NEC

Preliminary User's Manual

V850E/IF3, V850E/IG3

32-bit Single-Chip Microcontrollers

Hardware

V850E/IF3:
- μPD70F3451
- μPD70F3452

V850E/IG3:
- μPD70F3453
- μPD70F3454

Document No. U18279EJ1V0UD00 (1st edition)
Date Published August 2007 N

[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.

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

EEPROM is a trademark of NEC Electronics Corporation.

MINICUBE is a registered trademark of NEC Electronics Corporation in Japan and Germany or a trademark in the United States of America.

SuperFlash is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

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

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

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

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

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

(Note)

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

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

M5D 02. 11-1

PREFACE

Readers

This manual is intended for users who wish to understand the functions of the V850E/IF3 (μPD70F3451, 70F3452) and V850E/IG3 (μPD70F3453, 70F3454) and design application systems using the V850E/IF3 and V850E/IG3.

Purpose

This manual is intended to give users an understanding of the hardware functions of the V850E/IF3 and V850E/IG3 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 types
• CPU function	• Register set
• On-chip peripheral functions	• Instruction format and instruction set
• Flash memory programming	• Interrupts and exceptions
• Electrical specifications (target)	• Pipeline operation

How to Read This Manual

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

To understand the overall functions of the V850E/IF3 and V850E/IG3
→ Read this manual according to the **CONTENTS**.

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

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

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

To know the electrical specifications of the V850E/IF3 and V850E/IG3
→ See **CHAPTER 28 ELECTRICAL SPECIFICATIONS (TARGET)**.

The "yyy bit of the xxx register" is described as the "xxx.yyy bit" in this manual. Note with caution that even if "xxx.yyy" is described as is in a program, however, the compiler/assembler cannot recognize it correctly.

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:	Higher addresses on the top and lower addresses on the bottom	
Note:	Footnote for item marked with **Note** in the text	
Caution:	Information requiring particular attention	
Remark:	Supplementary information	
Numeric representation:	Binary	... xxxx or xxxxB
	Decimal	... xxxx
	Hexadecimal	... xxxxH
Prefix indicating power of 2 (address space, memory capacity):		
	K (kilo):	$2^{10} = 1,024$
	M (mega):	$2^{20} = 1,024^2$
	G (giga):	$2^{30} = 1,024^3$
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/IF3 and V850E/IG3

Document Name	Document No.
V850E1 Architecture User's Manual	U14559E
V850E/IF3, V850E/IG3 Hardware User's Manual	This manual
V850E/IF3, V850E/IG3 Sample Programs for Serial Communication (UARTA) Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Serial Communication (UARTB) Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Serial Communication (CSIB) Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Serial Communication (I^2C) Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for DMA Function Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Timer M Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Watchdog Timer Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Timer AA Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Timer AB Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Timer T Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Port Function Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Clock Generator Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Standby Function Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Interrupt Function Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for A/D Converters 0 and 1 Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for A/D Converter 2 Application Note	To be prepared
V850E/IF3, V850E/IG3 Sample Programs for Low-Voltage Detector (LVI) Function Application Note	To be prepared
V850E/IF3, V850E/IG3 6-Phase PWM Output Control by Timer AB, Timer Q Option, Timer AA, A/D Converters 0, 1 Application Note	To be prepared

Documents related to development tools (user's manuals)

Document Name		Document No.
QB-V850EIX3 In-Circuit Emulator		U18651E
QB-V850MINI On-Chip Debug Emulator		U17638E
QB-MINI2 On-Chip Debug Emulator with Programming Function		U18371E
CA850 Ver. 3.00 C Compiler Package	Operation	U17293E
	C Language	U17291E
	Assembly Language	U17292E
	Link Directives	U17294E
PM+ Ver. 6.30 Project Manager		U18416E
ID850QB Ver. 3.40 Integrated Debugger	Operation	U18604E
TW850 Ver. 2.00 Performance Analysis Tuning Tool		U17241E
SM+ System Simulator	Operation	U18601E
	User Open Interface	U18212E
RX850 Ver. 3.20 Real-Time OS	Basics	U13430E
	Installation	U17419E
	Technical	U13431E
	Task Debugger	U17420E
RX850 Pro Ver. 3.21 Real-Time OS	Basics	U18165E
	Installation	U17421E
	Technical	U13772E
	Task Debugger	U17422E
AZ850 Ver. 3.30 System Performance Analyzer		U17423E
PG-FP4 Flash Memory Programmer		U15260E

CONTENTS

CHAPTER 1 INTRODUCTION

The V850E/IF3 and V850E/IG3 are products of the NEC Electronics V850 single-chip microcontrollers. This chapter gives an outline of the V850E/IF3 and V850E/IG3.

1.1 Overview

The V850E/IF3 and V850E/IG3 are 32-bit single-chip microcontrollers that use the V850E1 CPU core and incorporates ROM/RAM and various peripheral functions such as DMA controller, timer/counter, watchdog timer, serial interfaces, A/D converter, and on-chip debug function.

In addition to high real-time response characteristics and 1-clock-pitch basic instructions, the V850E/IF3 and V850E/IG3 feature instructions such as multiply instructions, saturated operation instructions, and bit manipulation instructions realized by a hardware multiplier, as optimum instructions for digital servo control applications. Moreover, as a real-time control system, the V850E/IF3 and V850E/IG3 enable an extremely high cost-performance for applications such as motor inverter control.

Table 1-1 lists the V850E/IF3 and V850E/IG3 products.

Table 1-1. V850E/IF3, V850E/IG3 Product List

Function / Part Number		Package	ROM		RAM Size	Operating Frequency (MAX.)	Maskable Interrupt		Non-Maskable Interrupt
			Type	Size			External	Internal	
V850E/IF3	μPD70F3451	80GC	Flash memory	128 KB	8 KB	64 MHz	15	73	1
	μPD70F3452	80GC		256 KB	12 KB				
V850E/IG3	μPD70F3453	100GC		128 KB	8 KB		21	74	
	μPD70F3453	100GF		128 KB	8 KB				
	μPD70F3454	100GC		256 KB	12 KB				
	μPD70F3454	100GF		256 KB	12 KB				

Remarks 1. 80GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
100GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
100GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

2. The part numbers of the V850E/IG3 are shown as follows in this manual.

- GF versions
  μPD70F3453GF-GAS-AX, 70F3454GF-GAS-AX

Table 1-2 shows the differences in functions between the V850E/IF3 and V850E/IG3.

Table 1-2. Differences in Functions Between V850E/IF3 and V850E/IG3

Item		V850E/IF3	V850E/IG3
Port function	I/O	44	56
	Input	4	8
	On-chip pull-up resistor	44	56
Interrupt source		External interrupt: 15 Internal interrupt: 74	External interrupt: 21 Internal interrupt: 75
External bus function		None	Provided (μPD70F3454GC-8EA-A only)
Timers AA0 to AA4		Timer AA0 (without I/O) Timer AA1 (without I/O) Timer AA2 Timer AA3 (without I/O) Timer AA4	Timer AA0 (without I/O) Timer AA1 (without I/O) Timer AA2 Timer AA3 Timer AA4
Timers T0, T1		Timer T0 (without I/O) Timer T1	Timer T0 Timer T1
Motor control function	High-impedance output control pin	TOA2OFF TOB0OFF TOB1OFF	TOA2OFF TOA3OFF TOB0OFF TOB1OFF
A/D converter 2	Analog input	4 channels	8 channels
On-chip debug function	On-chip debug emulator	NIMICUBE2	NIMICUBE NIMICUBE2
Power supply for external pin		EV_{DD0}, EV_{DD1}	EV_{DD0} to EV_{DD2}
Package		80-pin plastic LQFP (14 × 14)	100-pin plastic LQFP (14 × 14) 100-pin plastic LQFP (14 × 20)

1.2 V850E/IF3

1.2.1 Features (V850E/IF3)

- Minimum instruction execution time:
  15.6 ns (at internal 64 MHz operation)

- General-purpose registers: 32 bits × 32

- CPU features:
  Signed multiplication (16 bits × 16 bits → 32 bits or 32 bits × 32 bits → 64 bits): 1 to 2 clocks
  Saturated operation instructions (with overflow/underflow detection function)
  32-bit shift instructions: 1 clock
  Bit manipulation instructions
  Load/store instructions with long/short format
  Signed load instructions

- Internal memory:

Part Number	Internal ROM	Internal RAM
μPD70F3451	128 KB (flash memory)	8 KB
μPD70F3452	256 KB (flash memory)	12 KB

- On-chip debug function: Supports MINICUBE®2.

- Interrupts/exceptions:
  Non-maskable interrupts: 1 source (external: none, internal: 1)
  Maskable interrupts: 88 sources (external: 15, internal: 73)
  Software exceptions: 32 sources
  Exception traps: 2 sources

- DMA controller: 4 channels
  Transfer unit: 8 bits/16 bits
  Maximum transfer count: 65536 (2^{16})
  Transfer type: 2-cycle
  Transfer modes: Single/single step/block
  Transfer targets: On-chip peripheral I/O ↔ Internal RAM
  On-chip peripheral I/O ↔ On-chip peripheral I/O
  Transfer request: On-chip peripheral I/O/software
  Next address setting function

- I/O lines: Total: 48 (Input ports: 4, I/O ports: 44)

○ Timer/counter function:
16-bit interval timer M (TMM): 4 channels
16-bit timer/event counter AA (TAA): 5 channels
16-bit timer/event counter AB (TAB): 2 channels
16-bit timer/event counter T (TMT): 2 channels
Motor control function (uses timer TAB: 2 channels (TAB0, TAB1), TAA: 2 channels (TAA0, TAA1))
- 16-bit accuracy 6-phase PWM function with deadtime: 2 channels
- High-impedance output control function
- A/D trigger generation by timer tuning operation function
- Arbitrary cycle setting function
- Arbitrary deadtime setting function

Watchdog timer: 1 channel

○ Serial interfaces:
Asynchronous serial interface A (UARTA)
Asynchronous serial interface B (UARTB)
Clocked serial interface B (CSIB)
I^2C bus interface (I^2C)

UARTA0/CSIB0:	1 channel
UARTA1/I^2C:	1 channel
UARTA2/CSIB1:	1 channel
UARTB/CSIB2:	1 channel

○ A/D converter:
12-bit resolution A/D converters (A/D converters 0 and 1): 5 channels + 5 channels (2 units)
- The one A/D converter 0 channel and three A/D converter 1 channels are provided with an operational amplifier for input level amplification and a comparator for overvoltage detection.

10-bit resolution A/D converter (A/D converter 2): 4 channels

○ Clock generator:
4 to 8 MHz resonator connectable (external clock input prohibited)
Multiplication function by PLL clock synthesizer (fixed to multiplication by eight, f_{XX} = 32 to 64 MHz)
CPU clock division function (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$)

○ Power-save function:
HALT/IDLE/STOP mode

○ Power-on-clear function:

○ Low-voltage detection function:

○ Package:
80-pin plastic LQFP (14 × 14)

○ Operation supply voltage:
$V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2}$ = 4.0 to 5.5 V (target)

1.2.2 Application fields (V850E/IF3)

- Consumer equipment (such as inverter air conditioners, washing machines, driers, refrigerators, etc.)
- Industrial equipment (such as motor control, general-purpose inverters, etc.)

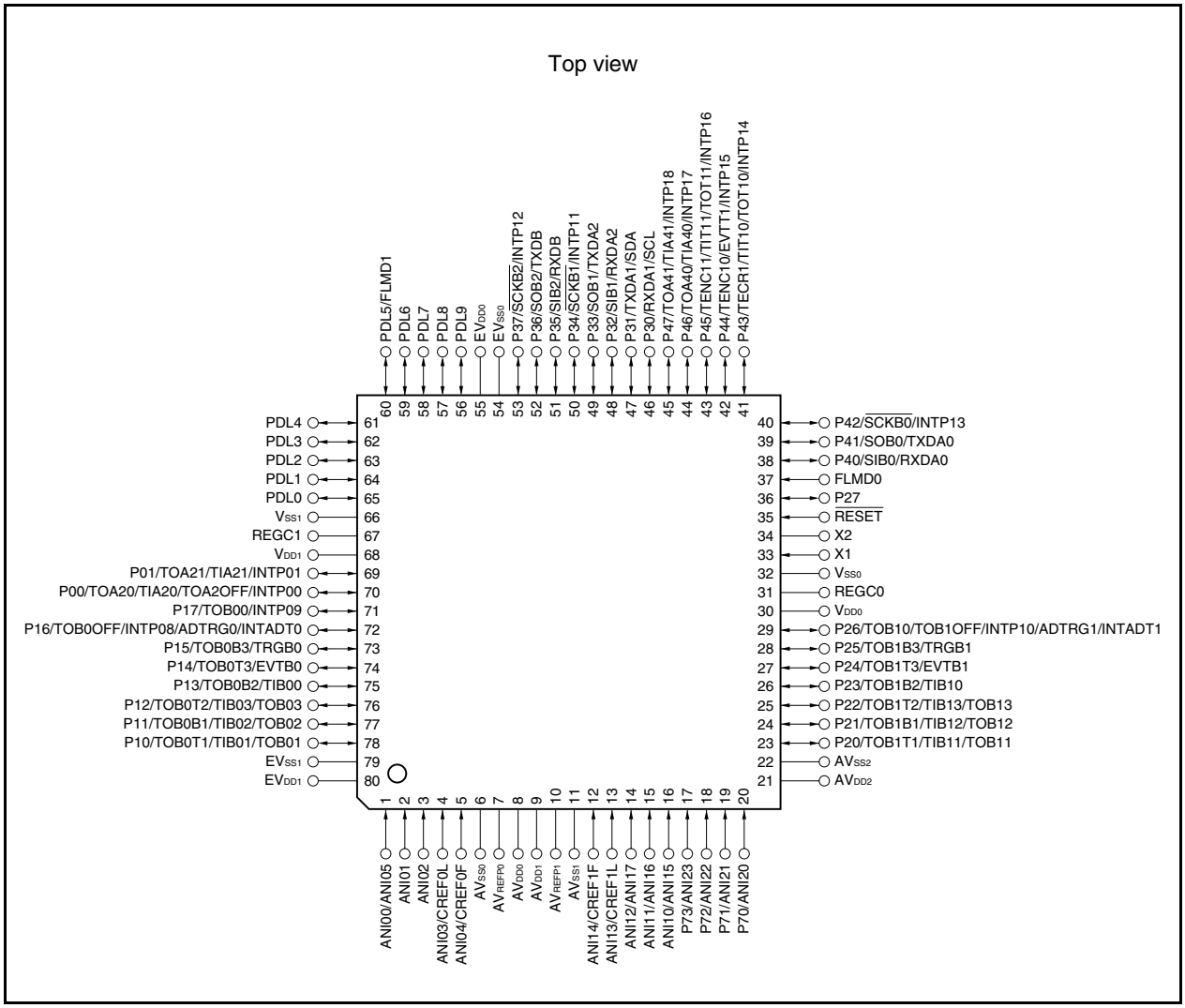
1.2.3 Ordering information (V850E/IF3)

Part Number	Package	Internal ROM
μPD70F3451GC-UBT-A	80-pin plastic LQFP (14 × 14)	Flash memory (128 KB)
μPD70F3452GC-UBT-A	80-pin plastic LQFP (14 × 14)	Flash memory (256 KB)

Remark The V850E/IF3 microcontrollers are lead-free products.

1.2.4 Pin configuration (V850E/IF3)

- **80-pin plastic LQFP (14 × 14)**
  μPD70F3451GC-UBT-A
  μPD70F3452GC-UBT-A

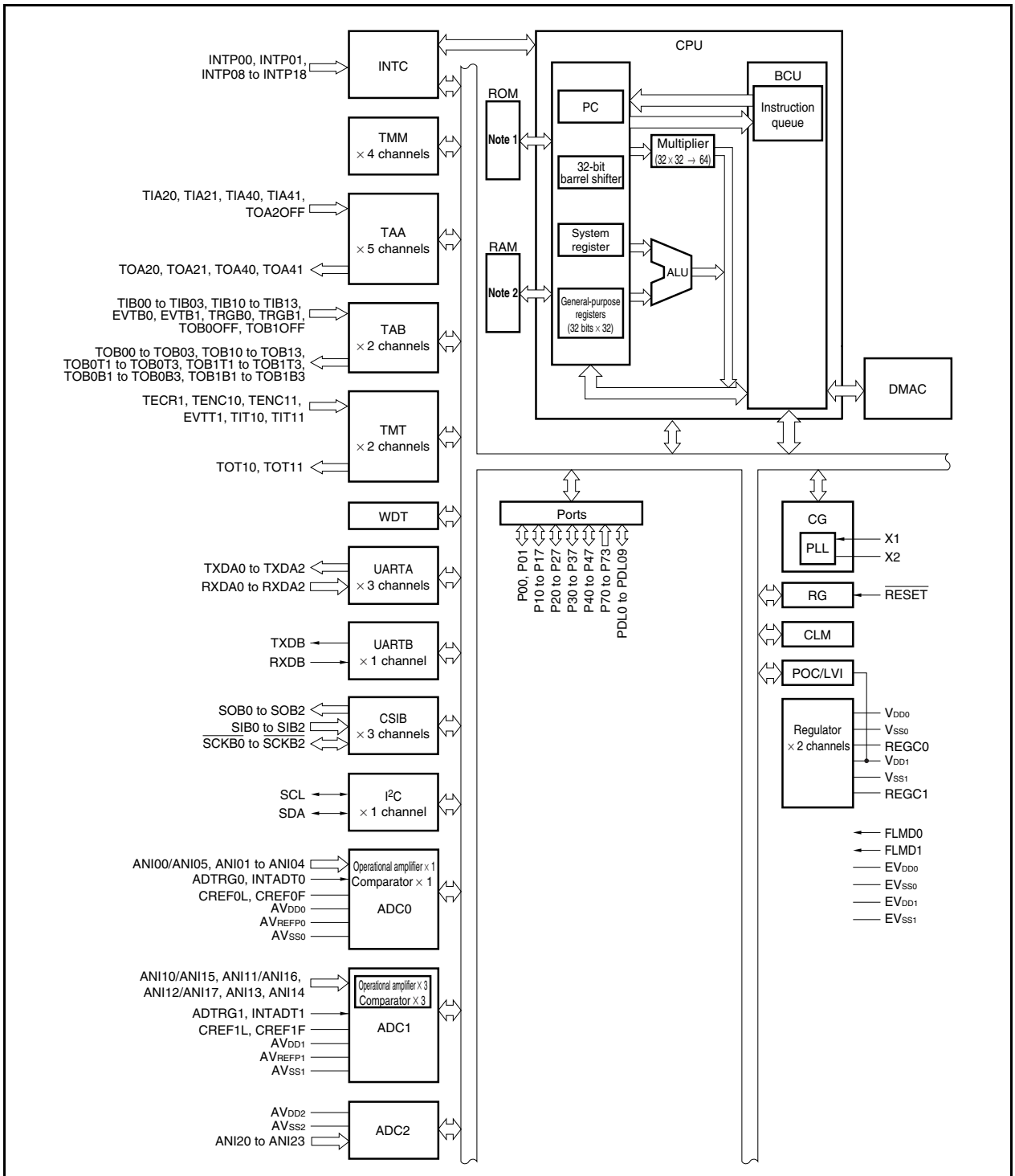


Pin Identification (V850E/IF3)

ADTRG0, ADTRG1:	A/D trigger input	SCL:	Serial clock
ANI00 to ANI05,		SDA:	Serial data
ANI10 to ANI17,		SIB0 to SIB2:	Serial input
ANI20 to ANI23:	Analog input	SOB0 to SOB2:	Serial output
AV_{DD0} to AV_{DD2}:	Analog power supply	TECR1:	Timer encoder clear input
AV_{REFP0}, AV_{REFP1}:	Analog reference voltage	TENC10, TENC11:	Timer encoder input
AV_{SS0} to AV_{SS2}:	Analog ground	TIA20, TIA21,	
CREF0F, CREF1F,		TIA40, TIA41,	
CREF0L, CREF1L:	Comparator reference voltage	TIB00 to TIB03,	
EV_{DD0}, EV_{DD1}:	Power supply for port	TIB10 to TIB13,	
EV_{SS0}, EV_{SS1}:	Ground for port	TIT10, TIT11:	Timer trigger input
EVTB0, EVTB1,		TOA20, TOA21,	
EVTT1:	Timer event count input	TOA40, TOA41,	
FLMD0, FLMD1:	Flash programming mode	TOB00 to TOB03,	
INTADT0, INTADT1,		TOB0B1 to TOB0B3,	
INTP00, INTP01,		TOB0T1 to TOB0T3,	
INTP08 to INTP18:	External interrupt input	TOB10 to TOB13,	
P00, P01:	Port 0	TOB1B1 to TOB1B3,	
P10 to P17:	Port 1	TOB1T1 to TOB1T3,	
P20 to P27:	Port 2	TOT10, TOT11:	Timer output
P30 to P37:	Port 3	TOA2OFF,	
P40 to P47:	Port 4	TOB0OFF, TOB1OFF:	Timer output off
P70 to P73:	Port 7	TRGB0, TRGB1:	Timer trigger input
PDL0 to PDL9:	Port DL	TXDA0 to TXDA2,	
REGC0, REGC1:	Regulator control	TXDB:	Transmit data
$\overline{\text{RESET}}$:	Reset	V_{DD0}, V_{DD1}:	Power supply
RXDA0 to RXDA2,		V_{SS0}, V_{SS1}:	Ground
RXDB:	Receive data	X1, X2:	Clock oscillator pin
$\overline{\text{SCKB0}}$ to $\overline{\text{SCKB2}}$:	Serial clock		

1.2.5 Function blocks (V850E/IF3)

(1) Internal block diagram



Notes 1. μPD70F3451: 128 KB (flash memory)
μPD70F3452: 256 KB (flash memory)

2. μPD70F3451: 8 KB
μPD70F3452: 12 KB

(2) Internal units

(a) CPU

The CPU uses five-stage pipeline control to enable single-clock execution of address calculations, arithmetic logic operations, data transfers, and almost all other instruction processing.

Other dedicated on-chip hardware, such as a multiplier (32 bits × 32 bits → 64 bits) and a barrel shifter (32 bits), help accelerate complex processing.

(b) Bus control unit (BCU)

The BCU controls the internal bus.

(i) DMA controller (DMAC)

This controller controls data transfer between on-chip peripheral I/O and internal RAM or on-chip peripheral I/O and on-chip peripheral I/O instead of the CPU.

The transfer type is two-cycle transfer, and single transfer, single-step transfer, and block transfer are used in transfer mode.

(c) ROM

This is flash memory that is mapped from address 00000000H.

During instruction fetch, ROM/flash memory can be accessed from the CPU in 1-clock cycles. The internal ROM capacity and area differ as follows depending on the product.

Part Number	Internal ROM Capacity	Internal ROM Area
μPD70F3451	128 KB (flash memory)	x0000000H to x001FFFFH
μPD70F3452	256 KB (flash memory)	x0000000H to x003FFFFH

(d) RAM

The internal RAM capacity and area differ as follows depending on the product.

During instruction fetch or data access, data can be accessed from the CPU in 1-clock cycles.

Part Number	Internal RAM Capacity	Internal RAM Area
μPD70F3451	8 KB	xFFFC000H to xFFFDFFFH
μPD70F3452	12 KB	xFFFC000H to xFFFEFFFH

(e) Interrupt controller (INTC)

This controller handles hardware interrupt requests (INTP00, INTP01, INTP08 to INTP18, INTADT0, INTADT1) from on-chip peripheral hardware and external hardware. Eight levels of interrupt priorities can be specified for these interrupt requests, and multiple-interrupt servicing control can be performed.

(f) Clock generator (CG)

The clock generator includes two basic operation modes: PLL mode (fixed to multiplication by eight) and clock-through mode. It generates four types of clocks (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$), and supplies one of them as the operating clock for the CPU (f_{CPU}).

(g) Timer/counter

The V850E/IF3 incorporates four 16-bit interval timer M (TMM) channels, five 16-bit timer/event counter AA (TAA) channels, two 16-bit timer/event counter AB (TAB) channels, and two 16-bit timer/event counter T (TMT) channels, and can measure pulse interval widths or frequency, enable an inverter function for motor control, and output a programmable pulse.

(h) Watchdog timer (WDT)

A watchdog timer is equipped to detect program loops, system abnormalities, etc.

It generates a non-maskable interrupt request signal (INTWDT) or internal reset signal (WDTRES) after an overflow occurs.

(i) Serial interface

The V850E/IF3 incorporates eight serial interface channels: for three asynchronous serial interface A (UARTA) channels, one asynchronous serial interface B (UARTB) channel, three clocked serial interface B (CSIB) channels, and one I^2C bus interface (I^2C) channel. Of these, UARTA0 and CSIB0, UARTA1 and I^2C, UARTA2 and CSIB1, and UARTB and CSIB2 share a pin.

For UARTA, data is transferred via the TXDAn and RXDAn pins (n = 0 to 2).

For UARTB, data is transferred via the TXDB and RXDB pins.

For CSIB, data is transferred via the SOBn, SIBn, and $\overline{\text{SCKBn}}$ pins (n = 0 to 2).

For I^2C, data is transferred via the SCL and SDA pins.

(j) A/D converter (ADC)

One channel is provided for each of the high-speed, high-resolution 12-bit A/D converters (ADC0, ADC1) (total of two channels), which have five analog input pins respectively, and one channel is provided for the 10-bit A/D converter (ADC2), which has four analog input pins.

Both one of the ADC0 channels and three of the ADC1 channels include an operational amplifier and a comparator so that these A/D converters can amplify an analog input voltage and detect overvoltage input.

(k) On-chip debug function

An on-chip debug function supporting MINICUBE2 can be used, so that a simple, inexpensive debug environment can be organized.

(l) Ports

As shown below, the following ports have general-purpose port functions and control pin functions.

Port	I/O	Alternate Function
Port 0	2-bit I/O	Timer/counter I/O, external interrupt input
Port 1	8-bit I/O	Timer/counter I/O, external trigger input of A/D converter 0, external interrupt input
Port 2	8-bit I/O	Timer/counter I/O, external trigger input of A/D converter 1, external interrupt input
Port 3	8-bit I/O	Serial interface I/O, external interrupt input
Port 4	8-bit I/O	Serial interface I/O, timer/counter I/O, external interrupt input
Port 7	4-bit input	A/D converter 2 input
Port DL	10-bit I/O	–

1.3 V850E/IG3

1.3.1 Features (V850E/IG3)

○ Minimum instruction execution time:
15.6 ns (at internal 64 MHz operation)

○ General-purpose registers: 32 bits × 32

○ CPU features: Signed multiplication (16 bits × 16 bits → 32 bits or 32 bits × 32 bits → 64 bits): 1 to 2 clocks
Saturated operation instructions (with overflow/underflow detection function)
32-bit shift instructions: 1 clock
Bit manipulation instructions
Load/store instructions with long/short format
Signed load instructions

○ Memory space (μPD70F3454GC-8EA-A only):
256 MB of linear address space (program/data sharing)
Chip select output function: 2 spaces
Memory block division function: 2 MB/block

- External bus interface: Multiplexed bus mode: 16-bit address bus
  8-bit/16-bit data bus
  Separate bus mode: 8-bit address bus
  8-bit/16-bit data bus
  8-bit/16-bit data bus sizing function
  External bus frequency switch function: 32/16 MHz
  Wait function
  • Programmable wait function
  • External wait function
  Idle state function
  Address setup wait function

○ Internal memory:

Part Number	Internal ROM	Internal RAM
μPD70F3453	128 KB (flash memory)	8 KB
μPD70F3454	256 KB (flash memory)	12 KB

○ On-chip debug function: Supports MINICUBE and MINICUBE2.

○ Interrupts/exceptions: Non-maskable interrupts: 1 source (external: none, internal: 1)
Maskable interrupts: 95 sources (external: 21, internal: 74)
Software exceptions: 32 sources
Exception traps: 2 sources

○ DMA controller: 4 channels
Transfer unit: 8 bits/16 bits
Maximum transfer count: 65,536 (2^{16})
Transfer type: 2-cycle
Transfer mode: Single/single step/block
Transfer target: On-chip peripheral I/O ↔ Internal RAM
On-chip peripheral I/O ↔ On-chip peripheral I/O
Transfer request: On-chip peripheral I/O/software
Next address setting function

○ I/O lines: Total: 64 (input ports: 8, I/O ports: 56)

○ Timer/counter function: 16-bit interval timer M (TMM): 4 channels
16-bit timer/event counter AA (TAA): 5 channels
16-bit timer/event counter AB (TAB): 2 channels
16-bit timer/event counter T (TMT): 2 channels
Motor control function (uses timer TAB: 2 channels (TAB0 and TAB1),
TAA: 2 channels (TAA0 and TAA1))
16-bit accuracy 6-phase PWM function with deadtime: 2 channels
High-impedance output control function
A/D trigger generation by timer tuning operation function
Arbitrary cycle setting function
Arbitrary deadtime setting function
Watchdog timer: 1 channel

○ Serial interfaces: Asynchronous serial interface A (UARTA)
Asynchronous serial interface B (UARTB)
Clocked serial interface B (CSIB)
I^2C bus interface (I^2C)

UARTA0/CSIB0: 1 channel
UARTA1/ I^2C: 1 channel
UARTA2/CSIB1: 1 channel
UARTB/CSIB2: 1 channel

○ A/D converter: 12-bit resolution A/D converters (A/D converters 0 and 1): 5 channels + 5 channels (2 units)
The one A/D converter 0 channel and three A/D converter 1 channels are provided with an operational amplifier for input level amplification and a comparator for overvoltage detection.
10-bit resolution A/D converter (A/D converter 2): 8 channels

○ Clock generator: 4 to 8 MHz resonator connectable (external clock input prohibited)
Multiplication function by PLL clock synthesizer (fixed to multiplication by eight, f_{XX} = 32 to 64 MHz)
CPU clock division function (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$)

- Power-save function: HALT/IDLE/STOP mode
- Power-on-clear function
- Low-voltage detection function
- Package: 100-pin plastic LQFP (fine pitch) (14 × 14)
  100-pin plastic LQFP (14 × 20)
- Operation supply voltage: $V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2}$ = 4.0 to 5.5 V (target)

1.3.2 Application fields (V850E/IG3)

- Consumer equipment (such as inverter air conditioners, washing machines, driers, refrigerators, etc.)
- Industrial equipment (such as motor control, general-purpose inverters, etc.)

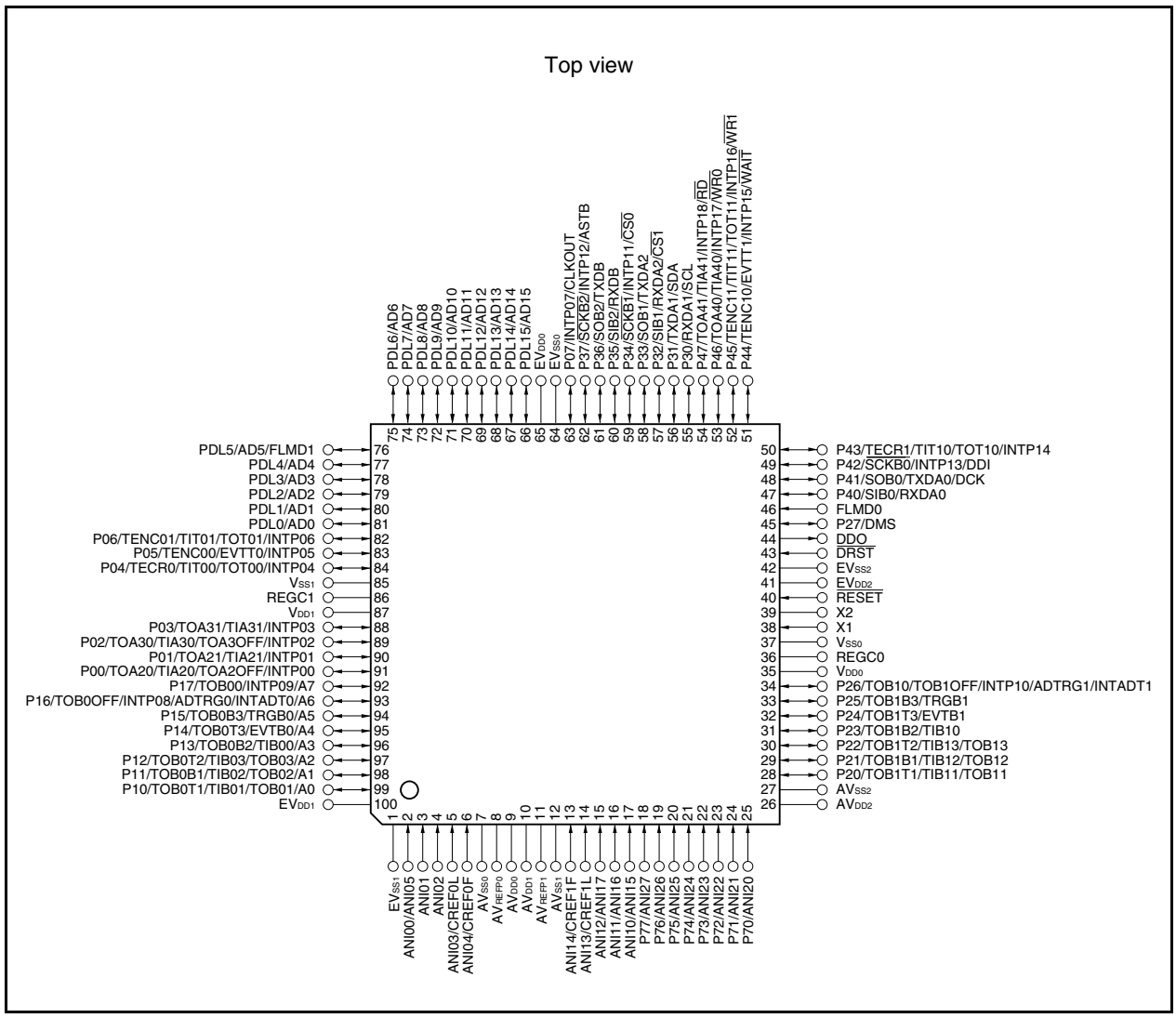
1.3.3 Ordering information (V850E/IG3)

Part Number	Package	Internal ROM
μPD70F3453GC-8EA-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory (128 KB)
μPD70F3453GF-GAS-AX	100-pin plastic LQFP (14 × 20)	Flash memory (128 KB)
μPD70F3454GC-8EA-A	100-pin plastic LQFP (fine pitch) (14 × 14)	Flash memory (256 KB)
μPD70F3454GF-GAS-AX	100-pin plastic LQFP (14 × 20)	Flash memory (256 KB)

Remark The V850E/IG3 microcontrollers are lead-free products.

1.3.4 Pin configuration (V850E/IG3)

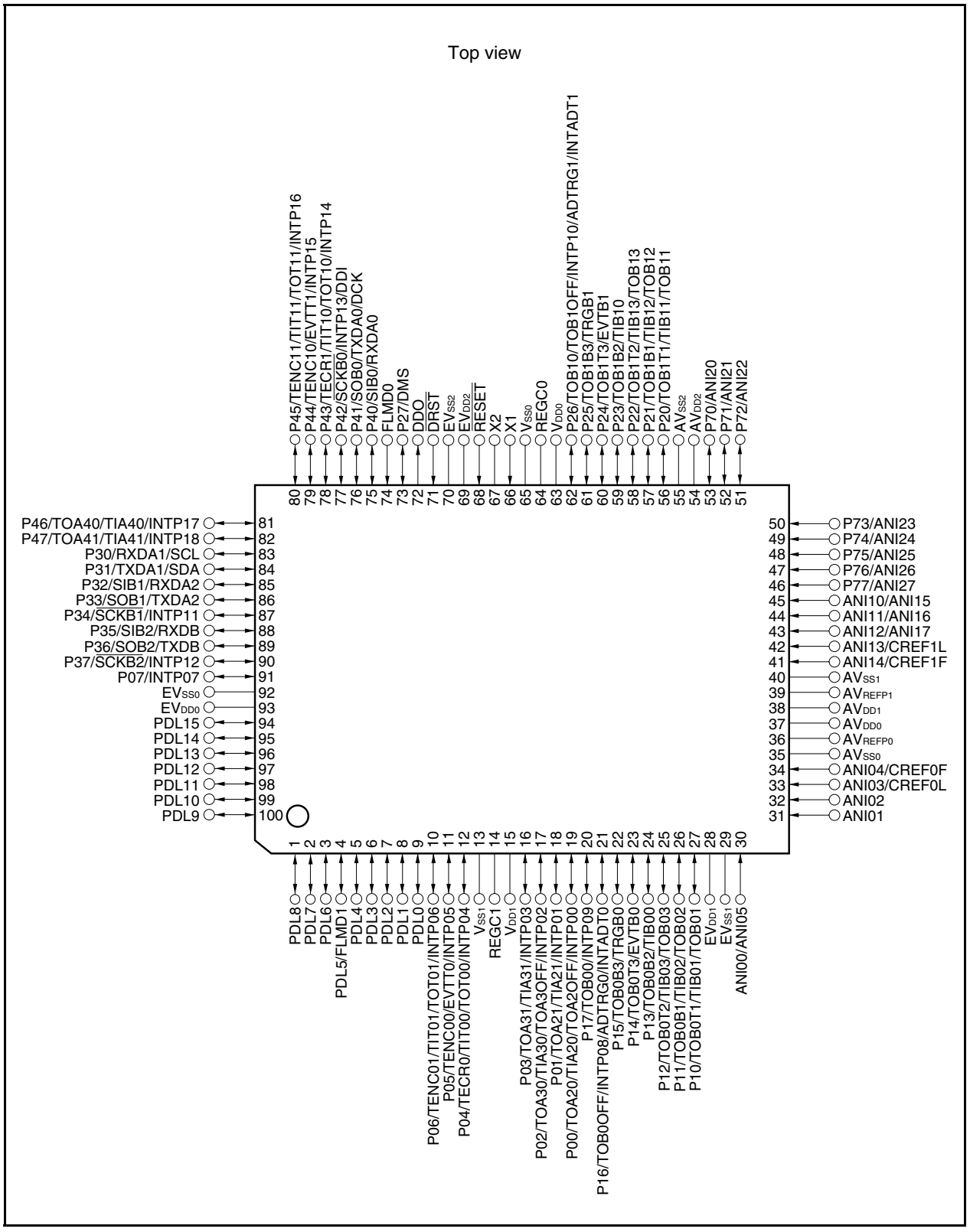
- **100-pin plastic LQFP (fine pitch) (14 × 14)**
  μPD70F3453GC-8EA-A
  μPD70F3454GC-8EA-A



- **100-pin plastic LQFP (14 × 20)**

  μPD70F3453GF-GAS-AX

  μPD70F3454GF-GAS-AX

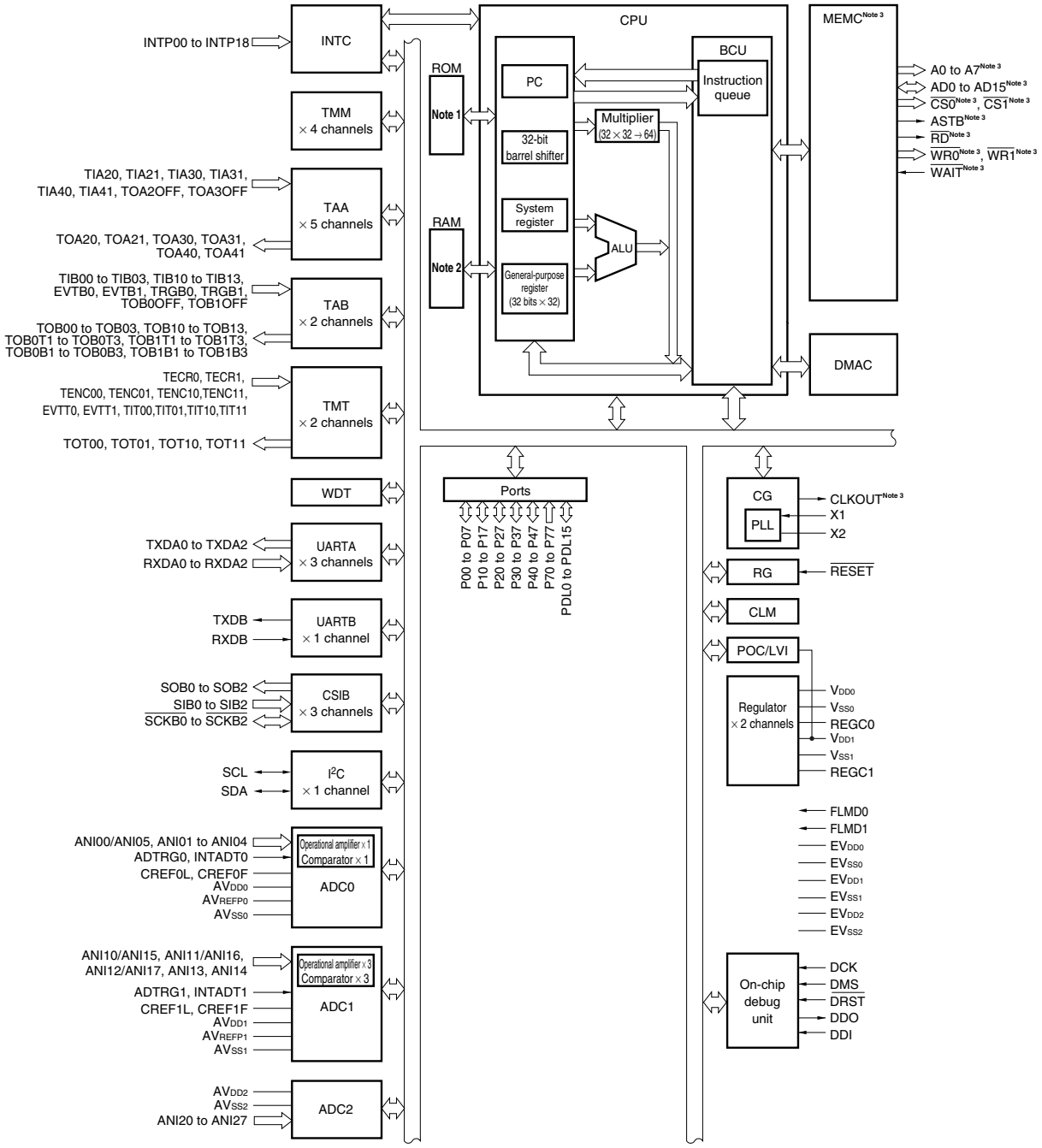


Pin Identification (V850E/IG3)

A0 to A7:	Address bus	$\overline{SCKB0}$ to $\overline{SCKB2}$:	Serial clock
AD0 to AD15:	Address/data bus	SCL:	Serial clock
ADTRG0, ADTRG1:	A/D trigger input	SDA:	Serial data
ANI00 to ANI05,		SIB0 to SIB2:	Serial input
ANI10 to ANI17,		SOB0 to SOB2:	Serial output
ANI20 to ANI27:	Analog input	TECR0, TECR1:	Timer encoder clear input
ASTB:	Address strobe	TENC00, TENC01,	
AV_{DD0} to AV_{DD2}:	Analog power supply	TENC10, TENC11:	Timer encoder input
AV_{REFP0}, AV_{REFP1}:	Analog reference voltage	TIA20, TIA21,	
AV_{SS0} to AV_{SS2}:	Analog ground	TIA30, TIA31,	
CLKOUT:	Clock output	TIA40, TIA41,	
CREF0F, CREF1F,		TIB00 to TIB03,	
CREF0L, CREF1L:	Comparator reference voltage	TIB10 to TIB13,	
$\overline{CS0}$, $\overline{CS1}$:	Chip select	TIT00, TIT01,	
DCK:	Debug clock	TIT10, TIT11:	Timer trigger input
DDI:	Debug data input	TOA20, TOA21,	
DDO:	Debug data output	TOA30, TOA31,	
DMS:	Debug mode select	TOA40, TOA41,	
$\overline{DRST}$:	Debug reset	TOB00 to TOB03,	
EV_{DD0} to EV_{DD2}:	Power supply for port	TOB0B1 to TOB0B3,	
EV_{SS0} to EV_{SS2}:	Ground for port	TOB0T1 to TOB0T3,	
EVTB0, EVTB1,		TOB10 to TOB13,	
EVTT0, EVTT1:	Timer event count input	TOB1B1 to TOB1B3,	
FLMD0, FLMD1:	Flash programming mode	TOB1T1 to TOB1T3,	
INTADT0, INTADT1,		TOT00, TOT01,	
INTP00 to INTP18:	External interrupt input	TOT10, TOT11:	Timer output
P00 to P07:	Port 0	TOA2OFF, TOA3OFF,	
P10 to P17:	Port 1	TOB0OFF, TOB1OFF:	Timer output off
P20 to P27:	Port 2	TRGB0, TRGB1:	Timer trigger input
P30 to P37:	Port 3	TXDA0 to TXDA2,	
P40 to P47:	Port 4	TXDB:	Transmit data
P70 to P77:	Port 7	V_{DD0}, V_{DD1}:	Power supply
PDL0 to PDL15:	Port DL	V_{SS0}, V_{SS1}:	Ground
$\overline{RD}$:	Read strobe	$\overline{WAIT}$:	Wait
REGC0, REGC1:	Regulator control	$\overline{WR0}$, $\overline{WR1}$:	Write strobe
$\overline{RESET}$:	Reset	X1, X2:	Clock oscillator pin
RXDA0 to RXDA2,			
RXDB:	Receive data		

1.3.5 Function blocks (V850E/IG3)

(1) Internal block diagram



Notes 1. μPD70F3453: 128 KB (flash memory)
μPD70F3454: 256 KB (flash memory)

2. μPD70F3453: 8 KB
μPD70F3454: 12 KB

3. μPD70F3454GC-8EA-A only

(2) Internal units

(a) CPU

The CPU uses five-stage pipeline control to enable single-clock execution of address calculations, arithmetic logic operations, data transfers, and almost all other instruction processing.

Other dedicated on-chip hardware, such as a multiplier (32 bits × 32 bits → 64 bits) and a barrel shifter (32 bits), help accelerate complex processing.

(b) Bus control unit (BCU)

The BCU starts the required external bus cycles in accordance with the physical address obtained by the CPU. If the CPU does not request the start of a bus cycle when an instruction is fetched from the external memory area (μPD70F3454GC-8EA-A only), the BCU generates a prefetch address and prefetches an instruction code. The prefetched instruction code is loaded to the internal instruction queue.

The BCU controls a memory controller (MEMC) and performs external memory access (μPD70F3454GC-8EA-A only).

(i) Memory controller (MEMC) (μPD70F3454GC-8EA-A only)

Controls access to SRAM, external ROM, and external I/O.

(ii) DMA controller (DMAC)

This controller controls data transfer between on-chip peripheral I/O and internal RAM or on-chip peripheral I/O and on-chip peripheral I/O instead of the CPU.

The transfer type is two-cycle transfer, and single transfer, single-step transfer, and block transfer are used in transfer mode.

(c) ROM

This is flash memory that is mapped from address 00000000H.

During instruction fetch, ROM/flash memory can be accessed from the CPU in 1-clock cycles. The internal ROM capacity and area differ as follows depending on the product.

Part Number	Internal ROM Capacity	Internal ROM Area
μPD70F3453	128 KB (flash memory)	x0000000H to x001FFFFH
μPD70F3454	256 KB (flash memory)	x0000000H to x003FFFFH

(d) RAM

The internal RAM capacity and area differ as follows depending on the product.

During instruction fetch or data access, data can be accessed from the CPU in 1-clock cycles.

Part Number	Internal RAM Capacity	Internal RAM Area
μPD70F3453	8 KB	xFFFC000H to xFFFDFFFH
μPD70F3454	12 KB	xFFFC000H to xFFFEFFFH

(e) Interrupt controller (INTC)

This controller handles hardware interrupt requests (INTP00 to INTP18, INTADT0, INTADT1) from on-chip peripheral hardware and external hardware. Eight levels of interrupt priorities can be specified for these interrupt requests, and multiple-interrupt servicing control can be performed.

(f) Clock generator (CG)

The clock generator includes two basic operation modes: PLL mode (fixed to multiplication by eight) and clock-through mode. It generates four types of clocks (f_{XX}, $f_{XX}/2$, $f_{XX}/4$, $f_{XX}/8$), and supplies one of them as the operating clock for the CPU (f_{CPU}).

(g) Timer/counter

The V850E/IG3 incorporates four 16-bit interval timer M (TMM) channels, five 16-bit timer/event counter AA (TAA) channels, two 16-bit timer/event counter AB (TAB) channels, and two 16-bit timer/event counter T (TMT) channels, and can measure pulse interval widths or frequency, enable an inverter function for motor control, and output a programmable pulse.

(h) Watchdog timer (WDT)

A watchdog timer is equipped to detect program loops, system abnormalities, etc.

It generates a non-maskable interrupt request signal (INTWDT) or internal reset signal (WDTRES) after an overflow occurs.

(i) Serial interface

The V850E/IG3 incorporates eight serial interface channels: for three asynchronous serial interface A (UARTA) channels, one asynchronous serial interface B (UARTB) channel, three clocked serial interface B (CSIB) channels, and one I^2C bus interface (I^2C) channel. Of these, UART0 and CSIB0, UARTA1 and I^2C, UART2 and CSIB1, and UARTB and CSIB2 share a pin.

For UARTA, data is transferred via the TXDAn and RXDAn pins (n = 0 to 2).

For UARTB, data is transferred via the TXDB and RXDB pins.

For CSIB, data is transferred via the SOBn, SIBn, and $\overline{SCKBn}$ pins (n = 0 to 2).

For I^2C, data is transferred via the SCL and SDA pins.

(j) A/D converter (ADC)

One channel is provided for each of the high-speed, high-resolution 12-bit A/D converters (ADC0, ADC1) (total of two channels), which have five analog input pins respectively, and one channel is provided for the 10-bit A/D converter (ADC2), which has eight analog input pins.

Both one of the ADC0 channels and three of the ADC1 channels include an operational amplifier and a comparator so that these A/D converters can amplify an analog input voltage and detect overvoltage input.

(k) On-chip debug function

An on-chip debug function supporting MINICUBE and MINICUBE2 can be used, so that a simple, inexpensive debug environment can be organized.

(l) Ports

As shown below, the following ports have general-purpose port functions and control pin functions.

Port	I/O	Alternate Function
Port 0	8-bit I/O	Timer/counter I/O, external interrupt input, external bus interface control signal output
Port 1	8-bit I/O	Timer/counter I/O, external bus interface control signal output, external trigger input of A/D converter 0, external interrupt input
Port 2	8-bit I/O	Timer/counter I/O, external trigger input of A/D converter 1, external interrupt input, debug input
Port 3	8-bit I/O	Serial interface I/O, external bus interface control signal output, external interrupt input
Port 4	8-bit I/O	Serial interface I/O, timer/counter I/O, debug input, external interrupt input, external bus interface control signal I/O
Port 7	8-bit input	A/D converter 2 input
Port DL	16-bit I/O	External bus interface control signal I/O

CHAPTER 2 PIN FUNCTIONS

The names and functions of the pins in the V850E/IF3 and V850E/IG3 are listed below. These pins can be divided into port pins and non-port pins according to their function.

2.1 List of Pin Functions

There are two power supplies for the I/O buffer of a pin: AV_{DD2} and EV_{DD0}, EV_{DD1}, EV_{DD2} (V850E/IG3 only). The relationship between each power supply and the pins is shown below.

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

(a) V850E/IF3

Power Supply	Corresponding Pins
AV_{DD2}	P70 to P73
EV_{DD0}, EV_{DD1}	P00, P01, P10 to P17, P20 to P27, P30 to P37, P40 to P47, PDL0 to PDL9, $\overline{\text{RESET}}$

(b) V850E/IG3

Power Supply	Corresponding Pins
AV_{DD2}	P70 to P77
EV_{DD0}, EV_{DD1}, EV_{DD2}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, PDL0 to PDL15, $\overline{\text{RESET}}$, DCK, DDI, DDO, DMS, $\overline{\text{DRST}}$

(1) Port pins

(1/3)

Pin Name	Pin No.			I/O	Function	Alternate-Function Pin
	IF3	IG3				
	GC	GC	GF			
P00	70	91	19	I/O	Port 0 V850E/IF3: 2-bit I/O port V850E/IG3: 8-bit I/O port Input data read/output data write is enabled in 1-bit units. An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function, and when TOA21 and TOA31 (V850E/IG3 only) pins (output pins of the alternate function) go into a high-impedance state).	TOA20/TIA20/TOA2OFF/INTP00
P01	69	90	18			TOA21/TIA21/INTP01
P02[Note 1]	–	89	17			TOA30[Note 1]/TIA30[Note 1]/TOA3OFF[Note 1]/INTP02[Note 1]
P03[Note 1]	–	88	16			TOA31[Note 1]/TIA31[Note 1]/INTP03[Note 1]
P04[Note 1]	–	84	12			TECR0[Note 1]/TIT00[Note 1]/TOT00[Note 1]/INTP04[Note 1]
P05[Note 1]	–	83	11			TENC00[Note 1]/EVTT0[Note 1]/INTP05[Note 1]
P06[Note 1]	–	82	10			TENC01[Note 1]/TIT01[Note 1]/TOT01[Note 1]/INTP06[Note 1]
P07[Note 1]	–	63	91			INTP07[Note 1]/CLKOUT[Note 2]
P10	78	99	27	I/O	Port 1 8-bit I/O port Input data read/output data write is enabled in 1-bit units. An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function, and when TOB0B1 to TOB0B3 and TOB0T1 to TOB0T3 pins (output pins of the alternate function) go into a high-impedance state).	TOB0T1/TIB01/TOB01/A0[Note 2]
P11	77	98	26			TOB0B1/TIB02/TOB02/A1[Note 2]
P12	76	97	25			TOB0T2/TIB03/TOB03/A2[Note 2]
P13	75	96	24			TOB0B2/TIB00/A3[Note 2]
P14	74	95	23			TOB0T3/EVTB0/A4[Note 2]
P15	73	94	22			TOB0B3/TRGB0/A5[Note 2]
P16	72	93	21			TOB0OFF/INTP08/ADTRG0/INTADT0/A6[Note 2]
P17	71	92	20			TOB00/INTP09/A7[Note 2]

Notes 1. V850E/IG3 only

2. μPD70F3454GC-8EA-A only

Remark IF3: V850E/IF3
IG3: V850E/IG3
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(2/3)

Pin Name	Pin No.			I/O	Function	Alternate-Function Pin
	IF3	IG3				
	GC	GC	GF			
P20	23	28	56	I/O	Port 2 8-bit I/O port Input data read/output data write is enabled in 1-bit units. An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function, and when TOB1B1 to TOB1B3 and TOB1T1 to TOB1T3 pins (output pins of the alternate function) go into a high-impedance state).	TOB1T1/TIB11/TOB11
P21	24	29	57			TOB1B1/TIB12/TOB12
P22	25	30	58			TOB1T2/TIB13/TOB13
P23	26	31	59			TOB1B2/TIB10
P24	27	32	60			TOB1T3/EVTB1
P25	28	33	61			TOB1B3/TRGB1
P26	29	34	62			TOB10/TOB1OFF/INTP10/ADTRG1/INTADT1
P27	36	45	73			DMS[Note 1]
P30	46	55	83	I/O	Port 3 8-bit I/O port Input data read/output data write is enabled in 1-bit units. An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function (including the $\overline{\text{SCKB1}}$ and $\overline{\text{SCKB2}}$ pins in the slave mode)). If the SCL or SDA pin is selected when the alternate function is to be used, N-ch open-drain output can be specified.	RXDA1/SCL
P31	47	56	84			TXDA1/SDA
P32	48	57	85			SIB1/RXDA2/$\overline{\text{CS1}}$[Note 2]
P33	49	58	86			SOB1/TXDA2
P34	50	59	87			$\overline{\text{SCKB1}}$/INTP11/$\overline{\text{CS0}}$[Note 2]
P35	51	60	88			SIB2/RXDB
P36	52	61	89			SOB2/TXDB
P37	53	62	90			$\overline{\text{SCKB2}}$/INTP12/ASTB[Note 2]
P40	38	47	75	I/O	Port 4 8-bit I/O port Input data read/output data write is enabled in 1-bit units. An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected when the pins are in the port mode and input mode, and when the pins function as input pins of the alternate function (including the $\overline{\text{SCKB0}}$ pin in the slave mode)).	SIB0/RXDA0
P41	39	48	76			SOB0/TXDA0/DCK[Note 1]
P42	40	49	77			$\overline{\text{SCKB0}}$/INTP13/DDI[Note 1]
P43	41	50	78			TECR1/TIT10/TOT10/INTP14
P44	42	51	79			TENC10/EVTT1/INTP15/$\overline{\text{WAIT}}$[Note 2]
P45	43	52	80			TENC11/TIT11/TOT11/INTP16/$\overline{\text{WR1}}$[Note 2]
P46	44	53	81			TOA40/TIA40/INTP17/$\overline{\text{WR0}}$[Note 2]
P47	45	54	82			TOA41/TIA41/INTP18/$\overline{\text{RD}}$[Note 2]

Notes 1. V850E/IG3 only

2. μPD70F3454GC-8EA-A only

Remark IF3: V850E/IF3
IG3: V850E/IG3
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(3/3)

Pin Name	Pin No.			I/O	Function	Alternate-Function Pin
	IF3	IG3				
	GC	GC	GF			
P70	20	25	53	Input	Port 7 V850E/IF3: 4-bit input port V850E/IG3: 8-bit input port	ANI20
P71	19	24	52			ANI21
P72	18	23	51			ANI22
P73	17	22	50			ANI23
P74Note 1	–	21	49			ANI24Note 1
P75Note 1	–	20	48			ANI25Note 1
P76Note 1	–	19	47			ANI26Note 1
P77Note 1	–	18	46			ANI27Note 1
PDL0	65	81	9	I/O	Port DL V850E/IF3: 10-bit I/O port V850E/IG3: 16-bit I/O port Input data read/output data write is enabled in 1-bit units. An on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected only when the pins are in the port mode and input mode).	AD0Note 2
PDL1	64	80	8			AD1Note 2
PDL2	63	79	7			AD2Note 2
PDL3	62	78	6			AD3Note 2
PDL4	61	77	5			AD4Note 2
PDL5	60	76	4			AD5Note 2/FLMD1
PDL6	59	75	3			AD6Note 2
PDL7	58	74	2			AD7Note 2
PDL8	57	73	1			AD8Note 2
PDL9	56	72	100			AD9Note 2
PDL10Note 1	–	71	99			AD10Note 2
PDL11Note 1	–	70	98			AD11Note 2
PDL12Note 1	–	69	97			AD12Note 2
PDL13Note 1	–	68	96			AD13Note 2
PDL14Note 1	–	67	95			AD14Note 2
PDL15Note 1	–	66	94			AD15Note 2

Notes 1. V850E/IG3 only

2. μPD70F3454GC-8EA-A only

Remark IF3: V850E/IF3
IG3: V850E/IG3
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(2) Non-port pins

(1/8)

Pin Name	Pin No.			I/O	Function	Alternate-Function Pin
	IF3	IG3				
	GC	GC	GF			
A0[Note 1]	–	99	–	Output	8-bit address bus for external memory	P10/TOB0T1/TIB01/TOB01
A1[Note 1]	–	98	–			P11/TOB0B1/TIB02/TOB02
A2[Note 1]	–	97	–			P12/TOB0T2/TIB03/TOB03
A3[Note 1]	–	96	–			P13/TOB0B2/TIB00
A4[Note 1]	–	95	–			P14/TOB0T3/EVTB0
A5[Note 1]	–	94	–			P15/TOB0B3/TRGB0
A6[Note 1]	–	93	–			P16/TOB0OFF/INTP08/ADTRG0/INTADT0
A7[Note 1]	–	92	–			P17/TOB00/INTP09
AD0[Note 1]	–	81	–	I/O	16-bit address/data bus for external memory	PDL0
AD1[Note 1]	–	80	–			PDL1
AD2[Note 1]	–	79	–			PDL2
AD3[Note 1]	–	78	–			PDL3
AD4[Note 1]	–	77	–			PDL4
AD5[Note 1]	–	76	–			FLMD1/PDL5
AD6[Note 1]	–	75	–			PDL6
AD7[Note 1]	–	74	–			PDL7
AD8[Note 1]	–	73	–			PDL8
AD9[Note 1]	–	72	–			PDL9
AD10[Note 1]	–	71	–			PDL10[Note 2]
AD11[Note 1]	–	70	–			PDL11[Note 2]
AD12[Note 1]	–	69	–			PDL12[Note 2]
AD13[Note 1]	–	68	–			PDL13[Note 2]
AD14[Note 1]	–	67	–			PDL14[Note 2]
AD15[Note 1]	–	66	–			PDL15[Note 2]
ADTRG0	72	93	21	Input	External trigger input for A/D converter 0	INTADT0/A6[Note 1]/P16/TOB0OFF/INTP08
ADTRG1	29	34	62	Input	External trigger input for A/D converter 1	INTADT1/P26/TOB10/TOB1OFF/INTP10

Notes 1. μPD70F3454GC-8EA-A only

2. V850E/IG3 only

Remark IF3: V850E/IF3
IG3: V850E/IG3
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(2/8)

Pin Name	Pin No.			I/O	Function	Alternate-Function Pin
	IF3	IG3				
	GC	GC	GF			
ANI00	1	2	30	Input	Analog input for A/D converter 0	ANI05
ANI01	2	3	31			–
ANI02	3	4	32			–
ANI03	4	5	33			CREF0L
ANI04	5	6	34			CREF0F
ANI05	1	2	30			ANI00
ANI10	16	17	45	Input	Analog input for A/D converter 1	ANI15
ANI11	15	16	44			ANI16
ANI12	14	15	43			ANI17
ANI13	13	14	42			CREF1L
ANI14	12	13	41			CREF1F
ANI15	16	17	45			ANI10
ANI16	15	16	44			ANI11
ANI17	14	15	43			ANI12
ANI20	20	25	53	Input	Analog input for A/D converter 2	P70
ANI21	19	24	52			P71
ANI22	18	23	51			P72
ANI23	17	22	50			P73
ANI24Note 1	–	21	49			P74Note 1
ANI25Note 1	–	20	48			P75Note 1
ANI26Note 1	–	19	47			P76Note 1
ANI27Note 1	–	18	46			P77Note 1
ASTBNote 2	–	62	–	Output	Address strobe output of external data bus	P37/$\overline{\text{SCKB2}}$/INTP12
AV_{DD0}	8	9	37	–	Positive power supply for A/D converter 0	–
AV_{DD1}	9	10	38	–	Positive power supply for A/D converter 1	–
AV_{DD2}	21	26	54	–	Positive power supply for A/D converter 2	–
AV_{REFP0}	7	8	36	–	Reference voltage input for A/D converter 0	–
AV_{REFP1}	10	11	39	–	Reference voltage input for A/D converter 1	–

Notes 1. V850E/IG3 only

2. μPD70F3454GC-8EA-A only

Remark IF3: V850E/IF3
IG3: V850E/IG3
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(3/8)

Pin Name	Pin No.			I/O	Function	Alternate-Function Pin
	IF3	IG3				
	GC	GC	GF			
AV_{SS0}	6	7	35	–	Ground potential for A/D converter 0	–
AV_{SS1}	11	12	40	–	Ground potential for A/D converter 1	–
AV_{SS2}	22	27	55	–	Ground potential for A/D converter 2	–
CLKOUT Note 1	–	63	–	Output	External bus clock output	P07 Note 2/INTP07 Note 2
CREF0L	4	5	33	–	Low range comparator reference voltage of A/D converter 0	ANI03
CREF1L	13	14	42	–	Low range comparator reference voltage of A/D converter 1	ANI13
CREF0F	5	6	34	–	Full range comparator reference voltage of A/D converter 0	ANI04
CREF1F	12	13	41	–	Full range comparator reference voltage of A/D converter 1	ANI14
$\overline{CS0}$ Note 1	–	59	–	Output	Chip select output	P34/$\overline{SCKB1}$/INTP11
$\overline{CS1}$ Note 1	–	57	–			P32/SIB1/RXDA2
DCK Note 2	–	48	76	Input	Debug clock input for on-chip debug emulator	P41/SOB0/TXDA0
DDI Note 2	–	49	77	Input	Debug data input for on-chip debug emulator	P42/$\overline{SCKB0}$/INTP13
DDO Note 2	–	44	72	Output	Debug data output for on-chip debug emulator	–
DMS Note 2	–	45	73	Input	Debug mode select for on-chip debug emulator	P27
$\overline{DRST}$ Note 2	–	43	71	Input	Debug reset input for on-chip debug emulator	–
EV_{DD0}	55	65	93	–	Positive power supply for external pin	–
EV_{DD1}	80	100	28			–
EV_{DD2} Note 2	–	41	69			–
EV_{SS0}	54	64	92	–	Ground potential for external pin	–
EV_{SS1}	79	1	29			–
EV_{SS2} Note 2	–	42	70			–
EVTB0	74	95	23	Input	External event count input of TAB0, TAB1	A4 Note 1/P14/TOB0T3
EVTB1	27	32	60			P24/TOB1T3

Notes 1. μPD70F3454GC-8EA-A only

2. V850E/IG3 only

Remark IF3: V850E/IF3
IG3: V850E/IG3
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(4/8)

Pin Name	Pin No.			I/O	Function	Alternate-Function Pin
	IF3	IG3				
	GC	GC	GF			
EVTT0[Note 1]	–	83	11	Input	External event count input of TMT0, TMT1/external trigger input	INTP05[Note 1]/P05[Note 1]/TENC00[Note 1]
EVTT1	42	51	79			INTP15/$\overline{\text{WAIT}}$[Note 2]/P44/TENC10
FLMD0	37	46	74	Input	Pin for setting flash memory programming mode	–
FLMD1	60	76	4			PDL5/AD5[Note 2]
INTADT0	72	93	21	Input	External maskable interrupt request input	P16/TOB0OFF/INTP08/ADTRG0/A6[Note 2]
INTADT1	29	34	62			P26/TOB10/TOB1OFF/INTP10/ADTRG1
INTP00	70	91	19			P00/TOA20/TIA20/TOA2OFF
INTP01	69	90	18			P01/TOA21/TIA21
INTP02[Note 1]	–	89	17			P02[Note 1]/TOA30[Note 1]/TIA30[Note 1]/TOA3OFF[Note 1]
INTP03[Note 1]	–	88	16			P03[Note 1]/TOA31[Note 1]/TIA31[Note 1]
INTP04[Note 1]	–	84	12			P04[Note 1]/TECR0[Note 1]/TIT00[Note 1]/TOT00[Note 1]
INTP05[Note 1]	–	83	11			P05[Note 1]/TENC00[Note 1]/EVTT0[Note 1]
INTP06[Note 1]	–	82	10			P06[Note 1]/TENC01[Note 1]/TIT01[Note 1]/TOT01[Note 1]
INTP07[Note 1]	–	63	91			CLKOUT[Note 2]/P07[Note 1]
INTP08	72	93	21			ADTRG0/INTADT0/A6[Note 2]/P16/TOB0OFF
INTP09	71	92	20			A7[Note 2]/P17/TOB00
INTP10	29	34	62			ADTRG1/INTADT1/P26/TOB10/TOB1OFF
INTP11	50	59	87			$\overline{\text{CS0}}$[Note 2]/P34/$\overline{\text{SCKB1}}$
INTP12	53	62	90			ASTB[Note 2]/P37/$\overline{\text{SCKB2}}$
INTP13	40	49	77			DDI[Note 1]/P42/$\overline{\text{SCKB0}}$
INTP14	41	50	78			P43/TECR1/TIT10/TOT10
INTP15	42	51	79			$\overline{\text{WAIT}}$[Note 2]/P44/TENC10/EVTT1
INTP16	43	52	80			$\overline{\text{WR1}}$[Note 2]/P45/TENC11/TIT11/TOT11
INTP17	44	53	81			$\overline{\text{WR0}}$[Note 2]/P46/TOA40/TIA40
INTP18	45	54	82			$\overline{\text{RD}}$[Note 2]/P47/TOA41/TIA41
$\overline{\text{RD}}$[Note 2]	–	54	–	Output	Read strobe output of external data bus	P47/TOA41/TIA41/INTP18
REGC0	31	36	64	–	Regulator output stabilization capacitance connection	–
REGC1	67	86	14			–

Notes 1. V850E/IG3 only

2. μPD70F3454GC-8EA-A only

Remark IF3: V850E/IF3
IG3: V850E/IG3
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(5/8)

Pin Name	Pin No.			I/O	Function	Alternate-Function Pin
	IF3	IG3				
	GC	GC	GF			
$\overline{\text{RESET}}$	35	40	68	Input	System reset input	–
RXDA0	38	47	75	Input	Serial receive data input of UARTA0 to UARTA2	P40/SIB0
RXDA1	46	55	83			SCL/P30
RXDA2	48	57	85			$\overline{\text{CS1}}$[Note 1]/P32/SIB1
RXDB	51	60	88	Input	Serial receive data input of UARTB0	P35/SIB2
$\overline{\text{SCKB0}}$	40	49	77	I/O	Serial clock I/O of CSIB0 to CSIB2	INTP13/DDI[Note 2]/P42
$\overline{\text{SCKB1}}$	50	59	87			INTP11/$\overline{\text{CS0}}$[Note 1]/P34
$\overline{\text{SCKB2}}$	53	62	90			INTP12/ASTB[Note 1]/P37
SCL	46	55	83	I/O	Serial clock I/O	P30/RXDA1
SDA	47	56	84	I/O	Serial transmit/receive data I/O	P31/TXDA1
SIB0	38	47	75	Input	Serial receive data input of CSIB0 to CSIB2	RXDA0/P40
SIB1	48	57	85			RXDA2/$\overline{\text{CS1}}$[Note 1]/P32
SIB2	51	60	88			RXDB/P35
SOB0	39	48	76	Output	Serial transmit data output of CSIB0 to CSIB2	TXDA0/DCK[Note 2]/P41
SOB1	49	58	86			TXDA2/P33
SOB2	52	61	89			TXDB/P36
TECR0[Note 2]	–	84	12	Input	Encoder clear input of TMT0, TMT1	TIT00[Note 2]/TOT00[Note 2]/INTP04[Note 2]/P04[Note 2]
TECR1	41	50	78			TIT10/TOT10/INTP14/P43
TENC00[Note 2]	–	83	11	Input	Encoder input of TMT0, TMT1	EVTT0[Note 2]/INTP05[Note 2]/P05[Note 2]
TENC01[Note 2]	–	82	10			TIT01[Note 2]/TOT01[Note 2]/INTP06[Note 2]/P06[Note 2]
TENC10	42	51	79			EVTT1/INTP15/$\overline{\text{WAIT}}$[Note 1]/P44
TENC11	43	52	80			TIT11/TOT11/INTP16/$\overline{\text{WR1}}$[Note 1]/P45
TIA20	70	91	19	Input	External event count input/external trigger input/capture trigger input of TAA2	TOA2OFF/INTP00/P00/TOA20
TIA21	69	90	18	Input	Capture trigger input of TAA2	INTP01/P01/TOA21
TIA30[Note 2]	–	89	17	Input	External event count input/external trigger input/capture trigger input of TAA3	TOA3OFF[Note 2]/INTP02[Note 2]/P02[Note 2]/TOA30[Note 2]
TIA31[Note 2]	–	88	16		Capture trigger input of TAA3	INTP03[Note 2]/P03[Note 2]/TOA31[Note 2]
TIA40	44	53	81	Input	External event count input/external trigger input/capture trigger input of TAA4	INTP17/$\overline{\text{WR0}}$[Note 1]/P46/TOA40
TIA41	45	54	82	Input	Capture trigger input of TAA4	INTP18/$\overline{\text{RD}}$[Note 1]/P47/TOA41

Notes 1. μPD70F3454GC-8EA-A only

2. V850E/IG3 only

Remark IF3: V850E/IF3
IG3: V850E/IG3
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(6/8)

Pin Name	Pin No.			I/O	Function	Alternate-Function Pin
	IF3	IG3				
	GC	GC	GF			
TIB00	75	96	24	Input	Capture trigger input of TAB0, TAB1	A3[Note 1]/P13/TOB0B2
TIB01	78	99	27			TOB01/A0[Note 1]/P10/TOB0T1
TIB02	77	98	26			TOB02/A1[Note 1]/P11/TOB0B1
TIB03	76	97	25			TOB03/A2[Note 1]/P12/TOB0T2
TIB10	26	31	59			P23/TOB1B2
TIB11	23	28	56			TOB11/P20/TOB1T1
TIB12	24	29	57			TOB12/P21/TOB1B1
TIB13	25	30	58			TOB13/P22/TOB1T2
TIT00[Note 2]	–	84	12	Input	Capture trigger input of TIT0	TOT00[Note 2]/INTP04[Note 2]/P04[Note 2]/TECR0[Note 2]
TIT01[Note 2]	–	82	10			TOT01[Note 2]/INTP06[Note 2]/P06[Note 2]/TENC01[Note 2]
TIT10	41	50	78	Input	Capture trigger input of TIT1	TOT10/INTP14/P43/TECR1
TIT11	43	52	80			TOT11/INTP16/$\overline{\text{WR1}}$[Note 1]/P45/TENC11
TOA20	70	91	19	Output	Timer output of TAA2	TIA20/TOA2OFF/INTP00/P00
TOA21	69	90	18			TIA21/INTP01/P01
TOA2OFF	70	91	19	Input	High-impedance output control signal input	INTP00/P00/TOA20/TIA20
TOA30[Note 2]	–	89	17	Output	Timer output of TAA3	TIA30[Note 2]/TOA3OFF[Note 2]/INTP02[Note 2]/P02[Note 2]
TOA31[Note 2]	–	88	16			TIA31[Note 2]/INTP03[Note 2]/P03[Note 2]
TOA3OFF[Note 2]	–	89	17	Input	High-impedance output control signal input	INTP02[Note 2]/P02[Note 2]/TOA30[Note 2]/TIA30[Note 2]
TOA40	44	53	81	Output	Timer output of TAA4	TIA40/INTP17/$\overline{\text{WR0}}$[Note 1]/P46
TOA41	45	54	82			TIA41/INTP18/$\overline{\text{RD}}$[Note 1]/P47
TOB00	71	92	20	Output	Timer output of TAB0	INTP09/A7[Note 1]/P17
TOB01	78	99	27			A0[Note 1]/P10/TOB0T1/TIB01
TOB02	77	98	26			A1[Note 1]/P11/TOB0B1/TIB02
TOB03	76	97	25			A2[Note 1]/P12/TOB0T2/TIB03
TOB0B1	77	98	26	Output	Pulse signal output for 6-phase PWM low arm of TAB0	TIB02/TOB02/A1[Note 1]/P11
TOB0B2	75	96	24			TIB00/A3[Note 1]/P13
TOB0B3	73	94	22			TRGB0/A5[Note 1]/P15

Notes 1. μPD70F3454GC-8EA-A only

2. V850E/IG3 only

Remark IF3: V850E/IF3

IG3: V850E/IG3

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(7/8)

Pin Name	Pin No.			I/O	Function	Alternate-Function Pin
	IF3	IG3				
	GC	GC	GF			
TOB0OFF	72	93	21	Input	6-phase PWM high-impedance output control signal input of TAB0	INTP08/ADTRG0/INTADT0/A6[Note 1]/P16
TOB0T1	78	99	27	Output	Pulse signal output for 6-phase PWM high arm of TAB0	TIB01/TOB01/A0[Note 1]/P10
TOB0T2	76	97	25			TIB03/TOB03/A2[Note 1]/P12
TOB0T3	74	95	23			EVTB0/A4[Note 1]/P14
TOB10	29	34	62	Output	Timer output of TAB1	TOB1OFF/INTP10/ADTRG1/INTADT1/P26
TOB11	23	28	56			P20/TOB1T1/TIB11
TOB12	24	29	57			P21/TOB1B1/TIB12
TOB13	25	30	58			P22/TOB1T2/TIB13
TOB1B1	24	29	57	Output	Pulse signal output for 6-phase PWM low arm of TAB1	TIB12/TOB12/P21
TOB1B2	26	31	59			TIB10/P23
TOB1B3	28	33	61			TRGB1/P25
TOB1OFF	29	34	62	Input	6-phase PWM high-impedance output control signal input of TAB1	INTP10/ADTRG1/INTADT1/P26/TOB10
TOB1T1	23	28	56	Output	Pulse signal output for 6-phase PWM high arm of TAB1	TIB11/TOB11/P20
TOB1T2	25	30	58			TIB13/TOB13/P22
TOB1T3	27	32	60			EVTB1/P24
TOT00[Note 2]	–	84	12	Output	Timer output of TMT0, TMT1	INTP04[Note 2]/P04[Note 2]/TECR0[Note 2]/TIT00[Note 2]
TOT01[Note 2]	–	82	10			INTP06[Note 2]/P06[Note 2]/TENC01[Note 2]/TIT01[Note 2]
TOT10	41	50	78			INTP14/P43/TECR1/TIT10
TOT11	43	52	80			INTP16/$\overline{\text{WR1}}$[Note 1]/P45/TENC11/TIT11
TRGB0	73	94	22	Input	External trigger input of TAB0, TAB1	A5[Note 1]/P15/TOB0B3
TRGB1	28	33	61			P25/TOB1B3
TXDA0	39	48	76	Output	Serial transmit data output of UARTA0 to UARTA2	DCK[Note 2]/P41/SOB0
TXDA1	47	56	84			SDA/P31
TXDA2	49	58	86			P33/SOB1
TXDB	52	61	89	Output	Serial transmit data output of UARTB0	P36/SOB2

Notes 1. μPD70F3454GC-8EA-A only

2. V850E/IG3 only

Remark IF3: V850E/IF3
IG3: V850E/IG3
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(8/8)

Pin Name	Pin No.			I/O	Function	Alternate-Function Pin
	IF3	IG3				
	GC	GC	GF			
V_{DD0}	30	35	63	–	Positive power supply for internal unit	–
V_{DD1}	68	87	15			–
V_{SS0}	32	37	65	–	Ground potential for internal unit	–
V_{SS1}	66	85	13			–
$\overline{WAIT}$ Note	–	51	–	Input	External wait request input	P44/TENC10/EVTT1/INTP15
$\overline{WR0}$ Note	–	53	–	Output	Write strobe output of external data bus	P46/TOA40/TIA40/INTP17
$\overline{WR1}$ Note	–	52	–			P45/TENC11/TIT11/TOT11/INTP16
X1	33	38	66	Input	Resonator connection pin for system clock	–
X2	34	39	67	–		–

Note μPD70F3454GC-8EA-A only

Remark IF3: V850E/IF3
IG3: V850E/IG3
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

2.2 Pin Status

The operation statuses of pins in the various operation modes are described below.

Table 2-2. Pin Operation Status in Operation Modes

Pin \ Operating Status	Reset	HALT Mode/ During DMA Transfer	IDLE Mode/ STOP Mode	Idle State
AD0 to AD15[Note 1] (PDL0 to PDL15)	Hi-Z[Note 2]	Operating	Hi-Z	Held
A0 to A7[Note 1] (P10 to P17)	Hi-Z[Note 2]	Operating	Hi-Z	Held
$\overline{CS0}$[Note 1], $\overline{CS1}$[Note 1] (P34, P32)	Hi-Z[Note 2]	Operating	H	Held
$\overline{WR0}$[Note 1], $\overline{WR1}$[Note 1] (P46, P45)	Hi-Z[Note 2]	Operating	H	H
$\overline{RD}$[Note 1] (P47)	Hi-Z[Note 2]	Operating	H	H
ASTB[Note 1] (P37)	Hi-Z[Note 2]	Operating	H	H
$\overline{WAIT}$[Note 1] (P44)	Hi-Z[Note 2]	Operating	–	–
CLKOUT[Note 1] (P07)	Hi-Z[Note 2]	Operating	Held	Operating

Notes 1. μPD70F3454GC-8EA-A only

2. Since the bus control pin is also used as a port pin, it is initialized to the input mode (port mode) after reset.

Remark Hi-Z: High impedance
Held: The state during the immediately preceding external bus cycle is held.
H: High-level output
–: Input without sampling (not acknowledged)

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

It is recommended that 1 to 10 kΩ resistors be used when connecting to AV_{SS2}, EV_{DD0}, EV_{DD1}, EV_{DD2} (V850E/IG3 only) or EV_{SS0}, EV_{SS1}, EV_{SS2} (V850E/IG3 only) via resistors.

(1/3)

Pin	Alternate-Function Pin Name	Pin No.			I/O Circuit Type	Recommended Connection
		IF3	IG3			
		GC	GC	GF		
P00	TOA20/TIA20/TOA2OFF/INTP00	70	91	19	5-AH	Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{DD2}[Note 1] or EV_{SS0}, EV_{SS1}, EV_{SS2}[Note 1] via a resistor. Output: Leave open.
P01	TOA21/TIA21/INTP01	69	90	18		
P02[Note 1]	TOA30[Note 1]/TIA30[Note 1]/TOA3OFF[Note 1]/INTP02[Note 1]	–	89	17		
P03[Note 1]	TOA31[Note 1]/TIA31[Note 1]/INTP03[Note 1]	–	88	16		
P04[Note 1]	TECR0[Note 1]/TIT00[Note 1]/TOT00[Note 1]/INTP04[Note 1]	–	84	12		
P05[Note 1]	TENC00[Note 1]/EVTT0[Note 1]/INTP05[Note 1]	–	83	11		
P06[Note 1]	TENC01[Note 1]/TIT01[Note 1]/TOT01[Note 1]/INTP06[Note 1]	–	82	10		
P07[Note 1]	INTP07[Note 1]/CLKOUT[Note 2]	–	63	91		
P10	TOB0T1/TIB01/TOB01/A0[Note 2]	78	99	27		
P11	TOB0B1/TIB02/TOB02/A1[Note 2]	77	98	26		
P12	TOB0T2/TIB03/TOB03/A2[Note 2]	76	97	25		
P13	TOB0B2/TIB00/A3[Note 2]	75	96	24		
P14	TOB0T3/EVTB0/A4[Note 2]	74	95	23		
P15	TOB0B3/TRGB0/A5[Note 2]	73	94	22		
P16	TOB0OFF/INTP08/ADTRG0/INTADT0/A6[Note 2]	72	93	21		
P17	TOB00/INTP09/A7[Note 2]	71	92	20		
P20	TOB1T1/TIB11/TOB11	23	28	56		
P21	TOB1B1/TIB12/TOB12	24	29	57		
P22	TOB1T2/TIB13/TOB13	25	30	58		
P23	TOB1B2/TIB10	26	31	59		
P24	TOB1T3/EVTB1	27	32	60		
P25	TOB1B3/TRGB1	28	33	61		
P26	TOB10/TOB1OFF/INTP10/ADTRG1/INTADT1	29	34	62		
P27	DMS[Note 1]	36	45	73		

Notes 1. V850E/IG3 only

2. μPD70F3454GC-8EA-A only

Remark IF3: V850E/IF3
IG3: V850E/IG3
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(2/3)

Pin	Alternate-Function Pin Name	Pin No.			I/O Circuit Type	Recommended Connection
		IF3	IG3			
		GC	GC	GF		
P30	RXDA1/SCL	46	55	83	5-AH	Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{DD2}[Note 2] or EV_{SS0}, EV_{SS1}, EV_{SS2}[Note 2] via a resistor. Output: Leave open.
P31	TXDA1/SDA	47	56	84		
P32	SIB1/RXDA2/$\overline{CS1}$[Note 1]	48	57	85		
P33	SOB1/TXDA2	49	58	86	5-AG	
P34	$\overline{SCKB1}$/INTP11/$\overline{CS0}$[Note 1]	50	59	87	5-AH	
P35	SIB2/RXDB	51	60	88		
P36	SOB2/TXDB	52	61	89	5-AG	
P37	$\overline{SCKB2}$/INTP12/ASTB[Note 1]	53	62	90	5-AH	
P40	SIB0/RXDA0	38	47	75		
P41	SOB0/TXDA0/DCK[Note 2]	39	48	76	5-AG	
P42	$\overline{SCKB0}$/INTP13/DDI[Note 2]	40	49	77	5-AH	
P43	TECR1/TIT10/TOT10/INTP14	41	50	78		
P44	TENC10/EVTT1/INTP15/$\overline{WAIT}$[Note 1]	42	51	79		
P45	TENC11/TIT11/TOT11/INTP16/$\overline{WR1}$[Note 1]	43	52	80		
P46	TOA40/TIA40/INTP17/$\overline{WR0}$[Note 1]	44	53	81		
P47	TOA41/TIA41/INTP18/$\overline{RD}$[Note 1]	45	54	82		
P70	ANI20	20	25	53	11-G	Independently connect to AV_{SS2} via a resistor.
P71	ANI21	19	24	52		
P72	ANI22	18	23	51		
P73	ANI23	17	22	50		
P74[Note 2]	ANI24[Note 2]	–	21	49		
P75[Note 2]	ANI25[Note 2]	–	20	48		
P76[Note 2]	ANI26[Note 2]	–	19	47		
P77[Note 2]	ANI27[Note 2]	–	18	46		
PDL0	AD0[Note 1]	65	81	9	5-AG	Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{DD2}[Note 2] or EV_{SS0}, EV_{SS1}, EV_{SS2}[Note 2] via a resistor. Output: Leave open.
PDL1	AD1[Note 1]	64	80	8		
PDL2	AD2[Note 1]	63	79	7		
PDL3	AD3[Note 1]	62	78	6		

Notes 1. μPD70F3454GC-8EA-A only

2. V850E/IG3 only

Remark IF3: V850E/IF3
IG3: V850E/IG3
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(3/3)

Pin	Alternate-Function Pin Name	Pin No.			I/O Circuit Type	Recommended Connection
		IF3	IG3			
		GC	GC	GF		
PDL4	AD4[Note 1]	61	77	5	5-AG	Input: Independently connect to EV_{DD0}, EV_{DD1}, EV_{DD2}[Note 2] or EV_{SS0}, EV_{SS1}, EV_{SS2}[Note 2] via a resistor. Output: Leave open.
PDL5	AD5[Note 1]/FLMD1	60	76	4		
PDL6	AD6[Note 1]	59	75	3		
PDL7	AD7[Note 1]	58	74	2		
PDL8	AD8[Note 1]	57	73	1		
PDL9	AD9[Note 1]	56	72	100		
PDL10[Note 2]	AD10[Note 1]	–	71	99		
PDL11[Note 2]	AD11[Note 1]	–	70	98		
PDL12[Note 2]	AD12[Note 1]	–	69	97		
PDL13[Note 2]	AD13[Note 1]	–	68	96		
PDL14[Note 2]	AD14[Note 1]	–	67	95		
PDL15[Note 2]	AD15[Note 1]	–	66	94		
ANI00	ANI05	1	2	30	7-C	Connect to AV_{SS0} or AV_{SS1}.
ANI01	–	2	3	31		
ANI02	–	3	4	32		
ANI03	CREF0L	4	5	33		
ANI04	CREF0F	5	6	34		
ANI10	ANI15	16	17	45		
ANI11	ANI16	15	16	44		
ANI12	ANI17	14	15	43		
ANI13	CREF1L	13	14	42		
ANI14	CREF1F	12	13	41		
DDO[Note 2]	–	–	44	72	3-C	Leave open (output when $\overline{DRST}$ is high-level).
$\overline{DRST}$[Note 2]	–	–	43	71	2-M	Leave open (on-chip pull-down resistor).
FLMD0	–	37	46	74	2	–
$\overline{RESET}$	–	35	40	68		Pull this pin up when the power-on-clear circuit (POC) is used.

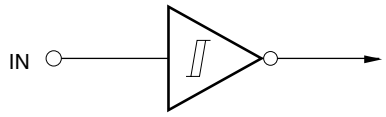
Notes 1. μPD70F3454GC-8EA-A only

2. V850E/IG3 only

Remark IF3: V850E/IF3
IG3: V850E/IG3
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

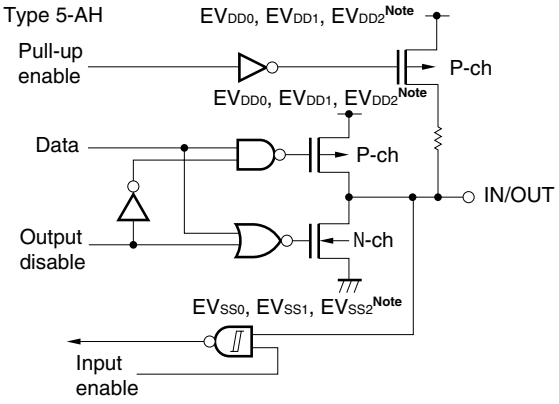
2.4 Pin I/O Circuits

Type 2

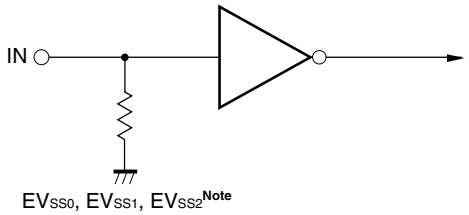


Schmitt-triggered input with hysteresis characteristics

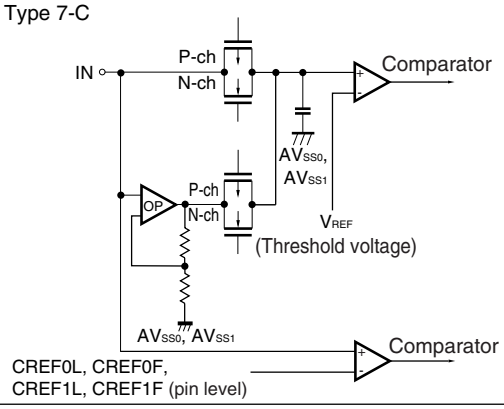
Type 5-AH



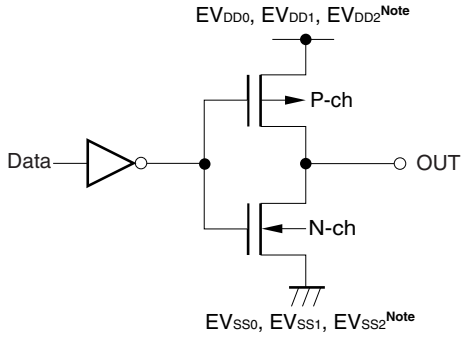
Type 2-M



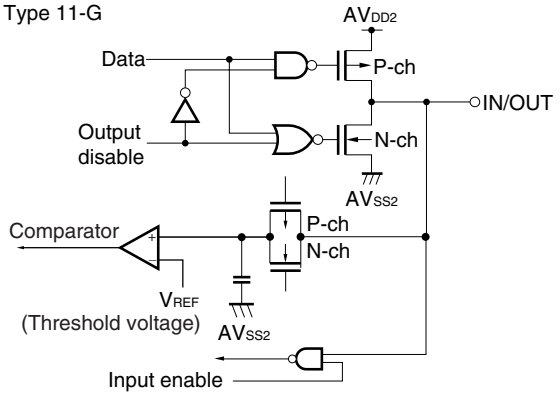
Type 7-C



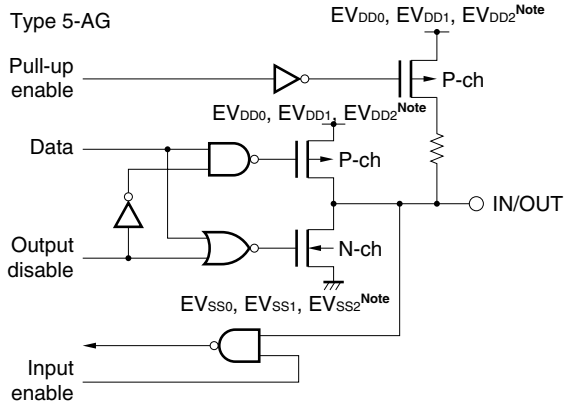
Type 3-C



Type 11-G



Type 5-AG



Note V850E/IG3 only

CHAPTER 3 CPU FUNCTION

The CPU of the V850E/IF3 and V850E/IG3 is based on RISC architecture and executes almost all the instructions in one clock cycle using 5-stage pipeline control.

3.1 Features

- Minimum instruction execution time: 15.6 ns (@ 64 MHz internal operation)
- Thirty-two 32-bit general-purpose registers
- Internal 32-bit architecture
- Five-stage pipeline control
- Multiply/divide instructions
- Saturated operation instructions
- One-clock 32-bit shift instruction
- Load/store instruction with long/short instruction format
- Four types of bit manipulation instructions
  - SET1
  - CLR1
  - NOT1
  - TST1

3.2 CPU Register Set

The registers of the V850E/IF3 and V850E/IG3 can be classified into two categories: a general-purpose program register set and a dedicated system register set. All the registers have a 32-bit width.

For details, refer to **V850E1 Architecture User's Manual**.

Figure 3-1. CPU Register Set

(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 (r0 to r31)

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 the registers after the registers have been used. r2 may be used by the real-time OS. If the real-time OS does not use r2, it can be used as a variable register.

Table 3-1. General-Purpose Registers

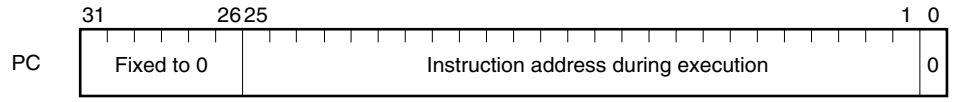
Name	Usage	Operation
r0	Zero register	Always holds 0
r1	Assembler-reserved register	Working register for generating 32-bit immediate data
r2	Address/data variable register (when r2 is not used by the real-time OS)	
r3	Stack pointer	Used to generate stack frame when function is called
r4	Global pointer	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 when memory is accessed
r31	Link pointer	Used by compiler when calling function

(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.

After reset: 00000000H



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

System Register 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 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 for future function expansion (operations that access these register numbers cannot be guaranteed).	×	×

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

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

Caution Even if bit 0 of EIPC, FEPC, or CTPC is set to 1 by 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, and 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)).

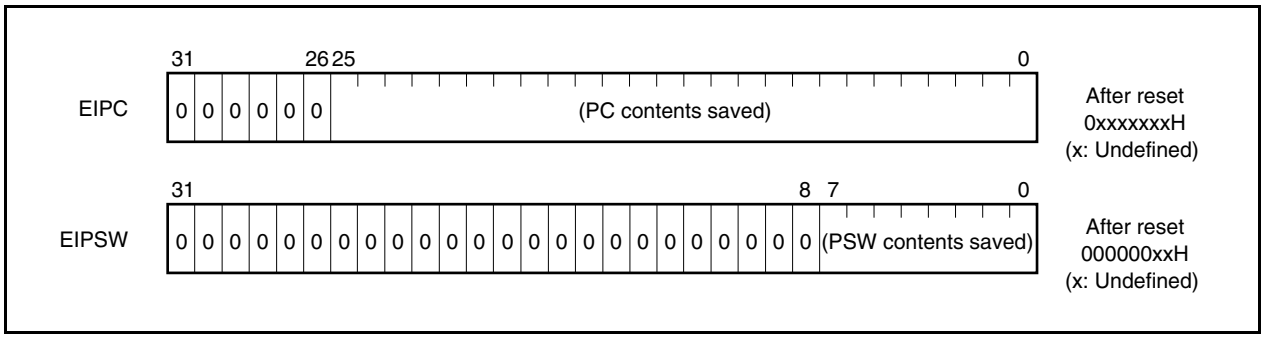
The address of the next instruction following the instruction executed when a software exception or maskable interrupt occurs is saved to EIPC, except for some instructions (see **20.9 Periods in Which CPU Does Not Acknowledge Interrupts**).

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.

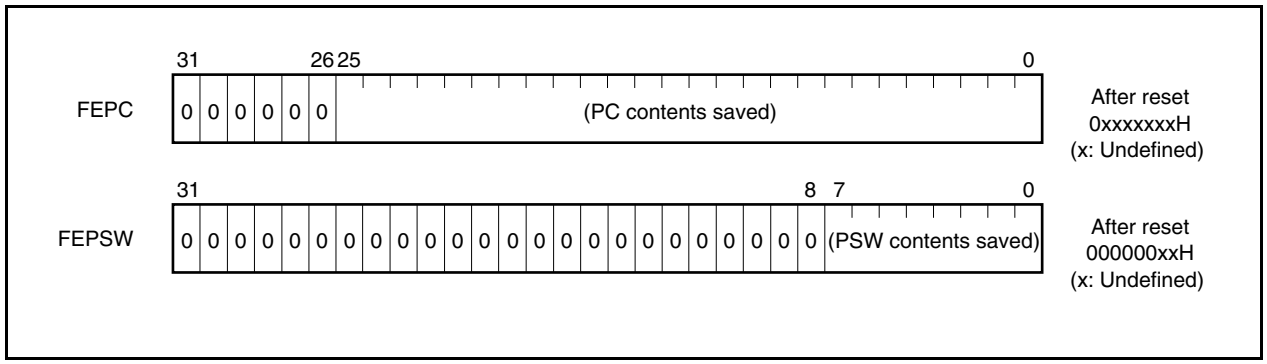
Upon occurrence of a non-maskable interrupt (NMI), the contents of the program counter (PC) are saved to FEPC and the contents of the program status word (PSW) are saved to FEPSW.

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.

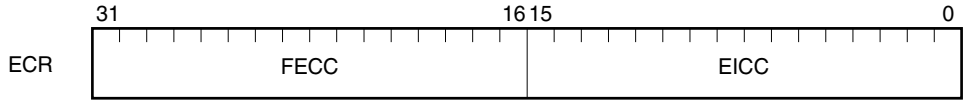
Bits 31 to 26 of FEPC and bits 31 to 8 of FEPSW are reserved (fixed to 0) for future function expansion.

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

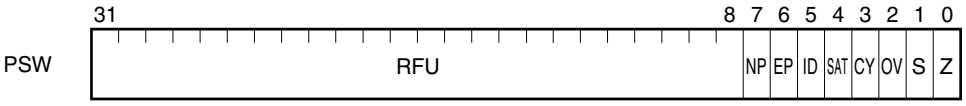
(4) Program status word (PSW)

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)

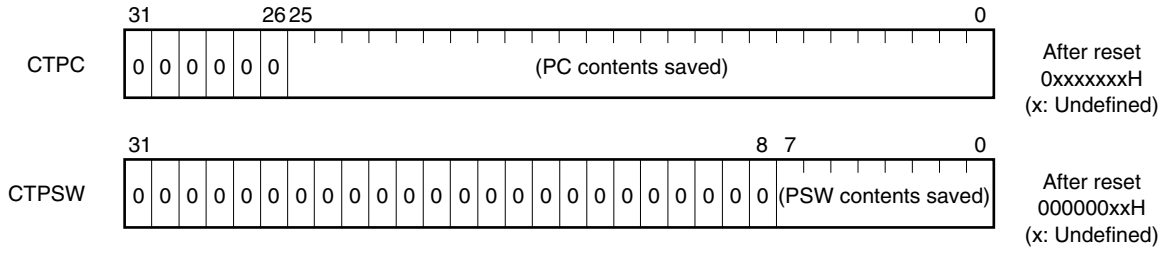
There are two CALLT execution status saving registers, CTPC and CTPSW.

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.

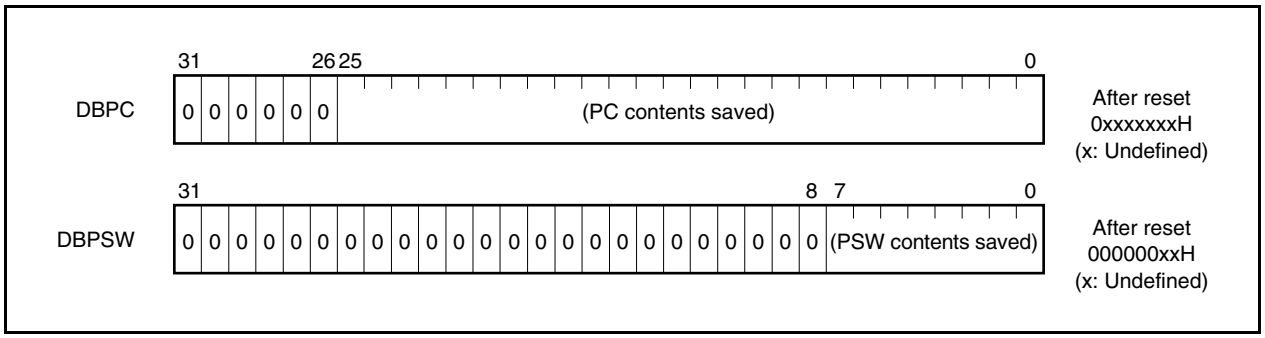
The contents saved to DBPC consist of the address of the next instruction after the instruction executed when an exception trap or debug trap occurs.

The current PSW contents are saved to DBPSW.

These registers can be read or written only in the period between DBTRAP instruction or illegal opcode 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.

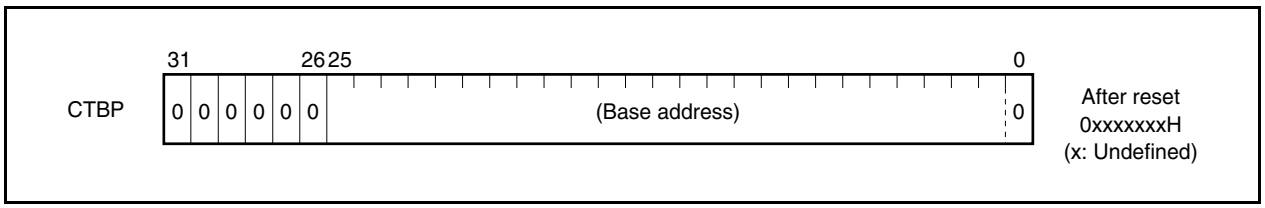
When the DBRET instruction has been executed, the values of DBPC and DBPSW are restored to the PC and PSW, respectively.



(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 Operating Modes

3.3.1 Operating modes

The V850E/IF3 and V850E/IG3 have the following operating modes. Mode specification is carried out using the FLMD0 and FLMD1 pins.

(1) Normal operation mode

In this mode, execution branches to the reset entry address in the internal ROM and instruction processing is started when system reset is released.

(2) Flash memory programming mode

If this mode is specified, a program can be written to the internal flash memory by the flash programmer.

3.3.2 Operating mode specification

The operating mode is specified according to the status (input level) of the FLMD0 and FLMD1 pins.

In the normal operating mode, input a low level to the FLMD0 pin after reset.

When the flash programmer is connected, a high level is input to the FLMD0 pin by the flash programmer in the flash memory programming mode; however, in the self-programming mode, input a high level via an external circuit.

Other than in the self-programming mode, fix the specifications of these pins in the application system, and do not change then during operation.

FLMD1	FLMD0	Operating Mode	Remarks
×	L	Normal operation mode	Internal ROM area is allocated from address 000000H.
L	H	Flash memory programming mode	–
H	H	Setting prohibited	

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

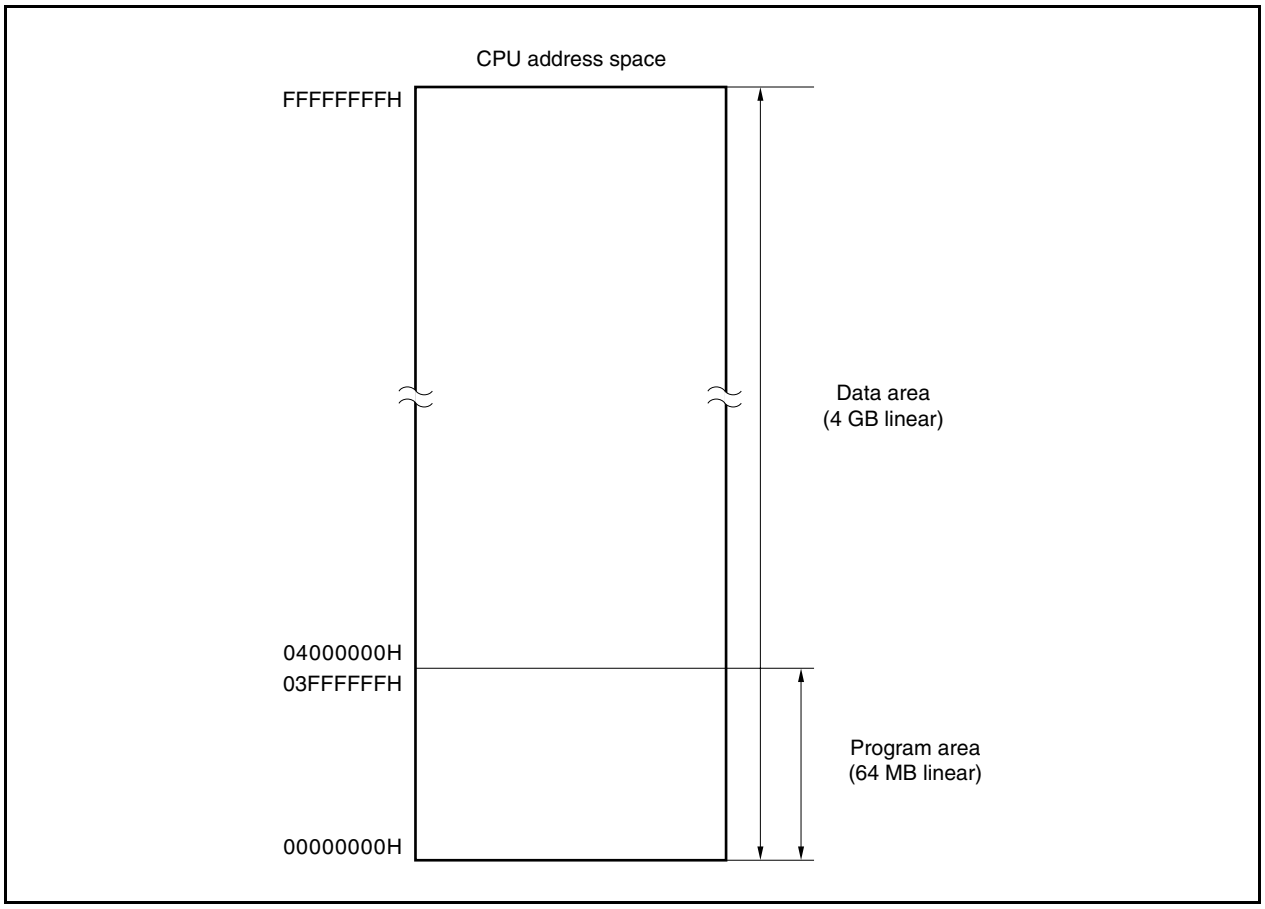
3.4 Address Space

3.4.1 CPU address space

The CPU of the V850E/IF3 and V850E/IG3 has 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 3-2 shows the CPU address space.

Figure 3-2. CPU Address Space

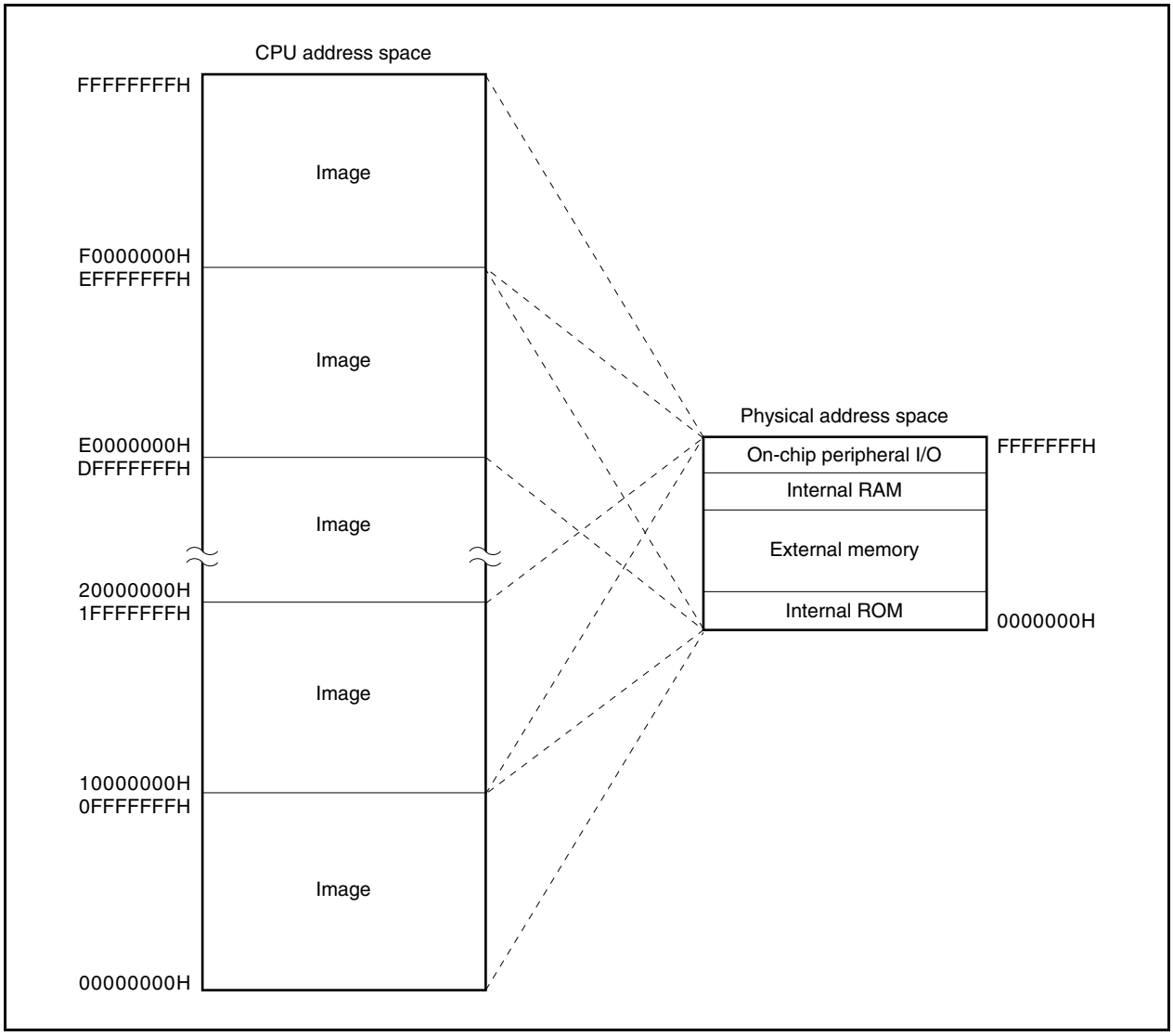


3.4.2 Image

A 256 MB physical address space is seen as 16 images 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 3-3 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 3-3. Images on Address Space



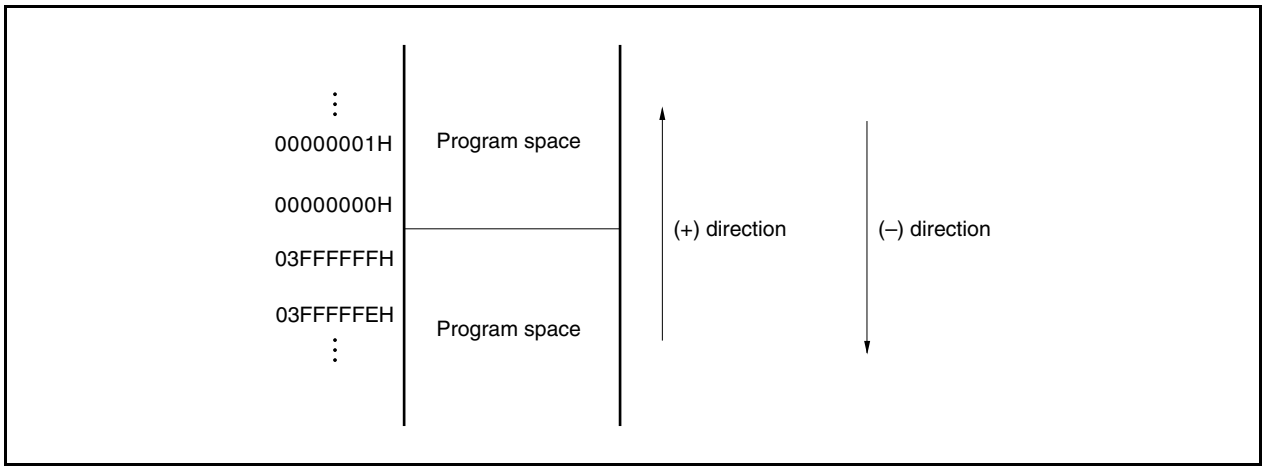
3.4.3 Wraparound of CPU address space

(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 a 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. Wraparound refers to a situation like this whereby the lower-limit address and upper-limit address become contiguous.

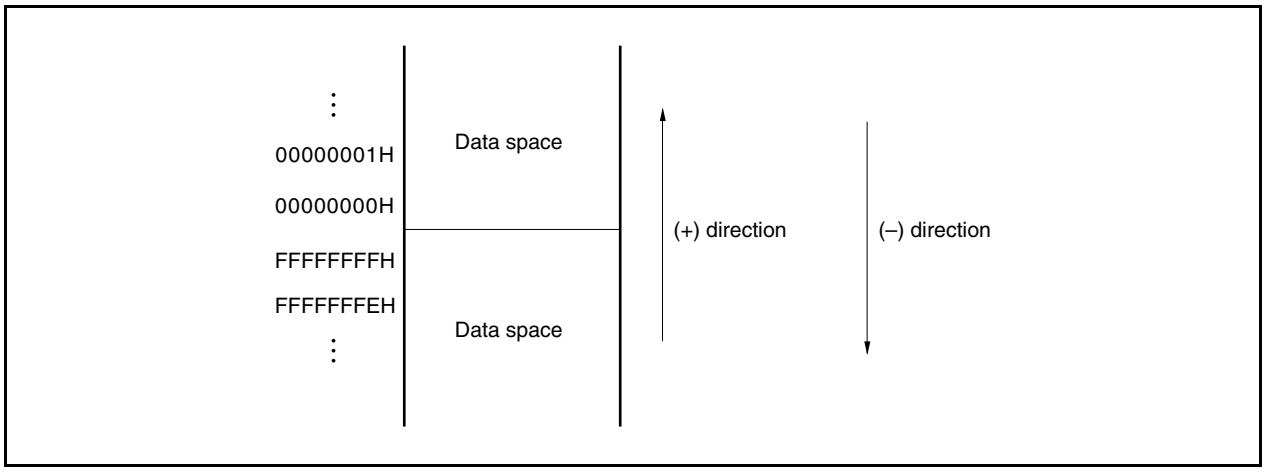
Caution The 4 KB area of 03FFF000H to 03FFFFFFH can be seen as an image of 0FFFF000H to 0FFFFFFFH. This area is access-prohibited. Therefore, do not execute any branch address calculation in which the result will reside in any part of this area.



(2) Data space

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

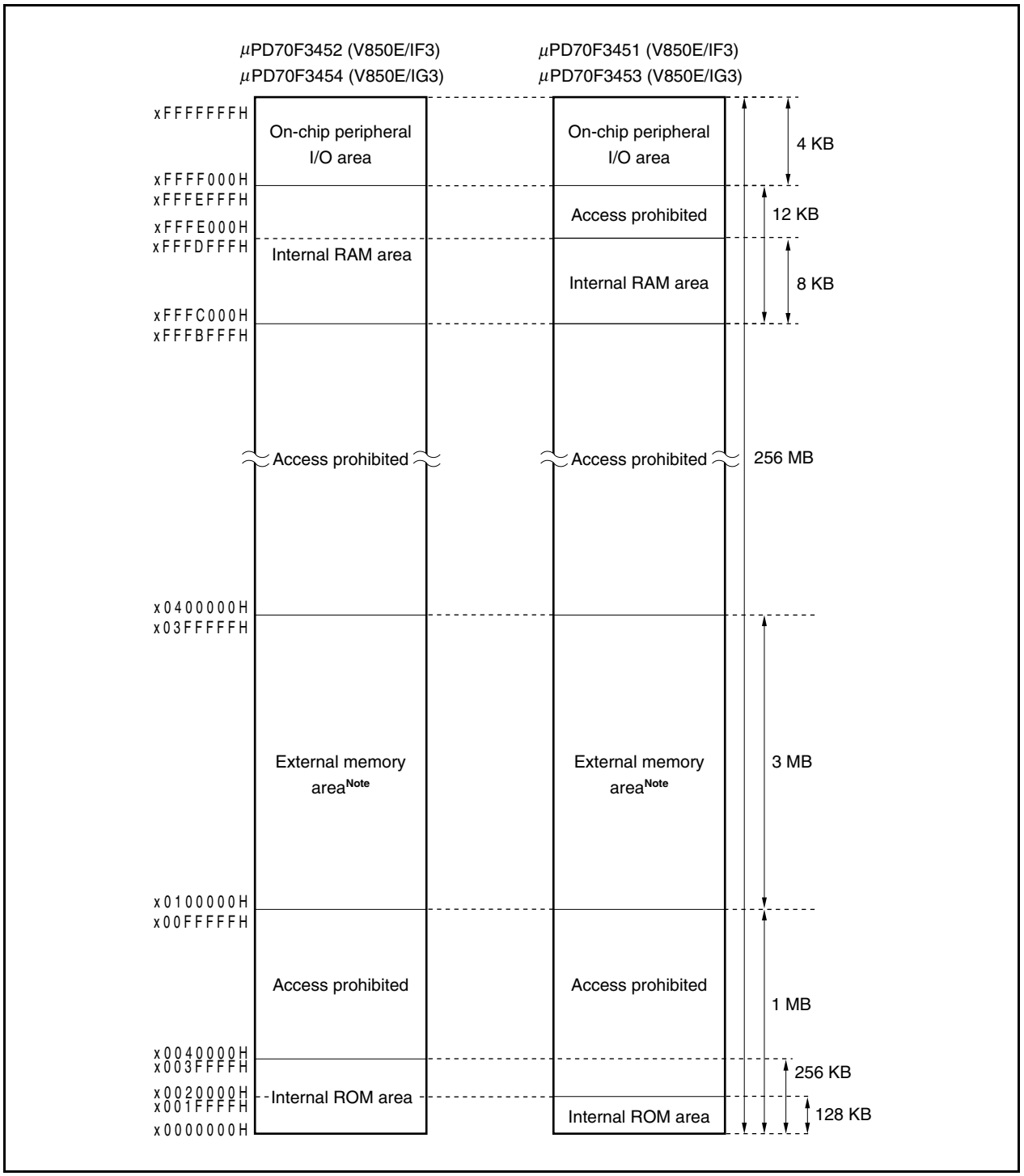
Therefore, the upper-limit address of the program space, address FFFFFFFFH, and the lower-limit address 00000000H are contiguous addresses, and the data space is wrapped around at the boundary of these addresses.



3.4.4 Memory map

The V850E/IF3 and V850E/IG3 reserve areas as shown in Figure 3-4.

Figure 3-4. Memory Map



Note μPD70F3454GC-8EA-A: External memory area
Others: Access prohibited area

3.4.5 Area

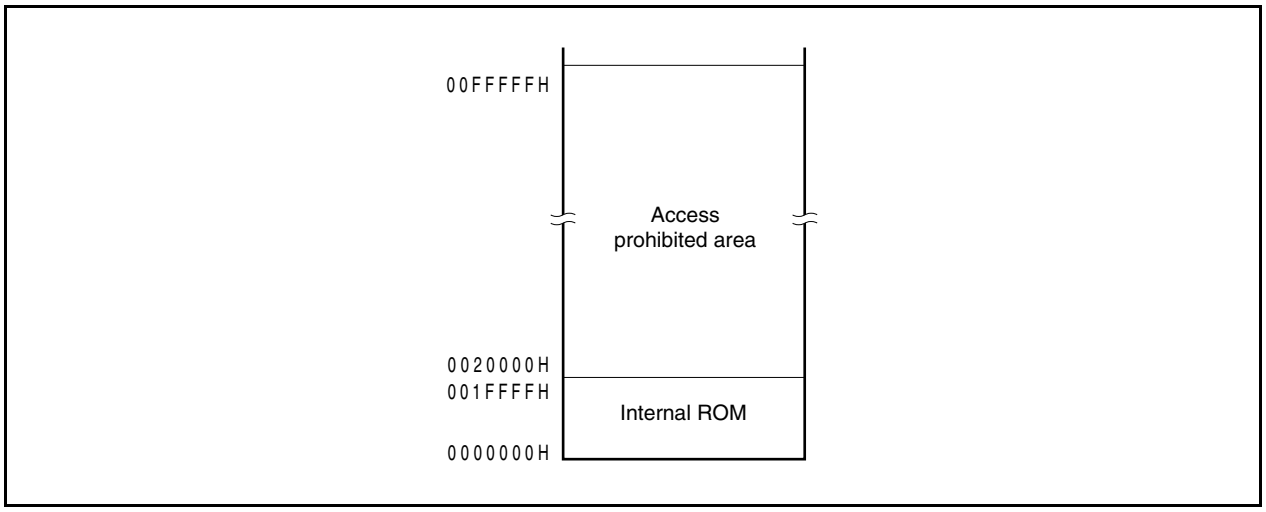
(1) Internal ROM area

1 MB of internal ROM area, addresses 00000H to FFFFFH, is reserved.

(a) μPD70F3451 (V850E/IF3), μPD70F3453 (V850E/IG3)

128 KB are provided at addresses 000000H to 01FFFFH as physical internal ROM.

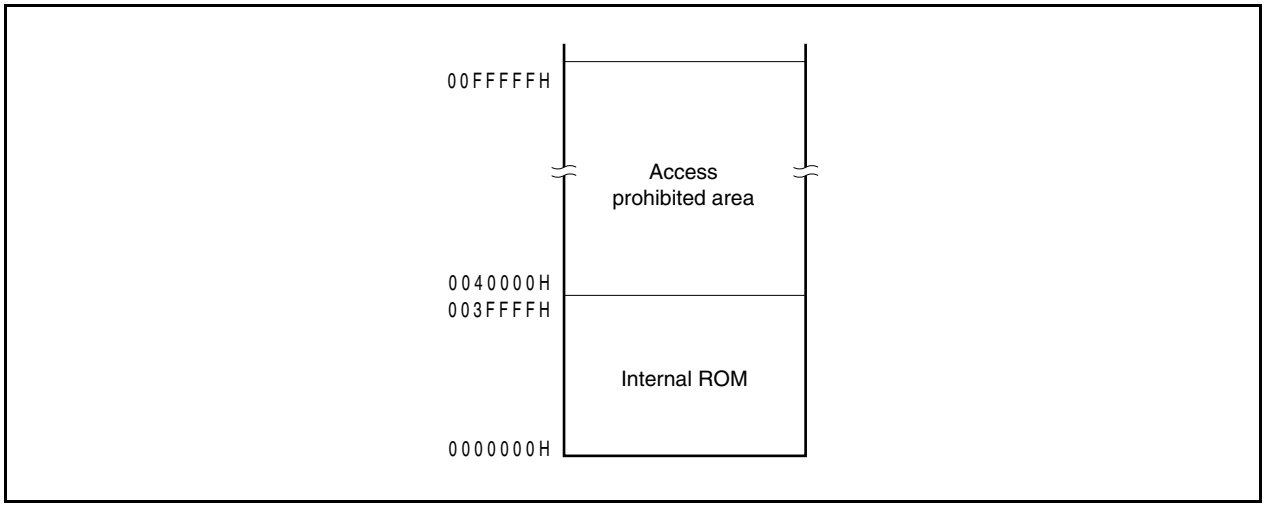
Figure 3-5. Internal ROM Area (128 KB)



(b) μPD70F3452 (V850E/IF3), μPD70F3454 (V850E/IG3)

256 KB are provided at addresses 000000H to 03FFFFH as physical internal ROM.

Figure 3-6. Internal ROM Area (256 KB)



(2) Internal RAM area

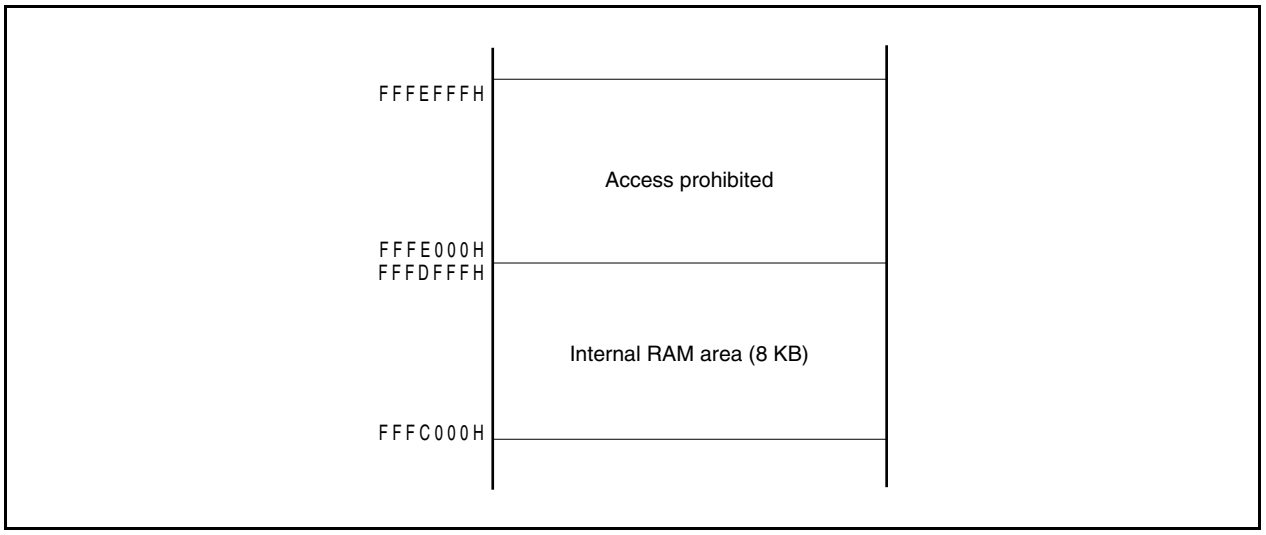
The 12 KB area of addresses FFFC000H to FFFEFFFH is reserved for the internal RAM area.

(a) μPD70F3451 (V850E/IF3), μPD70F3453 (V850E/IG3)

The 8 KB area of addresses FFFC000H to FFFDFFFH is provided as physical internal RAM.

Caution The following areas are access-prohibited.
Addresses FFFE000H to FFFEFFFH

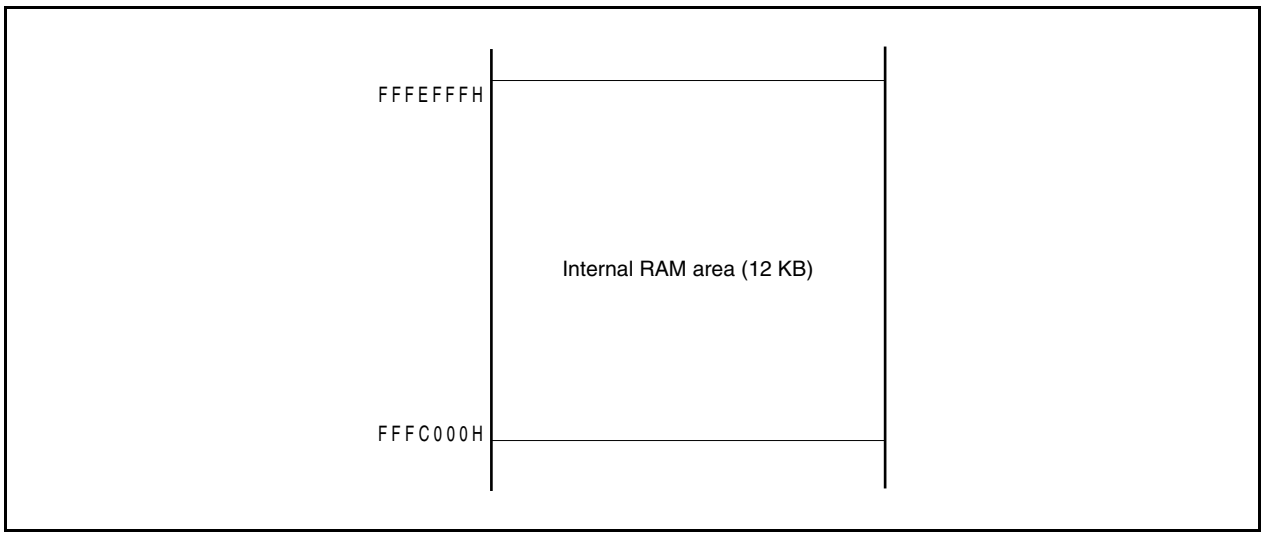
Figure 3-7. Internal RAM Area (8 KB)



(b) μPD70F3452 (V850E/IF3), μPD70F3454 (V850E/IG3)

The 12 KB area of addresses FFFC000H to FFFEFFFH is provided as physical internal RAM.

Figure 3-8. Internal RAM Area (12 KB)

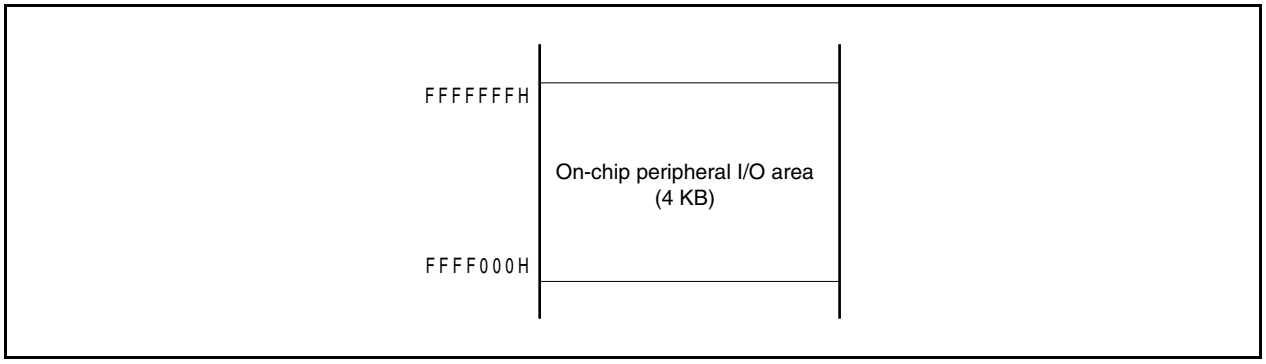


(3) On-chip peripheral I/O area

4 KB of memory, addresses FFFF000H to FFFFFFFH, is provided as an on-chip peripheral I/O area.
An image of addresses FFFF000H to FFFFFFFH can be seen at addresses 3FFF000H to 3FFFFFFH[Note].

Note Addresses 3FFF000H to 3FFFFFFH are access-prohibited. To access the on-chip peripheral I/O, specify addresses FFFF000H to FFFFFFFH.

Figure 3-9. On-Chip Peripheral I/O Area



On-chip peripheral I/O registers associated with the operating 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. In the V850E/IF3 and V850E/IG3, if a register is word accessed, halfword access is performed twice in the order of lower address, then higher address of the word area, disregarding the lower 2 bits of the address.

2. 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.

3. Addresses that are not defined as registers are reserved for future expansion. If these addresses are accessed, the operation is undefined and not guaranteed.
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 (μPD70F3454GC-8EA-A only)

3 MB (0100000H to 03FFFFFH) are available for the external memory area. For details, see **CHAPTER 18 BUS CONTROL FUNCTION**.

3.4.6 Recommended use of address space

The architecture of the V850E/IF3 and V850E/IG3 requires that a register that serves as a pointer be secured for address generation in operand data accessing of data space. Operand data access from instruction can be directly executed at the address in this pointer register area ±32 KB. However, because the general-purpose registers that can be used as a pointer register are limited, 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.

(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, unconditionally corresponds to the memory map of the program space.

(2) Data space

With the V850E/IF3 and V850E/IG3, 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 an address sign-extended to 32 bits.

(a) Application examples using wraparound

When R = r0 (zero register) is specified by the LD/ST disp16 [R] instruction, an addressing range of 00000000H ±32 KB can be referenced by the sign-extended disp16.

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

Example μPD70F3454 (V850E/IG3)

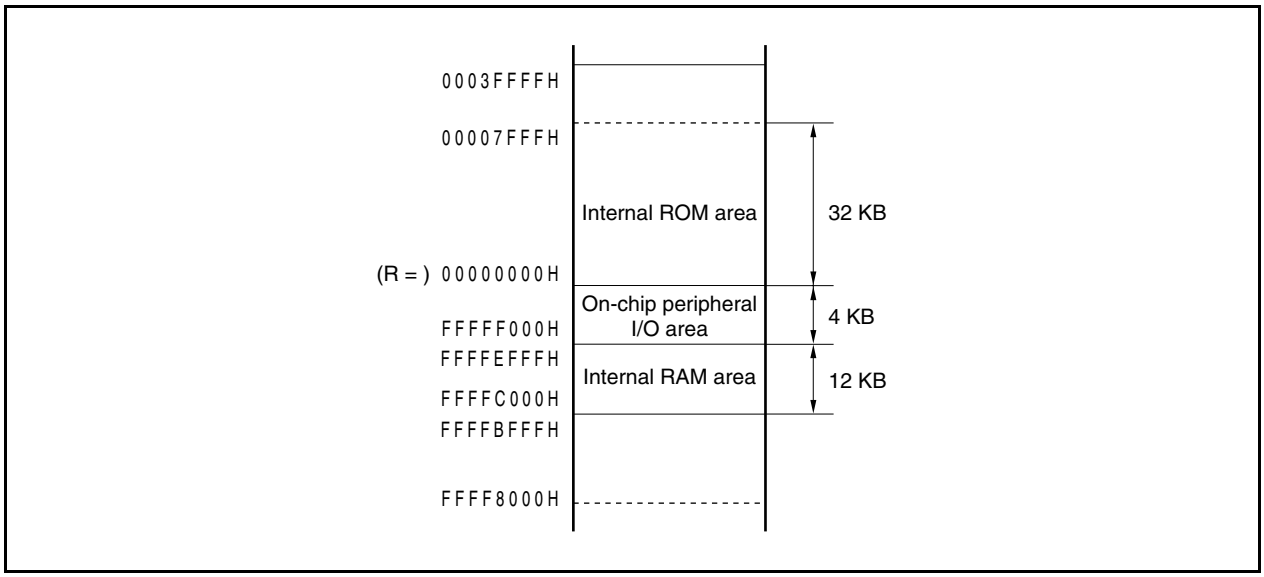
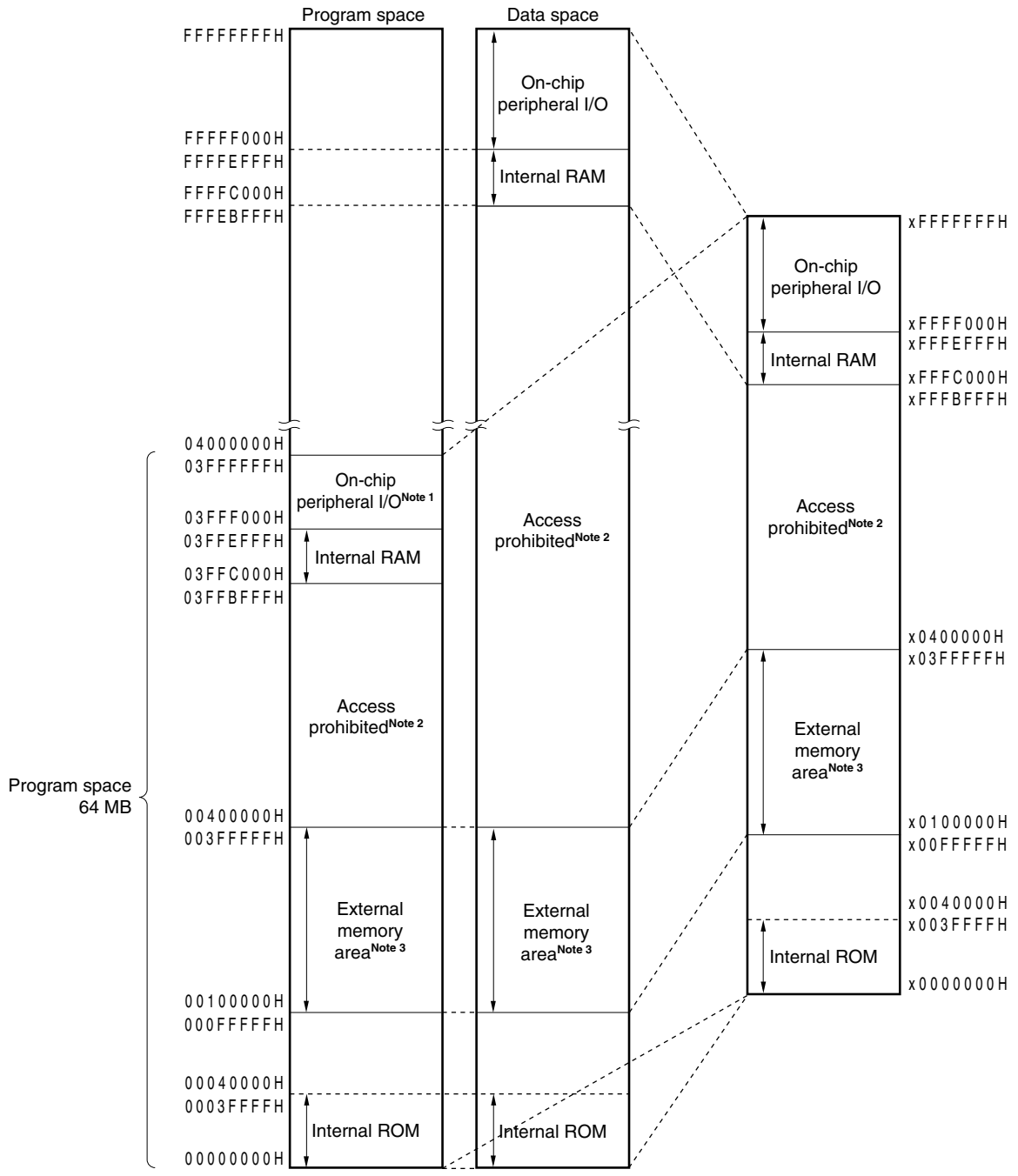


Figure 3-10. Recommended Memory Map



Notes 1. This area is access-prohibited. To access the on-chip peripheral I/O, specify addresses FFFF000H to FFFFFFFH.

2. The operation is not guaranteed if an access-prohibited area is accessed.

3. μPD70F3454GC-8EA-A: External memory area
Others: Access prohibited area Note 2

Remarks 1. The arrows indicate the recommended area.

2. This is a recommended memory map for the μPD70F3454 (V850E/IG3).

3.4.7 On-chip peripheral I/O registers

(1/14)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF004H	Port DL register	PDL	R/W			√	Undefined
FFFFF004H	Port DLL register	PDLL		√	√		Undefined
FFFFF005H	Port DLH register	PDLH		√	√		Undefined
FFFFF024H	Port DL mode register	PMDL				√	FFFFH
FFFFF024H	Port DL mode register L	PMDLL		√	√		FFH
FFFFF025H	Port DL mode register H	PMDLH		√	√		FFH
FFFFF044H	Port DL mode control register	PMCDL				√	0000H
FFFFF044H	Port DL mode control register L	PMCDLL		√	√		00H
FFFFF045H	Port DL mode control register H	PMCDLH		√	√		00H
FFFFF066H	Bus size configuration register	BSC				√	5555H
FFFFF06EH	System wait control register	VSWC			√		77H
FFFFF080H	DMA source address register 0L	DSA0L				√	Undefined
FFFFF082H	DMA source address register 0H	DSA0H				√	Undefined
FFFFF084H	DMA destination address register 0L	DDA0L				√	Undefined
FFFFF086H	DMA destination address register 0H	DDA0H				√	Undefined
FFFFF088H	DMA source address register 1L	DSA1L				√	Undefined
FFFFF08AH	DMA source address register 1H	DSA1H				√	Undefined
FFFFF08CH	DMA destination address register 1L	DDA1L				√	Undefined
FFFFF08EH	DMA destination address register 1H	DDA1H				√	Undefined
FFFFF090H	DMA source address register 2L	DSA2L				√	Undefined
FFFFF092H	DMA source address register 2H	DSA2H				√	Undefined
FFFFF094H	DMA destination address register 2L	DDA2L				√	Undefined
FFFFF096H	DMA destination address register 2H	DDA2H				√	Undefined
FFFFF098H	DMA source address register 3L	DSA3L				√	Undefined
FFFFF09AH	DMA source address register 3H	DSA3H				√	Undefined
FFFFF09CH	DMA destination address register 3L	DDA3L				√	Undefined
FFFFF09EH	DMA destination address register 3H	DDA3H				√	Undefined
FFFFF0C0H	DMA transfer count register 0	DBC0				√	Undefined
FFFFF0C2H	DMA transfer count register 1	DBC1				√	Undefined
FFFFF0C4H	DMA transfer count register 2	DBC2				√	Undefined
FFFFF0C6H	DMA transfer count register 3	DBC3				√	Undefined
FFFFF0D0H	DMA addressing control register 0	DADC0				√	0000H
FFFFF0D2H	DMA addressing control register 1	DADC1				√	0000H
FFFFF0D4H	DMA addressing control register 2	DADC2				√	0000H
FFFFF0D6H	DMA addressing control register 3	DADC3				√	0000H
FFFFF0E0H	DMA channel control register 0	DCHC0		√	√		00H
FFFFF0E2H	DMA channel control register 1	DCHC1		√	√		00H
FFFFF0E4H	DMA channel control register 2	DCHC2		√	√		00H
FFFFF0E6H	DMA channel control register 3	DCHC3		√	√		00H

(2/14)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF100H	Interrupt mask register 0	IMR0	R/W			√	FFFFH
FFFFF100H	Interrupt mask register 0L	IMR0L		√	√		FFH
FFFFF101H	Interrupt mask register 0H	IMR0H		√	√		FFH
FFFFF102H	Interrupt mask register 1	IMR1				√	FFFFH
FFFFF102H	Interrupt mask register 1L	IMR1L		√	√		FFH
FFFFF103H	Interrupt mask register 1H	IMR1H		√	√		FFH
FFFFF104H	Interrupt mask register 2	IMR2				√	FFFFH
FFFFF104H	Interrupt mask register 2L	IMR2L		√	√		FFH
FFFFF105H	Interrupt mask register 2H	IMR2H		√	√		FFH
FFFFF106H	Interrupt mask register 3	IMR3				√	FFFFH
FFFFF106H	Interrupt mask register 3L	IMR3L		√	√		FFH
FFFFF107H	Interrupt mask register 3H	IMR3H		√	√		FFH
FFFFF108H	Interrupt mask register 4	IMR4				√	FFFFH
FFFFF108H	Interrupt mask register 4L	IMR4L		√	√		FFH
FFFFF109H	Interrupt mask register 4H	IMR4H		√	√		FFH
FFFFF10AH	Interrupt mask register 5	IMR5				√	FFFFH
FFFFF10AH	Interrupt mask register 5L	IMR5L		√	√		FFH
FFFFF10BH	Interrupt mask register 5H	IMR5H		√	√		FFH
FFFFF110H	Interrupt control register	LVILIC		√	√		47H
FFFFF112H	Interrupt control register	LVIHIC		√	√		47H
FFFFF114H	Interrupt control register	PIC00		√	√		47H
FFFFF116H	Interrupt control register	PIC01		√	√		47H
FFFFF118H	Interrupt control register	PIC02[Note]		√	√		47H
FFFFF11AH	Interrupt control register	PIC03[Note]		√	√		47H
FFFFF11CH	Interrupt control register	PIC04[Note]		√	√		47H
FFFFF11EH	Interrupt control register	PIC05[Note]		√	√		47H
FFFFF120H	Interrupt control register	PIC06[Note]		√	√		47H
FFFFF122H	Interrupt control register	PIC07[Note]		√	√		47H
FFFFF124H	Interrupt control register	PIC08		√	√		47H
FFFFF126H	Interrupt control register	PIC09		√	√		47H
FFFFF128H	Interrupt control register	PIC10		√	√		47H
FFFFF12AH	Interrupt control register	PIC11		√	√		47H
FFFFF12CH	Interrupt control register	PIC12		√	√		47H
FFFFF12EH	Interrupt control register	PIC13		√	√		47H
FFFFF130H	Interrupt control register	PIC14		√	√		47H
FFFFF132H	Interrupt control register	PIC15		√	√		47H
FFFFF134H	Interrupt control register	PIC16		√	√		47H
FFFFF136H	Interrupt control register	PIC17		√	√		47H
FFFFF138H	Interrupt control register	PIC18		√	√		47H
FFFFF13AH	Interrupt control register	CMPIC0L		√	√		47H

Note V850E/IG3 only

(3/14)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF13CH	Interrupt control register	CMPIC0F	R/W	√	√		47H
FFFFF13EH	Interrupt control register	CMPIC1L		√	√		47H
FFFFF140H	Interrupt control register	CMPIC1F		√	√		47H
FFFFF142H	Interrupt control register	TB0OVIC		√	√		47H
FFFFF144H	Interrupt control register	TB0CCIC0		√	√		47H
FFFFF146H	Interrupt control register	TB0CCIC1		√	√		47H
FFFFF148H	Interrupt control register	TB0CCIC2		√	√		47H
FFFFF14AH	Interrupt control register	TB0CCIC3		√	√		47H
FFFFF14CH	Interrupt control register	TB1OVIC		√	√		47H
FFFFF14EH	Interrupt control register	TB1CCIC0		√	√		47H
FFFFF150H	Interrupt control register	TB1CCIC1		√	√		47H
FFFFF152H	Interrupt control register	TB1CCIC2		√	√		47H
FFFFF154H	Interrupt control register	TB1CCIC3		√	√		47H
FFFFF156H	Interrupt control register	TT0OVIC		√	√		47H
FFFFF158H	Interrupt control register	TT0CCIC0		√	√		47H
FFFFF15AH	Interrupt control register	TT0CCIC1		√	√		47H
FFFFF15CH	Interrupt control register	TT0IECIC[Note]		√	√		47H
FFFFF15EH	Interrupt control register	TT1OVIC		√	√		47H
FFFFF160H	Interrupt control register	TT1CCIC0		√	√		47H
FFFFF162H	Interrupt control register	TT1CCIC1		√	√		47H
FFFFF164H	Interrupt control register	TT1IECIC		√	√		47H
FFFFF166H	Interrupt control register	TA0OVIC		√	√		47H
FFFFF168H	Interrupt control register	TA0CCIC0		√	√		47H
FFFFF16AH	Interrupt control register	TA0CCIC1		√	√		47H
FFFFF16CH	Interrupt control register	TA1OVIC		√	√		47H
FFFFF16EH	Interrupt control register	TA1CCIC0		√	√		47H
FFFFF170H	Interrupt control register	TA1CCIC1		√	√		47H
FFFFF172H	Interrupt control register	TA2OVIC		√	√		47H
FFFFF174H	Interrupt control register	TA2CCIC0		√	√		47H
FFFFF176H	Interrupt control register	TA2CCIC1		√	√		47H
FFFFF178H	Interrupt control register	TA3OVIC		√	√		47H
FFFFF17AH	Interrupt control register	TA3CCIC0		√	√		47H
FFFFF17CH	Interrupt control register	TA3CCIC1		√	√		47H
FFFFF17EH	Interrupt control register	TA4OVIC		√	√		47H
FFFFF180H	Interrupt control register	TA4CCIC0		√	√		47H
FFFFF182H	Interrupt control register	TA4CCIC1		√	√		47H
FFFFF184H	Interrupt control register	DMAIC0		√	√		47H
FFFFF186H	Interrupt control register	DMAIC1		√	√		47H
FFFFF188H	Interrupt control register	DMAIC2		√	√		47H
FFFFF18AH	Interrupt control register	DMAIC3		√	√		47H

Note V850E/IG3 only

(4/14)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF18CH	Interrupt control register	UREIC	R/W	√	√		47H
FFFFF18EH	Interrupt control register	URIC		√	√		47H
FFFFF190H	Interrupt control register	UTIC		√	√		47H
FFFFF192H	Interrupt control register	UIFIC		√	√		47H
FFFFF194H	Interrupt control register	UTOIC		√	√		47H
FFFFF196H	Interrupt control register	UA0REIC		√	√		47H
FFFFF198H	Interrupt control register	UA0RIC		√	√		47H
FFFFF19AH	Interrupt control register	UA0TIC		√	√		47H
FFFFF19CH	Interrupt control register	CB0REIC		√	√		47H
FFFFF19EH	Interrupt control register	CB0RIC		√	√		47H
FFFFF1A0H	Interrupt control register	CB0TIC		√	√		47H
FFFFF1A2H	Interrupt control register	UA1REIC		√	√		47H
FFFFF1A4H	Interrupt control register	UA1RIC		√	√		47H
FFFFF1A6H	Interrupt control register	UA1TIC		√	√		47H
FFFFF1A8H	Interrupt control register	CB1REIC		√	√		47H
FFFFF1AAH	Interrupt control register	CB1RIC		√	√		47H
FFFFF1ACH	Interrupt control register	CB1TIC		√	√		47H
FFFFF1AEH	Interrupt control register	UA2REIC		√	√		47H
FFFFF1B0H	Interrupt control register	UA2RIC		√	√		47H
FFFFF1B2H	Interrupt control register	UA2TIC		√	√		47H
FFFFF1B4H	Interrupt control register	CB2REIC		√	√		47H
FFFFF1B6H	Interrupt control register	CB2RIC		√	√		47H
FFFFF1B8H	Interrupt control register	CB2TIC		√	√		47H
FFFFF1BAH	Interrupt control register	IICIC		√	√		47H
FFFFF1BCH	Interrupt control register	AD0IC		√	√		47H
FFFFF1BEH	Interrupt control register	AD1IC		√	√		47H
FFFFF1C0H	Interrupt control register	AD2IC		√	√		47H
FFFFF1C2H	Interrupt control register	TM0EQIC0		√	√		47H
FFFFF1C4H	Interrupt control register	TM1EQIC0		√	√		47H
FFFFF1C6H	Interrupt control register	TM2EQIC0		√	√		47H
FFFFF1C8H	Interrupt control register	TM3EQIC0		√	√		47H
FFFFF1CAH	Interrupt control register	ADT0IC		√	√		47H
FFFFF1CCH	Interrupt control register	ADT1IC		√	√		47H
FFFFF1FAH	In-service priority register	ISPR	R	√	√		00H
FFFFF1FCH	Command register	PRCMD	W		√		Undefined
FFFFF1FEH	Power save control register	PSC	R/W	√	√		00H
FFFFF200H	A/D0 conversion result register 0	AD0CR0	R			√	0000H
FFFFF201H	A/D0 conversion result register 0H	AD0CR0H			√		00H
FFFFF202H	A/D0 conversion result register 1	AD0CR1				√	0000H
FFFFF203H	A/D0 conversion result register 1H	AD0CR1H			√		00H

(5/14)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF204H	A/D0 conversion result register 2	AD0CR2	R			√	0000H
FFFFF205H	A/D0 conversion result register 2H	AD0CR2H			√		00H
FFFFF206H	A/D0 conversion result register 3	AD0CR3				√	0000H
FFFFF207H	A/D0 conversion result register 3H	AD0CR3H			√		00H
FFFFF208H	A/D0 conversion result register 4	AD0CR4				√	0000H
FFFFF209H	A/D0 conversion result register 4H	AD0CR4H			√		00H
FFFFF20AH	A/D0 conversion result register 5	AD0CR5				√	0000H
FFFFF20BH	A/D0 conversion result register 5H	AD0CR5H			√		00H
FFFFF20CH	A/D0 conversion result register 6	AD0CR6				√	0000H
FFFFF20DH	A/D0 conversion result register 6H	AD0CR6H			√		00H
FFFFF20EH	A/D0 conversion result register 7	AD0CR7				√	0000H
FFFFF20FH	A/D0 conversion result register 7H	AD0CR7H			√		00H
FFFFF210H	A/D0 conversion result register 8	AD0CR8				√	0000H
FFFFF211H	A/D0 conversion result register 8H	AD0CR8H			√		00H
FFFFF212H	A/D0 conversion result register 9	AD0CR9				√	0000H
FFFFF213H	A/D0 conversion result register 9H	AD0CR9H			√		00H
FFFFF214H	A/D0 conversion result register 10	AD0CR10				√	0000H
FFFFF215H	A/D0 conversion result register 10H	AD0CR10H			√		00H
FFFFF216H	A/D0 conversion result register 11	AD0CR11				√	0000H
FFFFF217H	A/D0 conversion result register 11H	AD0CR11H			√		00H
FFFFF218H	A/D0 conversion result register 12	AD0CR12				√	0000H
FFFFF219H	A/D0 conversion result register 12H	AD0CR12H			√		00H
FFFFF21AH	A/D0 conversion result register 13	AD0CR13				√	0000H
FFFFF21BH	A/D0 conversion result register 13H	AD0CR13H			√		00H
FFFFF21CH	A/D0 conversion result register 14	AD0CR14				√	0000H
FFFFF21DH	A/D0 conversion result register 14H	AD0CR14H			√		00H
FFFFF21EH	A/D0 conversion result register 15	AD0CR15				√	0000H
FFFFF21FH	A/D0 conversion result register 15H	AD0CR15H			√		00H
FFFFF220H	A/D converter 0 scan mode register	AD0SCM	R/W			√	0000H
FFFFF220H	A/D converter 0 scan mode register L	AD0SCML		√	√		00H
FFFFF221H	A/D converter 0 scan mode register H	AD0SCMH		√	√		00H
FFFFF222H	A/D converter 0 conversion time control register	AD0CTC		√	√		00H
FFFFF224H	A/D converter 0 conversion channel specification register	AD0CHEN				√	0000H
FFFFF224H	A/D converter 0 conversion channel specification register L	AD0CHENL		√	√		00H
FFFFF225H	A/D converter 0 conversion channel specification register H	AD0CHENH		√	√		00H
FFFFF230H	A/D converter 0 control register	AD0CTL0		√	√		00H
FFFFF231H	A/D converter 0 trigger select register	AD0TSEL		√	√		10H
FFFFF232H	A/D converter 0 channel specification register 1	AD0CH1		√	√		00H

(6/14)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF233H	A/D converter 0 channel specification register 2	AD0CH2	R/W	√	√		00H
FFFFF240H	A/D0 conversion result expansion register 0	AD0ECR0	R			√	0000H
FFFFF241H	A/D0 conversion result expansion register 0H	AD0ECR0H			√		00H
FFFFF242H	A/D0 conversion result expansion register 1	AD0ECR1				√	0000H
FFFFF243H	A/D0 conversion result expansion register 1H	AD0ECR1H			√		00H
FFFFF244H	A/D0 conversion result expansion register 2	AD0ECR2				√	0000H
FFFFF245H	A/D0 conversion result expansion register 2H	AD0ECR2H			√		00H
FFFFF246H	A/D0 conversion result expansion register 3	AD0ECR3				√	0000H
FFFFF247H	A/D0 conversion result expansion register 3H	AD0ECR3H			√		00H
FFFFF248H	A/D0 conversion result expansion register 4	AD0ECR4				√	0000H
FFFFF249H	A/D0 conversion result expansion register 4H	AD0ECR4H			√		00H
FFFFF254H	A/D converter 0 flag register	AD0FLG			√		00H
FFFFF255H	A/D converter 0 flag buffer register	AD0FLGB			√		00H
FFFFF260H	Operational amplifier 0 control register 0	OP0CTL0	R/W		√		00H
FFFFF261H	Comparator 0 control register 0	CMP0CTL0			√		00H
FFFFF262H	Comparator 0 control register 1	CMP0CTL1	R		√		00H
FFFFF263H	Comparator 0 control register 2	CMP0CTL2	R/W		√		00H
FFFFF264H	Comparator 0 control register 3	CMP0CTL3			√		00H
FFFFF270H	A/D converter 0 clock select register	AD0OCKS			√		00H
FFFFF274H	A/D converter 1 clock select register	AD1OCKS			√		00H
FFFFF278H	Comparator output digital noise elimination register 0L	CMPNFC0L			√		00H
FFFFF27AH	Comparator output digital noise elimination register 0F	CMPNFC0F			√		00H
FFFFF27CH	Comparator output digital noise elimination register 1L	CMPNFC1L			√		00H
FFFFF27EH	Comparator output digital noise elimination register 1F	CMPNFC1F			√		00H
FFFFF280H	A/D1 conversion result register 0	AD1CR0	R			√	0000H
FFFFF281H	A/D1 conversion result register 0H	AD1CR0H			√		00H
FFFFF282H	A/D1 conversion result register 1	AD1CR1				√	0000H
FFFFF283H	A/D1 conversion result register 1H	AD1CR1H			√		00H
FFFFF284H	A/D1 conversion result register 2	AD1CR2				√	0000H
FFFFF285H	A/D1 conversion result register 2H	AD1CR2H			√		00H
FFFFF286H	A/D1 conversion result register 3	AD1CR3				√	0000H
FFFFF287H	A/D1 conversion result register 3H	AD1CR3H			√		00H
FFFFF288H	A/D1 conversion result register 4	AD1CR4				√	0000H
FFFFF289H	A/D1 conversion result register 4H	AD1CR4H			√		00H
FFFFF28AH	A/D1 conversion result register 5	AD1CR5				√	0000H
FFFFF28BH	A/D1 conversion result register 5H	AD1CR5H			√		00H
FFFFF28CH	A/D1 conversion result register 6	AD1CR6				√	0000H
FFFFF28DH	A/D1 conversion result register 6H	AD1CR6H			√		00H
FFFFF28EH	A/D1 conversion result register 7	AD1CR7				√	0000H
FFFFF28FH	A/D1 conversion result register 7H	AD1CR7H			√		00H
FFFFF290H	A/D1 conversion result register 8	AD1CR8				√	0000H
FFFFF291H	A/D1 conversion result register 8H	AD1CR8H			√		00H

(7/14)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF292H	A/D1 conversion result register 9	AD1CR9	R			√	0000H
FFFFF293H	A/D1 conversion result register 9H	AD1CR9H			√		00H
FFFFF294H	A/D1 conversion result register 10	AD1CR10				√	0000H
FFFFF295H	A/D1 conversion result register 10H	AD1CR10H			√		00H
FFFFF296H	A/D1 conversion result register 11	AD1CR11				√	0000H
FFFFF297H	A/D1 conversion result register 11H	AD1CR11H			√		00H
FFFFF298H	A/D1 conversion result register 12	AD1CR12				√	0000H
FFFFF299H	A/D1 conversion result register 12H	AD1CR12H			√		00H
FFFFF29AH	A/D1 conversion result register 13	AD1CR13				√	0000H
FFFFF29BH	A/D1 conversion result register 13H	AD1CR13H			√		00H
FFFFF29CH	A/D1 conversion result register 14	AD1CR14				√	0000H
FFFFF29DH	A/D1 conversion result register 14H	AD1CR14H			√		00H
FFFFF29EH	A/D1 conversion result register 15	AD1CR15				√	0000H
FFFFF29FH	A/D1 conversion result register 15H	AD1CR15H			√		00H
FFFFF2A0H	A/D converter 1 scan mode register	AD1SCM	R/W			√	0000H
FFFFF2A0H	A/D converter 1 scan mode register L	AD1SCML		√	√		00H
FFFFF2A1H	A/D converter 1 scan mode register H	AD1SCMH		√	√		00H
FFFFF2A2H	A/D converter 1 conversion time control register	AD1CTC		√	√		00H
FFFFF2A4H	A/D converter 1 conversion channel specification register	AD1CHEN				√	0000H
FFFFF2A4H	A/D converter 1 conversion channel specification register L	AD1CHENL		√	√		00H
FFFFF2A5H	A/D converter 1 conversion channel specification register H	AD1CHENH		√	√		00H
FFFFF2B0H	A/D converter 1 control register	AD1CTL0		√	√		00H
FFFFF2B1H	A/D converter 1 trigger select register	AD1TSEL		√	√		10H
FFFFF2B2H	A/D converter 1 channel specification register 1	AD1CH1		√	√		00H
FFFFF2B3H	A/D converter 1 channel specification register 2	AD1CH2		√	√		00H
FFFFF2C0H	A/D1 conversion result expansion register 0	AD1ECR0	R			√	0000H
FFFFF2C1H	A/D1 conversion result expansion register 0H	AD1ECR0H			√		00H
FFFFF2C2H	A/D1 conversion result expansion register 1	AD1ECR1				√	0000H
FFFFF2C3H	A/D1 conversion result expansion register 1H	AD1ECR1H			√		00H
FFFFF2C4H	A/D1 conversion result expansion register 2	AD1ECR2				√	0000H
FFFFF2C5H	A/D1 conversion result expansion register 2H	AD1ECR2H			√		00H
FFFFF2C6H	A/D1 conversion result expansion register 3	AD1ECR3				√	0000H
FFFFF2C7H	A/D1 conversion result expansion register 3H	AD1ECR3H			√		00H
FFFFF2C8H	A/D1 conversion result expansion register 4	AD1ECR4				√	0000H
FFFFF2C9H	A/D1 conversion result expansion register 4H	AD1ECR4H			√		00H
FFFFF2D4H	A/D converter 1 flag register	AD1FLG			√		00H
FFFFF2D5H	A/D converter 1 flag buffer register	AD1FLGB			√		00H
FFFFF2E0H	Operational amplifier 1 control register 0	OP1CTL0	R/W		√		00H
FFFFF2E1H	Comparator 1 control register 0	CMP1CTL0			√		00H
FFFFF2E2H	Comparator 1 control register 1	CMP1CTL1	R		√		00H

(8/14)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF2E3H	Comparator 1 control register 2	CMP1CTL2	R/W		√		00H
FFFFF2E4H	Comparator 1 control register 3	CMP1CTL3			√		00H
FFFFF2F0H	A/D trigger falling edge specification register	ADTF		√	√		00H
FFFFF2F2H	A/D trigger rising edge specification register	ADTR		√	√		00H
FFFFF2F4H	Comparator output interrupt falling edge specification register	CMPOF		√	√		00H
FFFFF2F6H	Comparator output interrupt rising edge specification register	CMPOR		√	√		00H
FFFFF2F8H	A/DLDTRG1 input select register	ADLTS1			√		00H
FFFFF2FAH	A/DLDTRG2 input select register	ADLTS2			√		00H
FFFFF310H	Digital noise elimination 0 control register 14	INTNFC14			√		00H
FFFFF312H	Digital noise elimination 0 control register 15	INTNFC15			√		00H
FFFFF314H	Digital noise elimination 0 control register 16	INTNFC16			√		00H
FFFFF400H	Port 0 register	P0		√	√		Undefined
FFFFF402H	Port 1 register	P1		√	√		Undefined
FFFFF404H	Port 2 register	P2		√	√		Undefined
FFFFF406H	Port 3 register	P3		√	√		Undefined
FFFFF408H	Port 4 register	P4		√	√		Undefined
FFFFF420H	Port 0 mode register	PM0		√	√		FFH
FFFFF422H	Port 1 mode register	PM1		√	√		FFH
FFFFF424H	Port 2 mode register	PM2		√	√		FFH
FFFFF426H	Port 3 mode register	PM3		√	√		FFH
FFFFF428H	Port 4 mode register	PM4		√	√		FFH
FFFFF440H	Port 0 mode control register	PMC0		√	√		00H
FFFFF442H	Port 1 mode control register	PMC1		√	√		00H
FFFFF444H	Port 2 mode control register	PMC2		√	√		00H
FFFFF446H	Port 3 mode control register	PMC3		√	√		00H
FFFFF448H	Port 4 mode control register	PMC4		√	√		00H
FFFFF460H	Port 0 function control register	PFC0		√	√		00H
FFFFF462H	Port 1 function control register	PFC1		√	√		00H
FFFFF464H	Port 2 function control register	PFC2		√	√		00H
FFFFF466H	Port 3 function control register	PFC3		√	√		00H
FFFFF468H	Port 4 function control register	PFC4		√	√		00H
FFFFF480H	Bus cycle type configuration register 0	BCT0				√	CCCCH
FFFFF484H	Data wait control register 0	DWC0				√	7777H
FFFFF488H	Address wait control register	AWC				√	FFFFH
FFFFF48AH	Bus cycle control register	BCC				√	AAAAH
FFFFF48EH	Bus clock division control register	DVC			√		81H
FFFFF540H	TMM0 control register 0	TM0CTL0		√	√		00H
FFFFF544H	TMM0 compare register 0	TM0CMP0				√	0000H
FFFFF550H	TMM1 control register 0	TM1CTL0		√	√		00H

(9/14)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF554H	TMM1 compare register 0	TM1CMP0	R/W			√	0000H
FFFFF560H	TMM2 control register 0	TM2CTL0		√	√		00H
FFFFF564H	TMM2 compare register 0	TM2CMP0				√	0000H
FFFFF570H	TMM3 control register 0	TM3CTL0		√	√		00H
FFFFF574H	TMM3 compare register 0	TM3CMP0				√	0000H
FFFFF580H	TMT0 control register 0	TT0CTL0		√	√		00H
FFFFF581H	TMT0 control register 1	TT0CTL1		√	√		00H
FFFFF582H	TMT0 control register 2	TT0CTL2Note		√	√		00H
FFFFF583H	TMT0 I/O control register 0	TT0IOC0Note		√	√		00H
FFFFF584H	TMT0 I/O control register 1	TT0IOC1Note		√	√		00H
FFFFF585H	TMT0 I/O control register 2	TT0IOC2Note		√	√		00H
FFFFF586H	TMT0 I/O control register 3	TT0IOC3Note		√	√		00H
FFFFF587H	TMT0 option register 0	TT0OPT0		√	√		00H
FFFFF588H	TMT0 option register 1	TT0OPT1Note		√	√		00H
FFFFF58AH	TMT0 capture/compare register 0	TT0CCR0				√	0000H
FFFFF58CH	TMT0 capture/compare register 1	TT0CCR1				√	0000H
FFFFF58EH	TMT0 counter read buffer register	TT0CNT	R			√	0000H
FFFFF590H	TMT0 counter write register	TT0TCWNote	R/W			√	0000H
FFFFF5A0H	Digital noise elimination 2 control register 0	TTNFC0Note			√		00H
FFFFF5A2H	Digital noise elimination 2 control register 1	TTNFC1			√		00H
FFFFF5A4H	TMT0 capture input select register	TTISL0Note			√		Undefined
FFFFF5A6H	TMT1 capture input select register	TTISL1			√		Undefined
FFFFF5C0H	TMT1 control register 0	TT1CTL0		√	√		00H
FFFFF5C1H	TMT1 control register 1	TT1CTL1		√	√		00H
FFFFF5C2H	TMT1 control register 2	TT1CTL2		√	√		00H
FFFFF5C3H	TMT1 I/O control register 0	TT1IOC0		√	√		00H
FFFFF5C4H	TMT1 I/O control register 1	TT1IOC1		√	√		00H
FFFFF5C5H	TMT1 I/O control register 2	TT1IOC2		√	√		00H
FFFFF5C6H	TMT1 I/O control register 3	TT1IOC3		√	√		00H
FFFFF5C7H	TMT1 option register 0	TT1OPT0		√	√		00H
FFFFF5C8H	TMT1 option register 1	TT1OPT1		√	√		00H
FFFFF5CAH	TMT1 capture/compare register 0	TT1CCR0				√	0000H
FFFFF5CCH	TMT1 capture/compare register 1	TT1CCR1				√	0000H
FFFFF5CEH	TMT1 counter read buffer register	TT1CNT	R			√	0000H
FFFFF5D0H	TMT1 counter write register	TT1TCW	R/W			√	0000H
FFFFF5E0H	TAB0 control register 0	TAB0CTL0		√	√		00H
FFFFF5E1H	TAB0 control register 1	TAB0CTL1		√	√		00H
FFFFF5E2H	TAB0 I/O control register 0	TAB0IOC0		√	√		00H
FFFFF5E3H	TAB0 I/O control register 1	TAB0IOC1		√	√		00H
FFFFF5E4H	TAB0 I/O control register 2	TAB0IOC2		√	√		00H

Note V850E/IG3 only

(10/14)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF5E5H	TAB0 option register 0	TAB0OPT0	R/W	√	√		00H
FFFFF5E6H	TAB0 capture/compare register 0	TAB0CCR0				√	0000H
FFFFF5E8H	TAB0 capture/compare register 1	TAB0CCR1				√	0000H
FFFFF5EAH	TAB0 capture/compare register 2	TAB0CCR2				√	0000H
FFFFF5ECH	TAB0 capture/compare register 3	TAB0CCR3				√	0000H
FFFFF5EEH	TAB0 counter read buffer register	TAB0CNT	R			√	0000H
FFFFF600H	TAB0 option register 1	TAB0OPT1	R/W	√	√		00H
FFFFF601H	TAB0 option register 2	TAB0OPT2		√	√		00H
FFFFF602H	TAB0 I/O control register 3	TAB0IOC3		√	√		A8H
FFFFF603H	TAB0 option register 3	TAB0OPT3		√	√		00H
FFFFF604H	TAB0 deadtime compare register	TAB0DTC				√	0000H
FFFFF610H	High-impedance output control register 00	HZA0CTL0		√	√		00H
FFFFF611H	High-impedance output control register 01	HZA0CTL1		√	√		00H
FFFFF618H	High-impedance output control register 10	HZA1CTL0		√	√		00H
FFFFF619H	High-impedance output control register 11	HZA1CTL1[Note]		√	√		00H
FFFFF620H	TAB1 control register 0	TAB1CTL0		√	√		00H
FFFFF621H	TAB1 control register 1	TAB1CTL1		√	√		00H
FFFFF622H	TAB1 I/O control register 0	TAB1IOC0		√	√		00H
FFFFF623H	TAB1 I/O control register 1	TAB1IOC1		√	√		00H
FFFFF624H	TAB1 I/O control register 2	TAB1IOC2		√	√		00H
FFFFF625H	TAB1 option register 0	TAB1OPT0		√	√		00H
FFFFF626H	TAB1 capture/compare register 0	TAB1CCR0				√	0000H
FFFFF628H	TAB1 capture/compare register 1	TAB1CCR1				√	0000H
FFFFF62AH	TAB1 capture/compare register 2	TAB1CCR2				√	0000H
FFFFF62CH	TAB1 capture/compare register 3	TAB1CCR3				√	0000H
FFFFF62EH	TAB1 counter read buffer register	TAB1CNT	R			√	0000H
FFFFF640H	TAB1 option register 1	TAB1OPT1	R/W	√	√		00H
FFFFF641H	TAB1 option register 2	TAB1OPT2		√	√		00H
FFFFF642H	TAB1 I/O control register 3	TAB1IOC3		√	√		A8H
FFFFF643H	TAB1 option register 3	TAB1OPT3		√	√		00H
FFFFF644H	TAB1 deadtime compare register	TAB1DTC				√	0000H
FFFFF650H	High-impedance output control register 20	HZA2CTL0		√	√		00H
FFFFF651H	High-impedance output control register 21	HZA2CTL1		√	√		00H
FFFFF658H	High-impedance output control register 30	HZA3CTL0		√	√		00H
FFFFF659H	High-impedance output control register 31	HZA3CTL1		√	√		00H
FFFFF660H	TAA0 control register 0	TAA0CTL0		√	√		00H
FFFFF661H	TAA0 control register 1	TAA0CTL1		√	√		00H
FFFFF665H	TAA0 option register 0	TAA0OPT0		√	√		00H
FFFFF666H	TAA0 capture/compare register 0	TAA0CCR0				√	0000H
FFFFF668H	TAA0 capture/compare register 1	TAA0CCR1				√	0000H
FFFFF66AH	TAA0 counter read buffer register	TAA0CNT	R			√	0000H

Note V850E/IG3 only

(11/14)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF680H	TAA1 control register 0	TAA1CTL0	R/W	√	√		00H
FFFFF681H	TAA1 control register 1	TAA1CTL1		√	√		00H
FFFFF685H	TAA1 option register 0	TAA1OPT0		√	√		00H
FFFFF686H	TAA1 capture/compare register 0	TAA1CCR0				√	0000H
FFFFF688H	TAA1 capture/compare register 1	TAA1CCR1				√	0000H
FFFFF68AH	TAA1 counter read buffer register	TAA1CNT	R			√	0000H
FFFFF6A0H	TAA2 control register 0	TAA2CTL0	R/W	√	√		00H
FFFFF6A1H	TAA2 control register 1	TAA2CTL1		√	√		00H
FFFFF6A2H	TAA2 I/O control register 0	TAA2IOC0		√	√		00H
FFFFF6A3H	TAA2 I/O control register 1	TAA2IOC1		√	√		00H
FFFFF6A4H	TAA2 I/O control register 2	TAA2IOC2		√	√		00H
FFFFF6A5H	TAA2 option register 0	TAA2OPT0		√	√		00H
FFFFF6A6H	TAA2 capture/compare register 0	TAA2CCR0				√	0000H
FFFFF6A8H	TAA2 capture/compare register 1	TAA2CCR1				√	0000H
FFFFF6AAH	TAA2 counter read buffer register	TAA2CNT	R			√	0000H
FFFFF6C0H	Oscillation stabilization time select register	OSTS	R/W		√		04H
FFFFF6D0H	Watchdog timer mode register	WDTM			√		67H
FFFFF6D1H	Watchdog timer enable register	WDTE			√		1AH
FFFFF700H	Port 0 function control expansion register	PFCE0		√	√		00H
FFFFF702H	Port 1 function control expansion register	PFCE1		√	√		00H
FFFFF704H	Port 2 function control expansion register	PFCE2		√	√		00H
FFFFF706H	Port 3 function control expansion register	PFCE3		√	√		00H
FFFFF708H	Port 4 function control expansion register	PFCE4		√	√		00H
FFFFF802H	System status register	SYS		√	√		00H
FFFFF810H	DMA trigger factor register 0	DTFR0		√	√		00H
FFFFF812H	DMA trigger factor register 1	DTFR1		√	√		00H
FFFFF814H	DMA trigger factor register 2	DTFR2		√	√		00H
FFFFF816H	DMA trigger factor register 3	DTFR3		√	√		00H
FFFFF820H	Power save mode register	PSMR		√	√		00H
FFFFF828H	Processor clock control register	PCC		√	√		03H
FFFFF82CH	PLL control register	PLLCTL		√	√		01H
FFFFF870H	Clock monitor mode register	CLM		√	√		00H
FFFFF888H	Reset source flag register	RESF		√	√		00H/10H/01H
FFFFF890H	Low-voltage detection register	LVIM		√	√		00H
FFFFF891H	Low-voltage detection level select register	LVIS			√		00H
FFFFFA00H	UARTA0 control register 0	UA0CTL0		√	√		10H
FFFFFA01H	UARTA0 control register 1	UA0CTL1			√		00H
FFFFFA02H	UARTA0 control register 2	UA0CTL2			√		FFH
FFFFFA03H	UARTA0 option control register 0	UA0OPT0		√	√		14H
FFFFFA04H	UARTA0 status register	UA0STR		√	√		00H
FFFFFA06H	UARTA0 receive data register	UA0RX	R		√		FFH

(12/14)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFFA07H	UARTA0 transmit data register	UA0TX	R/W		√		FFH
FFFFFA10H	UARTA1 control register 0	UA1CTL0		√	√		10H
FFFFFA11H	UARTA1 control register 1	UA1CTL1			√		00H
FFFFFA12H	UARTA1 control register 2	UA1CTL2			√		FFH
FFFFFA13H	UARTA1 option control register 0	UA1OPT0		√	√		14H
FFFFFA14H	UARTA1 status register	UA1STR		√	√		00H
FFFFFA16H	UARTA1 receive data register	UA1RX	R		√		FFH
FFFFFA17H	UARTA1 transmit data register	UA1TX	R/W		√		FFH
FFFFFA20H	UARTA2 control register 0	UA2CTL0		√	√		10H
FFFFFA21H	UARTA2 control register 1	UA2CTL1			√		00H
FFFFFA22H	UARTA2 control register 2	UA2CTL2			√		FFH
FFFFFA23H	UARTA2 option control register 0	UA2OPT0		√	√		14H
FFFFFA24H	UARTA2 status register	UA2STR	R/W	√	√		00H
FFFFFA26H	UARTA2 receive data register	UA2RX	R		√		FFH
FFFFFA27H	UARTA2 transmit data register	UA2TX	R/W		√		FFH
FFFFFA40H	UARTB control register 0	UBCTL0		√	√		10H
FFFFFA42H	UARTB control register 2	UBCTL2				√	FFFFH
FFFFFA44H	UARTB status register	UBSTR		√	√		00H
FFFFFA46H	UARTB receive data register AP	UBRXAP	R			√	00FFH
FFFFFA46H	UARTB receive data register	UBRX			√		FFH
FFFFFA48H	UARTB transmit data register	UBTX	W		√		FFH
FFFFFA4AH	UARTBFIFO control register 0	UBFIC0	R/W	√	√		00H
FFFFFA4BH	UARTBFIFO control register 1	UBFIC1		√	√		00H
FFFFFA4CH	UARTBFIFO control register 2	UBFIC2				√	0000H
FFFFFA4CH	UARTBFIFO control register 2L	UBFIC2L			√		00H
FFFFFA4DH	UARTBFIFO control register 2H	UBFIC2H			√		00H
FFFFFA4EH	UARTBFIFO status register 0	UBFIS0	R		√		00H
FFFFFA4FH	UARTBFIFO status register 1	UBFIS1			√		10H
FFFFFB00H	TAA3 control register 0	TAA3CTL0	R/W	√	√		00H
FFFFFB01H	TAA3 control register 1	TAA3CTL1		√	√		00H
FFFFFB02H	TAA3 I/O control register 0	TAA3IOC0[Note]		√	√		00H
FFFFFB03H	TAA3 I/O control register 1	TAA3IOC1[Note]		√	√		00H
FFFFFB04H	TAA3 I/O control register 2	TAA3IOC2[Note]		√	√		00H
FFFFFB05H	TAA3 option register 0	TAA3OPT0		√	√		00H
FFFFFB06H	TAA3 capture/compare register 0	TAA3CCR0				√	0000H
FFFFFB08H	TAA3 capture/compare register 1	TAA3CCR1				√	0000H
FFFFFB0AH	TAA3 counter read buffer register	TAA3CNT	R			√	0000H
FFFFFB20H	TAA4 control register 0	TAA4CTL0	R/W	√	√		00H
FFFFFB21H	TAA4 control register 1	TAA4CTL1		√	√		00H
FFFFFB22H	TAA4 I/O control register 0	TAA4IOC0		√	√		00H
FFFFFB23H	TAA4 I/O control register 1	TAA4IOC1		√	√		00H

Note V850E/IG3 only

(13/14)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFFB24H	TAA4 I/O control register 2	TAA4IOC2	R/W	√	√		00H
FFFFFB25H	TAA4 option register 0	TAA4OPT0		√	√		00H
FFFFFB26H	TAA4 capture/compare register 0	TAA4CCR0				√	0000H
FFFFFB28H	TAA4 capture/compare register 1	TAA4CCR1				√	0000H
FFFFFB2AH	TAA4 counter read buffer register	TAA4CNT	R			√	0000H
FFFFFB40H	Digital noise elimination 1 control register 2	TANFC2	R/W		√		00H
FFFFFB42H	Digital noise elimination 1 control register 3	TANFC3[Note]			√		00H
FFFFFB44H	Digital noise elimination 1 control register 4	TANFC4			√		00H
FFFFFB80H	A/D converter 2 mode register 0	AD2M0		√	√		00H
FFFFFB81H	A/D converter 2 mode register 1	AD2M1		√	√		00H
FFFFFB82H	A/D converter 2 channel specification register	AD2S		√	√		00H
FFFFFB90H	A/D2 conversion result register 0	AD2CR0	R			√	0000H
FFFFFB91H	A/D2 conversion result register 0H	AD2CR0H			√		00H
FFFFFB92H	A/D2 conversion result register 1	AD2CR1				√	0000H
FFFFFB93H	A/D2 conversion result register 1H	AD2CR1H			√		00H
FFFFFB94H	A/D2 conversion result register 2	AD2CR2				√	0000H
FFFFFB95H	A/D2 conversion result register 2H	AD2CR2H			√		00H
FFFFFB96H	A/D2 conversion result register 3	AD2CR3				√	0000H
FFFFFB97H	A/D2 conversion result register 3H	AD2CR3H			√		00H
FFFFFB98H	A/D2 conversion result register 4	AD2CR4[Note]				√	0000H
FFFFFB99H	A/D2 conversion result register 4H	AD2CR4H[Note]			√		00H
FFFFFB9AH	A/D2 conversion result register 5	AD2CR5[Note]				√	0000H
FFFFFB9BH	A/D2 conversion result register 5H	AD2CR5H[Note]			√		00H
FFFFFB9CH	A/D2 conversion result register 6	AD2CR6[Note]				√	0000H
FFFFFB9DH	A/D2 conversion result register 6H	AD2CR6H[Note]			√		00H
FFFFFB9EH	A/D2 conversion result register 7	AD2CR7[Note]				√	0000H
FFFFFB9FH	A/D2 conversion result register 7H	AD2CR7H[Note]			√		00H
FFFFFBB0H	Port 7 register	P7	R	√	√		Undefined
FFFFFBB8H	Port 7 mode control register	PMC7	R/W	√	√		00H
FFFFFC00H	External interrupt falling edge specification register 0	INTF0		√	√		00H
FFFFFC02H	External interrupt falling edge specification register 1	INTF1		√	√		00H
FFFFFC04H	External interrupt falling edge specification register 2	INTF2		√	√		00H
FFFFFC20H	External interrupt rising edge specification register 0	INTR0		√	√		00H
FFFFFC22H	External interrupt rising edge specification register 1	INTR1		√	√		00H
FFFFFC24H	External interrupt rising edge specification register 2	INTR2		√	√		00H
FFFFFC40H	Pull-up resistor option register 0	PU0		√	√		00H
FFFFFC42H	Pull-up resistor option register 1	PU1		√	√		00H
FFFFFC44H	Pull-up resistor option register 2	PU2		√	√		00H
FFFFFC46H	Pull-up resistor option register 3	PU3		√	√		00H
FFFFFC48H	Pull-up resistor option register 4	PU4		√	√		00H

Note V850E/IG3 only

(14/14)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFFC66H	Port 3 function register	PF3	R/W	√	√		00H
FFFFFD00H	CSIB0 control register 0	CB0CTL0		√	√		01H
FFFFFD01H	CSIB0 control register 1	CB0CTL1		√	√		00H
FFFFFD02H	CSIB0 control register 2	CB0CTL2			√		00H
FFFFFD03H	CSIB0 status register	CB0STR		√	√		00H
FFFFFD04H	CSIB0 receive data register	CB0RX	R			√	0000H
FFFFFD04H	CSIB0 receive data register L	CB0RXL			√		00H
FFFFFD06H	CSIB0 transmit data register	CB0TX	R/W			√	0000H
FFFFFD06H	CSIB0 transmit data register L	CB0TXL			√		00H
FFFFFD10H	CSIB1 control register 0	CB1CTL0		√	√		01H
FFFFFD11H	CSIB1 control register 1	CB1CTL1		√	√		00H
FFFFFD12H	CSIB1 control register 2	CB1CTL2			√		00H
FFFFFD13H	CSIB1 status register	CB1STR		√	√		00H
FFFFFD14H	CSIB1 receive data register	CB1RX	R			√	0000H
FFFFFD14H	CSIB1 receive data register L	CB1RXL			√		00H
FFFFFD16H	CSIB1 transmit data register	CB1TX	R/W			√	0000H
FFFFFD16H	CSIB1 transmit data register L	CB1TXL			√		00H
FFFFFD20H	CSIB2 control register 0	CB2CTL0		√	√		01H
FFFFFD21H	CSIB2 control register 1	CB2CTL1		√	√		00H
FFFFFD22H	CSIB2 control register 2	CB2CTL2			√		00H
FFFFFD23H	CSIB2 status register	CB2STR		√	√		00H
FFFFFD24H	CSIB2 receive data register	CB2RX	R			√	0000H
FFFFFD24H	CSIB2 receive data register L	CB2RXL			√		00H
FFFFFD26H	CSIB2 transmit data register	CB2TX	R/W			√	0000H
FFFFFD26H	CSIB2 transmit data register L	CB2TXL			√		00H
FFFFFD80H	IIC shift register 0	IIC0			√		00H
FFFFFD82H	IIC control register 0	IICC0		√	√		00H
FFFFFD83H	Slave address register 0	SVA0			√		00H
FFFFFD84H	IIC clock select register 0	IICCL0		√	√		00H
FFFFFD85H	IIC function expansion register 0	IICX0		√	√		00H
FFFFFD86H	IIC status register 0	IICS0	R	√	√		00H
FFFFFD8AH	IIC flag register 0	IICF0	R/W	√	√		00H
FFFFFD90H	IIC OPS clock select register	IICOCKS			√		00H
FFFFFF44H	Pull-up resistor option register DL	PUDL				√	0000H
FFFFFF44H	Pull-up resistor option register DLL	PUDLL		√	√		00H
FFFFFF45H	Pull-up resistor option register DLH	PUDLH		√	√		00H

3.4.8 Special registers

Special registers are registers that are protected from being written with illegal data due to a program loop.

The V850E/IF3 and V850E/IG3 have the following four special registers.

- Power save control register (PSC)
- Processor clock control register (PCC)
- Reset source flag register (RESF)
- Clock monitor mode register (CLM)
- Low-voltage detection register (LVIM)

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

(1) Setting data to special registers

Set data to the special registers in the following sequence.

<1> Prepare data to be set to the special register in a general-purpose register.
<2> Write the data prepared in <1> to the command register.
<3> Write the setting data to the special register (by using the following instructions).
- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

(<4> to <8> Insert NOP instructions (5 instructions).)[Note]

[Example] With PSC register (setting standby mode)

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

There is no special sequence to read a special register.

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

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

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

An example of setting the special register (<3> in Example) by using the bit manipulation instruction is shown below.

```
CLR1 4, RESF[r0]
```

3. Before executing this processing, terminate all DMA transfer operations.

(2) Command register (PRCMD)

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

An illegal write operation to a special register can be checked by using the SYS.PRERR bit.

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

Reset makes this register undefined.

After reset: Undefined W Address: FFFFF1FCH

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

(3) System status register (SYS)

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

If this register is not written in the correct sequence including an access to the PRCMD register, data is not written to the intended register, a protection error occurs, and the PRERR flag is set. This register is cleared by writing "0" to it by an instruction from CPU.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF802H

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

PRERR	Protection error detection
0	Protection error did not occur.
1	Protection error occurred.

The PRERR flag operates under the following conditions.

(a) Set condition (PRERR flag = 1)

- When data is written to a special register without writing anything to the PRCMD register (when <3> is executed without executing <2> in **3.4.8 (1) Setting data to special registers**)
- When data is written to an on-chip peripheral I/O register other than a special register (including execution of a bit manipulation instruction) after writing data to the PRCMD register (if <3> in **3.4.8 (1) Setting data to special registers** is not the setting of a special register)

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

(b) Clear condition (PRERR flag = 0)

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

(ii) When the system is reset

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

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

3.4.9 System wait control register (VSWC)

The VSWC register is a register that controls the bus access wait for the on-chip peripheral I/O registers.

Access to on-chip peripheral I/O registers of the V850E1 CPU core is basically made in 3 clocks; however, in the V850E/IF3 and V850E/IG3, a wait set by the VSWC register is required in addition to those 3 clocks. Set 13H (set wait for 4 clocks) to VSWC.

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

CPU Clock Frequency (f_{CPU})	VSWC Set Value
500 kHz ≤ f_{CPU} ≤ 64 MHz	13

Caution When using the V850E/IF3 and V850E/IG3, the VSWC register must be set first. Set other registers if necessary after setting the VSWC register.

Remark When a register includes status flags that indicate the statuses of the on-chip peripheral functions (such as UAnSTR) or a register that indicates the count value of a timer (such as TAAnCNT) is accessed, a register access retry operation takes place if the timing at which the flag and count value changes and the timing of the register access overlap. Consequently, access to the on-chip peripheral I/O register may take a long time.

CHAPTER 4 PORT FUNCTIONS

4.1 Features

4.1.1 V850E/IF3

- ○ Input-only ports: 4
  I/O ports: 44
- ○ Input data read/output data write is enabled in 1-bit units.
- ○ On-chip pull-up resistor can be connected in 1-bit units (ports 0 to 4 and DL only).
  However, an on-chip pull-up resistor can only be connected when the pins are in input mode in the port mode, or when the pins function as input pins in the alternate-function mode. Moreover, an on-chip pull-up resistor can be connected to the TOB0T1 to TOB0T3, TOB0B1 to TOB0B3, and TOA21 pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOB0OFF, TOA2OFF, or TOB1OFF pin or software processing.

4.1.2 V850E/IG3

- ○ Input-only ports: 8
  I/O ports: 56
- ○ Input data read/output data write is enabled in 1-bit units.
- ○ On-chip pull-up resistor can be connected in 1-bit units (ports 0 to 4 and DL only).
  However, an on-chip pull-up resistor can only be connected when the pins are in input mode in the port mode, or when the pins function as input pins in the alternate-function mode. Moreover, an on-chip pull-up resistor can be connected to the TOB0T1 to TOB0T3, TOB0B1 to TOB0B3, TOA21, TOB1T1 to TOB1T3, TOB1B1 to TOB1B3, and TOA31 pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOB0OFF, TOB1OFF, TOA2OFF, or TOA3OFF pin or software processing.

4.2 Port Configuration

4.2.1 V850E/IF3

The V850E/IF3 incorporates a total of 48 input/output ports (including 4 input-only ports) labeled ports 0 to 4, 7, and DL. The port configuration is shown in Figure 4-1.

There are two power supply systems for the I/O buffer of a pin: AV_{DD2} and EV_{DD0}, EV_{DD1}. The relationship between each of these power supplies and the pin is shown in Table 4-1.

Figure 4-1. Port Configuration (V850E/IF3)

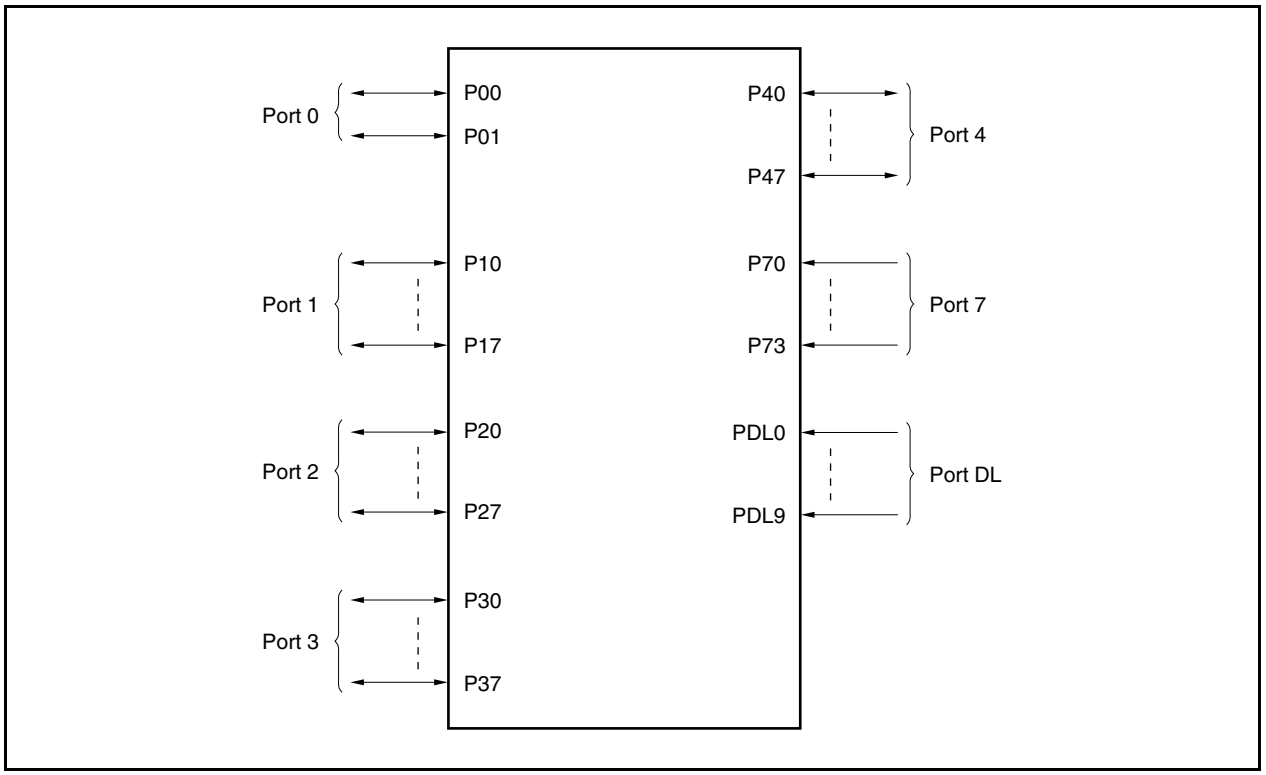


Table 4-1. Power Supplies for I/O Buffer of Each Pin (V850E/IF3)

Power Supply	Corresponding Pins
AV_{DD2}	P70 to P73
EV_{DD0}, EV_{DD1}	P00, P01, P10 to P17, P20 to P27, P30 to P37, P40 to P47, PDL0 to PDL9, $\overline{RESET}$

4.2.2 V850E/IG3

The V850E/IG3 incorporates a total of 64 input/output ports (including 8 input-only ports) labeled ports 0 to 4, 7, and DL. The port configuration is shown in Figure 4-2.

There are two power supply systems for the I/O buffer of a pin: AV_{DD2} and EV_{DD0}, EV_{DD1}, EV_{DD2}. The relationship between each of these power supplies and the pin is shown in Table 4-2.

Figure 4-2. Port Configuration (V850E/IG3)

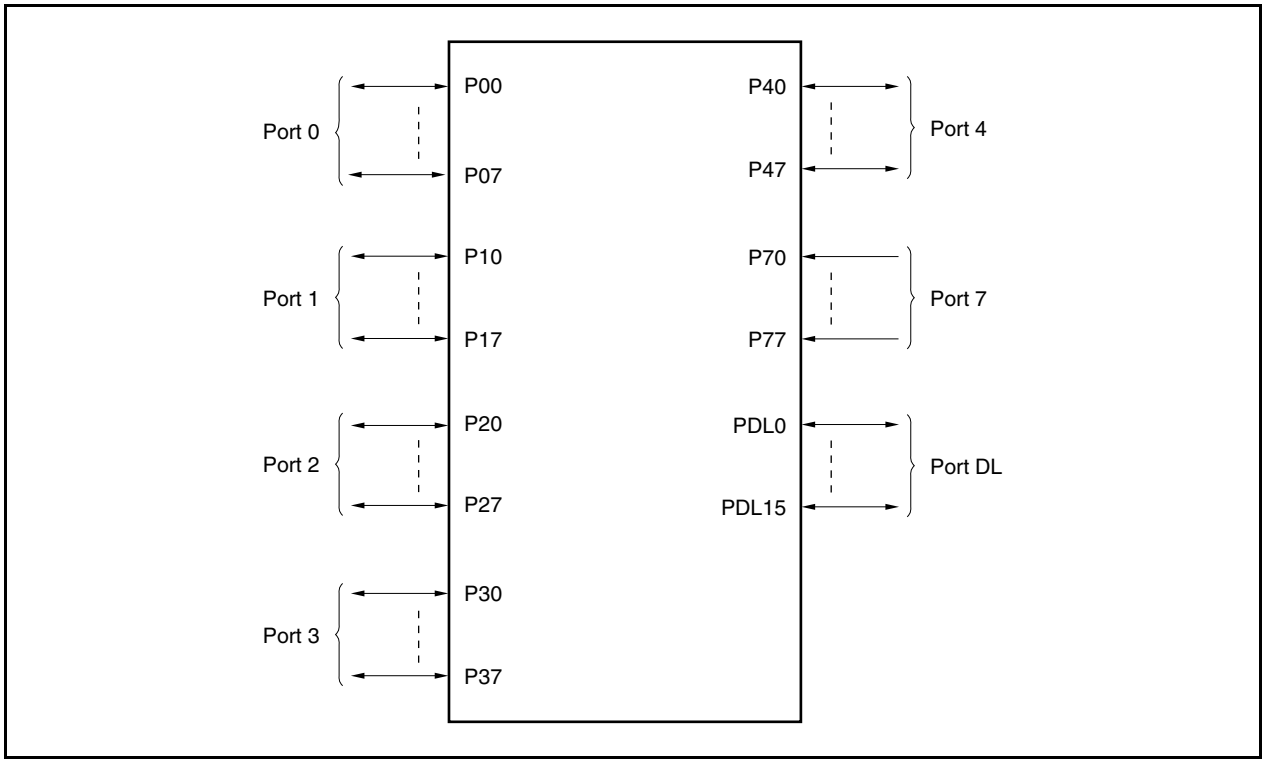


Table 4-2. Power Supplies for I/O Buffer of Each Pin (V850E/IG3)

Power Supply	Corresponding Pins
AV_{DD2}	P70 to P77
EV_{DD0}, EV_{DD1}, EV_{DD2}	P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, PDL0 to PDL15, $\overline{\text{RESET}}$, DCK, DDI, DDO, DMS, $\overline{\text{DRST}}$

4.3 Port Configuration

Table 4-3. Port Configuration (V850E/IF3)

Item	Configuration
Control registers	Port n register (Pn: n = 0 to 4, 7, DL) Port n mode register (PMn: n = 0 to 4, DL) Port n mode control register (PMCn: n = 0 to 4, 7, DL) Port n function control register (PFCn: n = 0 to 4) Port n function control expansion register (PFCEn: n = 0 to 4) Pull-up resistor option register (PUn: n = 0 to 4, DL) Port 3 function register (PF3)
Ports	Input-only: 4, I/O: 44
Pull-up resistor	Software control: 44

Table 4-4. Port Configuration (V850E/IG3)

Item	Configuration
Control registers	Port n register (Pn: n = 0 to 4, 7, DL) Port n mode register (PMn: n = 0 to 4, DL) Port n mode control register (PMCn: n = 0 to 4, 7, DL) Port n function control register (PFCn: n = 0 to 4) Port n function control expansion register (PFCEn: n = 0 to 4) Pull-up resistor option register (PUn: n = 0 to 4, DL) Port 3 function register (PF3)
Ports	Input-only: 8, I/O: 56
Pull-up resistor	Software control: 56

(1) Port n register (Pn)

Data is input from or output to an external device by writing or reading the Pn register.

The Pn register consists of a port latch that holds output data, and a circuit that reads the status of pins.

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

After reset: Undefined R/W

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

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

Data is written to or read from the Pn register as follows, regardless of the setting of the PMCn register.

Table 4-5. Writing/Reading Pn Register

Setting of PMn Register	Writing to Pn Register	Reading from Pn Register
Output mode (PMnm = 0)	Data is written to the output latch[Note 1]. In the port mode (PMCn = 0), the contents of the output latch are output from the pins.	The value of the output latch is read[Note 2].
Input mode (PMnm = 1)	Data is written to the output latch. The pin status is not affected[Note 1].	The pin status is read[Note 3].

Notes 1. The value written to the output latch is retained until a new value is written to the output latch.

2. Also, the value of the Pn register is read when the PMn register is in the output mode while the alternate function is set.

3. If the PMn register is in the input mode while the alternate function is set, the statuses of the pins at that time are read regardless of whether the alternate function is an input or output function.

(2) Port n mode register (PMn)

The PMn register specifies the input or output mode of the corresponding port pin.

Each bit of this register corresponds to one pin of port n, and the input or output mode can be specified in 1-bit units.

After reset: FFH R/W

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

PMnm	Control of I/O mode
0	Output mode
1	Input mode

(3) Port n mode control register (PMCn)

The PMCn register specifies the port mode or alternate function.

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

After reset: 00H R/W

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

PMCnm	Specification of operation mode
0	Port mode
1	Alternate function

(4) Port n function control register (PFCn)

The PFCn register specifies the alternate function of a port pin to be used if the pin has two alternate functions.

Each bit of this register corresponds to one pin of port n, and the alternate function of a port pin can be specified in 1-bit units.

After reset: 00H R/W

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

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

(5) Port n function control expansion register (PFCEn)

The PFCEn register specifies the alternate function of a port pin to be used if the pin has three or more alternate functions.

Each bit of this register corresponds to one pin of port n, and the alternate function of a port pin can be specified in 1-bit units.

After reset: 00H R/W

	7	6	5	4	3	2	1	0
PFCEn	PFCEn7	PFCEn6	PFCEn5	PFCEn4	PFCEn3	PFCEn2	PFCEn1	PFCEn0
PFCn	PFCn7	PFCn6	PFCn5	PFCn4	PFCn3	PFCn2	PFCn1	PFCn0

PFCEnm	PFCnm	Specification of alternate function
0	0	Alternate function 1
0	1	Alternate function 2
1	0	Alternate function 3
1	1	Alternate function 4

(6) Pull-up resistor option register (PUn)

PUn is a register that specifies the connection of an on-chip pull-up resistor.

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

After reset: 00H R/W

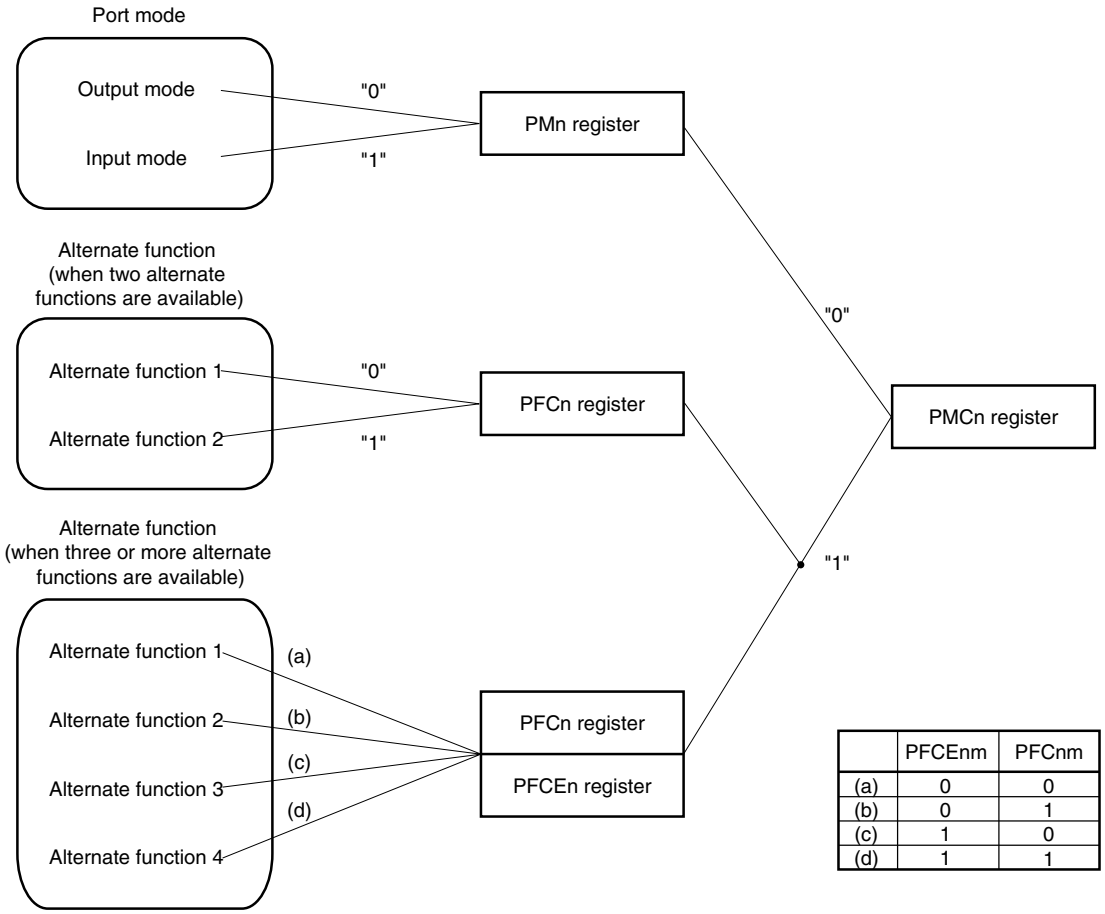
	7	6	5	4	3	2	1	0
PUn	PUn7	PUn6	PUn5	PUn4	PUn3	PUn2	PUn1	PUn0

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

(7) Port settings

Set the ports as follows.

Figure 4-3. Register Settings and Pin Functions



	PFCEnm	PFCnm
(a)	0	0
(b)	0	1
(c)	1	0
(d)	1	1

Caution To switch to external interrupt input (INTPn) from the port mode (by changing the PMCa.PMCam bit from 0 to 1), an external interrupt may be input if a wrong valid edge is detected. Therefore, be sure to set "no edge detection" by INTRk, INTFk, ADTR, or ADTF register, select external interrupt input (INTPn), and then specify the valid edge (V850E/IF3: n = 00, 01, 08 to 18, ADT0, ADT1, a = 0 to 4, m = 0 to 7, k = 0 to 2, V850E/IG3: n = 00 to 18, ADT0, ADT1, a = 0 to 4, m = 0 to 7, k = 0 to 2)).

When switching to the port mode from external interrupt input (INTPn) (by changing the PMCam bit = from 1 to 0), an edge may be detected. Therefore, be sure to set "no edge detection" by INTRk, INTFk, ADTR, or ADTF register, and then select the port mode.

Remark Switch to the alternate function using the following procedure (for n, see **Tables 4-3** and **4-4**).

<1> Set the PFCn and PFCEn registers.

<2> Set the PMCn register.

<3> Set the INTRk, INTFk, ADTR, and ADTF registers (when external interrupt pin is set).

If the PMCn register is set before setting the PFCn and PFCEn registers, an unexpected peripheral function may be selected while the PFCn and PFCEn registers are being set.

4.3.1 Port 0

Port 0 can be set to the input or output mode in 1-bit units.

The number of I/O pins differs from one product to another.

Generic Name	Number of I/O Ports
V850E/IF3	2-bit I/O port
V850E/IG3	8-bit I/O port

Port 0 has an alternate function as the following pins.

Table 4-6. Alternate-Function Pins of Port 0

Pin Name	Pin No.			Alternate-Function Pin Name	I/O	Pull-Up[Note 1]
	IF3	IG3				
	GC	GC	GF			
P00	70	91	19	TOA20/TIA20/TOA2OFF/INTP00	I/O	Provided
P01	69	90	18	TOA21/TIA21/INTP01	I/O	
P02[Note 2]	–	89	17	TOA30[Note 2]/TIA30[Note 2]/TOA3OFF[Note 2]/INTP02[Note 2]	I/O	
P03[Note 2]	–	88	16	TOA31[Note 2]/TIA31[Note 2]/INTP03[Note 2]	I/O	
P04[Note 2]	–	84	12	TECR0[Note 2]/TIT00[Note 2]/TOT00[Note 2]/INTP04[Note 2]	I/O	
P05[Note 2]	–	83	11	TENC00[Note 2]/EVTT0[Note 2]/INTP05[Note 2]	Input	
P06[Note 2]	–	82	10	TENC01[Note 2]/TIT01[Note 2]/TOT01[Note 2]/INTP06[Note 2]	I/O	
P07[Note 2]	–	63	91	INTP07[Note 2]/CLKOUT[Note 3]	I/O	

Remark IF3: V850E/IF3
IG3: V850E/IG3
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

Notes 1. Software pull-up function
2. V850E/IG3 only
3. μPD70F3454GC-8EA-A only

Cautions 1. To control the high-impedance output of a timer for motor control, be sure to set the PMC0.PMC0n bit to 1 and then specify the edge to be detected and enable the operation of the high-impedance output controller, because the output of the motor control timer may go into a high-impedance state if a wrong valid edge is detected (V850E/IF3: n = 0, V850E/IG3: n = 0, 2).

2. When P01 and P03 (V850E/IG3 only) are used as TOA21 and TOA31 (V850E/IG3 only), they go into a high-impedance state by inputting the following active signal.

- **Output of high impedance setting signal from high impedance output controller**
- **Output of clock stop detection signal from clock monitor**

Cautions 3. To switch to external interrupt input (INTP0n) from the port mode (by changing the PMC0.PMC0n bit from 0 to 1), an external interrupt may be input if a wrong valid edge is detected. Therefore, be sure to disable edge detection (INTF0.INTF0n bit = 0 and INTR0.INTR0n bit = 0), select external interrupt input (INTP0n), and then specify the valid edge (V850E/IF3: n = 0, 1, V850E/IG3: n = 0 to 7).

When switching to the port mode from external interrupt input (INTP0n) (by changing the PMC0n bit from 1 to 0), an edge may be detected. Therefore, be sure to disable edge detection (INTF0n bit = 0, INTR0n bit = 0), and then select the port mode.

4. To control high-impedance output of the external interrupt function and motor output control function, set the PMC0n bit to 1 (V850E/IF3: n = 0, 1, V850E/IG3: n = 0 to 7).

(1) Registers

(a) Port 0 register (P0)

After reset: Undefined R/W Address: FFFFF400H

	7	6	5	4	3	2	1	0
P0	P07[Note]	P06[Note]	P05[Note]	P04[Note]	P03[Note]	P02[Note]	P01	P00

P0n	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Note Valid only for the V850E/IG3.
With the V850E/IF3, the read value of this register is undefined.

Remark V850E/IF3: n = 0, 1
V850E/IG3: n = 0 to 7

(b) Port 0 mode register (PM0)

After reset: FFH R/W Address: FFFFF420H

	7	6	5	4	3	2	1	0
PM0	PM07[Note]	PM06[Note]	PM05[Note]	PM04[Note]	PM03[Note]	PM02[Note]	PM01	PM00

PM0n	Control of I/O mode (in port mode)
0	Output mode
1	Input mode

Note Valid only for the V850E/IG3.
With the V850E/IF3, be sure to set this bit to 1.

Remark V850E/IF3: n = 0, 1
V850E/IG3: n = 0 to 7

(c) Port 0 mode control register (PMC0)

After reset: 00H R/W Address: FFFFF440H

	7	6	5	4	3	2	1	0
PMC0	PMC07[Note 1]	PMC06[Note 1]	PMC05[Note 1]	PMC04[Note 1]	PMC03[Note 1]	PMC02[Note 1]	PMC01	PMC00

PMC07[Note 1]	Specification of operating mode of P07 pin
0	I/O port
1	INTP07 input[Note 2]/CLKOUT output[Note 3]

PMC06[Note 1]	Specification of operating mode of P06 pin
0	I/O port
1	TENC01 input[Note 2]/TIT01 input[Note 2]/TOT01 output[Note 2]/INTP06 input[Note 1]

PMC05[Note 1]	Specification of operating mode of P05 pin
0	I/O port
1	TENC00 input[Note 2]/EVTT0 input[Note 2]/INTP05 input[Note 2]

PMC04[Note 1]	Specification of operating mode of P04 pin
0	I/O port
1	TECR0 input[Note 2]/TIT00 input[Note 2]/TOT00 output[Note 2]/INTP04 input[Note 2]

PMC03[Note 1]	Specification of operating mode of P03 pin
0	I/O port
1	TOA31 output[Note 2]/TIA31 input[Note 2]/INTP03 input[Note 2]

PMC02[Note 1]	Specification of operating mode of P02 pin
0	I/O port
1	TOA30 output[Note 2]/TIA30 input[Note 2]/TOA3OFF input[Note 2]/INTP02 input[Note 2]

PMC01	Specification of operating mode of P01 pin
0	I/O port
1	TOA21 output/TIA21 input/INTP01 input

PMC00	Specification of operating mode of P00 pin
0	I/O port
1	TOA20 output/TIA20 input/TOA2OFF input/INTP00 input

Notes 1. Valid only in the V850E/IG3.
With the V850E/IF3, be sure to set these bits to 0.

2. V850E/IG3 only

3. μPD70F3454GC-8EA-A only

(d) Port 0 function control register (PFC0)

After reset: 00H R/W Address: FFFFF460H

	7	6	5	4	3	2	1	0
PFC0	PFC07[Note]	PFC06[Note]	PFC05[Note]	PFC04[Note]	PFC03[Note]	PFC02[Note]	PFC01	PFC00

Note Valid only in the V850E/IG3.
With the V850E/IF3, be sure to set these bits to 0.

Remark For the specifications of alternate functions, see **4.3.1 (1) (f) Settings of alternate functions of port 0**.

(e) Port 0 function control expansion register (PFCE0)

After reset: 00H R/W Address: FFFFF700H

	7	6	5	4	3	2	1	0
PFCE0	0	PFCE06[Note]	PFCE05[Note]	PFCE04[Note]	PFCE03[Note]	PFCE02[Note]	PFCE01	PFCE00

Note Valid only in the V850E/IG3.
With the V850E/IF3, be sure to set these bits to 0.

Remark For the specifications of alternate functions, see **4.3.1 (1) (f) Settings of alternate functions of port 0**.

(f) Setting of alternate function of port 0

PFC07[Note 1]	Specification of Alternate Function of P07[Note 1] Pin
0	INTP07 input[Note 1]
1	CLKOUT output[Note 2]

PFCE06[Note 1]	PFC06[Note 1]	Specification of Alternate Function of P06[Note 1] Pin
0	0	TENC01 input[Note 1]/TIT01 input[Note 1] (two functions are alternately used)
0	1	TOT01 output[Note 1]
1	0	INTP06 input[Note 1]
1	1	Setting prohibited

PFCE05[Note 1]	PFC05[Note 1]	Specification of Alternate Function of P05[Note 1] Pin
0	0	TENC00 input[Note 1]
0	1	EVTT0 input[Note 1]
1	0	INTP05 input[Note 1]
1	1	Setting prohibited

PFCE04[Note 1]	PFC04[Note 1]	Specification of Alternate Function of P04[Note 1] Pin
0	0	TECR0 input[Note 1]/TIT00 input[Note 1] (two functions are alternately used)
0	1	TOT00 output[Note 1]
1	0	INTP04 input[Note 1]
1	1	Setting prohibited

PFCE03[Note 1]	PFC03[Note 1]	Specification of Alternate Function of P03[Note 1] Pin
0	0	TOA31 output[Note 1]
0	1	TIA31 input[Note 1]
1	0	INTP03 input[Note 1]
1	1	Setting prohibited

PFCE02[Note 1]	PFC02[Note 1]	Specification of Alternate Function of P02[Note 1] Pin
0	0	TOA30 output[Note 1]
0	1	TIA30 input[Note 1]
1	0	TOA3OFF input[Note 1]/INTP02 input[Note 1] (two functions are alternately used)
1	1	Setting prohibited

Notes 1. V850E/IG3 only

2. μPD70F3454GC-8EA-A only

PFCE01	PFC01	Specification of Alternate Function of P01 Pin
0	0	TOA21 output
0	1	TIA21 input
1	0	INTP01 input
1	1	Setting prohibited

PFCE00	PFC00	Specification of Alternate Function of P00 Pin
0	0	TOA20 output
0	1	TIA20 input
1	0	TOA2OFF input/INTP00 input
1	1	Setting prohibited

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

After reset: 00H R/W Address: FFFFFC40H

	7	6	5	4	3	2	1	0
PU0	PU07Note 1	PU06Note 1	PU05Note 1	PU04Note 1	PU03Note 1	PU02Note 1	PU01	PU00

PU0n	Control of on-chip pull-up resistor connection
0	Do not connect
1	ConnectNote 2

Notes 1. Valid only in the V850E/IG3.
With the V850E/IF3, be sure to set this bit to 0.

2. An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode. Moreover, an on-chip pull-up resistor can be connected to the TOA21 and TOA31 (V850E/IG3 only) pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOA2OFF or TOA3OFF (V850E/IG3 only) pin, or software processing. An on-chip pull-up resistor cannot be connected when the pins are in output mode.

Remark V850E/IF3: n = 0, 1
V850E/IG3: n = 0 to 7

4.3.2 Port 1

Port 1 can be set to the input or output mode in 1-bit units.

Port 1 has an alternate function as the following pins.

Table 4-7. Alternate-Function Pins of Port 1

Pin Name	Pin No.			Alternate-Function Pin Name	I/O	Pull-Up[Note 1]
	IF3	IG3				
	GC	GC	GF			
P10	78	99	27	TOB0T1/TIB01/TOB01/A0[Note 2]	I/O	Provided
P11	77	98	26	TOB0B1/TIB02/TOB02/A1[Note 2]	I/O	
P12	76	97	25	TOB0T2/TIB03/TOB03/A2[Note 2]	I/O	
P13	75	96	24	TOB0B2/TIB00/A3[Note 2]	I/O	
P14	74	95	23	TOB0T3/EVTB0/A4[Note 2]	I/O	
P15	73	94	22	TOB0B3/TRGB0/A5[Note 2]	I/O	
P16	72	93	21	TOB0OFF/INTP08/ADTRG0/INTADT0/A6[Note 2]	I/O	
P17	71	92	20	TOB00/INTP09/A7[Note 2]	I/O	

Notes 1. Software pull-up function

2. μPD70F3454GC-8EA-A only

Caution When P10 to P15 are used as TOB0T1 to TOB0T3 and TOB0B1 to TOB0B3, they go into a high-impedance state by inputting the following active signal.

- **Output of high impedance setting signal from high impedance output controller**
- **Output of clock stop detection signal from clock monitor**

Remark IF3: V850E/IF3

IG3: V850E/IG3

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(1) Registers

(a) Port 1 register (P1)

After reset: Undefined R/W Address: FFFFF402H

	7	6	5	4	3	2	1	0
P1	P17	P16	P15	P14	P13	P12	P11	P10

P1n	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Remark n = 0 to 7

(b) Port 1 mode register (PM1)

After reset: FFH R/W Address: FFFFF422H

	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	Control of I/O mode (in port mode)
0	Output mode
1	Input mode

Remark n = 0 to 7

(c) Port 1 mode control register (PMC1)

After reset: 00H R/W Address: FFFFF442H

	7	6	5	4	3	2	1	0
PMC1	PMC17	PMC16	PMC15	PMC14	PMC13	PMC12	PMC11	PMC10

PMC17	Specification of operating mode of P17 pin
0	I/O port
1	TOB00 output/INTP09 input/A7 output[Note]

PMC16	Specification of operating mode of P16 pin
0	I/O port
1	TOB0OFF input/INTP08 input/ADTRG0 input/INTADT0 input/A6 output[Note]

PMC15	Specification of operating mode of P15 pin
0	I/O port
1	TOB0B3 output/TRGB0 input/A5 output[Note]

PMC14	Specification of operating mode of P14 pin
0	I/O port
1	TOB0T3 output/EVTB0 input/A4 output[Note]

PMC13	Specification of operating mode of P13 pin
0	I/O port
1	TOB0B2 output/TIB00 input/A3 output[Note]

PMC12	Specification of operating mode of P12 pin
0	I/O port
1	TOB0T2 output/TIB03 input/TOB03 output/A2 output[Note]

PMC11	Specification of operating mode of P11 pin
0	I/O port
1	TOB0B1 output/TIB02 input/TOB02 output/A1 output[Note]

PMC10	Specification of operating mode of P10 pin
0	I/O port
1	TOB0T1 output/TIB01 input/TOB01output/A0 output[Note]

Note μPD70F3454GC-8EA-A only

(d) Port 1 function control register (PFC1)

After reset: 00H R/W Address: FFFFF462H

	7	6	5	4	3	2	1	0
PFC1	PFC17	PFC16	PFC15	PFC14	PFC13	PFC12	PFC11	PFC10

Remark For the specifications of alternate functions, see **4.3.2 (1) (f) Settings of alternate functions of port 1**.

(e) Port 1 function control expansion register (PFCE1)

After reset: 00H R/W Address: FFFFF702H

	7	6	5	4	3	2	1	0
PFCE1	PFCE17	PFCE16	PFCE15	PFCE14	PFCE13	PFCE12	PFCE11	PFCE10

Remark For the specifications of alternate functions, see **4.3.2 (1) (f) Settings of alternate functions of port 1**.

(f) Settings of alternate functions of port 1

PFCE17	PFC17	Specification of Alternate Function of P17 Pin
0	0	TOB00 output
0	1	INTP09 input
1	0	A7 output[Note]
1	1	Setting prohibited

PFCE16	PFC16	Specification of Alternate Function of P16 Pin
0	0	TOB0OFF input/INTP08 input (two functions are alternately used)
0	1	ADTRG0 input/INTADT0 input (two functions are alternately used)
1	0	A6 output[Note]
1	1	Setting prohibited

PFCE15	PFC15	Specification of Alternate Function of P15 Pin
0	0	TOB0B3 output
0	1	TRGB0 input
1	0	A5 output[Note]
1	1	Setting prohibited

PFCE14	PFC14	Specification of Alternate Function of P14 Pin
0	0	TOB0T3 output
0	1	EVTB0 input
1	0	A4 output[Note]
1	1	Setting prohibited

PFCE13	PFC13	Specification of Alternate Function of P13 Pin
0	0	TOB0B2 output
0	1	TIB00 input
1	0	A3 output[Note]
1	1	Setting prohibited

PFCE12	PFC12	Specification of Alternate Function of P12 Pin
0	0	TOB0T2 output
0	1	TIB03 input
1	0	TOB03 output
1	1	A2 output[Note]

Note μPD70F3454GC-8EA-A only

PFCE11	PFC11	Specification of Alternate Function of P11 Pin
0	0	TOB0B1 output
0	1	TIB02 input
1	0	TOB02 output
1	1	A1 output[Note]

PFCE10	PFC10	Specification of Alternate Function of P10 Pin
0	0	TOB0T1 output
0	1	TIB01 input
1	0	TOB01 output
1	1	A0 output[Note]

Note μPD70F3454GC-8EA-A only

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

After reset: 00H R/W Address: FFFFFC42H

	7	6	5	4	3	2	1	0
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10

PU1n	Control of on-chip pull-up resistor connection
0	Do not connect
1	Connect[Note]

Note An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode. Moreover, an on-chip pull-up resistor can be connected to the TOB0T1 to TOB0T3 and TOB0B1 to TOB0B3 pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOB0OFF pin, or software processing. An on-chip pull-up resistor cannot be connected when the pins are in output mode.

Remark n = 0 to 7

4.3.3 Port 2

Port 2 can be set to the input or output mode in 1-bit units.

Port 2 has an alternate function as the following pins.

Table 4-8. Alternate-Function Pins of Port 2

Pin Name	Pin No.			Alternate-Function Pin Name	I/O	Pull-Up[Note 1]
	IF3	IG3				
	GC	GC	GF			
P20	23	28	56	TOB1T1/TIB11/TOB11	I/O	Provided
P21	24	29	57	TOB1B1/TIB12/TOB12	I/O	
P22	25	30	58	TOB1T2/TIB13/TOB13	I/O	
P23	26	31	59	TOB1B2/TIB10	I/O	
P24	27	32	60	TOB1T3/EVTB1	I/O	
P25	28	33	61	TOB1B3/TRGB1	I/O	
P26	29	34	62	TOB10/TOB1OFF/INTP10/ADTRG1/INTADT1	I/O	
P27	36	45	73	DMS[Notes 2, 3]	Input	

Notes 1. Software pull-up function

2. V850E/IG3 only

3. The P27 pin also functions as an on-chip debug pin. The on-chip debug function or port function (including the alternate functions) can be selected by using the level of the $\overline{\text{DRST}}$ pin, as shown in the table below.

Port 2 Functions	
Low-Level Input to $\overline{\text{DRST}}$ Pin	High-Level Input to $\overline{\text{DRST}}$ Pin
P27	DMS

Caution When P20 to P25 are used as TOB1T1 to TOB1T3 and TOB1B1 to TOB1B3, they go into a high-impedance state by inputting the following active signal.

- **Output of high impedance setting signal from high impedance output controller**
- **Output of clock stop detection signal from clock monitor**

Remark IF3: V850E/IF3

IG3: V850E/IG3

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(1) Registers

(a) Port 2 register (P2)

After reset: Undefined R/W Address: FFFFF404H

	7	6	5	4	3	2	1	0
P2	P27	P26	P25	P24	P23	P22	P21	P20

P2n	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Remark n = 0 to 7

(b) Port 2 mode register (PM2)

After reset: FFH R/W Address: FFFFF424H

	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

PM2n	Control of I/O mode (in port mode)
0	Output mode
1	Input mode

Remark n = 0 to 7

(c) Port 2 mode control register (PMC2)

After reset: 00H R/W Address: FFFFF444H

	7	6	5	4	3	2	1	0
PMC2	0	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20

PMC26	Specification of operating mode of P26 pin
0	I/O port
1	TOB10 output/TOB1OFF input/INTP10 input/ADTRG1 input/INTADT1 input

PMC25	Specification of operating mode of P25 pin
0	I/O port
1	TOB1B3 output/TRGB1 input

PMC24	Specification of operating mode of P24 pin
0	I/O port
1	TOB1T3 output/EVTB1 input

PMC23	Specification of operating mode of P23 pin
0	I/O port
1	TOB1B2 output/TIB10 input

PMC22	Specification of operating mode of P22 pin
0	I/O port
1	TOB1T2 output/TIB13 input/TOB13 output

PMC21	Specification of operating mode of P21 pin
0	I/O port
1	TOB1B1 output/TIB12 input/TOB12 output

PMC20	Specification of operating mode of P20 pin
0	I/O port
1	TOB1T1 output/TIB11 input/TOB11 output

(d) Port 2 function control register (PFC2)

After reset: 00H R/W Address: FFFFF464H

	7	6	5	4	3	2	1	0
PFC2	0	PFC26	PFC25	PFC24	PFC23	PFC22	PFC21	PFC20

Remark For the specifications of alternate functions, see **4.3.3 (1) (f) Settings of alternate functions of port 2**.

(e) Port 2 function control expansion register (PFCE2)

After reset: 00H R/W Address: FFFFF704H

	7	6	5	4	3	2	1	0
PFCE2	0	PFCE26	0	0	0	PFCE22	PFCE21	PFCE20

Remark For the specifications of alternate functions, see **4.3.3 (1) (f) Settings of alternate functions of port 2**.

(f) Settings of alternate functions of port 2

PFCE26	PFC26	Specification of Alternate Function of P26 Pin
0	0	TOB10 output
0	1	TOB1OFF input/INTP10 input (two functions are alternately used)
1	0	ADTRG1 input/INTADT1 input (two functions are alternately used)
1	1	Setting prohibited

PFC25	Specification of Alternate Function of P25 Pin
0	TOB1B3 output
1	TRGB1 input

PFC24	Specification of Alternate Function of P24 Pin
0	TOB1T3 output
1	EVTB1 input

PFC23	Specification of Alternate Function of P23 Pin
0	TOB1B2 output
1	TIB10 input

PFCE22	PFC22	Specification of Alternate Function of P22 Pin
0	0	TOB1T2 output
0	1	TIB13 input
1	0	TOB13 output
1	1	Setting prohibited

PFCE21	PFC21	Specification of Alternate Function of P21 Pin
0	0	TOB1B1 output
0	1	TIB12 input
1	0	TOB12 output
1	1	Setting prohibited

PFCE20	PFC20	Specification of Alternate Function of P20 Pin
0	0	TOB1T1 output
0	1	TIB11 input
1	0	TOB11 output
1	1	Setting prohibited

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

After reset: 00H R/W Address: FFFFFC44H

	7	6	5	4	3	2	1	0
PU2	PU27	PU26	PU25	PU24	PU23	PU22	PU21	PU20

PU2n	Control of on-chip pull-up resistor connection
0	Do not connect
1	Connect[Note]

Note An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode. Moreover, an on-chip pull-up resistor can be connected to the TOB1T1 to TOB1T3 and TOB1B1 to TOB1B3 pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOB1OFF pin, or software processing. An on-chip pull-up resistor cannot be connected when the pins are in output mode.

Remark n = 0 to 7

4.3.4 Port 3

Port 3 can be set to the input or output mode in 1-bit units.

Port 3 has an alternate function as the following pins.

Table 4-9. Alternate-Function Pins of Port 3

Pin Name	Pin No.			Alternate-Function Pin Name	I/O	Pull-Up[Note 1]
	IF3	IG3				
	GC	GC	GF			
P30	46	55	83	RXDA1/SCL	I/O	Provided
P31	47	56	84	TXDA1/SDA	I/O	
P32	48	57	85	SIB1/RXDA2/$\overline{CS1}$[Note 2]	I/O	
P33	49	58	86	SOB1/TXDA2	Output	
P34	50	59	87	$\overline{SCKB1}$/INTP11/$\overline{CS0}$[Note 2]	I/O	
P35	51	60	88	SIB2/RXDB	Input	
P36	52	61	89	SOB2/TXDB	Output	
P37	53	62	90	$\overline{SCKB2}$/INTP12/ASTB[Note 2]	I/O	

Notes 1. Software pull-up function

2. μPD70F3454GC-8EA-A only

Remark IF3: V850E/IF3

IG3: V850E/IG3

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(1) Registers

(a) Port 3 register (P3)

After reset: Undefined R/W Address: FFFFF406H

	7	6	5	4	3	2	1	0
P3	P37	P36	P35	P34	P33	P32	P31	P30

P3n	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Remark n = 0 to 7

(b) Port 3 mode register (PM3)

After reset: FFH R/W Address: FFFFF426H

	7	6	5	4	3	2	1	0
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

PM3n	Control of I/O mode (in port mode)
0	Output mode
1	Input mode

Remark n = 0 to 7

(c) Port 3 mode control register (PMC3)

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

PMC37	Specification of operating mode of P37 pin
0	I/O port
1	$\overline{\text{SCKB2}}$ I/O/NTP12 input/ASTB output[Note]

PMC36	Specification of operating mode of P36 pin
0	I/O port
1	SOB2 output/TXDB output

PMC35	Specification of operating mode of P35 pin
0	I/O port
1	SIB2 input/RXDB input

PMC34	Specification of operating mode of P34 pin
0	I/O port
1	$\overline{\text{SCKB1}}$ I/O/INTP11 input/$\overline{\text{CS0}}$ output[Note]

PMC33	Specification of operating mode of P33 pin
0	I/O port
1	SOB1 output/TXDA2 output

PMC32	Specification of operating mode of P32 pin
0	I/O port
1	SIB1 input/RXDA2 input/$\overline{\text{CS1}}$ output[Note]

PMC31	Specification of operating mode of P31 pin
0	I/O port
1	TXDA1 output/SDA I/O

PMC30	Specification of operating mode of P30 pin
0	I/O port
1	RXDA1 input/SCL I/O

Note μPD70F3454GC-8EA-A only

(d) Port 3 function control register (PFC3)

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

Remark For the specifications of alternate functions, see **4.3.4 (1) (f) Settings of alternate functions of port 3**.

(e) Port 3 function control expansion register (PFCE3)

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	PFCE37	PFCE36	PFCE35	PFCE34	0	PFCE32	PFCE31	PFCE30

Remark For the specifications of alternate functions, see **4.3.4 (1) (f) Settings of alternate functions of port 3**.

(f) Settings of alternate functions of port 3

PFCE37	PFC37	Specification of Alternate Function of P37 Pin
0	0	$\overline{\text{SCKB2}}$ input/output
0	1	INTP12 input
1	0	ASTB output[Note]
1	1	Setting prohibited

PFCE36	PFC36	Specification of Alternate Function of P36 Pin
0	0	SOB2 output
0	1	TXDB output
1	0	Setting prohibited
1	1	Setting prohibited

PFCE35	PFC35	Specification of Alternate Function of P35 Pin
0	0	SIB2 input
0	1	RXDB input
1	0	Setting prohibited
1	1	Setting prohibited

PFCE34	PFC34	Specification of Alternate Function of P34 Pin
0	0	$\overline{\text{SCKB1}}$ input/output
0	1	INTP11 input
1	0	$\overline{\text{CS0}}$ output[Note]
1	1	Setting prohibited

PFC33	Specification of Alternate Function of P33 Pin
0	SOB1 output
1	TXDA2 output

PFCE32	PFC32	Specification of Alternate Function of P32 Pin
0	0	SIB1 input
0	1	RXDA2 input
1	0	$\overline{\text{CS1}}$ output[Note]
1	1	Setting prohibited

Note μPD70F3454GC-8EA-A only

PFCE31	PFC31	Specification of Alternate Function of P31 Pin
0	0	TXDA1 output
0	1	SDA input/output
1	0	Setting prohibited
1	1	Setting prohibited

PFCE30	PFC30	Specification of Alternate Function of P30 Pin
0	0	RXDA1 input
0	1	SCL input/output
1	0	Setting prohibited
1	1	Setting prohibited

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

After reset: 00H R/W Address: FFFFFC46H

	7	6	5	4	3	2	1	0
PU3	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30

PU3n	Control of on-chip pull-up resistor connection
0	Do not connect
1	Connect[Note]

Note An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode (including the $\overline{\text{SCKB1}}$ and $\overline{\text{SCKB2}}$ pins in the slave mode). An on-chip pull-up resistor cannot be connected when the pins are in output mode.

Remark n = 0 to 7

(h) Port 3 function register (PF3)

After reset: 00H R/W Address: FFFFFC66H

	7	6	5	4	3	2	1	0
PF3	0	0	0	0	0	0	PF31	PF30

PF3n	Control of normal output/N-ch open-drain output (n = 0, 1)
0	Normal output (CMOS output)
1	N-ch open-drain output[Note]

Note When using I^2C, set as N-ch open-drain output.

4.3.5 Port 4

Port 4 can be set to the input or output mode in 1-bit units.

Port 4 has an alternate function as the following pins.

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

Pin Name	Pin No.			Alternate-Function Pin Name	I/O	Pull-Up[Note 1]
	IF3	IG3				
	GC	GC	GF			
P40	38	47	75	SIB0/RXDA0	Input	Provided
P41	39	48	76	SOB0/TXDA0/DCK[Notes 2, 3]	I/O	
P42	40	49	77	$\overline{\text{SCKB0}}$/INTP13/DDI[Notes 2, 3]	I/O	
P43	41	50	78	TECR1/TIT10/TOT10/INTP14	I/O	
P44	42	51	79	TENC10/EVTT1/INTP15/$\overline{\text{WAIT}}$[Note 4]	Input	
P45	43	52	80	TENC11/TIT11/TOT11/INTP16/$\overline{\text{WR1}}$[Note 4]	I/O	
P46	44	53	81	TOA40/TIA40/INTP17/$\overline{\text{WR0}}$[Note 4]	I/O	
P47	45	54	82	TOA41/TIA41/INTP18/$\overline{\text{RD}}$[Note 4]	I/O	

Notes 1. Software pull-up function

2. V850E/IG3 only

3. The P41 and P42 pins also function as on-chip debug pins. The on-chip debug function or port function (including the alternate functions) can be selected by using the level of the $\overline{\text{DRST}}$ pin, as shown in the table below.

Port 4 Functions	
Low-Level Input to $\overline{\text{DRST}}$ Pin	High-Level Input to $\overline{\text{DRST}}$ Pin
P41/SOB0/TXDA0	DCK
P42/$\overline{\text{SCKB0}}$/INTP13	DDI

4. μPD70F3454GC-8EA-A only

Remark IF3: V850E/IF3
IG3: V850E/IG3
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)
GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)
GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(1) Registers

(a) Port 4 register (P4)

After reset: Undefined R/W Address: FFFFF408H

	7	6	5	4	3	2	1	0
P4	P47	P46	P45	P44	P43	P42	P41	P40

P4n	Control of output data (in output mode)
0	Output 0.
1	Output 1.

Remark n = 0 to 7

(b) Port 4 mode register (PM4)

After reset: FFH R/W Address: FFFFF428H

	7	6	5	4	3	2	1	0
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40

PM4n	Control of I/O mode (in port mode)
0	Output mode
1	Input mode

Remark n = 0 to 7

(c) Port 4 mode control register (PMC4)

After reset: 00H R/W Address: FFFFF448H

	7	6	5	4	3	2	1	0
PMC4	PMC47	PMC46	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40

PMC47	Specification of operating mode of P47 pin
0	I/O port
1	TOA41 output/TIA41 input/INTP18 input/$\overline{RD}$ output[Note 1]

PMC46	Specification of operating mode of P46 pin
0	I/O port
1	TOA40 output/TIA40 input/INTP17 input/$\overline{WR0}$ output[Note 1]

PMC45	Specification of operating mode of P45 pin
0	I/O port
1	TENC11 input/TIT11 input/TOT11 output/INTP16 input/$\overline{WR1}$ output[Note 1]

PMC44	Specification of operating mode of P44 pin
0	I/O port
1	TENC10 input/EVTT1 input/INTP15 input/$\overline{WAIT}$ input[Note 1]

PMC43	Specification of operating mode of P43 pin
0	I/O port
1	TECR1 input/TIT10 input/TOT10 output/INTP14 input

PMC42	Specification of operating mode of P42 pin
0	I/O port
1	$\overline{SCKB0}$ input/output/INTP13 input/DDI input[Note 2]

PMC41	Specification of operating mode of P41 pin
0	I/O port
1	SOB0 output/TXDA0 output/DCK input[Note 2]

PMC40	Specification of operating mode of P40 pin
0	I/O port
1	SIB0 input/RXDA0 input

Notes 1. μPD70F3454GC-8EA-A only

2. V850E/IG3 only

(d) Port 4 function control register (PFC4)

After reset: 00H R/W Address: FFFFF468H

	7	6	5	4	3	2	1	0
PFC4	PFC47	PFC46	PFC45	PFC44	PFC43	PFC42	PFC41	PFC40

Remark For the specifications of alternate functions, see **4.3.5 (1) (f) Settings of alternate functions of port 4**.

(e) Port 4 function control expansion register (PFCE4)

After reset: 00H R/W Address: FFFFF708H

	7	6	5	4	3	2	1	0
PFCE4	PFCE47	PFCE46	PFCE45	PFCE44	PFCE43	PFCE42	PFCE41	0

Remark For the specifications of alternate functions, see **4.3.5 (1) (f) Settings of alternate functions of port 4**.

(f) Settings of alternate functions of port 4

PFCE47	PFC47	Specification of Alternate Function of P47 Pin
0	0	TOA41 output
0	1	TIA41 input
1	0	INTP18 input
1	1	$\overline{RD}$ output[Note]

PFCE46	PFC46	Specification of Alternate Function of P46 Pin
0	0	TOA40 output
0	1	TIA40 input
1	0	INTP17 input
1	1	$\overline{WR0}$ output[Note]

PFCE45	PFC45	Specification of Alternate Function of P45 Pin
0	0	TENC11 input/TIT11 input (two functions are alternately used)
0	1	TOT11 output
1	0	INTP16 input
1	1	$\overline{WR1}$ output[Note]

PFCE44	PFC44	Specification of Alternate Function of P44 Pin
0	0	TENC10 input
0	1	EVTT1 input
1	0	INTP15 input
1	1	$\overline{WAIT}$ input[Note]

PFCE43	PFC43	Specification of Alternate Function of P43 Pin
0	0	TECR1 input/TIT10 input (two functions are alternately used)
0	1	TOT10 output
1	0	INTP14 input
1	1	Setting prohibited

PFCE42	PFC42	Specification of Alternate Function of P42 Pin
0	0	$\overline{SCKB0}$ I/O
0	1	INTP13 input
1	0	Setting prohibited
1	1	Setting prohibited

Note μPD70F3454GC-8EA-A only

PFCE41	PFC41	Specification of Alternate Function of P41 Pin
0	0	SOB0 output
0	1	TXDA0 output
1	0	Setting prohibited
1	1	Setting prohibited

PFC40	Specification of Alternate Function of P40 Pin
0	SIB0 input
1	RXDA0 input

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

After reset: 00H R/W Address: FFFFFC48H

	7	6	5	4	3	2	1	0
PU4	PU47	PU46	PU45	PU44	PU43	PU42	PU41	PU40

PU4n	Control of on-chip pull-up resistor connection
0	Do not connect
1	Connect[Note]

Note An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode or when the pins function as input pins in the alternate-function mode (including the $\overline{\text{SCKB0}}$ pin in the slave mode). An on-chip pull-up resistor cannot be connected when the pins are in output mode.

Remark n = 0 to 7

4.3.6 Port 7

Port 7 is an input port with all its pins fixed to the input mode.

The number of input port pins differs depending on the product.

Generic Name	Number of I/O Ports
V850E/IF3	4-bit input-only port
V850E/IG3	8-bit input-only port

Port 7 has an alternate function as the following pins.

Table 4-11. Alternate-Function Pins of Port 7

Pin Name	Pin No.			Alternate-Function Pin Name	I/O	Pull-Up[Note 1]
	IF3	IG3				
	GC	GC	GF			
P70	20	25	53	ANI20	Input	None
P71	19	24	52	ANI21	Input	
P72	18	23	51	ANI22	Input	
P73	17	22	50	ANI23	Input	
P74[Note 2]	–	21	49	ANI24[Note 2]	Input	
P75[Note 2]	–	20	48	ANI25[Note 2]	Input	
P76[Note 2]	–	19	47	ANI26[Note 2]	Input	
P77[Note 2]	–	18	46	ANI27[Note 2]	Input	

Notes 1. Software pull-up function

2. V850E/IG3 only

Remark IF3: V850E/IF3

IG3: V850E/IG3

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(1) Registers

(a) Port 7 register (P7)

After reset: Undefined R Address: FFFFFBB0H

	7	6	5	4	3	2	1	0
P7	P77Note	P76Note	P75Note	P74Note	P73	P72	P71	P70

P7n	Control of input data
0	Input low level.
1	Input high level.

Note Valid only in the V850E/IG3.
With the V850E/IF3, the read value of this register is undefined.

Caution When using a port input pin and analog input pin (ANI2n) together, be sure to set (1) the bit (PMC7n) of the PMC7 register to be used as the ANI2n pin.

Remark V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

(b) Port 7 mode control register (PMC7)

After reset: 00H R/W Address: FFFFFBB8H

	7	6	5	4	3	2	1	0
PMC7	PMC77Note	PMC76Note	PMC75Note	PMC74Note	PMC73	PMC72	PMC71	PMC70

PMC7n	Specification of operating mode of P7n pin
0	Input port (reading P7n enabled. Input buffer is on when this bit is read)
1	ANI2n input (reading P7n disabled. Input buffer is off when this bit is read)

Note Valid only in the V850E/IG3.
With the V850E/IF3, be sure to set these bits to 0.

Cautions 1. Do not change to the port mode using A/D converter 2 during A/D conversion.

2. The PMC7 register enables or disables reading of the P7 register. When the PMC7n bit = 1, the input buffer does not turn on even when the P7 register is read. In this case, the read value of the P7n bit is fixed to the low level (V850E/IF3: n = 0 to 3, V850E/IG3: n = 0 to 7). This is to prevent through-current that may flow when the ANI2n input (intermediate level) is read.

Remark V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

4.3.7 Port DL

Port DL can be set to the input or output mode in 1-bit units.

Port DL has an alternate function as the following pins.

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

Pin Name	Pin No,			Alternate-Function Pin Name	I/O	Pull-Up[Note 1]
	IF3	IG3				
	GC	GC	GF			
PDL0	65	81	9	AD0[Note 3]	I/O	Provided
PDL1	64	80	8	AD1[Note 3]	I/O	
PDL2	63	79	7	AD2[Note 3]	I/O	
PDL3	62	78	6	AD3[Note 3]	I/O	
PDL4	61	77	5	AD4[Note 3]	I/O	
PDL5	60	76	4	AD5[Note 3]/FLMD1[Note 4]	I/O	
PDL6	59	75	3	AD6[Note 3]	I/O	
PDL7	58	74	2	AD7[Note 3]	I/O	
PDL8	57	73	1	AD8[Note 3]	I/O	
PDL9	56	72	100	AD9[Note 3]	I/O	
PDL10[Note 2]	–	71	99	AD10[Note 3]	I/O	
PDL11[Note 2]	–	70	98	AD11[Note 3]	I/O	
PDL12[Note 2]	–	69	97	AD12[Note 3]	I/O	
PDL13[Note 2]	–	68	96	AD13[Note 3]	I/O	
PDL14[Note 2]	–	67	95	AD14[Note 3]	I/O	
PDL15[Note 2]	–	66	94	AD15[Note 3]	I/O	

Notes 1. Software pull-up function

3. V850E/IG3 only

3. μPD70F3454GC-8EA-A only

4. This pin is used in the flash programming mode and does not have to be manipulated by a port control register. For details, see **CHAPTER 27 FLASH MEMORY**.

Remark IF3: V850E/IF3

IG3: V850E/IG3

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(1) Registers

(a) Port DL register (PDL)

After reset: Undefined R/W Address: PDL FFFFF004H
PDLL FFFFF004H, PDLH FFFFF005H

	15	14	13	12	11	10	9	8
PDL (PDLH[Note 1])	PDL15[Note 2]	PDL14[Note 2]	PDL13[Note 2]	PDL12[Note 2]	PDL11[Note 2]	PDL10[Note 2]	PDL9	PDL8

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

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

Notes 1. To read/write bits 8 to 15 of the PDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PDLH register.

2. Valid only in the V850E/IG3.
With the V850E/IF3, the read value of this register is undefined.

Remarks 1. The PDL register can be read or written in 16-bit units.
When the higher 8 bits of the PDL register are used as the PDLH register, and the lower 8 bits, as the PDLL register, these registers can be read or written in 8-bit or 1-bit units.

2. V850E/IF3: n = 0 to 9
V850E/IG3: n = 0 to 15

(b) Port DL mode register (PMDL)

After reset: FFFFH R/W Address: PMDL FFFFF024H
PMDLL FFFFF024H, PMDLH FFFFF025H

	15	14	13	12	11	10	9	8
PMDL (PMDLH[Note 1])	PMDL15[Note 2]	PMDL14[Note 2]	PMDL13[Note 2]	PMDL12[Note 2]	PDAL11[Note 2]	PDAL10[Note 2]	PMDL9	PMDL8

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

PMDLn	Control of I/O mode (in port mode)
0	Output mode
1	Input mode

Notes 1. To read/write bits 8 to 15 of the PMDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PMDLH register.

2. Valid only in the V850E/IG3.
With the V850E/IF3, be sure to set these bits to 1.

Remarks 1. The PMDL register can be read or written in 16-bit units.
When the higher 8 bits of the PMDL register are used as the PMDLH register, and the lower 8 bits, as the PMDLL register, these registers can be read or written in 8-bit or 1-bit units.

2. V850E/IF3: n = 0 to 9
V850E/IG3: n = 0 to 15

(c) Port DL mode control register (PMCDL)

After reset: 0000H R/W Address: PMCDL FFFFF044H
PMCDLL FFFFF044H, PMCDLH FFFFF045H

	15	14	13	12	11	10	9	8
PMCDL (PMCDLH[Note 1])	PMCDL15[Note 2]	PMCDL14[Note 2]	PMCDL13[Note 2]	PMCDL12[Note 2]	PMCDL11[Note 2]	PMCDL10[Note 2]	PMCDL9	PMCDL8

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

PMCDLn	Specification of operating mode of PMCDLn pin
0	I/O port
1	ADn input/output

Notes 1. To read/write bits 8 to 15 of the PMCDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PMCDLH register.

2. Valid only in the V850E/IG3.
With the V850E/IF3, be sure to set these bits to 0.

Remarks 1. The PMCDL register can be read or written in 16-bit units.
When the higher 8 bits of the PMCDL register are used as the PMCDLH register, and the lower 8 bits, as the PMCDLL register, these registers can be read or written in 8-bit or 1-bit units.

2. V850E/IF3: n = 0 to 9
V850E/IG3: n = 0 to 15

(d) Pull-up resistor option register DL (PUDL)

After reset: 0000H R/W Address: PUDL FFFFFF44H
PUDLL FFFFFF44H, PUDLH FFFFFF45H

	15	14	13	12	11	10	9	8
PUDL (PUDLH[Note 1])	PUDL15[Note 2]	PUDL14[Note 2]	PUDL13[Note 2]	PUDL12[Note 2]	PUDL11[Note 2]	PUDL10[Note 2]	PUDL9	PUDL8

	7	6	5	4	3	2	1	0
(PUDLL)	PUDL7	PUDL6	PUDL5	PUDL4	PUDL3	PUDL2	PUDL1	PUDL0

PUDLn	Control of on-chip pull-up resistor connection
0	Do not connect
1	Connect[Note 3]

Notes 1. To read/write bits 8 to 15 of the PUDL register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PUDLH register.

2. Valid only in the V850E/IG3.
With the V850E/IF3, be sure to set these bits to 0.

3. An on-chip pull-up resistor can be connected only when the pins are in input mode in the port mode.
An on-chip pull-up resistor cannot be connected when the pins are in output mode.

Remarks 1. The PUDL register can be read or written in 16-bit units.
When the higher 8 bits of the PUDL register are used as the PUDLH register, and the lower 8 bits, as the PUDLL register, these registers can be read or written in 8-bit or 1-bit units.

2. V850E/IF3: n = 0 to 9
V850E/IG3: n = 0 to 15

4.4 Output Data and Port Read Value for Each Setting

Table 4-14 shows the values used to select the alternate function of the respective pins, output data and port read values for each setting. In addition to the settings shown in Table 4-14, the setting of each peripheral function control register is required.

Table 4-13. Output Data and Port Read Value for Each Setting (1/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	
P00, P01, P02[Note], P03[Note]	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOA20, TOA21, TOA30[Note], TOA31[Note]	1	0	0	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	TIA20, TIA21, TIA30[Note], TIA31[Note]	1	0	1	0	–	Port latch	Alternate inpu
					1		Pin level	
	TOA2OFF, INTP00, INTP01, TOA3OFF[Note], INTP02[Note], INTP03[Note]	1	1	0	0	–	Port latch	Alternate inpu input (necess
					1		Pin level	
P04[Note], P06[Note]	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TECR0[Note], TIT00[Note], TENC01[Note], TIT01[Note]	1	0	0	0	–	Port latch	Alternate inpu
					1		Pin level	
	TOT00[Note], TOT01[Note]	1	0	1	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	INTP04[Note], INTP06[Note]	1	1	0	0	–	Port latch	Alternate inpu (necessary to
					1		Pin level	

Note V850E/IG3 only

Remark ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (2/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	
P05[Note 1]	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TENC00[Note 1]	1	0	0	0	–	Port latch	Alternate inpu
					1		Pin level	
	EVTT0[Note 1]	1	0	1	0	–	Port latch	Alternate inpu
					1		Pin level	
	INTP05[Note 1]	1	1	0	0	–	Port latch	Alternate inpu (necessary to
					1		Pin level	
P07[Note 1]	Output port	0	None	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	INTP07[Note 1]	1	None	0	0	–	Port latch	Alternate inpu (necessary to
					1		Pin level	
	CLKOUT[Note 2]	1	None	1	0	Alternate output (bus output)	Port latch	
					1		Pin level	

Notes 1. V850E/IG3 only

2. μPD70F3454GC-8EA-A only

Remark ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (3/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	
P10 to P12	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOB0T1, TOB0B1, TOB0T2	1	0	0	0	Alternate output 1 (timer output)	Port latch	
					1		Pin level	
	TIB01 to TIB03	1	0	1	0	–	Port latch	Alternate inpu
					1		Pin level	
	TOB01 to TOB03	1	1	0	0	Alternate output 2 (timer output)	Port latch	
					1		Pin level	
	A0[Note], A1[Note], A2[Note]	1	1	1	0	Alternate output 3 (bus output)	Port latch	
					1		Pin level	
P13 to P15	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOB0B2, TOB0T3, TOB0B3	1	0	0	0	Alternate output 1 (timer output)	Port latch	
					1		Pin level	
	TIB00, EVTB0, TRGB0	1	0	1	0	–	Port latch	Alternate inpu
					1		Pin level	
	A3[Note], A4[Note], A5[Note]	1	1	0	0	Alternate output 2 (bus output)	Port latch	
					1		Pin level	

Note μPD70F3454GC-8EA-A only

Remark ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (4/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	
P16	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOB0OFF/INTP08	1	0	0	0	Alternate output 1 (timer output)	Port latch	
					1		Pin level	
	ADTRG0/INTADT0	1	0	1	0	–	Port latch	Alternate inpu
					1		Pin level	input (necess
	A6[Note]	1	1	0	0	Alternate output 2 (bus output)	Port latch	
					1		Pin level	
P17	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOB00	1	0	0	0	Alternate output 1 (timer output)	Port latch	
					1		Pin level	
	INTP09	1	0	1	0	–	Port latch	Alternate inpu
					1		Pin level	(necessary to
	A7[Note]	1	1	0	0	Alternate output 2 (bus output)	Port latch	
					1		Pin level	
P20 to P22	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOB1T1, TOB1B1, TOB1T2	1	0	0	0	Alternate output 1 (timer output)	Port latch	
					1		Pin level	
	TIB11 to TIB13	1	0	1	0	–	Port latch	Alternate inpu
					1		Pin level	
	TOB11 to TOB13	1	1	0	0	Alternate output 2 (timer output)	Port latch	
					1		Pin level	

Note μPD70F3454GC-8EA-A only

Remark ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (5/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	
P23 to P25	Output port	0	None	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOB1B2, TOB1T3, TOB1B3	1	None	0	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	TIB10, EVTB1, TRGB1	1	None	1	0	–	Port latch	Alternate inpu
					1		Pin level	
P26	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOB10	1	0	0	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	TOB1OFF/INTP10	1	0	1	0	–	Port latch	Alternate inpu input (necess
					1		Pin level	
	ADTRG1/INTADT1	1	1	0	0	–	Port latch	Alternate inpu input (necess
					1		Pin level	
P27[Note]	Output port	None	None	None	0	Port latch	Port latch	
	Input port				1	–	Pin level	

Note The P27 pin is also used for on-chip debugging (V850E/IG3 only). Switching between on-chip debug function and port function (i can be set by using the $\overline{\text{DRST}}$ pin level. The following shows the setting method.

Port 2 Functions	
Low-Level Input to $\overline{\text{DRST}}$ Pin	High-Level Input to $\overline{\text{DRST}}$ Pin
P27	DMS

Remark ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (6/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	
P30	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	RXDA1	1	0	0	0	–	Port latch	Alternate inpu
					1		Pin level	
	SCL	1	0	1	0	Alternate I/O (serial I/O)	Port latch	Output in mas
					1		Pin level	Input in slave
P31	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TXDA1	1	0	0	0	Alternate output (serial output)	Port latch	
					1		Pin level	
	SDA	1	0	1	0	Alternate I/O (serial I/O)	Port latch	Output in mas
					1		Pin level	Input in slave
P32	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	SIB1	1	0	0	0	–	Port latch	Alternate inpu
					1		Pin level	
	RXDA2	1	0	1	0	–	Port latch	Alternate inpu
					1		Pin level	
	$\overline{CS1}$ Note	1	1	0	0	Alternate output (bus output)	Port latch	
					1		Pin level	

Note μPD70F3454GC-8EA-A only

Remark ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (7/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	
P33	Output port	0	None	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	SOB1	1	None	0	0	Alternate output 1 (serial output)	Port latch	
					1		Pin level	
	TXDA2	1	None	1	0	Alternate output 2 (serial output)	Port latch	
					1		Pin level	
P34	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	$\overline{\text{SCKB1}}$	1	0	0	0	Alternate I/O (serial I/O)	Port latch	Output in mas
					1		Pin level	Input in slave
	INTP11	1	0	1	0	–	Port latch	Alternate inpu
					1		Pin level	(necessary to
	$\overline{\text{CS0}}$[Note]	1	1	0	0	Alternate output (bus output)	Port latch	
					1		Pin level	
P35	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	SIB2	1	0	0	0	–	Port latch	Alternate inpu
					1		Pin level	
	RXDB	1	0	1	0	–	Port latch	Alternate inpu
					1		Pin level	

Note μPD70F3454GC-8EA-A only

Remark ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (8/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	
P36	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	SOB2	1	0	0	0	Alternate output 1 (serial output)	Port latch	
					1		Pin level	
	TXDB	1	0	1	0	Alternate output 2 (serial output)	Port latch	
					1		Pin level	
P37	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	$\overline{\text{SCKB2}}$	1	0	0	0	Alternate I/O (serial I/O)	Port latch	Output in mas
					1		Pin level	Input in slave
	INTP12	1	0	1	0	–	Port latch	Alternate inpu
					1		Pin level	(necessary to
	ASTB[Note]	1	1	0	0	Alternate output (bus output)	Port latch	
					1		Pin level	
P40	Output port	0	None	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	SIB0	1	None	0	0	–	Port latch	Alternate inpu
					1		Pin level	
	RXDA0	1	None	1	0	–	Port latch	Alternate inpu
					1		Pin level	

Note μPD70F3454GC-8EA-A only

Remark ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (9/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	
P41[Note]	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	SOB0	1	0	0	0	Alternate output 1 (serial output)	Port latch	
					1		Pin level	
	TXDA0	1	0	1	0	Alternate output 2 (serial output)	Port latch	
					1		Pin level	
P42[Note]	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	$\overline{\text{SCKB0}}$	1	0	0	0	Alternate I/O (serial I/O)	Port latch	Output in mas
					1		Pin level	Input in slave
	INTP13	1	0	1	0	–	Port latch	Alternate inpu
					1		Pin level	(necessary to

Note The P41 and P42 pins are also used for on-chip debugging (V850E/IG3 only). Switching between on-chip debug function a alternate function) can be set by using the $\overline{\text{DRST}}$ pin level. The following shows the setting method.

Port 4 Functions	
Low-Level Input to $\overline{\text{DRST}}$ Pin	High-Level Input to $\overline{\text{DRST}}$ Pin
P41/SOB0/TXDA0	DCK
P42/$\overline{\text{SCKB0}}$/INTP13	DDI

Remark ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (10/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	
P43	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TECR1/TIT10	1	0	0	0	–	Port latch	Alternate inpu
					1		Pin level	
	TOT10	1	0	1	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	INTP14	1	1	0	0	–	Port latch	Alternate inpu (necessary to
					1		Pin level	
P44	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TENC10	1	0	0	0	–	Port latch	Alternate inpu
					1		Pin level	
	EVTT1	1	0	1	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	INTP15	1	1	0	0	–	Port latch	Alternate inpu (necessary to
					1		Pin level	
	$\overline{\text{WAIT}}$Note	1	1	1	0	–	Port latch	Alternate inpu
					1		Pin level	

Note μPD70F3454GC-8EA-A only

Remark ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (11/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	
P45	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TENC11/TIT11	1	0	0	0	–	Port latch	Alternate inpu
					1		Pin level	
	TOT11	1	0	1	0	Alternate output 1 (timer output)	Port latch	
					1		Pin level	
	INTP16	1	1	0	0	–	Port latch	Alternate inpu (necessary to
					1		Pin level	
	$\overline{WR1}$[Note]	1	1	1	0	Alternate output 2 (bus output)	Port latch	
					1		Pin level	
P46, P47	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOA40, TOA41	1	0	0	0	Alternate output 1 (timer output)	Port latch	
					1		Pin level	
	TIA40, TIA41	1	0	1	0	–	Port latch	Alternate inpu
					1		Pin level	
	INTP17, INTP18	1	1	0	0	–	Port latch	Alternate inpu (necessary to
					1		Pin level	
	$\overline{WR0}$[Note], $\overline{RD}$[Note]	1	1	1	0	Alternate output 2 (bus output)	Port latch	
					1		Pin level	

Note μPD70F3454GC-8EA-A only

Remark ×: don't care

Table 4-13. Output Data and Port Read Value for Each Setting (12/12)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	
P70 to P73, P74 to P77[Note 1]	Input port	0	None	None	None	–	Pin level	Input-only por
	ANI20 to ANI23, ANI24 to ANI27[Note 1]	1				–	Low level	
PDL0 to PDL9, PDL10 to PDL15[Note 1]	Output port	0	None	None	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	AD0 to AD15[Note 2]	1	None	None	0	Alternate I/O (bus I/O)	Port latch	
					1		Pin level	

Notes 1. V850E/IG3 only

2. μPD70F3454GC-8EA-A only

3. The PDL5 pin is also used in flash programming mode. This pin does not have to be manipulated by a port control register. **FLASH MEMORY**.

Remark ×: don't care

4.5 Port Register Settings When Alternate Function Is Used

The following shows the port register settings when each port is used for an alternate function. When using a port pin as an alternate-function pin, refer to the description of each pin.

Table 4-14. Using Port Pin as Alternate-Function Pin (1/8)

Pin Name	Alternate Pin		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register
	Name	I/O					
P00	TOA20	Output	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	PFCE00 = 0	PFC00 = 0
	TIA20	Input	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	PFCE00 = 0	PFC00 = 1
	TOA2OFF	Input	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	PFCE00 = 1	PFC00 = 0
	INTP00	Input	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	PFCE00 = 1	PFC00 = 0
P01	TOA21	Output	P01 = Setting not required	PM01 = Setting not required	PMC01 = 1	PFCE01 = 0	PFC01 = 0
	TIA21	Input	P01 = Setting not required	PM01 = Setting not required	PMC01 = 1	PFCE01 = 0	PFC01 = 1
	INTP01	Input	P01 = Setting not required	PM01 = Setting not required	PMC01 = 1	PFCE01 = 1	PFC01 = 0
P02[Note]	TOA30[Note]	Output	P02 = Setting not required	PM02 = Setting not required	PMC02 = 1	PFCE02 = 0	PFC02 = 0
	TIA30[Note]	Input	P02 = Setting not required	PM02 = Setting not required	PMC02 = 1	PFCE02 = 0	PFC02 = 1
	TOA3OFF[Note]	Input	P02 = Setting not required	PM02 = Setting not required	PMC02 = 1	PFCE02 = 1	PFC02 = 0
	INTP02[Note]	Input	P02 = Setting not required	PM02 = Setting not required	PMC02 = 1	PFCE02 = 1	PFC02 = 0
P03[Note]	TOA31[Note]	Output	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	PFCE03 = 0	PFC03 = 0
	TIA31[Note]	Input	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	PFCE03 = 0	PFC03 = 1
	INTP03[Note]	Input	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	PFCE03 = 1	PFC03 = 0
P04[Note]	TECR0[Note]	Input	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	PFCE04 = 0	PFC04 = 0
	TIT00[Note]	Input	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	PFCE04 = 0	PFC04 = 0
	TOT00[Note]	Output	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	PFCE04 = 0	PFC04 = 1
	INTP04[Note]	Input	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	PFCE04 = 1	PFC04 = 0
P05[Note]	TENC00[Note]	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	PFCE05 = 0	PFC05 = 0
	EVTT0[Note]	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	PFCE05 = 0	PFC05 = 1
	INTP05[Note]	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	PFCE05 = 1	PFC05 = 0

Note V850E/IG3 only

Table 4-14. Using Port Pin as Alternate-Function Pin (2/8)

Pin Name	Alternate Pin		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register
	Name	I/O					
P06[Note 1]	TENC01[Note 1]	Input	P06 = Setting not required	PM06 = Setting not required	PMC06 = 1	PFCE06 = 0	PFC06 = 0
	TIT01[Note 1]	Input	P06 = Setting not required	PM06 = Setting not required	PMC06 = 1	PFCE06 = 0	PFC06 = 0
	TOT01[Note 1]	Output	P06 = Setting not required	PM06 = Setting not required	PMC06 = 1	PFCE06 = 0	PFC06 = 1
	INTP06[Note 1]	Input	P06 = Setting not required	PM06 = Setting not required	PMC06 = 1	PFCE06 = 1	PFC06 = 0
P07[Note 1]	INTP07[Note 1]	Input	P07 = Setting not required	PM07 = Setting not required	PMC07 = 1	–	PFC07 = 0
	CLKOUT[Note 2]	Output	P07 = Setting not required	PM07 = Setting not required	PMC07 = 1	–	PFC07 = 1
P10	TOB0T1	Output	P10 = Setting not required	PM10 = Setting not required	PMC10 = 1	PFCE10 = 0	PFC10 = 0
	TIB01	Input	P10 = Setting not required	PM10 = Setting not required	PMC10 = 1	PFCE10 = 0	PFC10 = 1
	TOB01	Output	P10 = Setting not required	PM10 = Setting not required	PMC10 = 1	PFCE10 = 1	PFC10 = 0
	A0[Note 2]	Output	P10 = Setting not required	PM10 = Setting not required	PMC10 = 1	PFCE10 = 1	PFC10 = 1
P11	TOB0B1	Output	P11 = Setting not required	PM11 = Setting not required	PMC11 = 1	PFCE11 = 0	PFC11 = 0
	TIB02	Input	P11 = Setting not required	PM11 = Setting not required	PMC11 = 1	PFCE11 = 0	PFC11 = 1
	TOB02	Output	P11 = Setting not required	PM11 = Setting not required	PMC11 = 1	PFCE11 = 1	PFC11 = 0
	A1[Note 2]	Output	P11 = Setting not required	PM11 = Setting not required	PMC11 = 1	PFCE11 = 1	PFC11 = 1
P12	TOB0T2	Output	P12 = Setting not required	PM12 = Setting not required	PMC12 = 1	PFCE12 = 0	PFC12 = 0
	TIB03	Input	P12 = Setting not required	PM12 = Setting not required	PMC12 = 1	PFCE12 = 0	PFC12 = 1
	TOB03	Output	P12 = Setting not required	PM12 = Setting not required	PMC12 = 1	PFCE12 = 1	PFC12 = 0
	A2[Note 2]	Output	P12 = Setting not required	PM12 = Setting not required	PMC12 = 1	PFCE12 = 1	PFC12 = 1
P13	TOB0B2	Output	P13 = Setting not required	PM13 = Setting not required	PMC13 = 1	PFCE13 = 0	PFC13 = 0
	TIB00	Input	P13 = Setting not required	PM13 = Setting not required	PMC13 = 1	PFCE13 = 0	PFC13 = 1
	A3[Note 2]	Output	P13 = Setting not required	PM13 = Setting not required	PMC13 = 1	PFCE13 = 1	PFC13 = 0

Notes 1. V850E/IG3 only

2. μPD70F3454GC-8EA-A only

Table 4-14. Using Port Pin as Alternate-Function Pin (3/8)

Pin Name	Alternate Pin		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register
	Name	I/O					
P14	TOB0T3	Output	P14 = Setting not required	PM14 = Setting not required	PMC14 = 1	PFCE14 = 0	PFC14 = 0
	EVTB0	Input	P14 = Setting not required	PM14 = Setting not required	PMC14 = 1	PFCE14 = 0	PFC14 = 1
	A4[Note]	Output	P14 = Setting not required	PM14 = Setting not required	PMC14 = 1	PFCE14 = 1	PFC14 = 0
P15	TOB0B3	Output	P15 = Setting not required	PM15 = Setting not required	PMC15 = 1	PFCE15 = 0	PFC15 = 0
	TRGB0	Input	P15 = Setting not required	PM15 = Setting not required	PMC15 = 1	PFCE15 = 0	PFC15 = 1
	A5[Note]	Output	P15 = Setting not required	PM15 = Setting not required	PMC15 = 1	PFCE15 = 1	PFC15 = 0
P16	TOB0OFF	Input	P16 = Setting not required	PM16 = Setting not required	PMC16 = 1	PFCE16 = 0	PFC16 = 0
	INTP08	Input	P16 = Setting not required	PM16 = Setting not required	PMC16 = 1	PFCE16 = 0	PFC16 = 0
	ADTRG0	Input	P16 = Setting not required	PM16 = Setting not required	PMC16 = 1	PFCE16 = 0	PFC16 = 1
	INTADT0	Input	P16 = Setting not required	PM16 = Setting not required	PMC16 = 1	PFCE16 = 0	PFC16 = 1
	A6[Note]	Output	P16 = Setting not required	PM16 = Setting not required	PMC16 = 1	PFCE16 = 1	PFC16 = 0
P17	TOB00	Output	P17 = Setting not required	PM17 = Setting not required	PMC17 = 1	PFCE17 = 0	PFC17 = 0
	INTP09	Input	P17 = Setting not required	PM17 = Setting not required	PMC17 = 1	PFCE17 = 0	PFC17 = 1
	A7[Note]	Output	P17 = Setting not required	PM17 = Setting not required	PMC17 = 1	PFCE17 = 1	PFC17 = 0
P20	TOB1T1	Output	P20 = Setting not required	PM20 = Setting not required	PMC20 = 1	PFCE20 = 0	PFC20 = 0
	TIB11	Input	P20 = Setting not required	PM20 = Setting not required	PMC20 = 1	PFCE20 = 0	PFC20 = 1
	TOB11	Output	P20 = Setting not required	PM20 = Setting not required	PMC20 = 1	PFCE20 = 1	PFC20 = 0
P21	TOB1B1	Output	P21 = Setting not required	PM21 = Setting not required	PMC21 = 1	PFCE21 = 0	PFC21 = 0
	TIB12	Input	P21 = Setting not required	PM21 = Setting not required	PMC21 = 1	PFCE21 = 0	PFC21 = 1
	TOB12	Output	P21 = Setting not required	PM21 = Setting not required	PMC21 = 1	PFCE21 = 1	PFC21 = 0
P22	TOB1T2	Output	P22 = Setting not required	PM22 = Setting not required	PMC22 = 1	PFCE22 = 0	PFC22 = 0
	TIB13	Input	P22 = Setting not required	PM22 = Setting not required	PMC22 = 1	PFCE22 = 0	PFC22 = 1
	TOB13	Output	P22 = Setting not required	PM22 = Setting not required	PMC22 = 1	PFCE22 = 1	PFC22 = 0

Note μPD70F3454GC-8EA-A only

Table 4-14. Using Port Pin as Alternate-Function Pin (4/8)

Pin Name	Alternate Pin		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register
	Name	I/O					
P23	TOB1B2	Output	P23 = Setting not required	PM23 = Setting not required	PMC23 = 1	–	PFC23 = 0
	TIB10	Input	P23 = Setting not required	PM23 = Setting not required	PMC23 = 1	–	PFC23 = 1
P24	TOB1T3	Output	P24 = Setting not required	PM24 = Setting not required	PMC24 = 1	–	PFC24 = 0
	EVTB1	Input	P24 = Setting not required	PM24 = Setting not required	PMC24 = 1	–	PFC24 = 1
P25	TOB1B3	Output	P25 = Setting not required	PM25 = Setting not required	PMC25 = 1	–	PFC25 = 0
	TRGB1	Input	P25 = Setting not required	PM25 = Setting not required	PMC25 = 1	–	PFC25 = 1
P26	TOB10	Output	P26 = Setting not required	PM26 = Setting not required	PMC26 = 1	PFCE26 = 0	PFC26 = 0
	TOB1OFF	Input	P26 = Setting not required	PM26 = Setting not required	PMC26 = 1	PFCE26 = 0	PFC26 = 1
	INTP10	Input	P26 = Setting not required	PM26 = Setting not required	PMC26 = 1	PFCE26 = 0	PFC26 = 1
	ADTRG1	Input	P26 = Setting not required	PM26 = Setting not required	PMC26 = 1	PFCE26 = 1	PFC26 = 0
	INTADT1	Input	P26 = Setting not required	PM26 = Setting not required	PMC26 = 1	PFCE26 = 1	PFC26 = 0
P27	DMS[Notes 1, 2]	Input	P27 = Setting not required	PM27 = Setting not required	–	–	–
P30	RXDA1	Input	P30 = Setting not required	PM30 = Setting not required	PMC30 = 1	PFCE30 = 0	PFC30 = 0
	SCL	I/O	P30 = Setting not required	PM30 = Setting not required	PMC30 = 1	PFCE30 = 0	PFC30 = 1
P31	TXDA1	Output	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	PFCE31 = 0	PFC31 = 0
	SDA	I/O	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	PFCE31 = 0	PFC31 = 1

Notes 1. V850E/IG3 only

2. The P27 pin is also used for on-chip debugging. Switching between on-chip debug function and port function can be set by following shows the setting method.

Port 2 Functions	
Low-Level Input to $\overline{\text{DRST}}$ Pin	High-Level Input to $\overline{\text{DRST}}$ Pin
P27	DMS

Table 4-14. Using Port Pin as Alternate-Function Pin (5/8)

Pin Name	Alternate Pin		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register
	Name	I/O					
P32	SIB1	Input	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 0	PFC32 = 0
	RXDA2	Input	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 0	PFC32 = 1
	$\overline{CS1}$[Note]	Output	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 1	PFC32 = 0
P33	SOB1	Output	P33 = Setting not required	PM33 = Setting not required	PMC33 = 1	–	PFC33 = 0
	TXDA2	Output	P33 = Setting not required	PM33 = Setting not required	PMC33 = 1	–	PFC33 = 1
P34	$\overline{SCKB1}$	I/O	P34 = Setting not required	PM34 = Setting not required	PMC34 = 1	PFCE34 = 0	PFC34 = 0
	INTP11	Input	P34 = Setting not required	PM34 = Setting not required	PMC34 = 1	PFCE34 = 0	PFC34 = 1
	$\overline{CS0}$[Note]	Output	P34 = Setting not required	PM34 = Setting not required	PMC34 = 1	PFCE34 = 1	PFC34 = 0
P35	SIB2	Input	P35 = Setting not required	PM35 = Setting not required	PMC35 = 1	PFCE35 = 0	PFC35 = 0
	RXDB	Input	P35 = Setting not required	PM35 = Setting not required	PMC35 = 1	PFCE35 = 0	PFC35 = 1
P36	SOB2	Output	P36 = Setting not required	PM36 = Setting not required	PMC36 = 1	PFCE36 = 0	PFC36 = 0
	TXDB	Output	P36 = Setting not required	PM36 = Setting not required	PMC36 = 1	PFCE36 = 0	PFC36 = 1
P37	$\overline{SCKB2}$	I/O	P30 = Setting not required	PM37 = Setting not required	PMC37 = 1	PFCE37 = 0	PFC37 = 0
	INTP12	Input	P30 = Setting not required	PM37 = Setting not required	PMC37 = 1	PFCE37 = 0	PFC37 = 1
	ASTB[Note]	Output	P30 = Setting not required	PM37 = Setting not required	PMC37 = 1	PFCE37 = 1	PFC37 = 0
P40	SIB0	Input	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	–	PFC40 = 0
	RXDA0	Input	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	–	PFC40 = 1

Note μPD70F3454GC-8EA-A only

Table 4-14. Using Port Pin as Alternate-Function Pin (6/8)

Pin Name	Alternate Pin		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register
	Name	I/O					
P41	SOB0	Output	P41 = Setting not required	PM41 = Setting not required	PMC41 = 1	PFCE41 = 0	PFC41 = 0
	TXDA0	Output	P41 = Setting not required	PM41 = Setting not required	PMC41 = 1	PFCE41 = 0	PFC41 = 1
	DCK[Notes 1, 2]	Input	P41 = Setting not required	PM41 = Setting not required	PMC41 = Setting not required	PFCE41 = Setting not required	PFC41 = Setting not required
P42	$\overline{\text{SCKB0}}$	I/O	P42 = Setting not required	PM42 = Setting not required	PMC42 = 1	PFCE42 = 0	PFC42 = 0
	INTP13	Input	P42 = Setting not required	PM42 = Setting not required	PMC42 = 1	PFCE42 = 0	PFC42 = 1
	DDI[Notes 1, 2]	Input	P42 = Setting not required	PM42 = Setting not required	PMC42 = Setting not required	PFCE42 = Setting not required	PFC42 = Setting not required
P43	TECR1	Input	P43 = Setting not required	PM43 = Setting not required	PMC43 = 1	PFCE43 = 0	PFC43 = 0
	TIT10	Input	P43 = Setting not required	PM43 = Setting not required	PMC43 = 1	PFCE43 = 0	PFC43 = 0
	TOT10	Output	P43 = Setting not required	PM43 = Setting not required	PMC43 = 1	PFCE43 = 0	PFC43 = 1
	INTP14	Input	P43 = Setting not required	PM43 = Setting not required	PMC43 = 1	PFCE43 = 1	PFC43 = 0
P44	TENC10	Input	P44 = Setting not required	PM44 = Setting not required	PMC44 = 1	PFCE44 = 0	PFC44 = 0
	EVTT1	Input	P44 = Setting not required	PM44 = Setting not required	PMC44 = 1	PFCE44 = 0	PFC44 = 1
	INTP15	Input	P44 = Setting not required	PM44 = Setting not required	PMC44 = 1	PFCE44 = 1	PFC44 = 0
	$\overline{\text{WAIT}}$[Note 3]	Input	P44 = Setting not required	PM44 = Setting not required	PMC44 = 1	PFCE44 = 1	PFC44 = 1

Notes 1. V850E/IG3 only

2. The P41 and P42 pins are also used for on-chip debugging. Switching between on-chip debug function and port function (in be set by using the $\overline{\text{DRST}}$ pin level. The following shows the setting method.

Port 4 Functions	
Low-Level Input to $\overline{\text{DRST}}$ Pin	High-Level Input to $\overline{\text{DRST}}$ Pin
P41/SOB0/TXDA0	DCK
P42/$\overline{\text{SCKB0}}$/INTP13	DDI

3. μPD70F3454GC-8EA-A only

Table 4-14. Using Port Pin as Alternate-Function Pin (7/8)

Pin Name	Alternate Pin		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register
	Name	I/O					
P45	TENC11	Input	P45 = Setting not required	PM45 = Setting not required	PMC45 = 1	PFCE45 = 0	PFC45 = 0
	TIT11	Input	P45 = Setting not required	PM45 = Setting not required	PMC45 = 1	PFCE45 = 0	PFC45 = 0
	TOT11	Output	P45 = Setting not required	PM45 = Setting not required	PMC45 = 1	PFCE45 = 0	PFC45 = 1
	INTP16	Input	P45 = Setting not required	PM45 = Setting not required	PMC45 = 1	PFCE45 = 1	PFC45 = 0
	$\overline{WR1}$ Note 1	Output	P45 = Setting not required	PM45 = Setting not required	PMC45 = 1	PFCE45 = 1	PFC45 = 1
P46	TOA40	Output	P46 = Setting not required	PM46 = Setting not required	PMC46 = 1	PFCE46 = 0	PFC46 = 0
	TIA40	Input	P46 = Setting not required	PM46 = Setting not required	PMC46 = 1	PFCE46 = 0	PFC46 = 1
	INTP17	Input	P46 = Setting not required	PM46 = Setting not required	PMC46 = 1	PFCE46 = 1	PFC46 = 0
	$\overline{WR0}$ Note 1	Output	P46 = Setting not required	PM46 = Setting not required	PMC46 = 1	PFCE46 = 1	PFC46 = 1
P47	TOA41	Output	P47 = Setting not required	PM47 = Setting not required	PMC47 = 1	PFCE47 = 0	PFC47 = 0
	TIA41	Input	P47 = Setting not required	PM47 = Setting not required	PMC47 = 1	PFCE47 = 0	PFC47 = 1
	INTP18	Input	P47 = Setting not required	PM47 = Setting not required	PMC47 = 1	PFCE47 = 1	PFC47 = 0
	$\overline{RD}$ Note 1	Output	P47 = Setting not required	PM47 = Setting not required	PMC47 = 1	PFCE47 = 1	PFC47 = 1
P70	ANI20	Input	P70 = Setting not required	–	PMC70 = 1	–	–
P71	ANI21	Input	P71 = Setting not required	–	PMC71 = 1	–	–
P72	ANI22	Input	P72 = Setting not required	–	PMC72 = 1	–	–
P73	ANI23	Input	P73 = Setting not required	–	PMC73 = 1	–	–
P74 Note 2	ANI24 Note 2	Input	P74 = Setting not required	–	PMC74 = 1	–	–
P75 Note 2	ANI25 Note 2	Input	P75 = Setting not required	–	PMC75 = 1	–	–
P76 Note 2	ANI26 Note 2	Input	P76 = Setting not required	–	PMC76 = 1	–	–
P77 Note 2	ANI27 Note 2	Input	P77 = Setting not required	–	PMC77 = 1	–	–

Notes 1. μPD70F3454GC-8EA-A only

2. V850E/IG3 only

Table 4-14. Using Port Pin as Alternate-Function Pin (8/8)

Pin Name	Alternate Pin		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register
	Name	I/O					
PDL0	AD0[Note 1]	I/O	PDL0 = Setting not required	PMDL0 = Setting not required	PMCDL0 = 1	–	–
PDL1	AD1[Note 1]	I/O	PDL1 = Setting not required	PMDL1 = Setting not required	PMCDL1 = 1	–	–
PDL2	AD2[Note 1]	I/O	PDL2 = Setting not required	PMDL2 = Setting not required	PMCDL2 = 1	–	–
PDL3	AD3[Note 1]	I/O	PDL3 = Setting not required	PMDL3 = Setting not required	PMCDL3 = 1	–	–
PDL4	AD4[Note 1]	I/O	PDL4 = Setting not required	PMDL4 = Setting not required	PMCDL4 = 1	–	–
PDL5	AD5[Note 1]	I/O	PDL5 = Setting not required	PMDL5 = Setting not required	PMCDL5 = 1	–	–
	FLMD1[Note 2]	Input	PDL5 = Setting not required	PMDL5 = Setting not required	PMCDL5 = Setting not required	–	–
PDL6	AD6[Note 1]	I/O	PDL6 = Setting not required	PMDL6 = Setting not required	PMCDL6 = 1	–	–
PDL7	AD7[Note 1]	I/O	PDL7 = Setting not required	PMDL7 = Setting not required	PMCDL7 = 1	–	–
PDL8	AD8[Note 1]	I/O	PDL8 = Setting not required	PMDL8 = Setting not required	PMCDL8 = 1	–	–
PDL9	AD9[Note 1]	I/O	PDL9 = Setting not required	PMDL9 = Setting not required	PMCDL9 = 1	–	–
PDL10[Note 3]	AD10[Note 1]	I/O	PDL10 = Setting not required	PMDL10 = Setting not required	PMCDL10 = 1	–	–
PDL11[Note 3]	AD11[Note 1]	I/O	PDL11 = Setting not required	PMDL11 = Setting not required	PMCDL11 = 1	–	–
PDL12[Note 3]	AD12[Note 1]	I/O	PDL12 = Setting not required	PMDL12 = Setting not required	PMCDL12 = 1	–	–
PDL13[Note 3]	AD13[Note 1]	I/O	PDL13 = Setting not required	PMDL13 = Setting not required	PMCDL13 = 1	–	–
PDL14[Note 3]	AD14[Note 1]	I/O	PDL14 = Setting not required	PMDL14 = Setting not required	PMCDL14 = 1	–	–
PDL15[Note 3]	AD15[Note 1]	I/O	PDL15 = Setting not required	PMDL15 = Setting not required	PMCDL15 = 1	–	–

Notes 1. μPD70F3454GC-8EA-A only

2. The PDL5 pin is also used for a pin (FLMD1) to be set in the flash programming mode. This pin does not need to be mar register. For details, see **CHAPTER 27 FLASH MEMORY**.

3. V850E/IG3 only

4.6 Noise Eliminator

A timing controller used to secure the noise elimination time is provided for the following pins. Input signals that change within the noise elimination time are not internally acknowledged.

Table 4-15. Noise Eliminator (1/2)

Target Pin		Filter Type	Noise Elimination Width	Sampling Clock
$\overline{\text{RESET}}$		Analog filter	Several 10 ns	–
$\overline{\text{DRST}}$[Note 1]				
FLMD0				
P00/TOA20/TIA20/TOA2OFF/INTP00	TIA20	Digital filter	2, 3 clocks	fxx, fxx/4 selectable
	TOA2OFF	Analog filter	Several 10 ns	–
	INTP00			
P01/TOA21/TIA21/INTP01	TIA21	Digital filter	2, 3 clocks	fxx, fxx/4 selectable
	INTP01	Analog filter	Several 10 ns	–
P02[Note 1]/TOA30[Note 1]/TIA30[Note 1]/TOA3OFF[Note 1]/INTP02[Note 1]	TIA30[Note 1]	Digital filter	2, 3 clocks	fxx, fxx/4 selectable
	TOA3OFF[Note 1]	Analog filter	Several 10 ns	–
	INTP02[Note 1]			
P03[Note 1]/TOA31[Note 1]/TIA31[Note 1]/INTP03[Note 1]	TIA31[Note 1]	Digital filter	2, 3 clocks	fxx, fxx/4 selectable
	INTP03[Note 1]	Analog filter	Several 10 ns	–
P04[Note 1]/TECR0[Note 1]/TIT00[Note 1]/TOT00[Note 1]/INTP04[Note 1]	TECR0[Note 1]/TIT00[Note 1]	Digital filter	2, 3 clocks	fxx, fxx/4, fxx/8, fxx/16, fxx/32, fxx/64 selectable
	INTP04[Note 1]	Analog filter	Several 10 ns	–
P05[Note 1]/TENC00[Note 1]/EVTT0[Note 1]/INTP05[Note 1]	TENC00[Note 1]	Digital filter	2, 3 clocks	fxx, fxx/4, fxx/8, fxx/16, fxx/32, fxx/64 selectable
	EVTT0[Note 1]			
	INTP05[Note 1]	Analog filter	Several 10 ns	–
P06[Note 1]/TENC01[Note 1]/TIT01[Note 1]/TOT01[Note 1]/INTP06[Note 1]	TENC01[Note 1]/TIT01[Note 1]	Digital filter	2, 3 clocks	fxx, fxx/4, fxx/8, fxx/16, fxx/32, fxx/64 selectable
	INTP06[Note 1]	Analog filter	Several 10 ns	–
P07[Note 1]/INTP07[Note 1]/CLKOUT[Note 2]	INTP07[Note 1]			
P10/TOB0T1/TIB01/TOB01/A0[Note 2]	TIB01	Digital filter	2, 3 clocks	fxx/4
P11/TOB0B1/TIB02/TOB02/A1[Note 2]	TIB02			
P12/TOB0T2/TIB03/TOB03/A2[Note 2]	TIB03			
P13/TOB0B2/TIB00/A3[Note 2]	TIB00			
P14/TOB0T3/EVTB0/A4[Note 2]	EVTB0			
P15/TOB0B3/TRGB0/A5[Note 2]	TRGB0			

Notes 1. V850E/IG3 only

2. μPD70F3454GC-8EA-A only

Cautions 1. The maskable interrupt pins are used to release the standby mode.

2. The digital filter uses clock sampling and therefore cannot acknowledge an input signal when the peripheral clock (fxx) is stopped.

3. The noise eliminator is valid only in the alternate-function mode.

Table 4-15. Noise Eliminator (2/2)

Target Pin		Filter Type	Noise Elimination Width	Sampling Clock
P16/TOB0OFF/INTP08/ADTRG0/INTADT0/ A6[Note 1]	TOB0OFF	Analog filter	Several 10 ns	–
	INTP08			
	ADTRG0			
	INTADT0			
P17/TOB00/INTP09/A7[Note 1]	INTP09			
P20/TOB1T1/TIB11/TOB11	TIB11	Digital filter	2, 3 clocks	fxx/4
P21/TOB1B1/TIB12/TOB12	TIB12			
P22/TOB1T2/TIB13/TOB13	TIB13			
P23/TOB1B2/TIB10	TIB10			
P24/TOB1T3/EVTB1	EVTB1			
P25/TOB1B3/TRGB1	TRGB1			
P26/TOB10/TOB1OFF/INTP10/ADTRG1/ INTADT1	TOB1OFF	Analog filter	Several 10 ns	–
	INTP10			
	ADTRG1			
	INTADT1			
P34/$\overline{\text{SCKB1}}$/INTP11/$\overline{\text{CS0}}$[Note 1]	INTP11			
P37/$\overline{\text{SCKB2}}$/INTP12/ASTB[Note 1]	INTP12			
P42/$\overline{\text{SCKB0}}$/INTP13/DDI[Note 2]	INTP13			
P43/TECR1/TIT10/TOT10/INTP14	TECR1/ TIT10	Digital filter	2, 3 clocks	fxx, fxx/4, fxx/8, fxx/16, fxx/32, fxx/64 selectable
	INTP14			fxx/4, fxx/16, fxx/64, fxx/128, fxx/256, fxx/512 selectable
P44/TENC10/EVTT1/INTP15/$\overline{\text{WAIT}}$[Note 1]	TENC10			fxx, fxx/4, fxx/8, fxx/16, fxx/32, fxx/64 selectable
	EVTT1			
	INTP15			fxx/4, fxx/16, fxx/64, fxx/128, fxx/256, fxx/512 selectable
P45/TENC11/TIT11/TOT11/INTP16/$\overline{\text{WR1}}$[Note 1]	TENC11/ TIT11			fxx/, fxx/4, fxx/8, fxx/16, fxx/32, fxx/64 selectable
	INTP16			fxx/4, fxx/16, fxx/64, fxx/128, fxx/256, fxx/512 selectable
P46/TOA40/TIA40/INTP17/$\overline{\text{WR0}}$[Note 1]	TIA40			fxx, fxx/4 selectable
	INTP17	Analog filter	Several 10 ns	–
P47/TOA41/TIA41/INTP18/$\overline{\text{RD}}$[Note 1]	TIA41	Digital filter	2, 3 clocks	fxx, fxx/4 selectable
	INTP18	Analog filter	Several 10 ns	–

Notes 1. μPD70F3454GC-8EA-A only

2. V850E/IG3 only

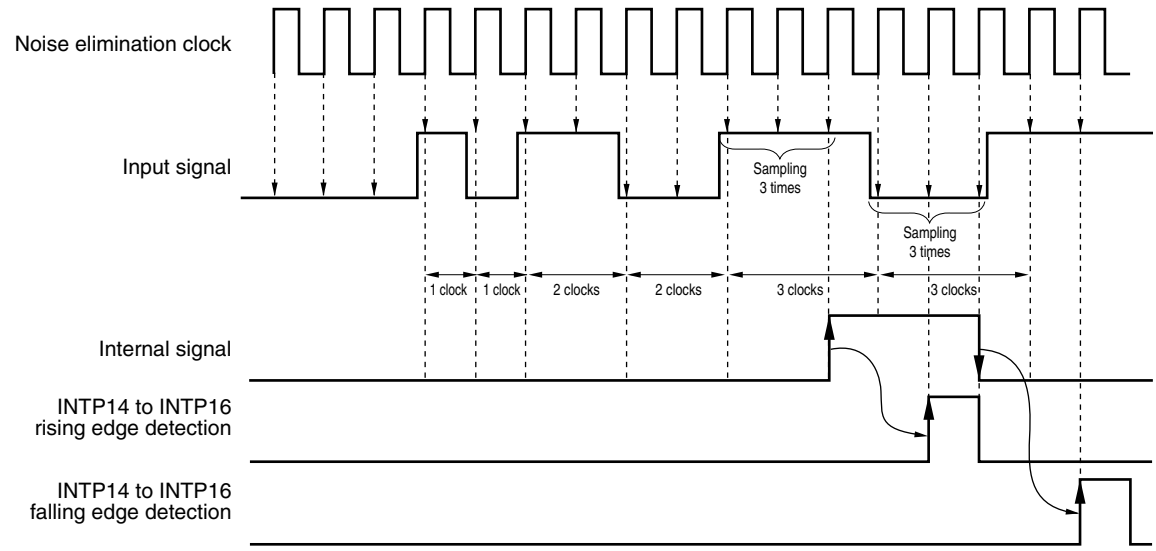
Cautions 1. The maskable interrupt pins (INTADT0, INTADT1, INTP14 to INTP16) are used to release the standby mode.

2. The digital filter uses clock sampling and therefore cannot acknowledge an input signal when the peripheral clock (fxx) is stopped.

3. The noise eliminator is valid only in the alternate-function mode.

An example of timing of noise elimination by digital filter for INTP14 to INTP16, timer AA input pin, and timer T input pin is shown below.

Figure 4-4. Example of Noise Elimination Timing



Caution If there are two or fewer noise elimination clocks while the INTP14 to INTP16 input signals are high level (or low level), the input signal is eliminated as noise. If it is sampled three times or more, the edge is detected as a valid input.

(1) Digital noise elimination 0 control register n (INTNFCn)

The INTNFCn register is used to select the sampling clock that is used to eliminate digital noise on the INTPn pin. If the same level is not detected on this pin three times in sequence using the clock selected by the INTNFCn register, the signal is eliminated as noise.

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

Reset sets this register to 00H.

Cautions 1. If the input signal lasts for the duration of 2 or 3 clocks, it is undefined whether the signal is detected as a valid edge or eliminated as noise. So that the signal is actually detected as a valid edge, the same signal level must be input for a duration of 3 clocks or more.

2. If noise is generated in synchronization with the sampling clock, eliminate the noise by attaching a filter to the input pin.

3. Noise is not eliminated if the pin is used as a normal input port pin.

After reset: 00H R/W Address: INTNFC14 FFFFF310H, INTNFC15 FFFFF312H, INTNFC16 FFFFF314H

INTNFCn (n = 14 to 16)

	7	6	5	4	3	2	1	0
INTNFCn	INTNFENn	0	0	0	0	INTNFCn2	INTNFCn1	INTNFCn0

INTNFENn	Setting of digital noise elimination
0	Disables digital noise elimination
1	Enables digital noise elimination

INTNFCn2	INTNFCn1	INTNFCn0	Sampling clock selection
0	0	0	$f_{XX}/4$
0	0	1	$f_{XX}/16$
0	1	0	$f_{XX}/64$
0	1	1	$f_{XX}/128$
1	0	0	$f_{XX}/256$
1	0	1	$f_{XX}/512$
Other than above			Setting prohibited

(2) Digital noise elimination 1 control register n (TANFCn)

The TANFCn register is used to select the sampling clock that is used to eliminate digital noise on the TIAn0 or TIAn1 pin. If the same level is not detected on these pins three times in sequence using the clock selected by the TANFCn register, the signal is eliminated as noise.

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

Reset sets this register to 00H.

Cautions 1. If the input signal lasts for the duration of 2 or 3 clocks, it is undefined whether the signal is detected as a valid edge or eliminated as noise. So that the signal is actually detected as a valid edge, the same signal level must be input for a duration of 3 clocks or more.

2. If noise is generated in synchronization with the sampling clock, eliminate the noise by attaching a filter to the input pin.

3. Noise is not eliminated if the pin is used as a normal input port pin.

4. The noise elimination function starts operating when the TAAnCTL0.TAAnCE bit is set to 1 (enabling count operations).

After reset: 00H R/W Address: TANFC2 FFFFFB40H, TANFC3 FFFFFB42H[Note], TANFC4 FFFFFB44H

	7	6	5	4	3	2	1	0
TANFCn	TANFENn	0	0	0	0	0	0	TANFCn0

(V850E/IF3 n = 2, 4)

(V850E/IG3 n = 2 to 4)

TANFENn	Setting of digital noise elimination
0	Disables digital noise elimination
1	Enables digital noise elimination

TANFCn0	Sampling clock selection
0	f_{XX}
1	$f_{XX}/4$

Note V850E/IG3 only

(3) Digital noise elimination 2 control register n (TTNFCn)

The TTNFCn register is used to select the sampling clock that is used to eliminate digital noise on the TITn0, TITn1, EVTTn, TENCn0, TENCn1, or TECRn pin. If the same level is not detected on these pins three times in sequence using the clock selected by the TTNFCn register, the signal is eliminated as noise.

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

Reset sets this register to 00H.

Cautions 1. If the input signal lasts for the duration of 2 or 3 clocks, it is undefined whether the signal is detected as a valid edge or eliminated as noise. So that the signal is actually detected as a valid edge, the same signal level must be input for a duration of 3 clocks or more.

2. If noise is generated in synchronization with the sampling clock, eliminate the noise by attaching a filter to the input pin.

3. Noise is not eliminated if the pin is used as a normal input port pin.

4. The noise elimination function starts operating when the TTnCTL0.TTnCE bit is set to 1 (enabling count operations).

After reset: 00H R/W Address: TTNFC0 FFFFF5A0H[Note], TTNFC1 FFFFF5A2H

	7	6	5	4	3	2	1	0
TTNFCn	TTNFENn	0	0	0	0	TTNFCn2	TTNFCn1	TTNFCn0

(V850E/IF3 n = 1)

(V850E/IG3 n = 0, 1)

TTNFENn	Setting of digital noise elimination
0	Disables digital noise elimination
1	Enables digital noise elimination

TTNFCn2	TTNFCn1	TTNFCn0	Sampling clock selection
0	0	0	f_{XX}
0	0	1	$f_{XX}/4$
0	1	0	$f_{XX}/8$
0	1	1	$f_{XX}/16$
1	0	0	$f_{XX}/32$
1	0	1	$f_{XX}/64$
Other than above			Setting prohibited

Note V850E/IG3 only

4.7 Cautions

4.7.1 Cautions on setting port pins

(1) Set the registers of a port in the following sequence.

<1> Set PFCn and PFCEn registers.
<2> Set PMCn register.
<3> Set INTFn and INTRn registers.

If the PMCn register is set before setting the PFCn and PFCEn registers, an unexpected peripheral function may be selected while the PFCn and PFCEn registers are being set.

(2) An on-chip pull-up resistor can only be connected when the pins are in input mode in the port mode, or when the pins function as input pins in the alternate-function mode.
Moreover, for the V850E/IF3, an on-chip pull-up resistor can be connected to the TOB0T1 to TOB0T3, TOB0B1 to TOB0B3, and TOA21 pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOB0OFF, TOA2OFF, and TOB1OFF pins or software processing.
For the V850E/IG3, an on-chip pull-up resistor can be connected to the TOB0T1 to TOB0T3, TOB0B1 to TOB0B3, TOA21, TOB1T1 to TOB1T3, TOB1B1 to TOB1B3, and TOA31 pins, these are output pins in the alternate-function mode, when these pins go into a high-impedance state due to the TOB0OFF, TOA2OFF, TOB1OFF, and TOA3OFF pins or software processing.
Set the on-chip pull-up resistor in the following sequence.

<1> Set PMCn register.
<2> Set PMn register.
<3> Set PU register.

(3) Set the N-ch open-drain in the following sequence.
• Used in port mode
<1> Set PMCn register.
<2> Set PFn register.

• Used as output pin in alternate-function mode of I^2C
<1> Set PFCn and PFCEn registers.
<2> Set PFn register.
<3> Set PMCn register.

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

When a 1-bit manipulation instruction is executed on a port that provides both input and output functions, the value of the output latch of an input port that is not subject to manipulation may be written in addition to the targeted bit.

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

<Example> When P20 pin is an output port, P21 to P27 pins are input ports (all pin statuses are high level), and the value of the port latch is 00H, if the output of P20 pin is changed from low level to high level via a bit manipulation instruction, the value of the port latch is FFH.

Explanation: The target bits of writing to and reading from the Pn register of a port whose PMnm bit is 1 are in the output latch status and pin status, respectively.

A bit manipulation instruction is executed in the following order in the V850E/IF3 and V850E/IG3.

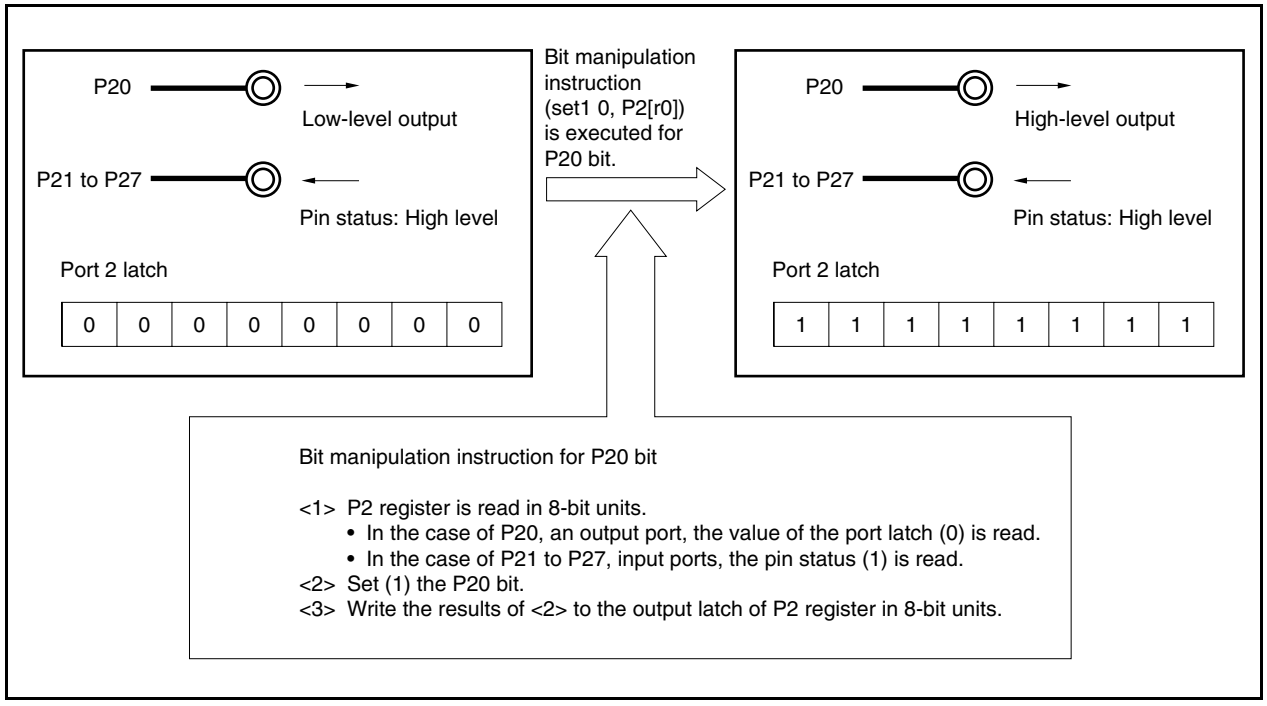
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the value of the output latch (0) of P20 pin, which is an output port, is read, while the pin statuses of P21 to P27 pins, which are input ports, are read. If the pin statuses of P21 to P27 pins are high level at this time, the read value is FEH.

The value is changed to FFH by the manipulation in <2>.

FFH is written to the output latch by the manipulation in <3>.

Figure 4-5. Bit Manipulation Instruction (P20 Pin)



CHAPTER 5 CLOCK GENERATOR

5.1 Overview

The features of clock generator are as follows.

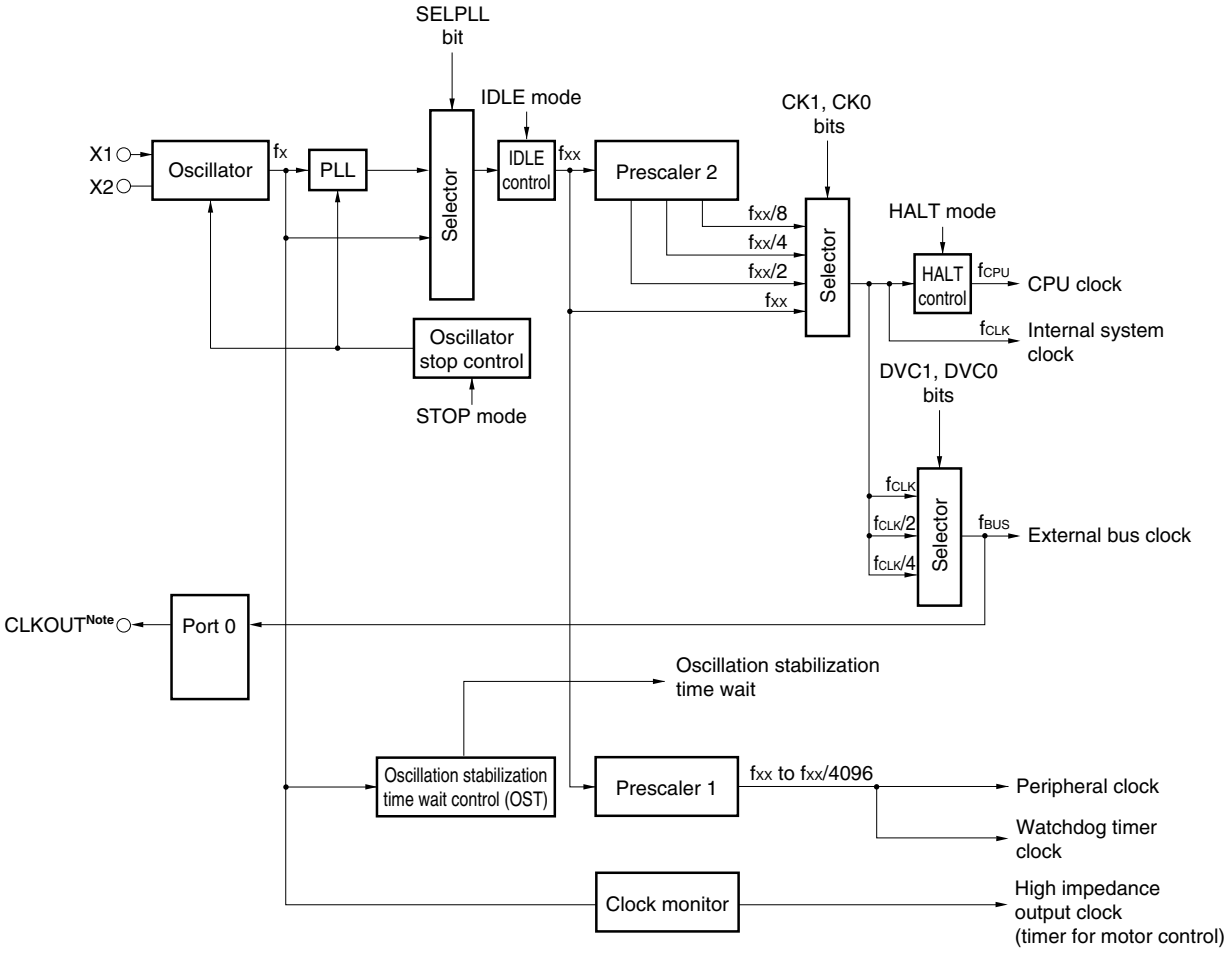
- ○ Oscillator
  - In PLL mode: f_X = 4 to 8 MHz (f_{XX} = 32 to 64 MHz)
  - In clock-through mode: f_X = 4 to 8 MHz (f_{XX} = 4 to 8 MHz)
- ○ Multiply (×8 fixed) function by PLL (Phase Locked Loop)
  - Clock-through mode/PLL mode selectable
- ○ Internal system clock generation
  - 4 steps (f_{XX}, f_{XX}/2, f_{XX}/4, f_{XX}/8)
- ○ Peripheral clock generation
- ○ Oscillation stabilization time selection

Caution The oscillation guaranteed range is 4 to 8 MHz.

Remark f_X: Oscillation frequency
f_{XX}: System clock frequency

5.2 Configuration

Figure 5-1. Clock Generator



Note μPD70F3454GC-8EA-A only

Caution Because f_{CPU} and f_{CLK} do not go through PLL immediately after reset, and $f_{XX}/8$ is selected by prescaler 2, if f_X = 8 MHz, f_{CPU} and f_{CLK} are 1 MHz.

Remark f_X: Oscillation frequency
f_{XX}: System clock frequency
f_{CPU}: CPU cock frequency
f_{CLK}: Internal system clock frequency
f_{BUS}: External bus clock frequency

Table 5-1. Operation Clock of Each Function Block

Function Block		Operation Clock
CPU		f_{CPU} (selected from f_{XX} to $f_{XX}/8$ by PCC register)
DMA, interrupt controller		f_{CLK} (selected from f_{XX} to $f_{XX}/8$ by PCC register)
TAA	TAA0, TAA1	f_{XX}
	TAA2 to TAA4	$f_{XX}/2$
TAB		f_{XX}
TMT		$f_{XX}/2$
TMM		$f_{XX}/2$
Watchdog timer		$f_{XX}/1024$
UARTA		f_{UCLK} (selected from $f_{XX}/2$ to $f_{XX}/4096$ by UAnCTL1 register)
UARTB		f_{XX}
CSIB		f_{CCLK} (selected from $f_{XX}/4$ to $f_{XX}/256$ by CBnCTL1 register)
I^2C		$f_{XX}/2$
Bus control function		f_{BUS} (selected from $f_{CLK}/1$, $f_{CLK}/2$, $f_{CLK}/4$ by DVC register)
A/D converters 0, 1		f_{AD01} (selected from $f_{XX}/2$ to $f_{XX}/4$ by ADnOCKS register)
A/D converter 2		$f_{AD2} = f_{XX}/2$

Remarks 1. f_{CPU}: CPU cock frequency
f_{XX}: Peripheral clock frequency
f_{CLK}: Internal system clock frequency
f_{UCLK}: Base clock frequency of UARTA0 to UARTA2
f_{CCLK}: Base clock frequency of CSIB0 to CSIB2
f_{AD01}: Base clock frequency of A/D converters 0 and 1
f_{AD2}: Operating clock frequency of A/D converter 2
f_{BUS}: External bus clock frequency

2. n = 0, 1

(1) Oscillator

The main resonator oscillates the following frequencies (f_X):

- In PLL mode (×8 fixed): f_X = 4 to 8 MHz (f_{XX} = 32 to 64 MHz)
- In clock-through mode: f_X = 4 to 8 MHz (f_{XX} = 4 to 8 MHz)

(2) IDLE control

All functions other than the oscillator, PLL, clock monitor operation, CSIB in slave mode, low-voltage detector (LVI), and power-on-clear circuit (POC) are stopped.

(3) HALT control

Only the CPU clock (f_{CPU}) is stopped.

(4) PLL

This circuit multiplies the clock generated by the oscillator (f_X) by 8.

It operates in two modes: clock-through mode in which f_X is output as is by setting the SELPLL bit of the PLL control register (PLLCTL), and PLL mode in which a multiplied clock is output.

(5) Prescaler 1

This prescaler generates the clock (f_{XX} to f_{XX}/4096) to be supplied to on-chip peripheral functions.

(6) Prescaler 2

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

The clock (f_{XX} to f_{XX}/8) to be supplied to the CPU clock (f_{CPU}) and internal system clock (f_{CLK}) is generated.

(7) Oscillation stabilization time wait control (OST)

This unit measures the time from when the clock generated by the oscillator was input until oscillation is stabilized. It also counts the PLL lockup time.

The count clock can be selected from $2^{14}/f_X$ to $2^{18}/f_X$.

(8) Clock monitor

The clock monitor samples the clock generated by the oscillator (f_X), by using the internal oscillation clock. When it detects stop of oscillation, output of the timer for motor control goes into a high-impedance state (for details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**).

5.3 Control Registers

The clock generator is controlled by the following six registers.

- PLL control register (PLLCTL)
- Processor clock control register (PCC)
- Power save control register (PSC)
- Power save mode register (PSMR)
- Oscillation stabilization time select register (OSTS)
- Clock monitor mode register (CLM)

(1) PLL control register (PLLCTL)

The PLLCTL register selects CPU operation clock.

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

Reset sets this register to 01H.

After reset: 01H R/W Address: FFFFF82CH

	7	6	5	4	3	2	<1>	0
PLLCTL	0	0	0	0	0	0	SELPLL	1

SELPLL	CPU operation clock selection
0	Clock-through mode
1	PLL mode

Cautions 1. Be sure to set bits 7 to 2 to “0” and set bit 0 to “1”.

2. Setting the SELPLL bit to 1 is enabled only when the PLL clock frequency is stabilized. If the SELPLL bit is rewritten when the PLL clock frequency is not stabilized (during unlock), 0 is written to the bit. Therefore, be sure to confirm that the PLL mode has been set. Use the following program for reference.

```
_loop:  set1   1, PLLCTL
        tst1   1, PLLCTL
        bz     _loop
(next instruction)
```

(2) Processor clock control register (PCC)

The PCC register is a special register. Data can be written to this register only in a combination of specific sequences (see **3.4.8 Special registers**).

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

Reset sets this register to 03H.

After reset: 03H R/W Address: FFFFF828H

	7	6	5	4	3	2	1	0
PCC	0	0	0	0	0	0	CK1	CK0

CK1	CK0	Clock selection (f_{CLK}/f_{CPU})
0	0	f_{XX}
0	1	$f_{XX}/2$
1	0	$f_{XX}/4$
1	1	$f_{XX}/8$

Cautions 1. Be sure to set bits 2 to 7 to "0".

2. Set the PCC register to 00H after the PLL mode is selected (PLLCTL.SELPLL bit = 1).

(3) Power save control register (PSC)

The PSC register is an 8-bit register that controls the standby function and specifies the standby mode by setting the STB bit. The PSC register is a special register (see **3.4.8 Special registers**). Data can be written to this register only in a combination of specific sequences.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF1FEH

	7	6	5	<4>	3	2	<1>	0
PSC	0	0	0	INTM	0	0	STB	0

INTM	Standby mode control[Note 2] by maskable interrupt request (INTxx[Note 1])
0	Standby mode release by INTxx request enabled
1	Standby mode release by INTxx request disabled

STB	Sets operation mode
0	Normal mode
1	Standby mode

Notes 1. For details, see **Table 20-1 Interrupt Source List**.

2. Setting is valid only in the IDLE mode and STOP mode.

Cautions 1. Be sure to set bits 0, 2, 3, and 5 to 7 to "0".

2. Before setting a standby mode by setting the STB bit to 1, be sure to set the PCC register to 03H and then set the STB bit to 1. Otherwise, the standby mode may not be set or released. After releasing the standby mode, change the value of the PCC register to the desired value.

3. To set the IDLE mode or STOP mode, set the PCC register to 03H, and the PSMR.PSM0 bit in that order and then set the STB bit to 1.

(4) Power save mode register (PSMR)

The PSMR register is an 8-bit register that controls the operation in the software standby mode.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF820H

	7	6	5	4	3	2	1	<0>
PSMR	0	0	0	0	0	0	0	PSM0

PSM0	Specifies operation in software standby mode
0	IDLE mode
1	STOP mode

Cautions 1. Be sure to set bits 1 to 7 to "0".

2. The PSM0 bit is valid only when the PSC.STB bit is 1.

(5) Oscillation stabilization time select register (OSTS)

The OSTS register selects the oscillation stabilization time until the oscillation stabilizes after the STOP mode is released by interrupt request.

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

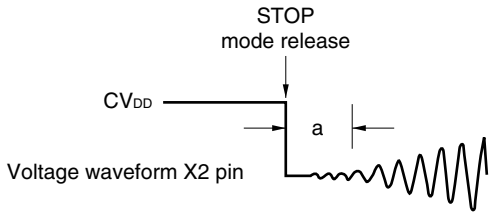
Reset sets this register to 04H.

After reset: 04H R/W Address: FFFFF6C0H

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	OSTS3	OSTS2	OSTS1	OSTS0

OSTS3	OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time (f_X = 8 MHz)
0	1	0	0	$2^{14}/f_X$ (2.05 ms)
0	1	0	1	$2^{15}/f_X$ (4.10 ms)
0	1	1	0	$2^{16}/f_X$ (8.19 ms)
0	1	1	1	$2^{17}/f_X$ (16.4 ms)
1	0	0	0	$2^{18}/f_X$ (32.8 ms)
Other than above				Setting prohibited

Cautions 1. The wait time does not include the time until the clock oscillation starts ("a" in the figure below) following release of the STOP mode.



2. The default value of the OSTS register after reset is 04H. If an 8 MHz resonator is used, therefore, the oscillation stabilization time is about 2 ms. Half the oscillation stabilization time is consumed by waiting for the lockup of PLL. Therefore, the actual stabilization time of the resonator is about 1 ms. When releasing reset, therefore, make sure that the oscillation stabilization time is secured during the active period of the reset signal. To release the STOP mode by an interrupt input other than a reset signal ($\overline{\text{RESET}}$ pin input, reset signal (LVIRES) generation by low-voltage detector (LVI), reset signal (POCRES) generation by power-on-clear circuit (POC)), the oscillation stabilization time is determined by the set value of the OSTS register. Therefore, set a time twice as long as that required for the resonator to stabilize to the OSTS register (because half the oscillation stabilization time is the stabilization time of PLL).

3. Be sure to set bits 4 to 7 to "0".

Remark f_X: Oscillation frequency

(6) Clock monitor mode register (CLM)

The CLM register sets clock monitor operation mode. The CLM register is a special register. It can be written only in a combination of specific sequences (see **3.4.8 Special registers**).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF870H

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

CLME	Clock monitor operation control
0	Clock monitor operation disabled
1	Clock monitor operation enabled

Cautions 1. The CLME bit is cleared to 0 only after reset.

2. When the CLME bit = 1, the clock monitor function is forcibly stopped if the following conditions are satisfied.

- **During oscillation stabilization time count after release of STOP mode**
- **During break (on-chip debug emulator)**

3. When the CLME bit = 1, output of the timer for motor control goes into a high-impedance state if oscillation (f_X) stop is detected. See Figure 10-4 for the target timer output.

5.4 PLL Function

5.4.1 Overview

The CPU and the operating clock of the peripheral macro can be switched between output of the oscillation frequency multiplied by 8, and clock-through mode.

When PLL function is used: Input clock (f_X) = 4 to 8 MHz, output clock (f_{XX}) = 32 to 64 MHz
Clock-through mode: Input clock (f_X) = 4 to 8 MHz, output clock (f_{XX}) = 4 to 8 MHz

5.4.2 PLL mode

In the PLL mode, the oscillation frequency (f_X) is multiplied by 8 with the PLL to generate a system clock (f_{XX}).

In the PLL mode, the clock is input from the oscillator to the PLL. A clock at a stable frequency must be supplied to the internal circuit after the lapse of the lockup time (frequency stabilization time) during which the phase is locked at a specific frequency and oscillation is stabilized. In the V850E/IF3 and V850E/IG3, the lockup time after release of reset is secured automatically.

Caution When a resonator of f_X = 8 MHz is used and if the oscillation stabilization time of that resonator must be 3 ms (MAX.), the reset input ($\overline{\text{RESET}}$ active) width must be 2 ms (MIN.).

5.4.3 Clock-through mode

In the clock-through mode, a system clock (f_{XX}) of the same frequency as the oscillation frequency (f_X) is generated.

5.5 Operation

5.5.1 Operation of each clock

The following table shows the operation status of each clock.

Table 5-2. Operation Status of Each Clock

Power Save Mode	Oscillator (f_X)	PLL	Internal System Clock (f_{CLK})	Peripheral Clock (f_{XX} to $f_{XX}/4096$)	External Bus Clock (f_{BUS})Note 1	CPU Clock (f_{CPU})	Watchdog Timer ClockNote 2
Normal operation	√	√	√	√	√	√	√
HALT mode	√	√	√	√	√	×	√
IDLE mode	√	√	×	×	×	×	×
In STOP mode and during oscillation stabilization time count after release of STOP mode	×Note 3	×Note 3	×	×	×	×	×
During $\overline{\text{RESET}}$ pin inputNote 4 and subsequent oscillation stabilization time count	√	× → √	√	×Note 5	×Note 6	√	×

Notes 1. μPD70F3454GC-8EA-A only

2. The peripheral clock ($f_{XX}/1024$) is used as the watchdog timer clock.

3. Operation continues during on-chip debugging.

4. $\overline{\text{RESET}}$ pin input, reset signal (WDTRES) generation by watchdog timer, reset signal (LVIRES) generation by low-voltage detector (LVI), or reset signal (POCRES) generation by power-on-clear circuit (POC)

5. The output from the prescaler (PRS) in not performed.

6. The clock is not output from the CLKOUT pin.

Remark √: Operating
×: Stopped

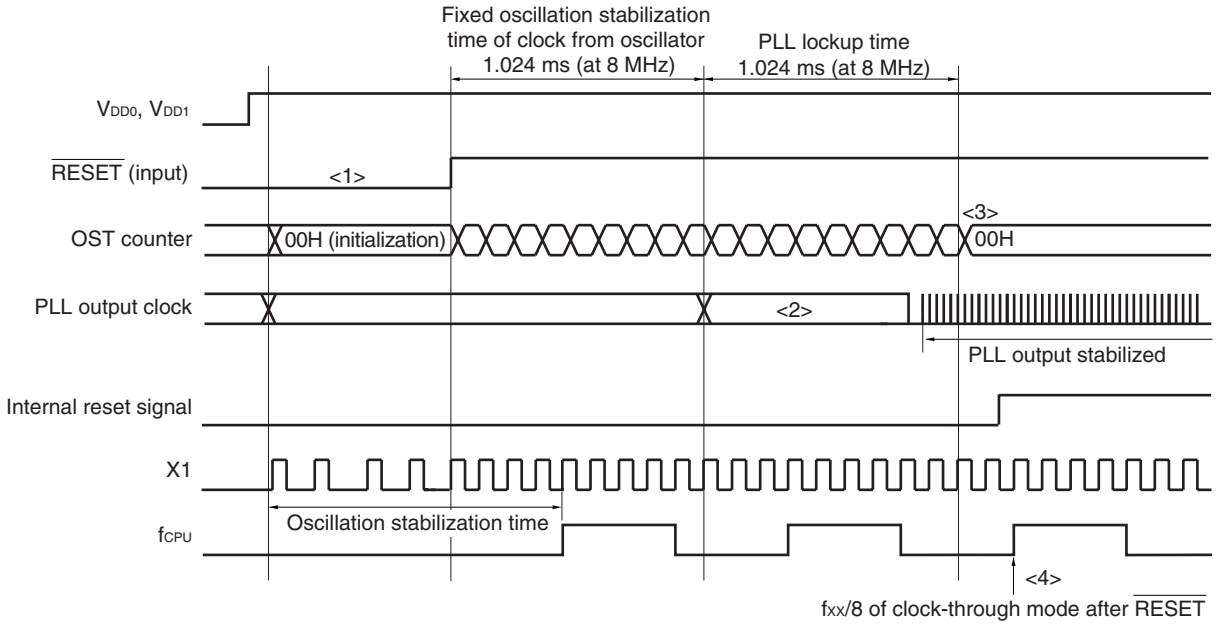
5.5.2 Clock output function

The clock output function is used to output the external bus clock (f_{BUS}) from the CLKOUT pin and supported only in the μPD70F3454GC-8EA-A.

When the internal system clock (f_{CLK}) in Table 5-2 is in the operable status (√), the pin can output the clock. When it is in the stopped status (×), the pin cannot output the clock.

5.5.3 Operation timing

(1) Power on (power-on reset)



<1> The oscillator is activated during the $\overline{\text{RESET}}$ period that follows power application.
Make sure that the low-level width of the $\overline{\text{RESET}}$ signal is “Oscillation stabilization time of the resonator – Fixed oscillation stabilization time” or more, taking the oscillation stabilization time into consideration.
PLL stops during the $\overline{\text{RESET}}$ period and fixed oscillation stabilization time.

<2> When the fixed oscillation stabilization time that elapses after the $\overline{\text{RESET}}$ signal is released expires, PLL stop is released, and counting the lockup time starts.

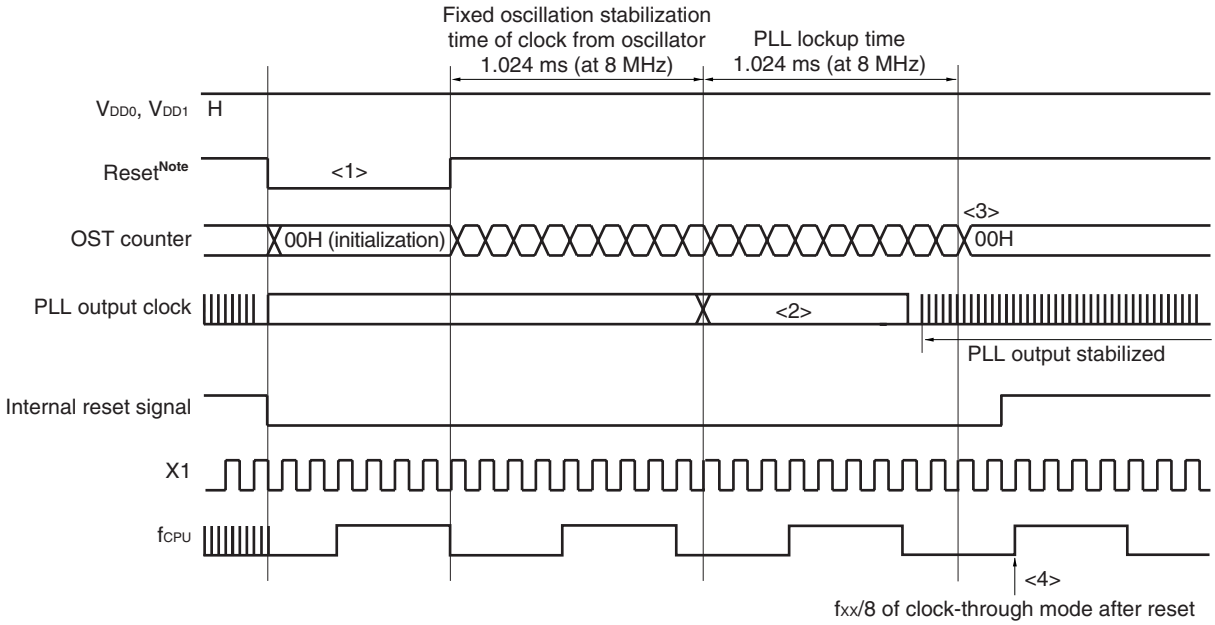
<3> PLL is locked when counting of the lockup time is over. The OST counter is initialized to 00H.

<4> When the lockup time expires, the CPU releases the reset signal and operates in the clock-through mode (f_X). The CPU operation clock (f_{CPU}) is $f_{XX}/8$. The PLL mode can be set by software.

Cautions 1. The clock generated by the oscillator starts oscillating during the $\overline{\text{RESET}}$ period. After the $\overline{\text{RESET}}$ signal is released, a specific wait time (fixed oscillation stabilization time) elapses.

2. To avoid malfunction due to noise, do not change the division ratio of the CPU operation clock (f_{CPU}) by using the PCC register before setting the PLL mode. Before changing the division ratio, be sure to select the PLL mode.

(2) Reset input with power on



<1> The oscillator continues operating during the reset period.
PLL stops during the reset period and fixed oscillation stabilization time.

<2> When the fixed oscillation stabilization time that elapses after the reset signal is released expires, PLL stop is released, and counting the lockup time starts.

<3> PLL is locked when counting of the lockup time is over. The OST counter is initialized to 00H.

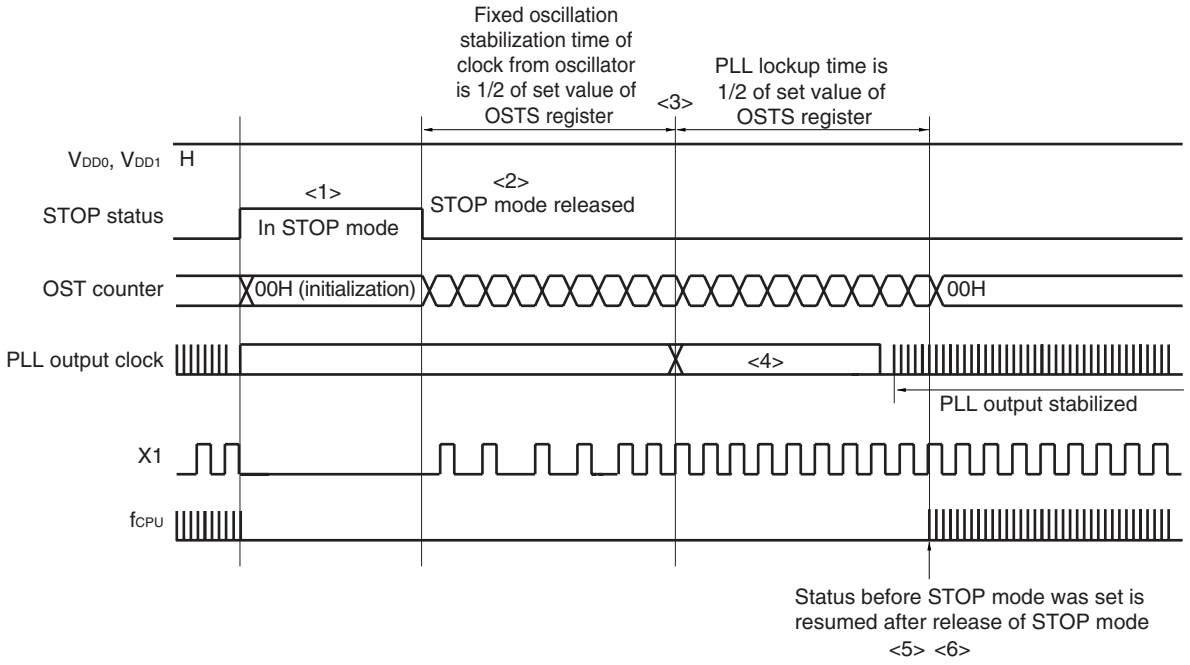
<4> When the lockup time expires, the CPU releases the reset signal and operates in the clock-through mode (f_X). The CPU operation clock (f_{CPU}) is $f_{XX}/8$. The PLL mode can be set by software.

Note $\overline{\text{RESET}}$ pin input, reset signal (WDTRES) generation by the watchdog timer, reset signal (LVIRES) generation by the low-voltage detector (LVI), or reset signal (POCRES) generation by the power-on-clear circuit (POC)

Cautions 1. The clock generated by the oscillator continues operating during a reset.
After the reset signal is released, a specific wait time (fixed oscillation stabilization time) elapses.

2. To avoid malfunction due to noise, do not change the division ratio of the CPU operation clock (f_{CPU}) by using the PCC register before setting the PLL mode. Before changing the division ratio, be sure to select the PLL mode.

(3) When releasing STOP mode by interrupt request



<1> When the STOP mode is set, both the oscillator and PLL stop.
At this time, PLL is stopped in the STOP mode. The OST counter is initialized.

<2> When the STOP mode is released, the oscillator is activated and the OST counter starts counting the oscillation stabilization time. At this time, PLL remains stopped.

<3> When half the oscillation stabilization time set to the OSTS has elapsed, PLL starts operating. The clock generated by the oscillator must be stabilized before PLL starts operating. The actual oscillation stabilization time is “1/2 the oscillation stabilization time”. Take this into consideration when setting a value to the OSTS register.

<4> After half the oscillation stabilization time has elapsed, the lockup wait time starts. The remaining count time of the OST counter is the lockup wait time.

<5> When the lockup time of PLL is over, clock supply to the internal circuitry is started. At this time, the status before the STOP mode was set is recovered.

<6> The operation to be performed when the STOP mode is released by a reset signal ($\overline{\text{RESET}}$ pin input, reset signal (LVIRES) generation by the low-voltage detector (LVI), reset signal (POCRES) generation by the power-on-clear circuit (POC)) is the same as that in **(1) Power on (power-on reset)** and **(2) Reset input with power on**.

5.6 Clock Monitor

(1) Clock monitor function

The clock monitor samples the clock generated by the oscillator, by using the internal oscillation clock. When it detects stop of oscillation, output of the timer for motor control goes into a high-impedance state (for details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**). The high-impedance state created by the clock monitor function is released by a reset signal ($\overline{\text{RESET}}$ pin input, reset signal (POCRES) generation by the power-on-clear circuit (POC)) and the pin enters the status after reset.

CHAPTER 6 16-BIT TIMER/EVENT COUNTER AA (TAA)

Timer AA (TAA) is a 16-bit timer/event counter.
The V850E/IF3 and V850E/IG3 incorporate TAA0 to TAA4.

6.1 Overview

The TAAn channels are outlined below (n = 0 to 4).

Table 6-1. TAAn Overview

Item	TAA0	TAA1	TAA2	TAA3	TAA4
Clock selection	8 ways	8 ways	8 ways	8 ways	8 ways
Capture trigger input pin	None	None	2	**Note 1**	2
External event count input pin	None	None	1	**Note 2**	1
External trigger input pin	None	None	1	**Note 2**	1
Timer counter	1	1	1	1	1
Capture/compare register	2 Note 3	2 Note 3	2	2 Note 4	2
Capture/compare match interrupt request signal	2 Note 3	2 Note 3	2	2 Note 4	2
Overflow interrupt request signal	1	1	1	1	1
Timer output pin	None	None	2	**Note 1**	2

Notes 1. V850E/IF3: None
V850E/IG3: 2

2. V850E/IF3: None
V850E/IG3: 1

3. Compare function only

4. In the V850E/IF3, compare function only

6.2 Functions

The functions of TAAn that can be realized differ from one channel to another, as shown in the table below (n = 0 to 4).

Table 6-2. TAAn Functions

Function	TAA0	TAA1	TAA2	TAA3	TAA4
Interval timer	√	√	√	√	√
External event counter	×	×	√	**Note 1**	√
External trigger pulse output	×	×	√	**Note 1**	√
One-shot pulse output	×	×	√	**Note 1**	√
PWM output	×	×	√	**Note 1**	√
Free-running timer	√Note 2	√Note 2	√	√Note 3	√
Pulse width measurement	×	×	√	**Note 1**	√
Timer tuning operation	√ (TAB0)	√ (TAB1)	×	×	×

Notes 1. V850E/IF3: ×
V850E/IG3: √

2. Compare function only

3. In the V850E/IF3, compare function only

6.3 Configuration

TAAn includes the following hardware (n = 0 to 4).

Table 6-3. Configuration of TAAn

Item	Configuration
Timer register	16-bit counter × 1
Registers	TAAn capture/compare registers 0, 1 (TAAnCCR0, TAAnCCR1) TAAn counter read buffer register (TAAnCNT) CCR0 and CCR1 buffer registers
Timer input	6 in total (TIA20, TIA21, TIA30[Note 1], TIA31[Note 1], TIA40, TIA41 pins)[Notes 2, 3]
Timer output	6 in total (TOA20, TOA21, TOA30[Note 1], TOA31[Note 1], TOA40, TOA41 pins)[Notes 2, 3]
Control registers	TAAn control registers 0, 1 (TAAnCTL0, TAAnCTL1) TAAm I/O control registers 0 to 2 (TAAmIOC0 to TAAmIOC2) TAAn option registers 0, 1 (TAAnOPT0, TAAnOPT1)

Notes 1. V850E/IG3 only

2. Not provided for TMP0 and TMP1

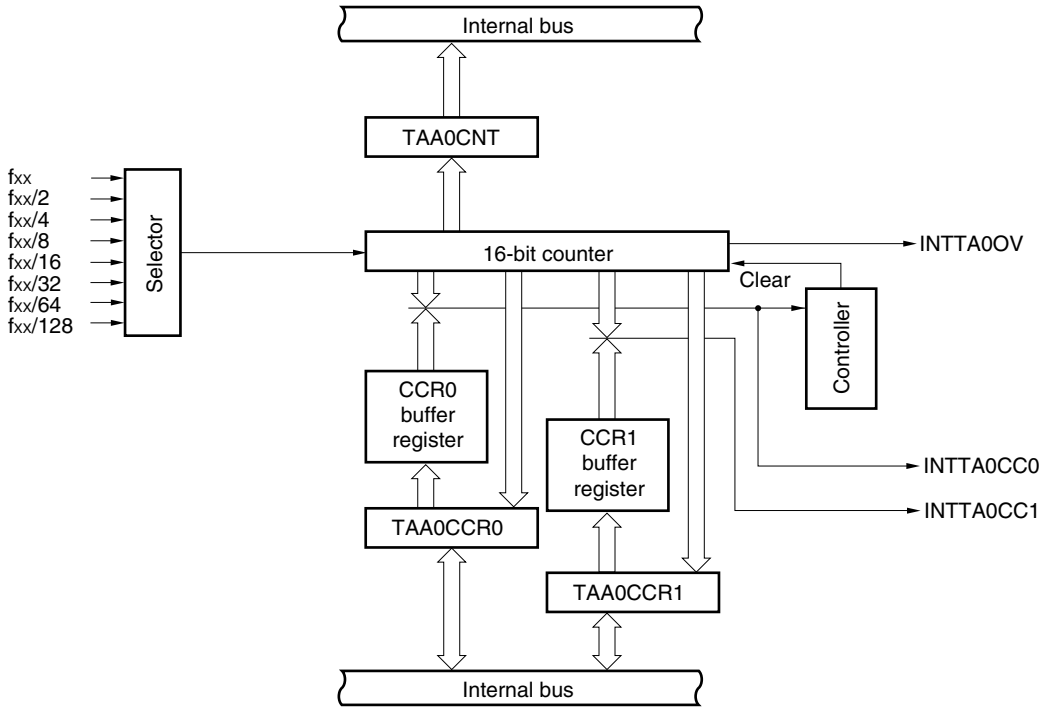
3. TIA20, TIA30, and TIA40 pins function as capture trigger input signal, external event count input signal, and external trigger input signal, and function alternately (alternate-function) as timer output pins (TOA20, TOA30, TOA40).

TIA21, TIA31, and TIA41 pins function alternately as capture trigger input signal and timer output pins (TOA21, TOA31, TOA41).

Remark V850E/IF3: n = 0 to 4, m = 2, 4

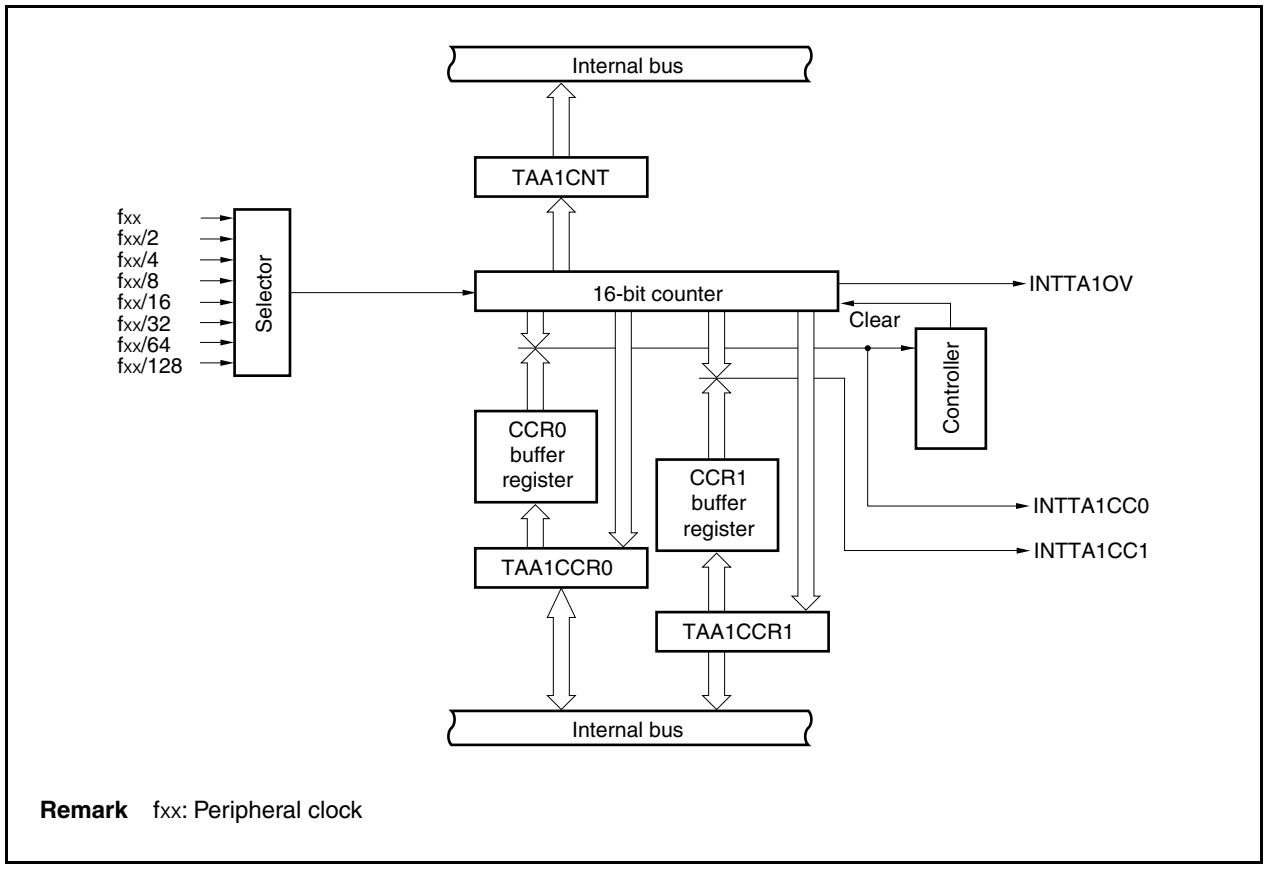
V850E/IG3: n = 0 to 4, m = 2 to 4

Figure 6-1. TAA0 Block Diagram



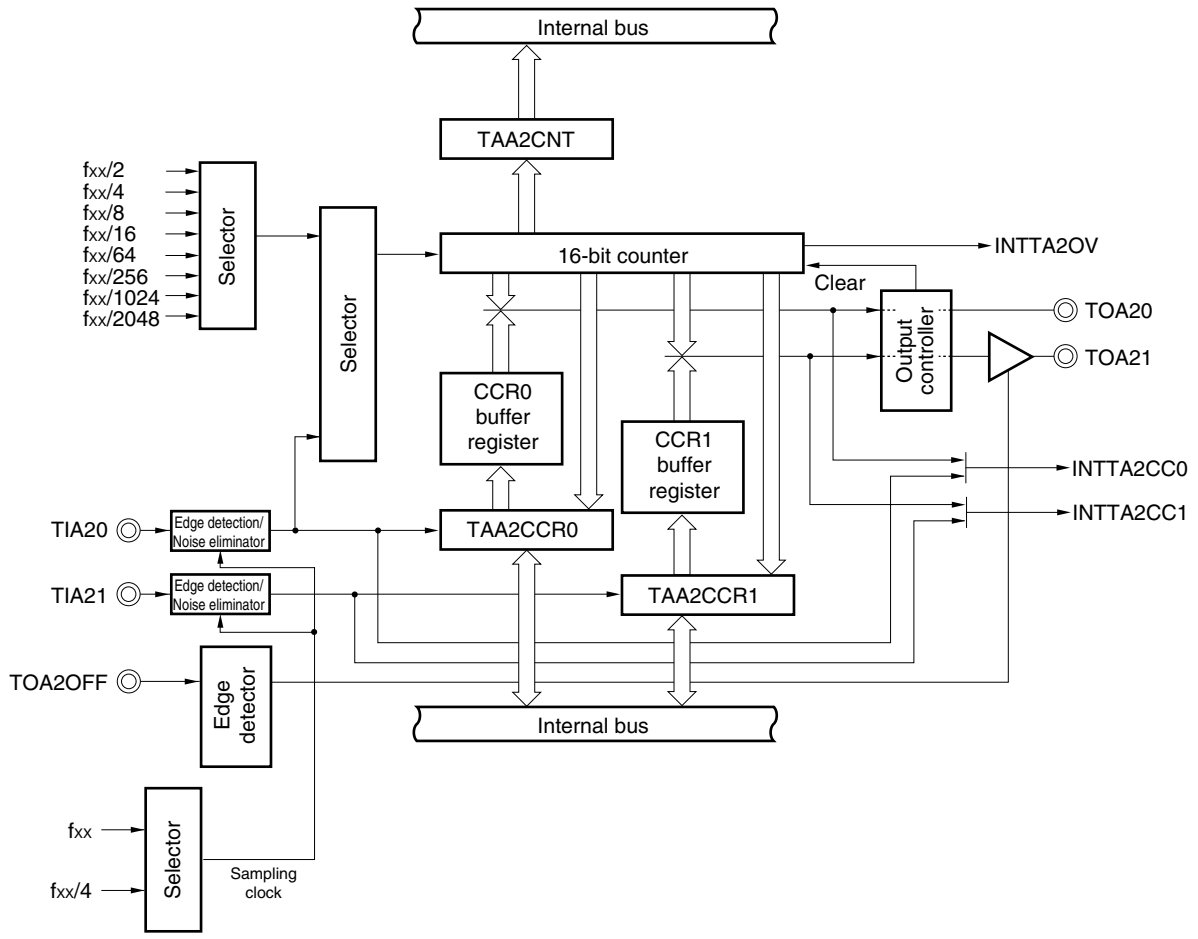
Remark fxx: Peripheral clock

Figure 6-2. TAA1 Block Diagram



Remark fxx: Peripheral clock

Figure 6-3. TAA2 Block Diagram

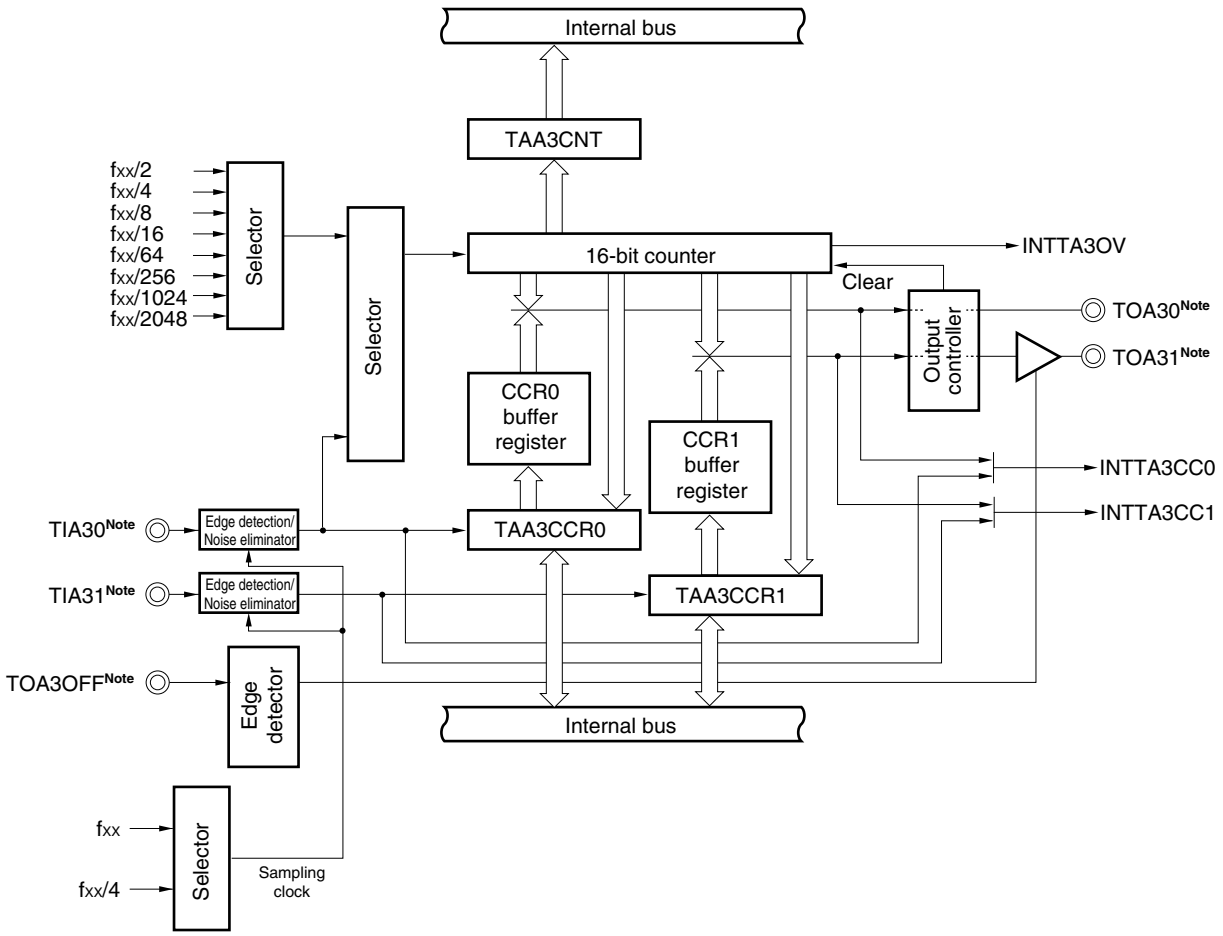


Remarks 1. fxx: Peripheral clock

2. For the TOA3OFF pin, see **10.3 (6) High-impedance output control registers 00, 01, 10, 11, 20, 21, 30, 31 (HZAyCTL0, HZAyCTL1)**.

3. For the noise eliminator, see **4.6 Noise Eliminator**.

Figure 6-4. TAA3 Block Diagram



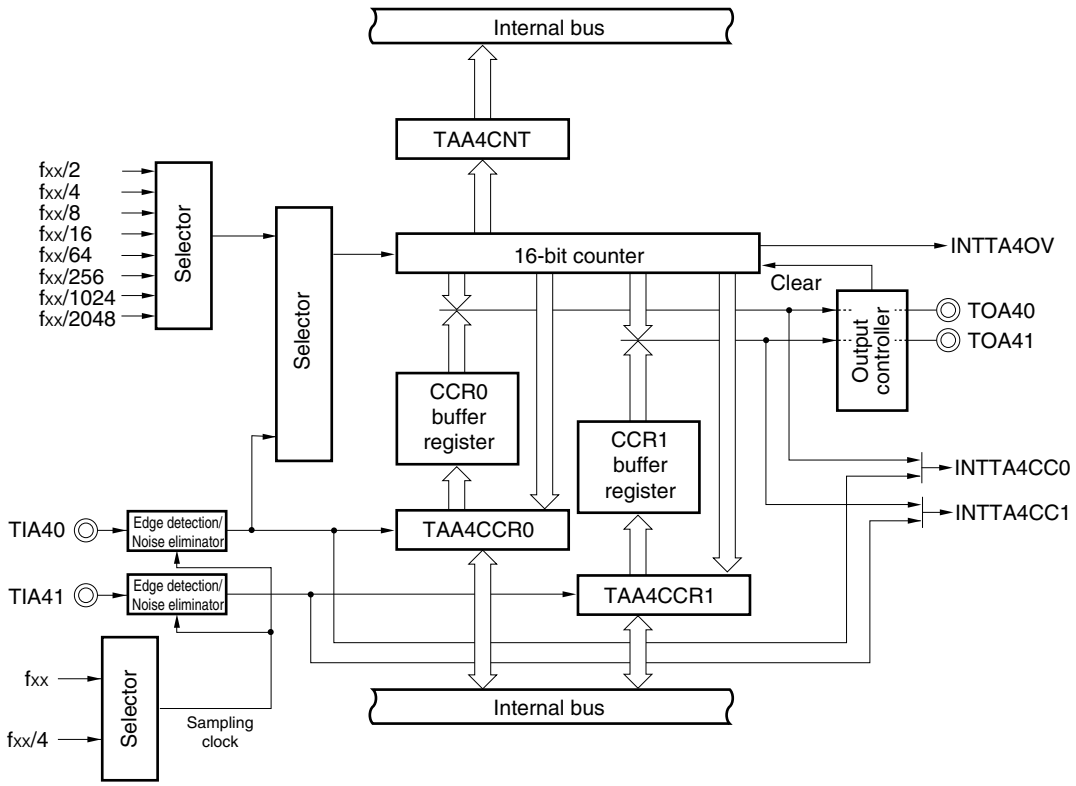
Note V850E/IG3 only

Remarks 1. fxx: Peripheral clock

2. For the TOA3OFF pin, see **10.3 (6) High-impedance output control registers 00, 01, 10, 11, 20, 21, 30, 31 (HZAyCTL0, HZAyCTL1)**.

3. For the noise eliminator, see **4.6 Noise Eliminator**.

Figure 6-5. TAA4 Block Diagram



Remarks 1. fxx: Peripheral clock

2. For the noise eliminator, see **4.6 Noise Eliminator**.

(1) 16-bit counter

This 16-bit counter can count internal clocks or external events.

The count value of this counter can be read by using the TAAnCNT register.

When the TAAnCTL0.TAAnCE bit = 0, the value of the 16-bit counter is FFFFH. If the TAAnCNT register is read at this time, 0000H is read.

Reset sets the TAAnCE bit to 0.

(2) CCR0 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TAAnCCR0 register is used as a compare register, the value written to the TAAnCCR0 register is transferred to the CCR0 buffer register. When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTAnCC0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is cleared to 0000H after reset, and the TAAnCCR0 register is cleared to 0000H.

(3) CCR1 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TAAnCCR1 register is used as a compare register, the value written to the TAAnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTAnCC1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is cleared to 0000H after reset, and the TAAnCCR1 register is cleared to 0000H.

(4) Edge detector

This circuit detects the valid edges input to the TIA20, TIA21, TIA30 (V850E/IG3 only), TIA31 (V850E/IG3 only), TIA40, and TIA41 pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TAAmIOC1 and TAAmIOC2 registers.

(5) Output controller

This circuit controls the output of the TOA20, TOA21, TOA30 (V850E/IG3 only), TOA31 (V850E/IG3 only), TOA40, and TOA41 pins. The output controller is controlled by the TAAmIOC0 registers.

(6) Selector

This selector selects the count clock for the 16-bit counter. Eight types of internal clocks or an external event can be selected as the count clock.

6.4 Registers

(1) TAAn control register 0 (TAAnCTL0)

The TAAnCTL0 register is an 8-bit register that controls the operation of TAAn.

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

Reset sets this register to 00H.

The same value can always be written to the TAAnCTL0 register by software.

After reset: 00H R/W Address: TAA0CTL0 FFFFF660H, TAA1CTL0 FFFFF680H, TAA2CTL0 FFFFF6A0H, TAA3CTL0 FFFFFB00H, TAA4CTL0 FFFFFB20H

	<7>	6	5	4	3	2	1	0
TAAnCTL0 (n = 0 to 4)	TAAnCE	0	0	0	0	TAAnCKS2	TAAnCKS1	TAAnCKS0

TAAnCE	TAAn operation control
0	TAAn operation disabled (TAAn reset asynchronously[Note])
1	TAAn operation enabled. TAAn operation start

TAAnCKS2	TAAnCKS1	TAAnCKS0	Internal count clock selection	
			TAA0, TAA1	TAA2 to TAA4
0	0	0	f_{XX}	$f_{XX}/2$
0	0	1	$f_{XX}/2$	$f_{XX}/4$
0	1	0	$f_{XX}/4$	$f_{XX}/8$
0	1	1	$f_{XX}/8$	$f_{XX}/16$
1	0	0	$f_{XX}/16$	$f_{XX}/64$
1	0	1	$f_{XX}/32$	$f_{XX}/256$
1	1	0	$f_{XX}/64$	$f_{XX}/1024$
1	1	1	$f_{XX}/128$	$f_{XX}/2048$

Note TAAnOPT0.TAAnOVF bit and 16-bit counter are reset simultaneously. Moreover, timer outputs (TOA20, TOA21, TOA30 (V850E/IG3 only), TOA31 (V850E/IG3 only), TOA40, TOA41 pins) are reset to the TAAmIOC0 register set status at the same time as the 16-bit counter (V850E/IF3: m = 2, 4, V850E/IG3: m = 2 to 4).

Cautions 1. Set the TAAnCKS2 to TAAnCKS0 bits when the TAAnCE bit = 0.
When the value of the TAAnCE bit is changed from 0 to 1, the TAAnCKS2 to TAAnCKS0 bits can be set simultaneously.
2. Be sure to set bits 3 to 6 to "0".

Remark f_{XX}: Peripheral clock

(2) TAAn control register 1 (TAAnCTL1)

The TAAnCTL1 register is an 8-bit register that controls the TAAn operation.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: TAA0CTL1 FFFFF661H, TAA1CTL1 FFFFF681H, TAA2CTL1 FFFFF6A1H, TAA3CTL1 FFFFFB01H, TAA4CTL1 FFFFFB21H

	7	6	5	4	3	2	1	0
TAAnCTL1	TAAaSYE Note 1	TAAmEST Note 2	TAAmEEE Note 2	0	0	TAAnMD2	TAAnMD1	TAAnMD0

V850E/IF3
n = 0 to 4
m = 2, 4
a = 0, 1

TAAaSYE Note 1	Operation mode selection
0	TAAa single mode
1	Tuning operation mode (see **10.4.5**)

V850E/IG3
n = 0 to 4
m = 2 to 4
a = 0, 1

TAAa can be used only as an A/D conversion start trigger factor of A/D converters 0 and 1 during the tuning operation. In the tuning operation mode, this bit always operates in synchronization with TABa.

TAAmEST Note 2	Software trigger control
0	–
1	Generates a valid signal for external trigger input. • In one-shot pulse output mode: A one-shot pulse is output with writing 1 to the TAAmEST bit as the trigger. • In external trigger pulse output mode: A PWM waveform is output with writing 1 to the TAAmEST bit as the trigger.

The read value of the TAAmEST bit is always 0.

Notes 1. This bit can be set only in TAA0 and TAA1. Be sure to set bit 7 of TAA2 to TAA4 to “0”.
For details of tuning operation mode, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

2. In the V850E/IF3, this bit can be set only in TAA2 and TAA4. Be sure to set bits 5 and 6 of TAA0, TAA1, and TAA3 to “0”.
In the V850E/IG3, this bit can be set only in TAA2 to TAA4. Be sure to set bits 5 and 6 of TAA0 and TAA1 to “0”.

(2/2)

TAAmEEE[Note 1]	Count clock selection
0	Disable operation with external event count input (TIAm0 pin). (Perform counting with the count clock selected by the TAAmCTL0.TAAmCKS0 to TAAmCTL0.TAAmCKS2 bits.)
1	Enable operation[Note 2] with external event count input (TIAm0 pin). (Perform counting at every valid edge of the external event count input signal (TIAm0 pin).)
The TAAmEEE bit selects whether counting is performed with the internal count clock or the valid edge of the external event count input.	

TAAnMD2	TAAnMD1	TAAnMD0	Timer mode selection
0	0	0	Interval timer mode
0	0	1	External event count mode
0	1	0	External trigger pulse mode
0	1	1	One-shot pulse output mode
1	0	0	PWM output mode
1	0	1	Free-running timer mode
1	1	0	Pulse width measurement mode
1	1	1	Setting prohibited

Notes 1. With the V850E/IF3, this bit can be set only in TAA2 and TAA4. Be sure to set bits 5 and 6 of TAA0, TAA1, and TAA3 to “0”.
With the V850E/IG3, this bit can be set only in TAA2 to TAA4. Be sure to set bits 5 and 6 of TAA0 and TAA1 to “0”.

2. Set the valid edge selection of capture trigger input (TIAm0 pin) and external trigger input (TIAm0 pin) to “No edge detection”.

Cautions 1. The TAAmEST bit is valid only in the external trigger pulse output mode or one-shot pulse output mode. In any other mode, writing 1 to this bit is ignored.

2. External event count input is selected in the external event count mode regardless of the value of the TAAmEEE bit.

3. Set the TAAaSYE, TAAmEEE, and TAAnMD2 to TAAnMD0 bits when the TAAnCTL0.TAAnCE bit = 0. (The same value can be written when the TAAnCE bit = 1.) The operation is not guaranteed when rewriting is performed with the TAAnCE bit = 1. If rewriting was mistakenly performed, clear the TAAnCE bit to 0 and then set the bits again.

4. Be sure to set bits 3 and 4 to “0”.

(3) TAAm I/O control register 0 (TAAmIOC0)

The TAAmIOC0 register is an 8-bit register that controls the timer output (TOAm0, TOAm1 pins).

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: TAA2IOC0 FFFFF6A2H, TAA3IOC0 FFFFFB02H[Note 1]
TAA4IOC0 FFFFFB22H

	7	6	5	4	3	<2>	1	<0>
TAAmIOC0	0	0	0	0	TAAmOL1	TAAmOE1	TAAmOL0	TAAmOE0

V850E/IF3 m = 2, 4

V850E/IG3 m = 2 to 4

TAAmOL1	TOAm1 pin output level setting[Note 2]
0	TOAm1 pin starts output at high level.
1	TOAm1 pin starts output at low level.

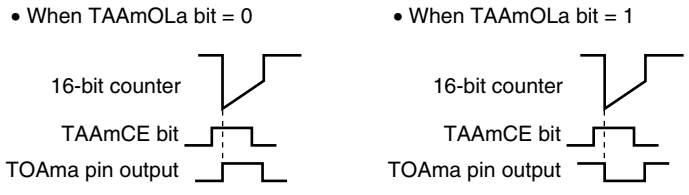
TAAmOE1	TOAm1 pin output setting
0	Timer output prohibited • Low level is output from the TOAm1 pin when the TAAmOL1 bit = 0. • High level is output from the TOAm1 pin when the TAAmOL1 bit = 1.
1	Timer output enabled (A pulse is output from the TOAm1 pin.)

TAAmOL0	TOAm0 pin output level setting[Note 2]
0	TOAm0 pin starts output at high level.
1	TOAm0 pin starts output at low level.

TAAmOE0	TOAm0 pin output setting
0	Timer output prohibited • Low level is output from the TOAm0 pin when the TAAmOL0 bit = 0. • High level is output from the TOAm0 pin when the TAAmOL0 bit = 1.
1	Timer output enabled (A pulse is output from the TOAm0 pin.)

Notes 1. V850E/IG3 only

2. The output level of the timer output pins (TOAm0 and TOAm1) specified by the TAAmOLa bit is shown below (a = 0, 1).



(2/2)

Cautions 1. If the setting of the TAAmIOC0 register is changed when TOAm0 and TOAm1 are set in the output mode, the output of the pins change. Set the port in the input mode and make the port go into a high-impedance state, noting changes in the pin status.

2. Rewrite the TAAmOL1, TAAmOE1, TAAmOL0, and TAAmOE0 bits when the TAAmCTL0.TAAmCE bit = 0. (The same value can be written when the TAAmCE bit = 1.) If rewriting was mistakenly performed, clear the TAAmCE bit to 0 and then set the bits again.

3. Even if the TAAmOL0 or TAAmOL1 bit is manipulated when the TAAmCE, TAAmOE0, and TAAmOE1 bits are 0, the output level of the TOAm0 and TOAm1 pins changes.

(4) TAAm I/O control register 1 (TAAmIOC1)

The TAAmIOC1 register is an 8-bit register that controls the valid edge for the capture trigger input signals (TIAm0, TIAm1 pins).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TAA2IOC1 FFFFF6A3H, TAA3IOC1 FFFFFB03H[Note], TAA4IOC1 FFFFFB23H

	7	6	5	4	3	2	1	0
TAAmIOC1	0	0	0	0	TAAmIS3	TAAmIS2	TAAmIS1	TAAmIS0

(V850E/IF3 m = 2, 4)

(V850E/IG3 m = 2 to 4)

TAAmIS3	TAAmIS2	Capture trigger input signal (TIAm1 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TAAmIS1	TAAmIS0	Capture trigger input signal (TIAm0 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Note V850E/IG3 only

Cautions 1. Rewrite the TAAmIS3 to TAAmIS0 bits when the TAAmCTL0.TAAmCE bit = 0. (The same value can be written when the TAAmCE bit = 1.) If rewriting was mistakenly performed, clear the TAAmCE bit to 0 and then set the bits again.

2. The TAAmIS3 to TAAmIS0 bits are valid only in the free-running timer mode (only when the TAAmOPT0.TAAmCCS1 and TAAmOPT0.TAAmCCS0 bits = 11) and the pulse width measurement mode. In all other modes, a capture operation is not possible.

(5) TAAm I/O control register 2 (TAAmIOC2)

The TAAmIOC2 register is an 8-bit register that controls the valid edge for the external event count input signal (TIAm0 pin) and external trigger input signal (TIAm0 pin).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TAA2IOC2 FFFFF6A4H, TAA3IOC2 FFFFFB04H[Note], TAA4IOC2 FFFFFB24H

	7	6	5	4	3	2	1	0
TAAmIOC2	0	0	0	0	TAAmEES1	TAAmEES0	TAAmETS1	TAAmETS0

(V850E/IF3 m = 2, 4)

(V850E/IG3 m = 2 to 4)

TAAmEES1	TAAmEES0	External event count input signal (TIAm0 pin) valid edge setting
0	0	No edge detection (external event count invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TAAmETS1	TAAmETS0	External trigger input signal (TIAm0 pin) valid edge setting
0	0	No edge detection (external trigger invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Note V850E/IG3 only

Cautions 1. Rewrite the TAAmEES1, TAAmEES0, TAAmETS1, and TAAmETS0 bits when the TAAmCTL0.TAAmCE bit = 0. (The same value can be written when the TAAmCE bit = 1.) If rewriting was mistakenly performed, clear the TAAmCE bit to 0 and then set the bits again.

2. The TAAmEES1 and TAAmEES0 bits are valid only when the TAAmCTL1.TAAmEEE bit = 1 or when the external event count mode (the TAAmCTL1.TAAmMD2 to TAAmCTL1.TAAmMD0 bits = 001) has been set.

3. The TAAmETS1 and TAAmETS0 bits are valid only in the external trigger pulse mode or one-shot pulse output mode.

(6) TAAn option register 0 (TAAnOPT0)

The TAAnOPT0 register is an 8-bit register that sets the capture/compare operation and detects overflow.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TAA0OPT0 FFFFF665H, TAA1OPT0 FFFFF685H, TAA2OPT0 FFFFF6A5H, TAA3OPT0 FFFFFB05H, TAA4OPT0 FFFFFB25H

	7	6	5	4	3	2	1	<0>
TAAnOPT0	0	0	TAAmCCS1Note	TAAmCCS0Note	0	0	0	TAAnOVF

V850E/IF3
n = 0 to 4
m = 2, 4

V850E/IG3
n = 0 to 4
m = 2 to 4

TAAmCCS1Note	TAAmCCR1 register capture/compare selection
0	Compare register selected
1	Capture register selected (cleared by TAAmCTL0.TAAmCE bit = 0)
The TAAmCCS1 bit setting is valid only in the free-running timer mode.	

TAAmCCS0Note	TAAmCCR0 register capture/compare selection
0	Compare register selected
1	Capture register selected (cleared by TAAmCTL0.TAAmCE bit = 0)
The TAAmCCS0 bit setting is valid only in the free-running timer mode.	

TAAnOVF	TAAn overflow detection flag
Set (1)	Overflow occurred
Reset (0)	0 is written to TAAnOVF bit or TAAnCTL0.TAAnCE bit = 0

- The TAAnOVF bit is set to 1 when the 16-bit counter value overflows from FFFFH to 0000H in the free-running timer mode or the pulse width measurement mode.
- An overflow interrupt request signal (INTTAnOV) is generated at the same time that the TAAnOVF bit is set to 1. The INTTAnOV signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode.
- The TAAnOVF bit is not cleared to 0 even when the TAAnOVF bit or the TAAnOPT0 register are read when the TAAnOVF bit = 1.
- Before clearing the TAAnOVF bit to 0 after generation of the INTTAnOV signal, be sure to confirm (by reading) that the TAAnOVF bit is set to 1.
- The TAAnOVF bit can be both read and written, but the TAAnOVF bit cannot be set to 1 by software. Writing 1 has no effect on the operation of TAAn.

Note With the V850E/IF3, this bit can be set only in TAA2 and TAA4. Be sure to set bits 4 and 5 of TAA0, TAA1, and TAA3 to “0”.

With the V850E/IG3, this bit can be set only in TAA2 to TAA4. Be sure to set bits 4 and 5 of TAA0 and TAA1 to “0”.

Cautions 1. Rewrite the TAAmCCS1 and TAAmCCS0 bits when the TAAmCE bit = 0. (The same value can be written when the TAAmCE bit = 1.) If rewriting was mistakenly performed, clear the TAAmCE bit to 0 and then set the bits again.

2. Be sure to set bits 1 to 3, 6, and 7 to “0”.

(7) TAAn capture/compare register 0 (TAAnCCR0)

The TAAmCCR0 register is a 16-bit register that can be used as a capture register or compare register depending on the mode. The TAAkCCR0 register is a 16-bit register that can only be used as a compare register.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TAAmOPT0.TAAmCCS0 bit. In the pulse width measurement mode, the TAAmCCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TAAnCCR0 register can be read or written during operation.

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

Reset sets this register to 0000H.

Remark V850E/IF3: n = 0 to 4, m = 2, 4, k = 0, 1, 3
V850E/IG3: n = 0 to 4, m = 2 to 4, k = 0, 1

After reset: 0000H R/W Address: TAA0CCR0 FFFFF666H, TAA1CCR0 FFFFF686H, TAA2CCR0 FFFFF6A6H, TAA3CCR0 FFFFFB06H, TAA4CCR0 FFFFFB26H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAAnCCR0 (n = 0 to 4)																

(a) Function as compare register

The TAAnCCR0 register can be rewritten even when the TAAnCTL0.TAAnCE bit = 1.

The set value of the TAAnCCR0 register is transferred to the CCR0 buffer register. When the value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTAnCC0) is generated. If TOAm0 pin output is enabled at this time, the output of the TOAm0 pin is inverted.

When the TAAnCCR0 register is used as a cycle register in the interval timer mode or the TAAmCCR0 register is used as a cycle register in external event count mode, external trigger pulse output mode, one-shot pulse output mode, or PWM output mode, the value of the 16-bit counter is cleared (0000H) if its count value matches the value of the CCR0 buffer register.

The compare register is not cleared by setting the TAAnCTL0.TAAnCE bit to 0.

(b) Function as capture register

When the TAAmCCR0 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TAAmCCR0 register if the valid edge of the capture trigger input pin (TIAm0 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TAAmCCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIAm0 pin) is detected.

Even if the capture operation and reading the TAAmCCR0 register conflict, the correct value of the TAAmCCR0 register can be read.

The capture register is cleared by setting the TAAmCTL0.TAAmCE bit to 0.

Remark V850E/IF3: n = 0 to 4, m = 2, 4
V850E/IG3: n = 0 to 4, m = 2 to 4

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 6-4. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter[Note 1]	Compare register	Anytime write
External trigger pulse output[Note 1]	Compare register	Batch write[Note 2]
One-shot pulse output[Note 1]	Compare register	Anytime write
PWM output[Note 1]	Compare register	Batch write[Note 2]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None

Notes 1. With the V850E/IF3, this mode is only for TAA2 and TAA4. With the V850E/IG3, this mode is only for TAA2 to TAA4.

2. Writing to the TAAmCCR1 register is the trigger.

Remark For anytime write and batch write, see **6.6 (2) Anytime write and batch write**.

(8) TAAn capture/compare register 1 (TAAnCCR1)

The TAAmCCR1 register is a 16-bit register that can be used as a capture register or compare register depending on the mode. The TAAkCCR1 register is a 16-bit register that can only be used as a compare register.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TAAmOPT0.TAAmCCS1 bit. In the pulse width measurement mode, the TAAmCCR1 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

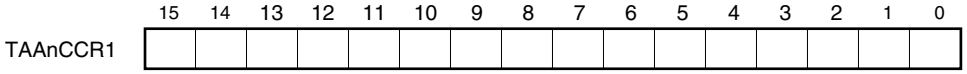
The TAAnCCR1 register can be read or written during operation.

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

Reset sets this register to 0000H.

Remark V850E/IF3: n = 0 to 4, m = 2, 4, k = 0, 1, 3
V850E/IG3: n = 0 to 4, m = 2 to 4, k = 0, 1

After reset: 0000H R/W Address: TAA0CCR1 FFFFF668H, TAA1CCR1 FFFFF688H, TAA2CCR1 FFFFF6A8H, TAA3CCR1 FFFFFB08H, TAA4CCR1 FFFFFB28H



TAAnCCR1
(n = 0 to 4)

(a) Function as compare register

The TAAnCCR1 register can be rewritten even when the TAAnCTL0.TAAnCE bit = 1.

The set value of the TAAnCCR1 register is transferred to the CCR1 buffer register. When the value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTAnCC1) is generated. If TOAm1 pin output is enabled at this time, the output of the TOAm1 pin is inverted.

The compare register is not cleared by setting the TAAnCTL0.TAAnCE bit to 0.

(b) Function as capture register

When the TAAmCCR1 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TAAmCCR1 register if the valid edge of the capture trigger input pin (TIAm1 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TAAmCCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIAm1 pin) is detected.

Even if the capture operation and reading the TAAmCCR1 register conflict, the correct value of the TAAmCCR1 register can be read.

The capture register is cleared by setting the TAAmCTL0.TAAmCE bit to 0.

Remark V850E/IF3: n = 0 to 4, m = 2, 4
V850E/IG3: n = 0 to 4, m = 2 to 4

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 6-5. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter[Note 1]	Compare register	Anytime write
External trigger pulse output[Note 1]	Compare register	Batch write[Note 2]
One-shot pulse output[Note 1]	Compare register	Anytime write
PWM output[Note 1]	Compare register	Batch write[Note 2]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None

Notes 1. In the V850E/IF3, this mode is only for TAA2 and TAA4. In the V850E/IG3, this mode is only for TAA2 to TAA4.

2. Writing to the TAAmCCR1 register is the trigger.

Remark For anytime write and batch write, see **6.6 (2) Anytime write and batch write**.

(9) TAAn counter read buffer register (TAAnCNT)

The TAAnCNT register is a read buffer register that can read the count value of the 16-bit counter.

If this register is read when the TAAnCTL0.TAAnCE bit = 1, the count value of the 16-bit timer can be read.

This register is read-only, in 16-bit units.

The value of the TAAnCNT register is cleared to 0000H when the TAAnCE bit = 0. If the TAAnCNT register is read at this time, the value of the 16-bit counter (FFFFH) is not read, but 0000H is read.

The value of the TAAnCNT register is cleared to 0000H after reset, and the TAAnCE bit is cleared to 0.

After reset: 0000H R Address: TAA0CNT FFFFF66AH, TAA1CNT FFFFF68AH, TAA2CNT FFFFF6AAH, TAA3CNT FFFFFB0AH, TAA4CNT FFFFFB2AH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAAnCNT																

(n = 0 to 4)

6.5 Timer Output Operations

The following table shows the operations and output levels of the TOAm0 and TOAm1 pins.

Table 6-6. Timer Output Control in Each Mode

Operation Mode	TOAm1 Pin	TOAm0 Pin
Interval timer mode	PWM output	
External event count mode	None	
External trigger pulse output mode	External trigger pulse output	PWM output
One-shot pulse output mode	One-shot pulse output	
PWM output mode	PWM output	
Free-running timer mode	PWM output (only when compare function is used)	
Pulse width measurement mode	None	

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

Table 6-7. Truth Table of TOAm0 and TOAm1 Pins Under Control of Timer Output Control Bits

TAAmIOC0.TAAmOLa Bit	TAAmIOC0.TAAmOEa Bit	TAAmCTL0.TAAmCE Bit	Level of TOAma Pin
0	0	×	Low-level output
	1	0	Low-level output
		1	Low level immediately before counting, high level after counting is started
1	0	×	High-level output
	1	0	High-level output
		1	High level immediately before counting, low level after counting is started

Remark V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

6.6 Operation

The functions of TAAn that can be achieved differ from one channel to another. The functions of each channel are shown below.

Table 6-8. TAA0 and TAA1 Specifications in Each Mode

Operation	Software Trigger Bit	External Trigger Input	Capture/Compare Register Setting	Compare Register Write Method
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode	None			
External trigger pulse output mode	None			
One-shot pulse output mode	None			
PWM output mode	None			
Free-running timer mode	Invalid	Invalid	Compare only	Anytime write
Pulse width measurement mode	None			

Remarks 1. TAAa does not have timer input pins (TIAa0, TIAa1) and timer output pins (TOAa0, TOAa1). It has interrupt request signals (INTTAaCC0, INTTAaCC1) on a match between the value of the 16-bit counter and the values of the TAAaCCR0 and TAAaCCR1 registers.

2. TAAa has a function to execute tuning with TABa. For details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

3. a = 0, 1

Table 6-9. TAA2 to TAA4 Specifications in Each Mode

Operation	TAAmCTL1.TAAmEST Bit (Software Trigger Bit)	TIAm0 Pin (External Trigger Input)	Capture/Compare Register Setting	Compare Register Write Method
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode[Note 1]	Invalid	Invalid	Compare only	Anytime write
External trigger pulse output mode[Note 2]	Valid	Valid	Compare only	Batch write
One-shot pulse output mode[Note 2]	Valid	Valid	Compare only	Anytime write
PWM output mode	Invalid	Invalid	Compare only	Batch write
Free-running timer mode	Invalid	Invalid	Switchable[Note 3]	Anytime write
Pulse width measurement mode[Note 2]	Invalid	Invalid	Capture only	Not applicable

Notes 1. When using the external event count mode, set the TIAm0 pin capture trigger input valid edge selection to “No edge detection”. (Clear the TAAmIOC1.TAAmIS1 and TAAmIOC1.TAAmIS0 bits to 00.)

2. When using the external trigger pulse output mode, one-shot pulse output mode, and pulse width measurement mode, select the internal clock as the count clock (by clearing the TAAmCTL1.TAAmEEE bit to 0).

3. In the V850E/IF3, this setting is compare only.

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

(1) Counter basic operation

This section explains the basic operation of the 16-bit counter. For details, refer to the description of the operation in each mode.

Remark V850E/IF3: n = 0 to 4, m = 2, 4
V850E/IG3: n = 0 to 4, m = 2 to 4

(a) Counter start operation

The 16-bit counter of TAAn starts counting from the default value FFFFH in all modes.
It counts up from FFFFH to 0000H, 0001H, 0002H, 0003H, and so on.

(b) Clear operation

The 16-bit counter is cleared to 0000H when its value matches the value of the compare register and is cleared, and when its value is captured and cleared. The counting operation from FFFFH to 0000H that takes place immediately after the counter has started counting or when the counter overflows is not a clearing operation. Therefore, the INTTAnCC0 and INTTAnCC1 interrupt signals are not generated.

(c) Overflow operation

The 16-bit counter overflows when the counter counts up from FFFFH to 0000H in the free-running mode or pulse width measurement mode. If the counter overflows, the TAAnOPT0.TAAnOVF bit is set to 1 and an interrupt request signal (INTTAnOV) is generated. Note that the INTTAnOV signal is not generated under the following conditions.

- Immediately after a counting operation has been started
- If the counter value matches the compare value FFFFH and is cleared
- When FFFFH is captured and cleared in the pulse width measurement mode and the counter counts up from FFFFH to 0000H

Caution After the overflow interrupt request signal (INTTAnOV) has been generated, be sure to check that the overflow flag (TAAnOVF bit) is set to 1.

(d) Counter read operation during counting operation

The value of the 16-bit counter of TAAn can be read by using the TAAnCNT register during the count operation. When the TAAnCTL0.TAAnCE bit = 1, the value of the 16-bit counter can be read by reading the TAAnCNT register. When the TAAnCTL0.TAAnCE bit = 0, the 16-bit counter is FFFFH and the TAAnCNT register is 0000H.

(e) Interrupt operation

TAAn generates the following three types of interrupt request signals.

- INTTAnCC0 interrupt: This signal functions as a match interrupt request signal of the CCR0 buffer register and as a capture interrupt request signal to the TAAnCCR0 register.
- INTTAnCC1 interrupt: This signal functions as a match interrupt request signal of the CCR1 buffer register and as a capture interrupt request signal to the TAAnCCR1 register.
- INTTAnOV interrupt: This signal functions as an overflow interrupt request signal.

(2) Anytime write and batch write

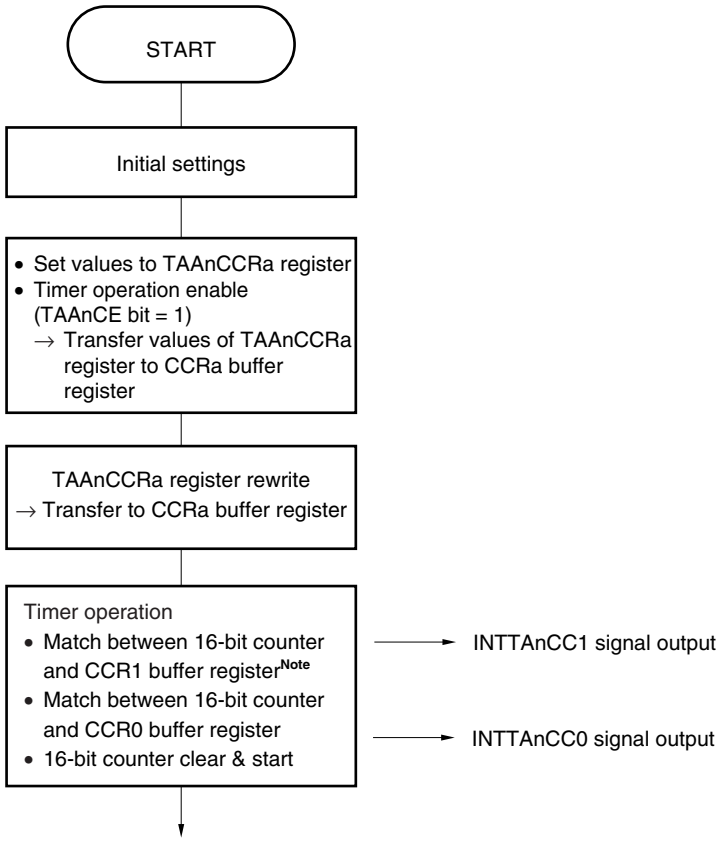
The TAAnCCR0 and TAAnCCR1 registers in TAAn can be rewritten during timer operation (TAAnCTL0.TAAnCE bit = 1), but the write method (anytime write, batch write) of the CCR0 and CCR1 buffer registers differs depending on the mode.

(a) Anytime write

In this mode, data is transferred at any time from the TAAnCCR0 and TAAnCCR1 registers to the CCR0 and CCR1 buffer registers during timer operation.

Remark n = 0 to 4

Figure 6-6. Flowchart of Basic Operation for Anytime Write

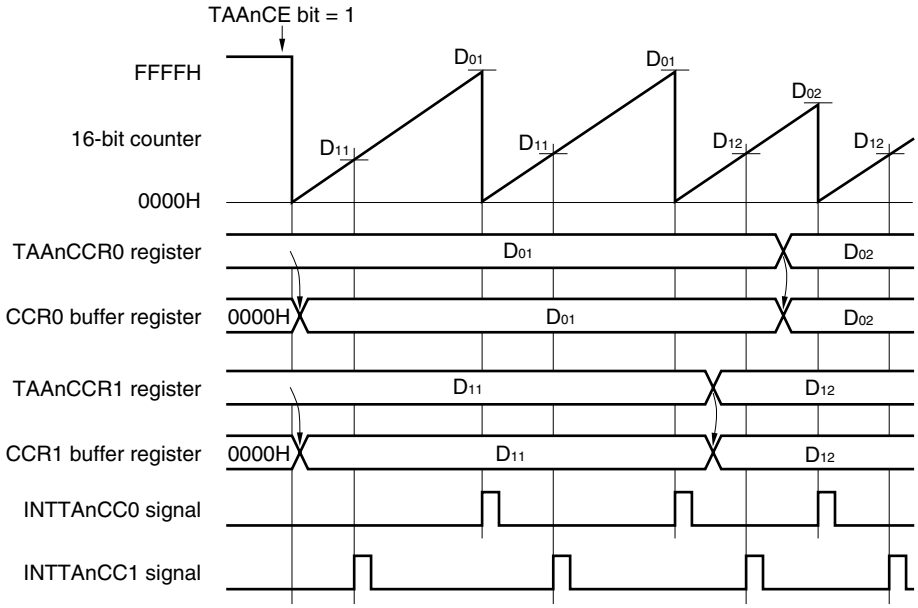


Note The 16-bit counter is not cleared upon a match between the 16-bit counter value and the CCR1 buffer register value. It is cleared upon a match between the 16-bit counter value and the CCR0 buffer register value.

Remarks 1. The above flowchart illustrates an example of the operation in the interval timer mode.

2. n = 0 to 4
a = 0, 1

Figure 6-7. Timing of Anytime Write



Remarks 1. D_{01}, D_{02}: Set values of the TAAnCCR0 register

D_{11}, D_{12}: Set values of the TAAnCCR1 register

2. The above timing chart illustrates an example of the operation in the interval timer mode.

3. n = 0 to 4

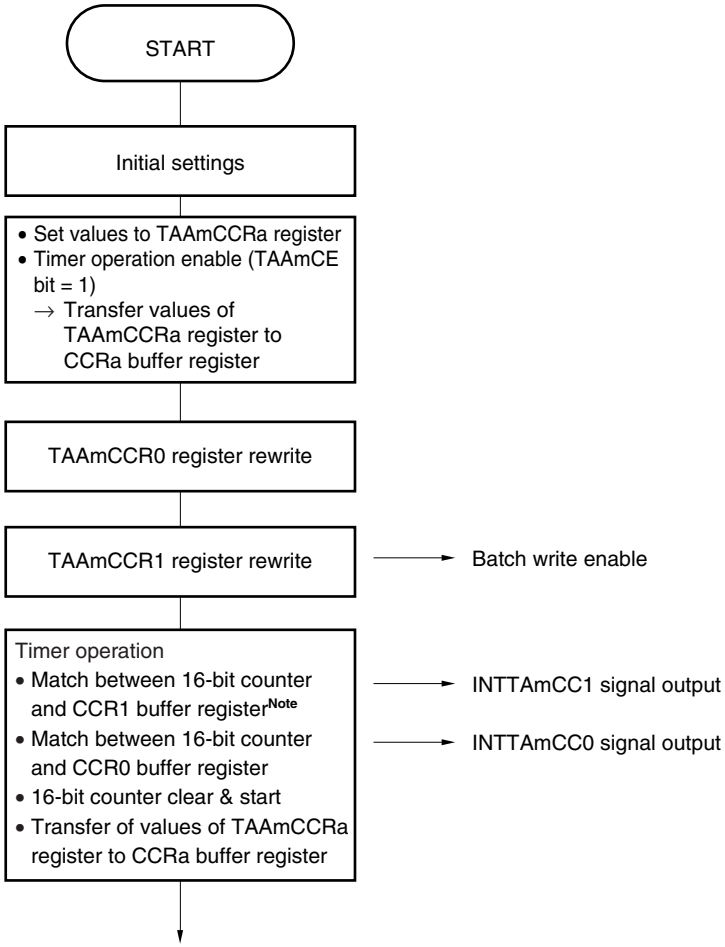
(b) Batch write

In this mode, data is transferred all at once from the TAAmCCR0 and TAAmCCR1 registers to the CCR0 and CCR1 buffer registers during timer operation. This data is transferred upon a match between the value of the CCR0 buffer register and the value of the 16-bit counter. Transfer is enabled by writing to the TAAmCCR1 register. Whether to enable or disable the next transfer timing is controlled by writing or not writing to the TAAmCCR1 register.

In order for the set value when the TAAmCCR0 and TAAmCCR1 registers are rewritten to become the 16-bit counter comparison value (in other words, in order for this value to be transferred to the CCR0 and CCR1 buffer registers), it is necessary to rewrite the TAAmCCR0 register and then write to the TAAmCCR1 register before the 16-bit counter value and the CCR0 buffer register value match. Therefore, the values of the TAAmCCR0 and TAAmCCR1 registers are transferred to the CCR0 and CCR1 buffer registers upon a match between the count value of the 16-bit counter and the value of the CCR0 buffer register. Thus even when wishing only to rewrite the value of the TAAmCCR0 register, also write the same value (same as preset value of the TAAmCCR1 register) to the TAAmCCR1 register.

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

Figure 6-8. Flowchart of Basic Operation for Batch Write



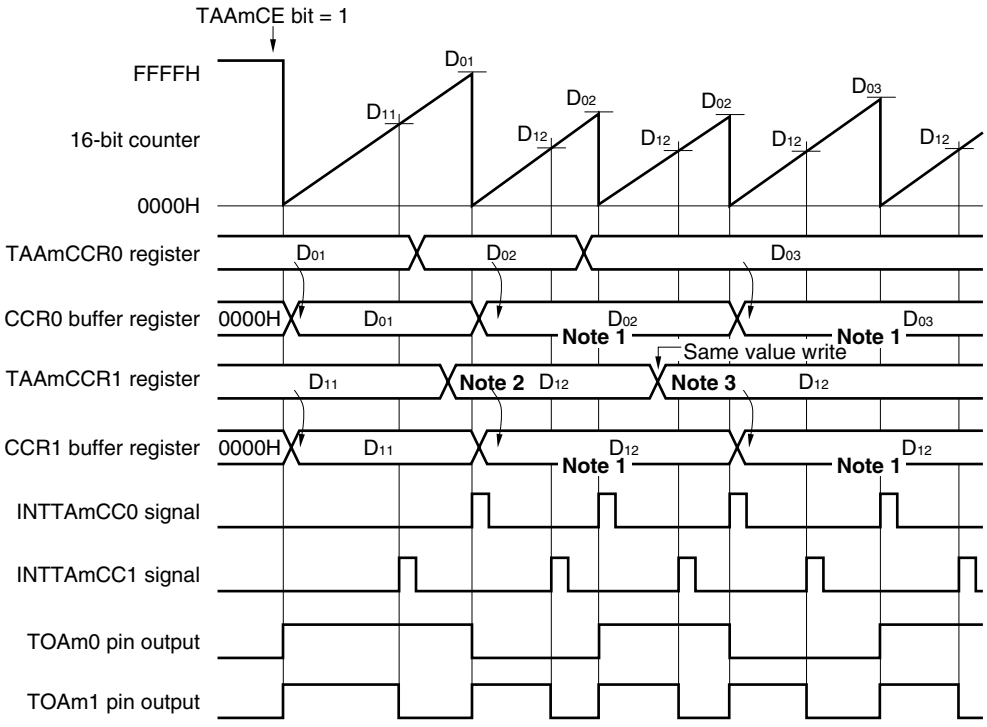
Note The 16-bit counter is not cleared upon a match between the 16-bit counter value and the CCR1 buffer register value. It is cleared upon a match between the 16-bit counter value and the CCR0 buffer register value.

Caution Writing to the TAAmCCR1 register includes enabling of batch write. Thus, rewrite the TAAmCCR1 register after rewriting the TAAmCCR0 register.

Remarks 1. The above flowchart illustrates an example of the operation in the PWM output mode.

2. V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

Figure 6-9. Timing of Batch Write



Notes 1. Because the TAAmCCR1 register was not rewritten, D_{03} is not transferred.

2. Because the TAAmCCR1 register has been written (D_{12}), data is transferred to the CCR1 buffer register upon a match between the value of the 16-bit counter and the value of the TAAmCCR0 register (D_{01}).

3. Because the TAAmCCR1 register has been written (D_{12}), data is transferred to the CCR1 buffer register upon a match between the value of the 16-bit counter and the value of the TAAmCCR0 register (D_{02}).

Remarks 1. D_{01}, D_{02}, D_{03}: Set values of TAAmCCR0 register
D_{11}, D_{12}: Set values of TAAmCCR1 register

2. The above timing chart illustrates the operation in the PWM output mode as an example.

3. V850E/IF3: m = 2, 4
V850E/IG3: m= 2 to 4

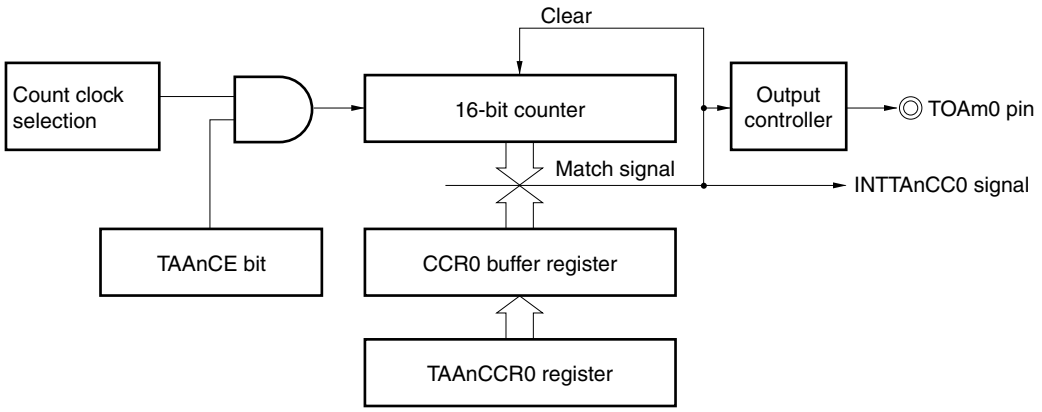
6.6.1 Interval timer mode (TAAnMD2 to TAAnMD0 bits = 000)

In the interval timer mode, an interrupt request signal (INTTAnCC0) is generated at the interval set by the TAAnCCR0 register if the TAAnCTL0.TAAnCE bit is set to 1. A PWM waveform with a duty factor of 50% whose half cycle is equal to the interval can be output from the TOAm0 pin.

The TAAnCCR1 register is not used in the interval timer mode. However, the set value of the TAAnCCR1 register is transferred to the CCR1 buffer register, and when the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTAnCC1) is generated. In addition, a PWM waveform with a duty factor of 50%, which is inverted when the INTTAmCC1 signal is generated, can be output from the TOAm1 pin.

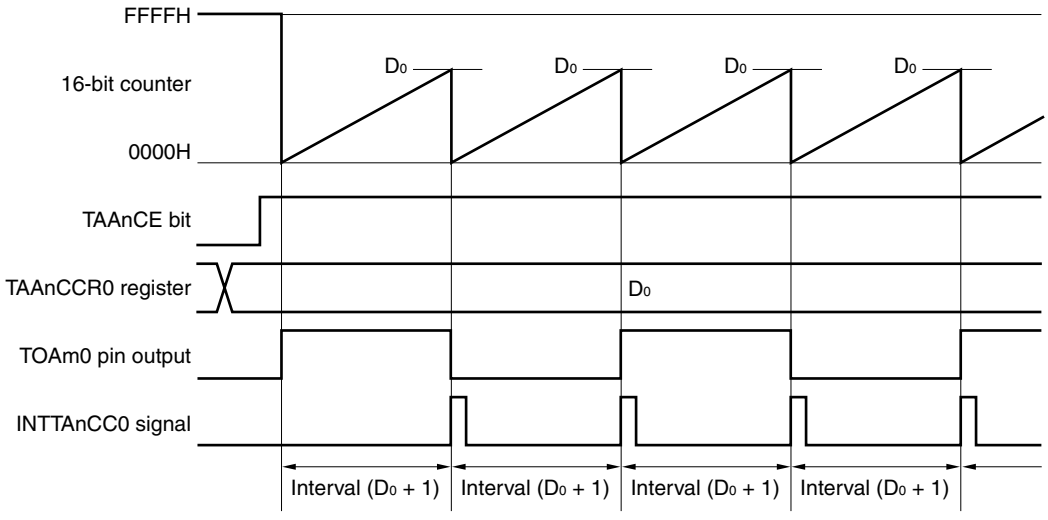
The value of the TAAnCCR0 and TAAnCCR1 registers can be rewritten even while the timer is operating.

Figure 6-10. Configuration of Interval Timer



Remark V850E/IF3: n = 0 to 4, m = 2, 4
V850E/IG3: n = 0 to 4, m = 2 to 4

Figure 6-11. Basic Timing of Operation in Interval Timer Mode



Remark V850E/IF3: n = 0 to 4, m = 2, 4
V850E/IG3: n = 0 to 4, m = 2 to 4

When the TAAnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H in synchronization with the count clock, and the counter starts counting. At this time, the output of the TOAm0 pin is inverted. Additionally, the set value of the TAAnCCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, the output of the TOAm0 pin is inverted, and a compare match interrupt request signal (INTTAnCC0) is generated.

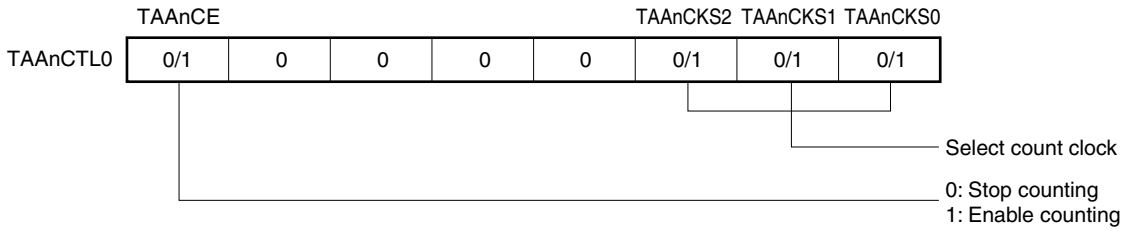
The interval can be calculated by the following expression.

Interval = (Set value of TAAnCCR0 register + 1) × Count clock cycle

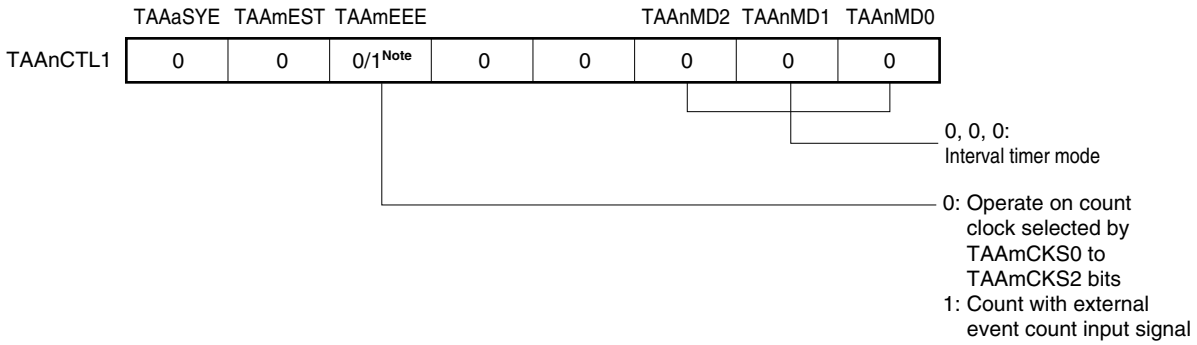
Remark V850E/IF3: n = 0 to 4, m = 2, 4
V850E/IG3: n = 0 to 4, m = 2 to 4

Figure 6-12. Register Setting for Interval Timer Mode Operation (1/3)

(a) TAAn control register 0 (TAAnCTL0)



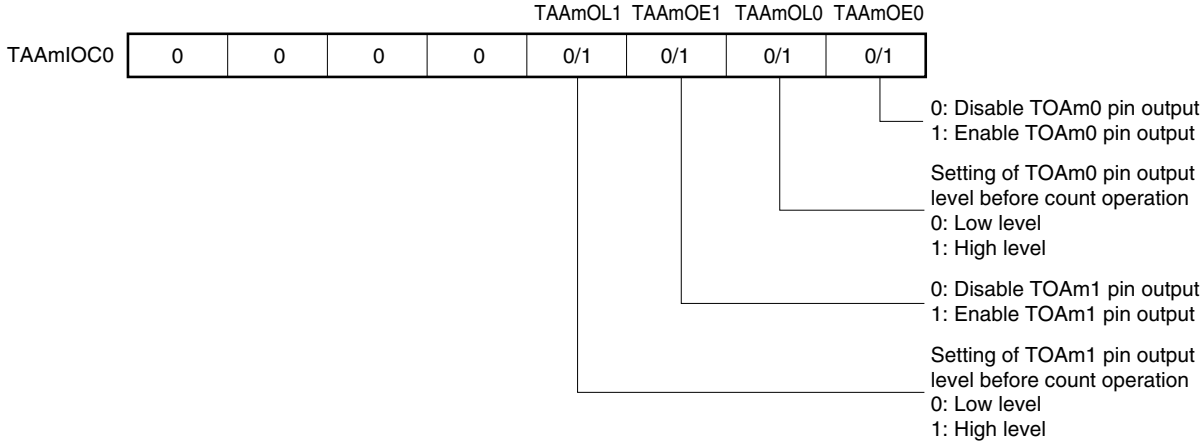
(b) TAAn control register 1 (TAAnCTL1)



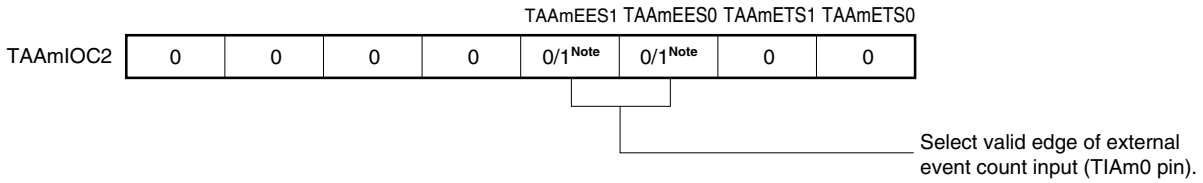
Note The TAAmEEE bit can be set to 1 only when timer output (TOAm1) is used. However, set the TAAmCCR0 and TAAmCCR1 registers to the same value.

Figure 6-12. Register Setting for Interval Timer Mode Operation (2/3)

(c) TAAm I/O control register 0 (TAAmIOC0)



(d) TAAm I/O control register 2 (TAAmIOC2)



Note The TAAmEES1 and TAAmEES0 bits can be set only when timer output (TOAm1) is used. However, set the TAAmCCR0 and TAAmCCR1 registers to the same value.

(e) TAAn counter read buffer register (TAAnCNT)

By reading the TAAnCNT register, the count value of the 16-bit counter can be read.

(f) TAAn capture/compare register 0 (TAAnCCR0)

If the TAAnCCR0 register is set to D_0, the interval is as follows.

Interval = $(D_0 + 1) \times$ Count clock cycle

Figure 6-12. Register Setting for Interval Timer Mode Operation (3/3)

(g) TAAn capture/compare register 1 (TAAnCCR1)

The TAAnCCR1 register is not used in the interval timer mode. However, the set value of the TAAnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, the TOAm1 pin output is inverted and a compare match interrupt request signal (INTTAnCC1) is generated.

By setting this register to the same value as the value set in the TAAnCCR0 register, a PWM waveform with a duty factor of 50% can be output from the TOAm1 pin.

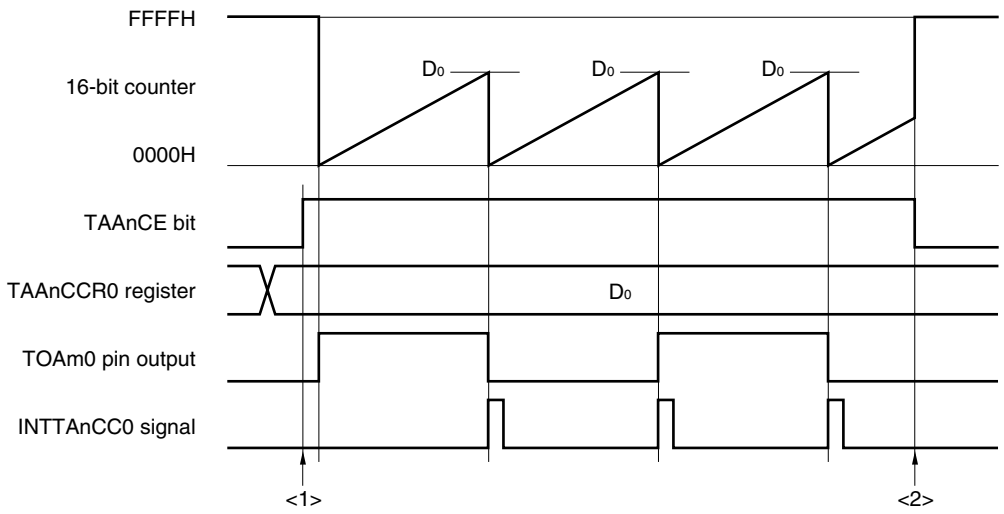
When the TAAnCCR1 register is not used, it is recommended to set the value to FFFFH. Also mask the register by the interrupt mask flag (TAAnCCIC1.TAAnCCMK1).

Remarks 1. TAAm I/O control register 1 (TAAmIOC1) and TAAn option register 0 (TAAnOPT0) are not used in the interval timer mode.

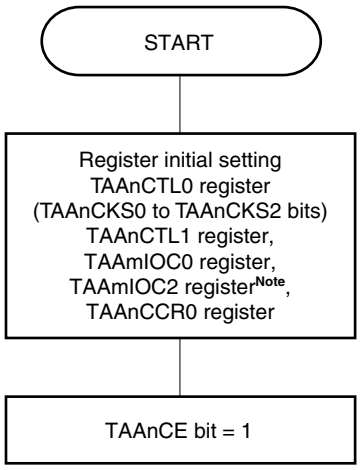
2. V850E/IF3: n = 0 to 4, m = 2, 4, a = 0, 1
V850E/IG3: n = 0 to 4, m = 2 to 4, a = 0, 1

(1) Interval timer mode operation flow

Figure 6-13. Software Processing Flow in Interval Timer Mode (1/2)



<1> Count operation start flow



Initial setting of these registers is performed before setting the TAAnCE bit to 1.

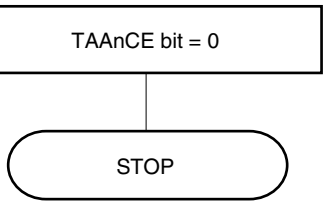
The TAAnCKS0 to TAAnCKS2 bits can be set at the same time when counting has been started (TAAnCE bit = 1).

Note The TAAmEES1 and TAAmEES0 bits can be set only when timer output (TOAm1) is used. However, set the TAAmCCR0 and TAAmCCR1 registers to the same value.

Remark V850E/IF3: n = 0 to 4, m = 2, 4
V850E/IG3: n = 0 to 4, m = 2 to 4

Figure 6-13. Software Processing Flow in Interval Timer Mode (2/2)

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TAAnCE bit to 0. The output level of the TOAm0 pin is as specified by the TAAmIOC0 register.

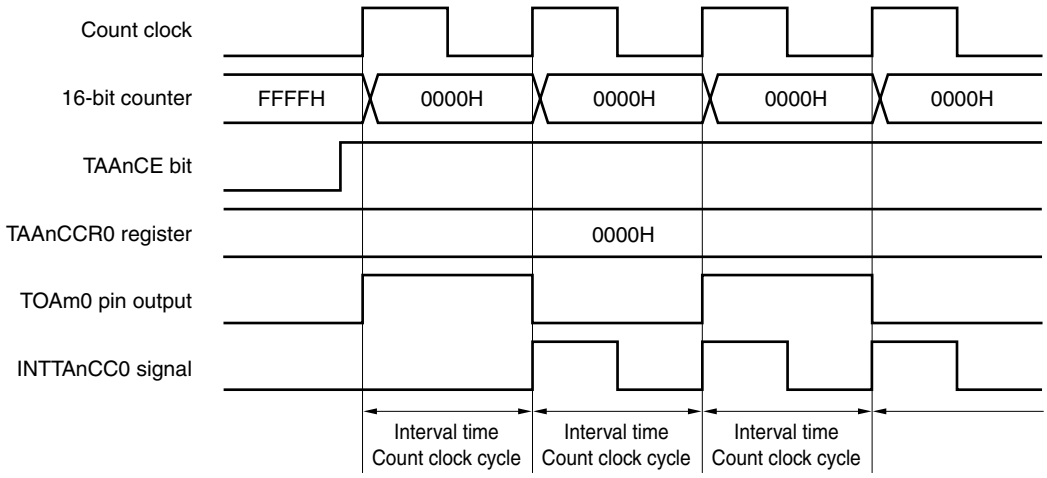
Remark V850E/IF3: n = 0 to 4, m = 2, 4
V850E/IG3: n = 0 to 4, m = 2 to 4

(2) Interval timer mode operation timing

(a) Operation if TAAnCCR0 register is set to 0000H

If the TAAnCCR0 register is set to 0000H, the INTTAnCC0 signal is generated at each count clock, and the output of the TOAm0 pin is inverted.

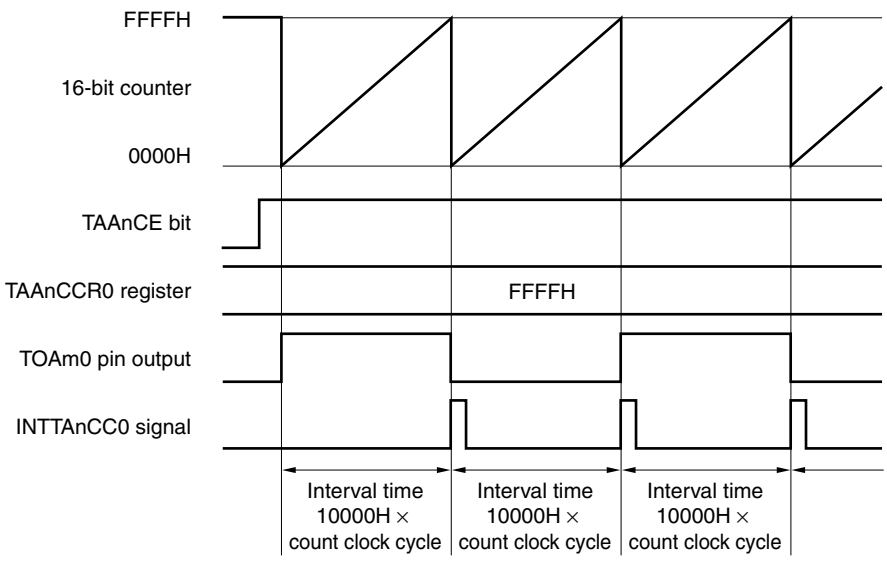
The value of the 16-bit counter is always 0000H.



Remark V850E/IF3: n = 0 to 4, m = 2, 4
V850E/IG3: n = 0 to 4, m = 2 to 4

(b) Operation if TAAnCCR0 register is set to FFFFH

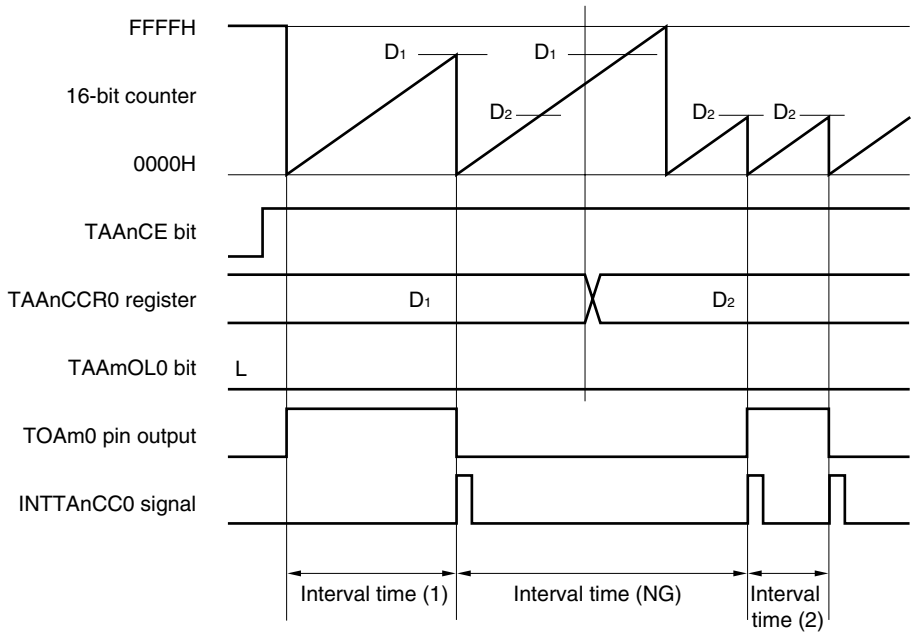
If the TAAnCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTAnCC0 signal is generated and the output of the TOAm0 pin is inverted. At this time, an overflow interrupt request signal (INTTAnOV) is not generated, nor is the overflow flag (TAAnOPT0.TAAnOVF bit) set to 1.



Remark V850E/IF3: n = 0 to 4, m = 2, 4
V850E/IG3: n = 0 to 4, m = 2 to 4

(c) Notes on rewriting TAAnCCR0 register

If the value of the TAAnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When an overflow may occur, stop counting and then change the set value.



Remarks 1. Interval time (1): $(D_1 + 1) \times$ Count clock cycle
Interval time (NG): $(10000H + D_2 + 1) \times$ Count clock cycle
Interval time (2): $(D_2 + 1) \times$ Count clock cycle

2. V850E/IF3: n = 0 to 4, m = 2, 4
V850E/IG3: n = 0 to 4, m = 2 to 4

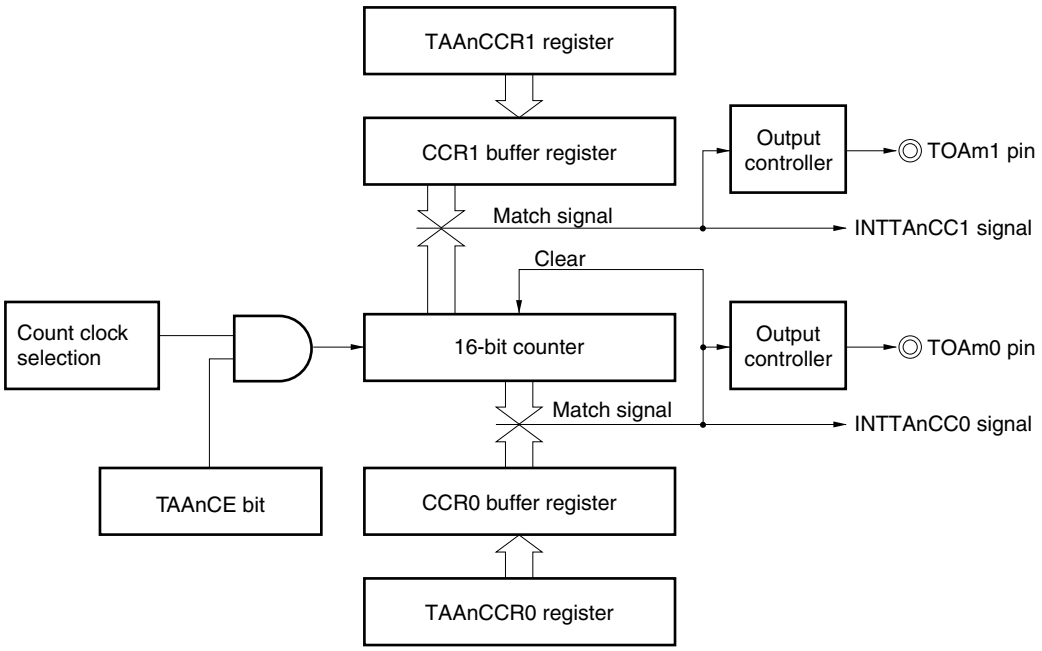
If the value of the TAAnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TAAnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTAnCC0 signal is generated and the output of the TOAm0 pin is inverted.

Therefore, the INTTAnCC0 signal may not be generated at the interval time "$(D_1 + 1) \times$ Count clock cycle" or "$(D_2 + 1) \times$ Count clock cycle" originally expected, but may be generated at an interval of "$(10000H + D_2 + 1) \times$ Count clock cycle".

(d) Operation of TAAnCCR1 register

Figure 6-14. Configuration of TAAnCCR1 Register



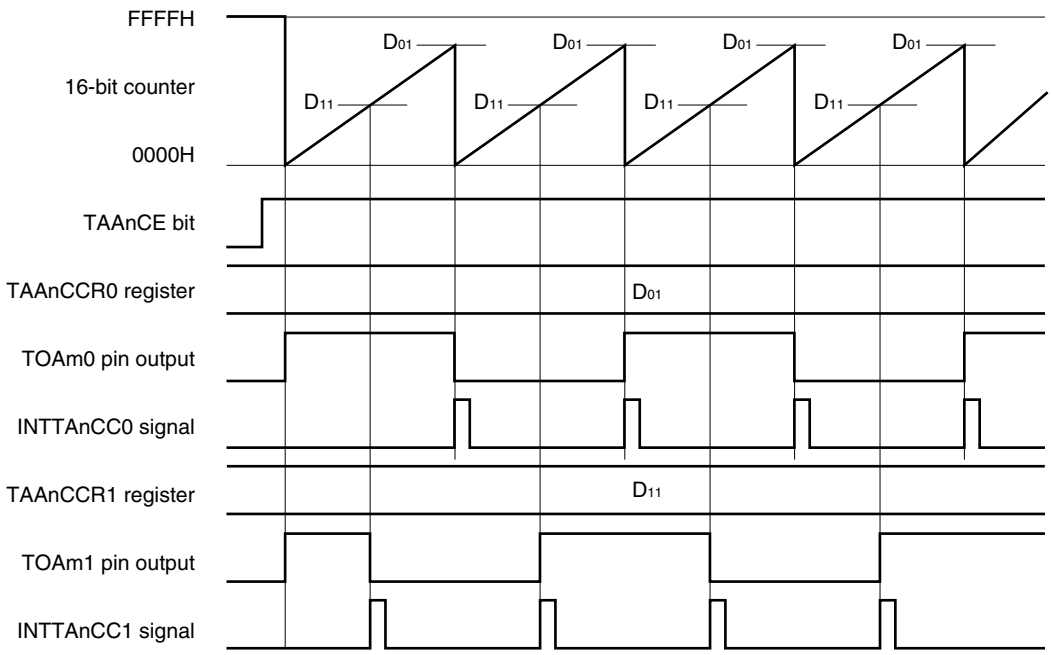
Remark V850E/IF3: n = 0 to 4, m = 2, 4
V850E/IG3: n = 0 to 4, m = 2 to 4

When the TAAnCCR1 register is set to the same value as the TAAnCCR0 register, the INTTAnCC0 signal is generated at the same timing as the INTTAnCC1 signal and the TOAm1 pin output is inverted. In other words, a PWM waveform with a duty factor of 50% can be output from the TOAm1 pin.

The following shows the operation when the TAAnCCR1 register is set to other than the value set in the TAAnCCR0 register.

If the set value of the TAAnCCR1 register is less than the set value of the TAAnCCR0 register, the INTTAnCC1 signal is generated once per cycle. At the same time, the output of the TOAm1 pin is inverted. The TOAm1 pin outputs a PWM waveform with a duty factor of 50% after outputting a short-width pulse.

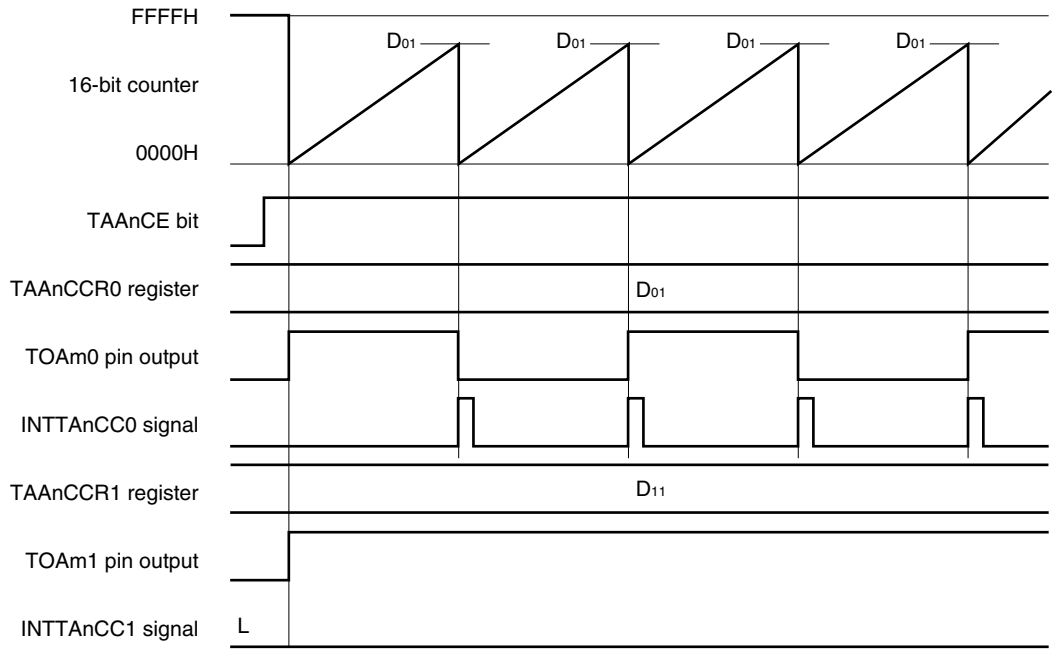
Figure 6-15. Timing Chart When $D_{01} \geq D_{11}$



Remark V850E/IF3: n = 0 to 4, m = 2, 4
V850E/IG3: n = 0 to 4, m = 2 to 4

If the set value of the TAAnCCR1 register is greater than the set value of the TAAnCCR0 register, the count value of the 16-bit counter does not match the value of the TAAnCCR1 register. Consequently, the INTTAnCC1 signal is not generated, nor is the output of the TOAm1 pin changed.
When the TAAnCCR1 register is not used, it is recommended to set its value to FFFFH.

Figure 6-16. Timing Chart When $D_{01} < D_{11}$



Remark V850E/IF3: n = 0 to 4, m = 2, 4
V850E/IG3: n = 0 to 4, m = 2 to 4

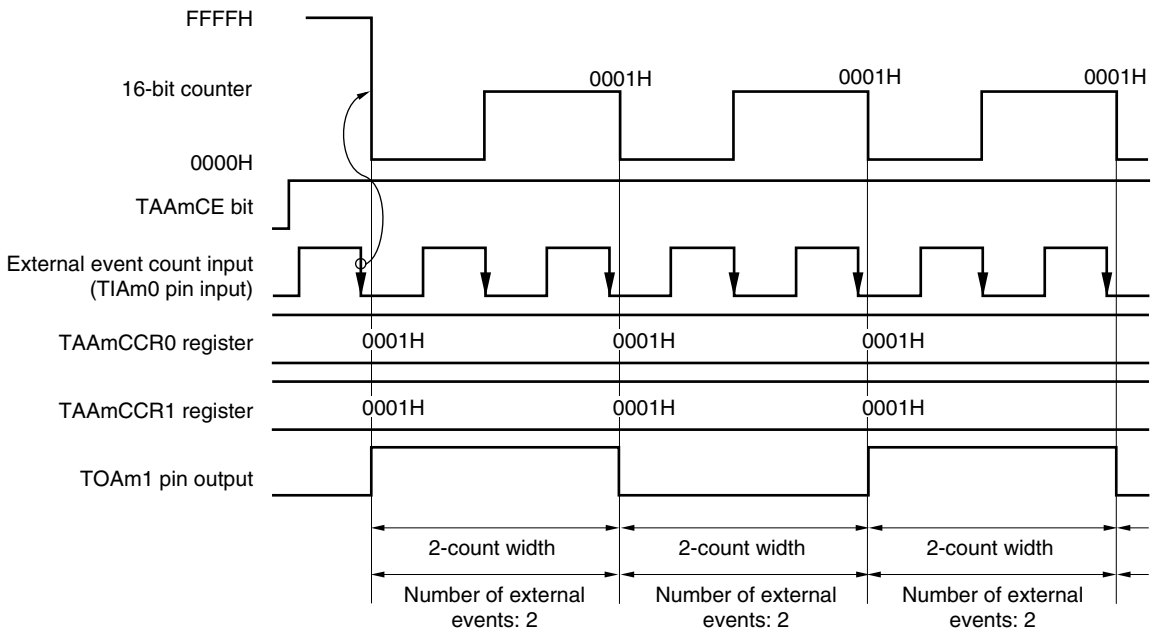
(3) Operation by external event count input (TIAm0)

(a) Operation

To count the 16-bit counter at the valid edge of the external event count input (TIAm0) in the interval timer mode, the 16-bit counter is cleared from FFFFH to 0000H by the valid edge of the external event count after the TAAmCE bit is set from 0 to 1.

When 0001H is set to both the TAAmCCR0 and TAAmCCR1 registers, the TOAm1 pin output is inverted each time the 16-bit counter counts twice.

The TAAmCTL1.TAAmEEE bit can be set to 1 in the interval timer mode only when the timer output (TOAm1) is used with the external event count input.



Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

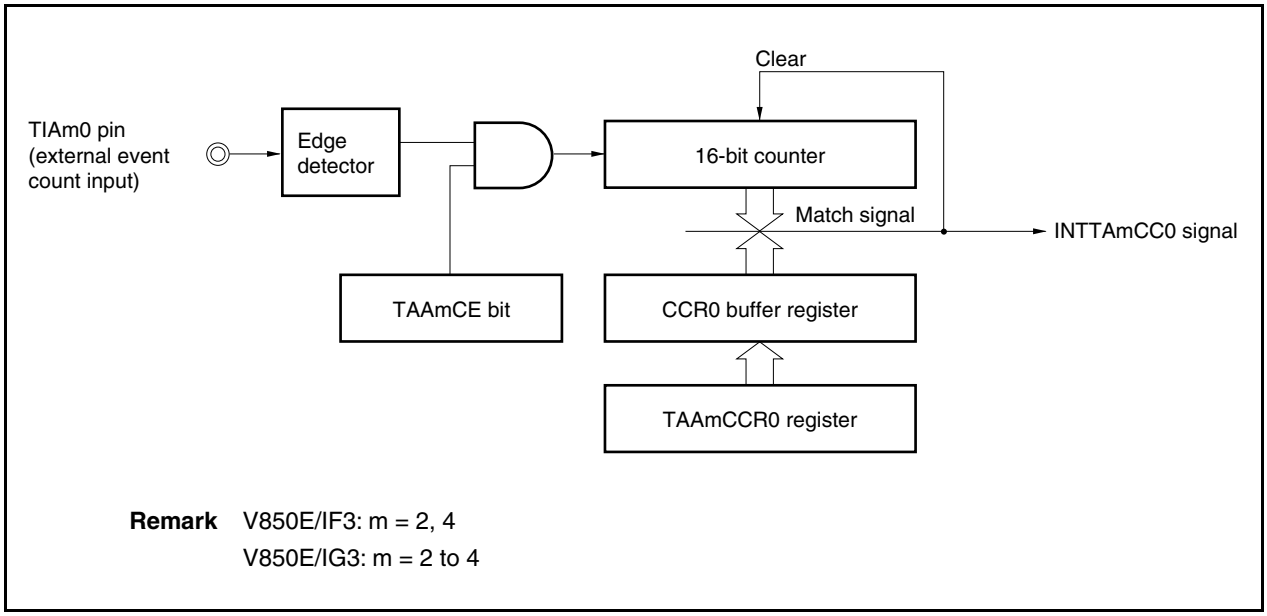
6.6.2 External event count mode (TAAmMD2 to TAAmMD0 bits = 001)

This mode is valid only in TAA2, TAA3 (V850E/IG3 only), and TAA4.

In the external event count mode, the valid edge of the external event count input (TIAm0) is counted when the TAAmCTL0.TAAmCE bit is set to 1, and an interrupt request signal (INTTAmCC0) is generated each time the number of edges set by the TAAmCCR0 register have been counted. The TOAm0 and TOAm1 pins cannot be used. When using the TOAm1 pin for external event count input, set the TAAmCTL1.TAAmEEE bit to 1 in the interval timer mode (see **6.6.1 (3) Operation by external event count input (TIAm0)**).

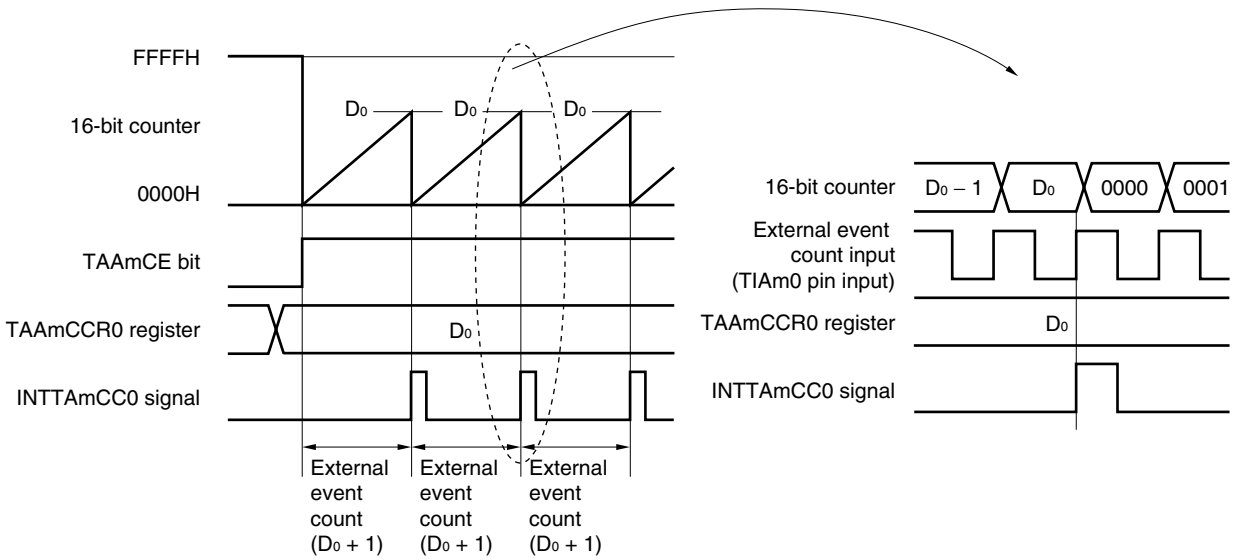
The TAAmCCR1 register is not used in the external event count mode.

Figure 6-17. Configuration in External Event Count Mode



Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

Figure 6-18. Basic Timing in External Event Count Mode



Remarks 1. This figure shows the basic timing when the rising edge is specified as the valid edge of the external event count input.

2. V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

When the TAAmCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H. The counter counts each time the valid edge of external event count input is detected. Additionally, the set value of the TAAmCCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTAmCC0) is generated.

The INTTAmCC0 signal is generated each time the valid edge of the external event count input has been detected "value set to TAAmCCR0 register + 1" times.

Figure 6-19. Register Setting for Operation in External Event Count Mode

(a) TAAm control register 0 (TAAmCTL0)

	TAAmCE					TAAmCKS2	TAAmCKS1	TAAmCKS0
TAAmCTL0	0/1	0	0	0	0	0	0	0

0: Stop counting
1: Enable counting

(b) TAAm control register 1 (TAAmCTL1)

	TAAaSYE	TAAmEST	TAAmEEE			TAAmMD2	TAAmMD1	TAAmMD0
TAAmCTL1	0	0	0	0	0	0	0	1

0, 0, 1:
External event count mode

(c) TAAm I/O control register 2 (TAAmIOC2)

					TAAmEES1	TAAmEES0	TAAmETS1	TAAmETS0
TAAmIOC2	0	0	0	0	0/1	0/1	0	0

Select valid edge
of external event
count input (TIAm0 pin)

(d) TAAm counter read buffer register (TAAmCNT)

The count value of the 16-bit counter can be read by reading the TAAmCNT register.

(e) TAAm capture/compare register 0 (TAAmCCR0)

If the TAAmCCR0 register is set to D_0, the count is cleared when the number of external events has reached (D_0 + 1) and the compare match interrupt request signal (INTTAmCC0) is generated.

(f) TAAm capture/compare register 1 (TAAmCCR1)

The TAAmCCR1 register is not used in the external event count mode. However, the set value of the TAAmCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTAmCC1) is generated.

When the TAAmCCR1 register is not used, it is recommended to set the value to FFFFH. Also mask the register by the interrupt mask flag (TAAmCCIC1.TAAmCCMK1).

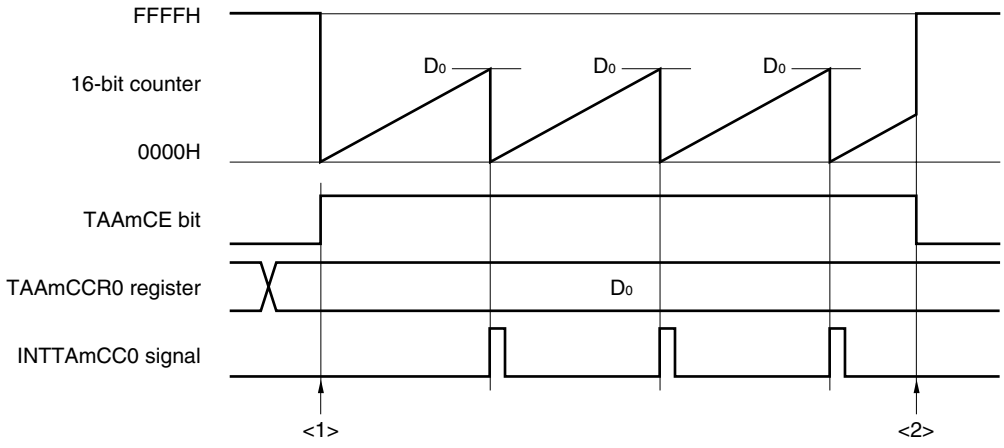
Caution Set the TAAmIOC0 register to 00H.

Remarks 1. TAAm I/O control register 1 (TAAmIOC1) and TAAm option register 0 (TAAmOPT0) are not used in the external event count mode.

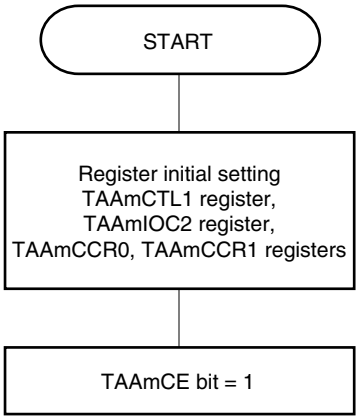
2. V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

(1) External event count mode operation flow

Figure 6-20. Software Processing Flow in External Event Count Mode

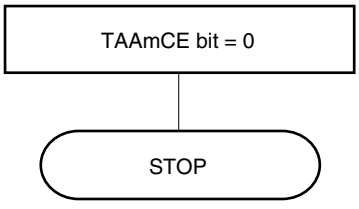


<1> Count operation start flow



Initial setting of these registers is performed before setting the TAAmCE bit to 1.

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TAAmCE bit to 0.

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

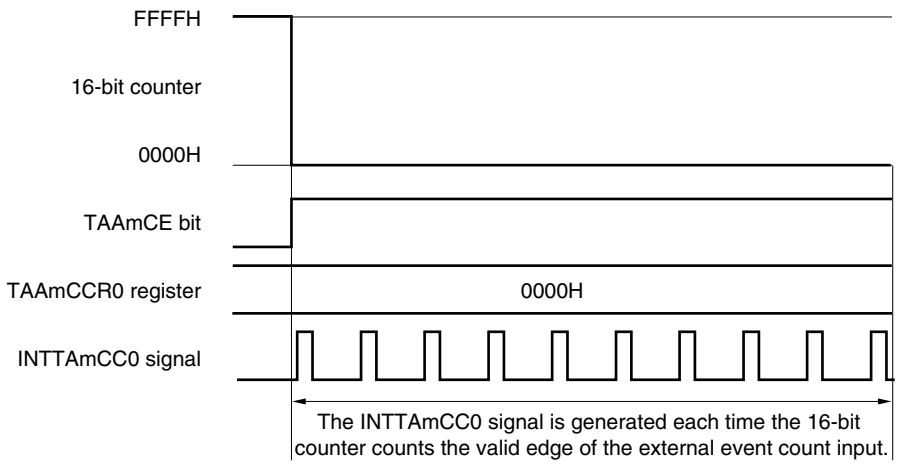
(2) Operation timing in external event count mode

Caution In the external event count mode, use of the timer output (TOAm0, TOAm1) is disabled. If using timer output (TOAm1) with external event count input (TIAm0), set the interval timer mode, and select the operation enabled by the external event count input for the count clock (TAAmCTL1.TAAmEEE bit = 1) (see 6.6.1 (3) Operation by external event count input (TIAm0)).

(a) Operation if TAAmCCR0 register is set to 0000H

When the TAAmCCR0 register is set to 0000H, the 16-bit counter is repeatedly cleared to 0000H and generates the INTTAmCC0 signal each time it has detected the valid edge of the external event count signal and its value has matched that of the CCR0 buffer register.

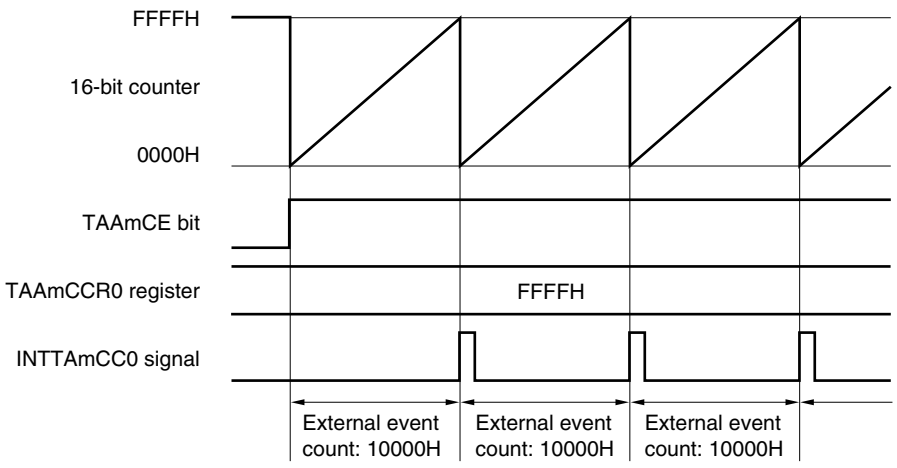
The value of the 16-bit counter is always 0000H.



Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

(b) Operation if TAAmCCR0 register is set to FFFFH

If the TAAmCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTTAmCC0 signal is generated. At this time, the TAAmOPT0.TAAmOVF bit is not set.

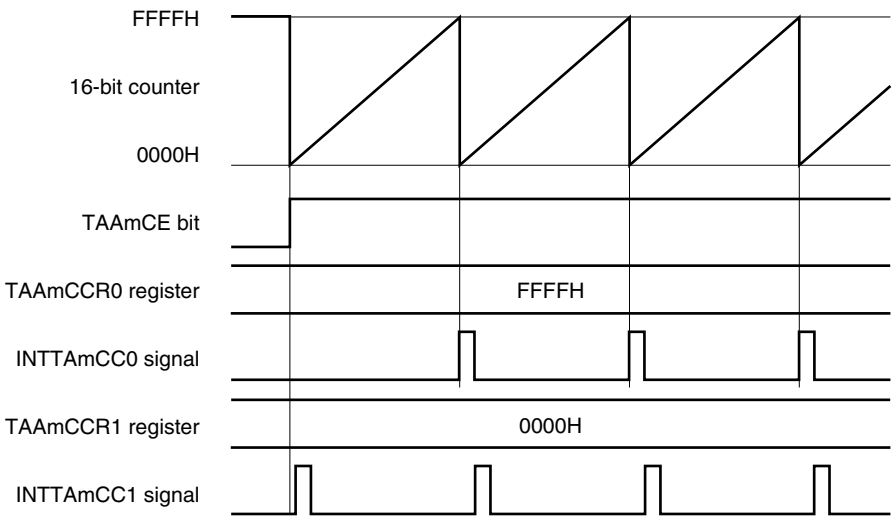


Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

(c) Operation with TAAmCCR0 register set to FFFFH and TAAmCCR1 register to 0000H

When the TAAmCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH each time it has detected the valid edge of the external event count signal. The counter is then cleared to 0000H in synchronization with the next count-up timing and the INTTAmCC0 signal is generated. At this time, the TAAmOPT0.TAAmOVF bit is not set.

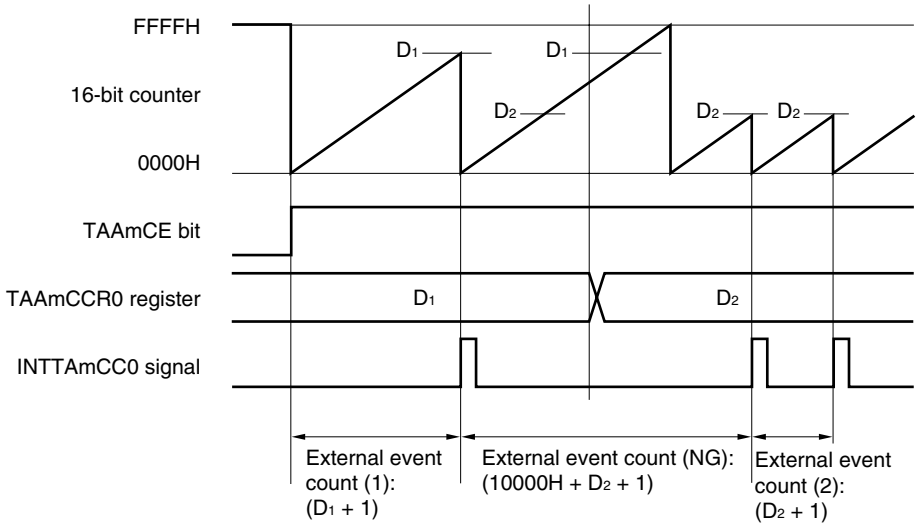
If the TAAmCCR1 register is set to 0000H, the INTTAmCC1 signal is generated when the 16-bit counter is cleared to 0000H.



Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

(d) Notes on rewriting the TAAmCCR0 register

If the value of the TAAmCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When the overflow may occur, stop counting once and then change the set value.



Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

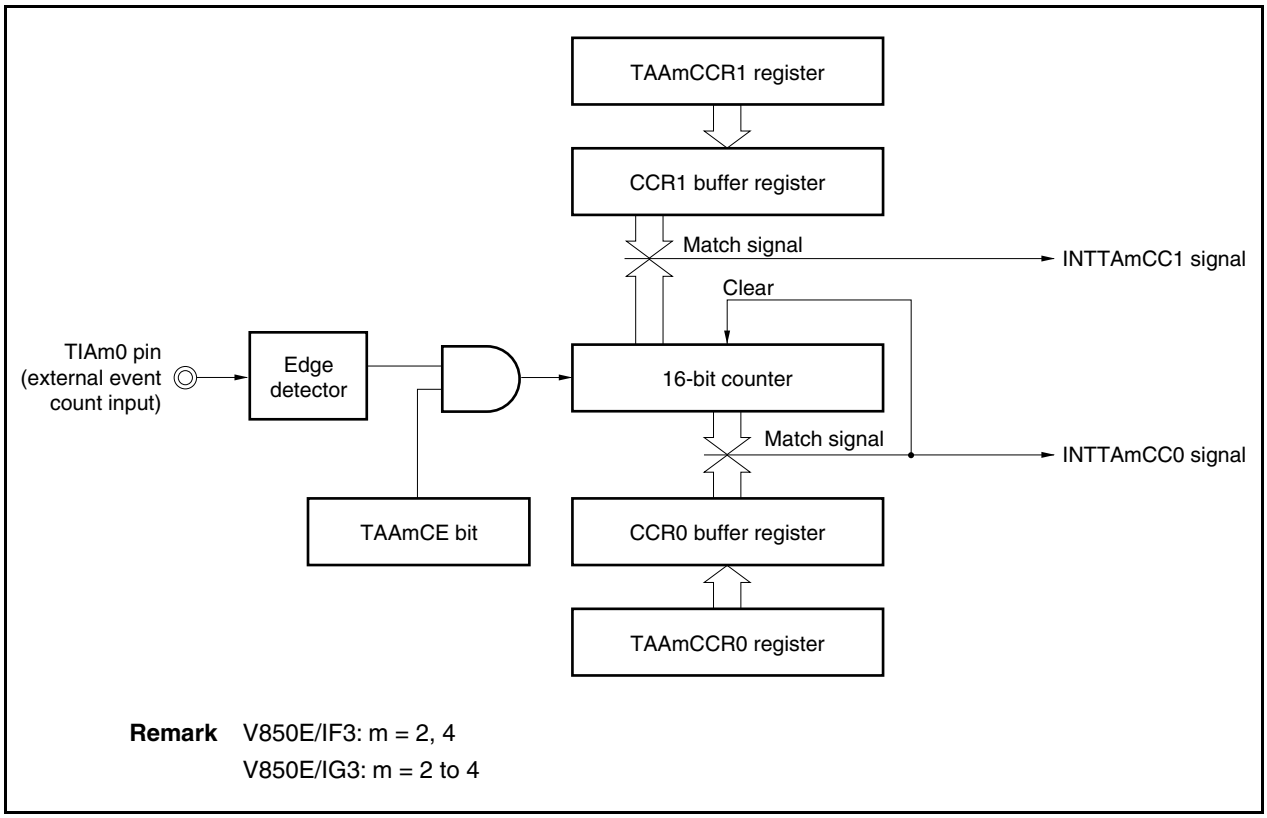
If the value of the TAAmCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TAAmCCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTAmCC0 signal is generated.

Therefore, the INTTAmCC0 signal may not be generated at the valid edge count of "(D_1 + 1) times" or "(D_2 + 1) times" originally expected, but may be generated at the valid edge count of "(10000H + D_2 + 1) times".

(e) Operation of TAAmCCR1 register

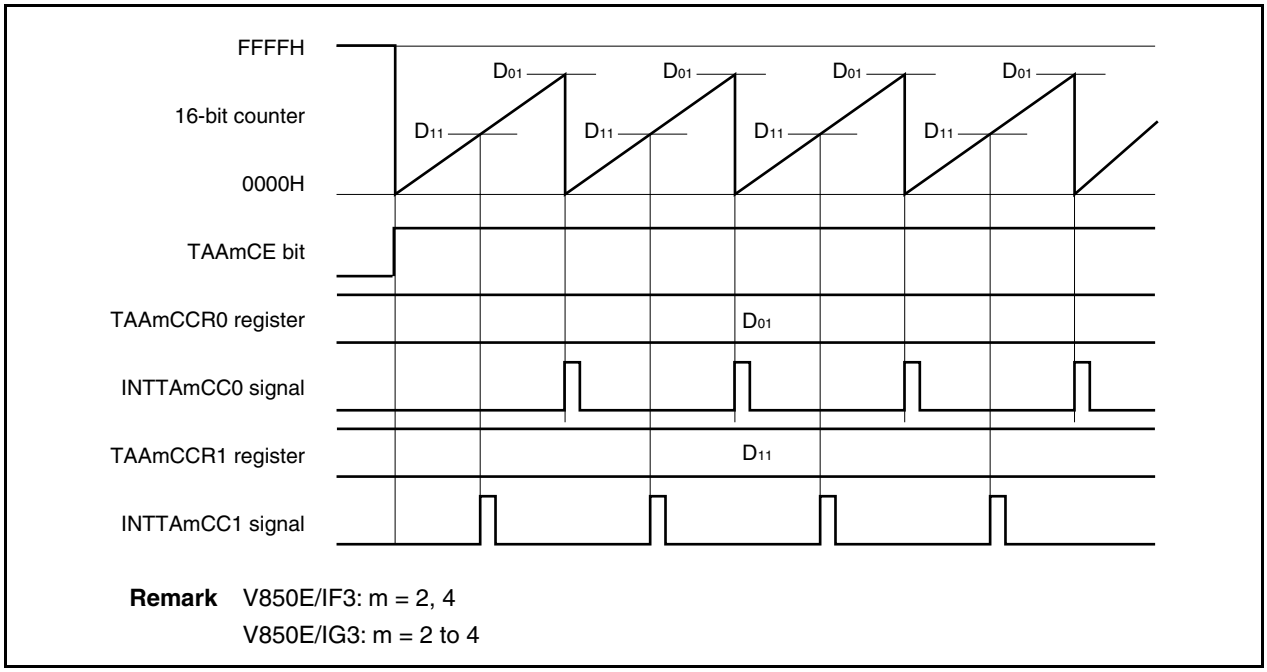
Figure 6-21. Configuration of TAAmCCR1 Register



Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

If the set value of the TAAmCCR1 register is smaller than the set value of the TAAmCCR0 register, the INTTAmCC1 signal is generated once per cycle.

Figure 6-22. Timing Chart When $D_{01} \geq D_{11}$

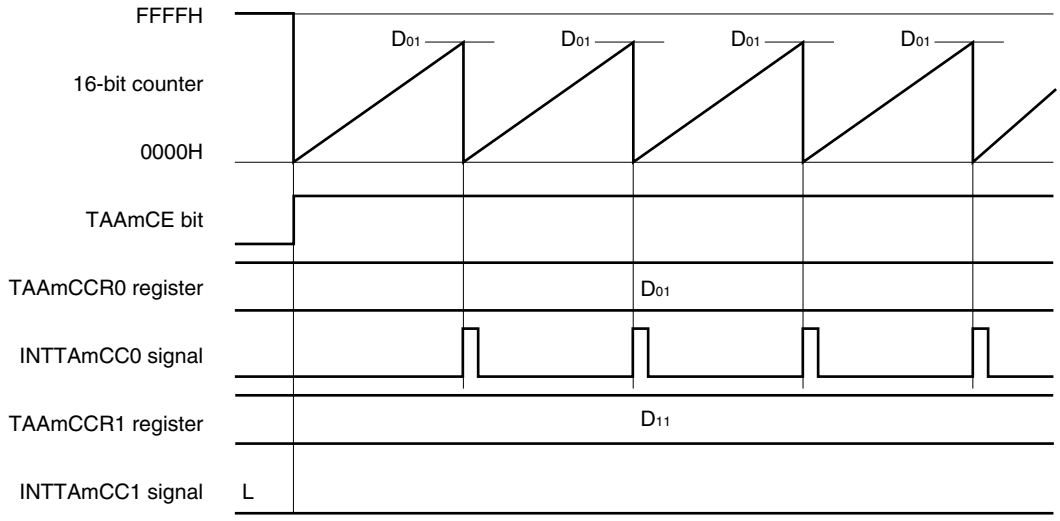


Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

If the set value of the TAAmCCR1 register is greater than the set value of the TAAmCCR0 register, the INTTAmCC1 signal is not generated because the count value of the 16-bit counter and the value of the TAAmCCR1 register do not match.

When the TAAmCCR1 register is not used, it is recommended to set its value to FFFFH.

Figure 6-23. Timing Chart When $D_{01} < D_{11}$



Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

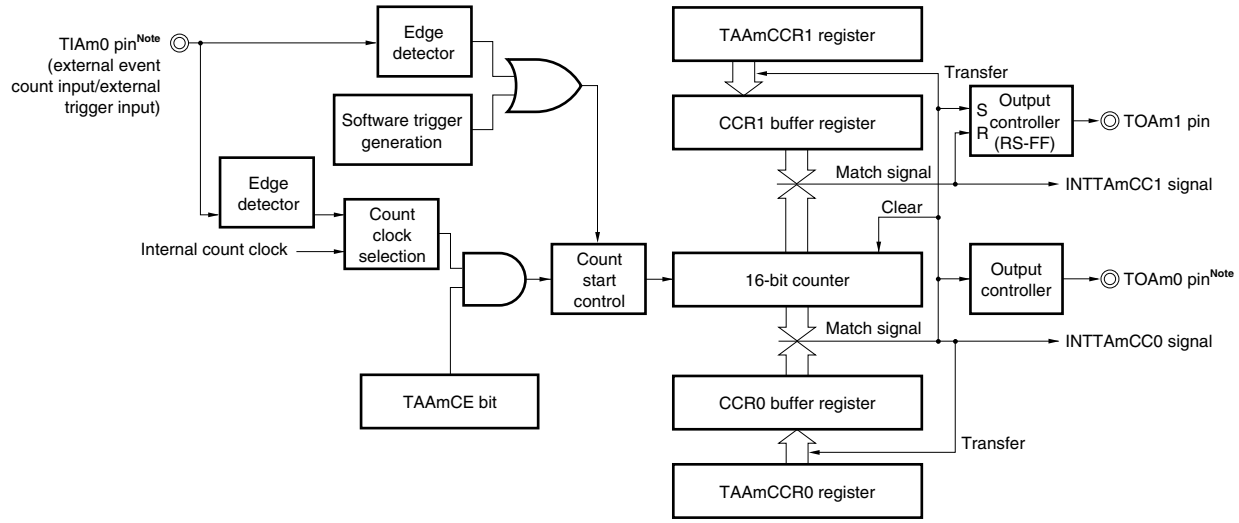
6.6.3 External trigger pulse output mode (TAAmMD2 to TAAmMD0 bits = 010)

This mode is valid only in TAA2, TAA3 (V850E/IG3 only), and TAA4.

In the external trigger pulse output mode, 16-bit timer/event counter AA waits for a trigger when the TAAmCTL0.TAAmCE bit is set to 1. When the valid edge of an external trigger input (TIAm0) is detected, 16-bit timer/event counter AA starts counting, and outputs a PWM waveform from the TOAm1 pin.

Pulses can also be output by generating a software trigger instead of using the external trigger. When using a software trigger, a PWM waveform with a duty factor of 50% that has the set value of the TAAmCCR0 register + 1 as half its cycle can also be output from the TOAm0 pin.

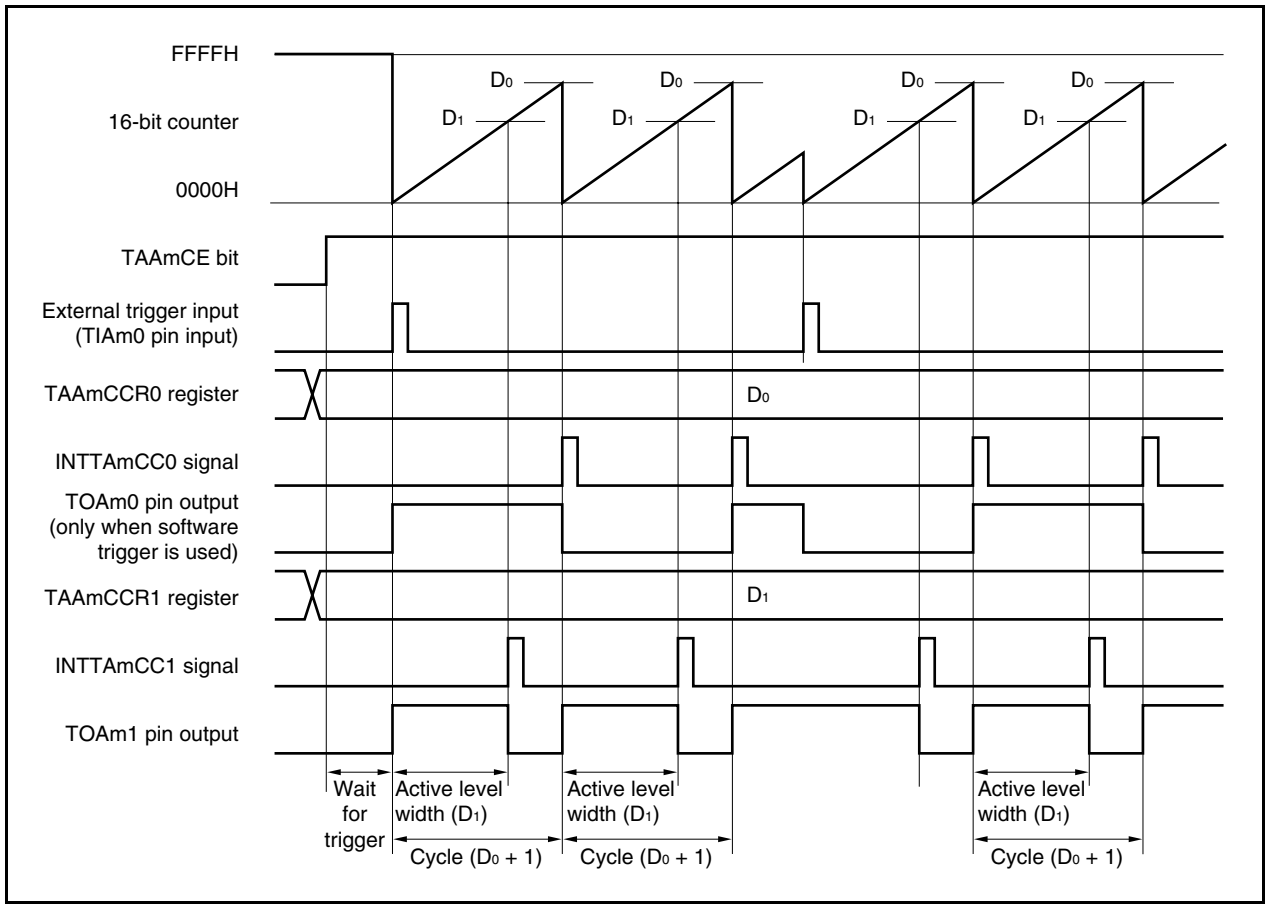
Figure 6-24. Configuration in External Trigger Pulse Output Mode



Note Because the external event count input pin (TIAm0), external trigger input pin (TIAm0), and timer output pin (TOAm0) are assigned to the same alternate-function pin, two or more functions cannot be used at the same time.

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

Figure 6-25. Basic Timing in External Trigger Pulse Output Mode



16-bit timer/event counter AA waits for a trigger when the TAAmCE bit is set to 1. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting at the same time, and outputs a PWM waveform from the TOAm1 pin. If the trigger is generated again while the counter is operating, the counter is cleared to 0000H and restarted. (The output of the TOAm0 pin is inverted. The TOAm1 pin outputs a high-level regardless of the status (high/low) when a trigger occurs.)

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TAAmCCR1 register) × Count clock cycle
Cycle = (Set value of TAAmCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TAAmCCR1 register)/(Set value of TAAmCCR0 register + 1)

The compare match interrupt request signal INTTAmCC0 is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal INTTAmCC1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

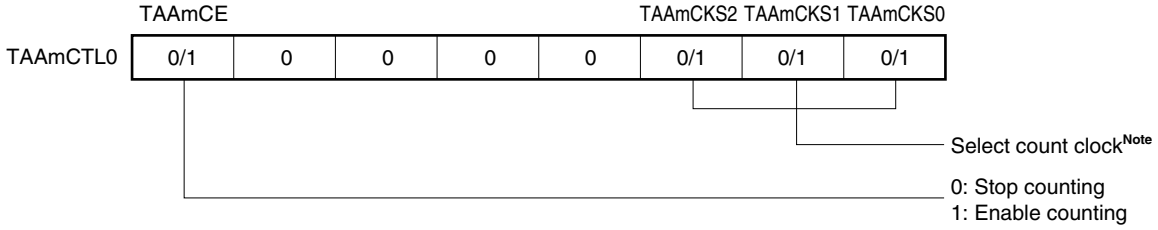
The value set to the TAAmCCRa register is transferred to the CCRa buffer register when the count value of the 16-bit counter matches the value of the CCRa buffer register and the 16-bit counter is cleared to 0000H.

The valid edge of an external trigger input (TIAm0), or setting the software trigger (TAAmCTL1.TAAmEST bit) to 1 is used as the trigger.

Remark V850E/IF3: m = 2, 4, a = 0, 1, V850E/IG3: m = 2 to 4, a = 0, 1

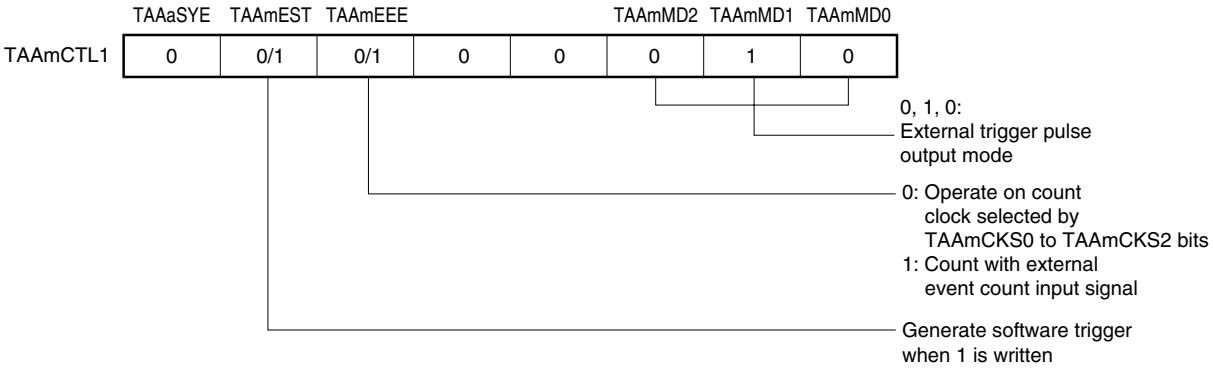
Figure 6-26. Setting of Registers in External Trigger Pulse Output Mode (1/2)

(a) TAAm control register 0 (TAAmCTL0)

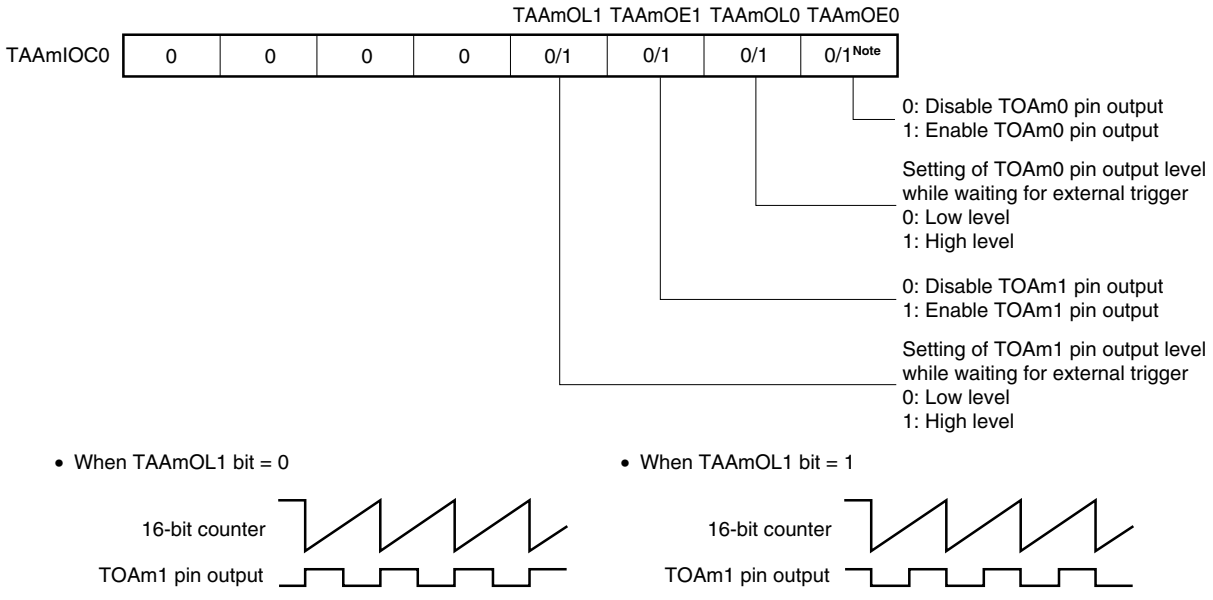


Note The setting is invalid when the TAAmCTL1.TAAmEEE bit = 1.

(b) TAAm control register 1 (TAAmCTL1)



(c) TAAm I/O control register 0 (TAAmIOC0)



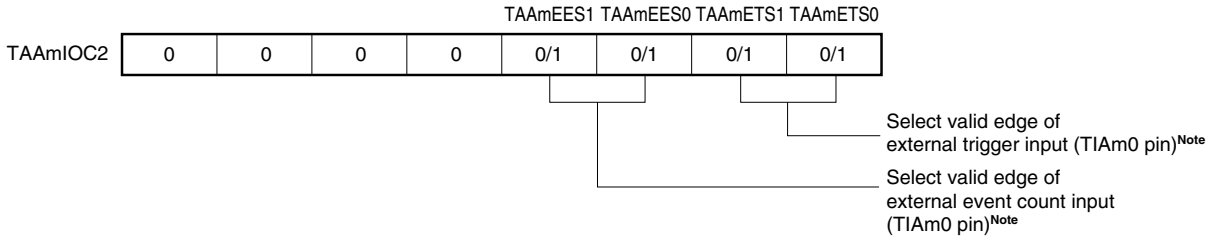
- When TAAmOL1 bit = 0

- When TAAmOL1 bit = 1

Note Clear this bit to 0 when the TOAm0 pin is not used in the external trigger pulse output mode.

Figure 6-26. Setting of Registers in External Trigger Pulse Output Mode (2/2)

(d) TAAm I/O control register 2 (TAAmIOC2)



Note Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

(e) TAAm counter read buffer register (TAAmCNT)

The value of the 16-bit counter can be read by reading the TAAmCNT register.

(f) TAAm capture/compare registers 0 and 1 (TAAmCCR0 and TAAmCCR1)

If D_0 is set to the TAAmCCR0 register and D_1 to the TAAmCCR1 register, the cycle and active level of the PWM waveform are as follows.

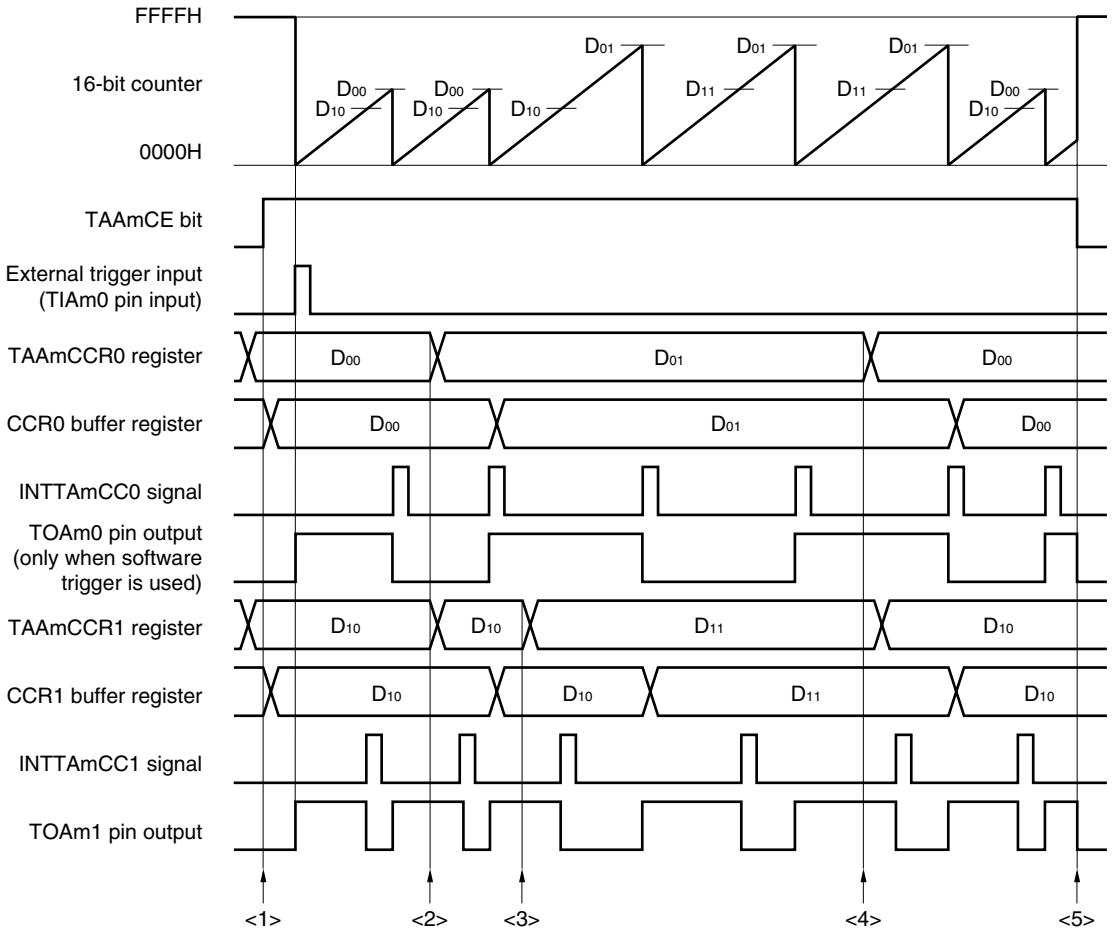
Cycle = $(D_0 + 1) \times$ Count clock cycle
Active level width = $D_1 \times$ Count clock cycle

Remarks 1. TAAm I/O control register 1 (TAAmIOC1) and TAAm option register 0 (TAAmOPT0) are not used in the external trigger pulse output mode.

2. V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

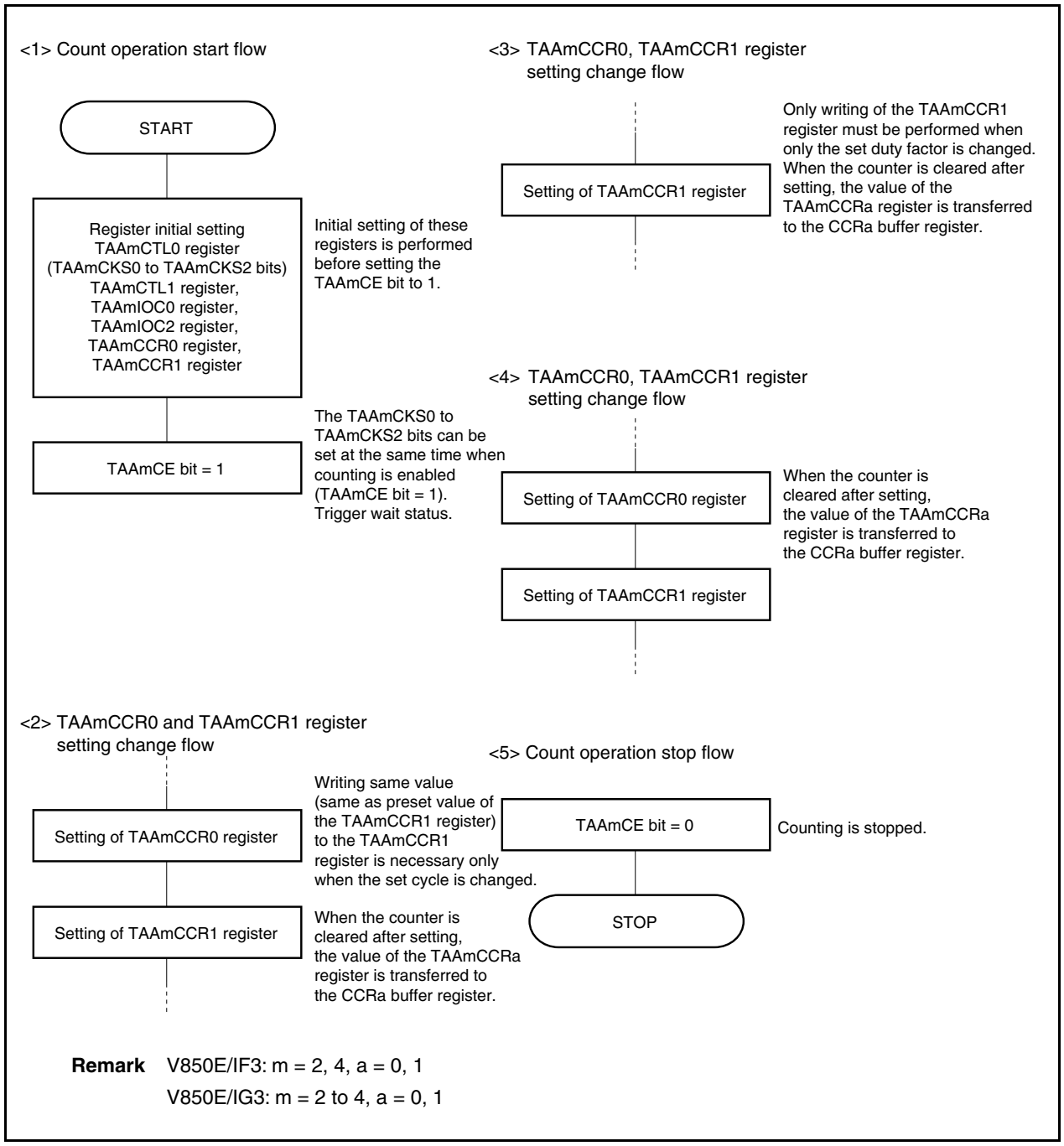
(1) Operation flow in external trigger pulse output mode

Figure 6-27. Software Processing Flow in External Trigger Pulse Output Mode (1/2)



Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

Figure 6-27. Software Processing Flow in External Trigger Pulse Output Mode (2/2)



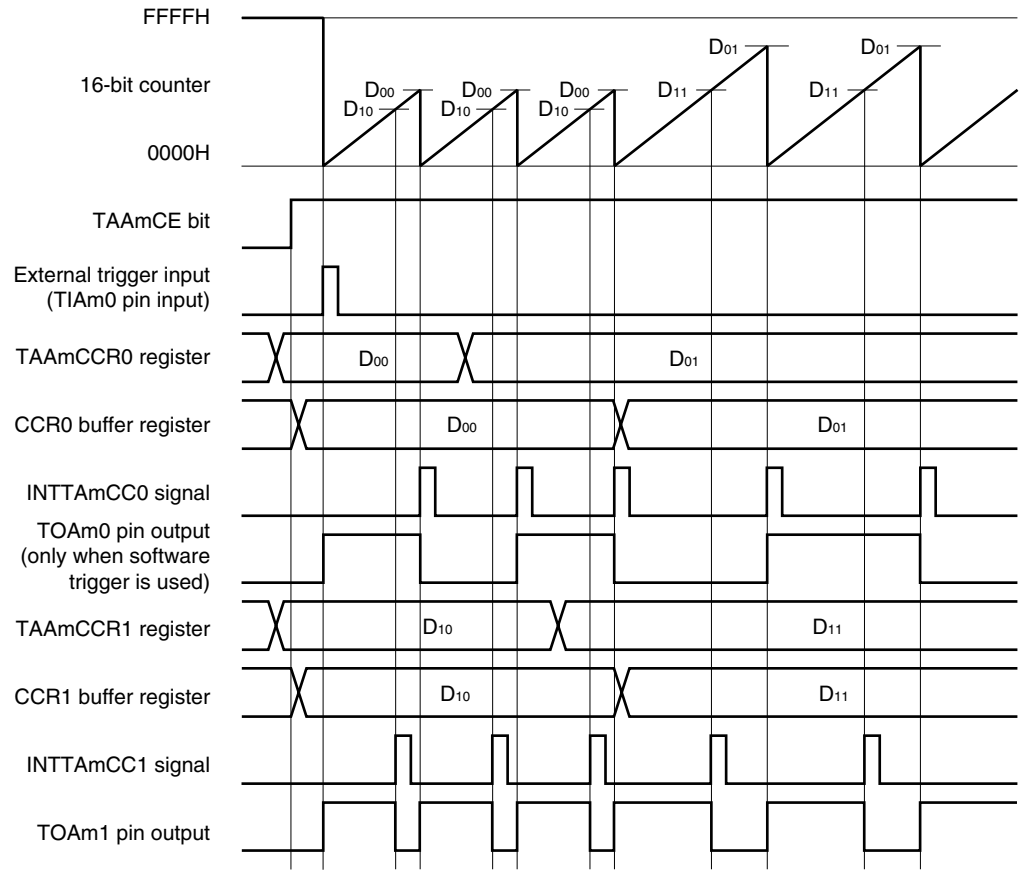
Remark V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

(2) External trigger pulse output mode operation timing

(a) Note on changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TAAmCCR1 register last.

Rewrite the TAAmCCRa register after writing the TAAmCCR1 register after the INTTAmCC0 signal is detected.



Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

In order to transfer data from the TAAmCCRa register to the CCRa buffer register, the TAAmCCR1 register must be written.

To change both the cycle and active level width of the PWM waveform at this time, first set the cycle to the TAAmCCR0 register and then set the active level width to the TAAmCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TAAmCCR0 register, and then write the same value (same as preset value of the TAAmCCR1 register) to the TAAmCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TAAmCCR1 register has to be set.

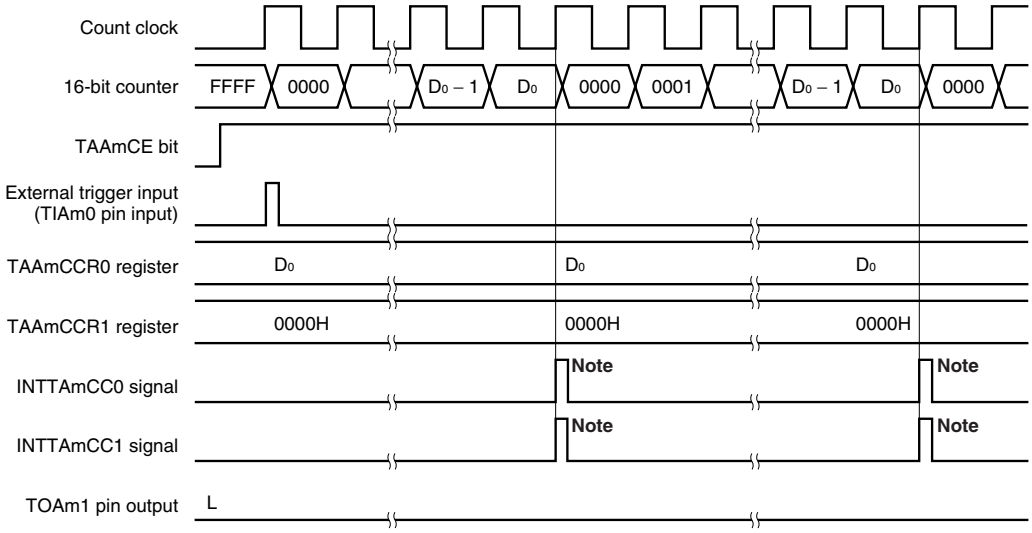
After data is written to the TAAmCCR1 register, the value written to the TAAmCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TAAmCCR0 or TAAmCCR1 register again after writing the TAAmCCR1 register once, do so after the INTTAmCC0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TAAmCCRa register to the CCRa buffer register conflicts with writing the TAAmCCRa register.

Remark V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

(b) 0%/100% output of PWM waveform

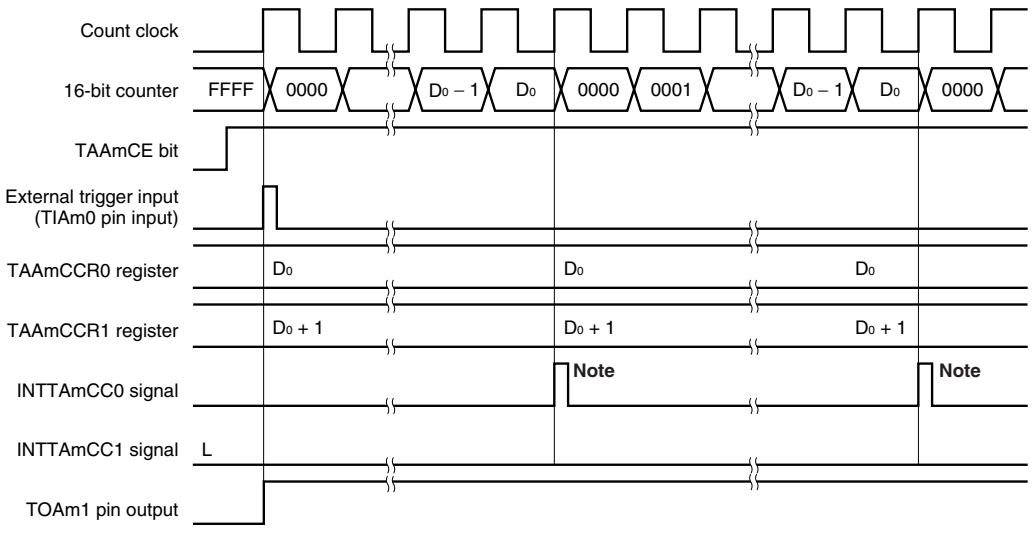
To output a 0% waveform, set the TAAmCCR1 register to 0000H. The 16-bit counter is cleared to 0000H and the INTTAmCC0 and INTTAmCC1 signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.



Note The timing is actually delayed by one operating clock (fxx).

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

To output a 100% waveform, set a value of (set value of TAAmCCR0 register + 1) to the TAAmCCR1 register. If the set value of the TAAmCCR0 register is FFFFH, 100% output cannot be produced.

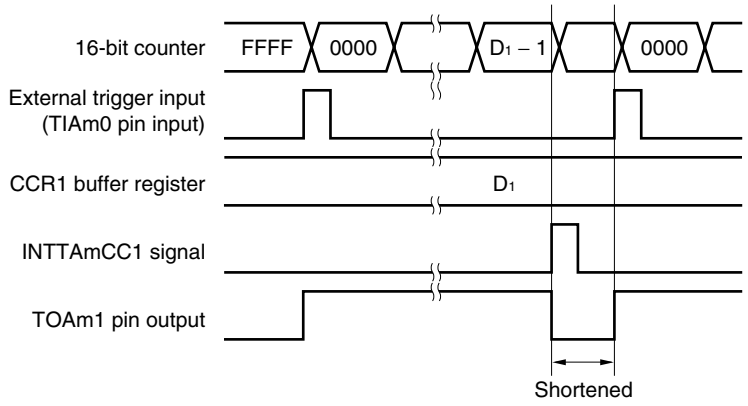


Note The timing is actually delayed by one operating clock (fxx).

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

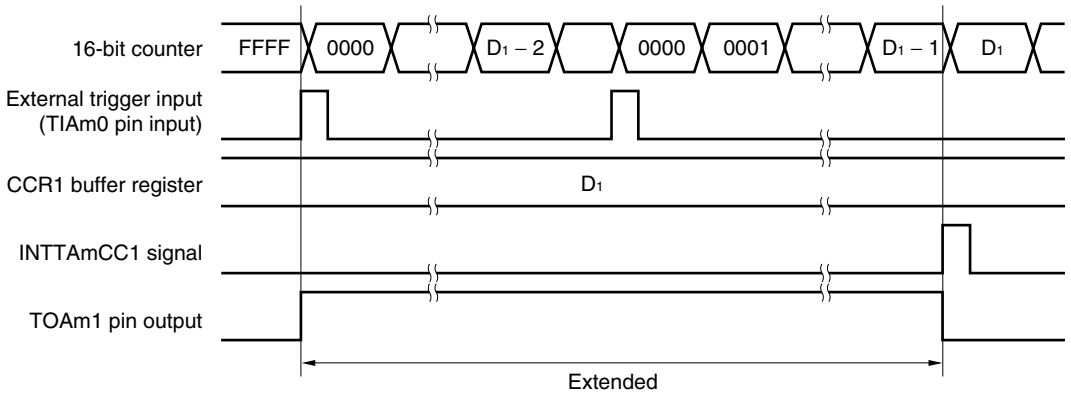
(c) Conflict between trigger detection and match with CCR1 buffer register

If the trigger is detected immediately after the INTTAmCC1 signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TOAm1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

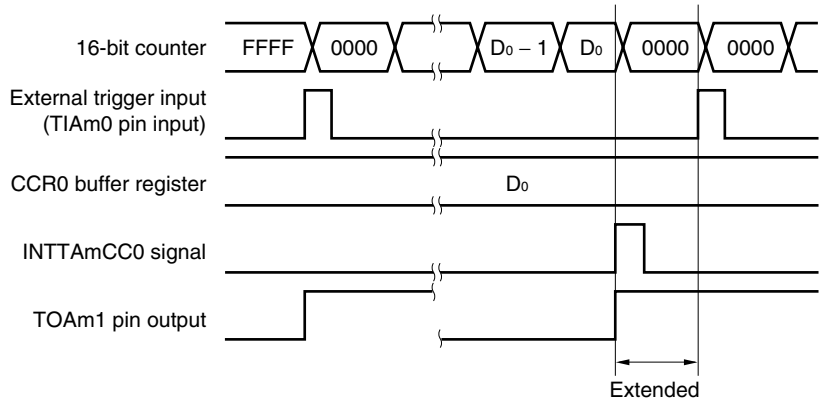
If the trigger is detected immediately before the INTTAmCC1 signal is generated, the INTTAmCC1 signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOAm1 pin remains active. Consequently, the active period of the PWM waveform is extended.



Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

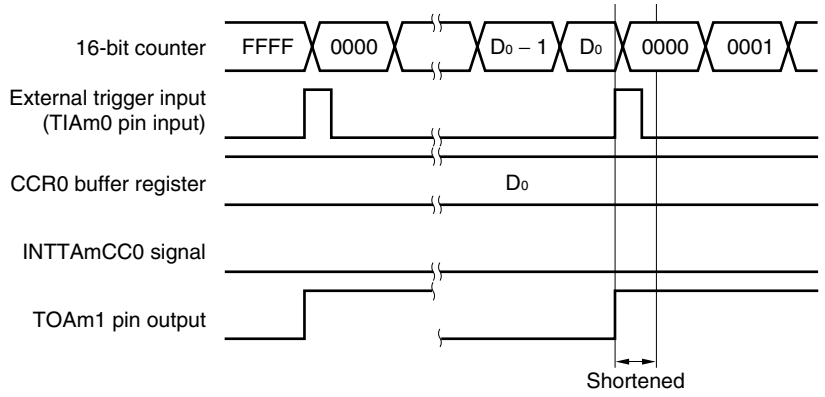
(d) Conflict between trigger detection and match with CCR0 buffer register

If the trigger is detected immediately after the INTTAmCC0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOAm1 pin is extended by time from generation of the INTTAmCC0 signal to trigger detection.



Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

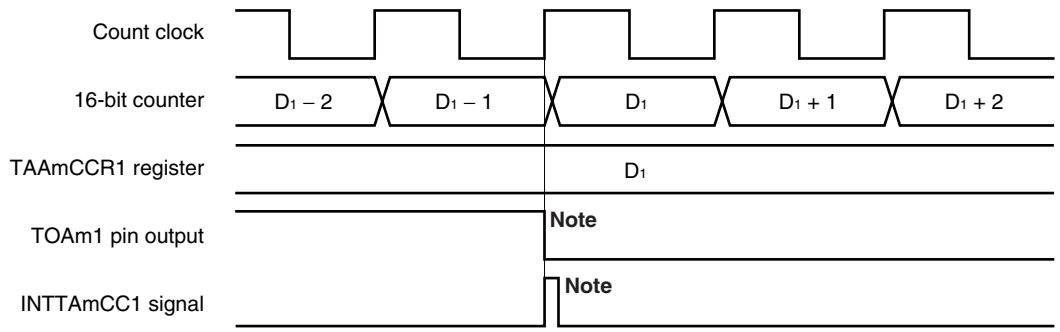
If the trigger is detected immediately before the INTTAmCC0 signal is generated, the INTTAmCC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOAm1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

(e) Generation timing of compare match interrupt request signal (INTTAmCC1)

The timing of generation of the INTTAmCC1 signal in the external trigger pulse output mode differs from the timing of INTTAmCC1 signals in other mode; the INTTAmCC1 signal is generated when the count value of the 16-bit counter matches the value of the TAAmCCR1 register.



Note The timing is actually delayed by one operating clock (fXX).

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

Usually, the INTTAmCC1 signal is generated in synchronization with the next count-up, after the count value of the 16-bit counter matches the value of the TAAmCCR1 register.

In the external trigger pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the timing of changing the output signal of the TOAm1 pin.

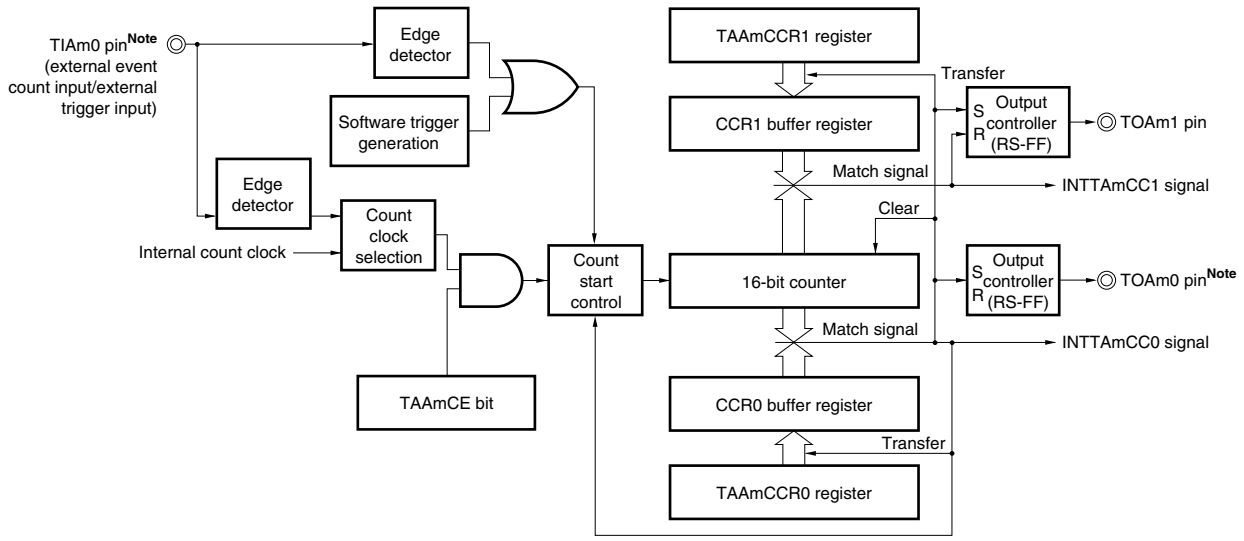
6.6.4 One-shot pulse output mode (TAAmMD2 to TAAmMD0 bits = 011)

This mode is valid only in TAA2, TAA3 (V850E/IG3 only), and TAA4.

In the one-shot pulse output mode, 16-bit timer/event counter AA waits for a trigger when the TAAmCTL0.TAAmCE bit is set to 1. When the valid edge of an external trigger input is detected, 16-bit timer/event counter AA starts counting, and outputs a one-shot pulse from the TOAm1 pin.

Instead of the external trigger input (TIAm0), a software trigger can also be generated to output the pulse. When the software trigger is used, the TOAm0 pin outputs the active level while the 16-bit counter is counting, and the inactive level when the counter is stopped (waiting for a trigger).

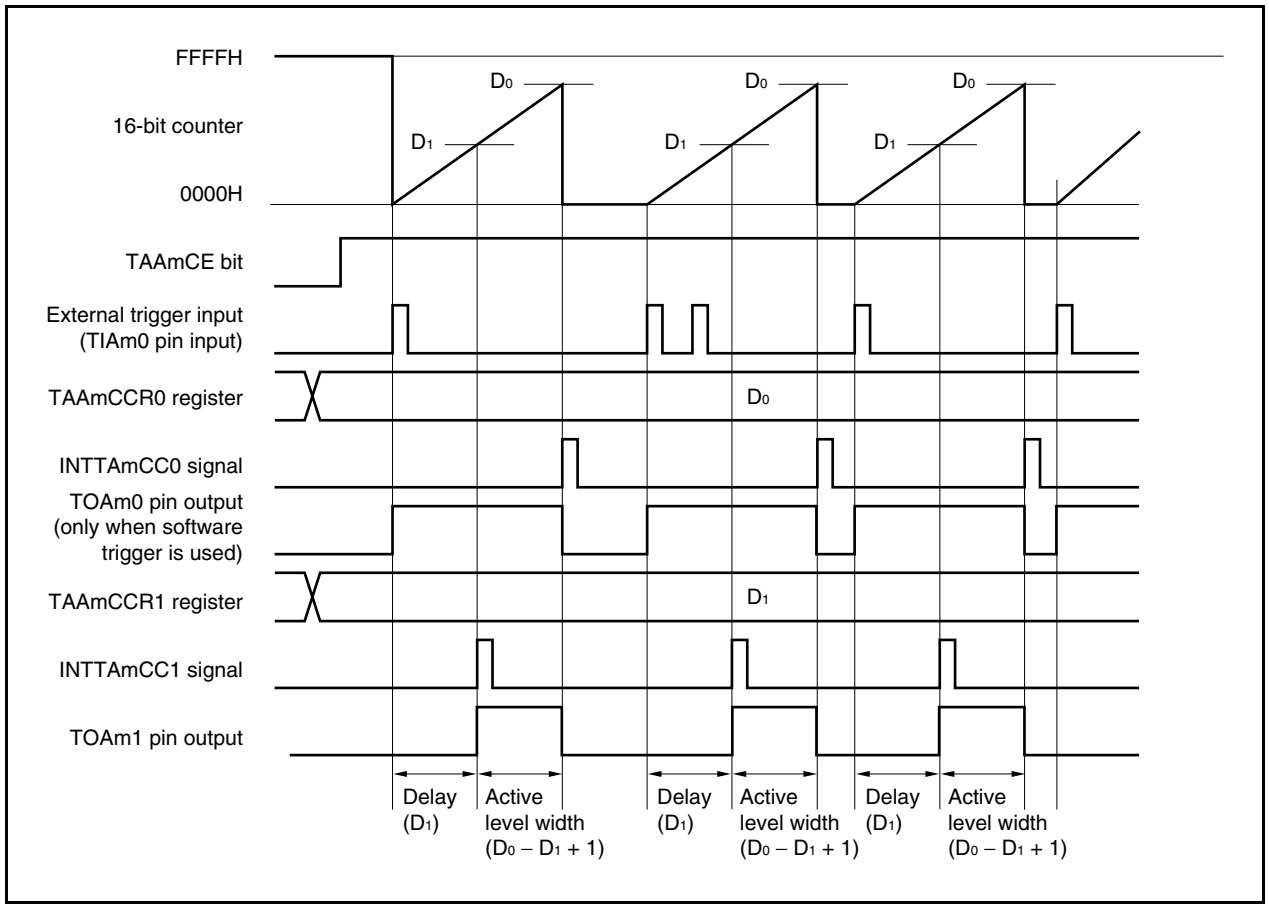
Figure 6-28. Configuration in One-Shot Pulse Output Mode



Note Because the external event count input pin (TIAm0), external trigger input pin (TIAm0), and timer output pin (TOAm0) are the same alternate-function pin, two or more functions cannot be used at the same time.

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

Figure 6-29. Basic Timing in One-Shot Pulse Output Mode



When the TAAmCE bit is set to 1, 16-bit timer/event counter AA waits for a trigger. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a one-shot pulse from the TOAm1 pin. After the one-shot pulse is output, the 16-bit counter is cleared to 0000H, stops counting, and waits for a trigger. When the trigger is generated again, the 16-bit counter starts counting from 0000H. If a trigger is generated again while the one-shot pulse is being output, it is ignored.

The output delay period and active level width of the one-shot pulse can be calculated as follows.

Output delay period = (Set value of TAAmCCR1 register) × Count clock cycle

Active level width = (Set value of TAAmCCR0 register − Set value of TAAmCCR1 register + 1) × Count clock cycle

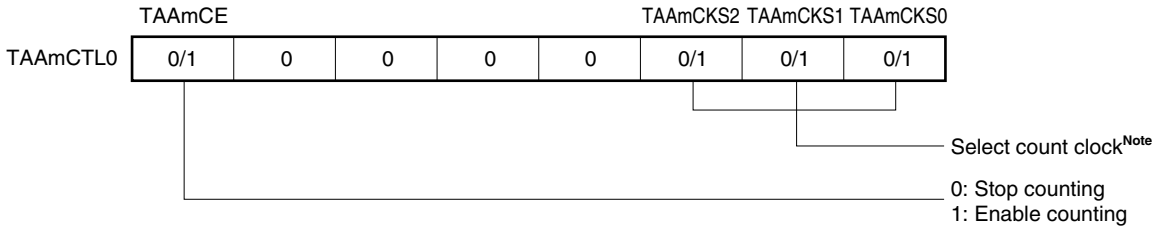
The compare match interrupt request signal (INTTAmCC0) is generated when the 16-bit counter counts after its count value matches the value of the CCR0 buffer register. The compare match interrupt request signal (INTTAmCC1) is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The valid edge of an external trigger input (TIAm0 pin) or setting the software trigger (TAAmCTL1.TAAnEST bit) to 1 is used as the trigger.

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

Figure 6-30. Setting of Registers in One-Shot Pulse Output Mode (1/2)

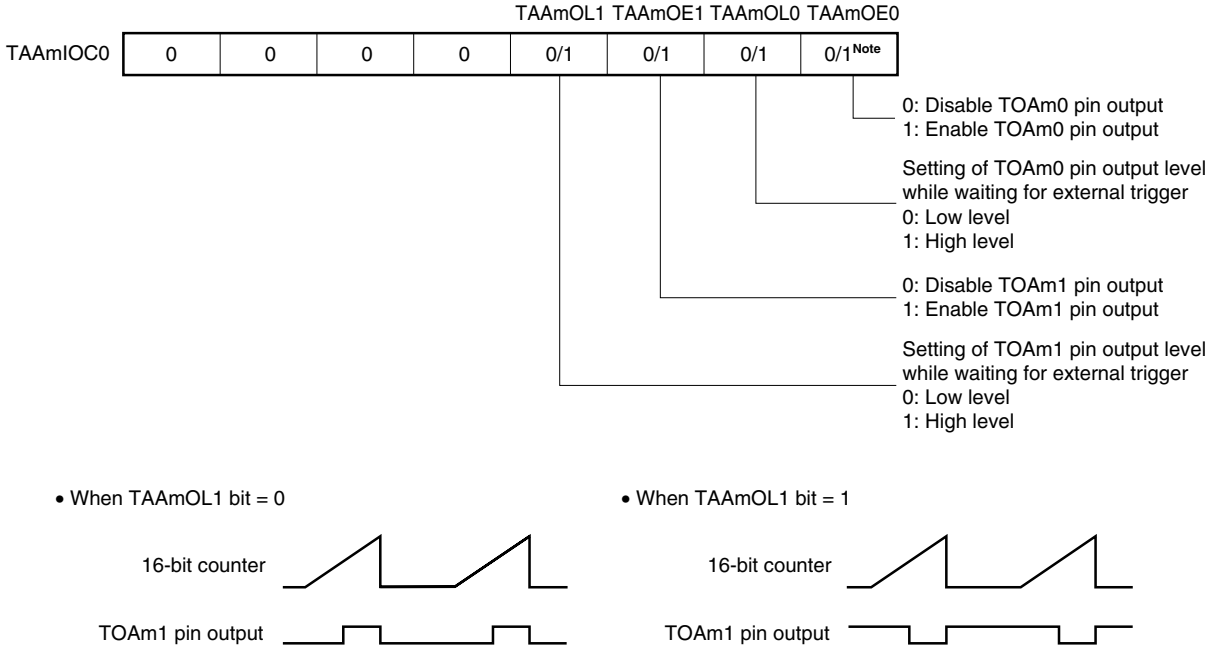
(a) TAAm control register 0 (TAAmCTL0)



Note The setting is invalid when the TAAmCTL1.TAAmEEE bit = 1.

(b) TAAm control register 1 (TAAmCTL1)

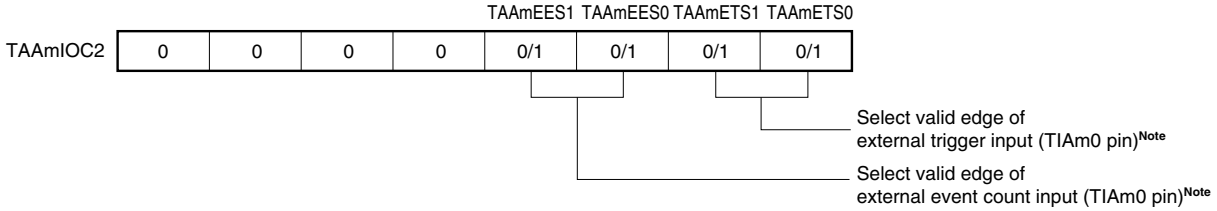
(c) TAAm I/O control register 0 (TAAmIOC0)



Note Clear this bit to 0 when the TOAm0 pin is not used in the one-shot pulse output mode.

Figure 6-30. Setting of Registers in One-Shot Pulse Output Mode (2/2)

(d) TAAm I/O control register 2 (TAAmIOC2)



Note Set the valid edge selection of the unused alternate external input signals to "No edge detection".

(e) TAAm counter read buffer register (TAAmCNT)

The value of the 16-bit counter can be read by reading the TAAmCNT register.

(f) TAAm capture/compare registers 0 and 1 (TAAmCCR0 and TAAmCCR1)

If D_0 is set to the TAAmCCR0 register and D_1 to the TAAmCCR1 register, the active level width and output delay period of the one-shot pulse are as follows.

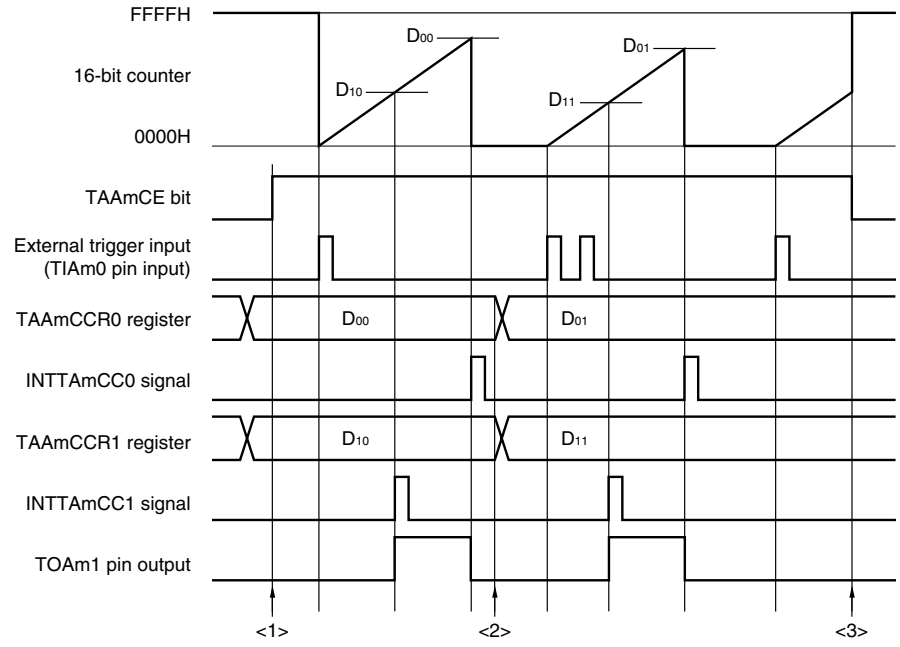
Active level width = $(D_0 - D_1 + 1) \times$ Count clock cycle
Output delay period = $D_1 \times$ Count clock cycle

Remarks 1. TAAm I/O control register 1 (TAAmIOC1) and TAAm option register 0 (TAAmOPT0) are not used in the one-shot pulse output mode.

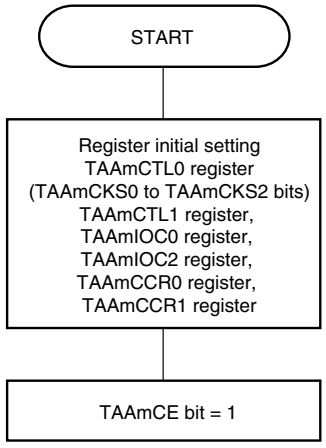
2. V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

(1) Operation flow in one-shot pulse output mode

Figure 6-31. Software Processing Flow in One-Shot Pulse Output Mode



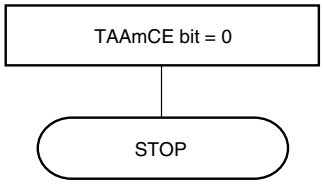
<1> Count operation start flow



Initial setting of these registers is performed before setting the TAAmCE bit to 1.

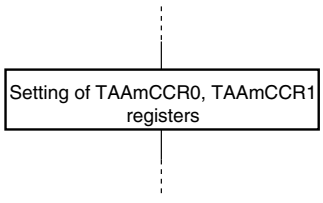
The TAAmCKS0 to TAAmCKS2 bits can be set at the same time when counting has been started (TAAmCE bit = 1).
Trigger wait status

<3> Count operation stop flow



Count operation is stopped

<2> TAAmCCR0, TAAmCCR1 register setting change flow



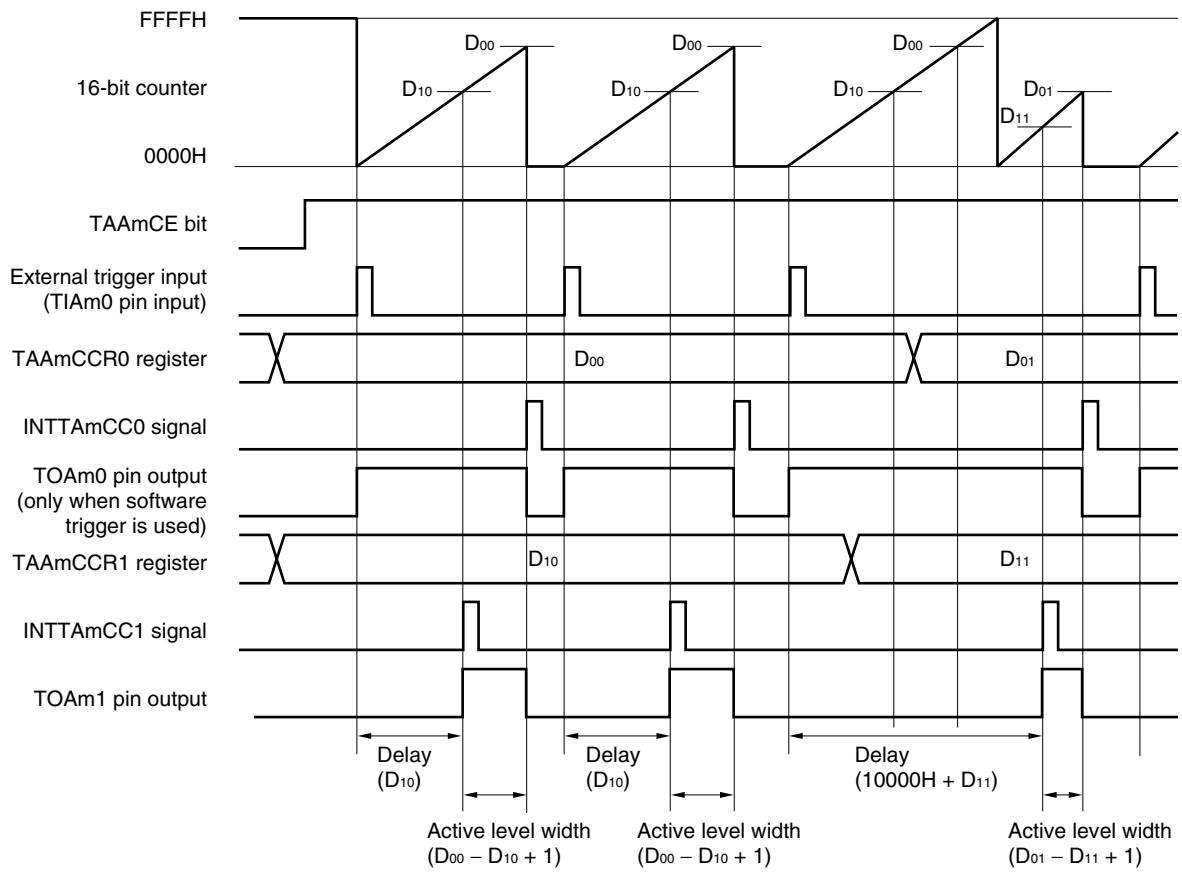
As rewriting the TAAmCCRa register immediately forwards to the CCRa buffer register, rewriting immediately after the generation of the INTTAmCC0 signal is recommended.

Remark V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

(2) Operation timing in one-shot pulse output mode

(a) Note on rewriting TAAmCCRa register

If the value of the TAAmCCRa register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When an overflow may occur, stop counting and then change the set value.



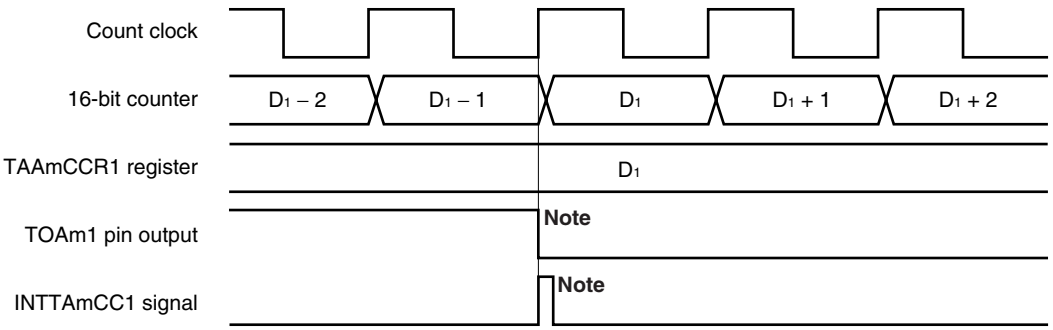
When the TAAmCCR0 register is rewritten from D_{00} to D_{01} and the TAAmCCR1 register from D_{10} to D_{11} where $D_{00} > D_{01}$ and $D_{10} > D_{11}$, if the TAAmCCR1 register is rewritten when the count value of the 16-bit counter is greater than D_{11} and less than D_{10} and if the TAAmCCR0 register is rewritten when the count value is greater than D_{01} and less than D_{00}, each set value is reflected as soon as the register has been rewritten and compared with the count value. The counter counts up to FFFFH and then counts up again from 0000H. When the count value matches D_{11}, the counter generates the INTTAmCC1 signal and asserts the TOAm1 pin output. When the count value matches D_{01}, the counter generates the INTTAmCC0 signal, deasserts the TOAm1 pin output, and stops counting.

Therefore, the counter may output a pulse with a delay period or active period different from that of the one-shot pulse that is originally expected.

Remark V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

(b) Generation timing of compare match interrupt request signal (INTTAmCC1)

The generation timing of the INTTAmCC1 signal in the one-shot pulse output mode is different from INTTAmCC1 signals; the INTTAmCC1 signal is generated when the count value of the 16-bit counter matches the value of the TAAmCCR1 register.



Note The timing is actually delayed by one operating clock (fxx).

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

Usually, the INTTAmCC1 signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TAAmCCR1 register.

In the one-shot pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the TOAm1 pin.

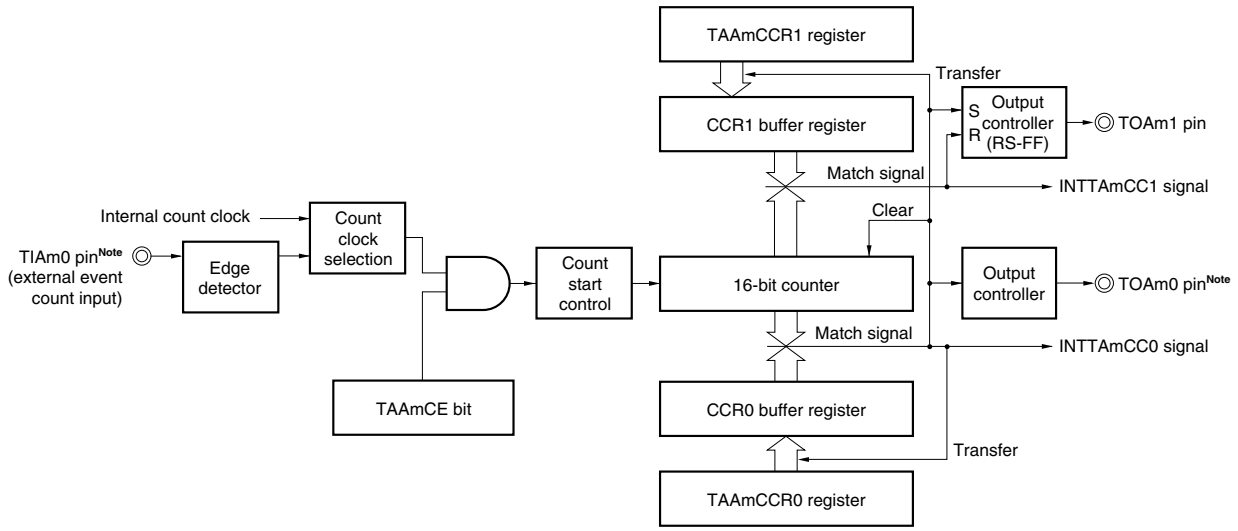
6.6.5 PWM output mode (TAAmMD2 to TAAmMD0 bits = 100)

This mode is valid only in TAA2, TAA3 (V850E/IG3 only), and TAA4.

In the PWM output mode, a PWM waveform is output from the TOAm1 pin when the TAAmCTL0.TAAmCE bit is set to 1.

In addition, a PWM waveform with a duty factor of 50% with the set value of the TAAmCCR0 register + 1 as half its cycle is output from the TOAm0 pin.

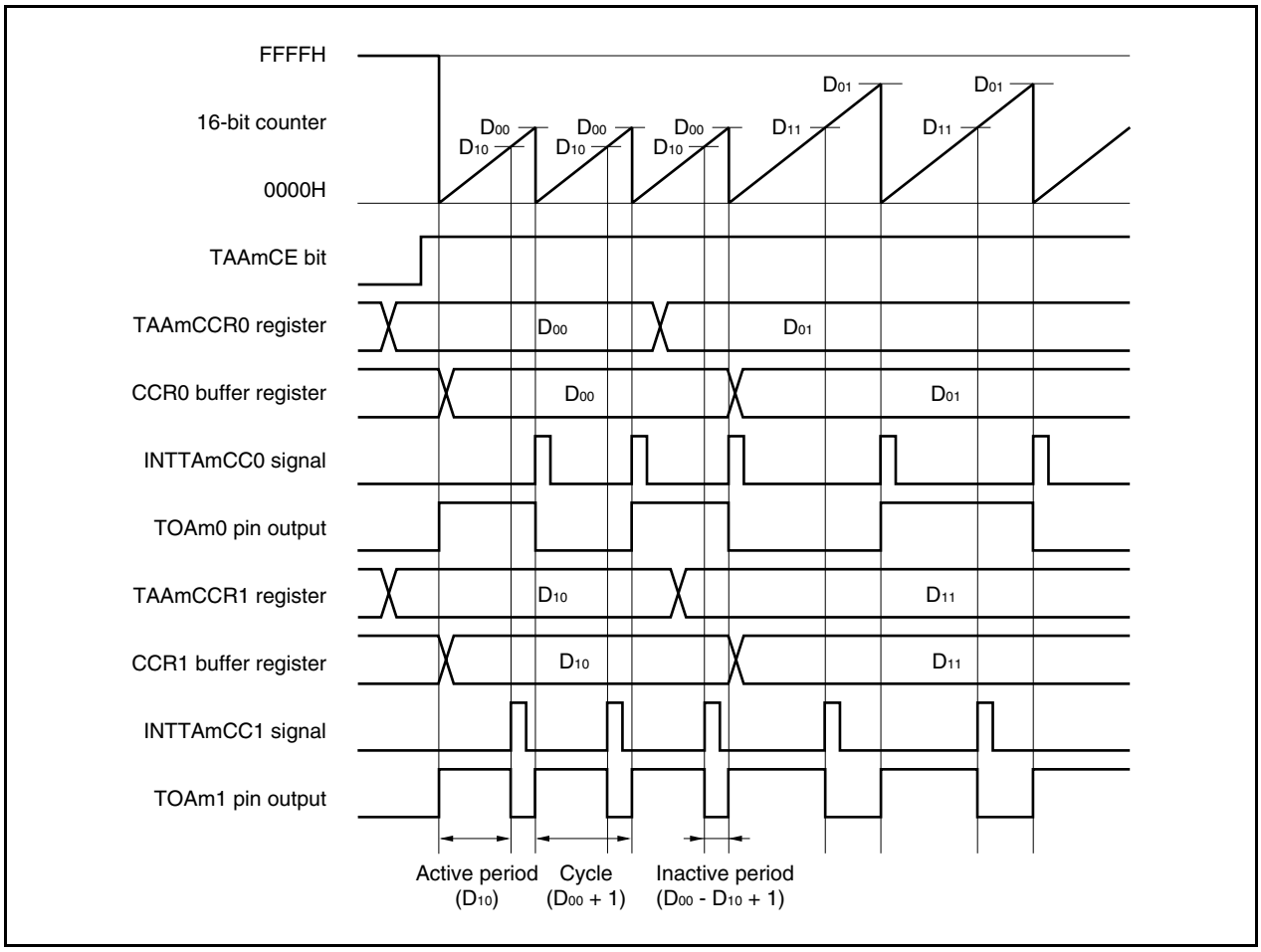
Figure 6-32. Configuration in PWM Output Mode



Note Because the external event count input pin (TIAm0) and timer output pin (TOAm0) are the same alternate-function pin, two or more functions cannot be used at the same time.

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

Figure 6-33. Basic Timing in PWM Output Mode



When the TAAmCE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a PWM waveform from the TOAm1 pin.

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TAAmCCR1 register) × Count clock cycle
Cycle = (Set value of TAAmCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TAAmCCR1 register)/(Set value of TAAmCCR0 register + 1)

The PWM waveform can be changed by rewriting the TAAmCCRa register while the counter is operating. The newly written value is reflected when the count value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

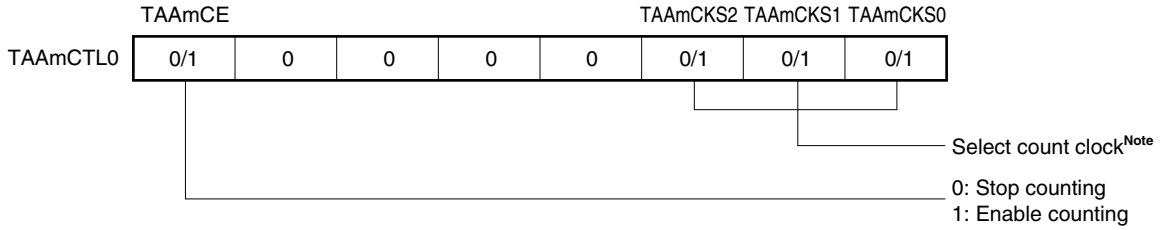
The compare match interrupt request signal INTTAmCC0 is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal INTTAmCC1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TAAmCCRa register is transferred to the CCRa buffer register when the count value of the 16-bit counter matches the value of the CCRa buffer register and the 16-bit counter is cleared to 0000H.

Remark V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

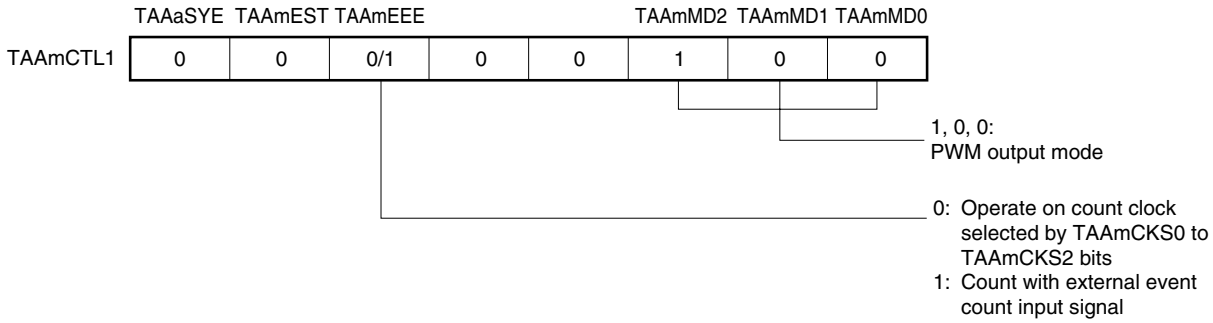
Figure 6-34. Setting of Registers in PWM Output Mode (1/2)

(a) TAAm control register 0 (TAAmCTL0)

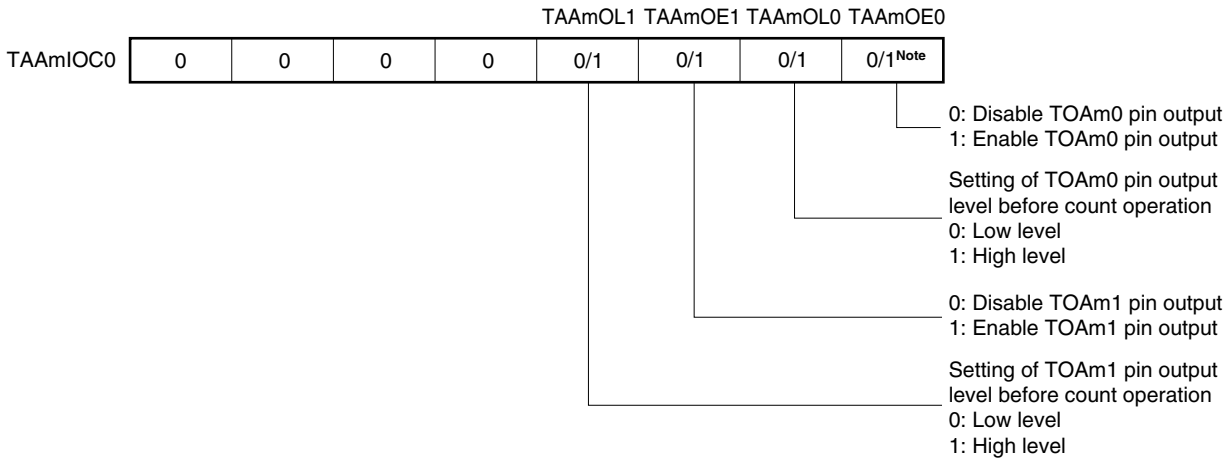


Note The setting is invalid when the TAAmCTL1.TAAmEEE bit = 1.

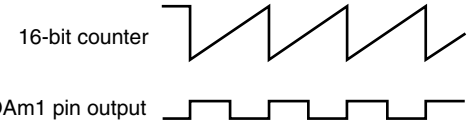
(b) TAAm control register 1 (TAAmCTL1)



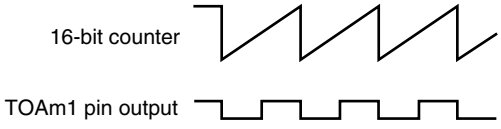
(c) TAAm I/O control register 0 (TAAmIOC0)



- When TAAmOL1 bit = 0



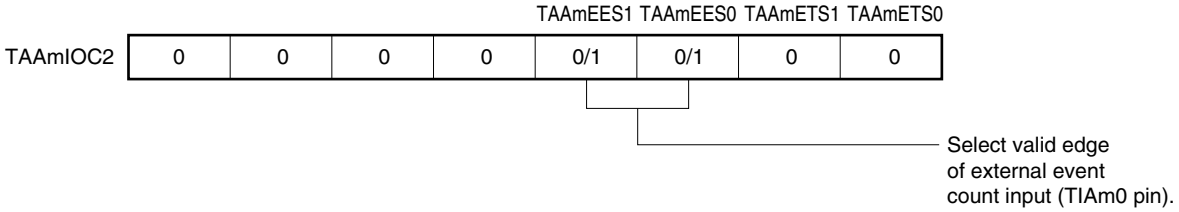
- When TAAmOL1 bit = 1



Note Clear this bit to 0 when the TOAm0 pin is not used in the PWM output mode.

Figure 6-34. Register Setting in PWM Output Mode (2/2)

(d) TAAm I/O control register 2 (TAAmIOC2)



(e) TAAm counter read buffer register (TAAmCNT)

The value of the 16-bit counter can be read by reading the TAAmCNT register.

(f) TAAm capture/compare registers 0 and 1 (TAAmCCR0 and TAAmCCR1)

If D_0 is set to the TAAmCCR0 register and D_1 to the TAAmCCR1 register, the cycle and active level of the PWM waveform are as follows.

Cycle = $(D_0 + 1) \times$ Count clock cycle

Active level width = $D_1 \times$ Count clock cycle

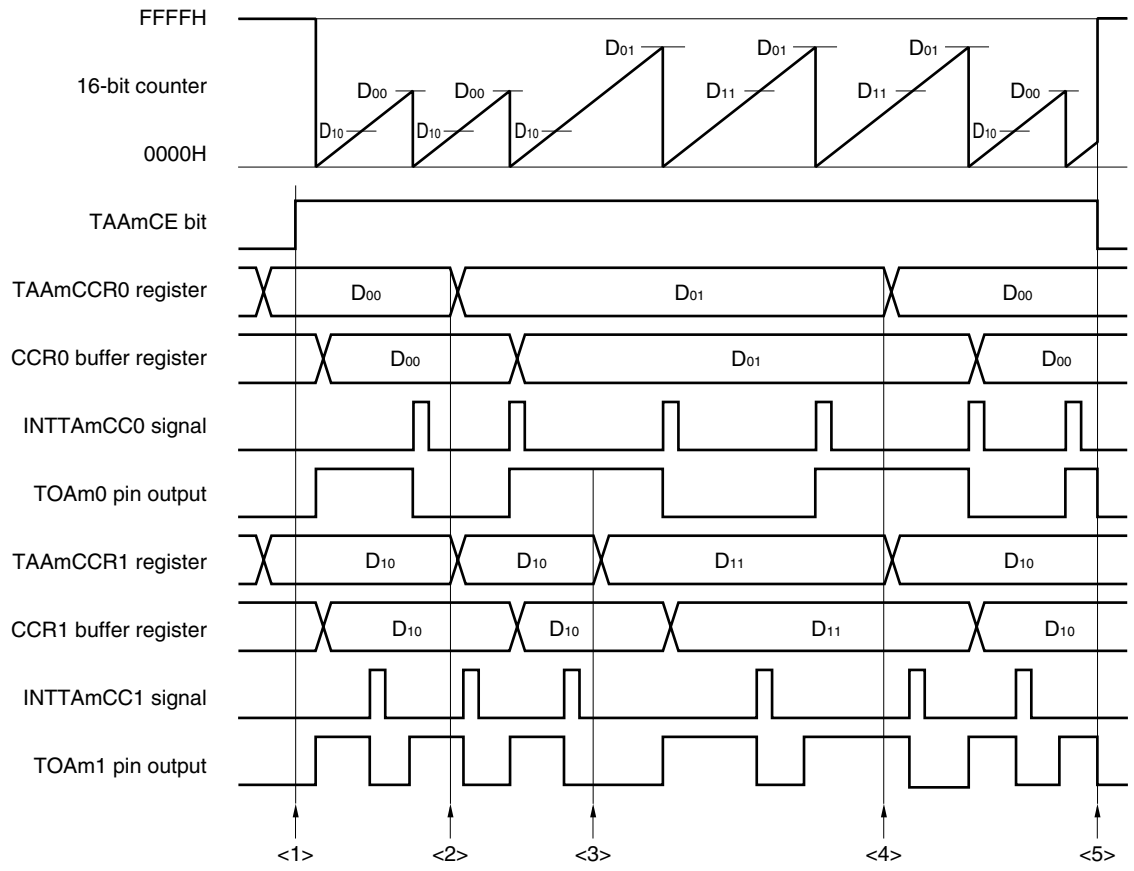
Remarks 1. TAAm I/O control register 1 (TAAmIOC1) and TAAm option register 0 (TAAmOPT0) are not used in the PWM output mode.

2. V850E/IF3: m = 2, 4, a = 0, 1

V850E/IG3: m = 2 to 4, a = 0, 1

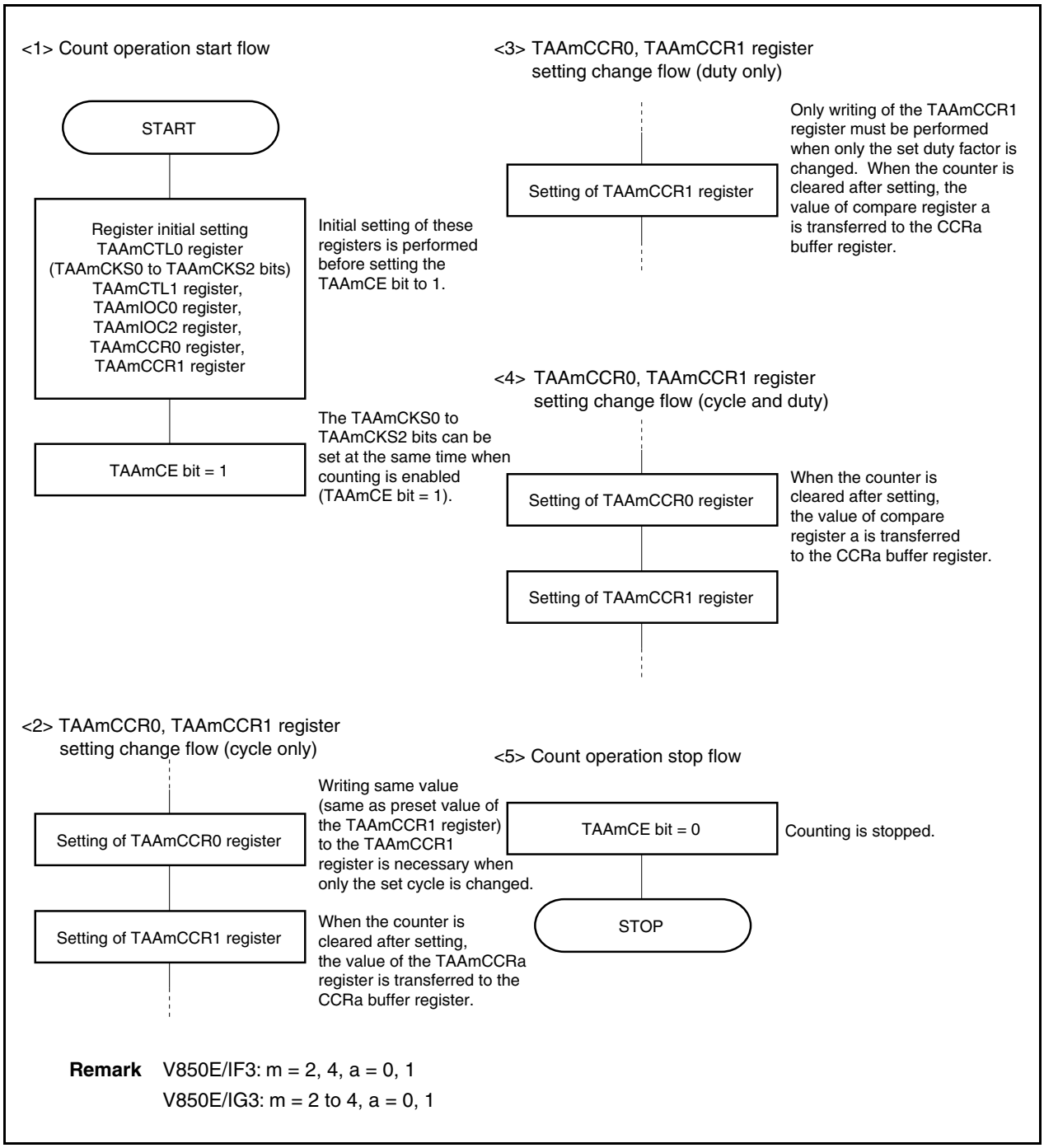
(1) Operation flow in PWM output mode

Figure 6-35. Software Processing Flow in PWM Output Mode (1/2)



Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

Figure 6-35. Software Processing Flow in PWM Output Mode (2/2)



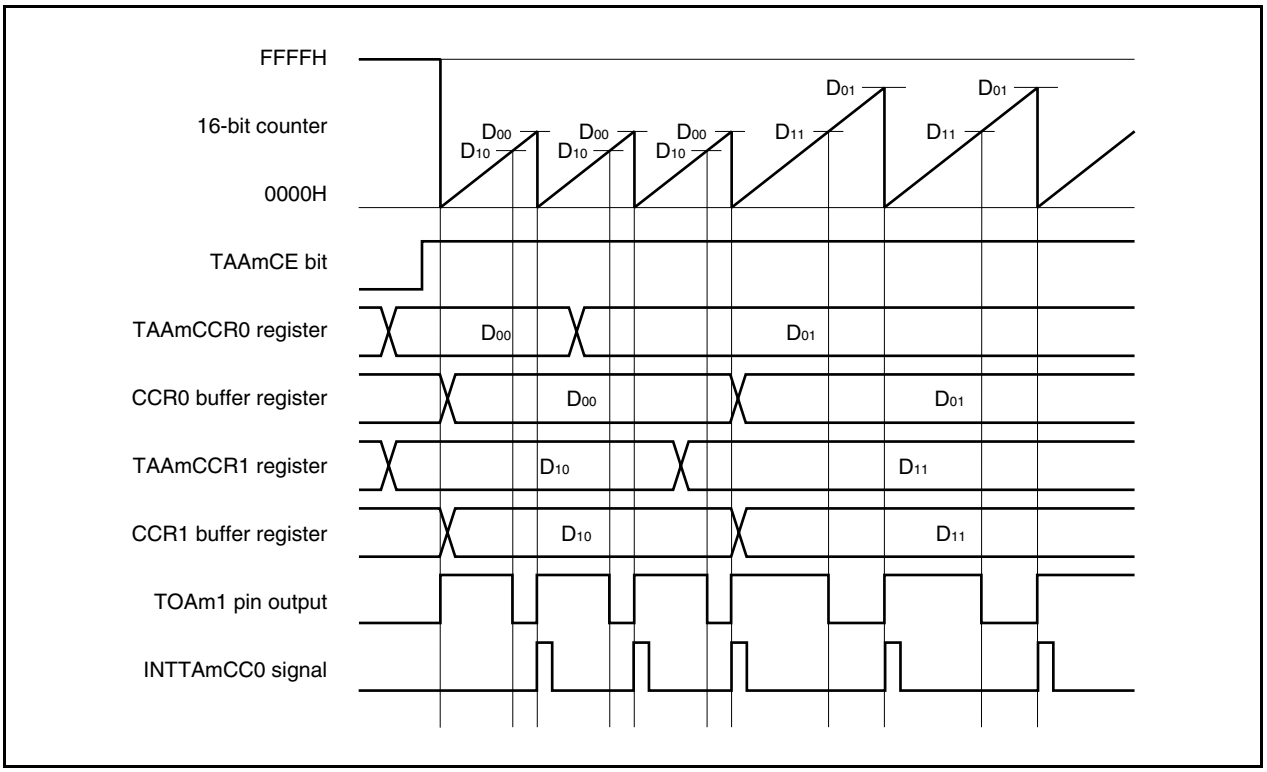
Remark V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

(2) PWM output mode operation timing

(a) Changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TAAmCCR1 register last.

Rewrite the TAAmCCRa register after writing the TAAmCCR1 register after the INTTAmCC1 signal is detected.



To transfer data from the TAAmCCRa register to the CCRa buffer register, the TAAmCCR1 register must be written.

To change both the cycle and active level width of the PWM waveform at this time, first set the cycle to the TAAmCCR0 register and then set the active level width to the TAAmCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TAAmCCR0 register, and then write the same value (same as preset value of the TAAmCCR1 register) to the TAAmCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TAAmCCR1 register has to be set.

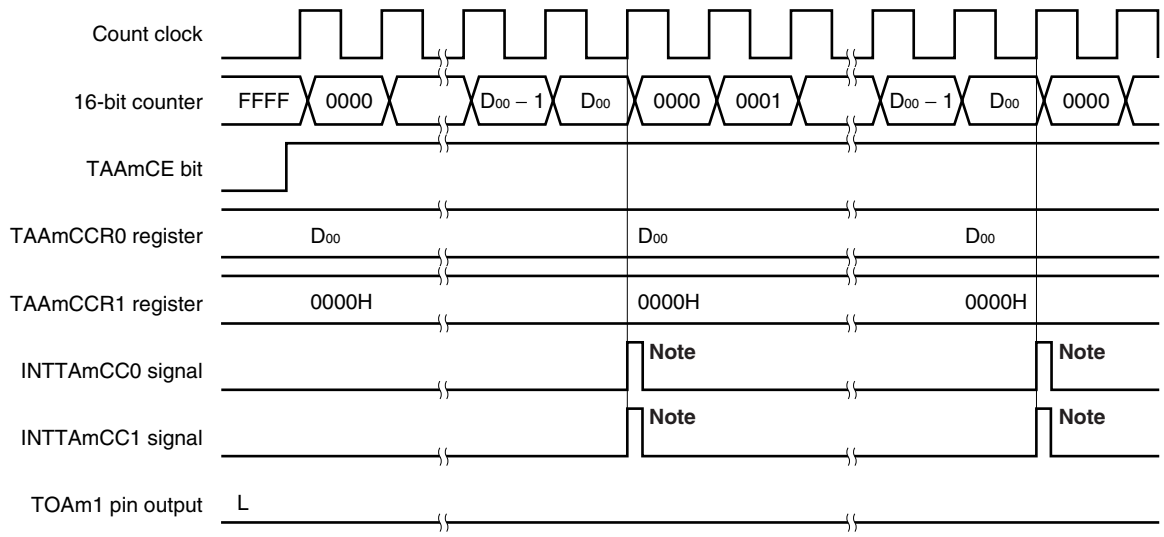
After data is written to the TAAmCCR1 register, the value written to the TAAmCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TAAmCCR0 or TAAmCCR1 register again after writing the TAAmCCR1 register once, do so after the INTTAmCC0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TAAmCCRa register to the CCRa buffer register conflicts with writing the TAAmCCRa register.

Remark V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

(b) 0%/100% output of PWM waveform

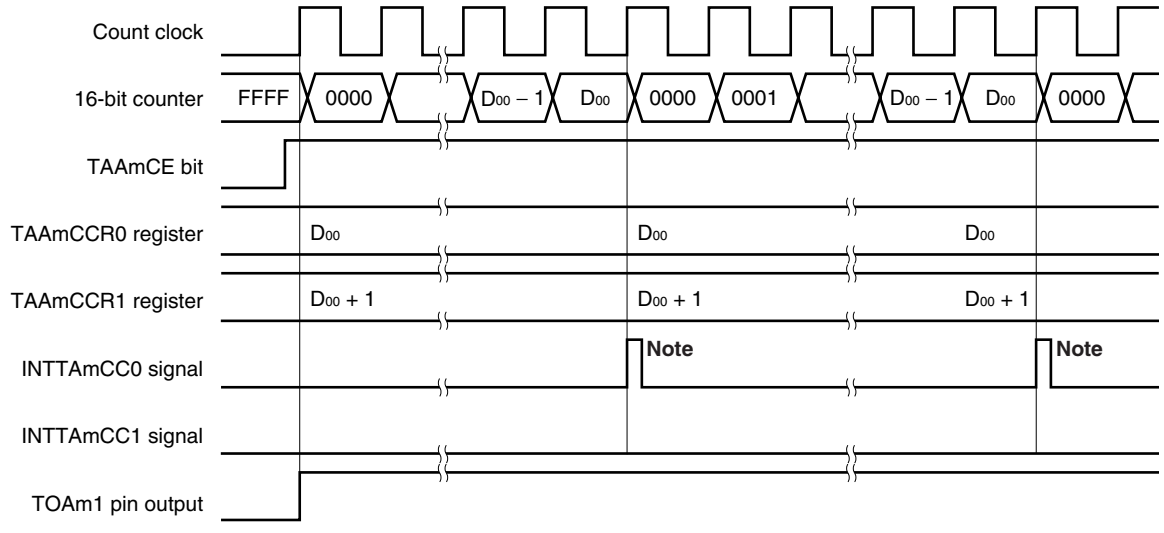
To output a 0% waveform, set the TAAmCCR1 register to 0000H. The 16-bit counter is cleared to 0000H and the INTTAmCC0 and INTTAmCC1 signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.



Note The timing is actually delayed by one operating clock (fxx).

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

To output a 100% waveform, set a value of (set value of TAAmCCR0 register + 1) to the TAAmCCR1 register. If the set value of the TAAmCCR0 register is FFFFH, 100% output cannot be produced.

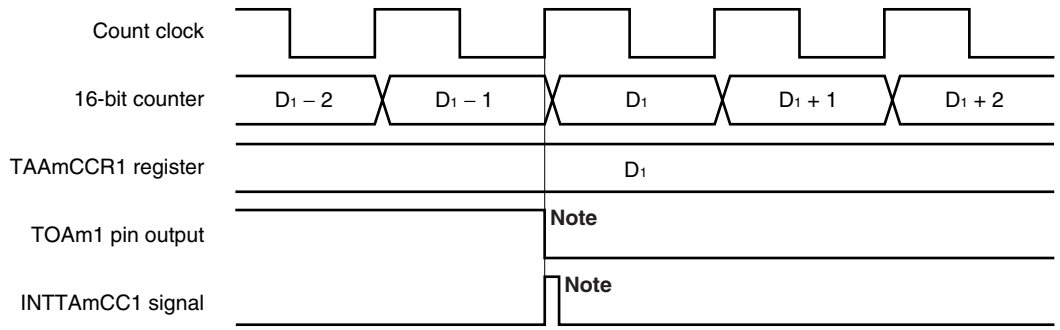


Note The timing is actually delayed by one operating clock (fxx).

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

(c) Generation timing of compare match interrupt request signal (INTTAmCC1)

The timing of generation of the INTTAmCC1 signal in the PWM output mode differs from the timing of INTTAmCC1 signals; the INTTAmCC1 signal is generated when the count value of the 16-bit counter matches the value of the TAAmCCR1 register.



Note The timing is actually delayed by one operating clock (fxx).

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

Usually, the INTTAmCC1 signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TAAmCCR1 register.

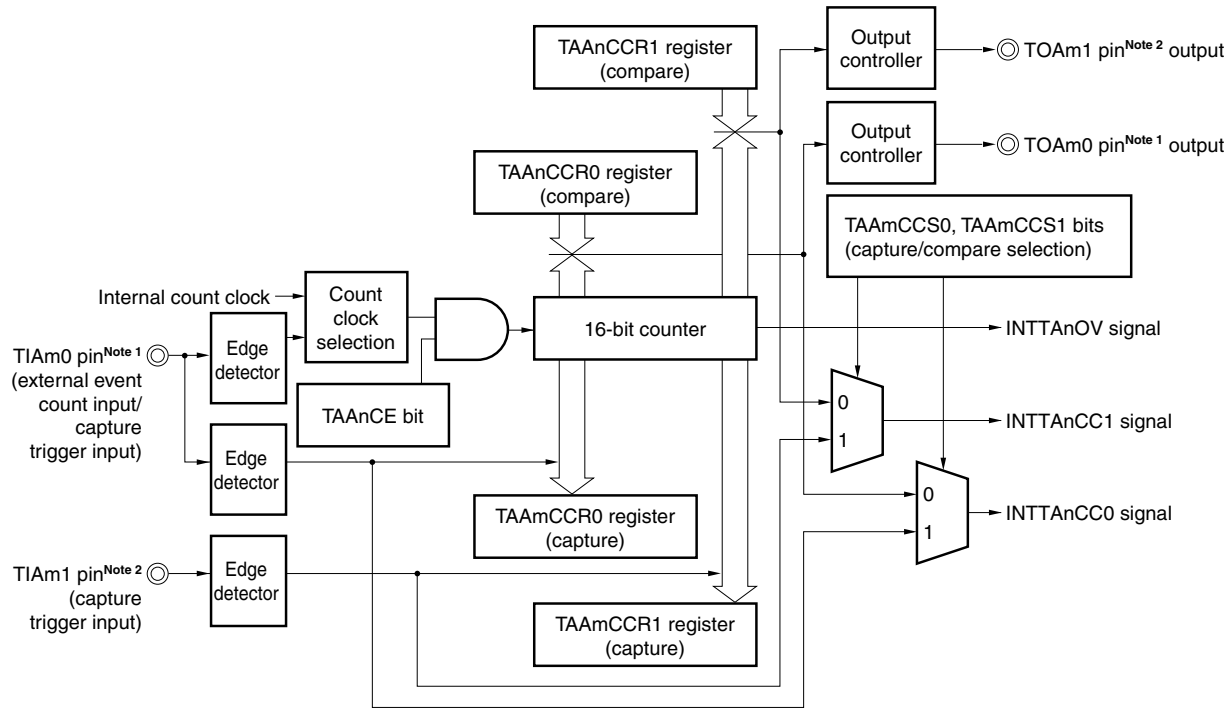
In the PWM output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the output signal of the TOAm1 pin.

6.6.6 Free-running timer mode (TAAnMD2 to TAAnMD0 bits = 101)

The compare function is valid for all of TAA0 to TAA4. The capture function is valid only for TAA2, TAA3 (V850E/IG3 only), and TAA4.

In the free-running timer mode, 16-bit timer/event counter AA starts counting when the TAAnCTL0.TAAnCE bit is set to 1. At this time, the TAAmCCR0 and TAAmCCR1 registers can be used as compare registers or capture registers, depending on the setting of the TAAmOPT0.TAAmCCS0 and TAAmOPT0.TAAmCCS1 bits.

Figure 6-36. Configuration in Free-Running Timer Mode



Notes 1. Because the external event count input pin (TIAm0), capture trigger input pin (TIAm0), and timer output pin (TOAm0) are the same alternate-function pin, two or more functions cannot be used at the same time.

2. Because the capture trigger input pin (TIAm1) and timer output pin (TOAm1) are the same alternate-function pin, two or more functions cannot be used at the same time.

Remark V850E/IF3: n = 0 to 4, m = 2, 4
V850E/IG3: n = 0 to 4, m = 2 to 4

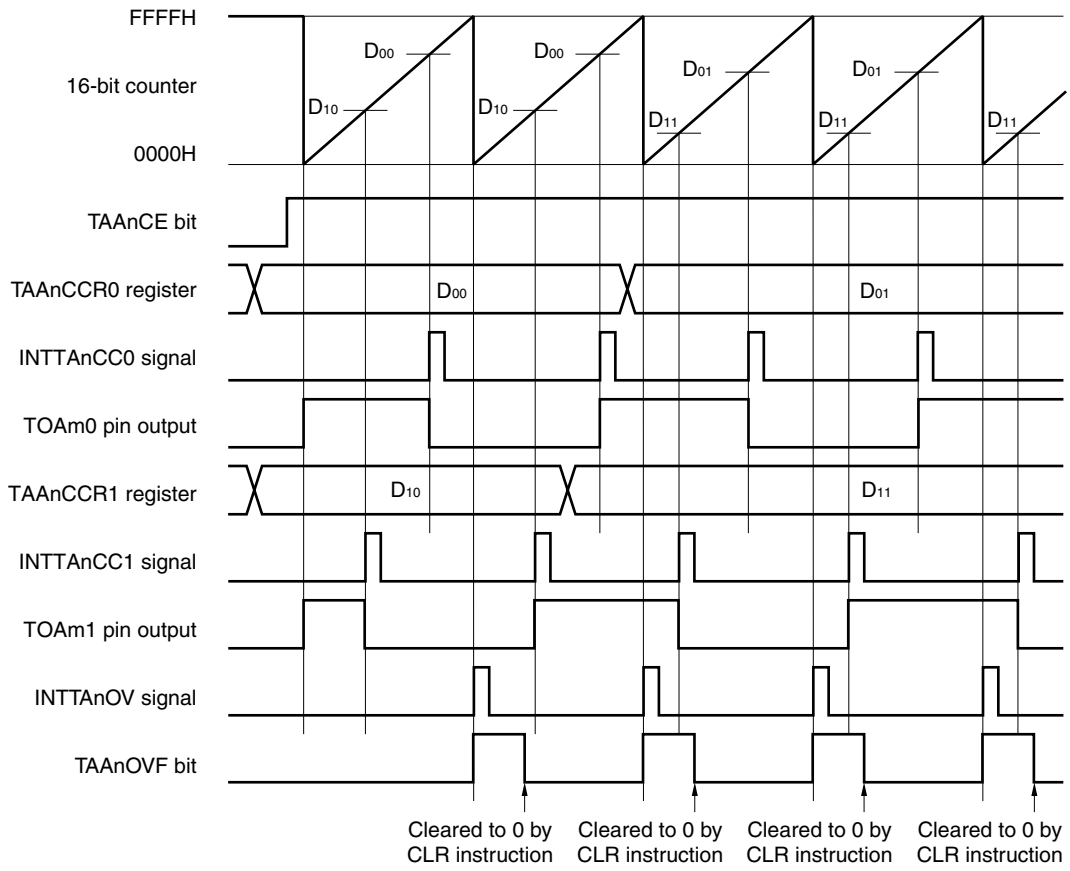
- Compare operation

  When the TAAnCE bit is set to 1, 16-bit timer/event counter AA starts counting, and the output signal of the TOAma pin is inverted. When the count value of the 16-bit counter later matches the set value of the TAAnCCRa register, a compare match interrupt request signal (INTTAnCCa) is generated, and the output signal of the TOAma pin is inverted.

  The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTAnOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TAAnOPT0.TAAnOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

  The TAAnCCRa register can be rewritten while the counter is operating. If it is rewritten, the new value is reflected at that time by anytime write, and compared with the count value.

Figure 6-37. Basic Timing in Free-Running Timer Mode (Compare Function)



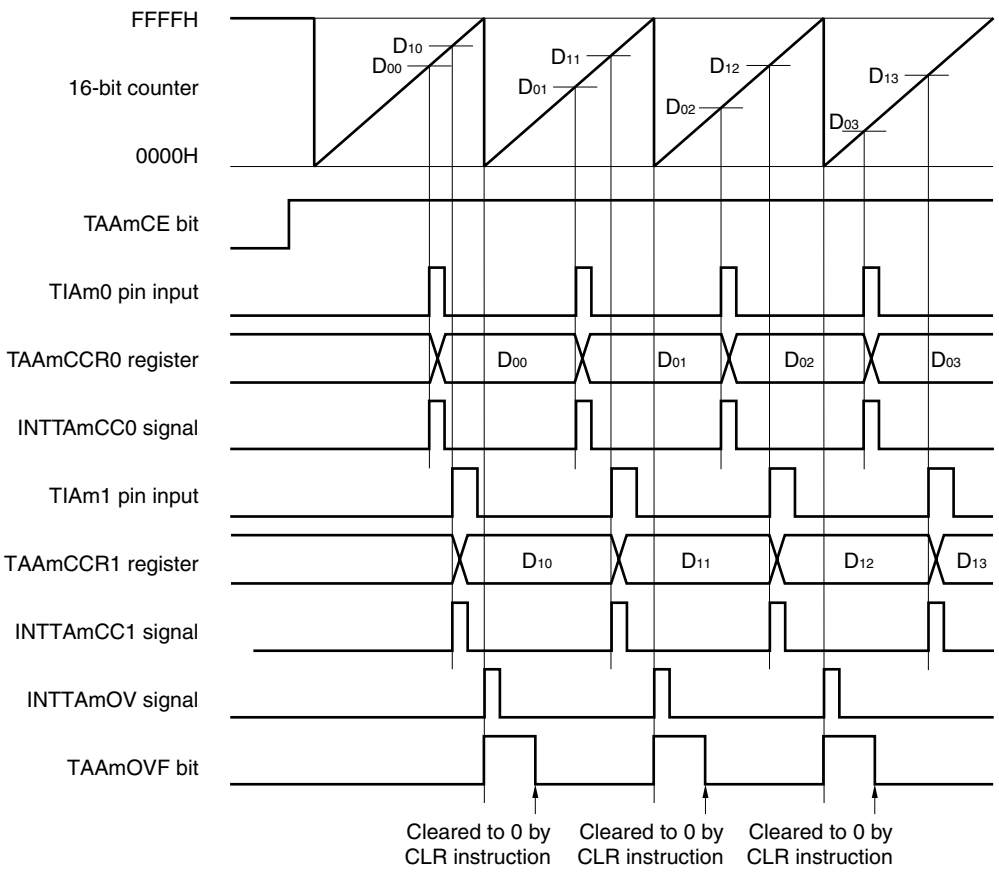
Remark V850E/IF3: n = 0 to 4, m = 2, 4, a = 0, 1
V850E/IG3: n = 0 to 4, m = 2 to 4, a = 0, 1

- Capture operation

  When the TAAmCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIAma pin is detected, the count value of the 16-bit counter is stored in the TAAmCCRa register, and a capture interrupt request signal (INTTAmCCa) is generated.

  The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTAmOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TAAmOPT0.TAAmOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

Figure 6-38. Basic Timing in Free-Running Timer Mode (Capture Function)



Remark V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

Figure 6-39. Register Setting in Free-Running Timer Mode (1/2)

(a) TAAn control register 0 (TAAnCTL0)

	TAAnCE					TAAnCKS2	TAAnCKS1	TAAnCKS0
TAAnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock[Note]

0: Stop counting
1: Enable counting

Note The setting is invalid when the TAAmCTL1.TAAmEEE bit = 1

(b) TAAn control register 1 (TAAnCTL1)

	TAAaSYE	TAAmEST	TAAmEEE			TAAnMD2	TAAnMD1	TAAnMD0
TAAnCTL1	0	0	0/1	0	0	1	0	1

1, 0, 1:
Free-running timer mode

0: Operate with count clock selected by TAAmCKS0 to TAAmCKS2 bits
1: Count on external event count input signal

(c) TAAm I/O control register 0 (TAAmIOC0)

					TAAmOL1	TAAmOE1	TAAmOL0	TAAmOE0
TAAmIOC0	0	0	0	0	0/1	0/1	0/1	0/1

0: Disable TOAm0 pin output
1: Enable TOAm0 pin output

Setting of TOAm0 pin output level before count operation
0: Low level
1: High level

0: Disable TOAm1 pin output
1: Enable TOAm1 pin output

Setting of TOAm1 pin output level before count operation
0: Low level
1: High level

(d) TAAm I/O control register 1 (TAAmIOC1)

					TAAmIS3	TAAmIS2	TAAmIS1	TAAmIS0
TAAmIOC1	0	0	0	0	0/1	0/1	0/1	0/1

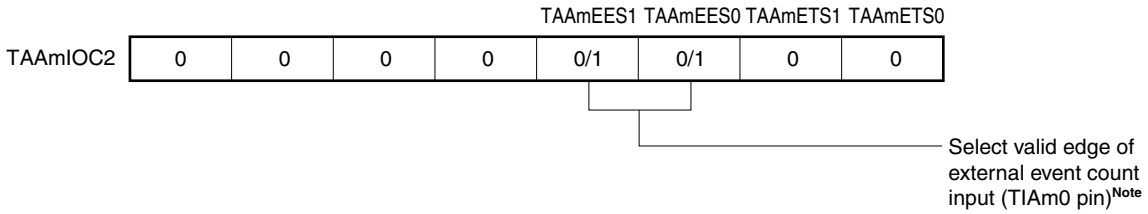
Select valid edge of TIAm0 pin input[Note]

Select valid edge of TIAm1 pin input[Note]

Note Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

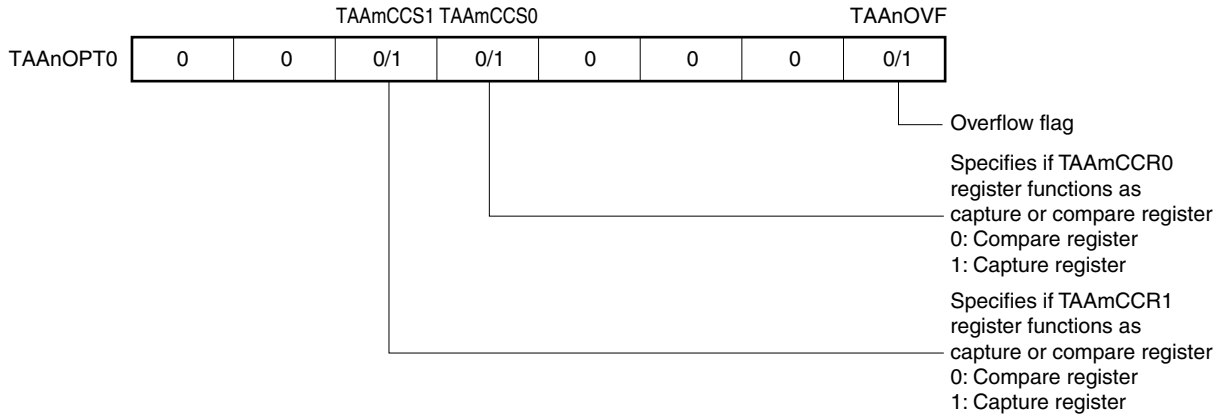
Figure 6-39. Register Setting in Free-Running Timer Mode (2/2)

(e) TAAm I/O control register 2 (TAAmIOC2)



Note Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

(f) TAAn option register 0 (TAAnOPT0)



(g) TAAn counter read buffer register (TAAnCNT)

The value of the 16-bit counter can be read by reading the TAAnCNT register.

(h) TAAn capture/compare registers 0 and 1 (TAAnCCR0 and TAAnCCR1)

These registers function as capture registers or compare registers depending on the setting of the TAAmOPT0.TAAmCCSa bit.

When the registers function as capture registers, they store the count value of the 16-bit counter when the valid edge input to the TIAma pin is detected.

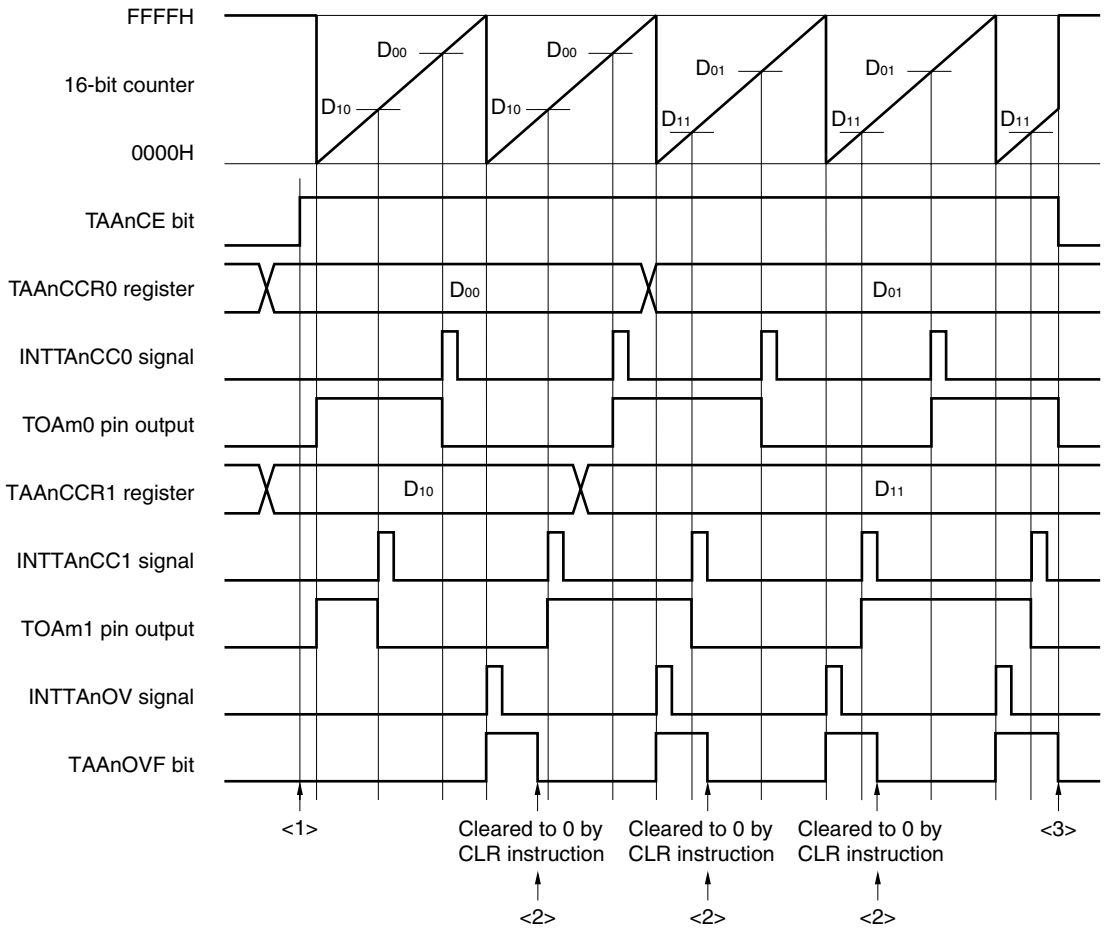
When the registers function as compare registers and when D_a is set to the TAAnCCRa register, the INTTAnCCa signal is generated when the counter reaches (D_a + 1), and the output signals of the TOAm0 and TOAm1 pins are inverted.

Remark V850E/IF3: n = 0 to 4, m = 2, 4, a = 0, 1
V850E/IG3: n = 0 to 4, m = 2 to 4, a = 0, 1

(1) Operation flow in free-running timer mode

(a) When using capture/compare register as compare register

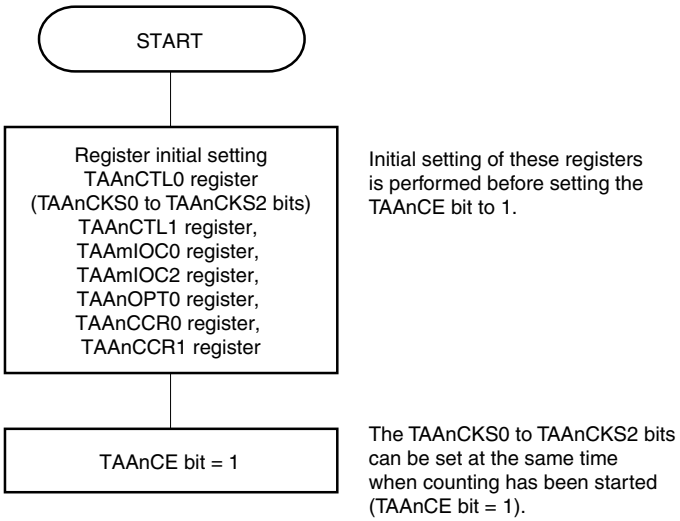
Figure 6-40. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)



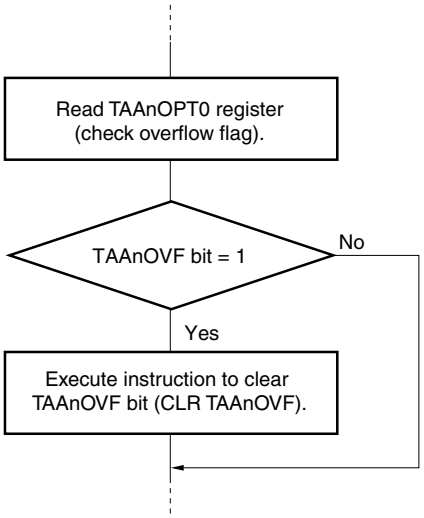
Remark n = 0 to 4

Figure 6-40. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)

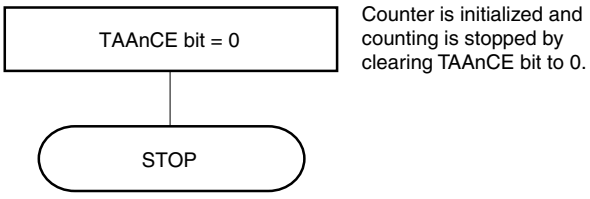
<1> Count operation start flow



<2> Overflow flag clear flow



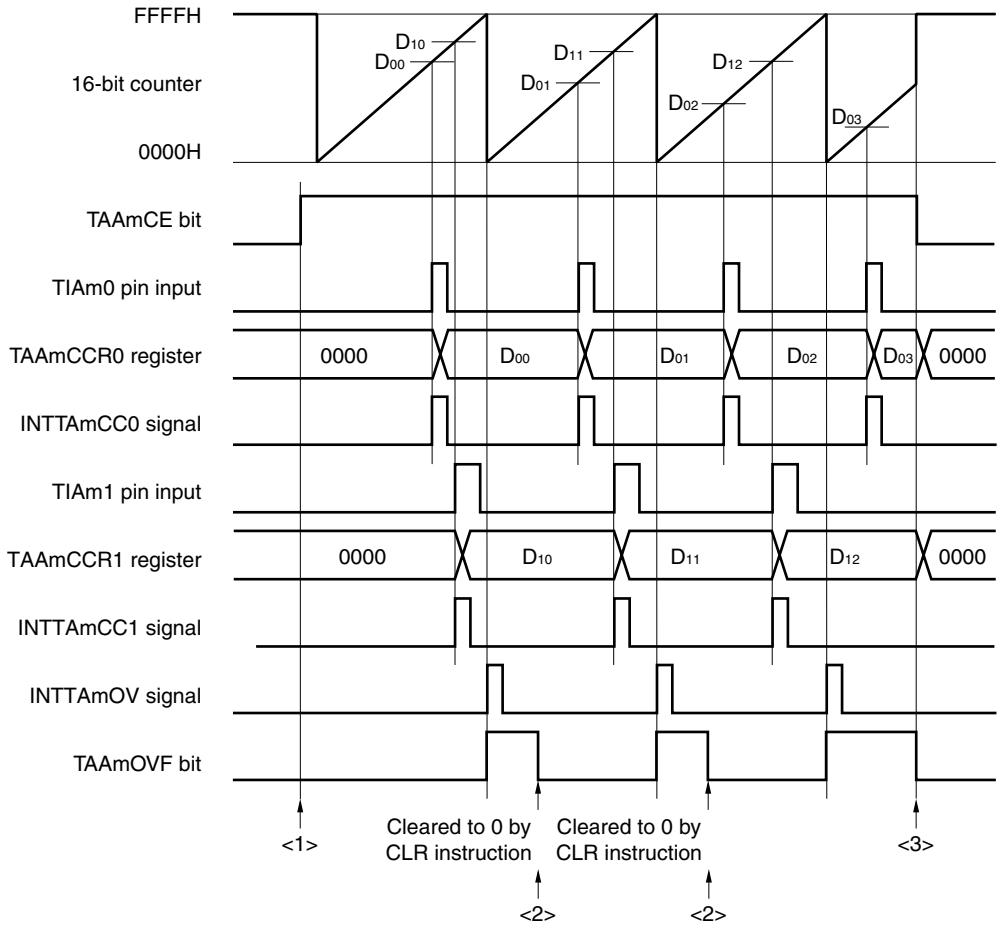
<3> Count operation stop flow



Remark V850E/IF3: n = 0 to 4, m = 2, 4
V850E/IG3: n = 0 to 4, m = 2 to 4

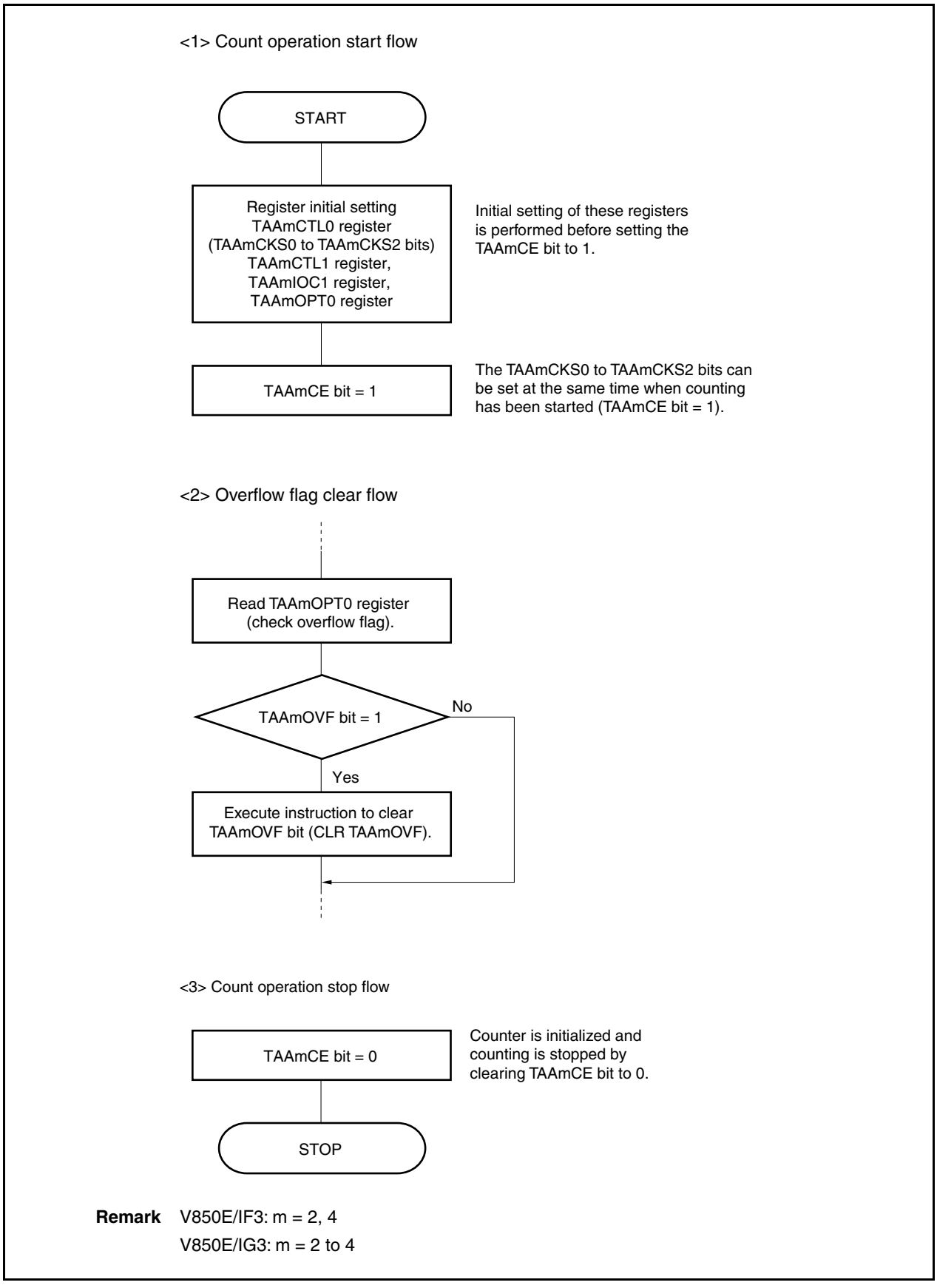
(b) When using capture/compare register as capture register

Figure 6-41. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)



Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

Figure 6-41. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)

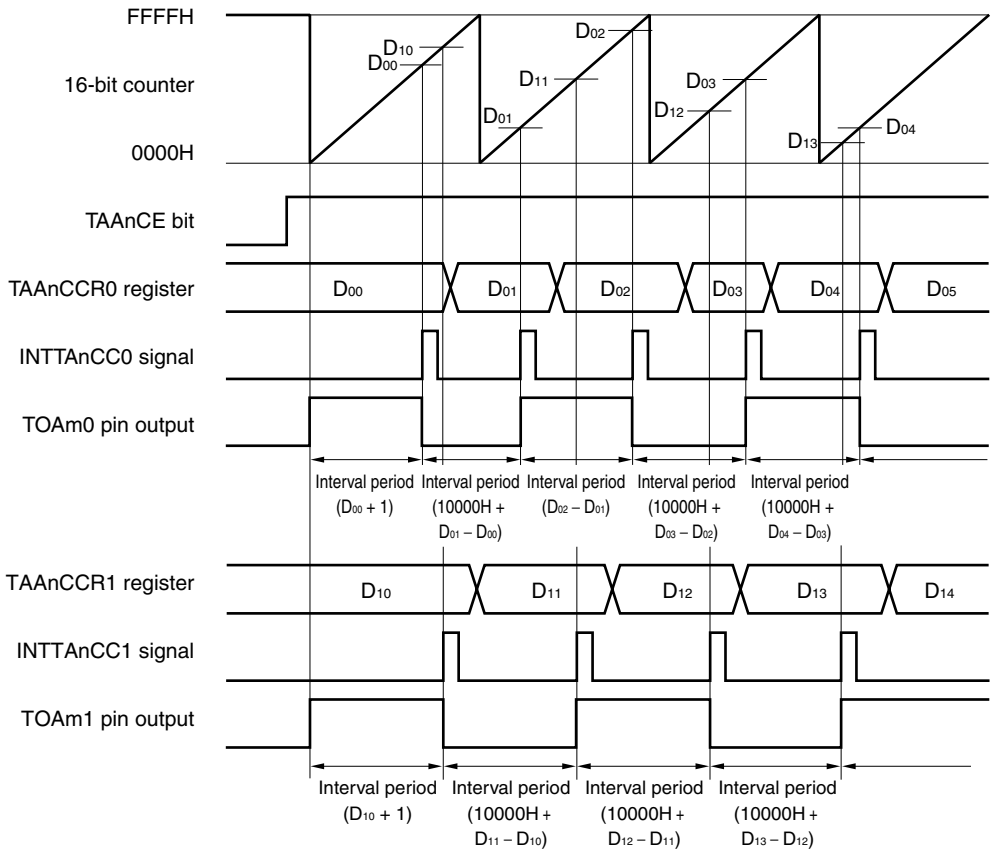


Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

(2) Operation timing in free-running timer mode

(a) Interval operation with compare register

When 16-bit timer/event counter AA is used as an interval timer with the TAAnCCRa register used as a compare register, software processing is necessary for setting a comparison value to generate the next interrupt request signal each time the INTTAnCCa signal has been detected.



When performing an interval operation in the free-running timer mode, two intervals can be set with one channel.

To perform the interval operation, the value of the corresponding TAAnCCRa register must be re-set in the interrupt servicing that is executed when the INTTAnCCa signal is detected.

The set value for re-setting the TAAnCCRa register can be calculated by the following expression, where "D_a" is the interval period.

Compare register default value: $D_a - 1$

Value set to compare register second and subsequent time: Previous set value + D_a

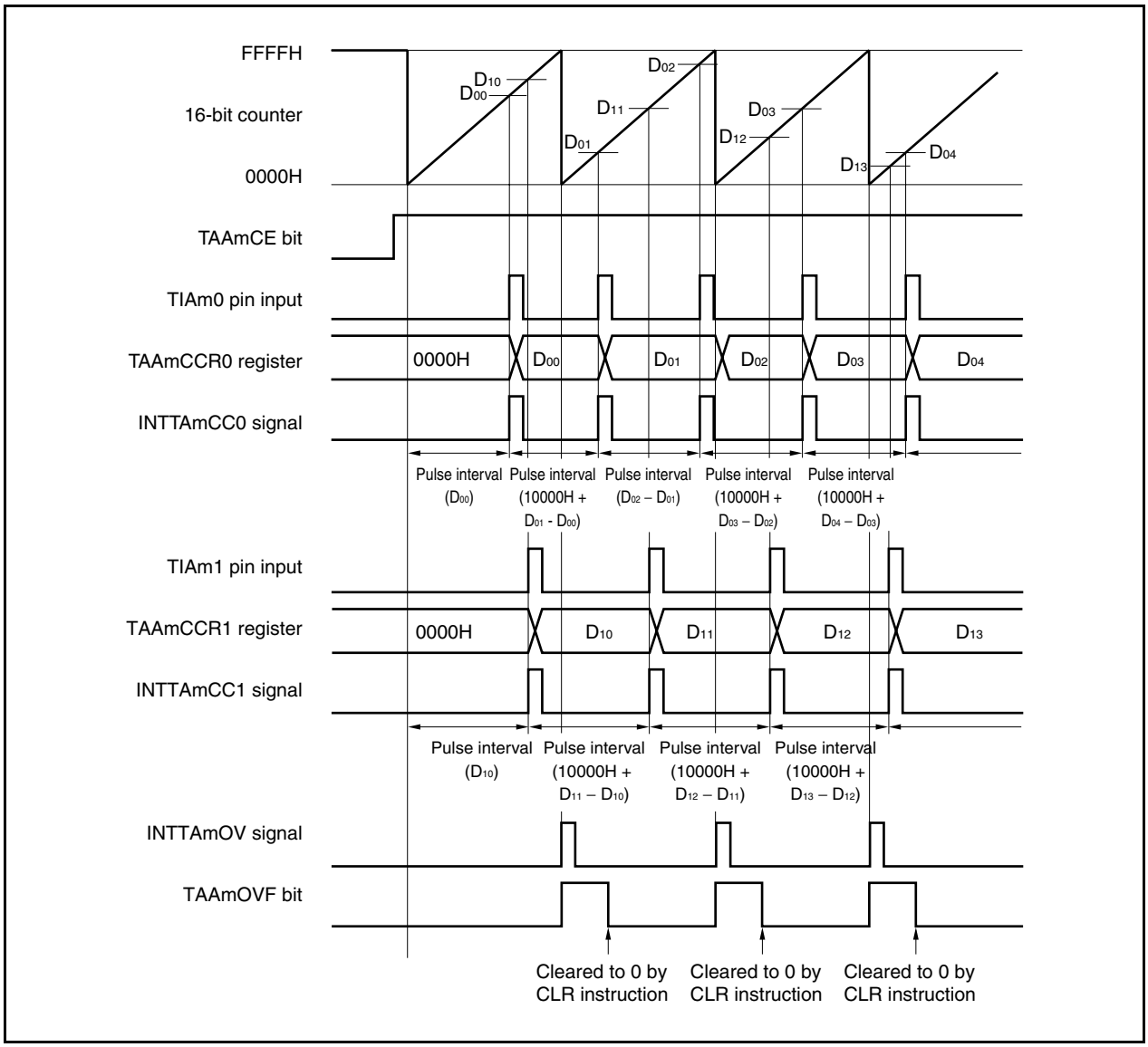
(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

Remark V850E/IF3: n = 0 to 4, m = 2, 4, a = 0, 1

V850E/IG3: n = 0 to 4, m = 2 to 4, a = 0, 1

(b) Pulse width measurement with capture register

When pulse width measurement is performed with the TAAmCCRa register used as a capture register, software processing is necessary for reading the capture register each time the INTTAmCCa signal has been detected and for calculating an interval.



When executing pulse width measurement in the free-running timer mode, two pulse widths can be measured with one channel.

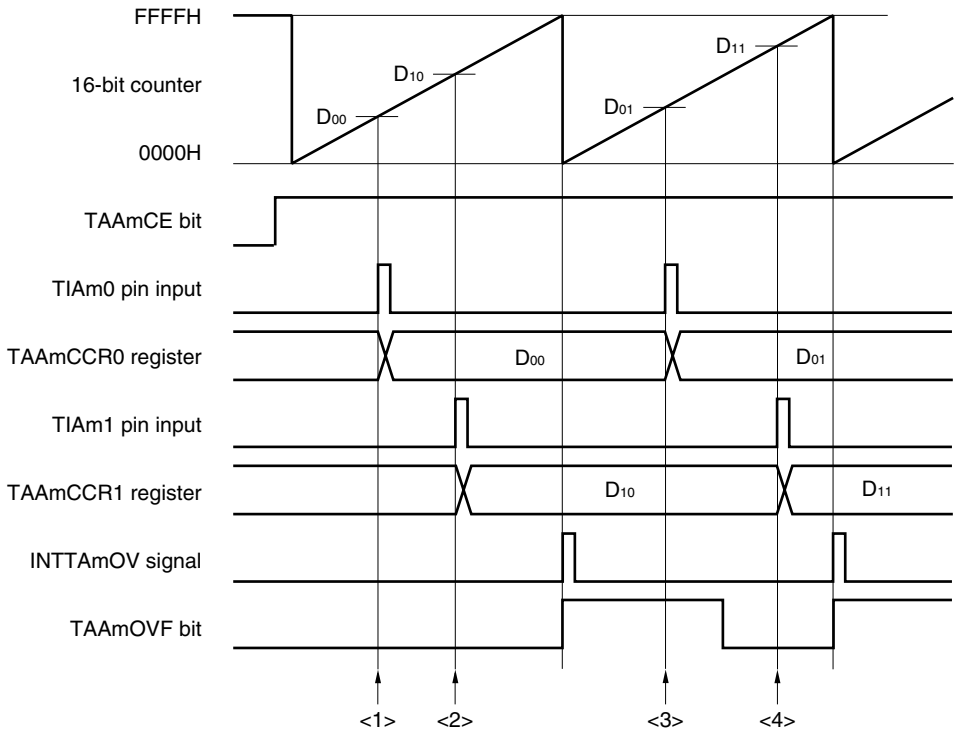
To measure a pulse width, the pulse width can be calculated by reading the value of the TAAmCCRa register in synchronization with the INTTAmCCa signal, and calculating the difference between the read value and the previously read value.

Remark V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

(c) Processing of overflow when two capture registers are used

Care must be exercised in processing the overflow flag when two capture registers are used. First, an example of incorrect processing is shown below.

Example of incorrect processing when two capture registers are used



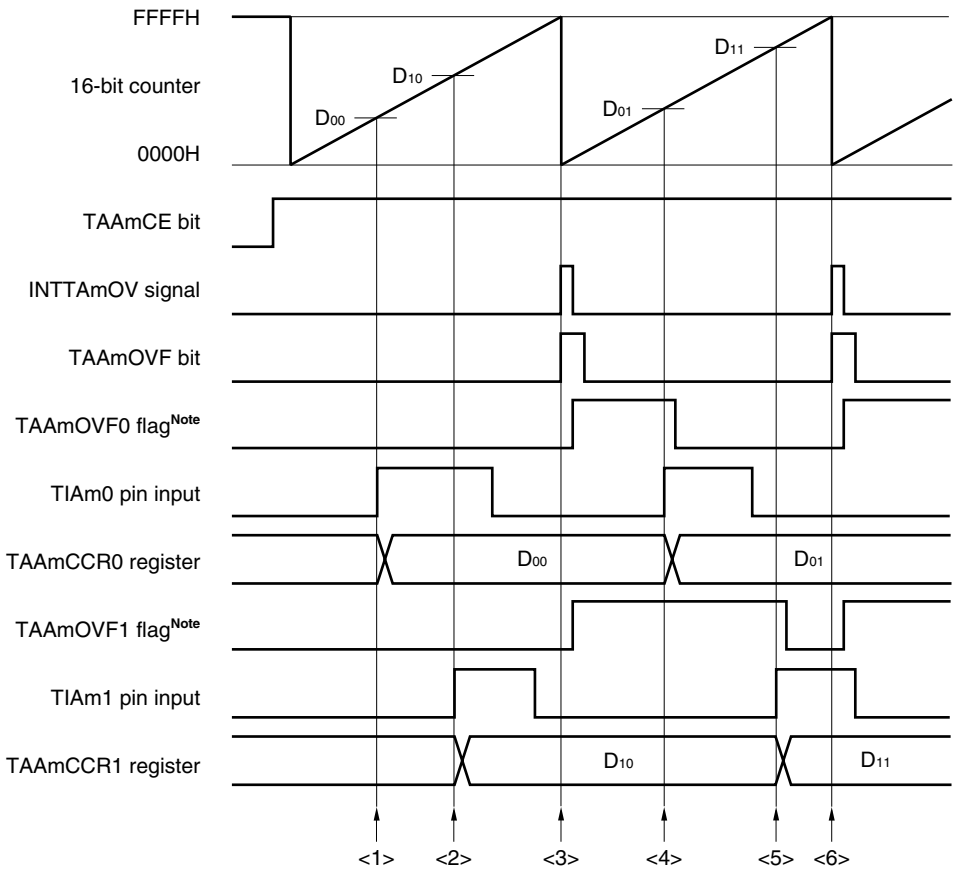
The following problem may occur when two pulse widths are measured in the free-running timer mode.

<1> Read the TAAmCCR0 register (setting of the default value of the TIAm0 pin input).
<2> Read the TAAmCCR1 register (setting of the default value of the TIAm1 pin input).
<3> Read the TAAmCCR0 register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).
<4> Read the TAAmCCR1 register.
Read the overflow flag. Because the flag is cleared in <3>, 0 is read.
Because the overflow flag is 0, the pulse width can be calculated by (D_{11} – D_{10}) (incorrect).

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

When two capture registers are used, and if the overflow flag is cleared to 0 by one capture register, the other capture register may not obtain the correct pulse width.
Use software when using two capture registers. An example of how to use software is shown below.

Example when two capture registers are used (using overflow interrupt)



Note The TAAmOVF0 and TAAmOVF1 flags are set on the internal RAM by software.

<1> Read the TAAmCCR0 register (setting of the default value of the TIAm0 pin input).

<2> Read the TAAmCCR1 register (setting of the default value of the TIAm1 pin input).

<3> An overflow occurs. Set the TAAmOVF0 and TAAmOVF1 flags to 1 in the overflow interrupt servicing, and clear the overflow flag to 0.

<4> Read the TAAmCCR0 register.
Read the TAAmOVF0 flag. If the TAAmOVF0 flag is 1, clear it to 0.
Because the TAAmOVF0 flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).

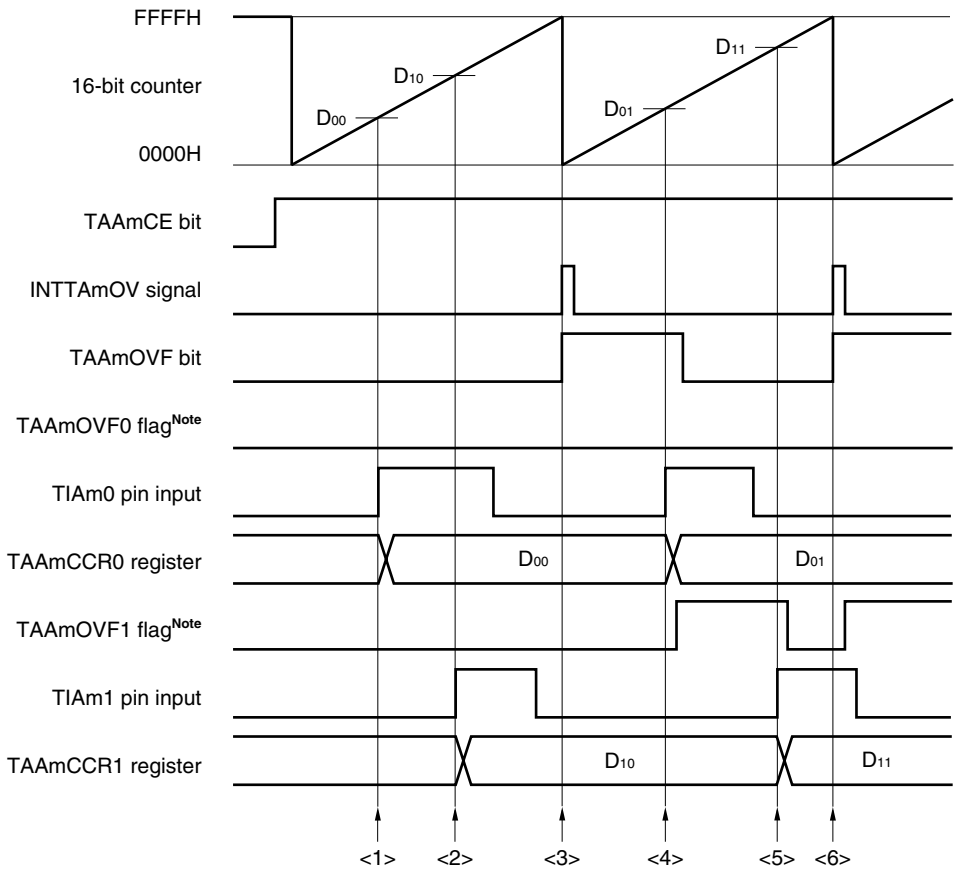
<5> Read the TAAmCCR1 register.
Read the TAAmOVF1 flag. If the TAAmOVF1 flag is 1, clear it to 0 (the TAAmOVF0 flag is cleared in <4>, and the TAAmOVF1 flag remains 1).
Because the TAAmOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} – D_{10}) (correct).

<6> Same as <3>

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

(2/2)

Example when two capture registers are used (without using overflow interrupt)



Note The TAAmOVF0 and TAAmOVF1 flags are set on the internal RAM by software.

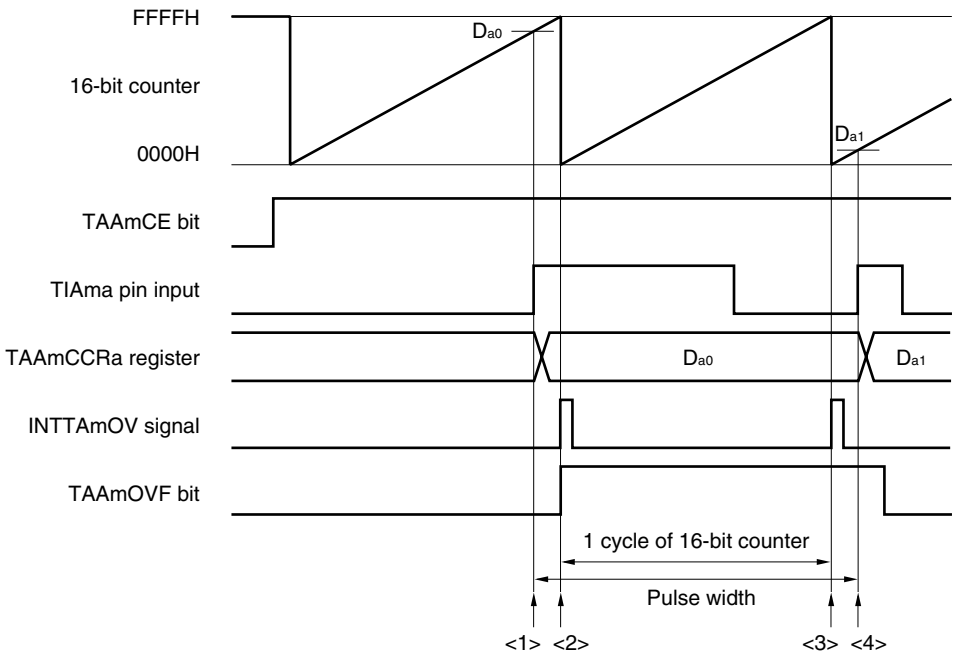
<1> Read the TAAmCCR0 register (setting of the default value of the TIAm0 pin input).
<2> Read the TAAmCCR1 register (setting of the default value of the TIAm1 pin input).
<3> An overflow occurs. Nothing is done by software.
<4> Read the TAAmCCR0 register.
Read the overflow flag. If the overflow flag is 1, set only the TAAmOVF1 flag to 1, and clear the overflow flag to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).
<5> Read the TAAmCCR1 register.
Read the overflow flag. Because the overflow flag is cleared in <4>, 0 is read.
Read the TAAmOVF1 flag. If the TAAmOVF1 flag is 1, clear it to 0.
Because the TAAmOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} – D_{10}) (correct).
<6> Same as <3>

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

(d) Processing of overflow if capture trigger interval is long

If the pulse width is greater than one cycle of the 16-bit counter, care must be exercised because an overflow may occur more than once from the first capture trigger to the next. First, an example of incorrect processing is shown below.

Example of incorrect processing when capture trigger interval is long



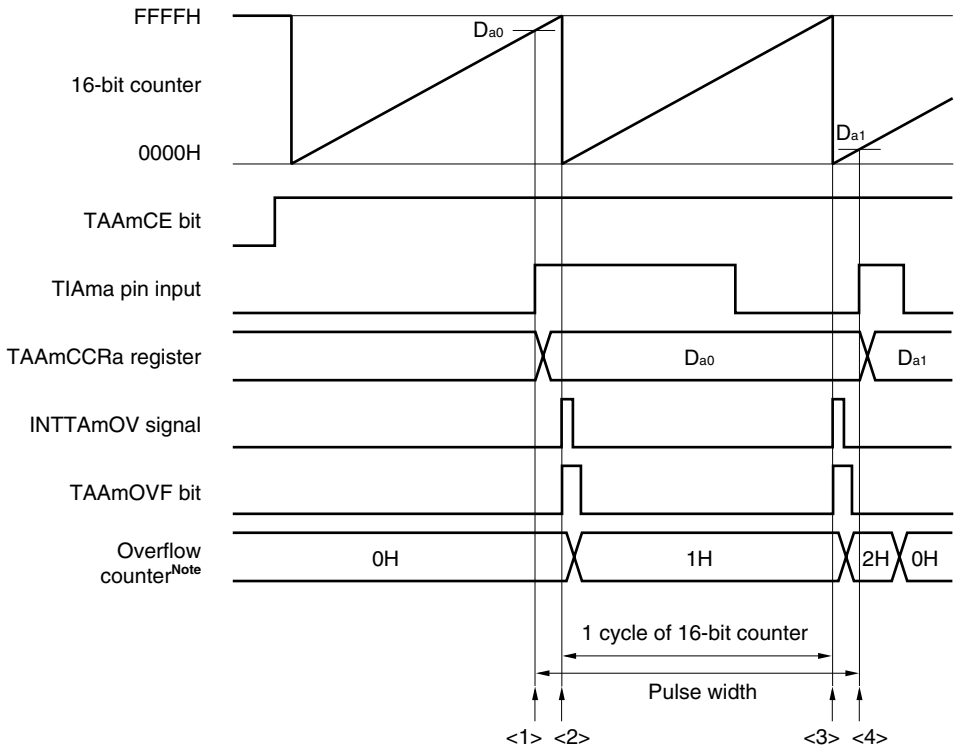
The following problem may occur when long pulse width is measured in the free-running timer mode.

<1> Read the TAAmCCRa register (setting of the default value of the TIAma pin input).
<2> An overflow occurs. Nothing is done by software.
<3> An overflow occurs a second time. Nothing is done by software.
<4> Read the TAAmCCRa register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{a1} – D_{a0}) (incorrect).
Actually, the pulse width must be (20000H + D_{a1} – D_{a0}) because an overflow occurs twice.

Remark V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

If an overflow occurs twice or more when the capture trigger interval is long, the correct pulse width may not be obtained.
If the capture trigger interval is long, slow the count clock to lengthen one cycle of the 16-bit counter, or use software. An example of how to use software is shown next.

Example when capture trigger interval is long



Note The overflow counter is set arbitrarily by software on the internal RAM.

<1> Read the TAAmCCRa register (setting of the default value of the TIAma pin input).

<2> An overflow occurs. Increment the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.

<3> An overflow occurs a second time. Increment the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.

<4> Read the TAAmCCRa register.

Read the overflow counter.

→ When the overflow counter is “N”, the pulse width can be calculated by ($N \times 10000H + D_{a1} - D_{a0}$).

In this example, the pulse width is ($20000H + D_{a1} - D_{a0}$) because an overflow occurs twice.

Clear the overflow counter (0H).

Remark V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

(e) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TAAmOVF bit to 0 with the CLR instruction after reading the TAAmOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TAAmOPT0 register after reading the TAAmOVF bit when it is 1.

6.6.7 Pulse width measurement mode (TAAmMD2 to TAAmMD0 bits = 110)

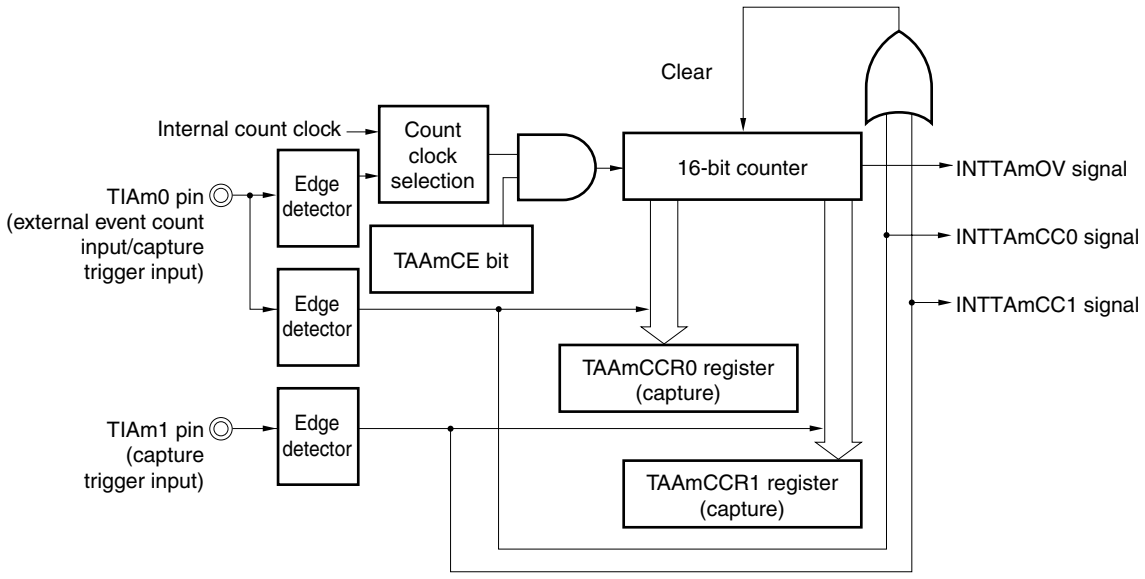
This mode is valid only in TAA2, TAA3 (V850E/IG3 only), and TAA4.

In the pulse width measurement mode, 16-bit timer/event counter AA starts counting when the TAAmCTL0.TAAmCE bit is set to 1. Each time the valid edge input to the TIAma pin has been detected, the count value of the 16-bit counter is stored in the TAAmCCRa register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TAAmCCRa register after a capture interrupt request signal (INTTAmCCa) occurs.

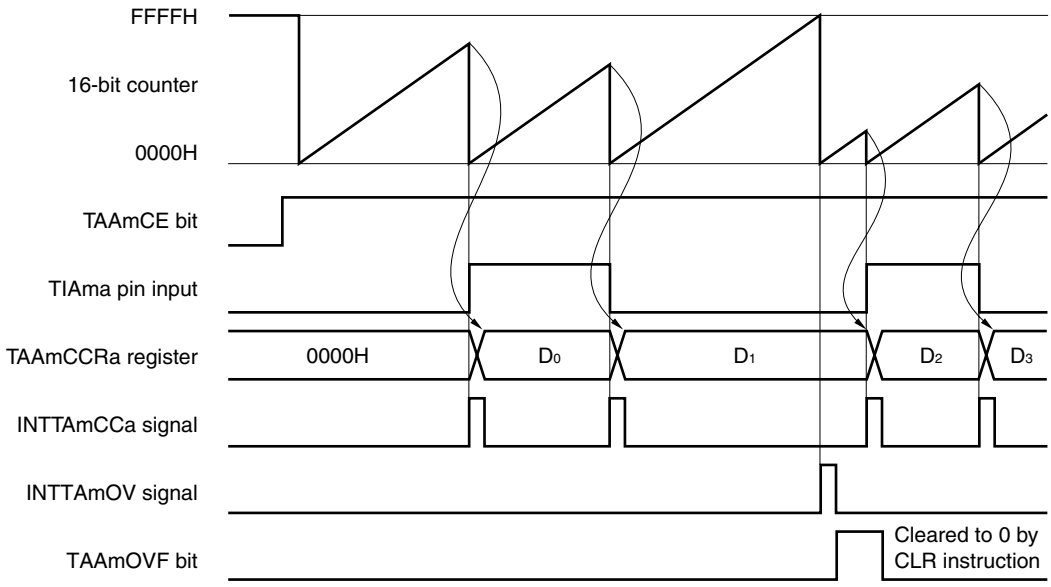
As shown in Figure 6-43, select either the TIAm0 or TIAm1 pin as the capture trigger input pin and set the unused pins to “No edge detection” by using the TAAmIOC1 register.

Figure 6-42. Configuration in Pulse Width Measurement Mode



Remark V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

Figure 6-43. Basic Timing in Pulse Width Measurement Mode



Remark V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

When the TAAmCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIAma pin is later detected, the count value of the 16-bit counter is stored in the TAAmCCRa register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTTAmCCa) is generated.

The pulse width is calculated as follows.

Pulse width = Captured value × Count clock cycle

If the valid edge is not input to the TIAma pin even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTAmOV) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TAAmOPT0.TAAmOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction via software.

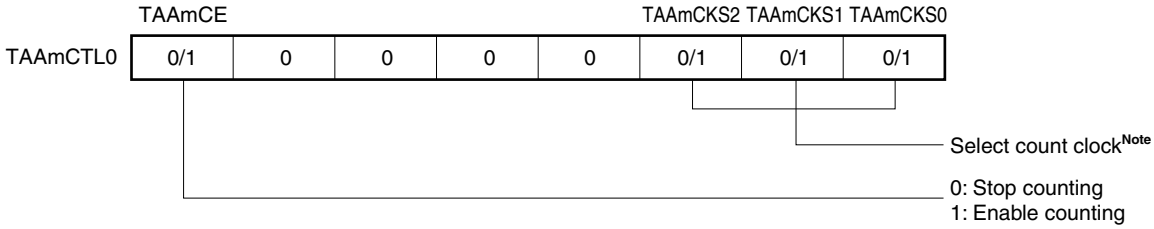
If the overflow flag is set to 1, the pulse width can be calculated as follows.

Pulse width = (10000H × TAAmOVF bit set (1) count + Captured value) × Count clock cycle

Remark V850E/IF3: m = 2, 4, a = 0, 1
V850E/IG3: m = 2 to 4, a = 0, 1

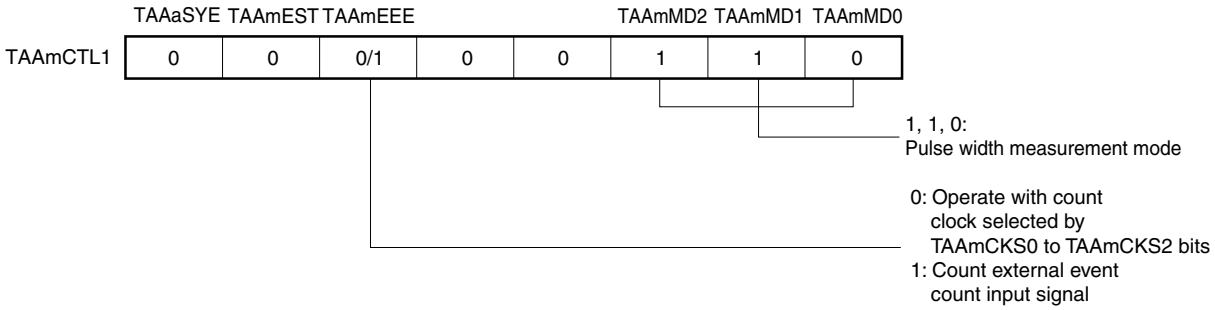
Figure 6-44. Register Setting in Pulse Width Measurement Mode (1/2)

(a) TAAm control register 0 (TAAmCTL0)

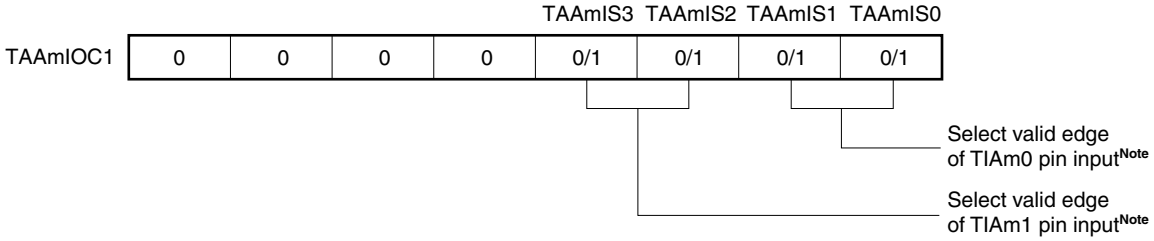


Note Setting is invalid when the TAAmCTL1.TAAmEEE bit = 1.

(b) TAAm control register 1 (TAAmCTL1)

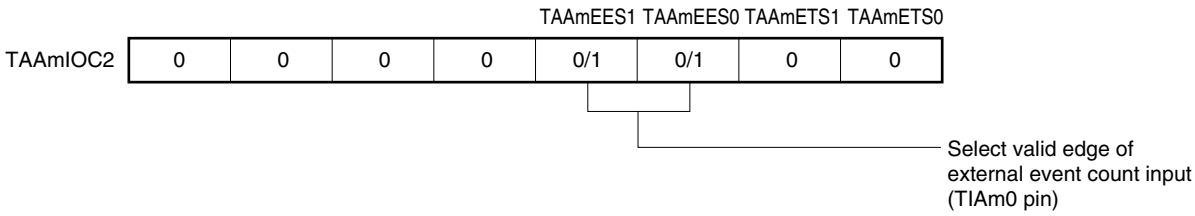


(c) TAAm I/O control register 1 (TAAmIOC1)



Note Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

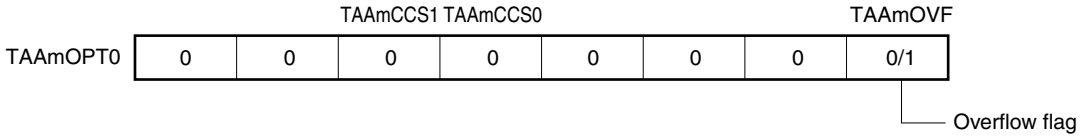
(d) TAAm I/O control register 2 (TAAmIOC2)



Note Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

Figure 6-44. Register Setting in Pulse Width Measurement Mode (2/2)

(e) TAAm option register 0 (TAAmOPT0)



(f) TAAm counter read buffer register (TAAmCNT)

The value of the 16-bit counter can be read by reading the TAAmCNT register.

(g) TAAm capture/compare registers 0 and 1 (TAAmCCR0 and TAAmCCR1)

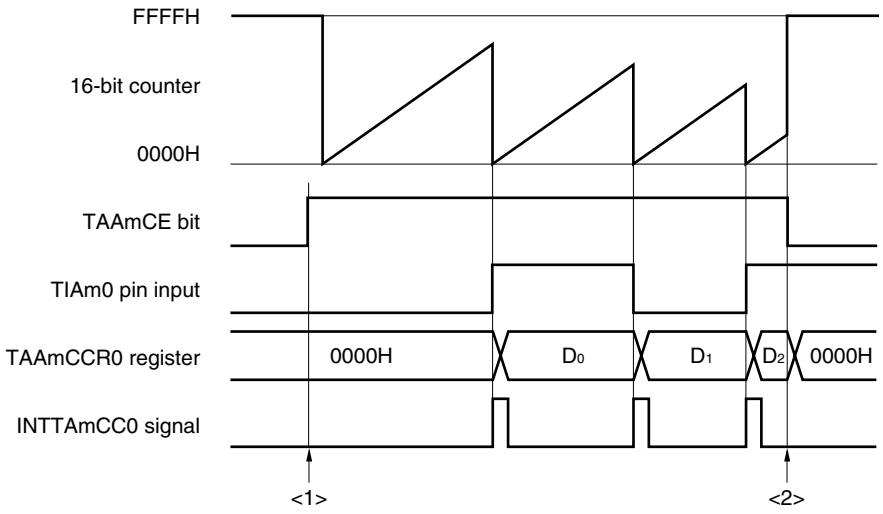
These registers store the count value of the 16-bit counter when the valid edge input to the TIAm0 and TIAm1 pins is detected.

Remarks 1. TAAm I/O control register 0 (TAAmIOC0) is not used in the pulse width measurement mode.

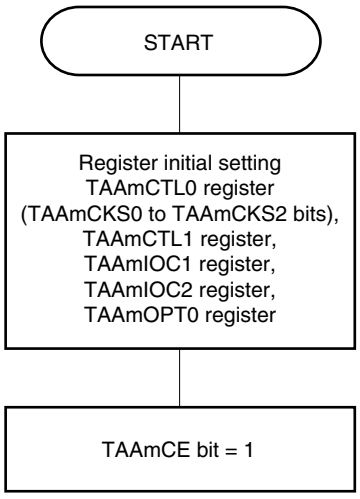
2. V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

(1) Operation flow in pulse width measurement mode

Figure 6-45. Software Processing Flow in Pulse Width Measurement Mode



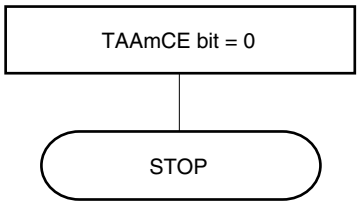
<1> Count operation start flow



Initial setting of these registers is performed before setting the TAAmCE bit to 1.

The TAAmCKS0 to TAAmCKS2 bits can be set at the same time when counting has been started (TAAmCE bit = 1).

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TAAmCE bit to 0.

Remark V850E/IF3: m = 2, 4
V850E/IG3: m = 2 to 4

(2) Operation timing in pulse width measurement mode

(a) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TAAmOVF bit to 0 with the CLR instruction after reading the TAAmOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TAAmOPT0 register after reading the TAAmOVF bit when it is 1.

CHAPTER 7 16-BIT TIMER/EVENT COUNTER AB (TAB)

Timer AB (TAB) is a 16-bit timer/event counter.
The V850E/IF3 and V850E/IG3 incorporate TAB0 and TAB1.

7.1 Overview

An outline of TABn is shown below (n = 0, 1).

- Clock selection: 8 ways
- Capture/trigger input pins: 4
- External event count input pins: 1
- External trigger input pins: 1
- Timer/counters: 1
- Capture/compare registers: 4
- Capture/compare match interrupt request signals: 4
- Overflow interrupt request signal: 1
- Timer output pins[Note]: 4

Note This is the number of output pins of TABn; it does not include the output pins of TMQOPn. For details of the output pins of TMQOPn, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

7.2 Functions

TABn has the following functions (n = 0, 1).

- 6-phase PWM output[Note]
- Interval timer
- External event counter
- External trigger pulse output
- One-shot pulse output
- PWM output
- Free-running timer
- Pulse width measurement

Note This is connected to TMQOPn. For details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

7.3 Configuration

TABn includes the following hardware (n = 0, 1).

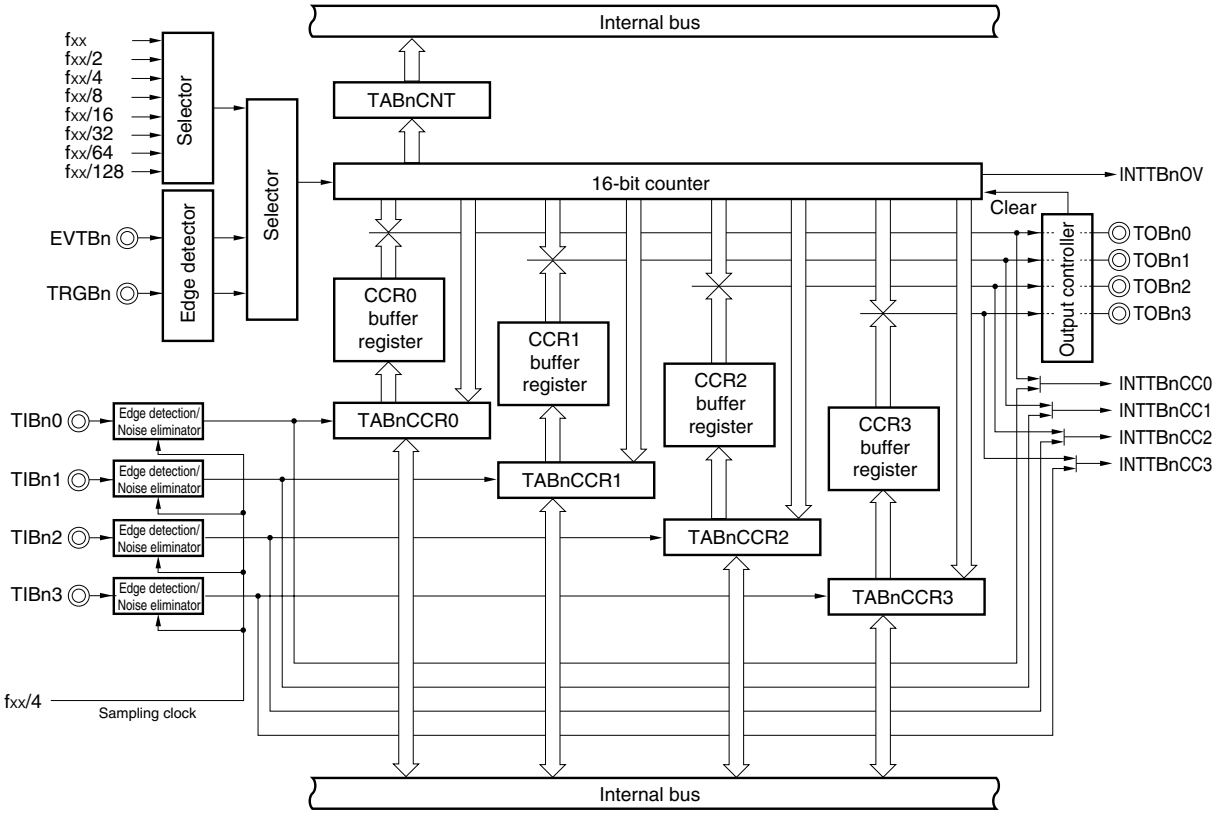
Table 7-1. TABn Configuration

Item	Configuration
Timer register	16-bit counter × 1
Registers	TABn counter read buffer register (TABnCNT): 2 in total TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3): 8 in total CCR0 to CCR3 buffer registers: 8 in total
Timer input	12 in total (TIB00 to TIB03, TIB10 to TIB13, EVTB0, EVTB1, TRGB0, TRGB1 pins)[Note]
Timer output	8 in total (TOB00 to TOB03, TOB10 to TOB13 pins)[Note]
Control registers	TABn control registers 0, 1 (TABnCTL0, TABnCTL1) TABn I/O control registers 0 to 2 (TABnIOC0 to TABnIOC2) TABn option register 0 (TABnOPT0)

Note The TIBn1 to TIBn3 pins function alternately as timer output pins (TOBn1 to TOBn3).

Remark n = 0, 1

Figure 7-1. TABn Block Diagram



Remarks 1. fxx: Peripheral clock

2. For the noise eliminator, see **4.6 Noise Eliminator**.

3. n = 0, 1

(1) 16-bit counter

This 16-bit counter can count internal clocks or external events.

The count value of this counter can be read by using the TABnCNT register.

When the TABnCTL0.TABnCE bit = 0, the value of the 16-bit counter is FFFFH. If the TABnCNT register is read at this time, 0000H is read.

Reset sets the TABnCE bit to 0.

(2) CCR0 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TABnCCR0 register is used as a compare register, the value written to the TABnCCR0 register is transferred to the CCR0 buffer register. When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTBnCC0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is cleared to 0000H after reset, and the TABnCCR0 register is cleared to 0000H.

(3) CCR1 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TABnCCR1 register is used as a compare register, the value written to the TABnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTBnCC1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is cleared to 0000H after reset, and the TABnCCR1 register is cleared to 0000H.

(4) CCR2 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TABnCCR2 register is used as a compare register, the value written to the TABnCCR2 register is transferred to the CCR2 buffer register. When the count value of the 16-bit counter matches the value of the CCR2 buffer register, a compare match interrupt request signal (INTTBnCC2) is generated.

The CCR2 buffer register cannot be read or written directly.

The CCR2 buffer register is cleared to 0000H after reset, and the TABnCCR2 register is cleared to 0000H.

(5) CCR3 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TABnCCR3 register is used as a compare register, the value written to the TABnCCR3 register is transferred to the CCR3 buffer register. When the count value of the 16-bit counter matches the value of the CCR3 buffer register, a compare match interrupt request signal (INTTBnCC3) is generated.

The CCR3 buffer register cannot be read or written directly.

The CCR3 buffer register is cleared to 0000H after reset, and the TABnCCR3 register is cleared to 0000H.

(6) Edge detector

This circuit detects the valid edges input to the TIBn0 to TIBn3, EVTBn, and TRGBn pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TABnIOC1 and TABnIOC2 registers.

(7) Output controller

This circuit controls the output of the TOBn0 to TOBn3 pins. The output controller is controlled by the TABnIOC0 register.

(8) Selector

This selector selects the count clock for the 16-bit counter. Eight types of internal clocks or an external event can be selected as the count clock.

7.4 Registers

(1) TABn control register 0 (TABnCTL0)

The TABnCTL0 register is an 8-bit register that controls the operation of TABn.

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

Reset sets this register to 00H.

The same value can always be written to the TABnCTL0 register by software.

After reset: 00H R/W Address: TAB0CTL0 FFFFF5E0H, TAB1CTL0 FFFFF620H

	<7>	6	5	4	3	2	1	0
TABnCTL0	TABnCE	0	0	0	0	TABnCKS2	TABnCKS1	TABnCKS0

(n = 0, 1)

TABnCE	TABn operation control
0	TABn operation disabled (TABn reset asynchronously[Note])
1	TABn operation enabled.

TABnCKS2	TABnCKS1	TABnCKS0	Internal count clock selection
0	0	0	f_{XX}
0	0	1	$f_{XX}/2$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/8$
1	0	0	$f_{XX}/16$
1	0	1	$f_{XX}/32$
1	1	0	$f_{XX}/64$
1	1	1	$f_{XX}/128$

Note The TABnOPT0.TABnOVF bit and the 16-bit counter are reset simultaneously. Moreover, timer outputs (TOBn0 to TOBn3 pins) are reset to the TABnIOC0 register set status at the same time as the 16-bit counter.

Cautions 1. Set the TABnCKS2 to TABnCKS0 bits when the TABnCE bit = 0.
When the value of the TABnCE bit is changed from 0 to 1, the TABnCKS2 to TABnCKS0 bits can be set simultaneously.
2. Be sure to set bits 3 to 6 to "0".

Remark f_{XX}: Peripheral clock

(2) TABn control register 1 (TABnCTL1)

The TABnCTL1 register is an 8-bit register that controls the operation of TABn.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0CTL1 FFFFF5E1H, TAB1CTL1 FFFFF621H

	7	6	5	4	3	2	1	0
TABnCTL1 (n = 0, 1)	0	TABnEST	TABnEEE	0	0	TABnMD2	TABnMD1	TABnMD0

TABnEST	Software trigger control
0	–
1	Generate a valid signal for external trigger input. • In one-shot pulse output mode: A one-shot pulse is output with writing 1 to the TABnEST bit as the trigger. • In external trigger pulse output mode: A PWM waveform is output with writing 1 to the TABnEST bit as the trigger.
Read value of the TABnEST bit is always 0.	

TABnEEE	Count clock selection
0	Disable operation with external event count input (EVTBn pin). (Perform counting with the count clock selected by the TABnCTL0.TABnCKS0 to TABnCKS2 bits.)
1	Enable operation with external event count input (EVTBn pin). (Perform counting at the valid edge of the external event count input signal (EVTBn pin).)
The TABnEEE bit selects whether counting is performed with the internal count clock or the valid edge of the external event count input.	

TABnMD2	TABnMD1	TABnMD0	Timer mode selection
0	0	0	Interval timer mode
0	0	1	External event count mode
0	1	0	External trigger pulse output mode
0	1	1	One-shot pulse output mode
1	0	0	PWM output mode
1	0	1	Free-running timer mode
1	1	0	Pulse width measurement mode
1	1	1	6-phase PWM output mode[Note]

Note The 6-phase PWM output mode cannot be used when only TABn is used. For details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

Cautions 1. The TABnEST bit is valid only in the external trigger pulse output mode or one-shot pulse output mode. In any other mode, writing 1 to this bit is ignored.

2. External event count input is selected in the external event count mode regardless of the value of the TABnEEE bit.

3. Set the TABnEEE and TABnMD2 to TABnMD0 bits when the TABnCTL0.TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) The operation is not guaranteed when rewriting is performed with the TABnCE bit = 1. If rewriting was mistakenly performed, clear the TABnCE bit to 0 and then set the bits again.

4. Be sure to set bits 3, 4, and 7 to "0".

(3) TABn I/O control register 0 (TABnIOC0)

The TABnIOC0 register is an 8-bit register that controls the timer output (TOBn0 to TOBn3, TOBnT1 to TOBnT3 pins).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0IOC0 FFFFF5E2H, TAB1IOC0 FFFFF622H

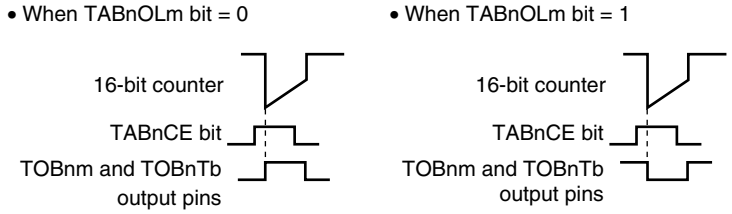
	7	<6>	5	<4>	3	<2>	1	<0>
TABnIOC0	TABnOL3	TABnOE3	TABnOL2	TABnOE2	TABnOL1	TABnOE1	TABnOL0	TABnOE0

n = 0, 1
b = 1 to 3

TABnOLm	TOBnm, TOBnTb pin output level setting[Note] (m = 0 to 3)
0	TOBnm and TOBnTb pins start output at high level.
1	TOBnm and TOBnTb pins start output at low level.

TABnOEm	TOBnm, TOBnTb pin output setting (m = 0 to 3)
0	Timer output disabled • When TABnOLm bit = 0: Low level is output from the TOBnm and TOBnTb pins • When TABnOLm bit = 1: High level is output from the TOBnm and TOBnTb pins
1	Timer output enabled (A pulse is output from the TOBnm and TOBnTb pins).

Note The output level of the timer output pins (TOBnm and TOBnTb) specified by the TABnOLm bit is shown below.



Cautions 1. If the setting of the TABnIOC0 register is changed when TOBnm and TOBnTb are set in the output mode, the output of the pins change. Set the port in the input mode and make the port go into a high-impedance state, noting changes in the pin status.

2. Rewrite the TABnOLm and TABnOEm bits when the TABnCTL0.TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) If rewriting was mistakenly performed, clear (0) the TABnCE bit and then set the bits again.

3. If the TABnOLm bit is manipulated when the TABnCE and TABnOEm bits are 0, the output level of the TOBnm and TOBnTb pins changes.

4. To generate the TOBnTb pin output and the A/D conversion start trigger signal of A/D converters 0 and 1 in the 6-phase PWM output mode, be sure to set the TOBnTb pin output using the TABnIOC0 register. At this time, be sure to clear the TABnOL0 bit to 0 and set the TABnOE0 bit to 1 (b = 1 to 3).

Remark m = 0 to 3

(4) TABn I/O control register 1 (TABnIOC1)

The TABnIOC1 register is an 8-bit register that controls the valid edge of the capture trigger input signals (TIBn0 to TIBn3 pins).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0IOC1 FFFFF5E3H, TAB1IOC1 FFFFF623H

	7	6	5	4	3	2	1	0
TABnIOC1 (n = 0, 1)	TABnIS7	TABnIS6	TABnIS5	TABnIS4	TABnIS3	TABnIS2	TABnIS1	TABnIS0

TABnIS7	TABnIS6	Capture trigger input signal (TIBn3 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TABnIS5	TABnIS4	Capture trigger input signal (TIBn2 pin) valid edge detection
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TABnIS3	TABnIS2	Capture trigger input signal (TIBn1 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TABnIS1	TABnIS0	Capture trigger input signal (TIBn0 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Cautions 1. Rewrite the TABnIS7 to TABnIS0 bits when the TABnCTL0.TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) If rewriting was mistakenly performed, clear the TABnCE bit to 0 and then set the bits again.

2. The TABnIS7 to TABnIS0 bits are valid only in the free-running timer mode (only when the TABnOPT0.TABnCCSm bit = 1) and the pulse width measurement mode (m = 0 to 3). In all other modes, a capture operation is not possible.

(5) TABn I/O control register 2 (TABnIOC2)

The TABnIOC2 register is an 8-bit register that controls the valid edge of the external event count input signal (EVTBn pin) and external trigger input signal (TRGBn pin).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0IOC2 FFFFF5E4H, TAB1IOC2 FFFFF624H

	7	6	5	4	3	2	1	0
TABnIOC2	0	0	0	0	TABnEES1	TABnEES0	TABnETS1	TABnETS0

(n = 0, 1)

TABnEES1	TABnEES0	External event count input signal (EVTBn pin) valid edge setting
0	0	No edge detection (external event count invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TABnETS1	TABnETS0	External trigger input signal (TRGBn pin) valid edge setting
0	0	No edge detection (external trigger invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Cautions 1. Rewrite the TABnEES1, TABnEES0, TABnETS1, and TABnETS0 bits when the TABnCTL0.TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) If rewriting was mistakenly performed, clear the TABnCE bit to 0 and then set the bits again.

2. The TABnEES1 and TABnEES0 bits are valid only when the TABnCTL1.TABnEEE bit = 1 or when the external event count mode (TABnCTL1.TABnMD2 to TABnCTL1.TABnMD0 bits = 001) has been set.

3. The TABnETS1 and TABnETS0 bits are valid only in the external trigger pulse output mode or one-shot pulse output mode.

(6) TABn option register 0 (TABnOPT0)

The TABnOPT0 register is an 8-bit register used to set the capture/compare operation and detect an overflow.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0OPT0 FFFFF5E5H, TAB1OPT0 FFFFF625H

	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
TABnOPT0	TABnCCS3	TABnCCS2	TABnCCS1	TABnCCS0	0	TABnCMS[Note]	TABnCUF[Note]	TABnOVF

(n = 0, 1)

TABnCCSm	TABnCCRm register capture/compare selection (m = 0 to 3)
0	Compare register selected
1	Capture register selected (cleared by TABnCTL0.TABnCE bit = 0)
The TABnCCSm bit setting is valid only in the free-running timer mode.	

TABnOVF	TABn overflow flag
Set (1)	Overflow occurred
Reset (0)	TABnOVF bit 0 written or TABnCTL0.TABnCE bit = 0
• The TABnOVF bit is set to 1 when the 16-bit counter count value overflows from FFFFH to 0000H in the free-running timer mode or the pulse width measurement mode. • An overflow interrupt request signal (INTTBnOV) is generated at the same time that theTABnOVF bit is set to 1. The INTTBnOV signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode. • The TABnOVF bit is not cleared to 0 even when the TABnOVF bit or the TABnOPT0 register are read when the TABnOVF bit = 1. • Before clearing the TABnOVF bit to 0 after generation of the INTTBnOV signal, be sure to confirm (by reading) that the TABnOVF bit is set to 1. • The TABnOVF bit can be both read and written, but the TABnOVF bit cannot be set to 1 by software. Writing 1 has no influence on the operation of TABn.	

Note For details of the TABnCMS and TABnCUF bits, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

Cautions 1. Rewrite the TABnCCS3 to TABnCCS0 bits when the TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) If rewriting was mistakenly performed, clear the TABnCE bit to 0 and then set the bits again.

2. Be sure to set bit 3 to "0".

(7) TABn capture/compare register 0 (TABnCCR0)

The TABnCCR0 register is a 16-bit register that can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TABnOPT0.TABnCCS0 bit. In the pulse width measurement mode, the TABnCCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TABnCCR0 register can be read or written during operation.

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

Reset sets this register to 0000H.

After reset: 0000H R/W Address: TAB0CCR0 FFFFF5E6H, TAB1CCR0 FFFFF626H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TABnCCR0																

(n = 0, 1)

(a) Function as compare register

The TABnCCR0 register can be rewritten even when the TABnCTL0.TABnCE bit = 1.

The set value of the TABnCCR0 register is transferred to the CCR0 buffer register. When the value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTBnCC0) is generated. If TOBn0 pin output is enabled at this time, the output of the TOBn0 pin is inverted.

When the TABnCCR0 register is used as a cycle register in the interval timer mode, external event count mode, external trigger pulse output mode, one-shot pulse output mode, or PWM output mode, the value of the 16-bit counter is cleared (0000H) if its count value matches the value of the CCR0 buffer register.

The compare register is not cleared by setting the TABnCTL0.TABnCE bit to 0.

(b) Function as capture register

When the TABnCCR0 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TABnCCR0 register if the valid edge of the capture trigger input pin (TIBn0 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TABnCCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIBn0 pin) is detected.

Even if the capture operation and reading the TABnCCR0 register conflict, the correct value of the TABnCCR0 register can be read.

The capture register is cleared by setting the TABnCTL0.TABnCE bit = 0.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 7-2. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write[Note]
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write[Note]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	None

Note Writing to the TABnCCR1 register is the trigger.

Remark For anytime write and batch write, see **7.6 (2) Anytime write and batch write**.

(8) TABn capture/compare register 1 (TABnCCR1)

The TABnCCR1 register, which consists of 16 bits, can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TABnOPT0.TABnCCS1 bit. In the pulse width measurement mode, the TABnCCR1 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TABnCCR1 register can be read or written during operation.

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

Reset sets this register to 0000H.

After reset: 0000H R/W Address: TAB0CCR1 FFFFF5E8H, TAB1CCR1 FFFFF628H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TABnCCR1																

(n = 0, 1)

(a) Function as compare register

The TABnCCR1 register can be rewritten even when the TABnCTL0.TABnCE bit = 1.

The set value of the TABnCCR1 register is transferred to the CCR1 buffer register. When the value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTBnCC1) is generated. If TOBn1 pin output is enabled at this time, the output of the TOBn1 pin is inverted.

The compare register is not cleared by setting the TABnCTL0.TABnCE bit to 0.

(b) Function as capture register

When the TABnCCR1 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TABnCCR1 register if the valid edge of the capture trigger input pin (TIBn1 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TABnCCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIBn1 pin) is detected.

Even if the capture operation and reading the TABnCCR1 register conflict, the correct value of the TABnCCR1 register can be read.

The capture register is cleared by setting the TABnCTL0.TABnCE bit to 0.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 7-3. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write[Note]
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write[Note]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	None

Note Writing to the TABnCCR1 register is the trigger.

Remark For anytime write and batch write, see **7.6 (2) Anytime write and batch write**.

(9) TABn capture/compare register 2 (TABnCCR2)

The TABnCCR2 register is a 16-bit register that can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TABnOPT0.TABnCCS2 bit. In the pulse width measurement mode, the TABnCCR2 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TABnCCR2 register can be read or written during operation.

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

Reset sets this register to 0000H.

After reset: 0000H R/W Address: TAB0CCR2 FFFFF5EAH, TAB1CCR2 FFFFF62AH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TABnCCR2																

(n = 0, 1)

(a) Function as compare register

The TABnCCR2 register can be rewritten even when the TABnCTL0.TABnCE bit = 1.

The set value of the TABnCCR2 register is transferred to the CCR2 buffer register. When the value of the 16-bit counter matches the value of the CCR2 buffer register, a compare match interrupt request signal (INTTBnCC2) is generated. If TOBn2 pin output is enabled at this time, the output of the TOBn2 pin is inverted.

The compare register is not cleared by setting the TABnCTL0.TABnCE bit to 0.

(b) Function as capture register

When the TABnCCR2 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TABnCCR2 register if the valid edge of the capture trigger input pin (TIBn2 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TABnCCR2 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIBn2 pin) is detected.

Even if the capture operation and reading the TABnCCR2 register conflict, the correct value of the TABnCCR2 register can be read.

The capture register is cleared by setting the TABnCTL0.TABnCE bit to 0.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 7-4. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write[Note]
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write[Note]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	None

Note Writing to the TABnCCR1 register is the trigger.

Remark For anytime write and batch write, see **7.6 (2) Anytime write and batch write**.

(10) TABn capture/compare register 3 (TABnCCR3)

The TABnCCR3 register, which consists of 16 bits, can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TABnOPT0.TABnCCS3 bit. In the pulse width measurement mode, the TABnCCR3 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TABnCCR3 register can be read or written during operation.

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

Reset sets this register to 0000H.

After reset: 0000H R/W Address: TAB0CCR3 FFFFF5ECH, TAB1CCR3 FFFFF62CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TABnCCR3																

(n = 0, 1)

(a) Function as compare register

The TABnCCR3 register can be rewritten even when the TABnCTL0.TABnCE bit = 1.

The set value of the TABnCCR3 register is transferred to the CCR3 buffer register. When the value of the 16-bit counter matches the value of the CCR3 buffer register, a compare match interrupt request signal (INTTBnCC3) is generated. If TOBn3 pin output is enabled at this time, the output of the TOBn3 pin is inverted.

The compare register is not cleared by setting the TABnCTL0.TABnCE bit to 0.

(b) Function as capture register

When the TABnCCR3 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TABnCCR3 register if the valid edge of the capture trigger input pin (TIBn3 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TABnCCR3 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIBn3 pin) is detected.

Even if the capture operation and reading the TABnCCR3 register conflict, the correct value of the TABnCCR3 register can be read.

The capture register is cleared by setting the TABnCTL0.TABnCE bit to 0.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 7-5. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write[Note]
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write[Note]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	None

Note Writing to the TABnCCR1 register is the trigger.

Remark For anytime write and batch write, see **7.6 (2) Anytime write and batch write**.

(11) TABn counter read buffer register (TABnCNT)

The TABnCNT register is a read buffer register that can read the count value of the 16-bit counter.

If this register is read when the TABnCTL0.TABnCE bit = 1, the count value of the 16-bit timer can be read.

This register is read-only, in 16-bit units.

The value of the TABnCNT register is set to 0000H when the TABnCE bit = 0. If the TABnCNT register is read at this time, the value of the 16-bit counter (FFFFH) is not read, but 0000H is read.

The value of the TABnCNT register is set to 0000H after reset, and the TABnCE bit is cleared to 0.

After reset: 0000H R Address: TAB0CNT FFFFF5EEH, TAB1CNT FFFFF62EH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TABnCNT																

(n = 0, 1)

7.5 Timer Output Operations

The following table shows the operations and output levels of the TOBn0 to TOBn3 pins.

Table 7-6. Timer Output Control in Each Mode

Operation Mode	TOBn0 Pin	TOBn1 Pin	TOBn2 Pin	TOBn3 Pin
Interval timer mode	PWM output			
External event count mode	None			
External trigger pulse output mode	PWM output	External trigger pulse output	External trigger pulse output	External trigger pulse output
One-shot pulse output mode		One-shot pulse output	One-shot pulse output	One-shot pulse output
PWM output mode		PWM output	PWM output	PWM output
Free-running timer mode	PWM output (only when compare function is used)			
Pulse width measurement mode	None			

Remark n = 0, 1

Table 7-7. Truth Table of TOBn0 to TOBn3 Pins Under Control of Timer Output Control Bits

TABnIOC0.TABnOLa Bit	TABnIOC0.TABnOEa Bit	TABnCTL0.TABnCE bit	Level of TOBna Pin
0	0	×	Low-level output
	1	0	Low-level output
		1	Low level immediately before counting, high level after counting is started
1	0	×	High-level output
	1	0	High-level output
		1	High level immediately before counting, low level after counting is started

Remark n = 0, 1
a = 0 to 3

7.6 Operation

TABn can perform the following functions.

Table 7-8. TABn Specifications in Each Mode

Operation	TABnCTL1.TABnEST Bit (Software Trigger Bit)	TRGBn Pin (External Trigger Input)	Capture/Compare Register Setting	Compare Register Write
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode	Invalid	Invalid	Compare only	Anytime write
External trigger pulse output mode	Valid	Valid	Compare only	Batch write
One-shot pulse output mode	Valid	Valid	Compare only	Anytime write
PWM output mode	Invalid	Invalid	Compare only	Batch write
Free-running timer mode	Invalid	Invalid	Switching enabled	Anytime write
Pulse width measurement mode	Invalid	Invalid	Capture only	Not applicable

Remarks 1. TABn has a function to execute tuning with TAAn. For details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

2. n = 0, 1

(1) Counter basic operation

This section explains the basic operation of the 16-bit counter. For details, refer to the description of the operation in each mode.

Remark n = 0, 1
a = 0 to 3

(a) Counter start operation

The 16-bit counter of TABn starts counting from the default value FFFFH in all modes.
It counts up from FFFFH to 0000H, 0001H, 0002H, 0003H, and so on.

(b) Clear operation

The 16-bit counter is cleared to 0000H when its value matches the value of the compare register and when its value is captured. The count operation from FFFFH to 0000H that takes place immediately after the counter has started counting or when the counter overflows is not a clearing operation. Therefore, the INTTBnCCa interrupt signal is not generated.

(c) Overflow operation

The 16-bit counter overflows when the counter counts up from FFFFH to 0000H in the free-running mode or pulse width measurement mode. If the counter overflows, the TABnOPT0.TABnOVF bit is set to 1 and an interrupt request signal (INTTBnOV) is generated. Note that the INTTBnOV signal is not generated under the following conditions.

- Immediately after a count operation has been started
- If the counter value matches the compare value FFFFH and is cleared
- When FFFFH is captured in the pulse width measurement mode and the counter counts up from FFFFH to 0000H

Caution After the overflow interrupt request signal (INTTBnOV) has been generated, be sure to check that the overflow flag (TABnOVF bit) is set to 1.

(d) Counter read operation during count operation

The value of the 16-bit counter of TABn can be read by using the TABnCNT register during the count operation.
When the TABnCTL0.TABnCE bit = 1, the value of the 16-bit counter can be read by reading the TABnCNT register. When the TABnCE bit = 0, the 16-bit counter is FFFFH and the TABnCNT register is 0000H.

(e) Interrupt operation

TABn generates the following five interrupt request signals.

- INTTBnCC0 interrupt: This signal functions as a match interrupt request signal of the CCR0 buffer register and as a capture interrupt request signal to the TABnCCR0 register.
- INTTBnCC1 interrupt: This signal functions as a match interrupt request signal of the CCR1 buffer register and as a capture interrupt request signal to the TABnCCR1 register.
- INTTBnCC2 interrupt: This signal functions as a match interrupt request signal of the CCR2 buffer register and as a capture interrupt request signal to the TABnCCR2 register.
- INTTBnCC3 interrupt: This signal functions as a match interrupt request signal of the CCR3 buffer register and as a capture interrupt request signal to the TABnCCR3 register.
- INTTBnOV interrupt: This signal functions as an overflow interrupt request signal.

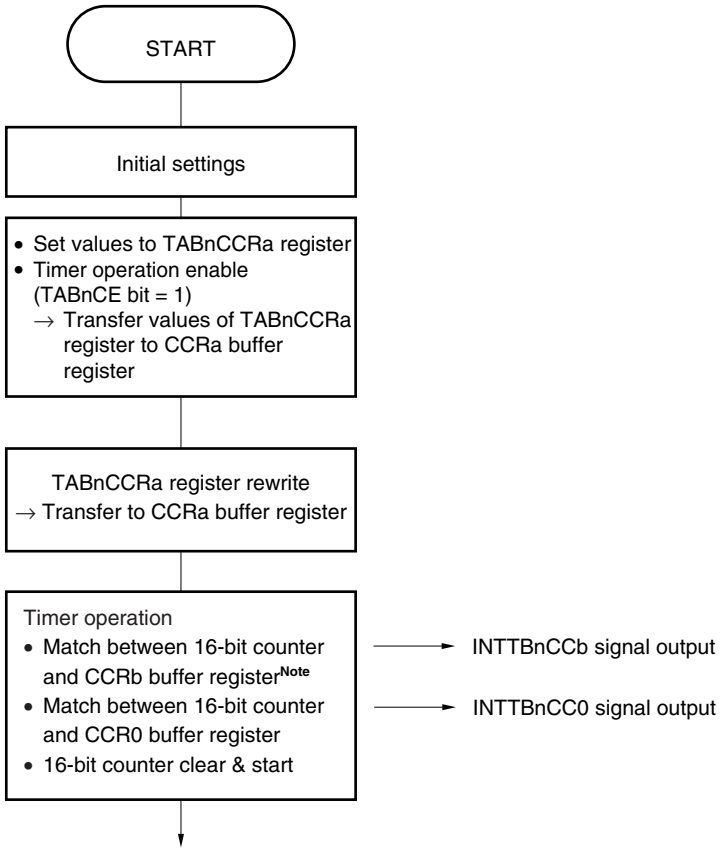
(2) Anytime write and batch write

The TABnCCR0 to TABnCCR3 registers can be rewritten in the TABn during timer operation (TABnCTL0.TABnCE bit = 1), but the write method (anytime write, batch write) of the CCR0 to CCR3 buffer registers differs depending on the mode.

(a) Anytime write

In this mode, data is transferred at any time from the TABnCCR0 to TABnCCR3 registers to the CCR0 to CCR3 buffer registers during the timer operation.

Figure 7-2. Flowchart of Basic Operation for Anytime Write

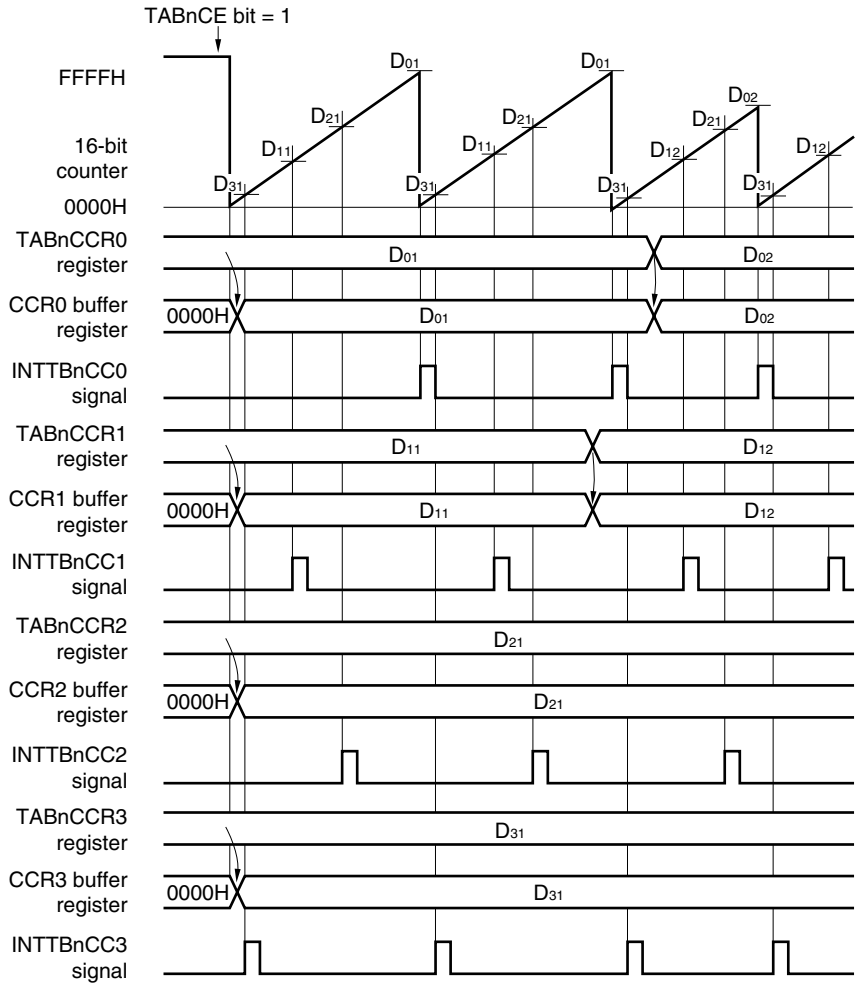


Note The 16-bit counter is not cleared upon a match between the 16-bit counter value and the CCRb buffer register value. It is cleared upon a match between the 16-bit counter value and the CCR0 buffer register value.

Remarks 1. The above flowchart illustrates an example of the operation in the interval timer mode.

2. n = 0, 1
a = 0 to 3
b = 1 to 3

Figure 7-3. Timing of Anytime Write



Remarks 1. D_{01}, D_{02}: Set values of TABnCCR0 register

D_{11}, D_{12}: Set values of TABnCCR1 register

D_{21}: Set value of TABnCCR2 register

D_{31}: Set value of TABnCCR3 register

2. The above timing chart illustrates an example of the operation in the interval timer mode.

3. n = 0, 1

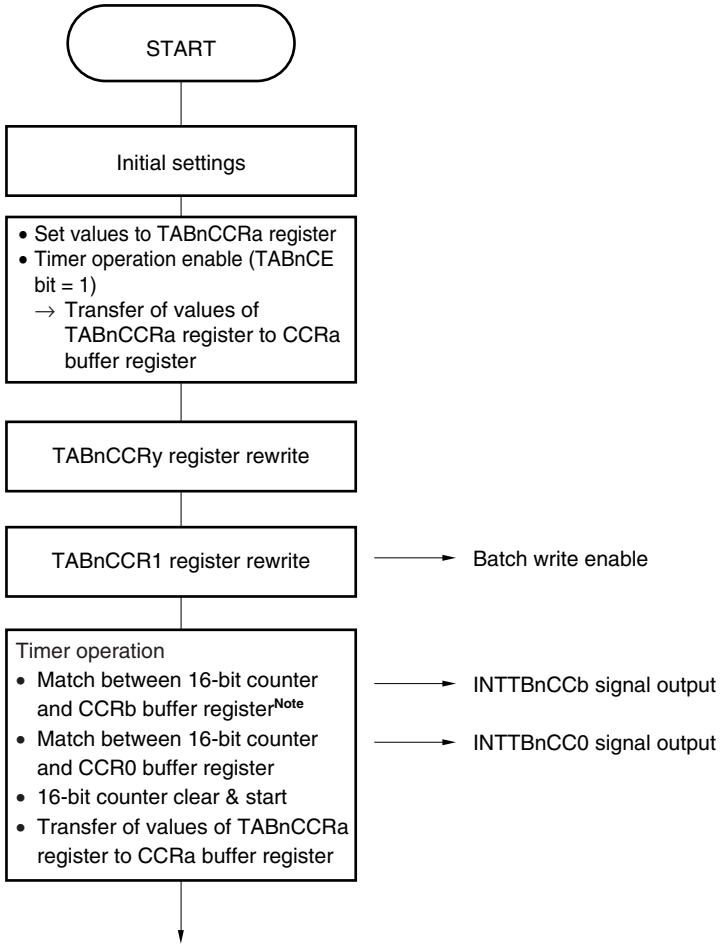
(b) Batch write

In this mode, data is transferred all at once from the TABnCCR0 to TABnCCR3 registers to the CCR0 to CCR3 buffer registers during timer operation. This data is transferred upon a match between the value of the CCR0 buffer register and the value of the 16-bit counter. Transfer is enabled by writing to the TABnCCR1 register.

Whether to enable or disable the next transfer timing is controlled by writing or not writing to the TABnCCR1 register.

In order for the set value when the TABnCCR0 to TABnCCR3 registers are rewritten to become the 16-bit counter comparison value (in other words, in order for this value to be transferred to the CCR0 to CCR3 buffer registers), it is necessary to rewrite TABnCCR0 and finally write to the TABnCCR1 register before the 16-bit counter value and the CCR0 buffer register value match. The values of the TABnCCR0 to TABnCCR3 registers are transferred to the CCR0 to CCR3 buffer registers upon a match between the count value of the 16-bit counter and the value of the CCR0 buffer register. Thus, even when wishing only to rewrite the value of the TABnCCR0, TABnCCR2, or TABnCCR3 register, also write the same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register.

Figure 7-4. Flowchart of Basic Operation for Batch Write



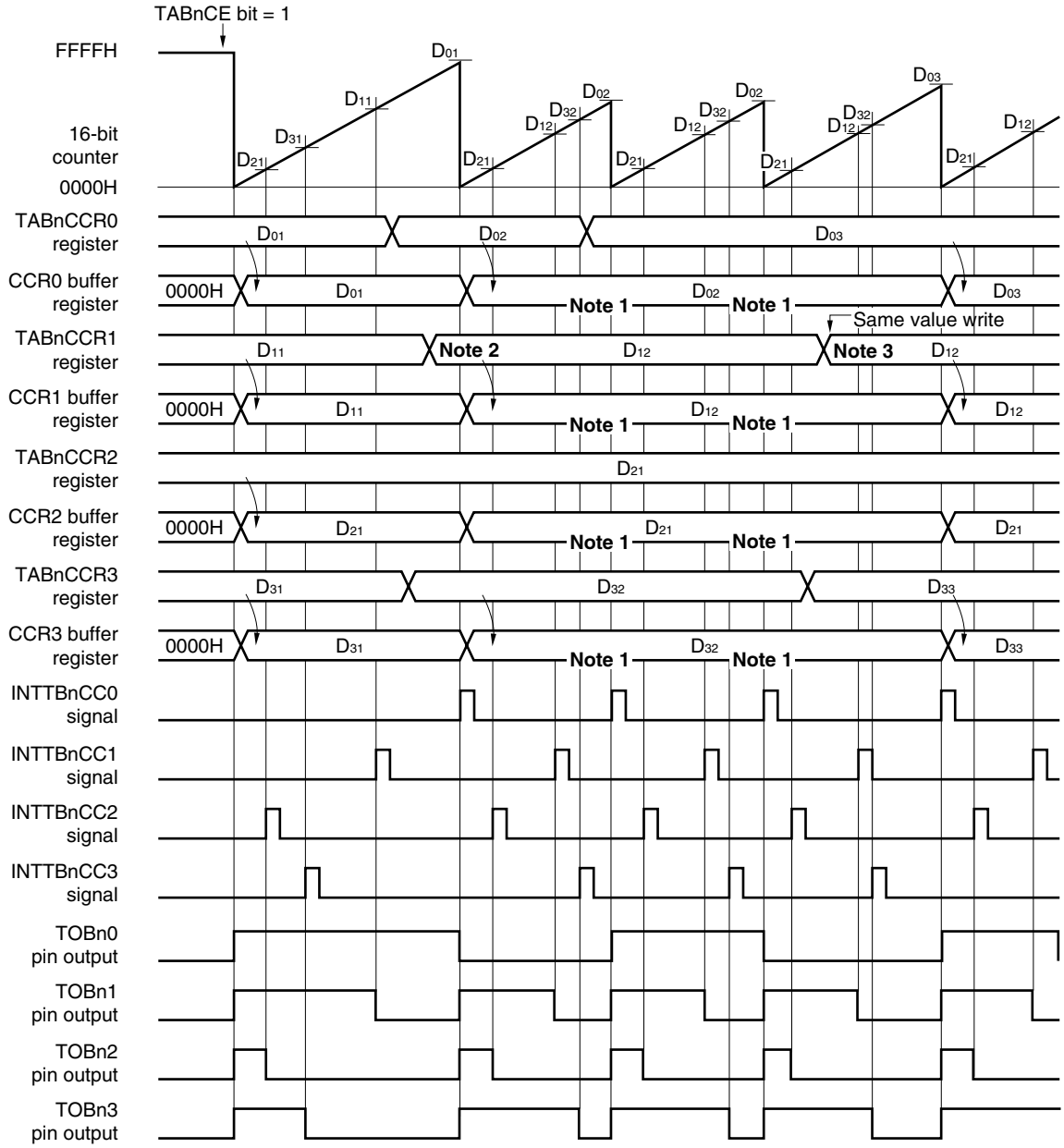
Note The 16-bit counter is not cleared upon a match between the 16-bit counter value and the CCRb buffer register value. It is cleared upon a match between the 16-bit counter value and the CCR0 buffer register value.

Caution Writing to the TABnCCR1 register includes enabling of batch write. Thus, rewrite the TABnCCR1 register after rewriting the TABnCCR0, TABnCCR2, and TABnCCR3 registers.

Remarks 1. The above flowchart illustrates an example of the operation in the PWM output mode.

2. a = 0 to 3
b = 1 to 3
y = 0, 2, 3

Figure 7-5. Timing of Batch Write



Notes 1. Because the TABnCCR1 register was not rewritten, D_{02} is not transferred.

2. Because TABnCCR1 register has been written (D_{12}), data is transferred to the CCR1 buffer register upon a match between the value of the 16-bit timer and the value of the TABnCCR0 register (D_{01}).

3. Because TABnCCR1 register has been written (D_{12}), data is transferred to the CCR1 buffer register upon a match between the value of the 16-bit timer and the value of the TABnCCR0 register (D_{12}).

Remarks 1. D_{01}, D_{02}, D_{03}: Set values of TABnCCR0 register
D_{11}, D_{12}: Set values of TABnCCR1 register
D_{21}: Set value of TABnCCR2 register
D_{31}, D_{32}, D_{33}: Set values of TABnCCR3 register

2. The above timing chart illustrates the operation in the PWM output mode as an example.

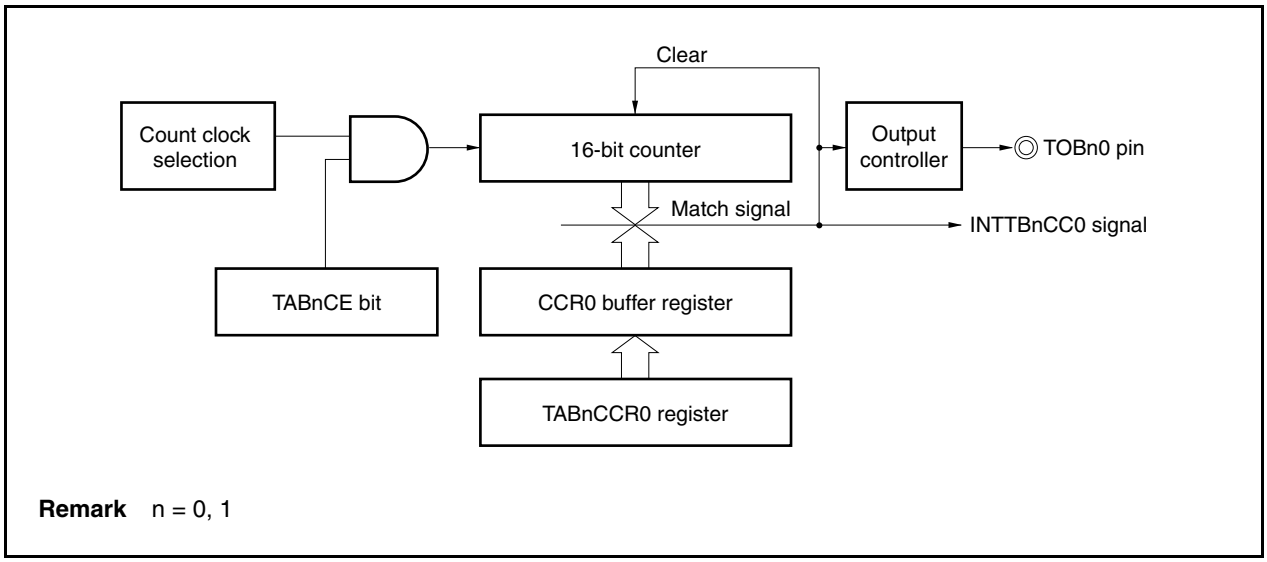
7.6.1 Interval timer mode (TABnMD2 to TABnMD0 bits = 000)

In the interval timer mode, an interrupt request signal (INTTBnCC0) is generated at the interval set by the TABnCCR0 register if the TABnCTL0.TABnCE bit is set to 1. A PWM waveform with a duty factor of 50% whose half cycle is equal to the interval can be output from the TOBn0 pin.

The TABnCCR1 to TABnCCR3 registers are not used in the interval timer mode. However, the set value of the TABnCCR1 to TABnCCR3 registers is transferred to the CCR1 to CCR3 buffer registers and, when the count value of the 16-bit counter matches the value of the CCR1 to CCR3 buffer registers, compare match interrupt request signals (INTTBnCC1 to INTTBnCC3) are generated. In addition, a PWM waveform with a duty factor of 50%, which is inverted when the INTTBnCC1 to INTTBnCC3 signals are generated, can be output from the TOBn1 to TOBn3 pins.

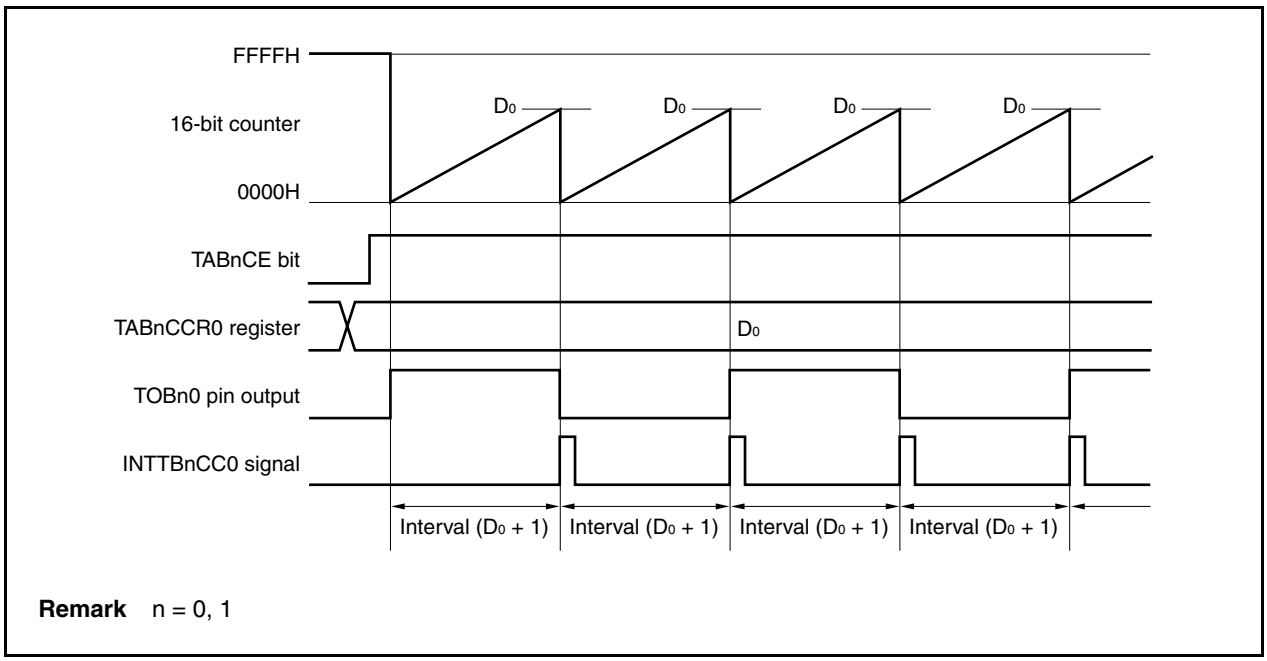
The value of the TABnCCR1 to TABnCCR3 registers can be rewritten even while the timer is operating.

Figure 7-6. Interval Timer Configuration



Remark n = 0, 1

Figure 7-7. Basic Timing of Operation in Interval Timer Mode



Remark n = 0, 1

When the TABnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H in synchronization with the count clock, and the counter starts counting. At this time, the output of the TOBn0 pin is inverted. Additionally, the set value of the TABnCCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, the output of the TOBn0 pin is inverted, and a compare match interrupt request signal (INTTBnCC0) is generated.

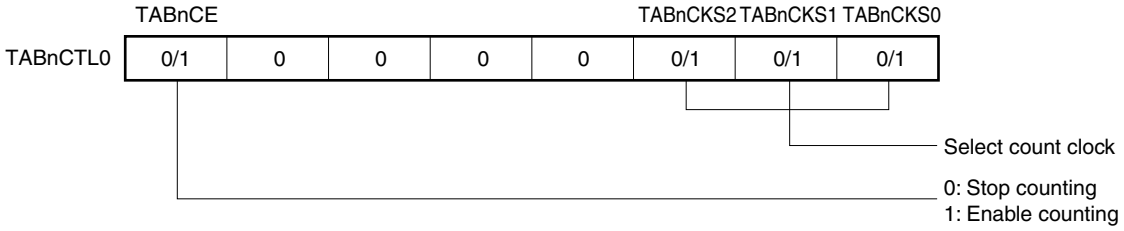
The interval can be calculated by the following expression.

Interval = (Set value of TABnCCR0 register + 1) × Count clock cycle

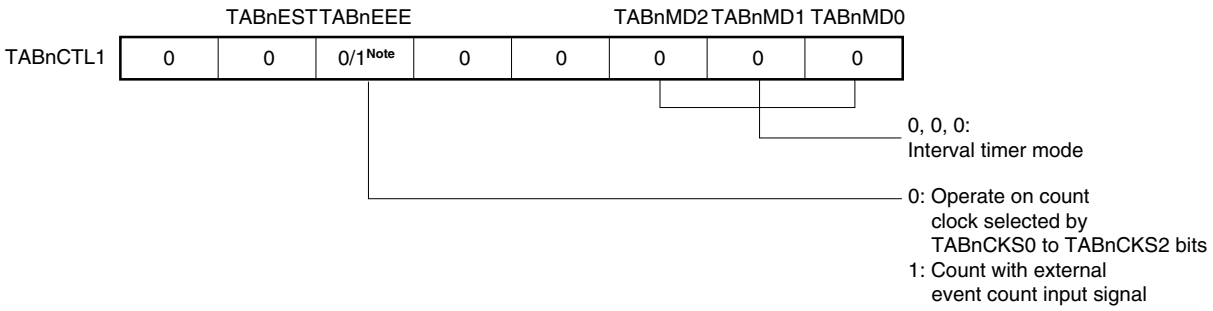
Remark n = 0, 1

Figure 7-8. Register Setting for Interval Timer Mode Operation (1/3)

(a) TABn control register 0 (TABnCTL0)



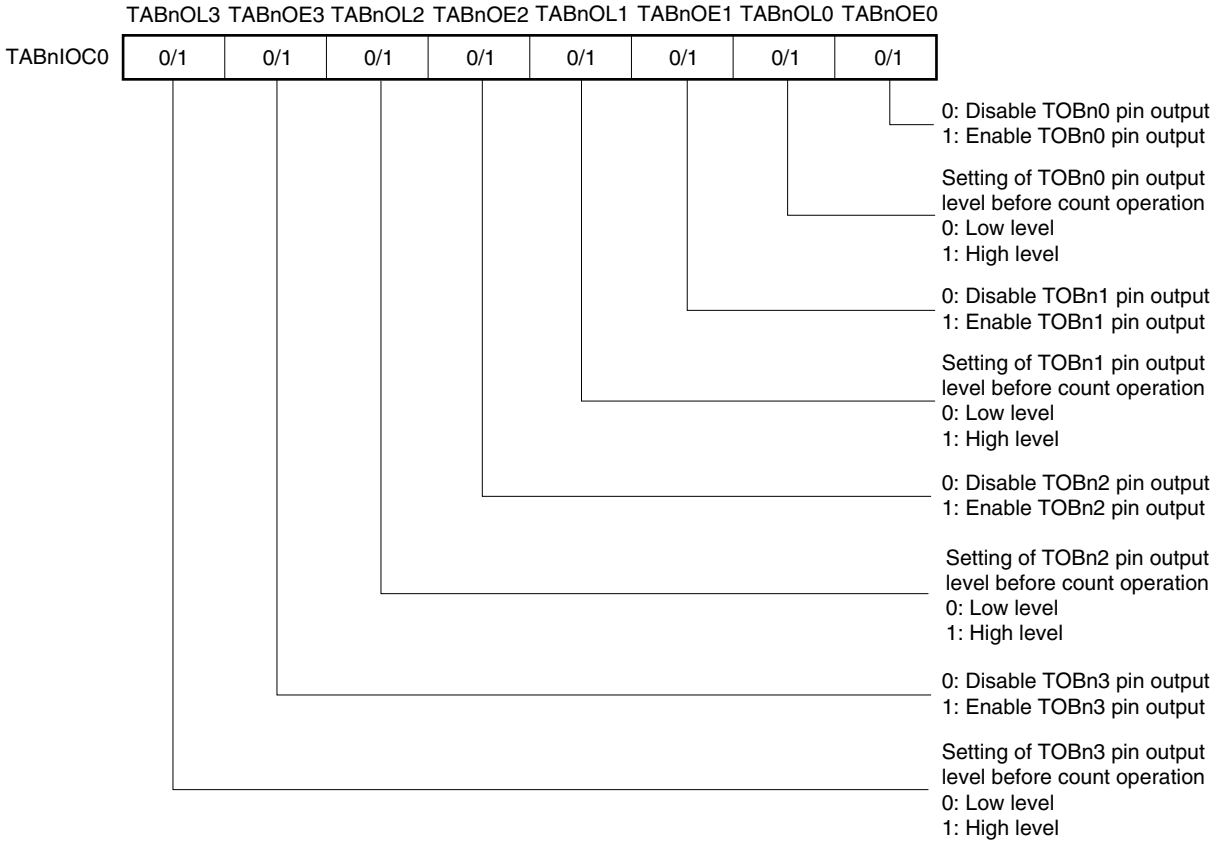
(b) TABn control register 1 (TABnCTL1)



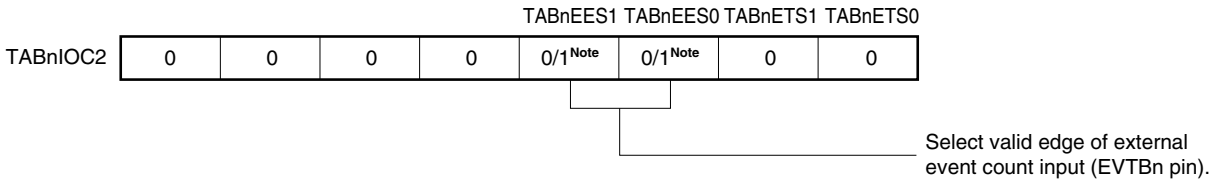
Note The TABnEEE bit can be set to 1 only when timer output (TOBn0 to TOBn3) is used. However, set the TABnCCR0 to TABnCCR3 registers to the same value.

Figure 7-8. Register Setting for Interval Timer Mode Operation (2/3)

(c) TABn I/O control register 0 (TABnIOC0)



(d) TABn I/O control register 2 (TABnIOC2)



Note The TABnEES1 and TABnEES0 bits can be set only when timer output (TOBn0 to TOBn3) is used. However, set the TABnCCR0 to TABnCCR3 registers to the same value.

(e) TABn counter read buffer register (TABnCNT)

By reading the TABnCNT register, the count value of the 16-bit counter can be read.

(f) TABn capture/compare register 0 (TABnCCR0)

If the TABnCCR0 register is set to D_0, the interval is as follows.

Interval = $(D_0 + 1) \times$ Count clock cycle

Figure 7-8. Register Setting for Interval Timer Mode Operation (3/3)

(g) TABn capture/compare registers 1 to 3 (TABnCCR1 to TABnCCR3)

The TABnCCR1 to TABnCCR3 registers are not used in the interval timer mode. However, the set values of the TABnCCR1 to TABnCCR3 registers are transferred to the CCR1 to CCR3 buffer registers. When the count value of the 16-bit counter matches the value of the CCR1 to CCR3 buffer registers, the TOBn1 to TOBn3 pin outputs are inverted and a compare match interrupt request signal (INTTBnCC1 to INTTBnCC3) is generated.

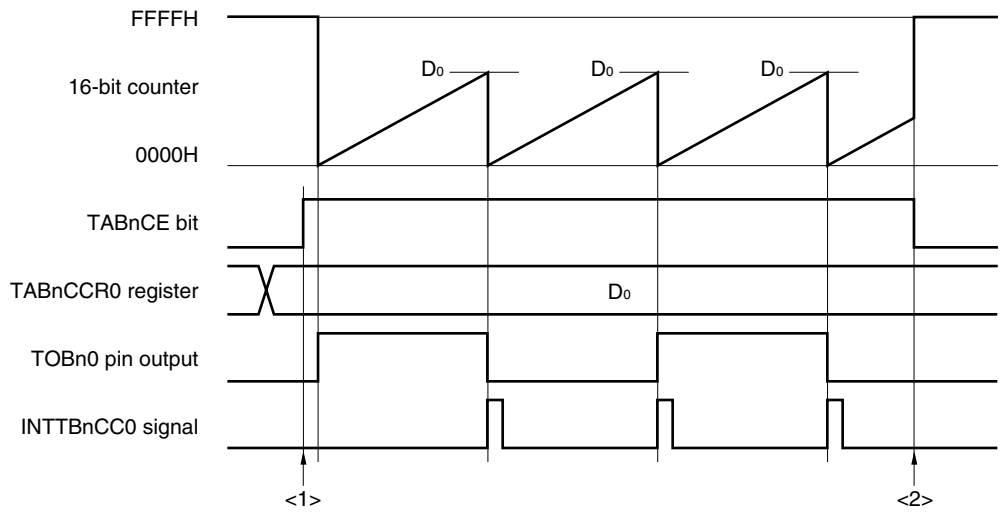
When the TABnCCR1 to TABnCCR3 registers are not used, it is recommended to set their values to FFFFH. Also mask the registers by the interrupt mask flags (TABnCCIC1.TABnCCMK1 to TABnCCIC3.TABnCCMK3).

Remarks 1. TABn I/O control register 1 (TABnIOC1) and TABn option register 0 (TABnOPT0) are not used in the interval timer mode.

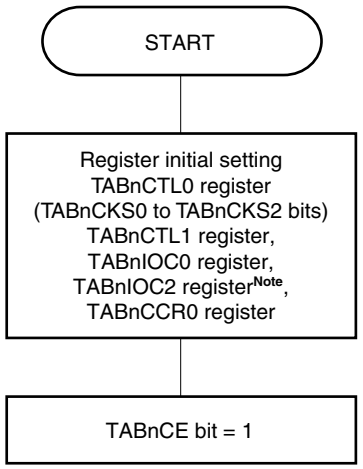
2. n = 0, 1

(1) Interval timer mode operation flow

Figure 7-9. Software Processing Flow in Interval Timer Mode (1/2)



<1> Count operation start flow



Initial setting of these registers is performed before setting the TABnCE bit to 1.

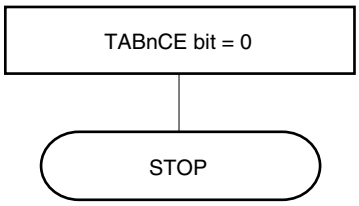
The TABnCKS0 to TABnCKS2 bits can be set at the same time as when counting starts (TABnCE bit = 1).

Note The TABnEES1 and TABnEES0 bits can be set only when timer output (TOBn0 to TOBn3) is used. However, set the TABnCCR0 to TABnCCR3 registers to the same value.

Remark n = 0, 1

Figure 7-9. Software Processing Flow in Interval Timer Mode (2/2)

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TABnCE bit to 0. The output level of the TOBn0 pin is as specified by the TABnIOC0 register.

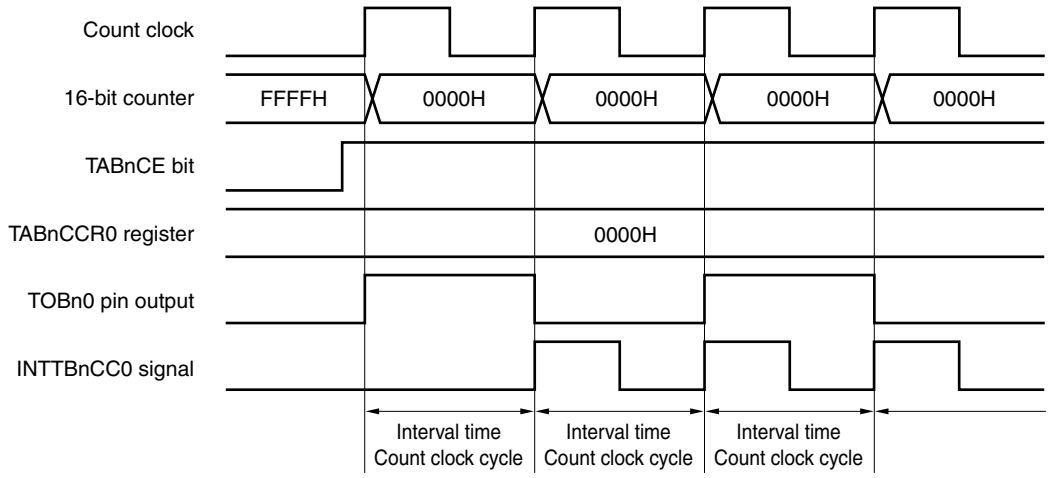
Remark n = 0, 1

(2) Interval timer mode operation timing

(a) Operation if TABnCCR0 register is set to 0000H

If the TABnCCR0 register is set to 0000H, the INTTBnCC0 signal is generated at each count clock, and the output of the TOBn0 pin is inverted.

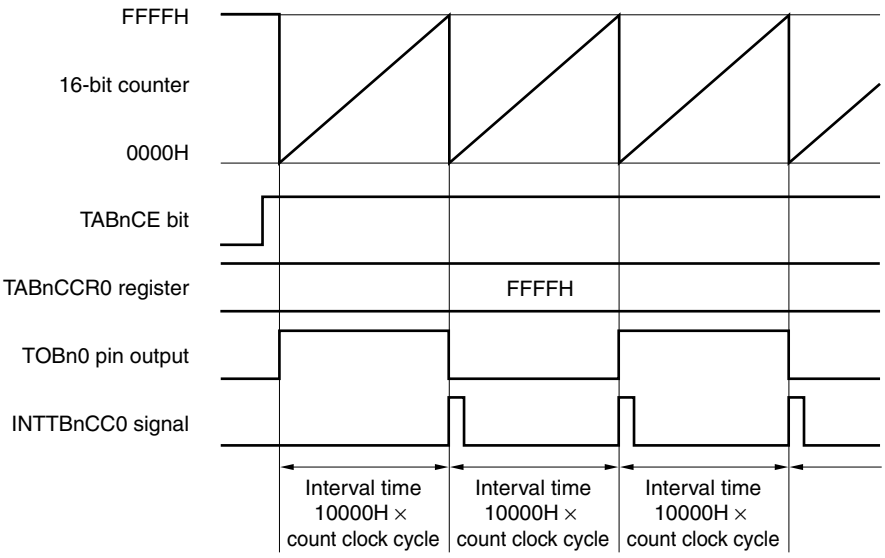
The value of the 16-bit counter is always 0000H.



Remark n = 0, 1

(b) Operation if TABnCCR0 register is set to FFFFH

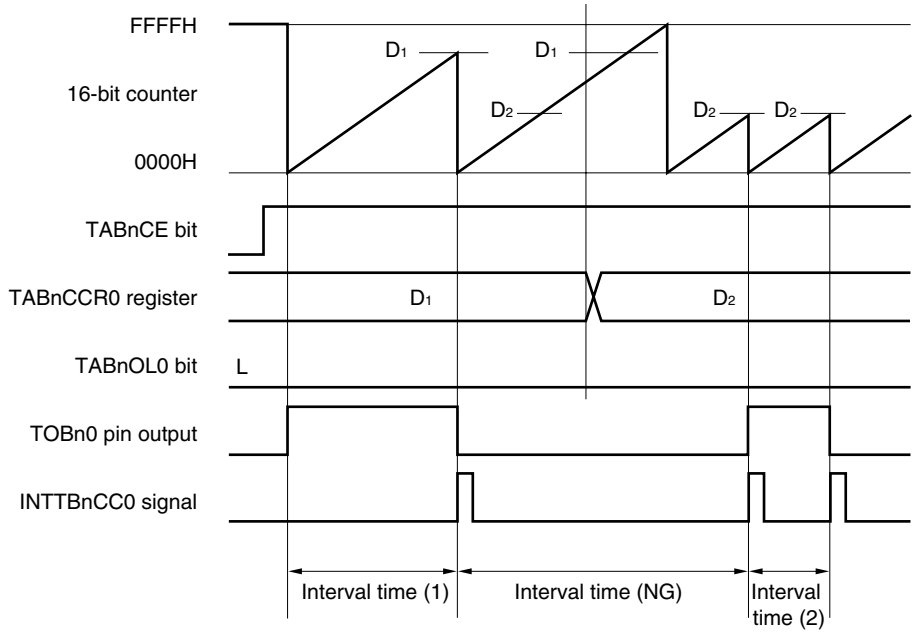
If the TABnCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTBnCC0 signal is generated and the output of the TOBn0 pin is inverted. At this time, an overflow interrupt request signal (INTTBnOV) is not generated, nor is the overflow flag (TABnOPT0.TABnOVF bit) set to 1.



Remark n = 0, 1

(c) Notes on rewriting TABnCCR0 register

If the value of the TABnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When the overflow may occur, stop counting once and then change the set value.



Remarks 1. Interval time (1): $(D_1 + 1) \times$ Count clock cycle
Interval time (NG): $(10000H + D_2 + 1) \times$ Count clock cycle
Interval time (2): $(D_2 + 1) \times$ Count clock cycle

2. n = 0, 1

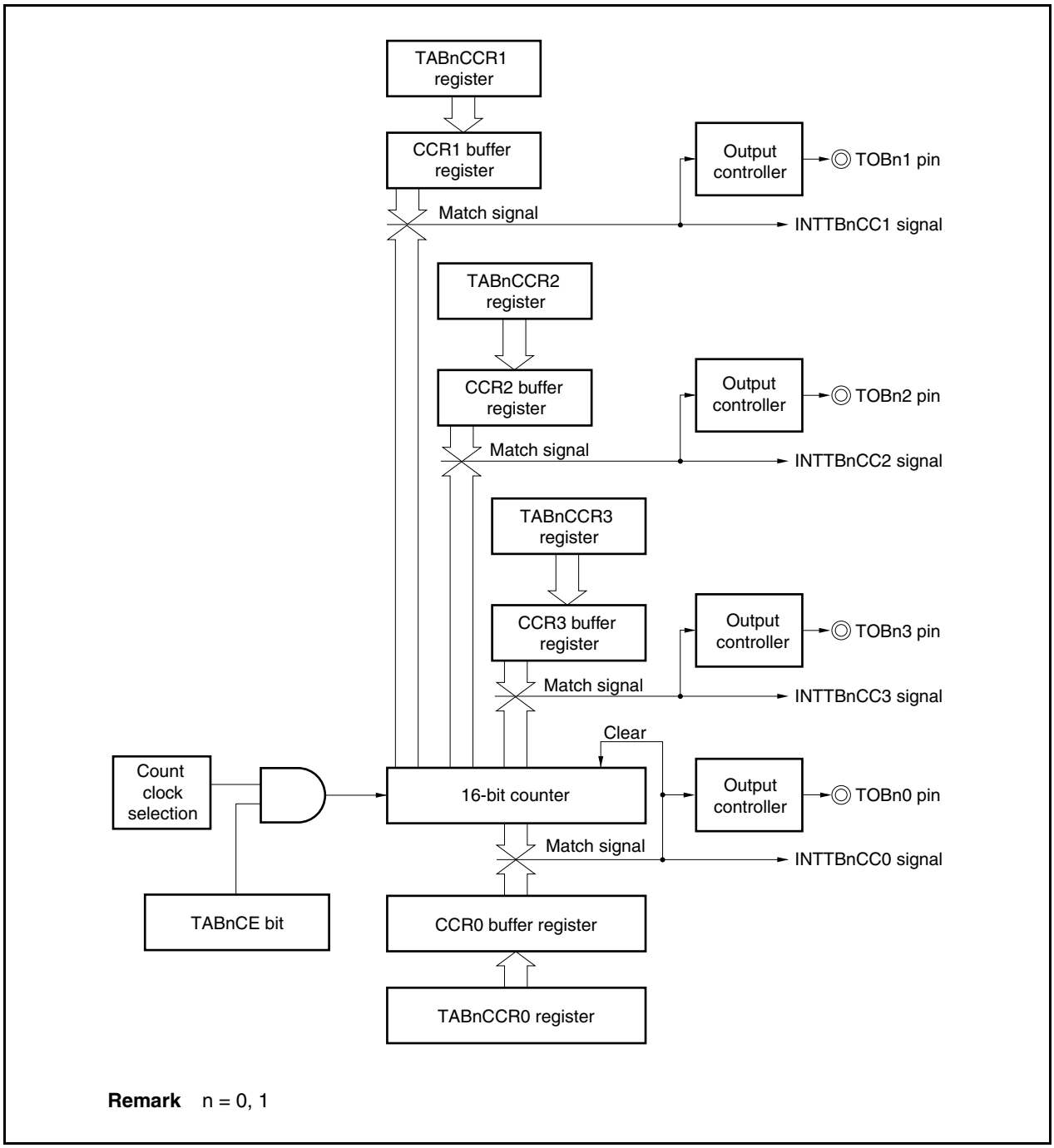
If the value of the TABnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TABnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTBnCC0 signal is generated and the output of the TOBn0 pin is inverted.

Therefore, the INTTBnCC0 signal may not be generated at the interval time "$(D_1 + 1) \times$ Count clock cycle" or "$(D_2 + 1) \times$ Count clock cycle" originally expected, but may be generated at an interval of "$(10000H + D_2 + 1) \times$ Count clock period".

(d) Operation of TABnCCR1 to TABnCCR3 registers

Figure 7-10. Configuration of TABnCCR1 to TABnCCR3 Registers



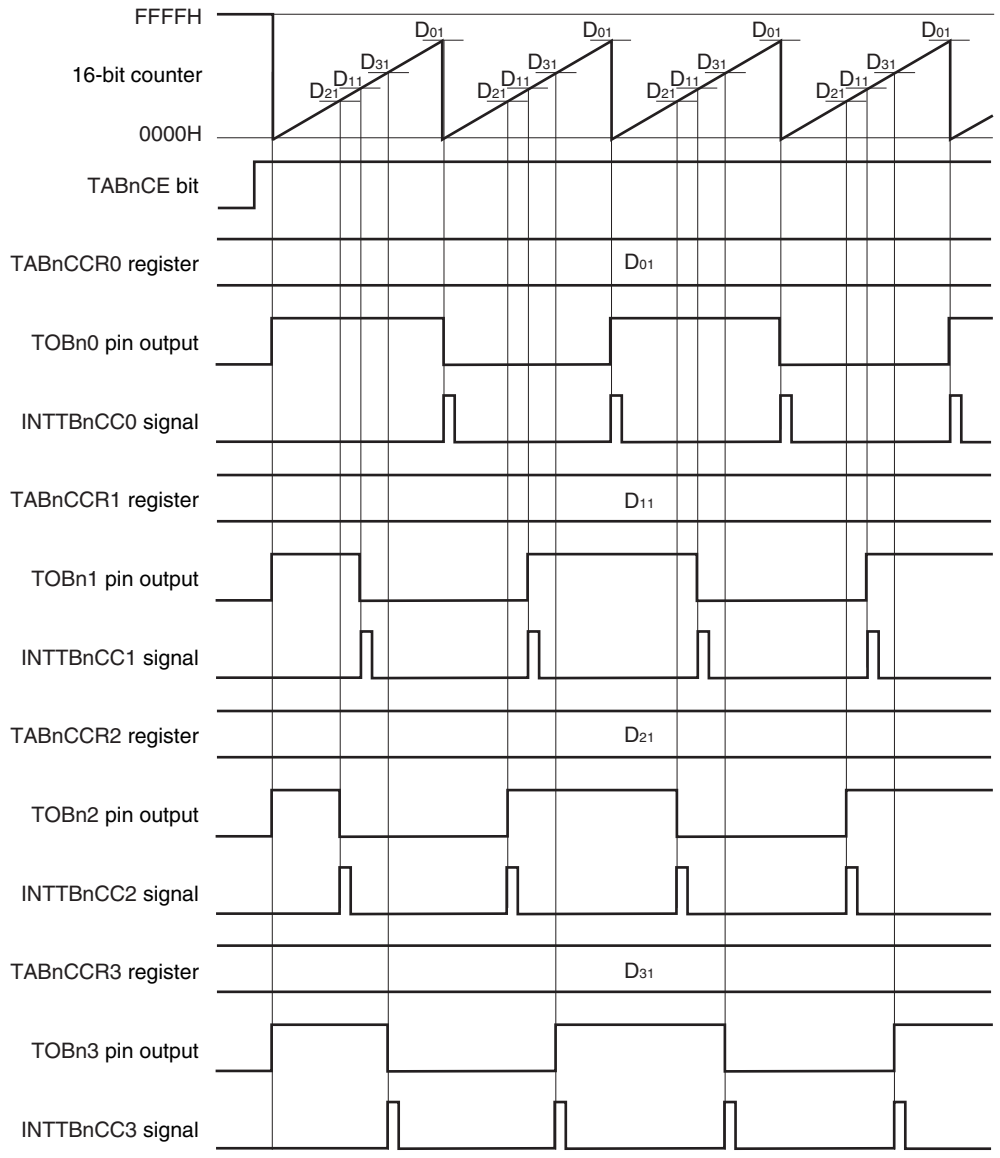
Remark n = 0, 1

When the TABnCCRb register is set to the same value as the TABnCCR0 register, the INTTBnCCb signal is generated at the same timing as the INTTBnCC0 signal and the TOBnb pin output is inverted. In other words, a PWM waveform with a duty factor of 50% can be output from the TOBnb pin.

The following shows the operation when the TABnCCRb register is set to other than the value set in the TABnCCR0 register.

If the set value of the TABnCCRb register is less than the set value of the TABnCCR0 register, the INTTBnCCb signal is generated once per cycle. At the same time, the output of the TOBnb pin is inverted. The TOBnb pin outputs a PWM waveform with a duty factor of 50% after outputting a short-width pulse.

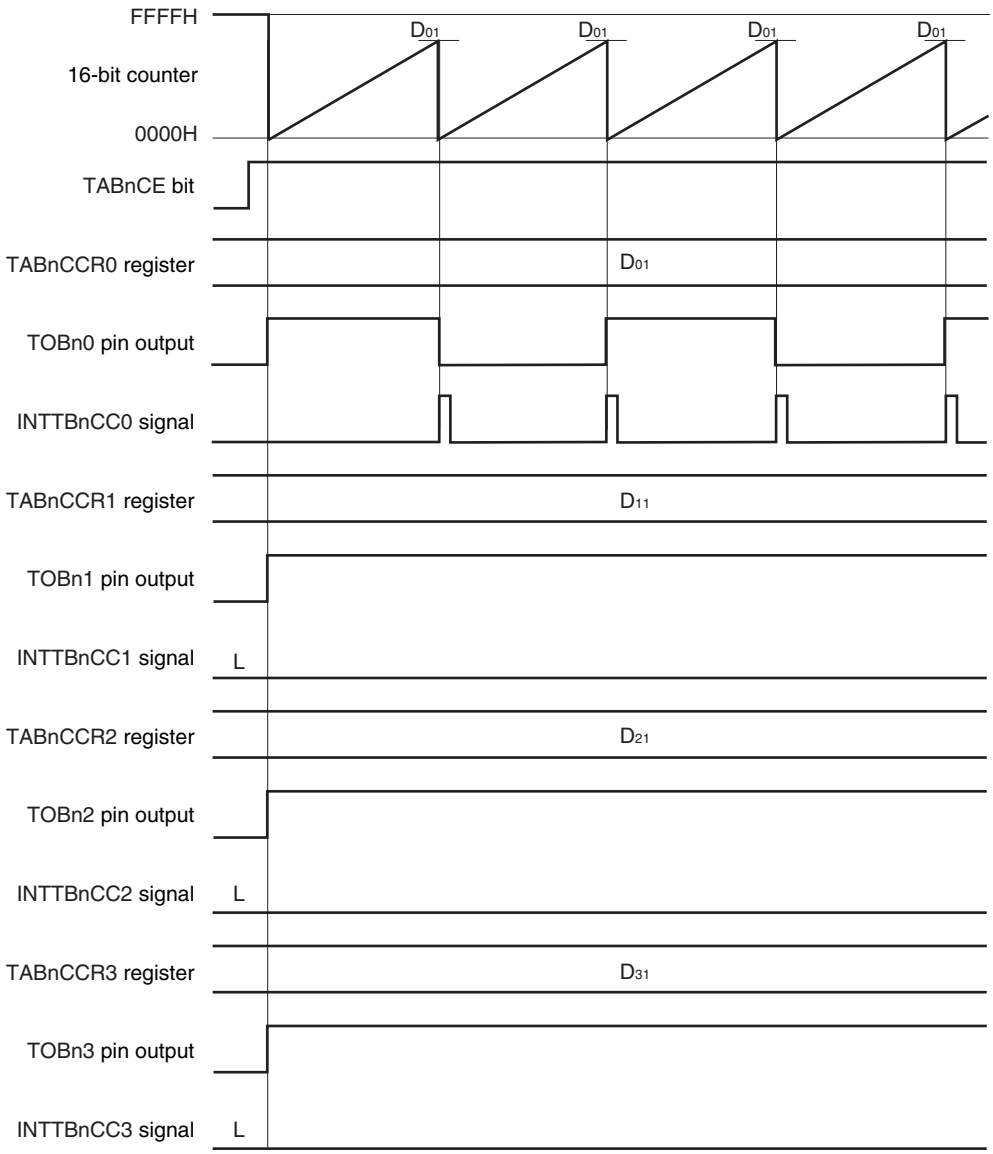
Figure 7-11. Timing Chart When $D_{01} \geq D_{b1}$



Remark n = 0, 1
b = 1 to 3

If the set value of the TABnCCRb register is greater than the set value of the TABnCCR0 register, the count value of the 16-bit counter does not match the value of the TABnCCRb register. Consequently, the INTTBnCCb signal is not generated, nor is the output of the TOBnb pin changed.
When the TABnCCRb register is not used, it is recommended to set its value to FFFFH.

Figure 7-12. Timing Chart When $D_{01} < D_{b1}$



Remark n = 0, 1
b = 1 to 3

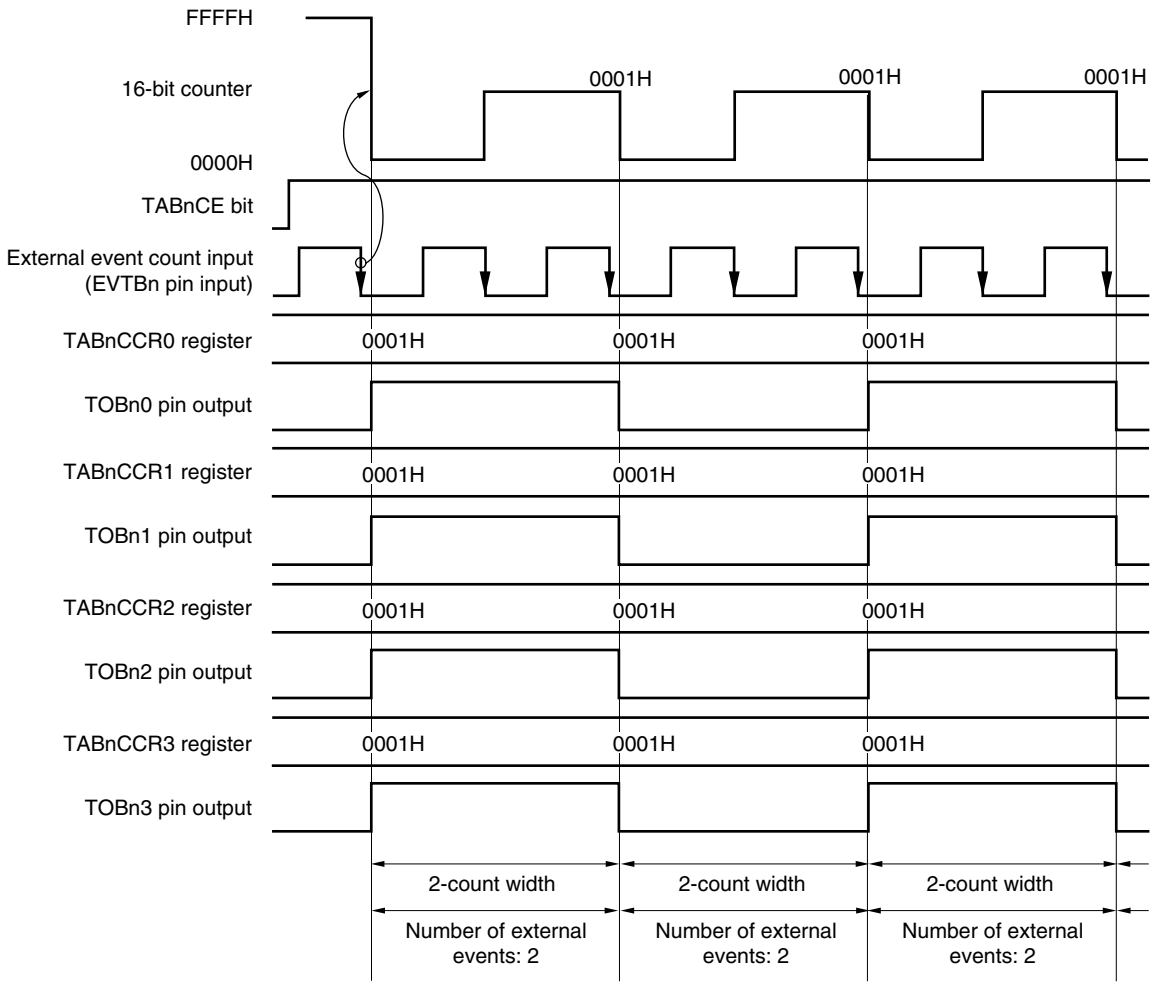
(3) Operation by external event count input (EVTBn)

(a) Operation

To count the 16-bit counter at the valid edge of the external event count input (EVTBn) in the interval timer mode, the 16-bit counter is cleared from FFFFH to 0000H by the valid edge of the external event count input after the TABnCE bit is set from 0 to 1.

When 0001H is set to both the TABnCCR0 and TABnCCRb registers, the output of the TOBn0 and TOBnb pins is inverted each time the 16-bit counter counts twice (b = 1 to 3).

The TABnCTL0.TABnEEE bit can be set to 1 in the interval timer mode only when the timer output (TOBn0, TOBnb) is used with the external event count input.



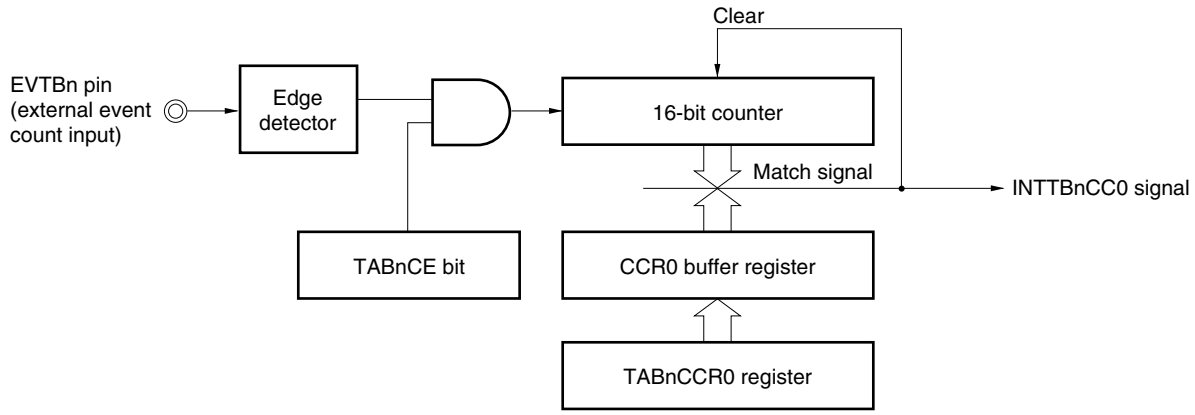
Remark n = 0, 1
b = 1 to 3

7.6.2 External event count mode (TABnMD2 to TABnMD0 bits = 001)

In the external event count mode, the valid edge of the external event count input (EVTBn) is counted when the TABnCTL0.TABnCE bit is set to 1, and an interrupt request signal (INTTBnCC0) is generated each time the specified number of edges set by the TABnCCR0 register have been counted. The TOBn0 to TOBn3 pins cannot be used. When using the TOBn0 to TOBn3 pins for external event count input, set the TABnCTL1.TABnEEE bit to 1 in the interval timer mode (see **7.6.1 (3) Operation by external event count input (EVTBn)**).

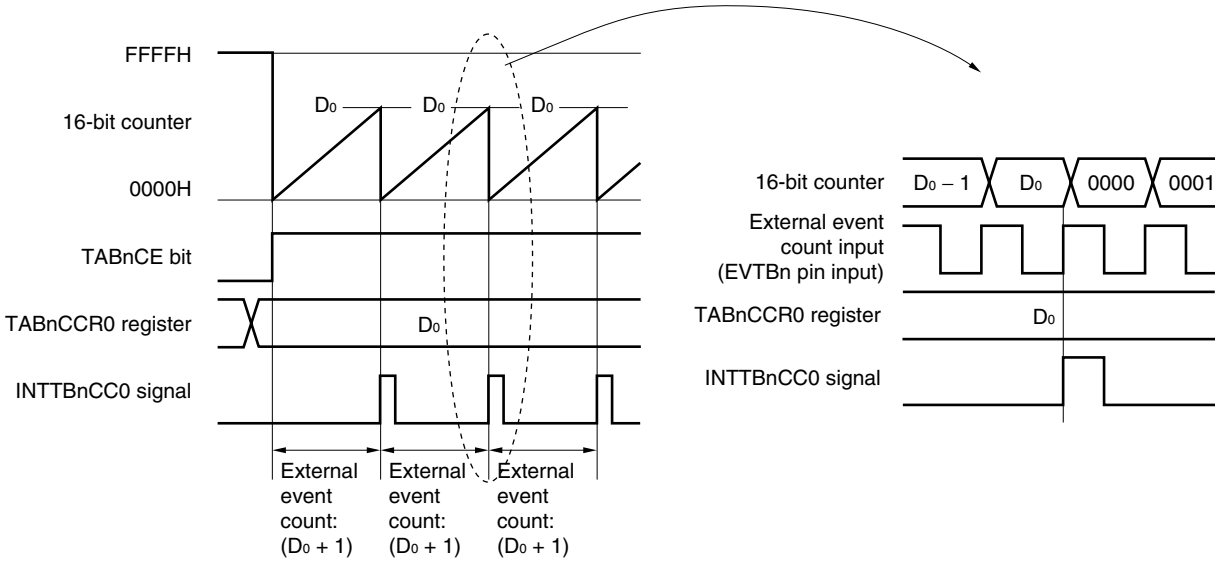
The TABnCCR1 to TABnCCR3 registers are not used in the external event count mode.

Figure 7-13. Configuration in External Event Count Mode



Remark n = 0, 1

Figure 7-14. Basic Timing in External Event Count Mode



Remarks 1. This figure shows the basic timing when the rising edge is specified as the valid edge of the external event count input.

2. n = 0, 1

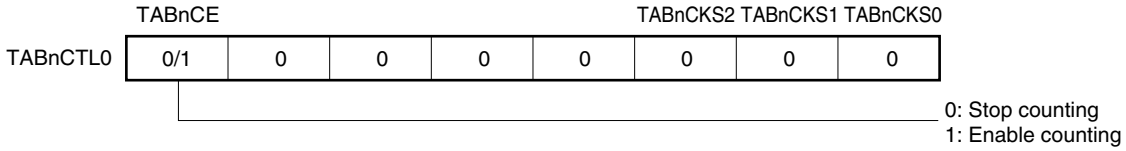
When the TABnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H. The counter counts each time the valid edge of external event count input is detected. Additionally, the set value of the TABnCCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTBnCC0) is generated.

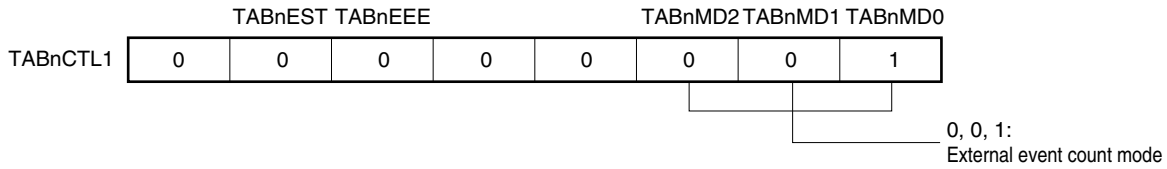
The INTTBnCC0 signal is generated each time the valid edge of the external event count has been detected "value set to TABnCCR0 register + 1" times.

Figure 7-15. Register Setting for Operation in External Event Count Mode (1/2)

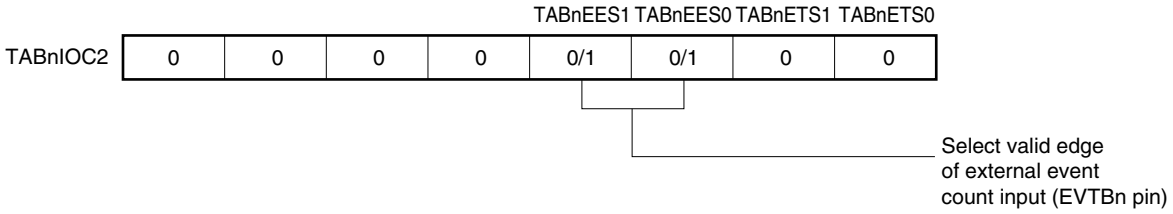
(a) TABn control register 0 (TABnCTL0)



(b) TABn control register 1 (TABnCTL1)



(c) TABn I/O control register 2 (TABnIOC2)



(d) TABn counter read buffer register (TABnCNT)

The count value of the 16-bit counter can be read by reading the TABnCNT register.

(e) TABn capture/compare register 0 (TABnCCR0)

If the TABnCCR0 register is set to D_0, the compare match interrupt request signal (INTTBnCC0) is generated when the number of external events has reached (D_0 + 1).

Figure 7-15. Register Setting for Operation in External Event Count Mode (2/2)

(f) TABn capture/compare registers 1 to 3 (TABnCCR1 to TABnCCR3)

The TABnCCR1 to TABnCCR3 registers are not used in the external event count mode. However, the set value of the TABnCCR1 to TABnCCR3 registers are transferred to the CCR1 to CCR3 buffer registers. When the count value of the 16-bit counter matches the value of the CCR1 to CCR3 buffer registers, compare match interrupt request signals (INTTBnCC1 to INTTBnCC3) are generated.

When the TABnCCR1 to TABnCCR3 registers are not used, it is recommended to set their values to FFFFH. Also mask the registers by the interrupt mask flags (TABnCCIC1.TABnCCMK1 to TABnCCIC3.TABnCCMK3).

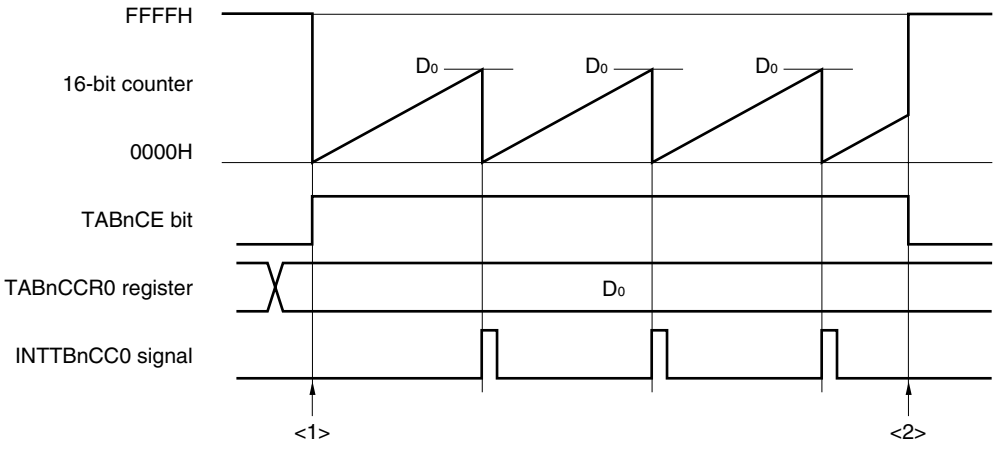
Caution Set the TABnIOC0 register to 00H.

Remarks 1. TABn I/O control register 1 (TABnIOC1) and TABn option register 0 (TABnOPT0) are not used in the external event count mode.

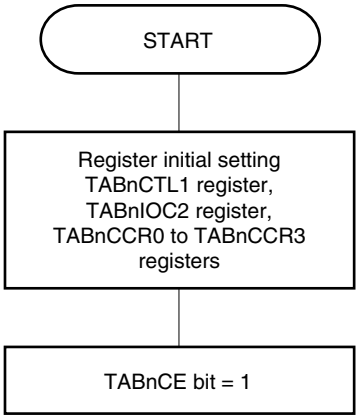
2. n = 0, 1

(1) External event count mode operation flow

Figure 7-16. Software Processing Flow in External Event Count Mode

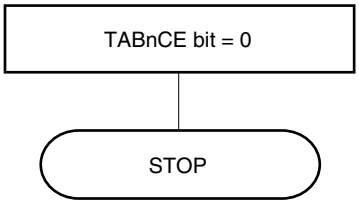


<1> Count operation start flow



Initial setting of these registers is performed before setting the TABnCE bit to 1.

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TABnCE bit to 0.

Remark n = 0, 1

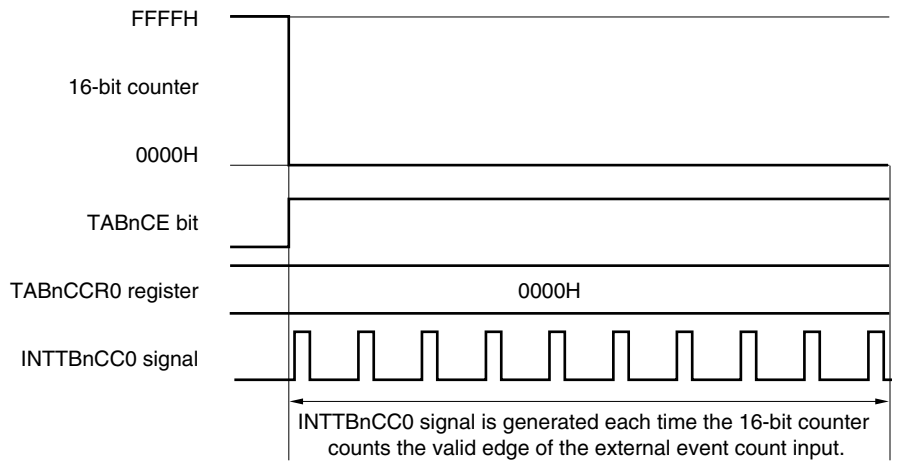
(2) Operation timing in external event count mode

Caution In the external event count mode, use of the timer output (TOBn0 to TOBn3) is disabled. If using timer output (TOBn0 to TOBn3) with external event count input (EVTBn), set the interval timer mode, and select the operation enabled by the external event count input for the count clock (TABnCTL1.TABnEEE bit = 1) (see 7.6.1 (3) Operation by external event count input (EVTBn)).

(a) Operation if TABnCCR0 register is set to 0000H

When the TABnCCR0 register is set to 0000H, the 16-bit counter is repeatedly cleared to 0000H and generates an INTTBnCC0 signal each time it has detected the valid edge of the external event count signal and its value has matched that of the CCR0 buffer register.

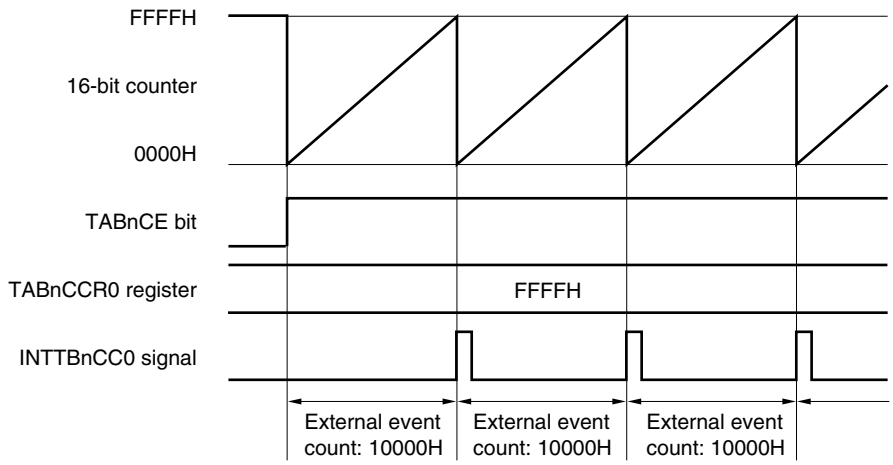
The value of the 16-bit counter is always 0000H.



Remark n = 0, 1

(b) Operation if TABnCCR0 register is set to FFFFH

If the TABnCCR0 register is set to FFFFH, the 16-bit counter counts to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTTBnCC0 signal is generated. At this time, the TABnOPT0.TABnOVF bit is not set.

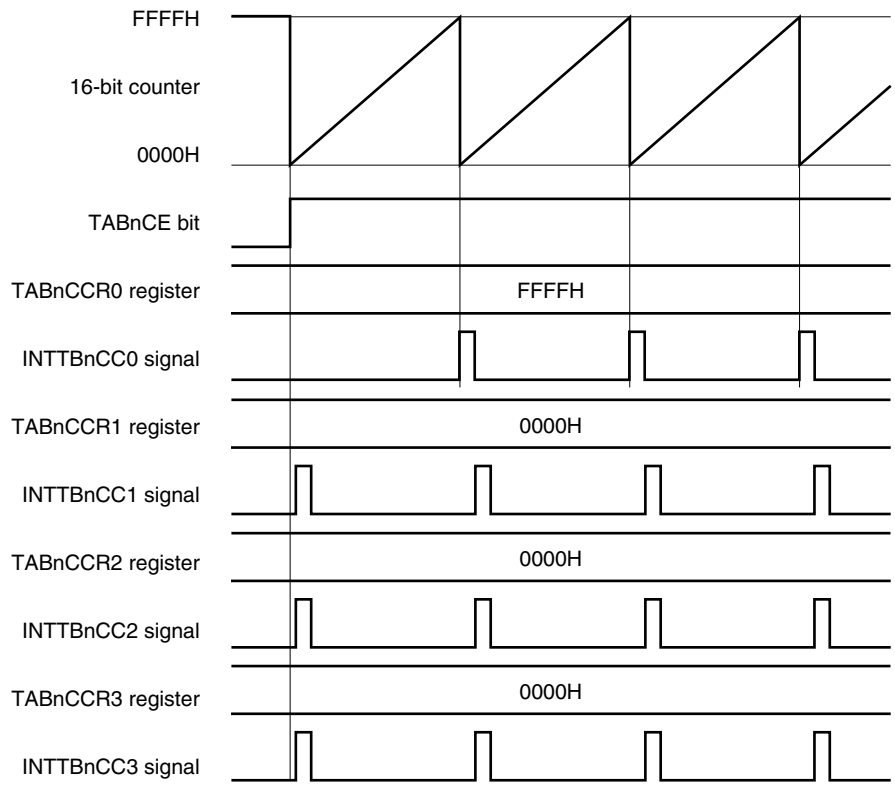


Remark n = 0, 1

(c) Operation with TABnCCR0 set to FFFFH and TABnCCRb register to 0000H

When the TABnCCR0 register is set to FFFFH, the 16-bit counter counts to FFFFH each time it has detected the valid edge of the external event count signal. The counter is then cleared to 0000H in synchronization with the next count-up timing and the INTTBnCC0 signal is generated. At this time, the TABnOPT0.TABnOVF bit is not set.

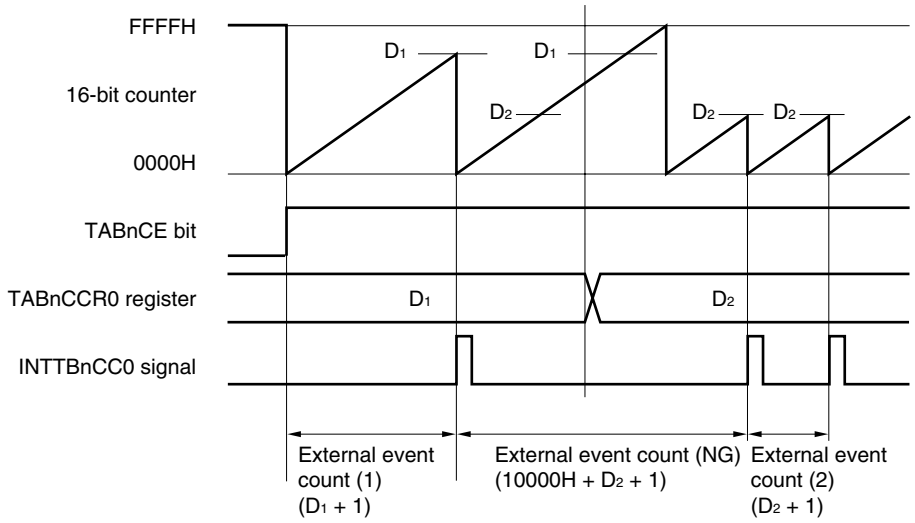
If the TABnCCRb register is set to 0000H, the INTTBnCCb signal is generated when the 16-bit counter is cleared to 0000H.



Remark n = 0, 1
b = 1 to 3

(d) Notes on rewriting the TABnCCR0 register

If the value of the TABnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When the overflow may occur, stop counting once and then change the set value.



Remark n = 0, 1

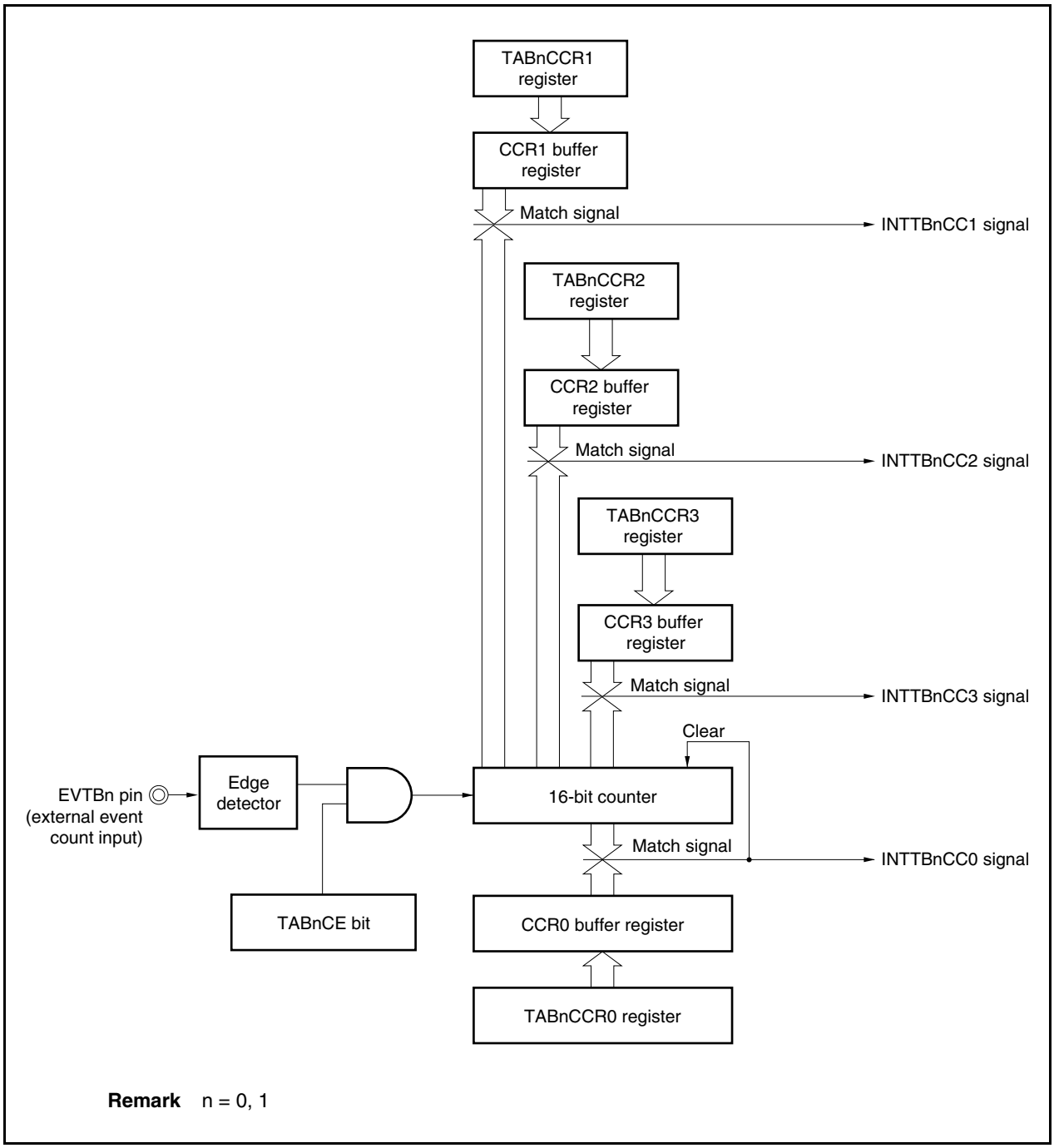
If the value of the TABnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TABnCCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTBnCC0 signal is generated.

Therefore, the INTTBnCC0 signal may not be generated at the valid edge count of "(D_1 + 1) times" or "(D_2 + 1) times" originally expected, but may be generated at the valid edge count of "(10000H + D_2 + 1) times".

(e) Operation of TABnCCR1 to TABnCCR3 registers

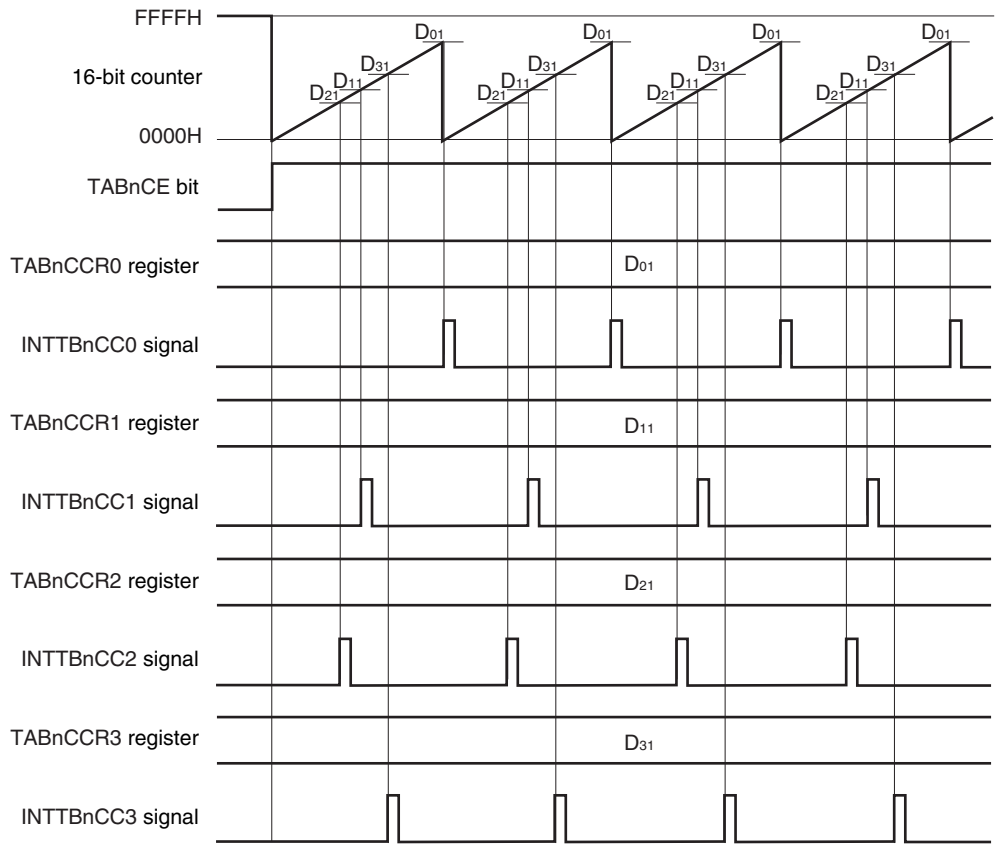
Figure 7-17. Configuration of TABnCCR1 to TABnCCR3 Registers



Remark n = 0, 1

If the set value of the TABnCCRb register is smaller than the set value of the TABnCCR0 register, the INTTBnCCb signal is generated once per cycle.

Figure 7-18. Timing Chart When $D_{01} \geq D_{b1}$

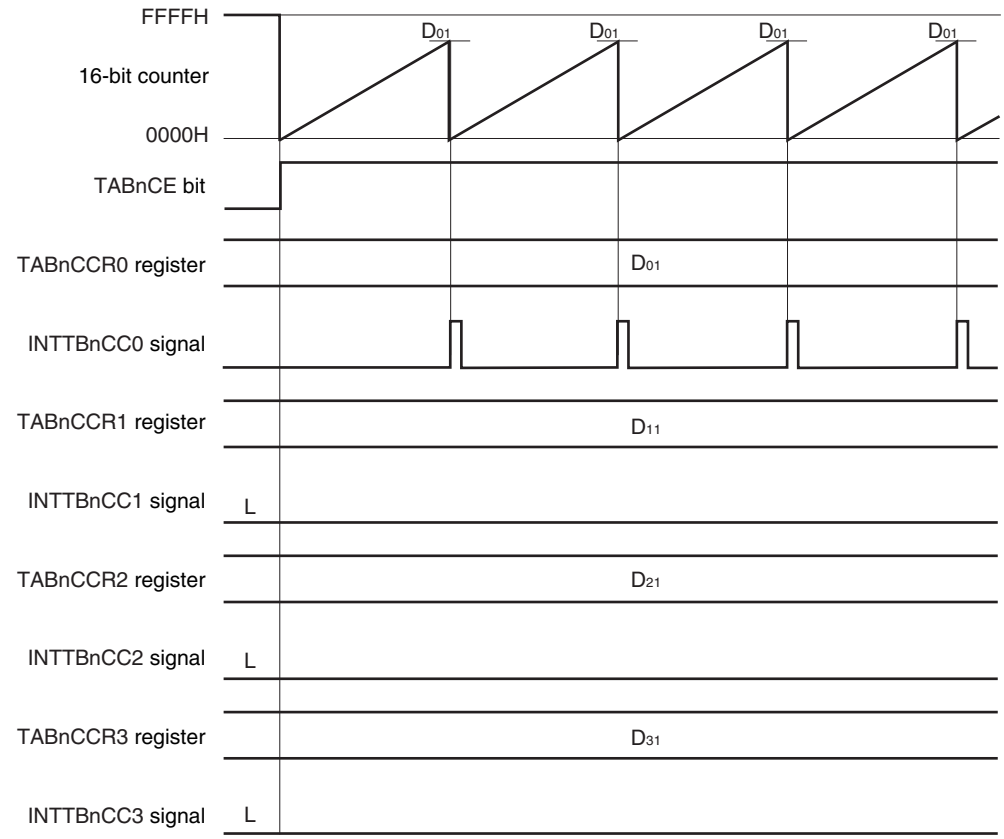


Remark n = 0, 1
b = 1 to 3

If the set value of the TABnCCRb register is greater than the set value of the TABnCCR0 register, the INTTBnCCb signal is not generated because the count value of the 16-bit counter and the value of the TABnCCRb register do not match.

When the TABnCCRb register is not used, it is recommended to set its value to FFFFH.

Figure 7-19. Timing Chart When $D_{01} < D_{b1}$



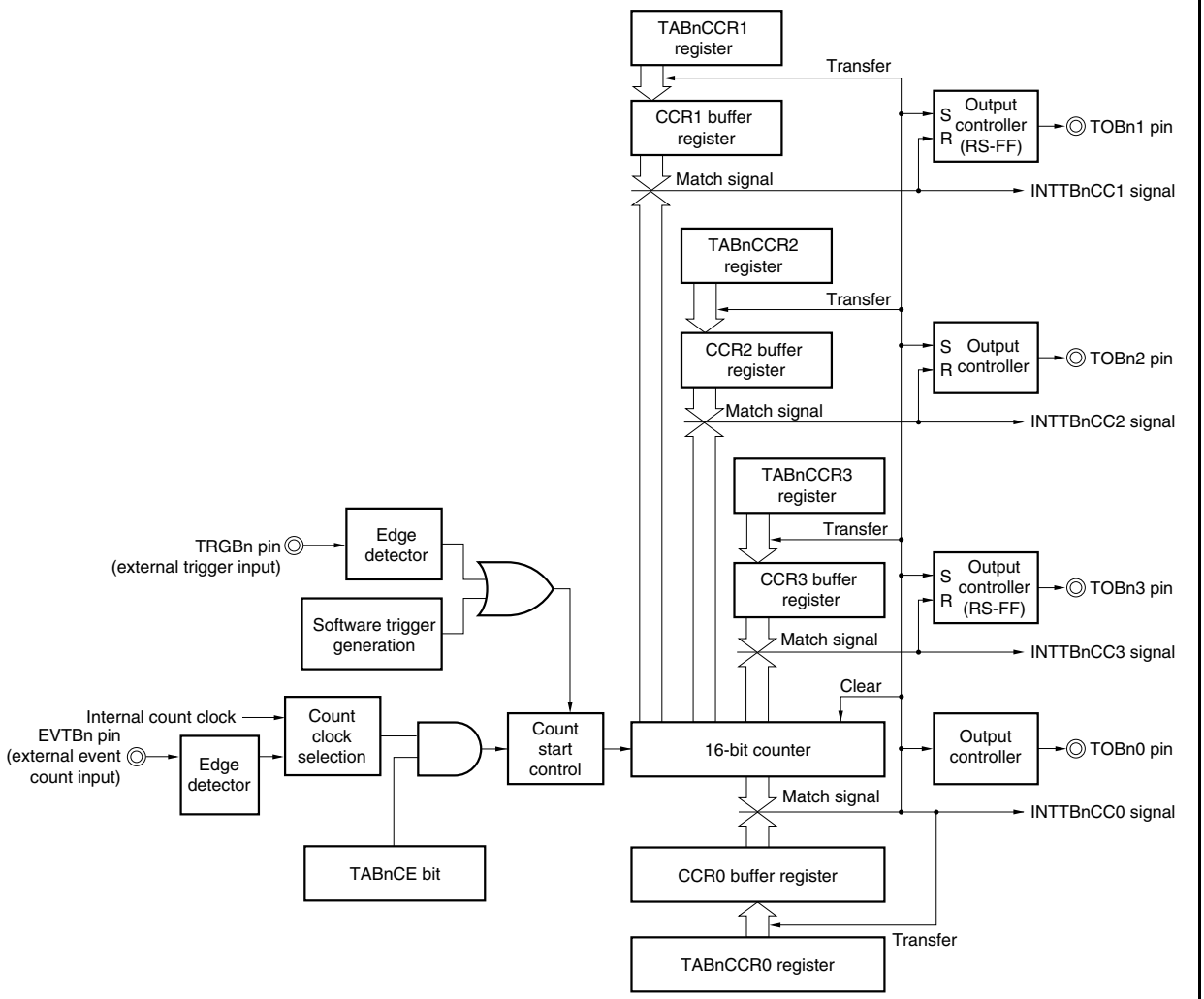
Remark n = 0, 1
b = 1 to 3

7.6.3 External trigger pulse output mode (TABnMD2 to TABnMD0 bits = 010)

In the external trigger pulse output mode, 16-bit timer/event counter AB waits for a trigger when the TABnCTL0.TABnCE bit is set to 1. When the valid edge of an external trigger input signal (TRGBn) is detected, 16-bit timer/event counter AB starts counting, and outputs a PWM waveform (up to 3-phase) from the TOBn1 to TOBn3 pins. A PWM waveform with a duty factor of 50% whose half cycle is the set value of the TABnCCR0 register + 1 can also be output from the TOBn0 pin.

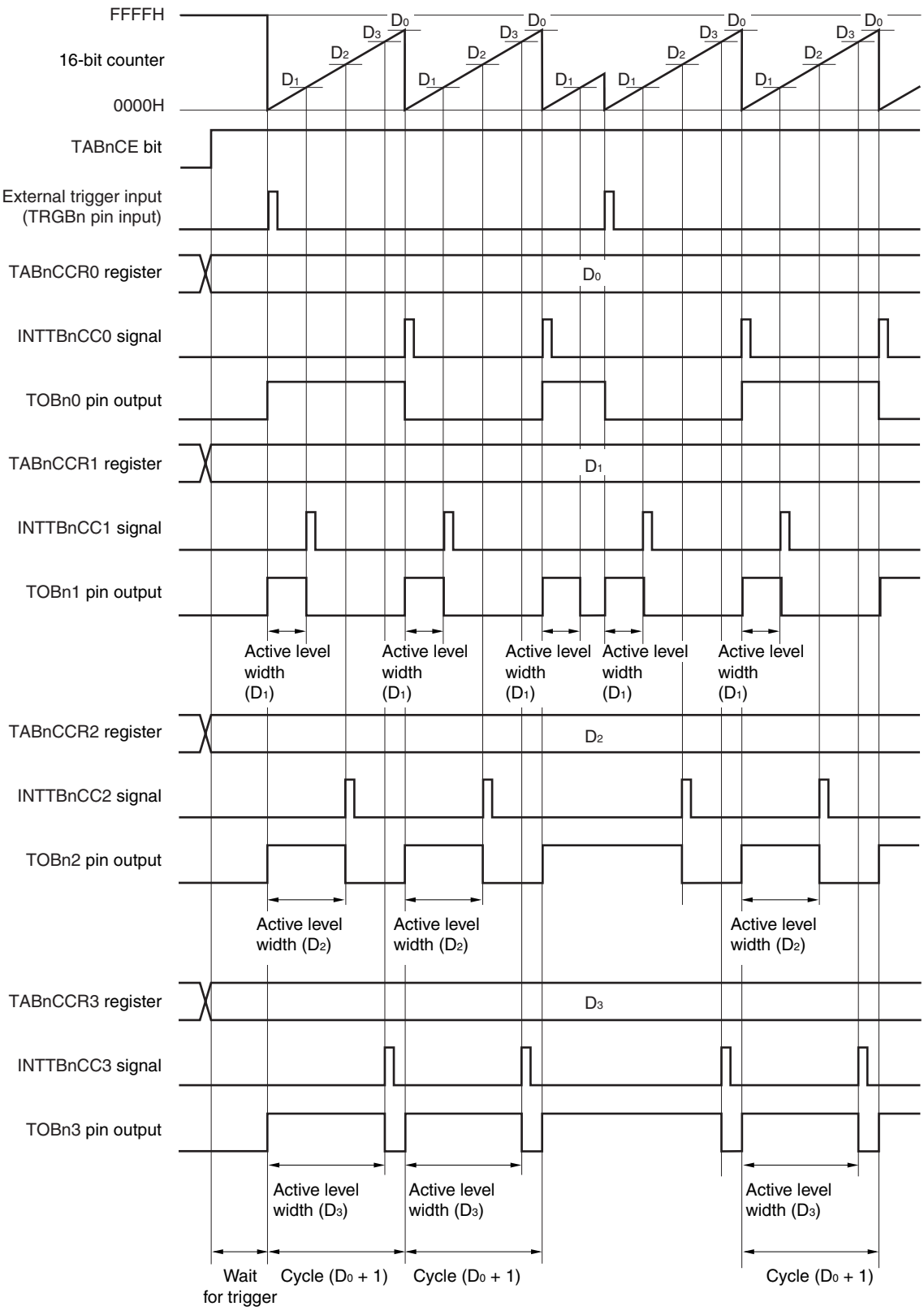
Pulses can also be output by generating a software trigger instead of using the external trigger input.

Figure 7-20. Configuration in External Trigger Pulse Output Mode



Remark n = 0, 1

Figure 7-21. Basic Timing in External Trigger Pulse Output Mode



Remark n = 0, 1

16-bit timer/event counter AB waits for a trigger when the TABnCE bit is set to 1. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting at the same time, and outputs a PWM waveform from the TOBnb pin. If the trigger is generated again while the counter is operating, the counter is cleared to 0000H and restarted. (The output of the TOBn0 pin is inverted. The TOBnb pin outputs a high-level regardless of the status (high/low) when a trigger occurs.)

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TABnCCRb register) × Count clock cycle
Cycle = (Set value of TABnCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TABnCCRb register)/(Set value of TABnCCR0 register + 1)

The compare match request signal INTTBnCC0 is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal INTTBnCCb is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.

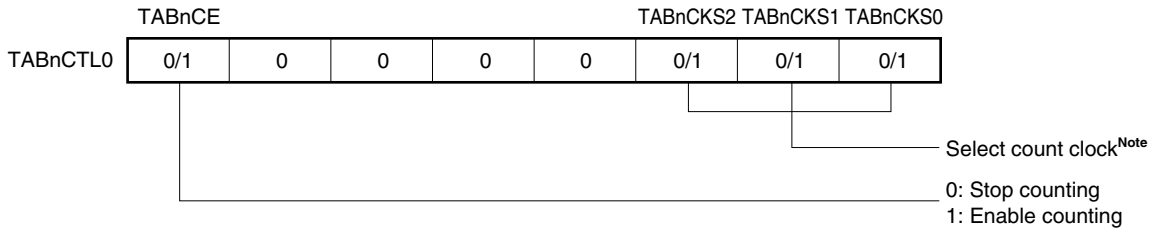
The value set to the TABnCCRa register is transferred to the CCRa buffer register when the count value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

The valid edge of an external trigger input signal (TRGBn), or setting the software trigger (TABnCTL1.TABnEST bit) to 1 is used as the trigger.

Remark n = 0, 1
a = 0 to 3
b = 1 to 3

Figure 7-22. Setting of Registers in External Trigger Pulse Output Mode (1/3)

(a) TABn control register 0 (TABnCTL0)



Note The setting is invalid when the TABnCTL1.TABnEEE bit = 1.

Figure 7-22. Setting of Registers in External Trigger Pulse Output Mode (2/3)

(b) TABn control register 1 (TABnCTL1)

	TABnEST	TABnEEE				TABnMD2	TABnMD1	TABnMD0
TABnCTL1	0	0/1	0/1	0	0	0	1	0

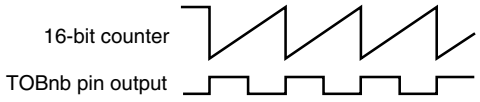
- 0, 1, 0: External trigger pulse output mode
- 0: Operate on count clock selected by TABnCKS0 to TABnCKS2 bits
  1: Count with external event count input signal
- Generate software trigger when 1 is written

(c) TABn I/O control register 0 (TABnIOC0)

	TABnOL3	TABnOE3	TABnOL2	TABnOE2	TABnOL1	TABnOE1	TABnOL0	TABnOE0
TABnIOC0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

- 0: Disable TOBn0 pin output
  1: Enable TOBn0 pin output
- Setting of TOBn0 pin output level while waiting for external trigger
  0: Low level
  1: High level
- 0: Disable TOBn1 pin output
  1: Enable TOBn1 pin output
- Setting of TOBn1 pin output level while waiting for external trigger
  0: Low level
  1: High level
- 0: Disable TOBn2 pin output
  1: Enable TOBn2 pin output
- Setting of TOBn2 pin output level while waiting for external trigger
  0: Low level
  1: High level
- 0: Disable TOBn3 pin output
  1: Enable TOBn3 pin output
- Setting of TOBn3 pin output level while waiting for external trigger
  0: Low level
  1: High level

• When TABnOLb bit = 0



• When TABnOLb bit = 1

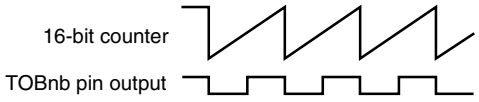
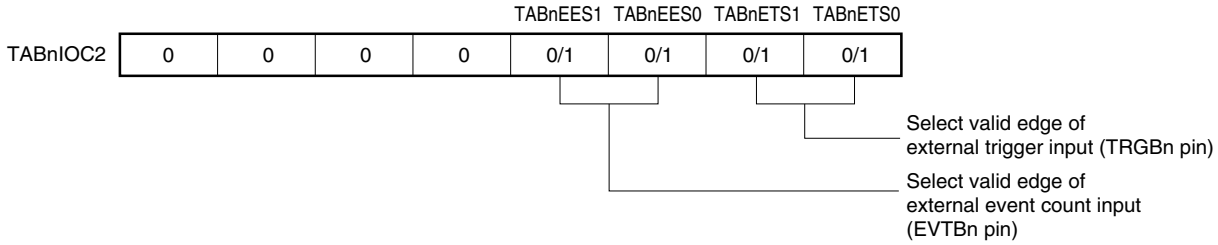


Figure 7-22. Setting of Registers in External Trigger Pulse Output Mode (3/3)

(d) TABn I/O control register 2 (TABnIOC2)



(e) TABn counter read buffer register (TABnCNT)

The value of the 16-bit counter can be read by reading the TABnCNT register.

(f) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)

If D_0 is set to the TABnCCR0 register, D_1 to the TABnCCR1 register, D_2 to the TABnCCR2 register, and D_3, to the TABnCCR3 register, the cycle and active level of the PWM waveform are as follows.

Cycle = $(D_0 + 1) \times$ Count clock cycle
TOBn1 pin PWM waveform active level width = $D_1 \times$ Count clock cycle
TOBn2 pin PWM waveform active level width = $D_2 \times$ Count clock cycle
TOBn3 pin PWM waveform active level width = $D_3 \times$ Count clock cycle

Remarks 1. TABn I/O control register 1 (TABnIOC1) and TABn option register 0 (TABnOPT0) are not used in the external trigger pulse output mode.

2. n = 0, 1
b = 1 to 3

(1) Operation flow in external trigger pulse output mode

Figure 7-23. Software Processing Flow in External Trigger Pulse Output Mode (1/2)

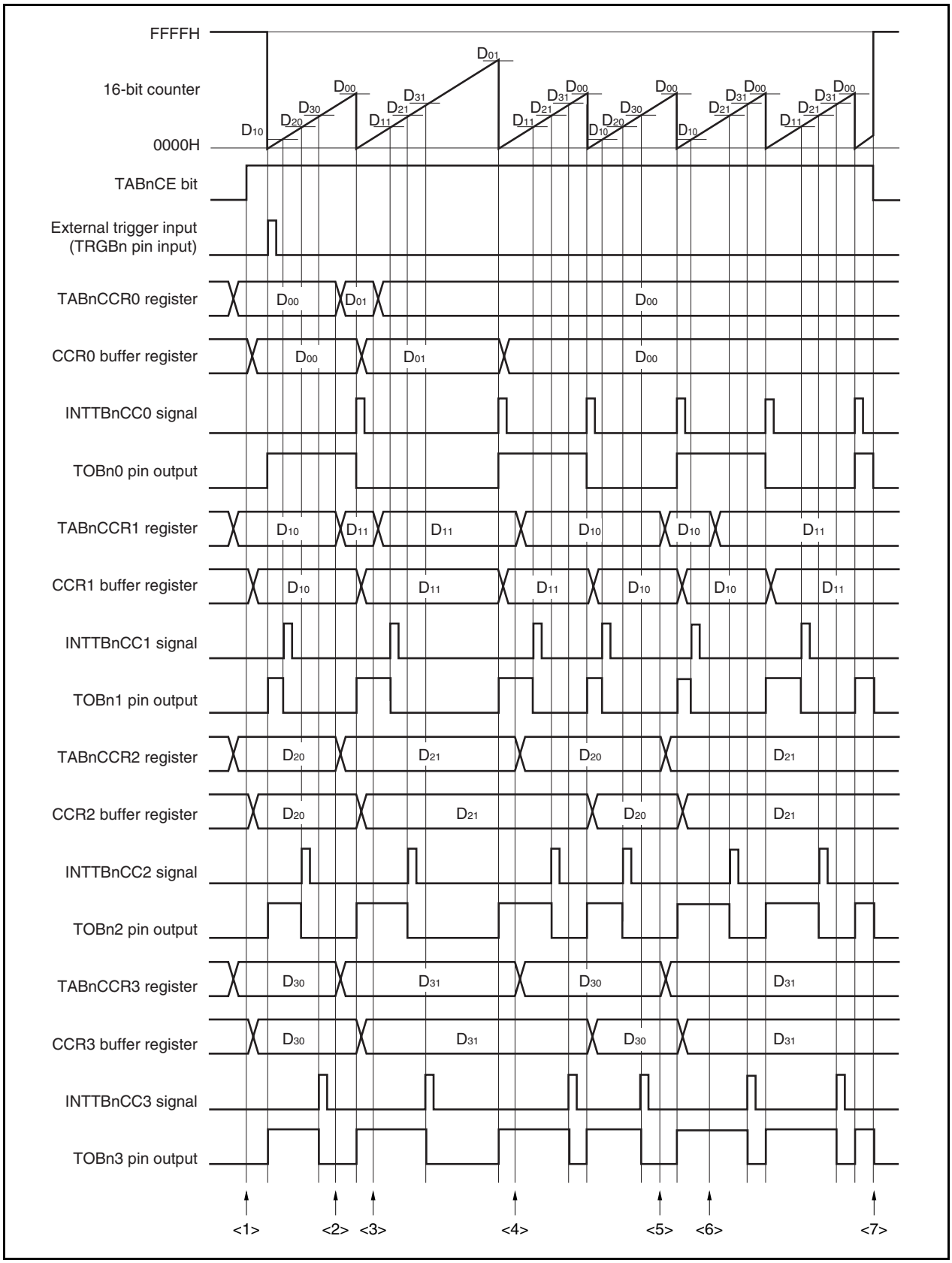
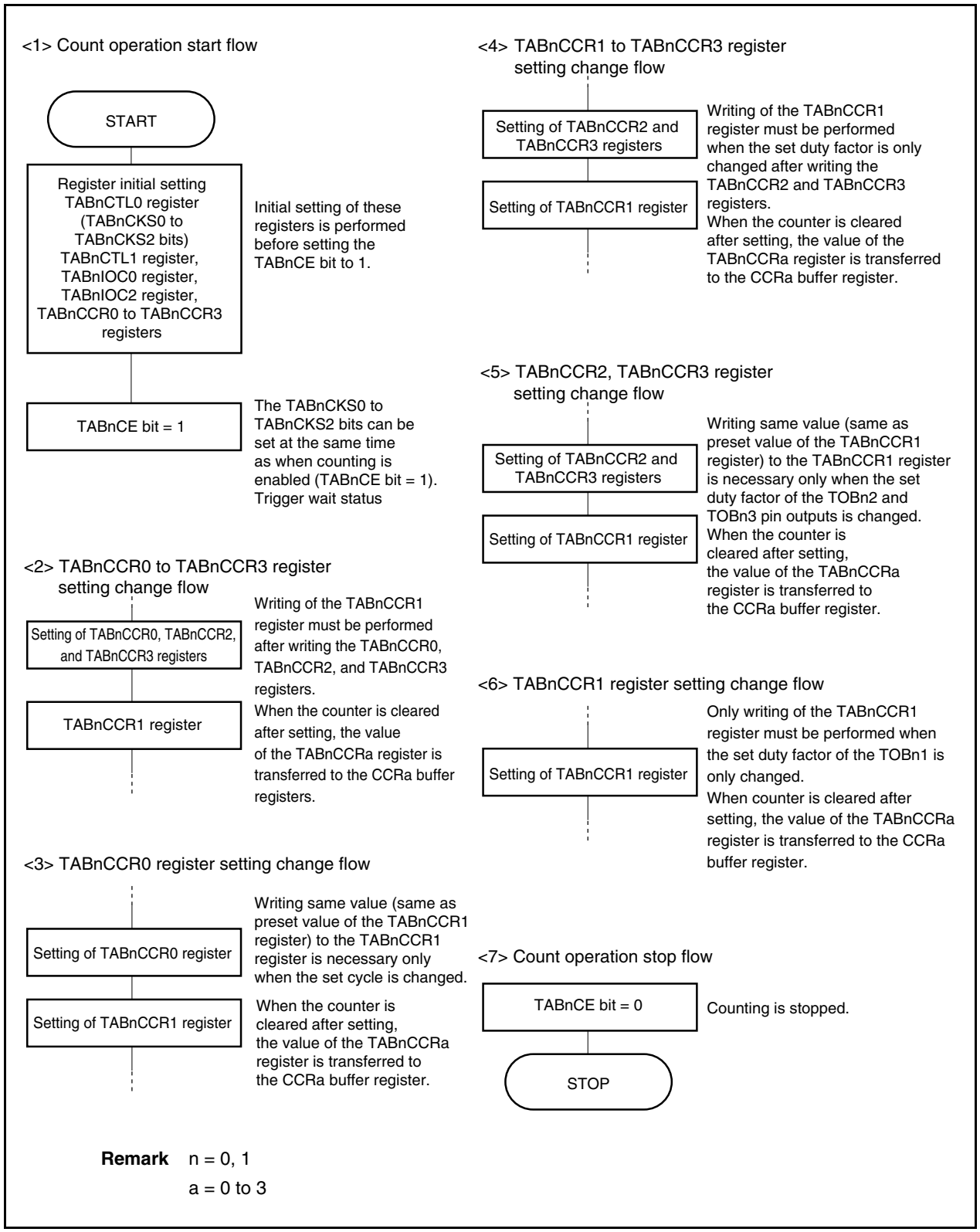


Figure 7-23. Software Processing Flow in External Trigger Pulse Output Mode (2/2)



Remark n = 0, 1

a = 0 to 3

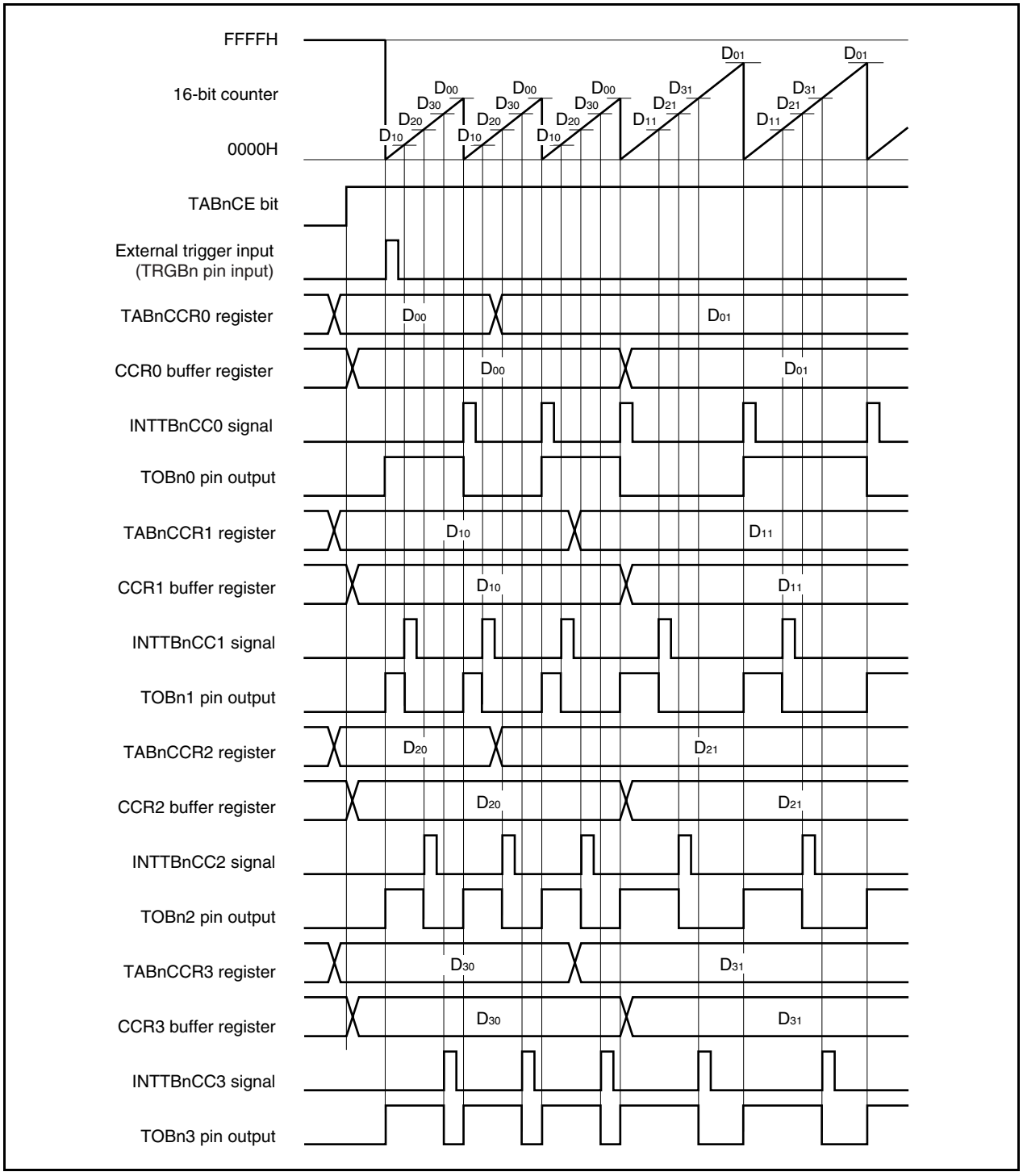
(2) External trigger pulse output mode operation timing

(a) Note on changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TABnCCR1 register last.

Rewrite the TABnCCRb register after writing the TABnCCR1 register after the INTTBnCC0 signal is detected.

Remark n = 0, 1
b = 1 to 3



In order to transfer data from the TABnCCRa register to the CCRa buffer register, the TABnCCR1 register must be written.

To change both the cycle and active level width of the PWM waveform at this time, first set the cycle to the TABnCCR0 register, set the active level width to the TABnCCR2 and TABnCCR3 registers, and then set an active level to the TABnCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TABnCCR0 register, and then write the same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, first set an active level to the TABnCCR2 and TABnCCR3 registers and then set an active level to the TABnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform output by the TOBn1 pin, only the TABnCCR1 register has to be set.

To change only the active level width (duty factor) of the PWM waveform output by the TOBn2 and TOBn3 pins, first set an active level width to the TABnCCR2 and TABnCCR3 registers, and then write the same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register.

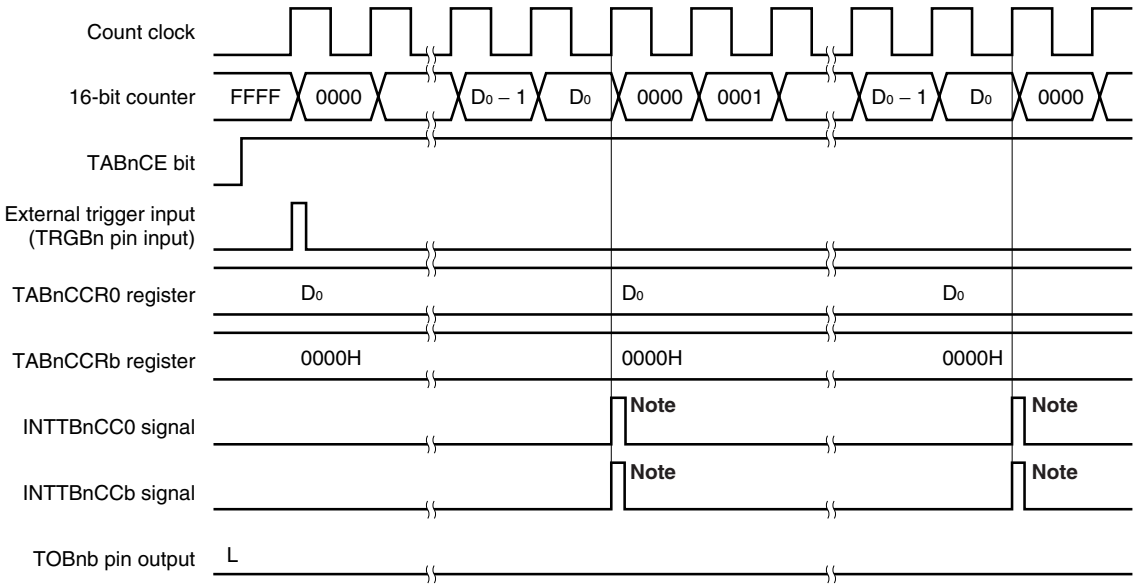
After data is written to the TABnCCR1 register, the value written to the TABnCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TABnCCR0 to TABnCCR3 registers again after writing the TABnCCR1 register once, do so after the INTTBnCC0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because timing of transferring data from the TABnCCRa register to the CCRa buffer register conflicts with writing the TABnCCRa register.

Remark n = 0, 1
a = 0 to 3

(b) 0%/100% output of PWM waveform

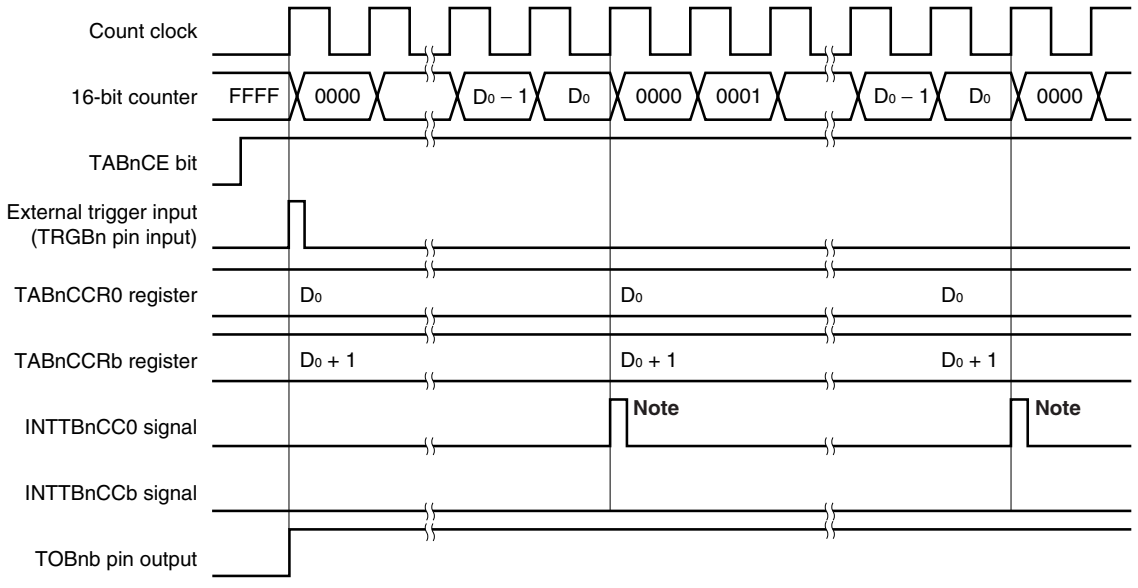
To output a 0% waveform, set the TABnCCRb register to 0000H. The 16-bit counter is cleared to 0000H and the INTTBnCC0 and INTTBnCCb signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.



Note The timing is actually delayed by one operating clock (fxx).

Remark n = 0, 1, b = 1 to 3

To output a 100% waveform, set a value of (set value of TABnCCR0 register + 1) to the TABnCCRb register. If the set value of the TABnCCR0 register is FFFFH, 100% output cannot be produced.

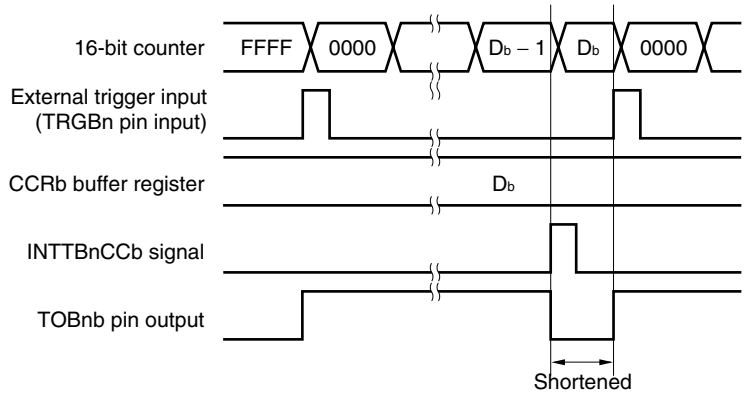


Note The timing is actually delayed by one operating clock (fxx).

Remark n = 0, 1, b = 1 to 3

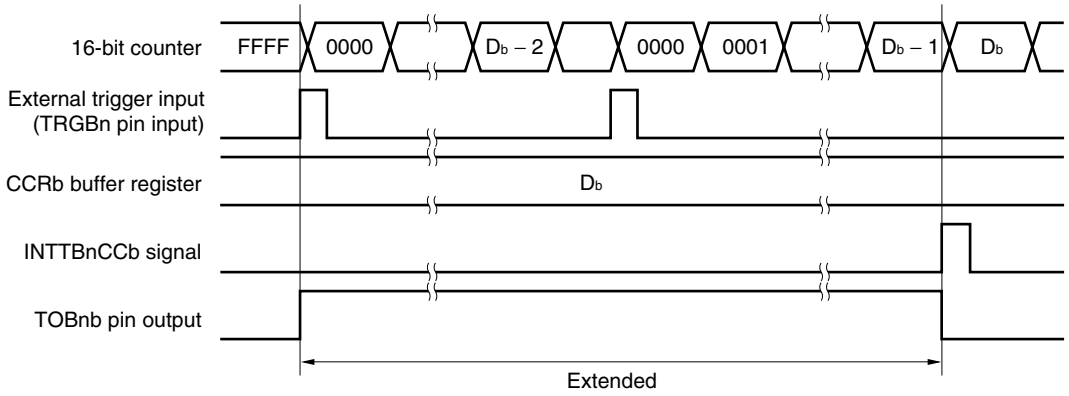
(c) Conflict between trigger detection and match with CCRb buffer register

If the trigger is detected immediately after the INTTBnCCb signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TOBnb pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark n = 0, 1
b = 1 to 3

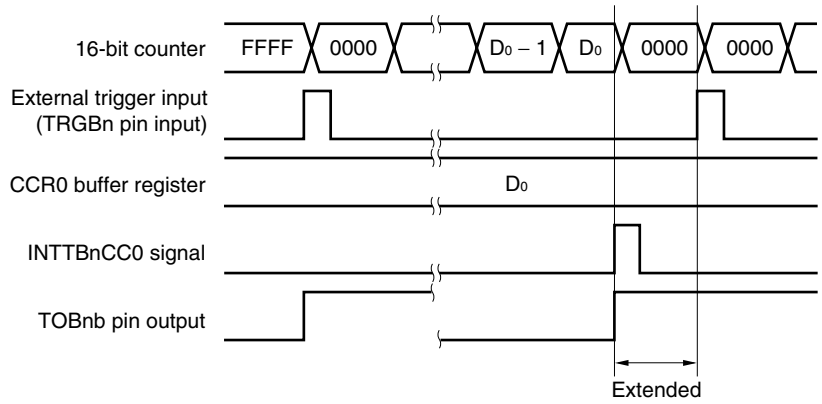
If the trigger is detected immediately before the INTTBnCCb signal is generated, the INTTBnCCb signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOBnb pin remains active. Consequently, the active period of the PWM waveform is extended.



Remark n = 0, 1
b = 1 to 3

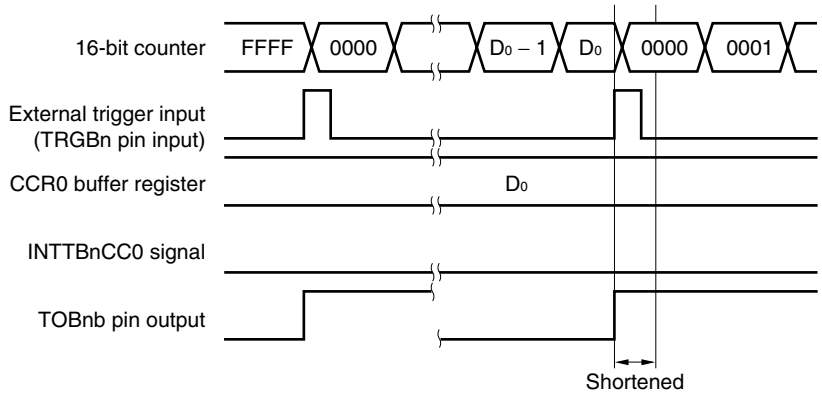
(d) Conflict between trigger detection and match with CCR0 buffer register

If the trigger is detected immediately after the INTTBnCC0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOBnb pin is extended by time from generation of the INTTBnCC0 signal to trigger detection.



Remark n = 0, 1
b = 1 to 3

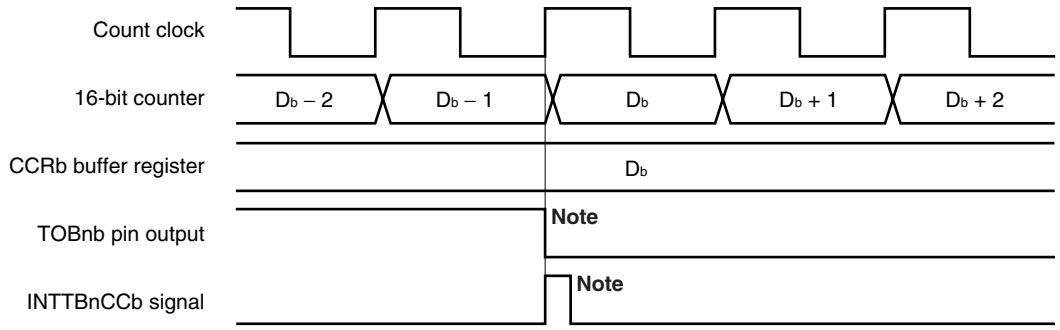
If the trigger is detected immediately before the INTTBnCC0 signal is generated, the INTTBnCC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOBnb pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark n = 0, 1
b = 1 to 3

(e) Generation timing of compare match interrupt request signal (INTTBnCCb)

The timing of generation of the INTTBnCCb signal in the external trigger pulse output mode differs from the timing of INTTBnCCb signals in other mode; the INTTBnCCb signal is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.



Note The timing is actually delayed by one operating clock (fxx).

Remark n = 0, 1
b = 1 to 3

Usually, the INTTBnCCb signal is generated in synchronization with the next count up after the count value of the 16-bit counter matches the value of the CCRb buffer register.

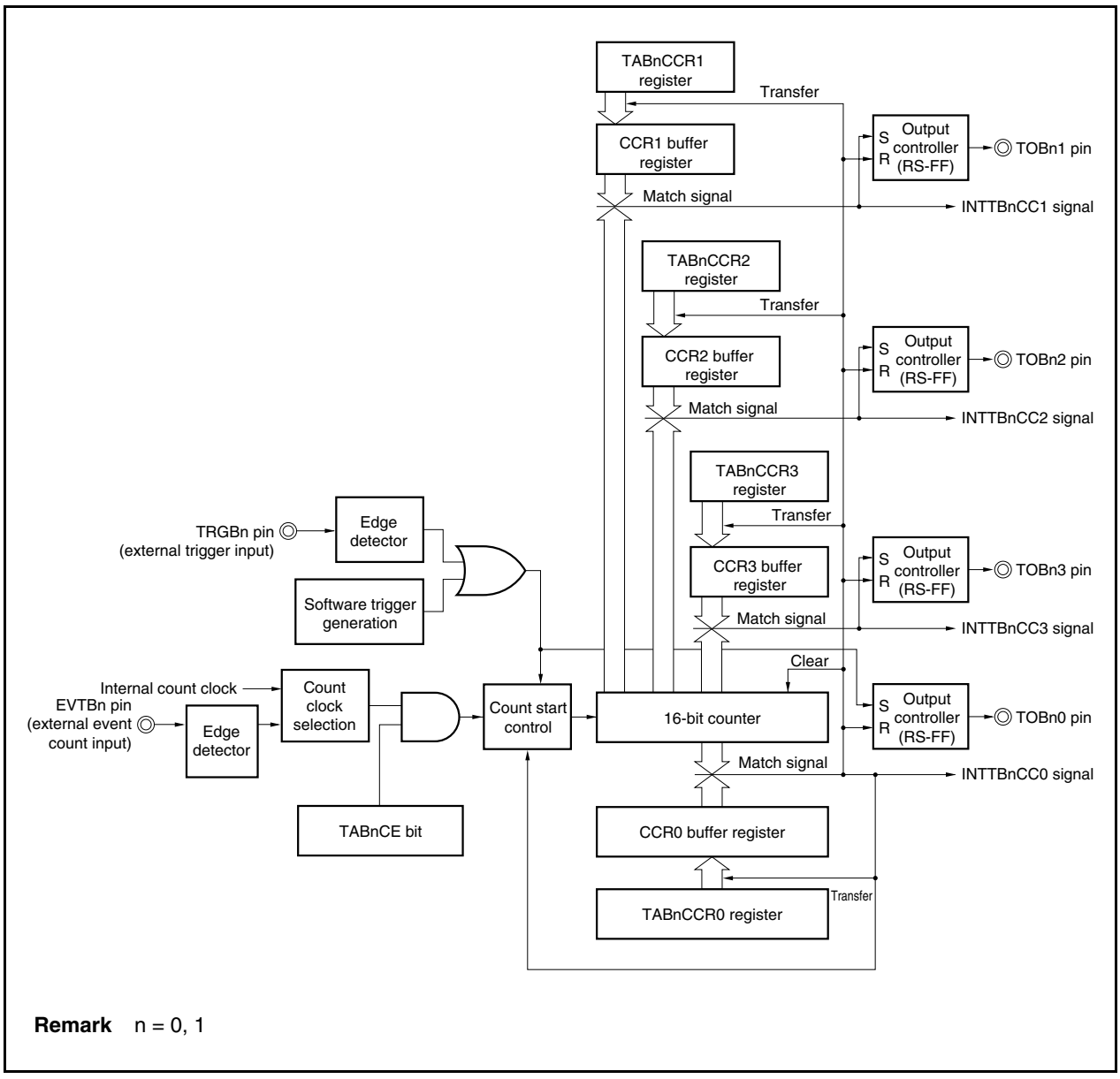
In the external trigger pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the timing of changing the output signal of the TOBnb pin.

7.6.4 One-shot pulse output mode (TABnMD2 to TABnMD0 bits = 011)

In the one-shot pulse output mode, 16-bit timer/event counter AB waits for a trigger when the TABnCTL0.TABnCE bit is set to 1. When the valid edge of an external trigger input (TRGBn) is detected, 16-bit timer/event counter AB starts counting, and outputs a one-shot pulse from the TOBn1 to TOBn3 pins. The TOBn0 pin outputs the active level while the 16-bit counter is counting, and the inactive level when the counter is stopped (waiting for a trigger).

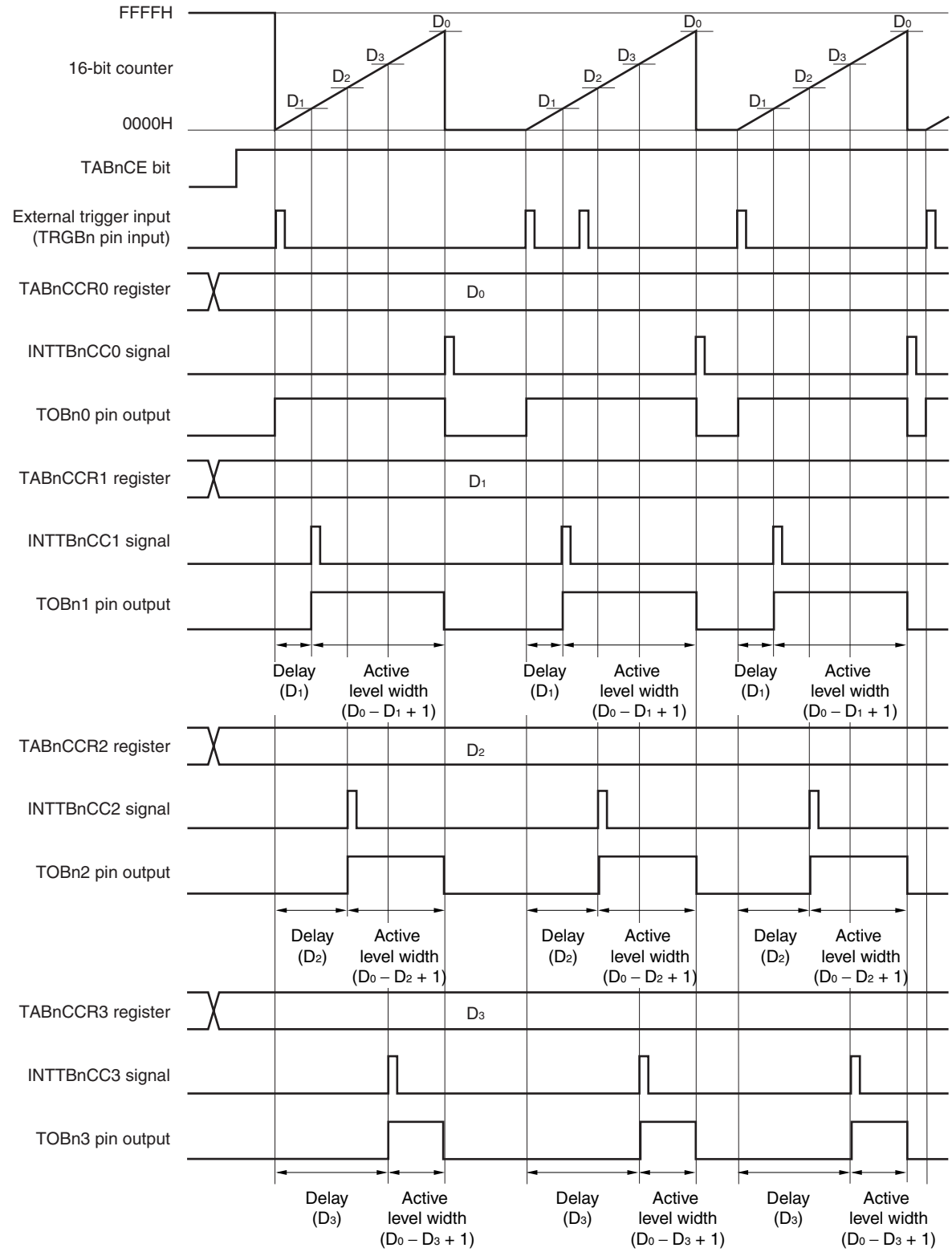
Instead of the external trigger input, a software trigger can also be generated to output the pulse.

Figure 7-24. Configuration in One-Shot Pulse Output Mode



Remark n = 0, 1

Figure 7-25. Basic Timing in One-Shot Pulse Output Mode



Remark n = 0, 1

When the TABnCE bit is set to 1, 16-bit timer/event counter AB waits for a trigger. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a one-shot pulse from the TOBnb pin. After the one-shot pulse is output, the 16-bit counter is set to 0000H, stops counting, and waits for a trigger. When the trigger is generated again, the 16-bit counter starts counting from 0000H. If a trigger is generated again while the one-shot pulse is being output, it is ignored.

The output delay period and active level width of the one-shot pulse can be calculated as follows.

Output delay period = (Set value of TABnCCRb register) × Count clock cycle

Active level width = (Set value of TABnCCR0 register − Set value of TABnCCRb register + 1) × Count clock cycle

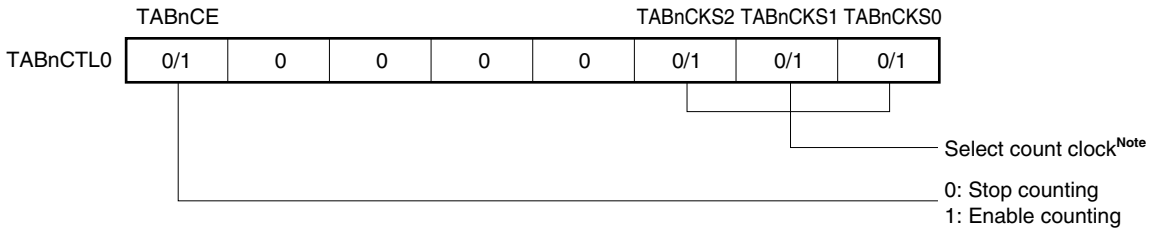
The compare match interrupt request signal INTTBnCC0 is generated when the 16-bit counter counts after its count value matches the value of the CCR0 buffer register. The compare match interrupt request signal INTTBnCCb is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.

The valid edge of an external trigger input (TRGBn) or setting the software trigger (TABnCTL1.TABnEST bit) to 1 is used as the trigger.

Remark n = 0, 1, b = 1 to 3

Figure 7-26. Setting of Registers in One-Shot Pulse Output Mode (1/3)

(a) TABn control register 0 (TABnCTL0)



Note The setting is invalid when the TABnCTL1.TABnEEE bit = 1.

(b) TABn control register 1 (TABnCTL1)

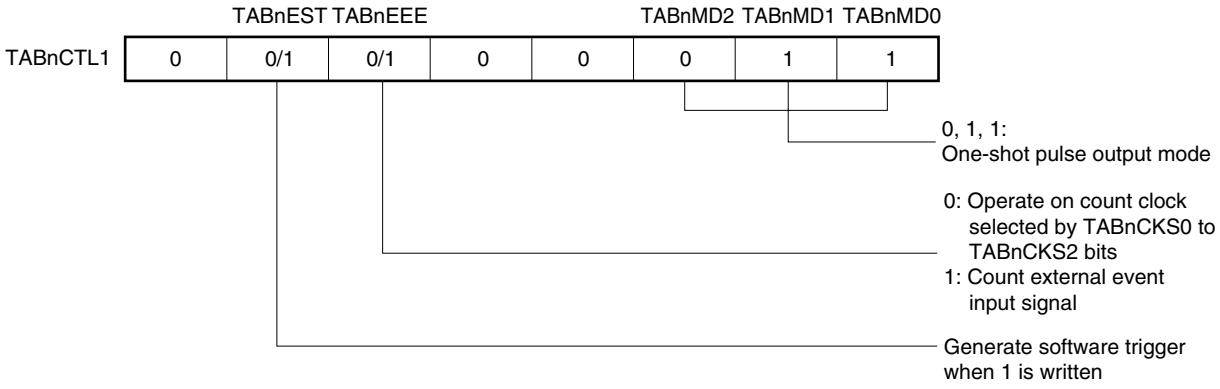
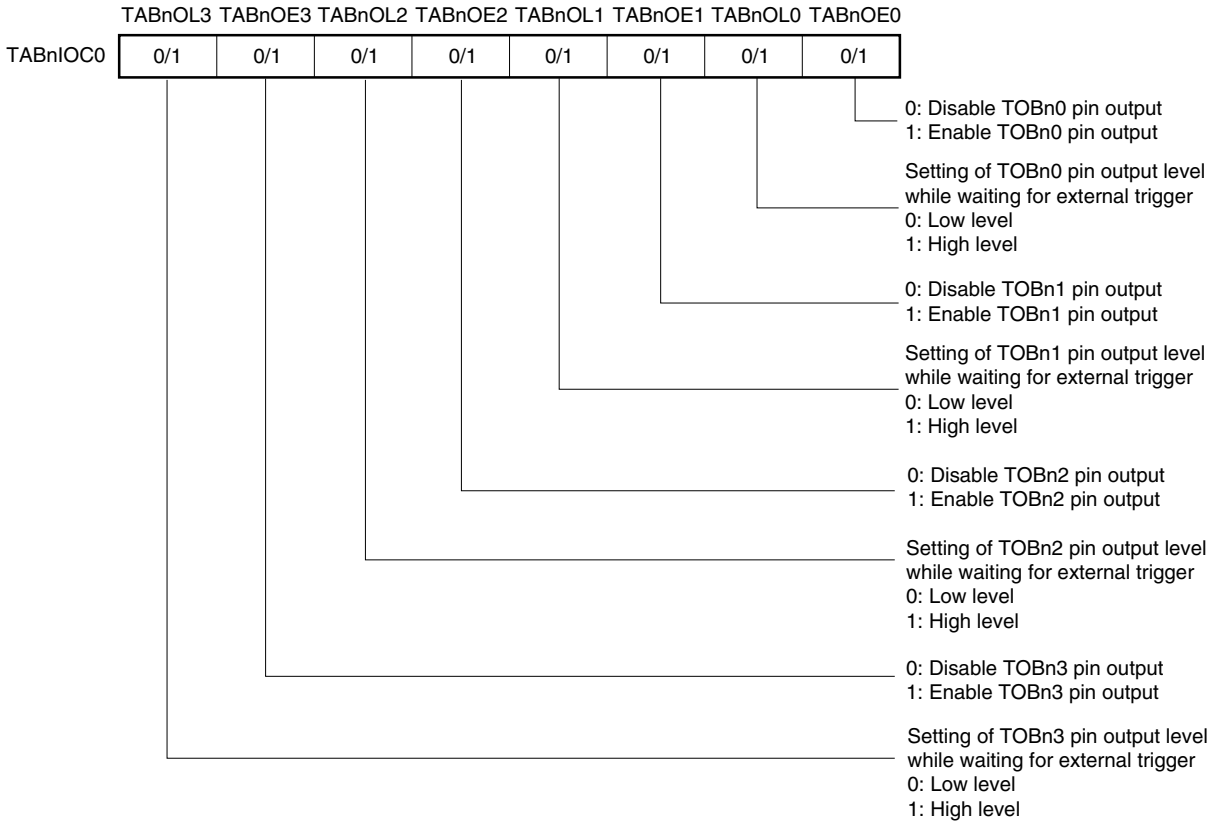
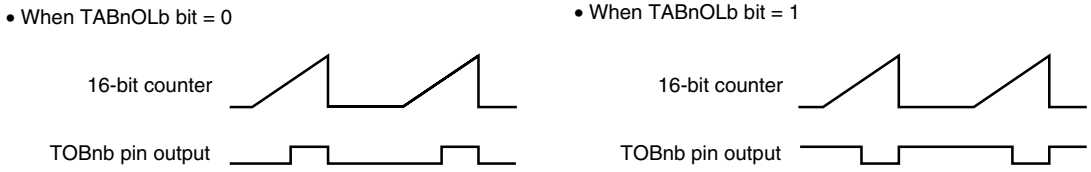


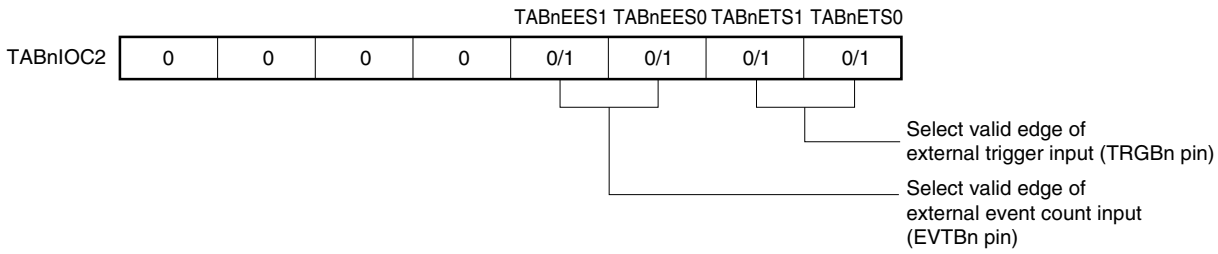
Figure 7-26. Register Setting in One-Shot Pulse Output Mode (2/3)

(c) TABn I/O control register 0 (TABnIOC0)





(d) TABn I/O control register 2 (TABnIOC2)



(e) TABn counter read buffer register (TABnCNT)

The value of the 16-bit counter can be read by reading the TABnCNT register.

Figure 7-26. Register Setting in One-Shot Pulse Output Mode (3/3)

(f) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)

If D_0 is set to the TABnCCR0 register and D_b to the TABnCCRb register, the active level width and output delay period of the one-shot pulse are as follows.

Active level width = $(D_b - D_0 + 1) \times$ Count clock cycle

Output delay period = $D_b \times$ Count clock cycle

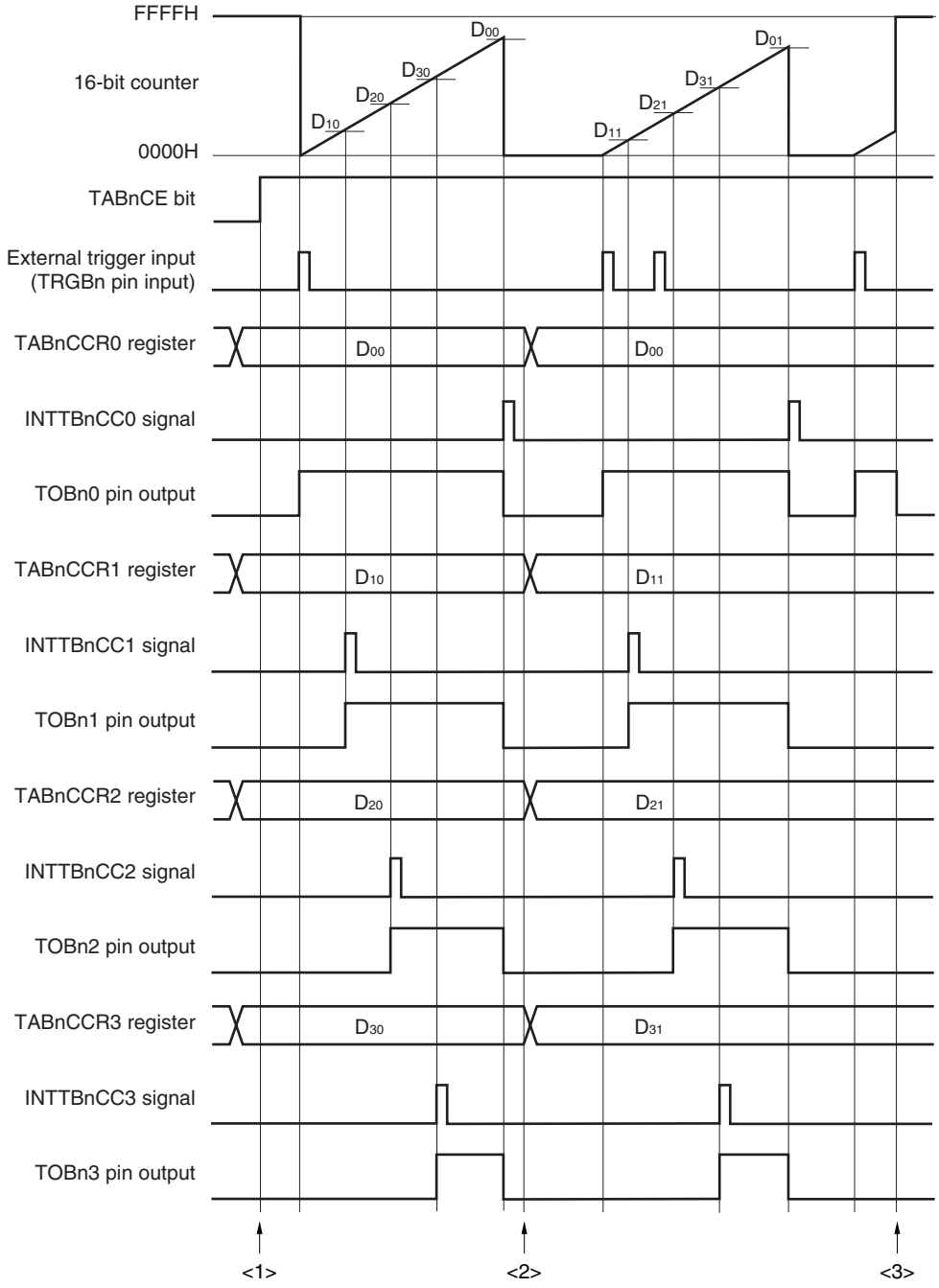
Remarks 1. TABn I/O control register 1 (TABnIOC1) and TABn option register 0 (TABnOPT0) are not used in the one-shot pulse output mode.

2. n = 0, 1

b = 1 to 3

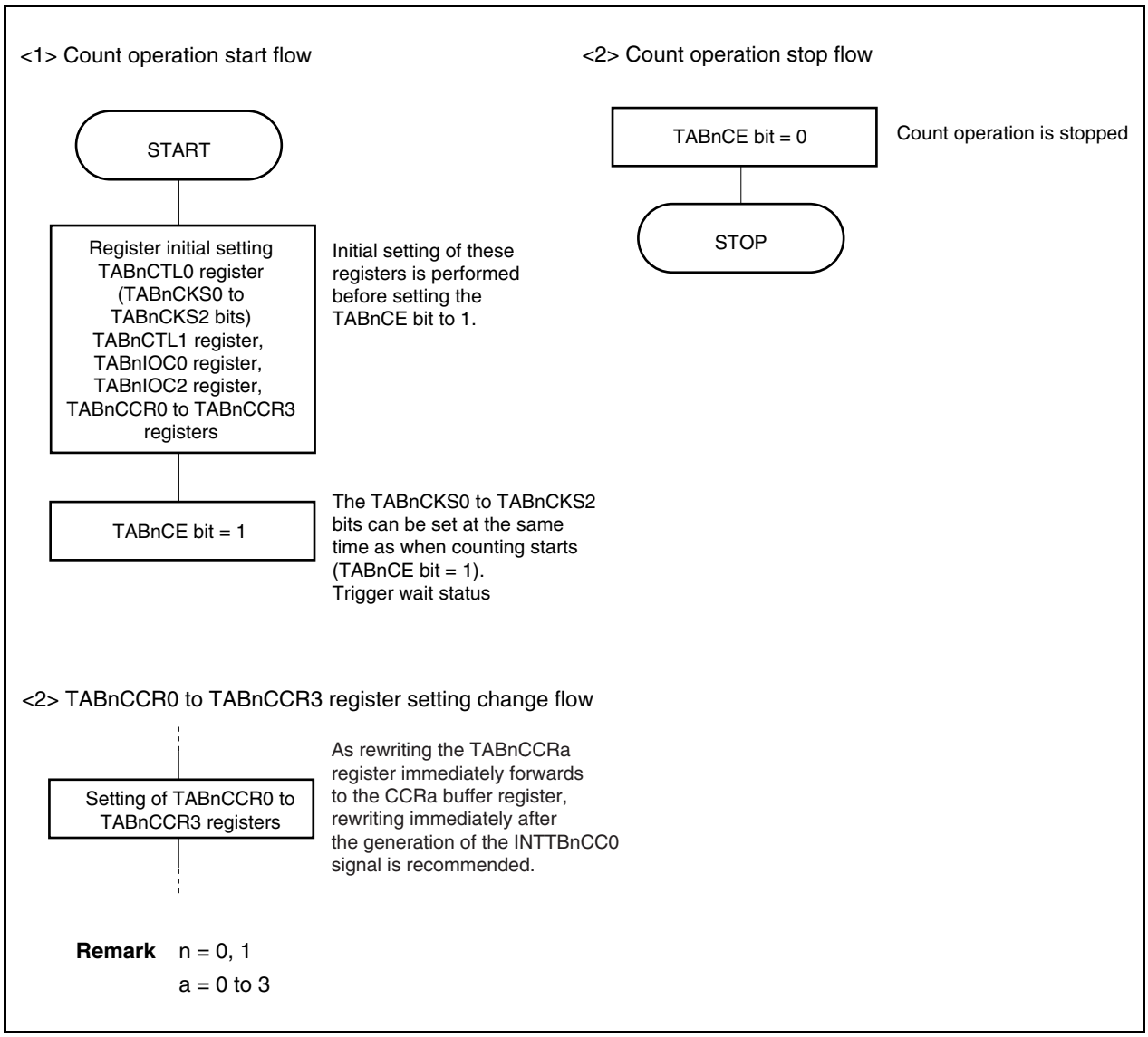
(1) Operation flow in one-shot pulse output mode

Figure 7-27. Software Processing Flow in One-Shot Pulse Output Mode (1/2)



Remark n = 0, 1

Figure 7-27. Software Processing Flow in One-Shot Pulse Output Mode (2/2)



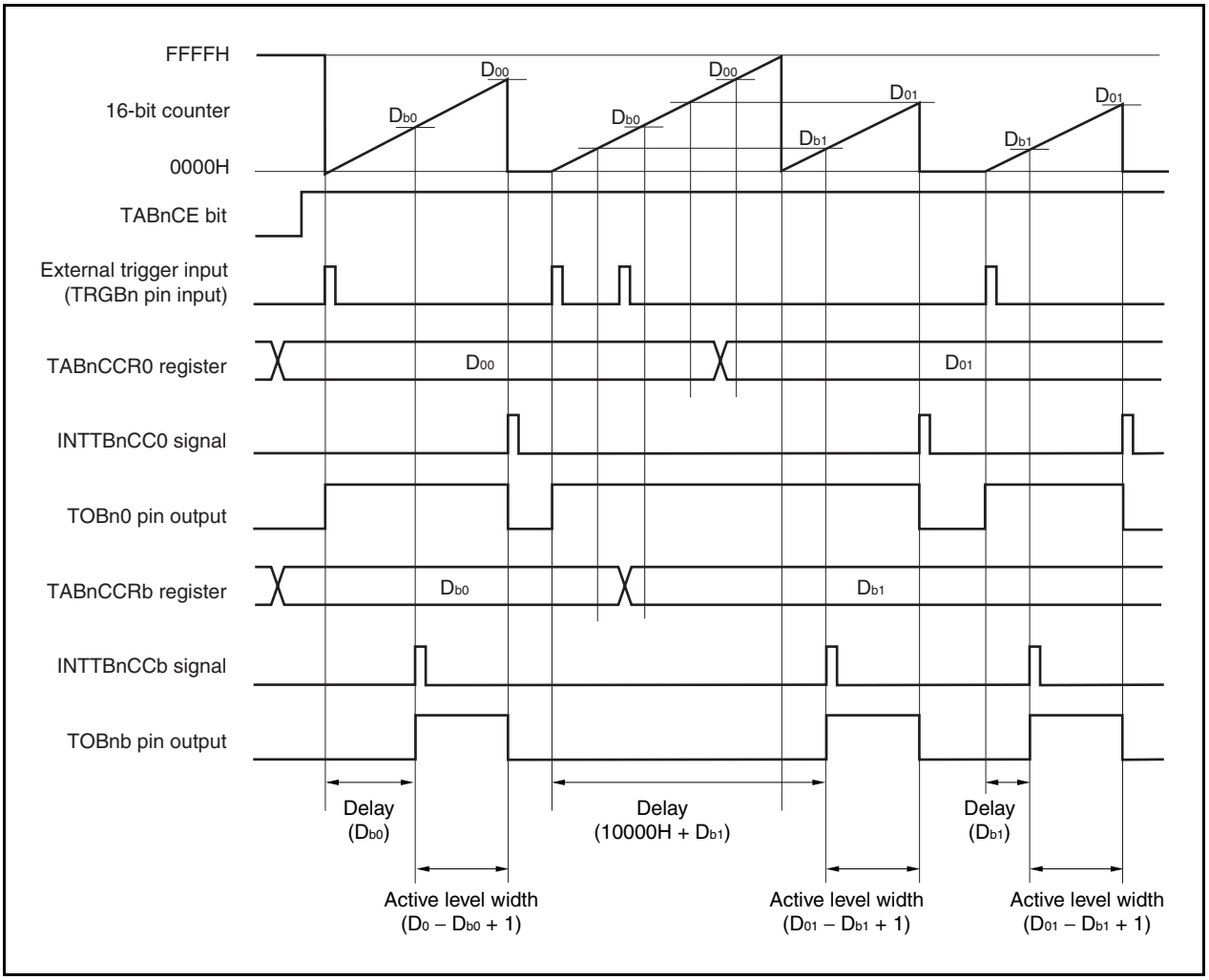
Remark n = 0, 1

a = 0 to 3

(2) Operation timing in one-shot pulse output mode

(a) Note on rewriting TABnCCRa register

To change the set value of the TABnCCRa register to a smaller value, stop counting once, and then change the set value. When the overflow may occur, stop counting once, and then change the set value.



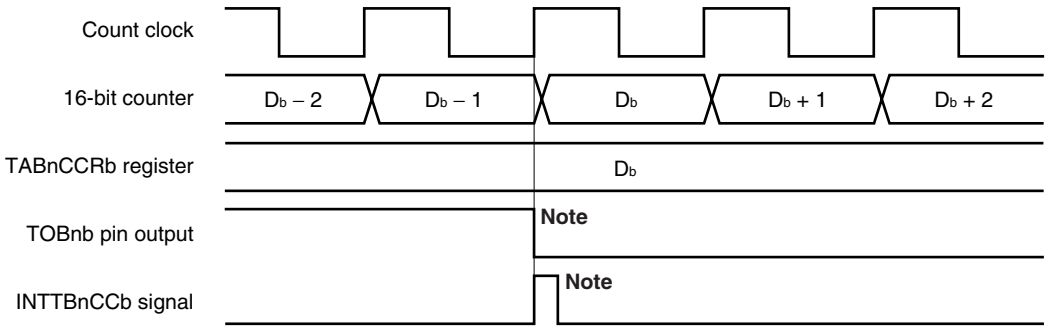
When the TABnCCR0 register is rewritten from D_{00} to D_{01} and the TABnCCRb register from D_{b0} to D_{b1} where $D_{00} > D_{01}$ and $D_{b0} > D_{b1}$, if the TABnCCRb register is rewritten when the count value of the 16-bit counter is greater than D_{b1} and less than D_{b0} and if the TABnCCR0 register is rewritten when the count value is greater than D_{01} and less than D_{00}, each set value is reflected as soon as the register has been rewritten and compared with the count value. The counter counts up to FFFFH and then counts up again from 0000H. When the count value matches D_{b1}, the counter generates the INTTBnCCb signal and asserts the TOBnb pin. When the count value matches D_{01}, the counter generates the INTTBnCC0 signal, deasserts the TOBnb pin, and stops counting.

Therefore, the counter may output a pulse with a delay period or active period different from that of the one-shot pulse that is originally expected.

Remark n = 0, 1, a = 0 to 3, b = 1 to 3

(b) Generation timing of compare match interrupt request signal (INTTBnCCb)

The generation timing of the INTTBnCCb signal in the one-shot pulse output mode is different from INTTBnCCb signals in other mode; the INTTBnCCb signal is generated when the count value of the 16-bit counter matches the value of the TABnCCRb register.



Note The timing is actually delayed by one operating clock (fxx).

Remark n = 0, 1
b = 1 to 3

Usually, the INTTBnCCb signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TABnCCRb register.

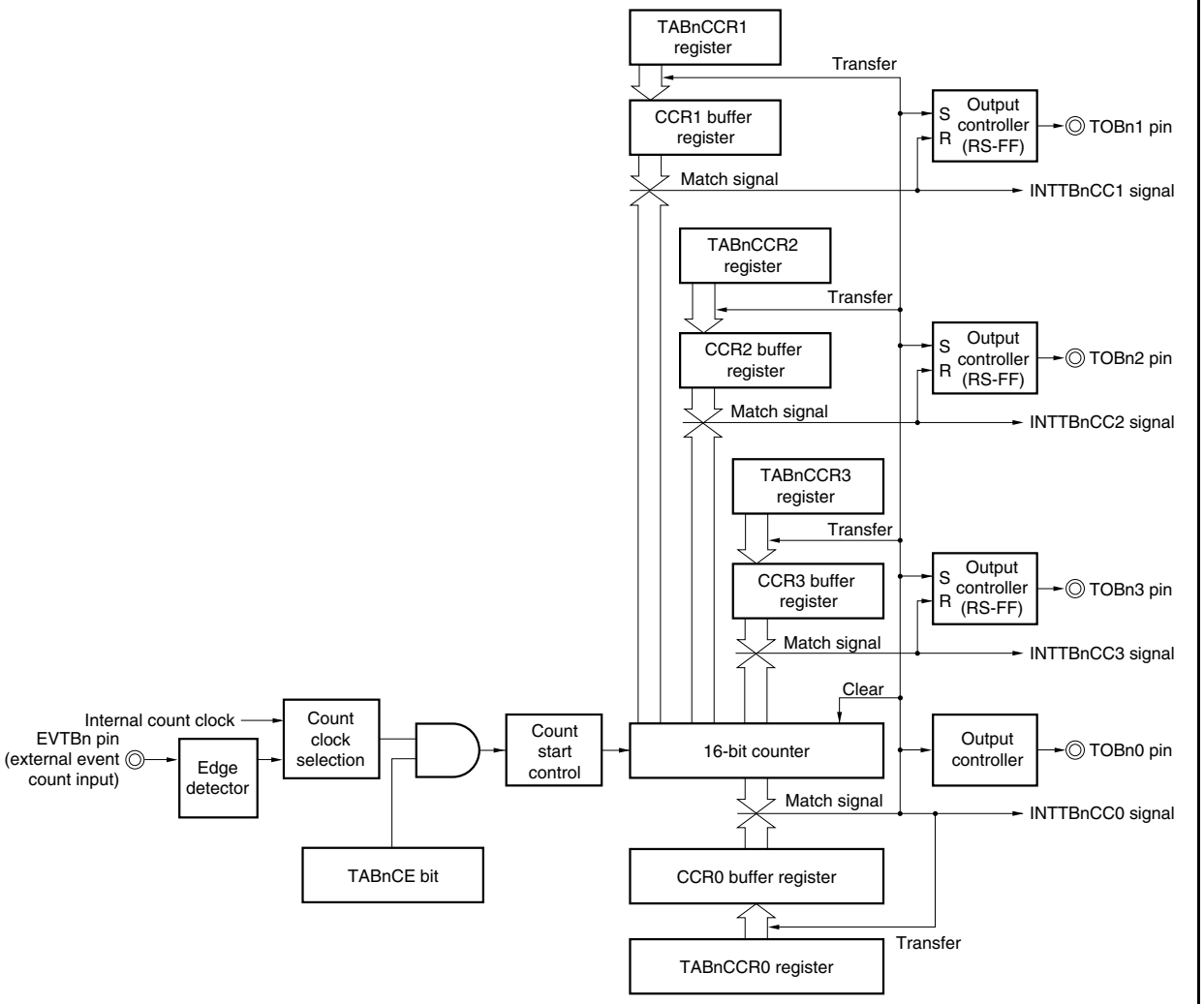
In the one-shot pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the TOBnb pin.

7.6.5 PWM output mode (TABnMD2 to TABnMD0 bits = 100)

In the PWM output mode, a PWM waveform is output from the TOBn1 to TOBn3 pins when the TABnCTL0.TABnCE bit is set to 1.

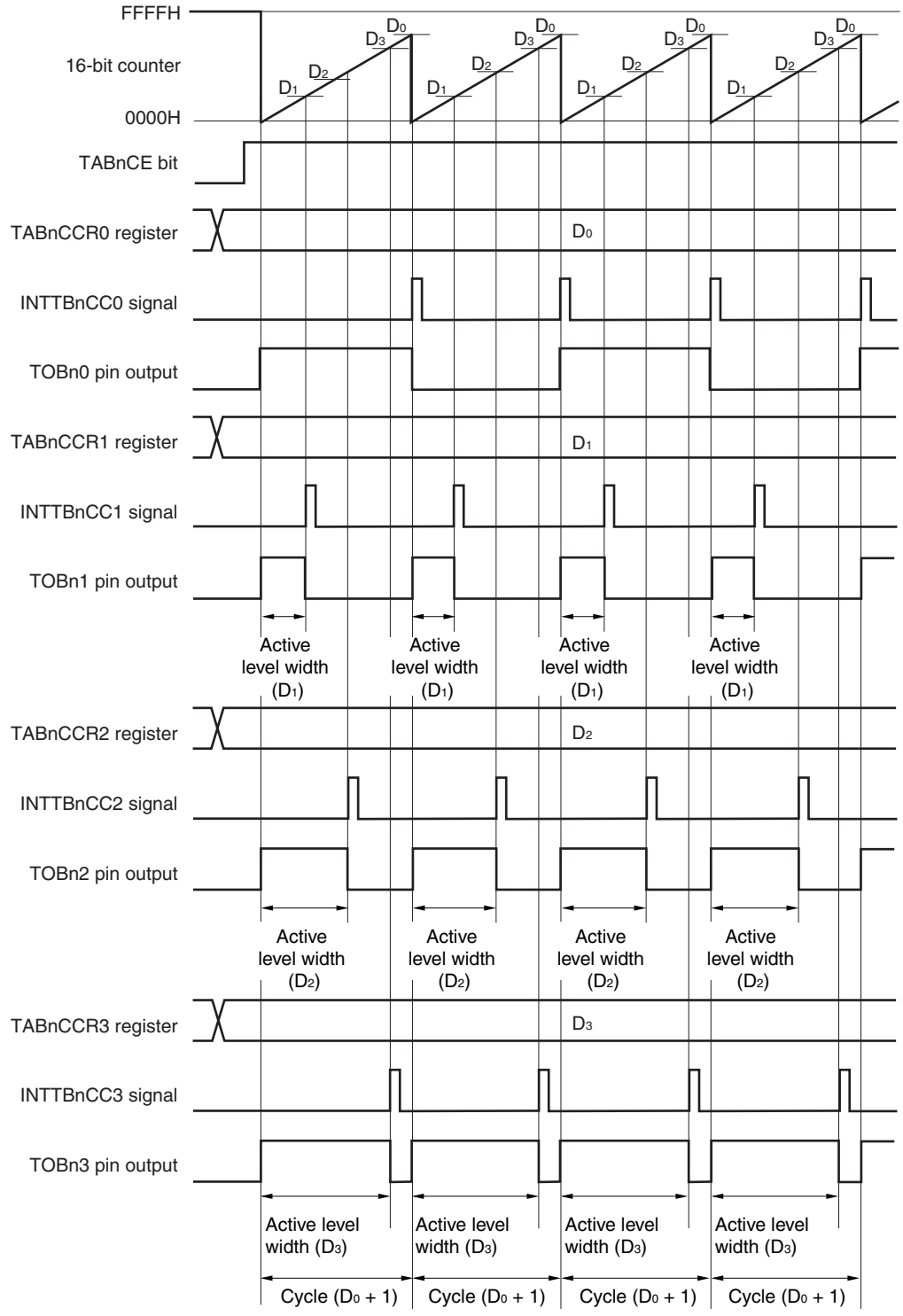
In addition, a PWM waveform with a duty factor of 50% with the set value of the TABnCCR0 register + 1 as half its cycle is output from the TOBn0 pin.

Figure 7-28. Configuration in PWM Output Mode



Remark n = 0, 1

Figure 7-29. Basic Timing in PWM Output Mode



Remark n = 0, 1

When the TABnCE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs PWM waveform from the TOBnb pin.

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TABnCCRb register) × Count clock cycle
Cycle = (Set value of TABnCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TABnCCRb register)/(Set value of TABnCCR0 register + 1)

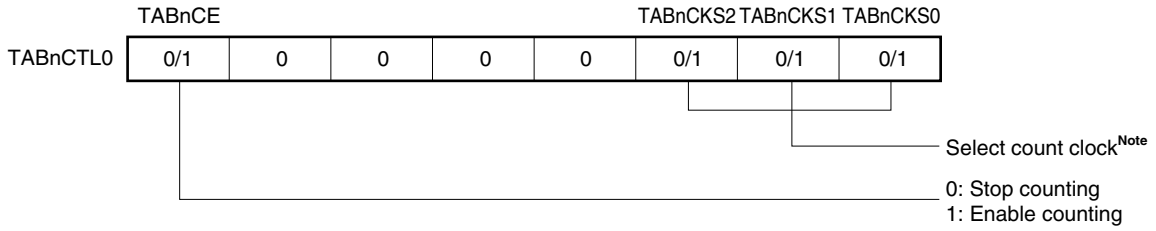
The PWM waveform can be changed by rewriting the TABnCCRa register while the counter is operating. The newly written value is reflected when the count value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

The compare match interrupt request signal INTTBnCC0 is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal INTTBnCCb is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.

Remark n = 0, 1
a = 0 to 3
b = 1 to 3

Figure 7-30. Setting of Registers in PWM Output Mode (1/3)

(a) TABn control register 0 (TABnCTL0)



Note The setting is invalid when the TABnCTL1.TABnEEE bit = 1.

(b) TABn control register 1 (TABnCTL1)

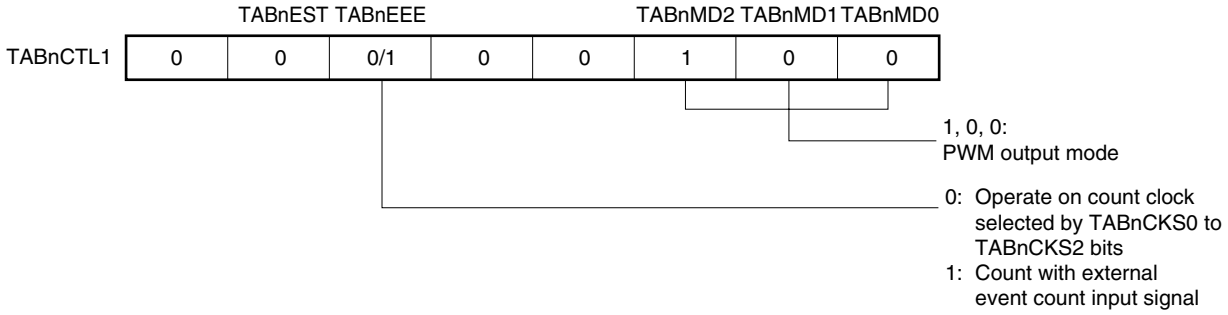
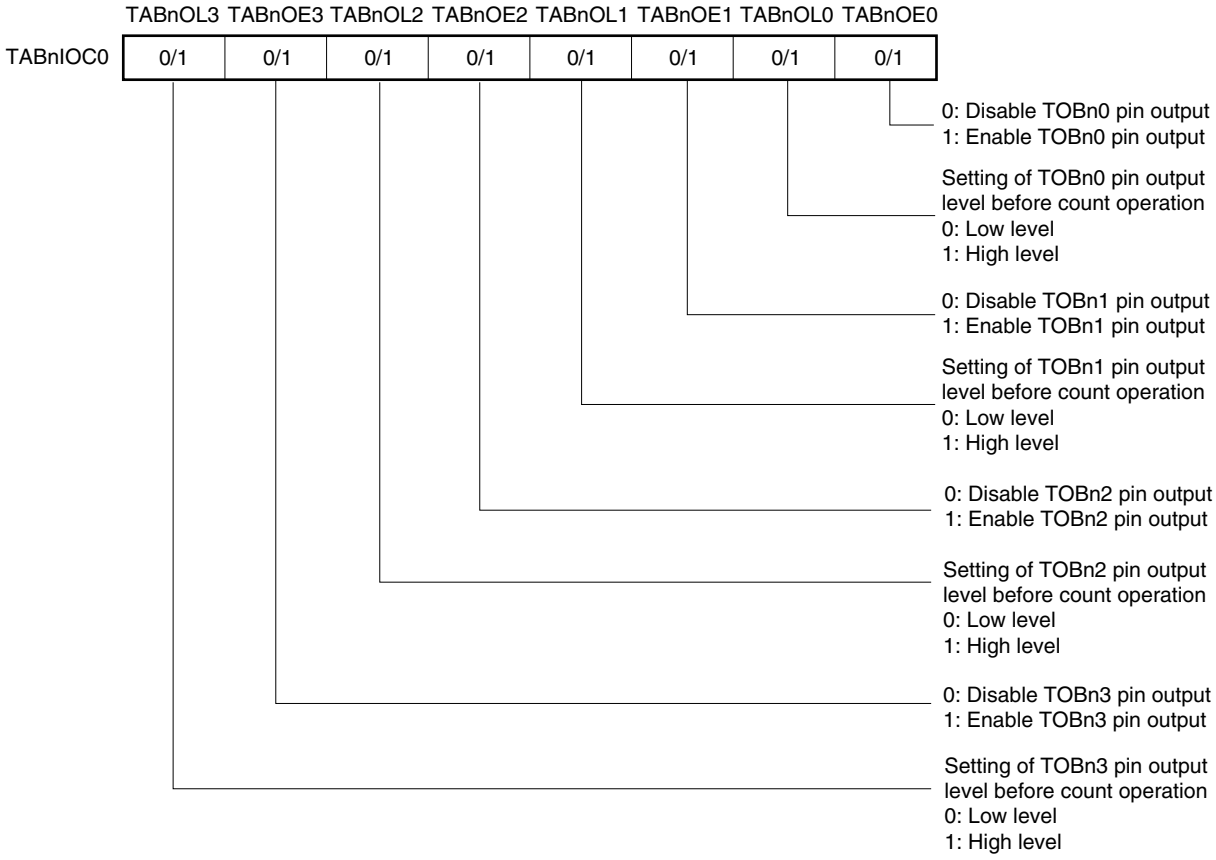
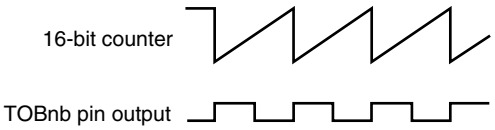


Figure 7-30. Setting of Registers in PWM Output Mode (2/3)

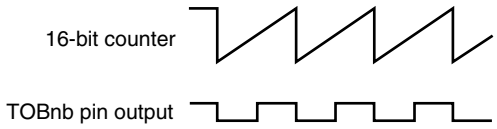
(c) TABn I/O control register 0 (TABnIOC0)



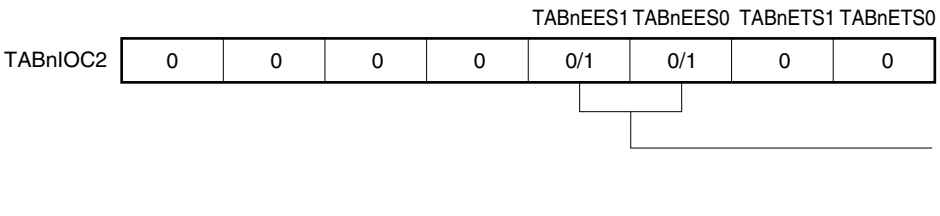
- When TABnOLb bit = 0



- When TABnOLb bit = 1



(d) TABn I/O control register 2 (TABnIOC2)



(e) TABn counter read buffer register (TABnCNT)

The value of the 16-bit counter can be read by reading the TABnCNT register.

Figure 7-30. Register Setting in PWM Output Mode (3/3)

(f) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)

If D_0 is set to the TABnCCR0 register and D_b to the TABnCCRb register, the cycle and active level of the PWM waveform are as follows.

PWM waveform cycle = $(D_0 + 1) \times$ Count clock cycle

PWM waveform active level width = $D_b \times$ Count clock cycle

Remarks 1. TABn I/O control register 1 (TABnIOC1) and TABn option register 0 (TABnOPT0) are not used in the PWM output mode.

2. n = 0, 1

b = 1 to 3

(1) Operation flow in PWM output mode

Figure 7-31. Software Processing Flow in PWM Output Mode (1/2)

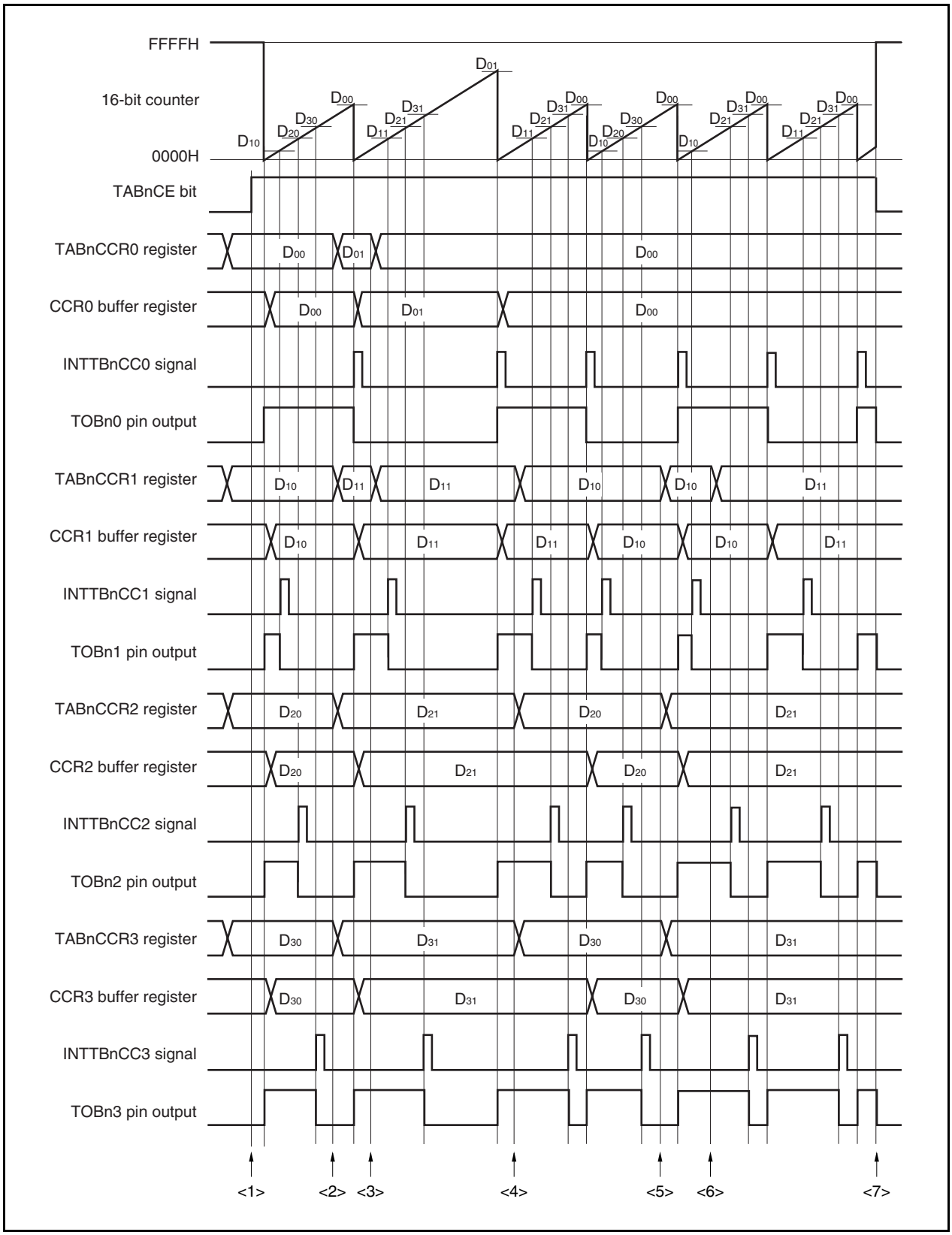
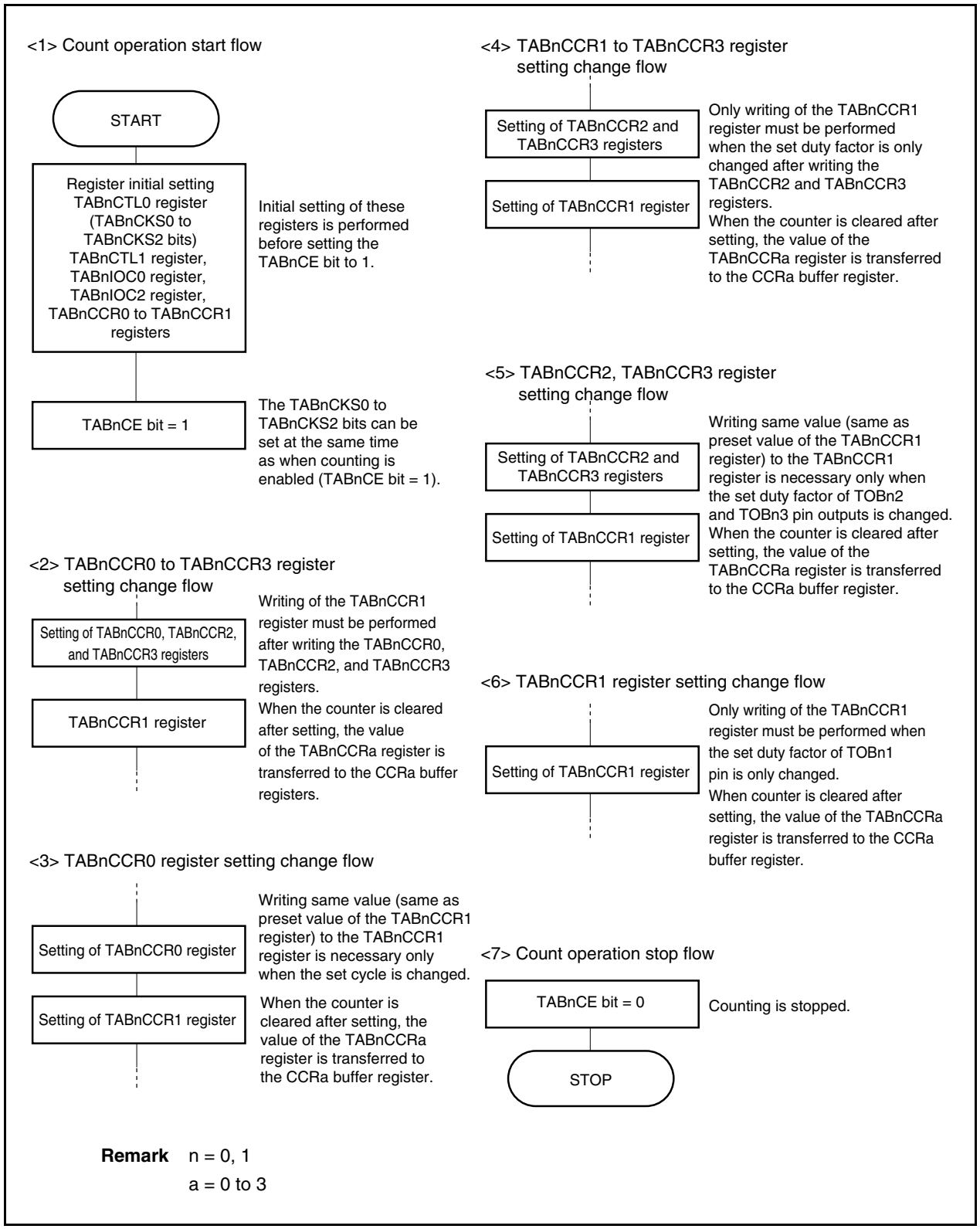


Figure 7-31. Software Processing Flow in PWM Output Mode (2/2)



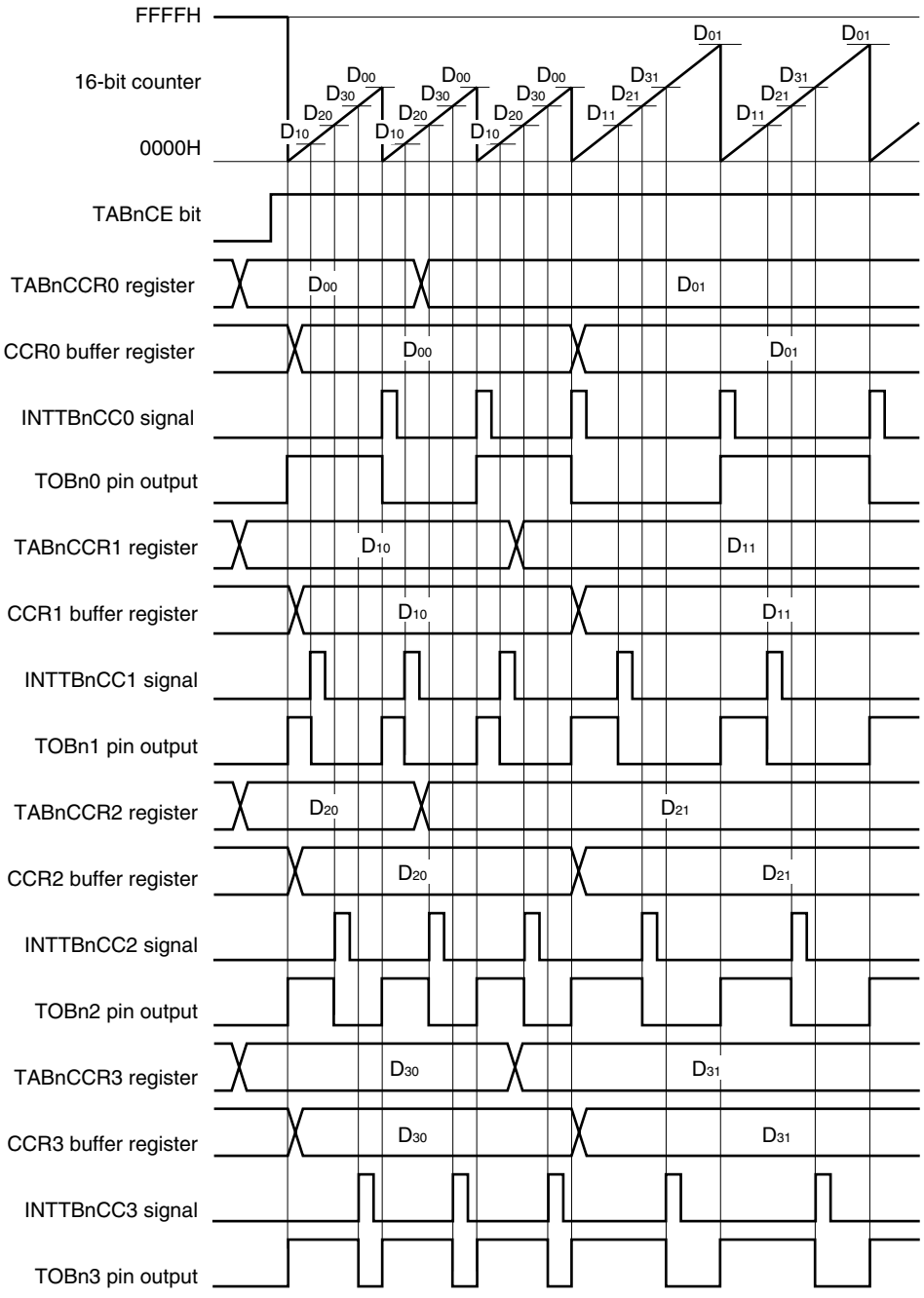
Remark n = 0, 1
a = 0 to 3

(2) PWM output mode operation timing

(a) Changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TABnCCR1 register last.

Rewrite the TABnCCRa register after writing the TABnCCR1 register after the INTTBnCC1 signal is detected.



Remark n = 0, 1

To transfer data from the TABnCCRa register to the CCRa buffer register, the TABnCCR1 register must be written.

To change both the cycle and active level of the PWM waveform at this time, first set the cycle to the TABnCCR0 register, set the active level width to the TABnCCR2 and TABnCCR3 registers, and then set an active level width to the TABnCCR1 register.

To change only the cycle of the PWM waveform, first set a cycle to the TABnCCR0 register, and then write the same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register.

To change only the active level width (duty factor) of PWM waveform, first set the active level to the TABnCCR2 and TABnCCR3 registers, and then set an active level to the TABnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform output by the TOBn1 pin, only the TABnCCR1 register has to be set.

To change only the active level width (duty factor) of the PWM waveform output by the TOBn2 and TOBn3 pins, first set an active level width to the TABnCCR2 and TABnCCR3 registers, and then write the same value (same as preset value of the TABnCCR1 register) to the TABnCCR1 register.

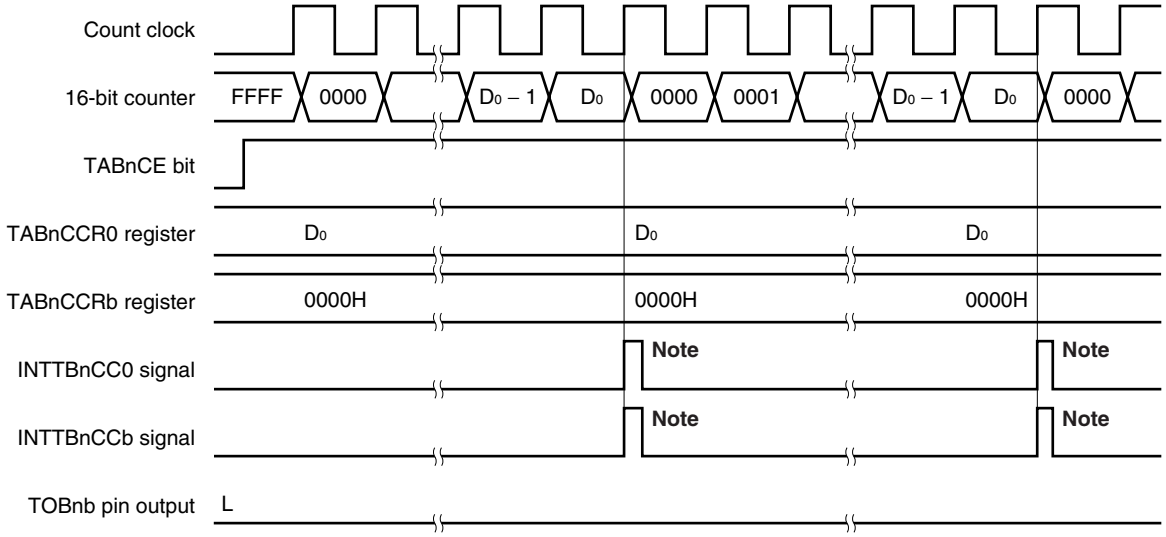
After the TABnCCR1 register is written, the value written to the TABnCCRa register is transferred to the CCRa buffer register in synchronization with the timing of clearing the 16-bit counter, and is used as a value to be compared with the value of the 16-bit counter.

To write the TABnCCR0 to TABnCCR3 registers again after writing the TABnCCR1 register once, do so after the INTTBnCC0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TABnCCRa register to the CCRa buffer register conflicts with writing the TABnCCRa register.

Remark n = 0, 1
a = 0 to 3

(b) 0%/100% output of PWM waveform

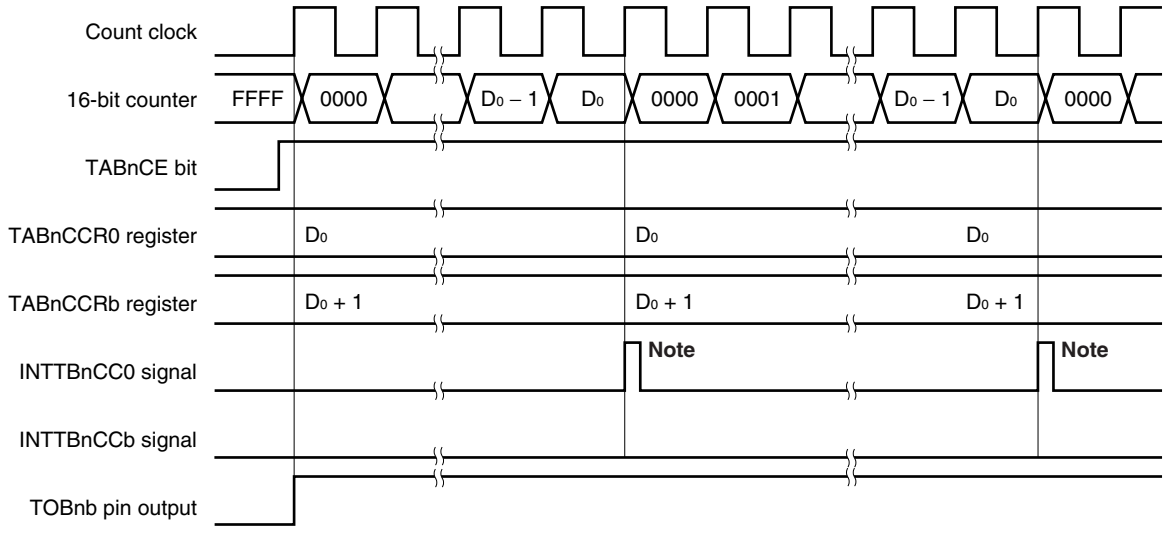
To output a 0% waveform, set the TABnCCRb register to 0000H. The 16-bit counter is cleared to 0000H and the INTTBnCC0 and INTTBnCCb signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.



Note The timing is actually delayed by one operating clock (fxx).

Remark n = 0, 1, b = 1 to 3

To output a 100% waveform, set a value of (set value of TABnCCR0 register + 1) to the TABnCCRb register. If the set value of the TABnCCR0 register is FFFFH, 100% output cannot be produced.

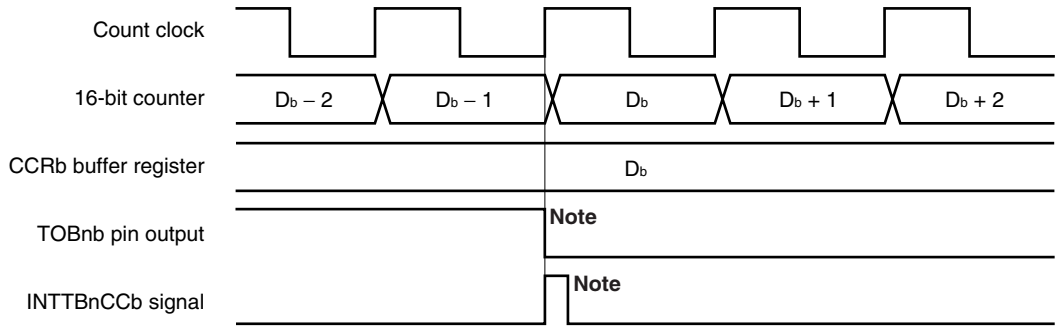


Note The timing is actually delayed by one operating clock (fxx).

Remark n = 0, 1, b = 1 to 3

(c) Generation timing of compare match interrupt request signal (INTTBnCCb)

The timing of generation of the INTTBnCCb signal in the PWM output mode differs from the timing of INTTBnCCb signals in other mode; the INTTBnCCb signal is generated when the count value of the 16-bit counter matches the value of the TABnCCRb register.



Note Actually, the timing is delayed by one operating clock (fxx).

Remark n = 0, 1
b = 1 to 3

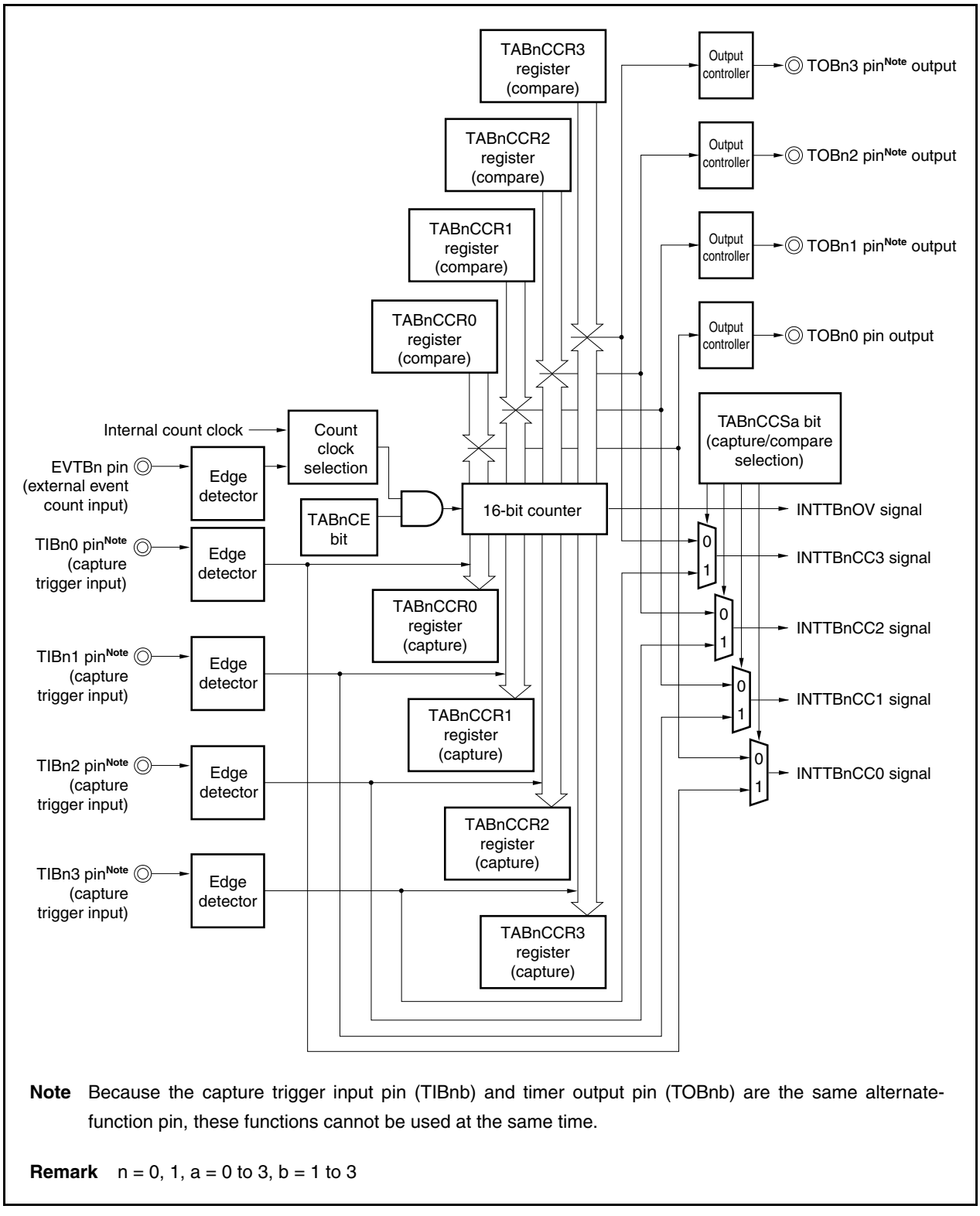
Usually, the INTTBnCCb signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TABnCCRb register.

In the PWM output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the output signal of the TOBnb pin.

7.6.6 Free-running timer mode (TABnMD2 to TABnMD0 bits = 101)

In the free-running timer mode, 16-bit timer/event counter AB starts counting when the TABnCTL0.TABnCE bit is set to 1. At this time, the TABnCCRa register can be used as a compare register or a capture register, depending on the setting of the TABnOPT0.TABnCCSa bit.

Figure 7-32. Configuration in Free-Running Timer Mode



Note Because the capture trigger input pin (TIBnb) and timer output pin (TOBnb) are the same alternate-function pin, these functions cannot be used at the same time.

Remark n = 0, 1, a = 0 to 3, b = 1 to 3

- Compare operation

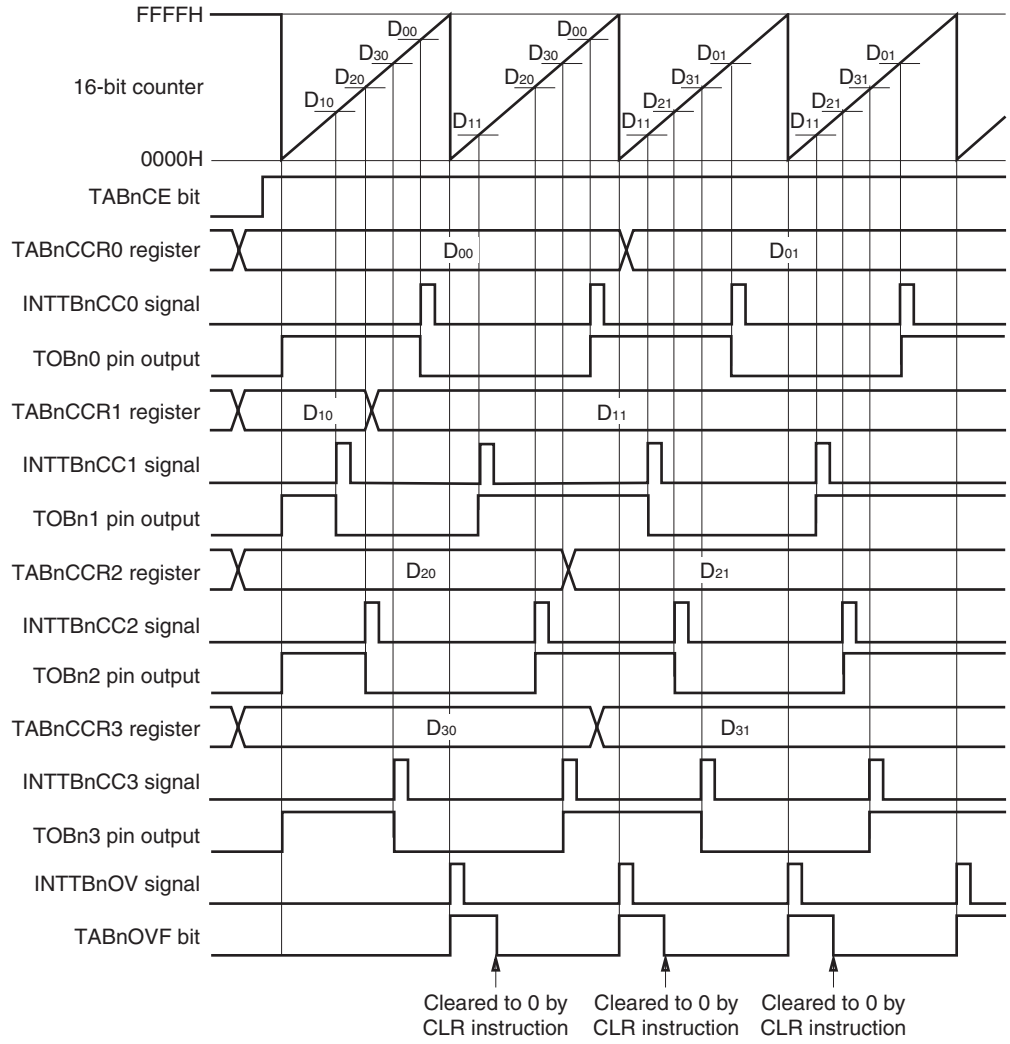
  When the TABnCE bit is set to 1, 16-bit timer/event counter AB starts counting, and the output signals of the TOBn0 to TOBn3 pins are inverted. When the count value of the 16-bit counter later matches the set value of the TABnCCRa register, a compare match interrupt request signal (INTTBnCCa) is generated, and the output signals of the TOBn0 to TOBn3 pins are inverted.

  The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTBnOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TABnOPT0.TABnOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

  The TABnCCRa register can be rewritten while the counter is operating. If it is rewritten, the new value is reflected at that time, and compared with the count value.

Remark n = 0, 1
a = 0 to 3

Figure 7-33. Basic Timing in Free-Running Timer Mode (Compare Function)



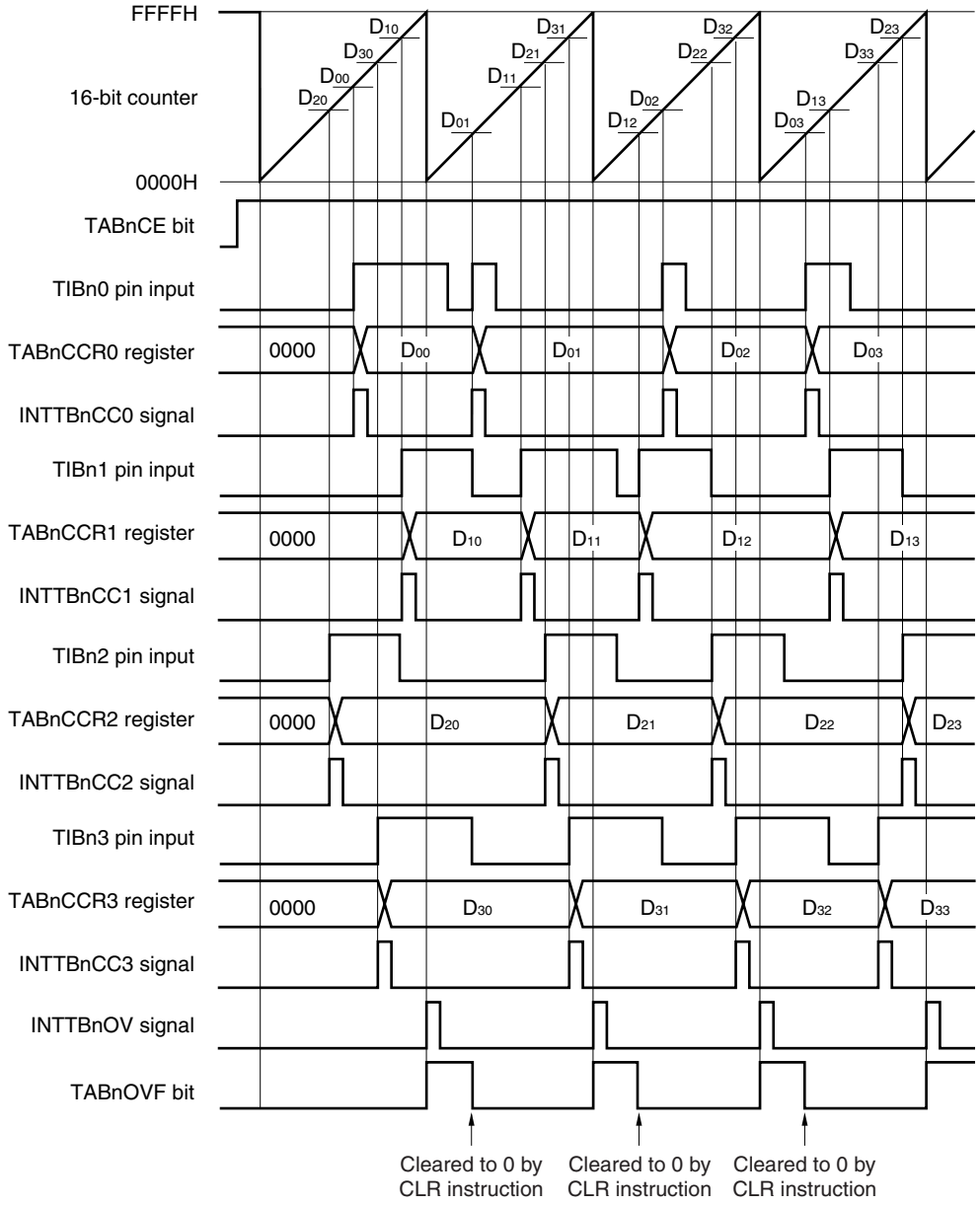
Remark n = 0, 1

- Capture operation

  When the TABnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIBna pin is detected, the count value of the 16-bit counter is stored in the TABnCCRa register, and a capture interrupt request signal (INTTBnCCa) is generated.

  The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTBnOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TABnOPT0.TABnOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

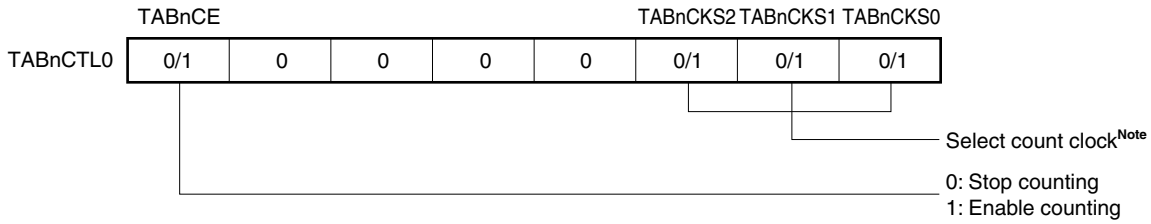
Figure 7-34. Basic Timing in Free-Running Timer Mode (Capture Function)



Remark n = 0, 1
a = 0 to 3

Figure 7-35. Register Setting in Free-Running Timer Mode (1/3)

(a) TABn control register 0 (TABnCTL0)



Note The setting is invalid when the TABnCTL1.TABnEEE bit = 1

(b) TABn control register 1 (TABnCTL1)

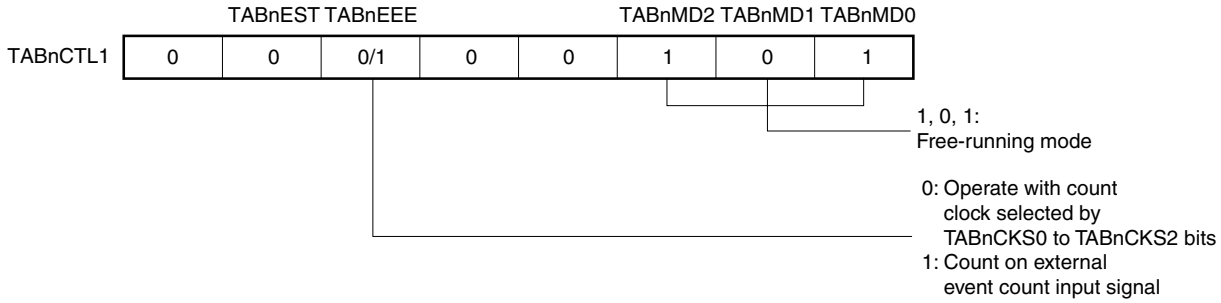
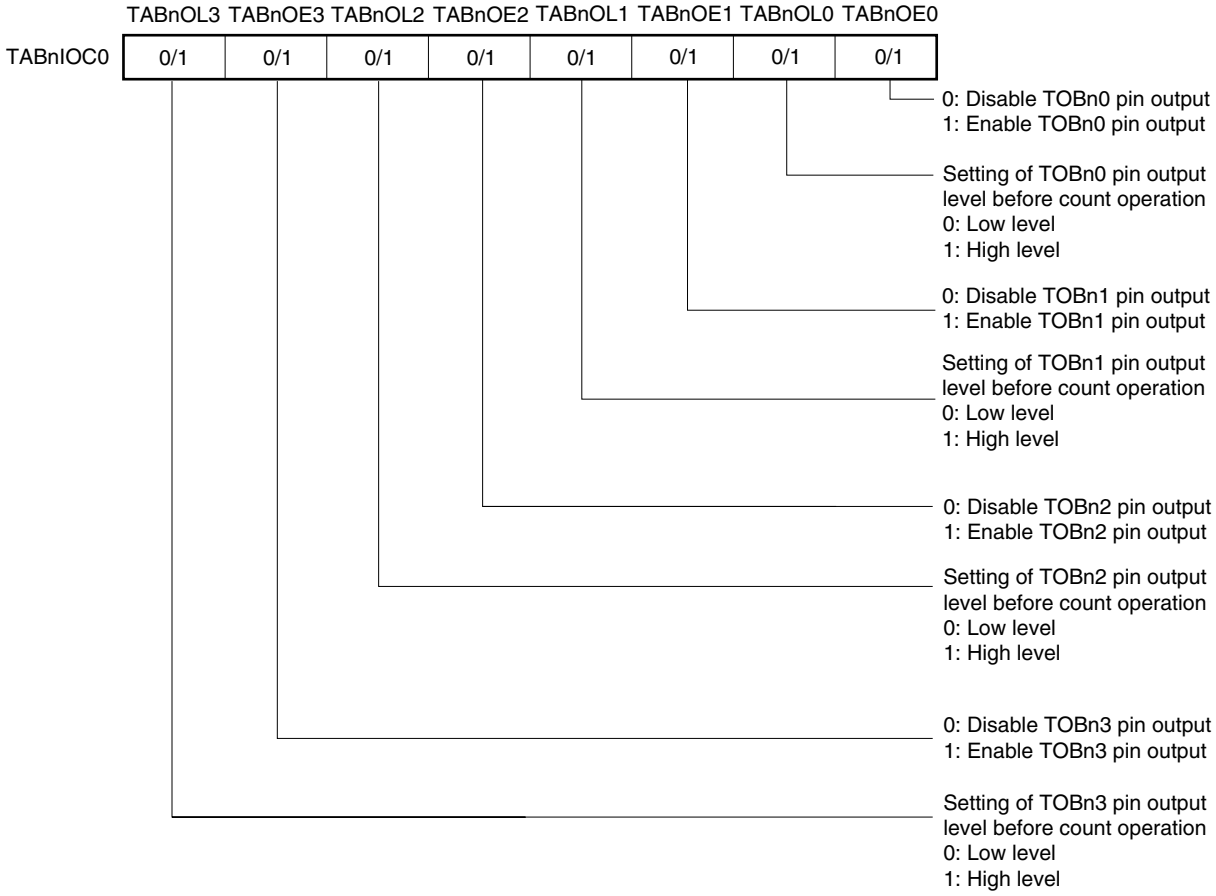


Figure 7-35. Register Setting in Free-Running Timer Mode (2/3)

(c) TABn I/O control register 0 (TABnIOC0)



(d) TABn I/O control register 1 (TABnIOC1)

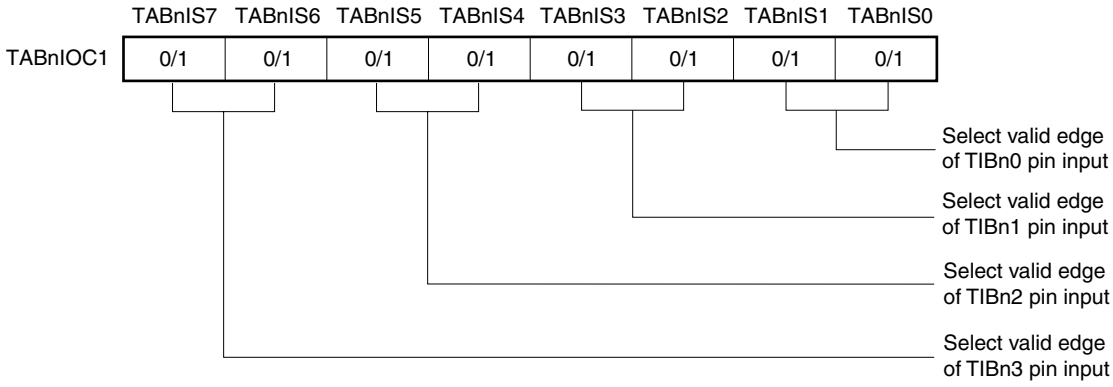
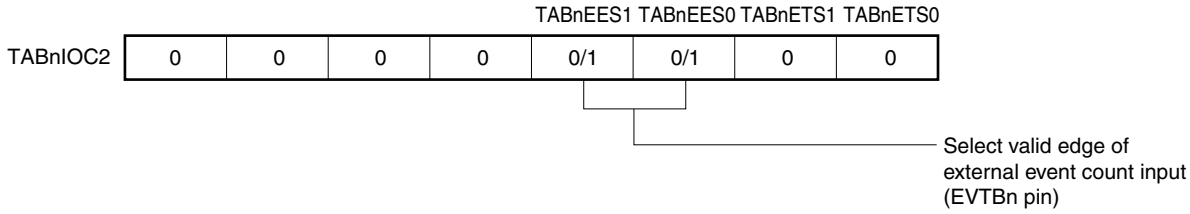
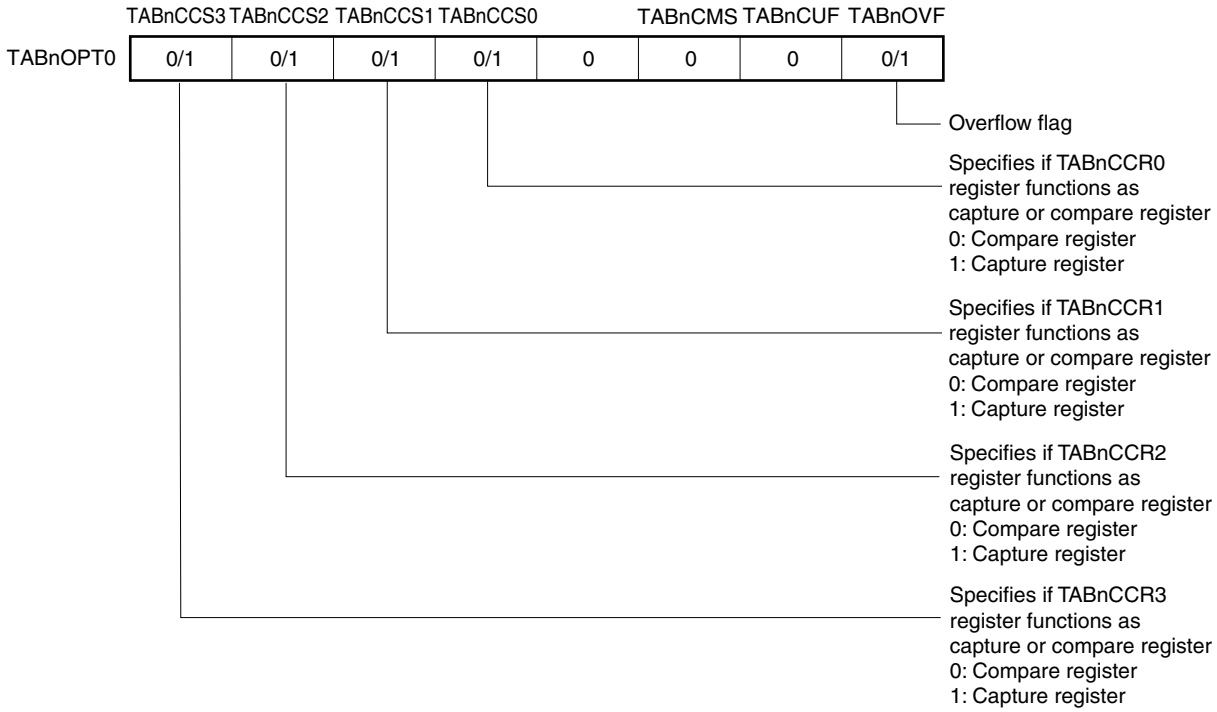


Figure 7-35. Register Setting in Free-Running Timer Mode (3/3)

(e) TABn I/O control register 2 (TABnIOC2)



(f) TABn option register 0 (TABnOPT0)



(g) TABn counter read buffer register (TABnCNT)

The value of the 16-bit counter can be read by reading the TABnCNT register.

(h) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)

These registers function as capture registers or compare registers depending on the setting of the TABnOPT0.TABnCCSa bit.

When the registers function as capture registers, they store the count value of the 16-bit counter when the valid edge input to the TIBna pin is detected.

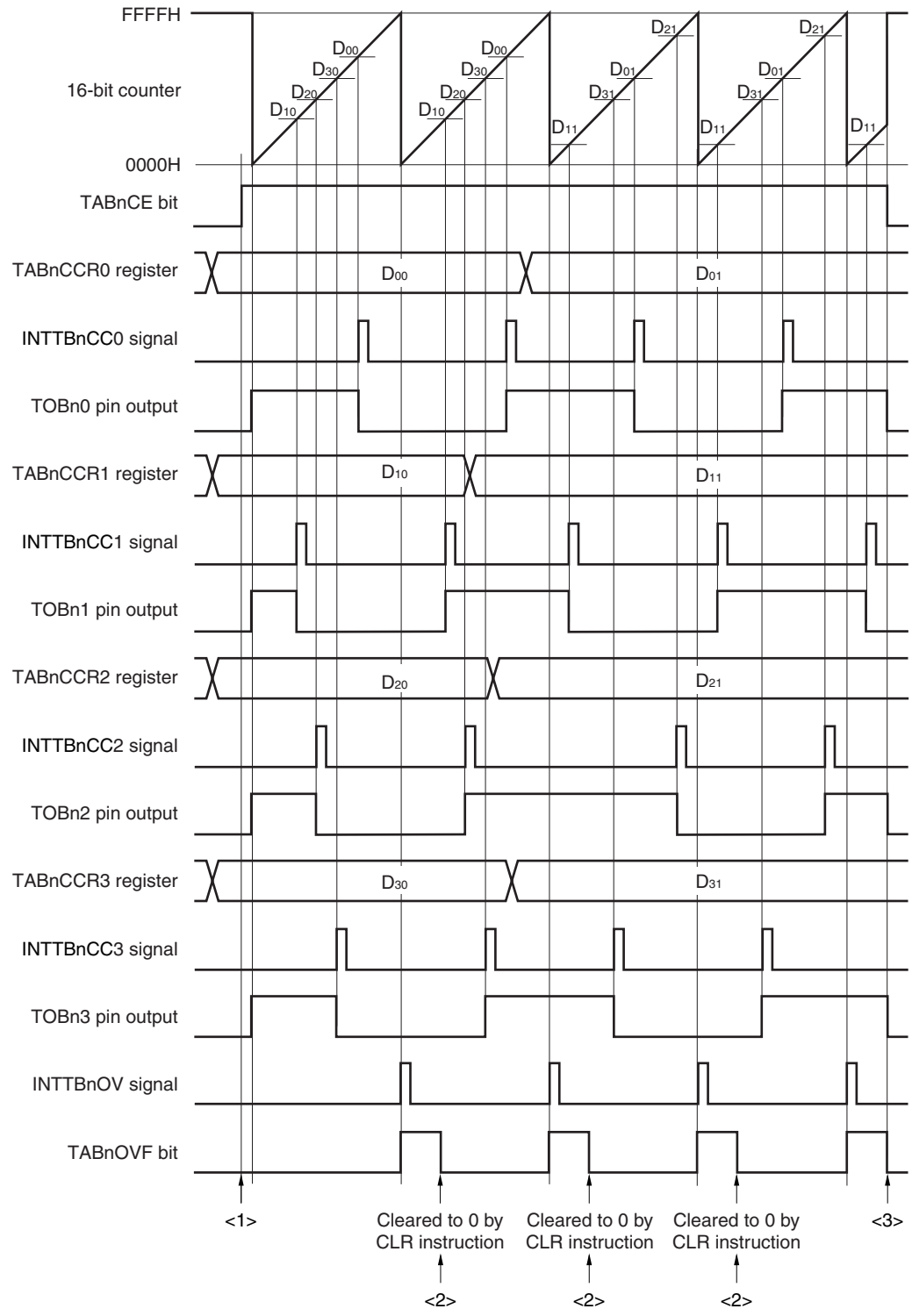
When the registers function as compare registers and when D_a is set to the TABnCCRa register, the INTTBnCCa signal is generated when the counter reaches (D_a + 1), and the output signals of the TOBn0 to TOBn3 pins are inverted.

Remark n = 0, 1
a = 0 to 3

(1) Operation flow in free-running timer mode

(a) When using capture/compare register as compare register

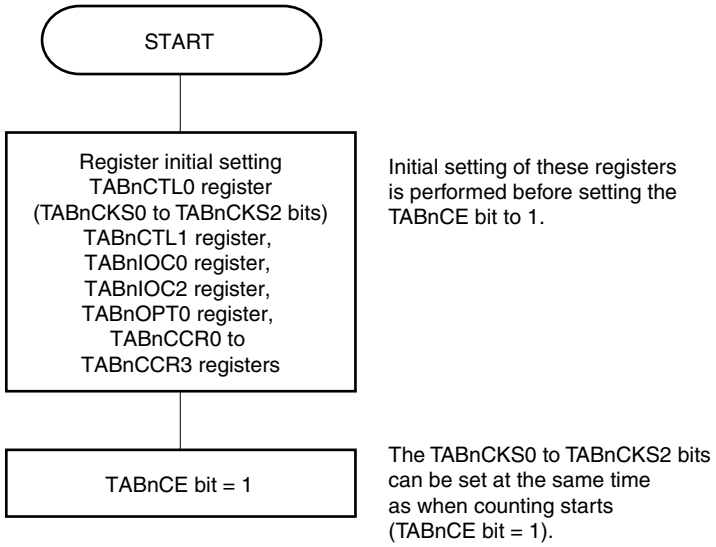
Figure 7-36. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)



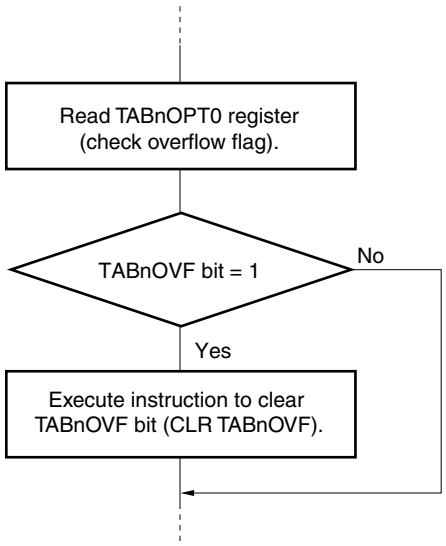
Remark n = 0, 1

Figure 7-36. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)

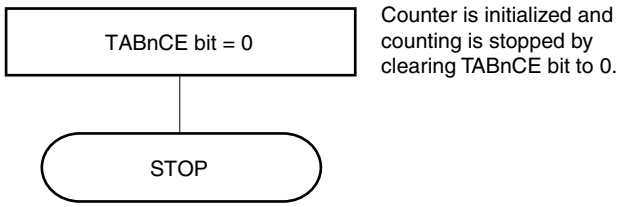
<1> Count operation start flow



<2> Overflow flag clear flow



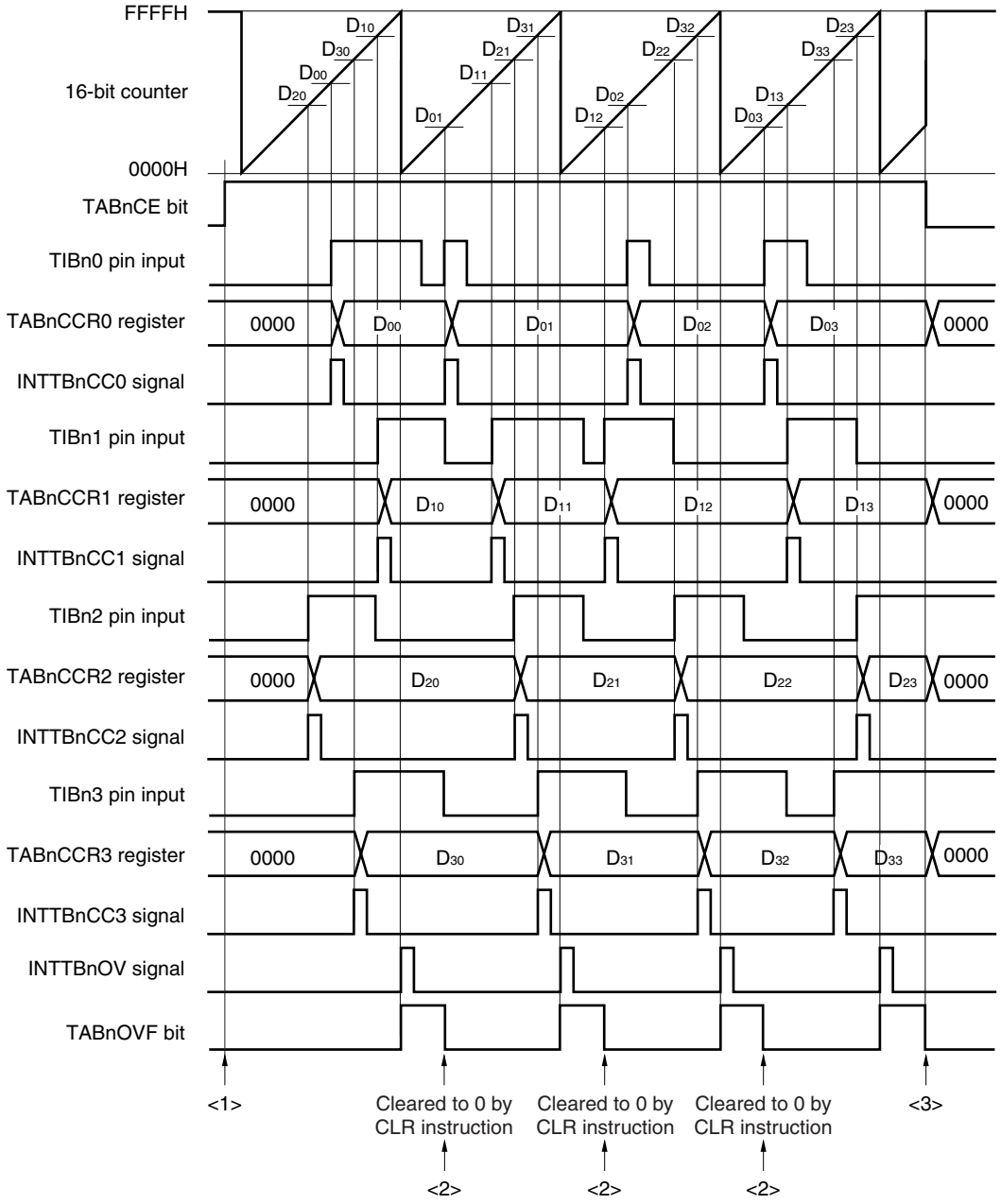
<3> Count operation stop flow



Remark n = 0, 1

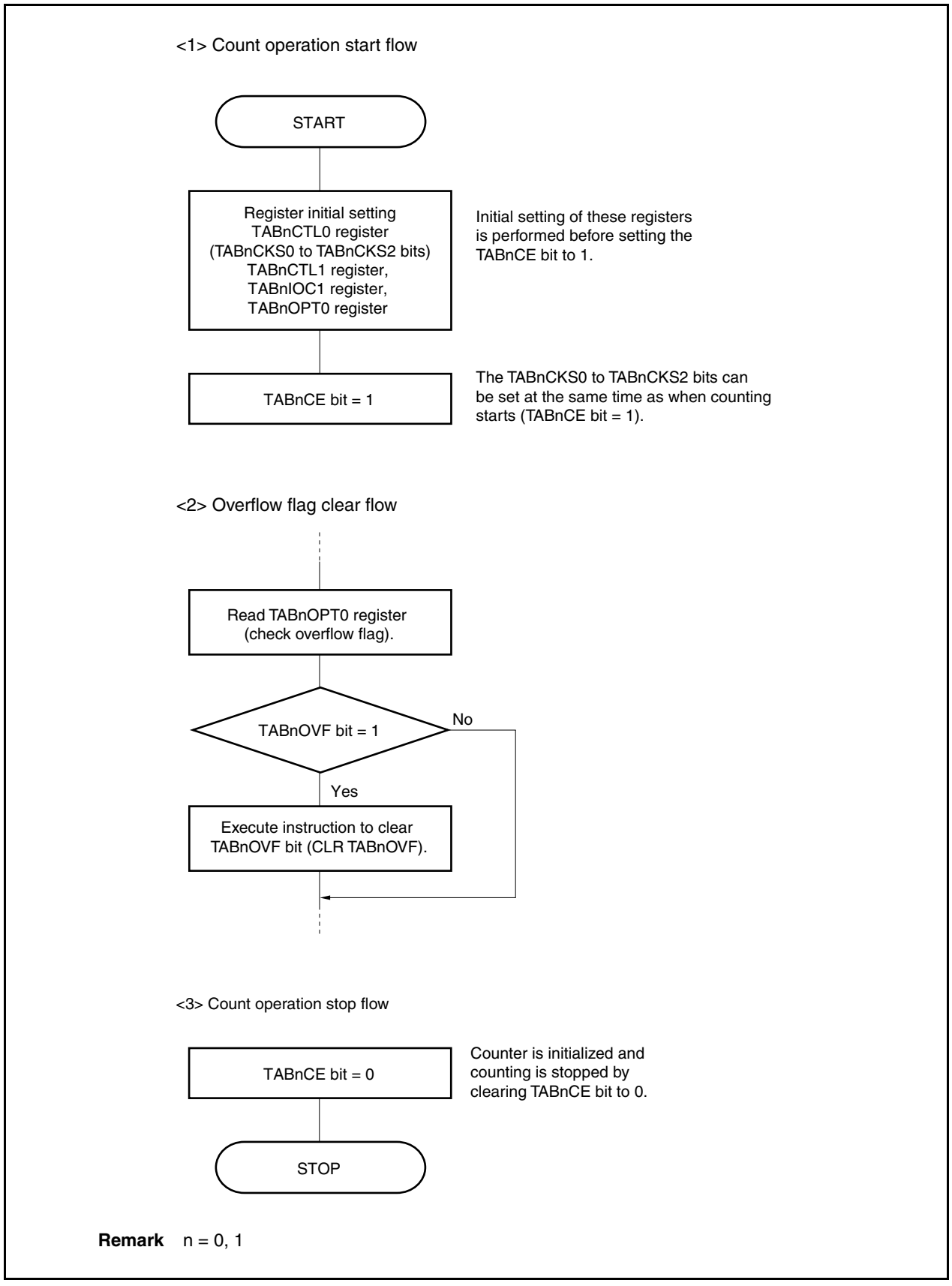
(b) When using capture/compare register as capture register

Figure 7-37. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)



Remark n = 0, 1

Figure 7-37. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)

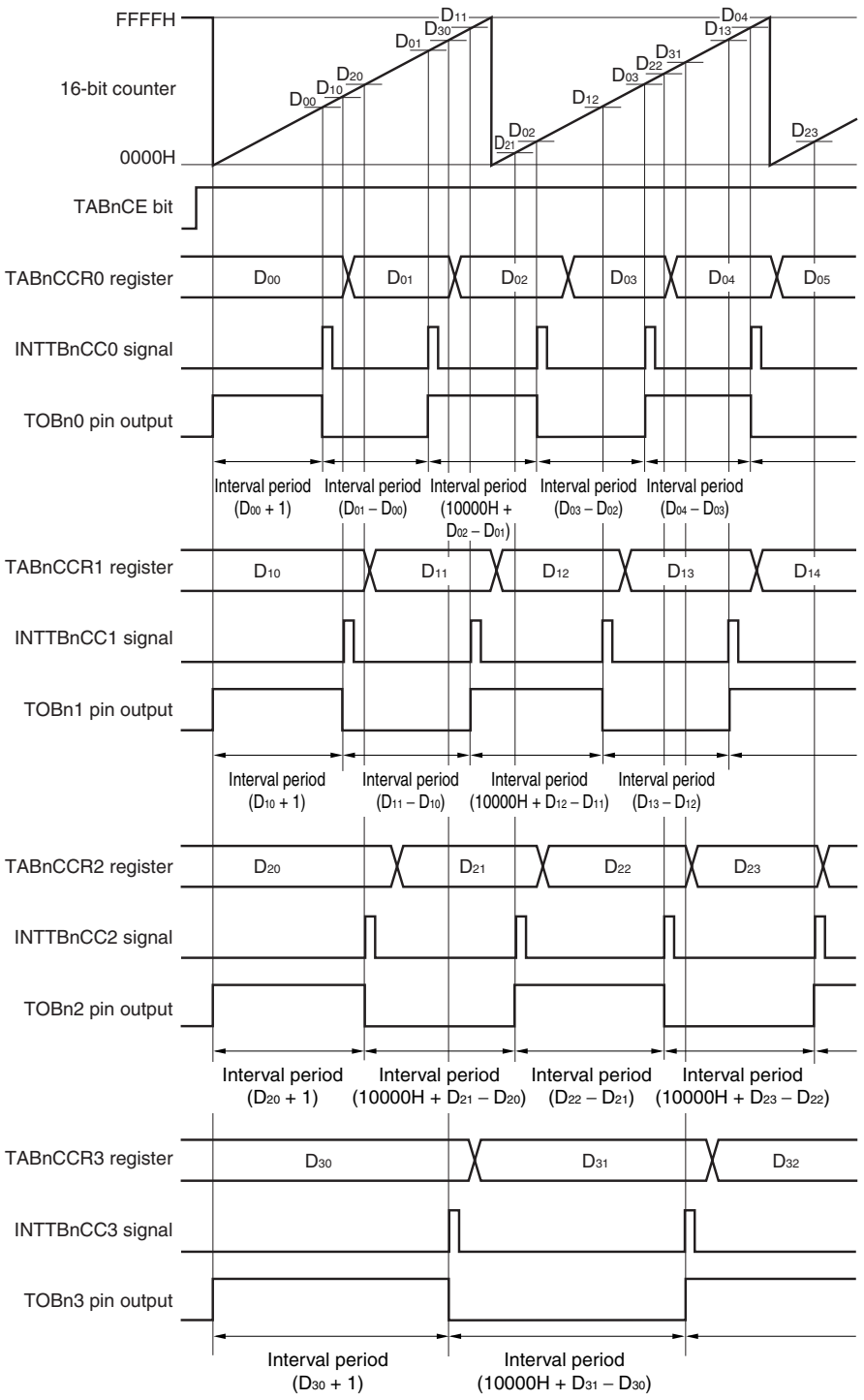


Remark n = 0, 1

(2) Operation timing in free-running timer mode

(a) Interval operation with compare register

When 16-bit timer/event counter AB is used as an interval timer with the TABnCCRa register used as a compare register, software processing is necessary for setting a comparison value to generate the next interrupt request signal each time the INTTBnCCa signal has been detected.



Remark n = 0, 1, a = 0 to 3

When performing an interval operation in the free-running timer mode, four intervals can be set with one channel.

To perform the interval operation, the value of the corresponding TABnCCRa register must be re-set in the interrupt servicing that is executed when the INTTBnCCa signal is detected.

The set value for re-setting the TABnCCRa register can be calculated by the following expression, where "D_a" is the interval period.

Compare register default value: $D_a - 1$

Value set to compare register second and subsequent time: Previous set value + D_a

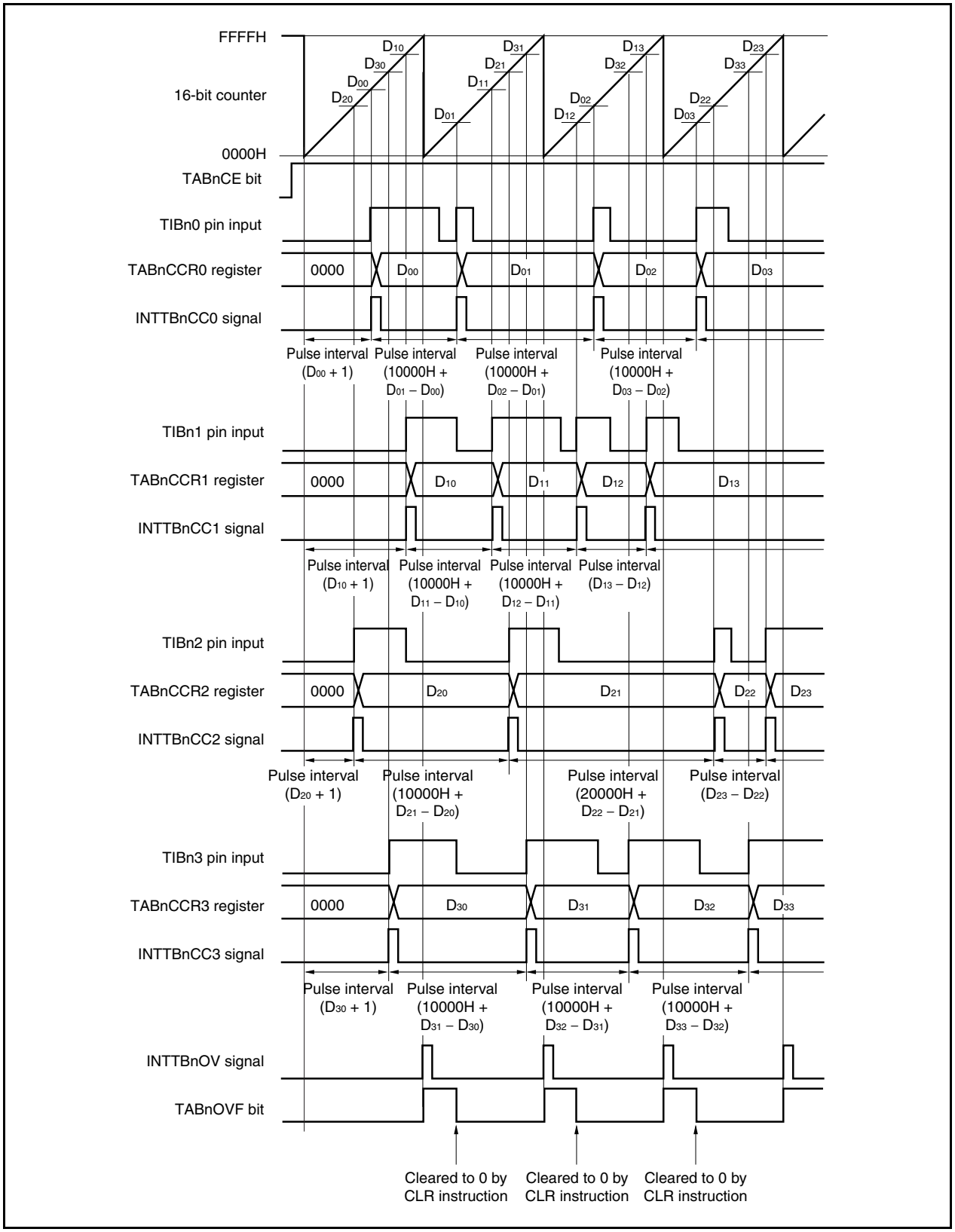
(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

Remark n = 0, 1

a = 0 to 3

(b) Pulse width measurement with capture register

When pulse width measurement is performed with the TABnCCRa register used as a capture register, software processing is necessary for reading the capture register each time the INTTBnCCa signal has been detected and for calculating an interval.



When executing pulse width measurement in the free-running timer mode, four pulse widths can be measured with one channel.

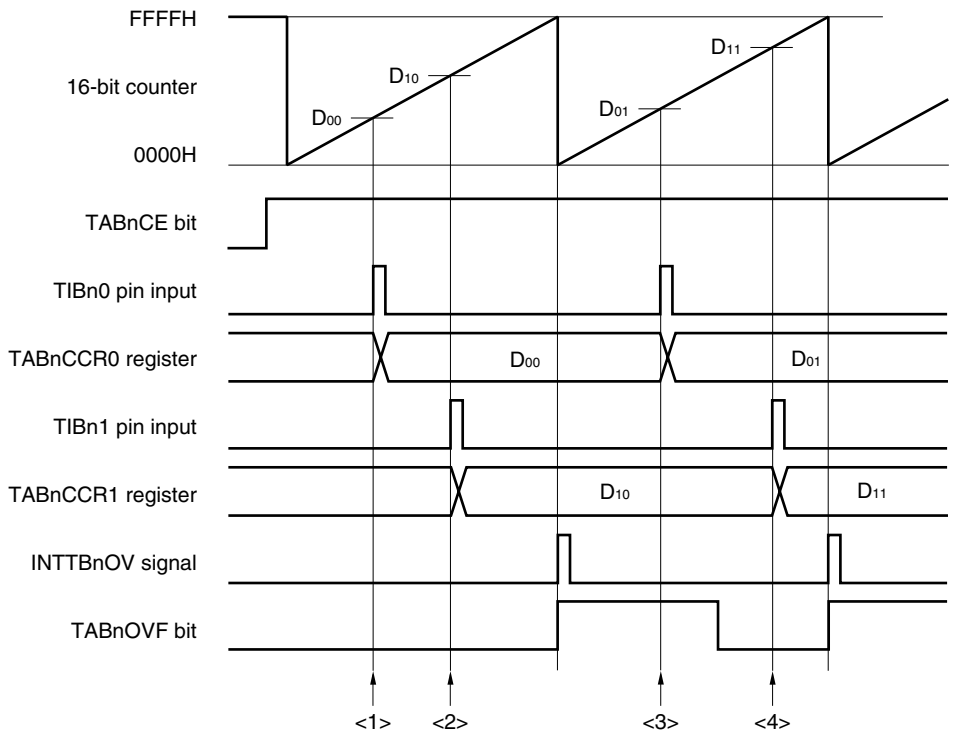
To measure a pulse width, the pulse width can be calculated by reading the value of the TABnCCRa register in synchronization with the INTTBnCCa signal, and calculating the difference between the read value and the previously read value.

Remark n = 0, 1
a = 0 to 3

(c) Processing of overflow when two capture registers are used

Care must be exercised in processing the overflow flag when two capture registers are used. First, an example of incorrect processing is shown below.

Example of incorrect processing when two capture registers are used



The following problem may occur when two pulse widths are measured in the free-running timer mode.

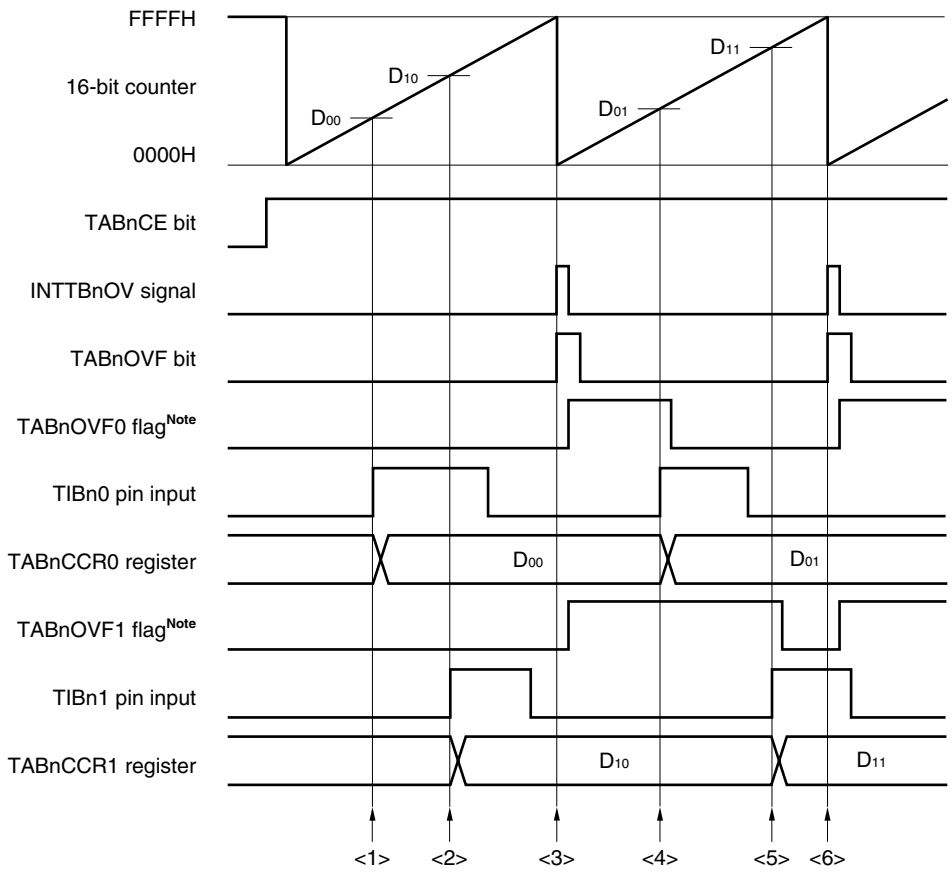
<1> Read the TABnCCR0 register (setting of the default value of the TIBn0 pin input).
<2> Read the TABnCCR1 register (setting of the default value of the TIBn1 pin input).
<3> Read the TABnCCR0 register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} − D_{00}).
<4> Read the TABnCCR1 register.
Read the overflow flag. Because the flag is cleared in <3>, 0 is read.
Because the overflow flag is 0, the pulse width can be calculated by (D_{11} − D_{10}) (incorrect).

Remark n = 0, 1

When two capture registers are used, and if the overflow flag is cleared to 0 by one capture register, the other capture register may not obtain the correct pulse width.
Use software when using two capture registers. An example of how to use software is shown below.

(1/2)

Example when two capture registers are used (using overflow interrupt)



Note The TABnOVF0 and TABnOVF1 flags are set on the internal RAM by software.

<1> Read the TABnCCR0 register (setting of the default value of the TIBn0 pin input).

<2> Read the TABnCCR1 register (setting of the default value of the TIBn1 pin input).

<3> An overflow occurs. Set the TABnOVF0 and TABnOVF1 flags to 1 in the overflow interrupt servicing, and clear the overflow flag to 0.

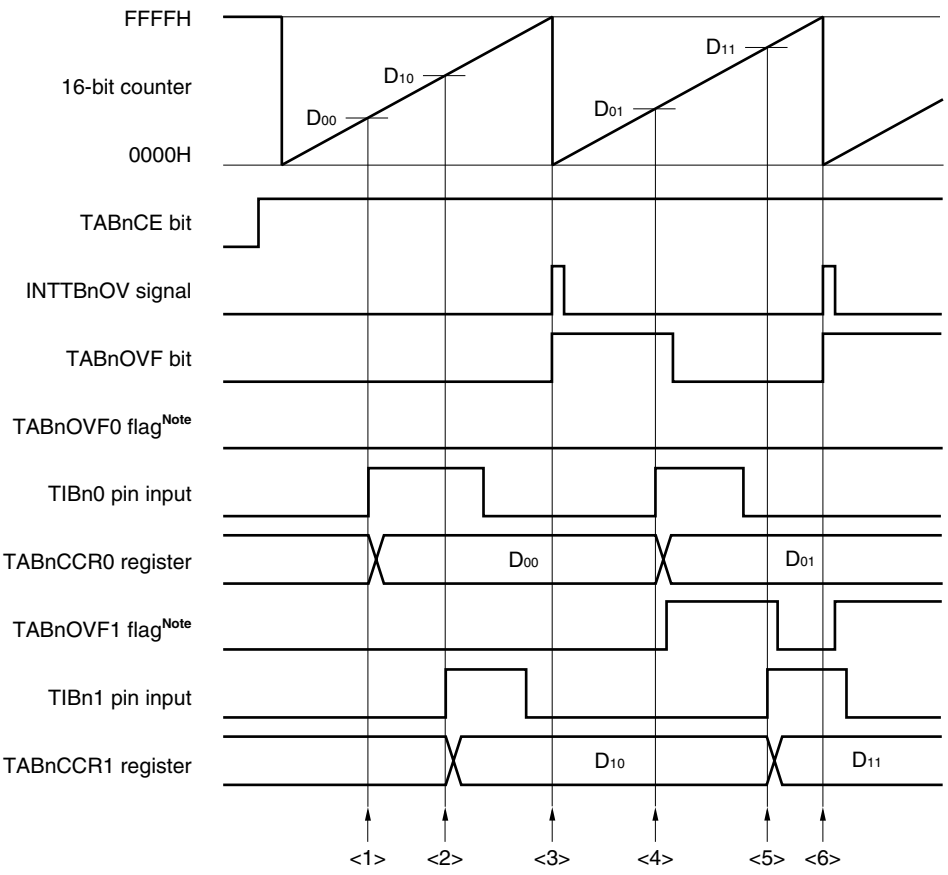
<4> Read the TABnCCR0 register.
Read the TABnOVF0 flag. If the TABnOVF0 flag is 1, clear it to 0.
Because the TABnOVF0 flag is 1, the pulse width can be calculated by (10000H + D_{01} − D_{00}).

<5> Read the TABnCCR1 register.
Read the TABnOVF1 flag. If the TABnOVF1 flag is 1, clear it to 0 (the TABnOVF0 flag is cleared in <4>, and the TABnOVF1 flag remains 1).
Because the TABnOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} − D_{10}) (correct).

<6> Same as <3>

Remark n = 0, 1

Example when two capture registers are used (without using overflow interrupt)



Note The TABnOVF0 and TABnOVF1 flags are set on the internal RAM by software.

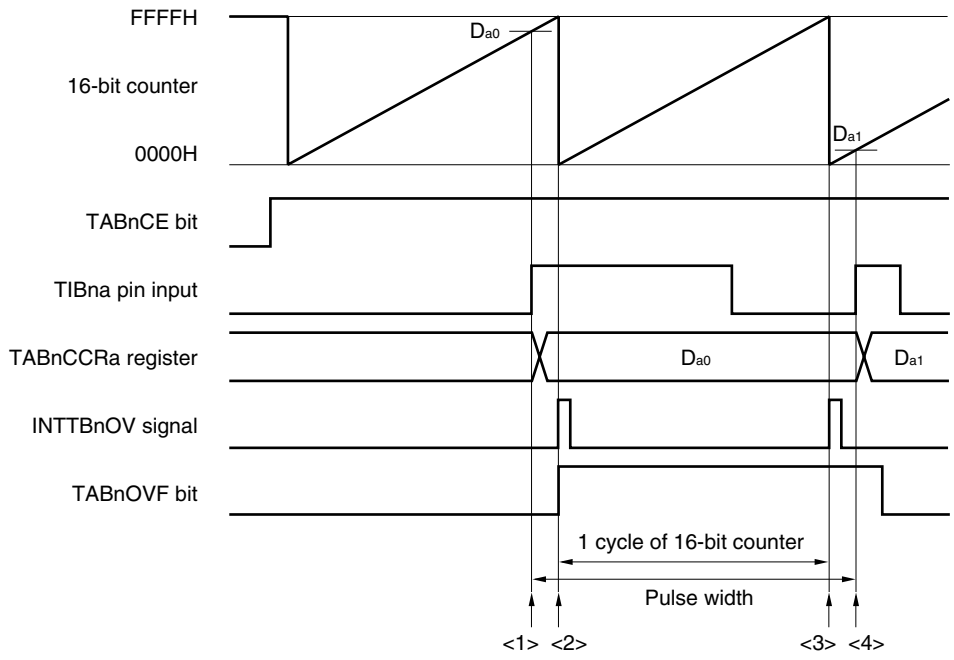
<1> Read the TABnCCR0 register (setting of the default value of the TIBn0 pin input).
<2> Read the TABnCCR1 register (setting of the default value of the TIBn1 pin input).
<3> An overflow occurs. Nothing is done by software.
<4> Read the TABnCCR0 register.
Read the overflow flag. If the overflow flag is 1, set only the TABnOVF1 flag to 1, and clear the overflow flag to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + $D_{01} - D_{00}$).
<5> Read the TABnCCR1 register.
Read the overflow flag. Because the overflow flag is cleared in <4>, 0 is read.
Read the TABnOVF1 flag. If the TABnOVF1 flag is 1, clear it to 0.
Because the TABnOVF1 flag is 1, the pulse width can be calculated by (10000H + $D_{11} - D_{10}$) (correct).
<6> Same as <3>

Remark n = 0, 1

(d) Processing of overflow if capture trigger interval is long

If the pulse width is greater than one cycle of the 16-bit counter, care must be exercised because an overflow may occur more than once from the first capture trigger to the next. First, an example of incorrect processing is shown below.

Example of incorrect processing when capture trigger interval is long



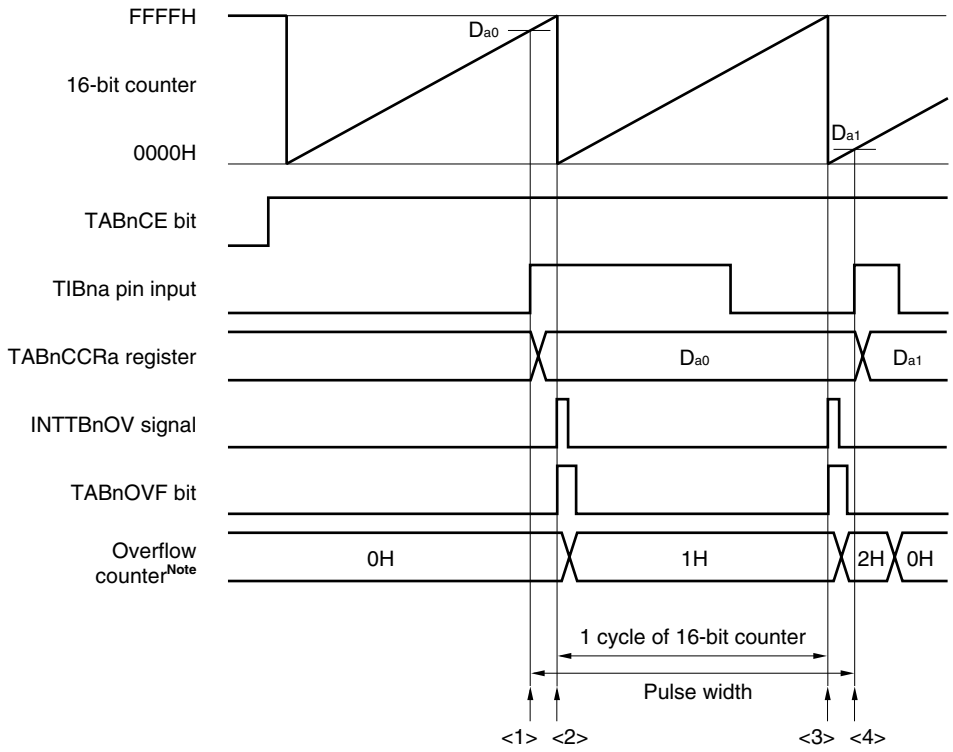
The following problem may occur when a long pulse width in the free-running timer mode.

<1> Read the TABnCCRa register (setting of the default value of the TIBna pin input).
<2> An overflow occurs. Nothing is done by software.
<3> An overflow occurs a second time. Nothing is done by software.
<4> Read the TABnCCRa register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{a1} – D_{a0}) (incorrect).
Actually, the pulse width must be (20000H + D_{a1} – D_{a0}) because an overflow occurs twice.

Remark n = 0, 1, a = 0 to 3

If an overflow occurs twice or more when the capture trigger interval is long, the correct pulse width may not be obtained.
If the capture trigger interval is long, slow the count clock to lengthen one cycle of the 16-bit counter, or use software. An example of how to use software is shown next.

Example when capture trigger interval is long



Note The overflow counter is set arbitrarily by software on the internal RAM.

<1> Read the TABnCCRa register (setting of the default value of the TIBna pin input).

<2> An overflow occurs. Increment the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.

<3> An overflow occurs a second time. Increment (+1) the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.

<4> Read the TABnCCRa register.

Read the overflow counter.

→ When the overflow counter is “N”, the pulse width can be calculated by (N × 10000H + D_{a1} – D_{a0}).

In this example, the pulse width is (20000H + D_{a1} – D_{a0}) because an overflow occurs twice.

Clear the overflow counter (0H).

Remark n = 0, 1

a = 0 to 3

(e) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TABnOVF bit to 0 with the CLR instruction after reading the TABnOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TABnOPT0 register after reading the TABnOVF bit when it is 1.

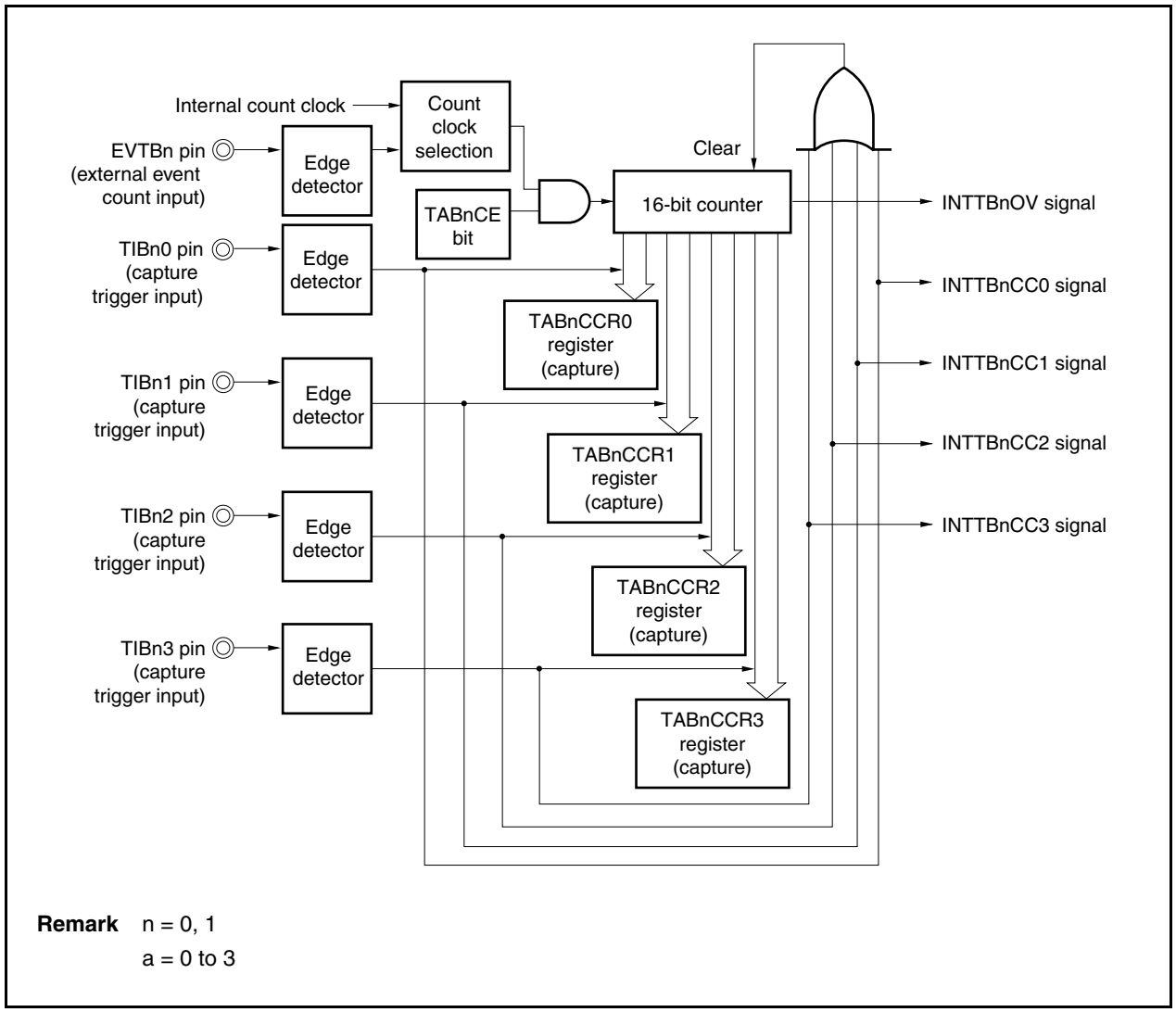
7.6.7 Pulse width measurement mode (TABnMD2 to TABnMD0 bits = 110)

In the pulse width measurement mode, 16-bit timer/event counter AB starts counting when the TABnCTL0.TABnCE bit is set to 1. Each time the valid edge input to the TIBna pin has been detected, the count value of the 16-bit counter is stored in the TABnCCRa register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TABnCCRa register after a capture interrupt request signal (INTTBnCCa) occurs.

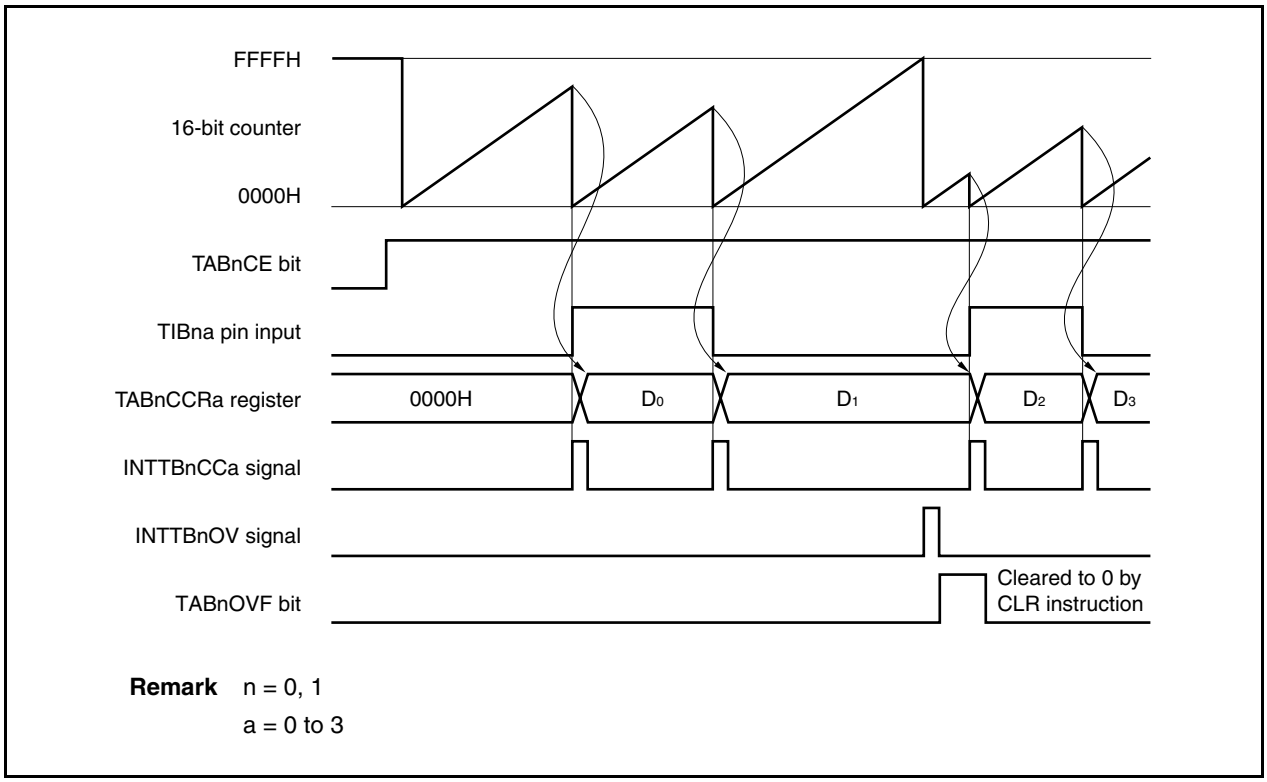
As shown in Figure 7-39, select either of the TIBn0 to TIBn3 pins as the capture trigger input pin. Specify “No edge detection” by using the TABnIOC1 register for the unused pins.

Figure 7-38. Configuration in Pulse Width Measurement Mode



Remark n = 0, 1
a = 0 to 3

Figure 7-39. Basic Timing in Pulse Width Measurement Mode



Remark n = 0, 1
a = 0 to 3

When the TABnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIBna pin is later detected, the count value of the 16-bit counter is stored in the TABnCCRa register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTTBnCCa) is generated.

The pulse width is calculated as follows.

Pulse width = Captured value × Count clock cycle

If the valid edge is not input even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTBnOV) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TABnOPT0.TABnOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction via software.

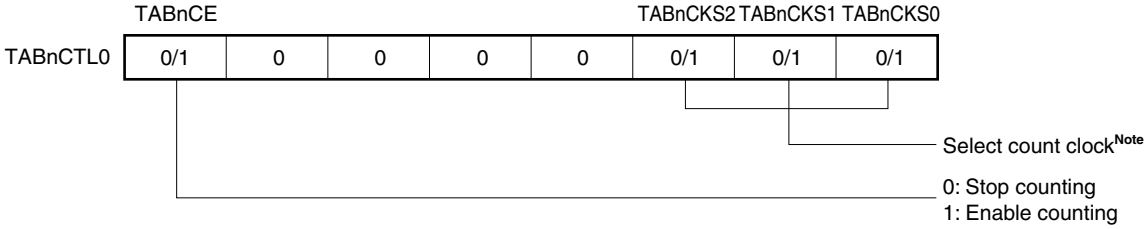
If the overflow flag is set to 1, the pulse width can be calculated as follows.

Pulse width = (10000H × TABnOVF bit set (1) count + Captured value) × Count clock cycle

Remark n = 0, 1
a = 0 to 3

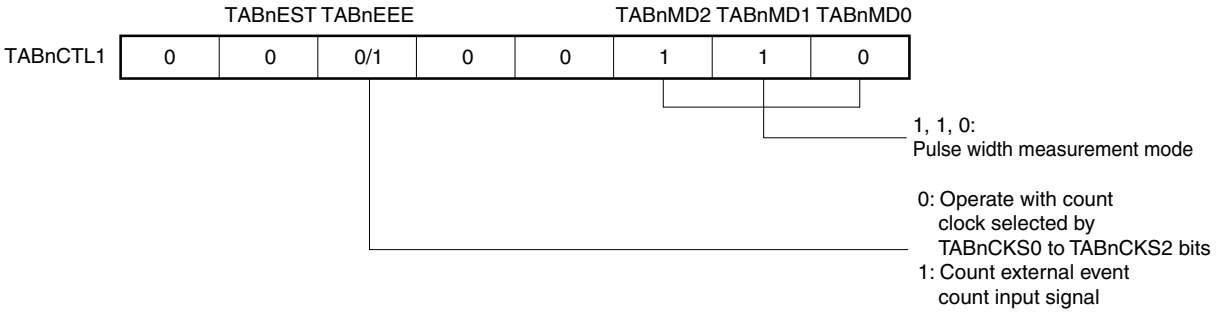
Figure 7-40. Register Setting in Pulse Width Measurement Mode (1/2)

(a) TABn control register 0 (TABnCTL0)

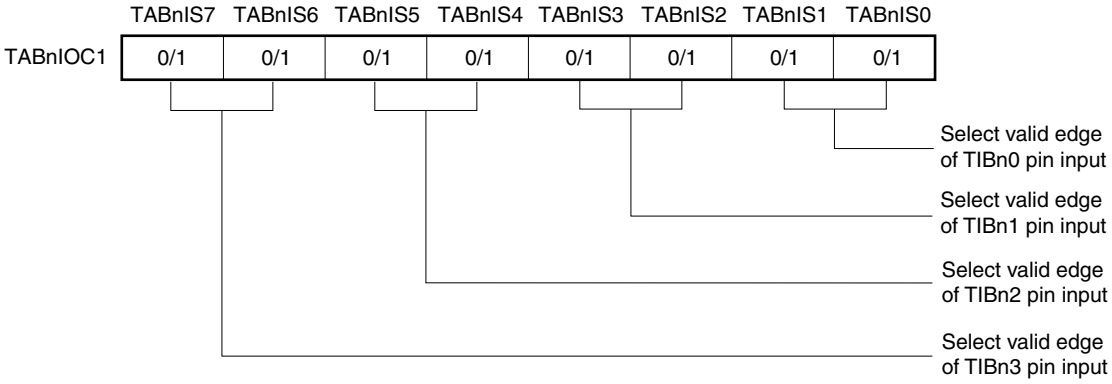


Note Setting is invalid when the TABnCTL1.TABnEEE bit = 1.

(b) TABn control register 1 (TABnCTL1)



(c) TABn I/O control register 1 (TABnIOC1)



(d) TABn I/O control register 2 (TABnIOC2)

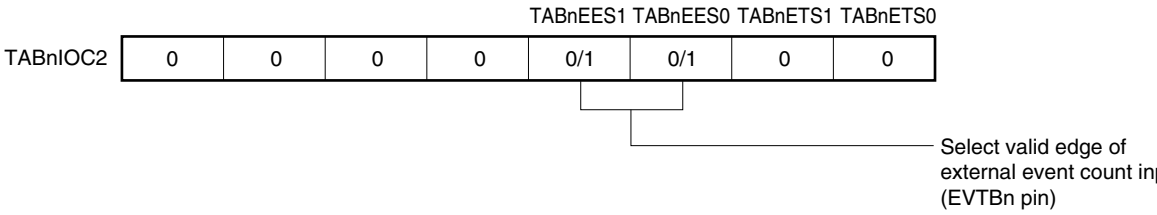


Figure 7-40. Register Setting in Pulse Width Measurement Mode (2/2)

(e) TABn option register 0 (TABnOPT0)

	TABnCCS3	TABnCCS2	TABnCCS1	TABnCCS0		TABnCMS	TABnCUF	TABnOVF
TABnOPT0	0	0	0	0	0	0	0	0/1

Overflow flag

(f) TABn counter read buffer register (TABnCNT)

The value of the 16-bit counter can be read by reading the TABnCNT register.

(g) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)

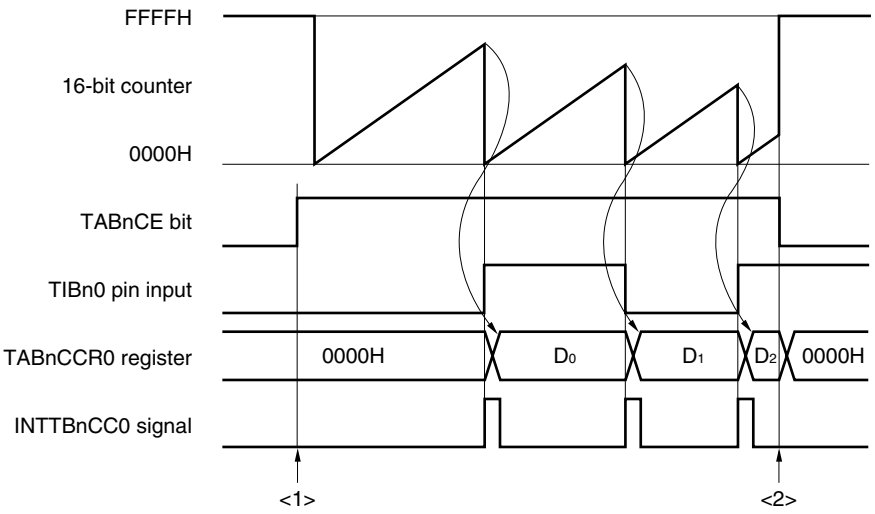
These registers store the count value of the 16-bit counter when the valid edge input to the TIBna pin is detected.

Remarks 1. TABn I/O control register 0 (TABnIOC0) is not used in the pulse width measurement mode.

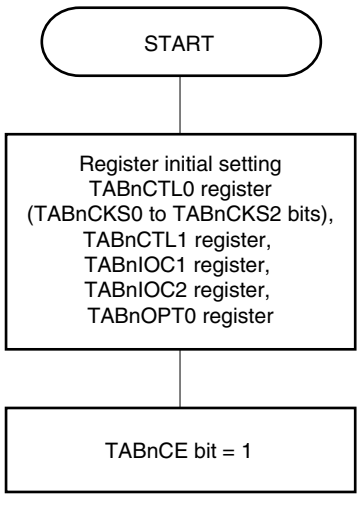
2. n = 0, 1
a = 0 to 3

(1) Operation flow in pulse width measurement mode

Figure 7-41. Software Processing Flow in Pulse Width Measurement Mode



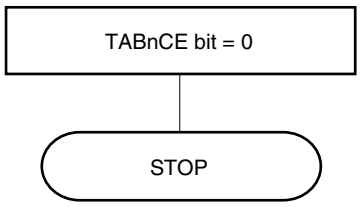
<1> Count operation start flow



Initial setting of these registers is performed before setting the TABnCE bit to 1.

The TABnCKS0 to TABnCKS2 bits can be set at the same time as when counting starts (TABnCE bit = 1).

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TABnCE bit to 0.

Remark n = 0, 1

(2) Operation timing in pulse width measurement mode

(a) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TABnOVF bit to 0 with the CLR instruction after reading the TABnOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TABnOPT0 register after reading the TABnOVF bit when it is 1.

CHAPTER 8 16-BIT TIMER/EVENT COUNTER T (TMT)

Timer T (TMT) is a 16-bit timer/event counter.

An encoder count function and other functions are added to the timer AA (TAA). However, TMT does not have a function to operate with an external event count input when it operates in the interval timer mode.

The V850E/IF3 and V850E/IG3 incorporate TMT0 and TMT1.

8.1 Overview

The TMTn channels are outlined below (n = 0, 1).

Table 8-1. TMTn Overview

Item	TMT0	TMT1
Clock selection	8 ways	8 ways
Capture trigger input pin	**Note 1**	2
External event count input pin	**Note 2**	1
External trigger input pin	**Note 2**	1
Timer counter	1	1
Capture/compare register	2 Note 3	2
Capture/compare match interrupt request signal	2 Note 3	2
Overflow interrupt request signal	1	1
Encoder clear interrupt request signal	**Note 1**	1
Timer output pin	**Note 4**	2

Notes 1. V850E/IF3: None
V850E/IG3: 2

2. V850E/IF3: None
V850E/IG3: 1

3. In the V850E/IF3, compare function only

8.2 Functions

The TMTn functions that can be implemented differ from one channel to another, as shown in the table below (n = 0, 1).

Table 8-2. TMTn Functions

Function	TMT0	TMT1
Interval timer	√	√
External event counter	**Note 1**	√
External trigger pulse output	**Note 1**	√
One-shot pulse output	**Note 1**	√
PWM output	**Note 1**	√
Free-running timer	√Note 2	√
Pulse width measurement	×	√
Triangular-wave PWM output mode	**Note 1**	√
Encoder count function	**Note 1**	√

Notes 1. V850E/IF3: ×
V850E/IG3: √

2. In the V850E/IF3, compare function only

8.3 Configuration

TMTn includes the following hardware (n = 0, 1).

Table 8-3. Configuration of TMTn

Item	Configuration
Timer register	16-bit counter × 1
Registers	TMTn capture/compare registers 0, 1 (TTnCCR0, TTnCCR1) TMTn counter read buffer register (TTnCNT) TMTm counter write register (TTmTCW) CCR0 and CCR1 buffer registers
Timer input	2 in total (TIT00[Note 1], TIT01[Note 1], TIT10, TIT11, EVTT0[Note 1], EVTT1, TENC00[Note 1], TENC01[Note 1], TENC10, TENC11, TECR0[Note 1], TECR1 pins)[Note 2]
Timer output	4 in total (TOT00[Note 1], TOT01[Note 1], TOT10, TOT11 pins)[Note 2]
Control registers	TMTn control registers 0, 1 (TTnCTL0, TTnCTL1) TMTm control register 2 (TTmCTL2) TMTm I/O control registers 0 to 3 (TTmIOC0 to TTmIOC3) TMTn option register 0 (TTnOPT0) TMTm option register 1 (TTmOPT1) TMTm capture input select register (TTISLm)

Notes 1. V850E/IG3 only

2. TIT00/TECR0 and TIT10/TECR1 pins function alternately as capture trigger input pins (TIT00, TIT10), encoder clear input pins (TECR0, TECR1), and timer output pins (TOT00, TOT10).

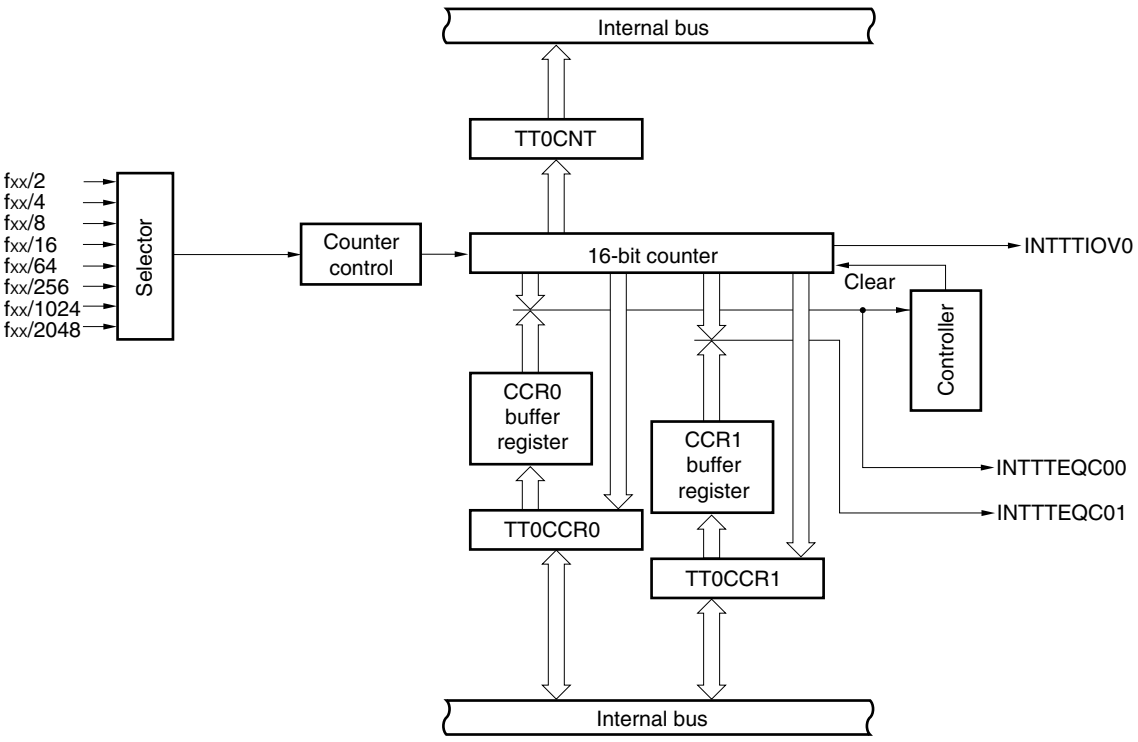
TENC00/EVTT0 and TENC10/EVTT1 pins function alternately as encoder input pins (TENC00, TENC10), external event count input pins (EVTT0, EVTT1), and external trigger input pins (EVTT0, EVTT1).

TIT01/TENC01 and TIT11/TENC11 pins function alternately as capture trigger input pins (TIT01, TIT11), encoder input pins (TENC01, TENC11), and timer output pins (TOT01, TOT11).

Remark V850E/IF3: n = 0, 1, m = 1

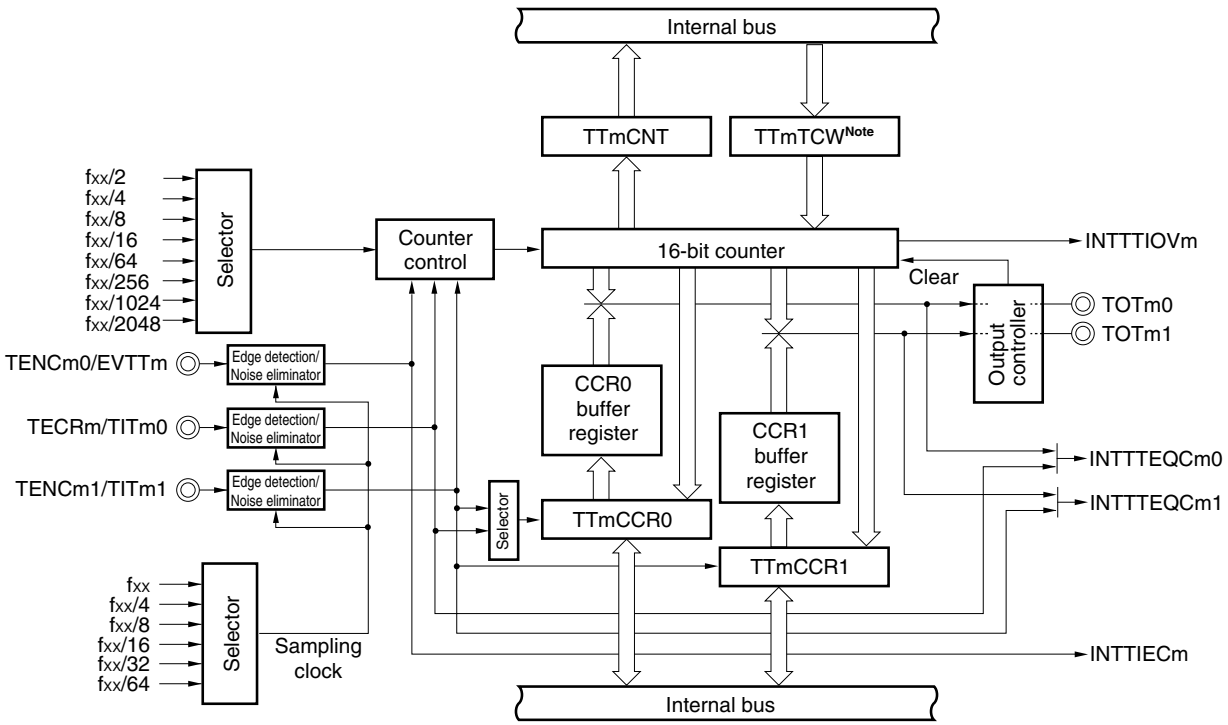
V850E/IG3: n = 0, 1, m = 0, 1

Figure 8-1. Block Diagram of TMT0 for V850E/IF3



Remark fxx: Peripheral clock

Figure 8-2. Block Diagram of TMT1 for V850E/IF3 and TMT0 and TMT1 for V850E/IG3



Note The initial value set from the TTmTCW register to the 16-bit counter is valid only in the encoder compare mode.

Rewrite the TTmTCW register when the TTmCTL0.TTmCE bit = 0.

The value of the TTmTCW register is transferred to the 16-bit counter when the TTmCE bit = 1.

Remarks 1. fxx: Peripheral clock

2. For the noise eliminator, see **4.6 Noise Eliminator**.

3. V850E/IF3: m = 1

V850E/IG3: m = 0, 1

(1) 16-bit counter

This 16-bit counter can count internal clocks or external events.

The count value of this counter can be read by using the TTnCNT register.

When the TTnCTL0.TTnCE bit = 0, the value of the 16-bit counter is FFFFH. If the TTnCNT register is read at this time, 0000H is read.

Reset sets the TTnCE bit to 0.

(2) CCR0 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TTnCCR0 register is used as a compare register, the value written to the TTnCCR0 register is transferred to the CCR0 buffer register. When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTTEQCn0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is set to 0000H after reset, and the TTnCCR0 register is set to 0000H.

(3) CCR1 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TTnCCR1 register is used as a compare register, the value written to the TTnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTTEQCn1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is set to 0000H after reset, and the TTnCCR1 register is set to 0000H.

(4) Edge detector

This circuit detects the valid edges input to the TIT00 (V850E/IG3 only), TIT01 (V850E/IG3 only), TIT10, TIT11, EVTT0 (V850E/IG3 only), EVTT1, TENC00 (V850E/IG3 only), TENC01 (V850E/IG3 only), TENC10, TENC11, TECR0 (V850E/IG3 only), and TECR1 pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TTmIOC1, TTmIOC2, and TTmIOC3 registers.

(5) Output controller

This circuit controls the output of the TOT00 (V850E/IG3 only), TOT01 (V850E/IG3 only), TOT10, and TOT11 pins. The output controller is controlled by the TTmIOC0 registers.

(6) Selector

This selector selects the count clock for the 16-bit counter. Eight types of internal clocks or an external event can be selected as the count clock.

(7) Counter control

The count operation is controlled by the timer mode selected by the TTnCTL1 register.

8.4 Registers

(1) TMTn control register 0 (TTnCTL0)

The TTnCTL0 register is an 8-bit register that controls the operation of TMTn.

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

Reset sets this register to 00H.

The same value can always be written to the TTnCTL0 register by software.

After reset: 00H R/W Address: TT0CTL0 FFFFF580H, TT1CTL0 FFFFF5C0H

	<7>	6	5	4	3	2	1	0
TTnCTL0	TTnCE	0	0	0	0	TTnCKS2	TTnCKS1	TTnCKS0

(n = 0, 1)

TTnCE	TMTn operation control
0	TMTn operation disabled (TMTn reset asynchronously[Note])
1	TMTn operation enabled. TMTn operation start

TTnCKS2	TTnCKS1	TTnCKS0	Internal count clock selection
0	0	0	$f_{XX}/2$
0	0	1	$f_{XX}/4$
0	1	0	$f_{XX}/8$
0	1	1	$f_{XX}/16$
1	0	0	$f_{XX}/64$
1	0	1	$f_{XX}/256$
1	1	0	$f_{XX}/1024$
1	1	1	$f_{XX}/2048$

Note TTnOPT0.TTnOVF bit and 16-bit counter are reset simultaneously. Moreover, timer outputs (TOT00 (V850E/IG3 only), TOT01 (V850E/IG3 only), TOT10, and TOT11 pins) are reset to the TTmIOC0 register set status at the same time as the 16-bit counter (V850E/IF3: m = 1, V850E/IG3: m = 0, 1).

Cautions 1. Set the TTnCKS2 to TTnCKS0 bits when the TTnCE bit = 0.
When the value of the TTnCE bit is changed from 0 to 1, the TTnCKS2 to TTnCKS0 bits can be set simultaneously.
2. Be sure to set bits 3 to 6 to "0".

Remark f_{XX}: Peripheral clock

(2) TMTn control register 1 (TTnCTL1)

The TTnCTL1 register is an 8-bit register that controls the TMTn operation.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: TT0CTL1 FFFFF581H, TT1CTL1 FFFFF5C1H

	7	6	5	4	3	2	1	0
TTnCTL1	0	TTmEST[Note]	TTmEEE[Note]	0	TTnMD3	TTnMD2	TTnMD1	TTnMD0

V850E/IF3 (n = 0, 1, m = 1)

V850E/IG3 (n = 0, 1, m = 0, 1)

TTmEST[Note]	Software trigger control
0	–
1	Generate a valid signal for external trigger input. • In one-shot pulse output mode: A one-shot pulse is output with writing 1 to the TTmEST bit as the trigger. • In external trigger pulse output mode: A PWM waveform is output with writing 1 to the TTmEST bit as the trigger.
The read value of the TTmEST bit is always 0.	

TTmEEE[Note]	Count clock selection
0	Disable operation with external event count input (EVTTm pin). (Perform counting with the count clock selected by the TTmCTL0.TTmCKS0 to TTmCTL0.TTmCKS2 bits.)
1	Enable operation with external event count input (EVTTm pin). (Perform counting at the valid edge of the external event count input signal (EVTTm pin).)
The TTmEEE bit selects whether counting is performed with the internal count clock or the valid edge of the external event count input.	

TTnMD3	TTnMD2	TTnMD1	TTnMD0	Timer mode selection
0	0	0	0	Interval timer mode
0	0	0	1	External event count mode
0	0	1	0	External trigger pulse output mode
0	0	1	1	One-shot pulse output mode
0	1	0	0	PWM output mode
0	1	0	1	Free-running timer mode
0	1	1	0	Pulse width measurement mode
0	1	1	1	Triangular-wave PWM output mode
1	0	0	0	Encoder compare mode
Other than above				Setting prohibited

Note In the V850E/IF3, this bit can be set only in TMT1. Be sure to set bits 5 and 6 of TMT0 to “0”.

(2/2)

Cautions 1. The TTmEST bit is valid only in the external trigger pulse output mode or one-shot pulse output mode. In any other mode, writing 1 to this bit is ignored.

2. The TTmEEE bit is valid only in the interval timer mode, external trigger pulse output mode, one-shot pulse output mode, PWM output mode, free-running timer mode, pulse width measurement mode, or triangular-wave PWM output mode. In any other mode, writing 1 to this bit is ignored.

3. External event count input (EVTTm) or encoder inputs (TENCm0, TENCm1) is selected in the external event count mode or encoder compare mode regardless of the value of the TTmEEE bit.

4. Set the TTmEEE and TTnMD3 to TTnMD0 bits when the TTnCTL0.TTnCE bit = 0. (The same value can be written when the TTnCE bit = 1.) The operation is not guaranteed when rewriting is performed with the TTnCE bit = 1. If rewriting was mistakenly performed, clear the TTnCE bit to 0 and then set the bits again.

5. Be sure to set bits 4 and 7 to "0".

(3) TMTm control register 2 (TTmCTL2)

The TTmCTL2 register is an 8-bit register that controls the encoder count function operation.

The TTmCTL2 register is valid only in the encoder compare mode.

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

Reset sets this register to 00H.

Caution For details of each bit of the TTmCTL2 register, see 8.6.9 (5) Controlling bits of TTmCTL2 register.

(1/2)

After reset: 00H R/W Address: TT0CTL2 FFFFF582H[Note], TT1CTL2 FFFFF5C2H

	7	6	5	4	3	2	1	0
TTmCTL2	TTmECC	0	0	TTmLDE	TTmECM1	TTmECM0	TTmUDS1	TTmUDS0

(V850E/IF3 m = 1)

(V850E/IG3 m = 0, 1)

TTmECC	Encoder counter control
0	Normal operation
1	Holds count value of 16-bit counter when TTmCTL0.TTmCE bit = 0.

TTmLDE	Transfer setting to 16-bit counter
0	Disables transfer of set value of TTmCCR0 to 16-bit counter in case of underflow.
1	Enables transfer of set value of TTmCCR0 to 16-bit counter in case of underflow.

TTmECM1	Control of encoder clear operation 1
0	The 16-bit counter is not cleared to 0000H when its count value matches value of CCR1 register.
1	The 16-bit counter is cleared to 0000H when its count value matches value of CCR1 register.

TTmECM0	Control of encoder clear operation 0
0	The 16-bit counter is not cleared to 0000H when its count value matches value of CCR0 register.
1	The 16-bit counter is cleared to 0000H when its count value matches value of CCR0 register.

Note V850E/IG3 only

(2/2)

TTmUDS1	TTmUDS0	Up/down count selection
0	0	When valid edge of TENCm0 input is detected Counts down when TENCm1 = high level. Counts up when TENCm1 = low level.
0	1	Counts up when valid edge of TENCm0 input is detected. Counts down when valid edge of TENCm1 input is detected.
1	0	Counts down when rising edge of TENCm0 input is detected. Counts up when falling edge of TENCm0 input is detected. However, count operation is performed only when TENCm1 = low level.
1	1	Both rising and falling edges of TENCm0 and TENCm1 are detected. Count operation is automatically identified by combination of edge detection and level detection.

Cautions 1. The TTmECC bit is valid only in the encoder compare mode. In any other mode, writing "1" to this bit is ignored.

If the TTmCTL0.TTmCE bit is cleared to 0 while the TTmECC bit = 1, the values of the timer/counter and capture registers (TTmCCR0 and TTmCCR1), and the TTmOPT1, TTmEUF, TTmEOF, and TTmESF flags are retained.

If the TTmCE bit is set from 0 to 1 when the TTmECC bit = 1, the value of the TTmTCW register is not transferred to the 16-bit counter.

2. The TTmLDE bit is valid only when the TTmECM1 and TTmECM0 bits = 00, 01. Writing "1" to this bit is ignored when the TTmECM1 and TTmECM0 bits = 10, 11.

3. The edge detection of the TENCm0 and TENCm1 inputs specified by the TTmIOC3.TTmEIS1 and TTmIOC3.TTmEIS0 bits is invalid and fixed to both the rising and falling edges when the TTmUDS1 and TTmUDS0 bits = 10, 11.

4. Set the TTmLDE, TTmECM1, TTmECM0, TTmUDS1, and TTmUDS0 bits when the TTmCTL0.TTmCE bit = 0 (the same value can be written to these bits when the TTmCE bit = 1). If the value of these bits is changed when the TTmCE bit = 1, the operation cannot be guaranteed. If it is changed by mistake, clear the TTmCE bit and then set the correct value.

5. Be sure to set bits 5 and 6 to "0".

(4) TMTm I/O control register 0 (TTmIOC0)

The TTmIOC0 register is an 8-bit register that controls the timer output (TOTm0, TOTm1 pins).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TT0IOC0 FFFFF583H[Note 1], TT1IOC0 FFFFF5C3H

TTmIOC0 (V850E/IF3 m = 1, V850E/IG3 m = 0, 1)

7	6	5	4	3	<2>	1	<0>
0	0	0	0	TTmOL1	TTmOE1	TTmOL0	TTmOE0

TTmOL1	TOTm1 pin output level setting[Note 2]
0	TOTm1 pin starts output at high level.
1	TOTm1 pin starts output at low level.

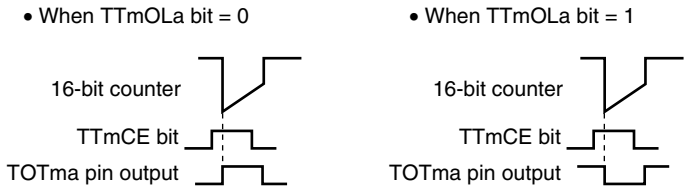
TTmOE1	TOTm1 pin output setting
0	Timer output prohibited • Low level is output from the TOTm1 pin when the TTmOL1 bit = 0. • High level is output from the TOTm1 pin when the TTmOL1 bit = 1.
1	Timer output enabled (A pulse is output from the TOTm1 pin.)

TTmOL0	TOTm0 pin output level setting[Note 2]
0	TOTm0 pin starts output at high level.
1	TOTm0 pin starts output at low level.

TTmOE0	TOTm0 pin output setting
0	Timer output prohibited • Low level is output from the TOTm0 pin when the TTmOL0 bit = 0. • High level is output from the TOTm0 pin when the TTmOL0 bit = 1.
1	Timer output enabled (A pulse is output from the TOTm0 pin.)

Notes 1. V850E/IG3 only

2. The output level of the timer output pins (TOTm0 and TOTm1) specified by the TTmOLa bit is shown below (a = 0, 1).



Cautions 1. If the setting of the TTmIOC0 register is changed when TOTm0 and TOTm1 outputs are set for the port mode, the output of the pins change. Set the port in the input mode and make the port go into a high-impedance state, noting changes in the pin status.

2. Rewrite the TTmOL1, TTmOE1, TTmOL0, and TTmOE0 bits when the TTmCTL0.TTmCE bit = 0. (The same value can be written when the TTmCE bit = 1.) If rewriting was mistakenly performed, clear the TTmCE bit to 0 and then set the bits again.

3. Even if the TTmOL0 or TTmOL1 bit is manipulated when the TTmCE, TTmOE0, and TTmOE1 bits are 0, the output level of the TOTm0 and TOTm1 pins changes.

(5) TMTm I/O control register 1 (TTmIOC1)

The TTmIOC1 register is an 8-bit register that controls the valid edge for the capture trigger input signals (TITm0, TITm1 pins).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TT0IOC1 FFFFF584H[Note], TT1IOC1 FFFFF5C4H

	7	6	5	4	3	2	1	0
TTmIOC1	0	0	0	0	TTmIS3	TTmIS2	TTmIS1	TTmIS0

(V850E/IF3 m = 1)

(V850E/IG3 m = 0, 1)

TTmIS3	TTmIS2	Capture trigger input signal (TITm1 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TTmIS1	TTmIS0	Capture trigger input signal (TITm0 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Note V850E/IG3 only

Cautions 1. Rewrite the TTmIS3 to TTmIS0 bits when the TTmCTL0.TTmCE bit = 0. (The same value can be written when the TTmCE bit = 1.) If rewriting was mistakenly performed, clear the TTmCE bit to 0 and then set the bits again.

2. The TTmIS3 and TTmIS2 bits are valid only in the free-running timer mode (only when the TTmOPT0.TTmCCS1 bit = 1) and the pulse width measurement mode. In all other modes, a capture operation is not possible.

The TTmIS1 and TTmIS0 bits are valid only in the free-running timer mode (only when the TTmOPT0. TTmCCS0 bit = 1) and the pulse width measurement mode. In all other modes, a capture operation is not possible.

(6) TMTm I/O control register 2 (TTmIOC2)

The TTmIOC2 register is an 8-bit register that controls the valid edge for the external event count input signal (EVTTm pin) and external trigger input signal (EVTTm pin).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TT0IOC2 FFFFF585H[Note], TT1IOC2 FFFFF5C5H

	7	6	5	4	3	2	1	0
TTmIOC2	0	0	0	0	TTmEES1	TTmEES0	TTmETS1	TTmETS0

V850E/IF3 m = 1

V850E/IG3 m = 0, 1

TTmEES1	TTmEES0	External event count input signal (EVTTm pin) valid edge setting
0	0	No edge detection (external event count invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TTmETS1	TTmETS0	External trigger input signal (EVTTm pin) valid edge setting
0	0	No edge detection (external trigger invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Note V850E/IG3 only

Cautions 1. Rewrite the TTmEES1, TTmEES0, TTmETS1, and TTmETS0 bits when the TTmCTL0.TTmCE bit = 0. (The same value can be written when the TTmCE bit = 1.) If rewriting was mistakenly performed, clear the TTmCE bit to 0 and then set the bits again.

2. The TTmEES1 and TTmEES0 bits are valid only when the TTmCTL1.TTmEEE bit = 1 or when the external event count mode (the TTmCTL1.TTmMD3 to TTmCTL1.TTmMD0 bits = 0001) has been set.

3. The TTmETS1 and TTmETS0 bits are valid only in the external trigger pulse mode or one-shot pulse output mode.

(7) TMTm I/O control register 3 (TTmIOC3)

The TTmIOC3 register is an 8-bit register that controls the encoder clear function operation.

The TTmIOC3 register is valid only in the encoder compare mode.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: TT0IOC3 FFFFF586H[Note], TT1IOC3 FFFFF5C6H

	7	6	5	4	3	2	1	0
TTmIOC3	TTmSCE	TTmZCL	TTmBCL	TTmACL	TTmECS1	TTmECS0	TTmEIS1	TTmEIS0

(V850E/IF3 m = 1)

(V850E/IG3 m = 0, 1)

TTmSCE	Encoder clear selection
0	Clears 16-bit counter on detection of edge of encoder clear signal (TECRm pin).
1	Clears 16-bit counter on detection of clear level condition of the TENCm0, TENCm1, and TECRm pins.

- Clears 16-bit counter to 0000H when valid edge of TECRm pin specified by the TTmECS1 and TTmECS0 bits is detected when the TTmSCE bit = 0.
- Clears 16-bit counter to 0000H when clear level conditions of the TTmZCL, TTmBCL, and TTmACL bits match input levels of the TECRm, TENCm1, and TENCm0 pins when TTmSCE bit = 1.
- Setting of the TTmZCL, TTmBCL, and TTmACL bits is valid and that of the TTmECS1 and TTmECS0 bits is invalid when the TTmSCE bit = 1.
  Encoder clear interrupt request signal (INTTIECn) is not generated.
- Setting of the TTmZCL, TTmBCL, and TTmACL bits is invalid and setting of the TTmECS1 and TTmECS0 bits is valid when the TTmSCE bit = 0.
  The INTTIECn signal is generated when valid edge specified by the TTmECS1 and TTmECS0 bits is detected.
- Be sure to set the TTmCTL2.TTmUDS1 and TTmCTL2.TTmUDS0 bits to 10 or 11 when the TTmSCE bit = 1.
  Operation is not guaranteed if the TTmUDS1 and TTmUDS0 bits = 00 or 01 and the TTmSCE bit = 1.

TTmZCL	Clear level selection of encoder clear signal (TECRm pin)
0	Clears low level of the TECRm pin.
1	Clears high level of the TECRm pin.
Setting of the TTmZCL bit is valid only when the TTmSCE bit = 1.	

TTmBCL	Clear level selection of encoder input signal (TENCm1 pin)
0	Clears low level of the TENCm1 pin.
1	Clears high level of the TENCm1 pin.
Setting of the TTmBCL bit is valid only when the TTmSCE bit = 1.	

TTmACL	Clear level selection of encoder input signal (TENCm0 pin)
0	Clears low level of the TENCm0 pin.
1	Clears high level of the TENCm0 pin.
Setting of the TTmACL bit is valid only when the TTmSCE bit = 1.	

Note V850E/IG3 only

(2/2)

TTmECS1	TTmECS0	Valid edge setting of encoder clear signal (TECRm pin)
0	0	Detects no edge (clearing encoder is invalid).
0	1	Detects rising edge.
1	0	Detects falling edge.
1	1	Detects both edges.

TTmEIS1	TTmEIS0	Valid edge setting of encoder input signals (TENCm0, TENCm1 pins)
0	0	Detects no edge (inputting encoder is invalid).
0	1	Detects rising edge.
1	0	Detects falling edge.
1	1	Detects both edges.

Cautions 1. Rewrite the TTmSCE, TTmZCL, TTmBCL, TTmACL, TTmECS1, TTmECS0, TTmEIS1, and TTmEIS0 bits when the TTmCTL0.TTmCE bit = 0. (The same value can be written to these bits when the TTmCE bit = 1.) If rewriting was mistakenly performed, clear the TTmCE bit to 0 and then set these bits again.

2. The TTmECS1 and TTmECS0 bits are valid only when the TTmSCE bit = 0 and the encoder compare mode is set.

3. The TTmEIS1 and TTmEIS0 bits are valid only when the TTmCTL2.TTmUDS1 and TTmCTL2.TTmUDS0 bits = 00 or 01.

(8) TMTn option register 0 (TTnOPT0)

The TTnOPT0 register is an 8-bit register that sets the capture/compare operation and detects overflow.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TT0OPT0 FFFFF587H, TT1OPT0 FFFFF5C7H

	7	6	5	4	3	2	1	<0>
TTnOPT0	0	0	TTmCCS1[Note]	TTmCCS0[Note]	0	0	0	TTnOVF

V850E/IF3 (n = 0, 1, m = 1)

V850E/IG3 (n = 0, 1, m = 0, 1)

TTmCCS1[Note]	TTmCCR1 register capture/compare selection
0	Compare register selected
1	Capture register selected (cleared by the TTmCTL0.TTmCE bit = 0)
The TTmCCS1 bit setting is valid only in the free-running timer mode.	

TTmCCS0[Note]	TTmCCR0 register capture/compare selection
0	Compare register selected
1	Capture register selected (cleared by the TTmCTL0.TTmCE bit = 0)
The TTmCCS0 bit setting is valid only in the free-running timer mode.	

TTnOVF	TMTn overflow detection flag
Set (1)	Overflow occurred
Reset (0)	0 written to TTnOVF bit or TTnCTL0.TTnCE bit = 0

- The TTnOVF bit is set to 1 when the 16-bit counter value overflows from FFFFH to 0000H in the free-running timer mode or the pulse width measurement mode.
- An overflow interrupt request signal (INTTTIOVn) is generated at the same time that the TTnOVF bit is set to 1. The INTTTIOVn signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode.
- The TTnOVF bit is not cleared to 0 even when the TTnOVF bit or the TTnOPT0 register are read when the TTnOVF bit = 1.
- Before clearing the TTnOVF bit to 0 after generation of the INTTTIOVn signal, be sure to confirm (by reading) that the TTnOVF bit is set to 1.
- The TTnOVF bit can be both read and written, but the TTnOVF bit cannot be set to 1 by software. Writing 1 has no effect on the operation of TMTn.

Note In the V850E/IF3, this bit can be set only in TMT1. Be sure to set bits 4 and 5 of TMT0 to “0”.

Cautions 1. Rewrite the TTmCCS1 and TTmCCS0 bits when the TTmCE bit = 0. (The same value can be written when the TTmCE bit = 1.) If rewriting was mistakenly performed, clear the TTmCE bit to 0 and then set these bits again.

2. Be sure to set bits 1 to 3, 6, and 7 to “0”.

(9) TMTm option register 1 (TTmOPT1)

The TTmOPT1 register is an 8-bit register that detects the overflow, underflow, and count-up/down operation of the encoder count function.

The TTmOPT1 register is valid only in the encoder compare mode.

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

Reset sets this register to 00H.

This register can be rewritten even when the TTmCTL0.TTmCE bit = 1.

(1/2)

After reset: 00H R/W Address: TT0OPT1 FFFFF588H[Note], TT1OPT1 FFFFF5C8H

	7	6	5	4	3	<2>	<1>	<0>
TTmOPT1	0	0	0	0	0	TTmEUF	TTmEOF	TTmESF

V850E/IF3 m = 1

TTmEUF	TMTm underflow detection flag
Set (1)	Underflow occurs.
Reset (0)	Cleared by writing to TTmEUF bit or when TTmCTL0.TTmCE bit = 0

V850E/IG3 m = 0, 1

- The TTmEUF bit is set to 1 when 16-bit counter underflows from 0000H to FFFFH in encoder compare mode.
- When the TTmCTL2.TTmLDE bit = 1, TTmEUF bit is set to 1 when value of 16-bit counter is changed from 0000H to set value of the TTmCCR0 register.
- Overflow interrupt request signal (INTTTIOVm) is generated as soon as the TTmEUF bit is set to 1.
- The TTmEUF bit is not cleared to 0 even if the TTmEUF bit or TTmOPT1 register is read when the TTmEUF bit = 1.
- Status of the TTmEUF bit is retained even if the TTmCTL0.TTmCE bit is cleared to 0 when the TTmCTL2.TTmECC bit = 1.
- Before clearing the TTmEUF bit to 0 after the INTTTIOVm signal is generated, be sure to confirm (read) that the TTmEUF bit is set to 1.
- The TTmEUF bit can be read or written, but it cannot be set to 1 by software. Setting this bit to 1 does not affect operation of TMTm.

Note V850E/IG3 only

(2/2)

TTmEOF	Overflow detection flag for TMTm encoder function
Set (1)	Overflow occurs.
Reset (0)	Cleared by writing 0 to the TTmEOF bit or when the TTmCTL0.TTmCE bit = 0
• The TTmEOF bit is set to 1 when 16-bit counter overflows from FFFFH to 0000H in encoder compare mode. • As soon as the TTmEOF bit has been set to 1, an overflow interrupt request signal (INTTTIOVm) is generated. At this time, the TTmOPT0.TTmOVF bit is not set to 1. • The TTmEOF bit is not cleared to 0 even if the TTmEOF bit or TTmOPT1 register is read when the TTmEOF bit = 1. • Status of the TTmEOF bit is retained even if the TTmCTL0.TTmCE bit is cleared to 0 when the TTmCTL2.TTmECC bit = 1. • Before clearing the TTmEOF bit to 0 after the INTTTIOVm signal is generated, be sure to confirm (read) that the TTmEOF bit is set to 1. • The TTmEOF bit can be read or written, but it cannot be set to 1 by software. Writing 1 to this bit does not affect operation of TMTm.	

TTmESF	TMTm count-up/-down operation status detection flag
0	TMTm is counting up.
1	TMTm is counting down.
• This bit is cleared to 0 if the TTmCTL0.TTmCE bit = 0 when the TTmCTL2.TTmECC bit = 0. • Status of the TTmESF bit is retained even if the TTmCE bit = 0 when the TTmECC bit = 1.	

Caution Be sure to set bits 3 to 7 to “0”.

(10) TMTm capture input select register (TTISLm)

The TTISLm register is used to select which of TITm0 or TITm1 pin is used to input a capture trigger input signal when the TTmCCR0 register is used as a capture register.

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

Reset makes this register undefined.

After reset: Undefined R/W Address: TTISL0 FFFFF5A4H[Note], TTISL1 FFFFF5A6H

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

V850E/IF3 m = 1

TTISLm	Capture trigger input signal selection of TTmCCR0 register
0	TITm0 input
1	TITm1 input

V850E/IG3 m = 0, 1

Note V850E/IG3 only

(11) TMTn capture/compare register 0 (TTnCCR0)

The TTmCCR0 register is a 16-bit register that can be used as a capture register or compare register depending on the mode. The TT0CCR0 register of the V850E/IF3 is a 16-bit registers that can only be used as a compare register.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TTmOPT0.TTmCCS0 bit. In the pulse width measurement mode, the TTmCCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TTnCCR0 register can be read or written during operation.

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

Reset sets this register to 0000H.

Remark V850E/IF3: n = 0, 1, m = 1
V850E/IG3: n = 0, 1, m = 0, 1

After reset: 0000H R/W Address: TT0CCR0 FFFFF58AH, TT1CCR0 FFFFF5CAH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTnCCR0 (n = 0, 1)																

(a) Function as compare register

The TTnCCR0 register can be rewritten even when the TTnCTL0.TTnCE bit = 1.

The set value of the TTnCCR0 register is transferred to the CCR0 buffer register. When the value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTTEQCn0) is generated. If TOTm0 pin output is enabled at this time, the output of the TOTm0 pin is inverted.

When the TTnCCR0 register is used as a cycle register in the interval timer mode, or when the TTmCCR0 register is used as a cycle register in the external event count mode, external trigger pulse output mode, one-shot pulse output mode, PWM output mode, triangular-wave PWM output mode, or encoder compare mode, the value of the 16-bit counter is cleared (0000H) if its count value matches the value of the CCR0 buffer register.

The compare register is not cleared by setting the TTnCTL0.TTnCE bit to 0.

(b) Function as capture register

When the TTmCCR0 register is used as a capture register in the free-running timer mode (when the TTmCCR0 register is used as a capture register), the count value of the 16-bit counter is stored in the TTmCCR0 register if the valid edge of the capture trigger input pin (TITm0 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TTmCCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TITm0 pin) is detected.

Even if the capture operation and reading the TTmCCR0 register conflict, the correct value of the TTmCCR0 register can be read.

The capture register is cleared by setting the TTmCTL0.TTmCE bit to 0.

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 8-4. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	TTnCCR0 Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter[Note 1]	Compare register	Anytime write
External trigger pulse output[Note 1]	Compare register	Batch write[Note 2]
One-shot pulse output[Note 1]	Compare register	Anytime write
PWM output[Note 1]	Compare register	Batch write[Note 2]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None
Triangular-wave WPM output[Note 1]	Compare register	Batch write[Note 2]
Encoder compare[Note 1]	Compare register	Anytime write

Notes 1. In the V850E/IF3, this mode can be set only in TMT1.

2. Writing to the TTnCCR1 register is the trigger.

Remark For anytime write and batch write, see **8.6 (2) Anytime write and batch write**.

(12) TMTn capture/compare register 1 (TTnCCR1)

The TTmCCR1 register is a 16-bit register that can be used as a capture register or compare register depending on the mode. The TT0CCR1 register of the V850E/IF3 is a 16-bit registers that can only be used as a compare register.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TTmOPT0.TTmCCS1 bit. In the pulse width measurement mode, the TTmCCR1 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TTnCCR1 register can be read or written during operation.

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

Reset sets this register to 0000H.

Remark V850E/IF3: n = 0, 1, m = 1
V850E/IG3: n = 0, 1, m = 0, 1

After reset: 0000H R/W Address: TT0CCR1 FFFFF58CH, TT1CCR1 FFFFF5CCH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTnCCR1 (n = 0, 1)																

(a) Function as compare register

The TTnCCR1 register can be rewritten even when the TTnCTL0.TTnCE bit = 1.

The set value of the TTnCCR1 register is transferred to the CCR1 buffer register. When the value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTTEQCn1) is generated. If TOTm1 pin output is enabled at this time, the output of the TOTm1 pin is inverted.

The compare register is not cleared by setting the TTnCTL0.TTnCE bit to 0.

(b) Function as capture register

When the TTnCCR1 register is used as a capture register in the free-running timer mode (when the TTmCCR1 register is used as a capture register), the count value of the 16-bit counter is stored in the TTmCCR1 register if the valid edge of the capture trigger input pin (TITm1 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TTmCCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TITm1 pin) is detected.

Even if the capture operation and reading the TTmCCR1 register conflict, the correct value of the TTmCCR1 register can be read.

The capture register is cleared by setting the TTmCTL0.TTmCE bit to 0.

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 8-5. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	TTnCCR1 Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter[Note 1]	Compare register	Anytime write
External trigger pulse output[Note 1]	Compare register	Batch write[Note 2]
One-shot pulse output[Note 1]	Compare register	Anytime write
PWM output[Note 1]	Compare register	Batch write[Note 2]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None
Triangular-wave PWM output[Note 1]	Compare register	Batch write[Note 2]
Encoder compare[Note 1]	Compare register	Anytime write

Notes 1. In the V850E/IF3, this mode can be set only in TMT1.

2. Writing to the TTnCCR1 register is the trigger.

Remark For anytime write and batch write, see **8.6 (2) Anytime write and batch write**.

(13) TMTm counter write register (TTmTCW)

The TTmTCW register is used to set the initial value of the 16-bit counter.

The TTmTCW register is valid only in the encoder compare mode.

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

Rewrite the TTmTCW register when the TTmCTL0.TTmCE bit = 0.

The value of the TTmTCW register is transferred to the 16-bit counter when the TTmCE bit is set (1).

Reset sets this register to 0000H.

After reset: 0000H R/W Address: TT0TCW FFFFF590H[Note], TT1TCW FFFFF5D0H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTmTCW																

(V850E/IF3 m = 1)

(V850E/IG3 m = 0, 1)

Note V850E/IG3 only

(14) TMTn counter read buffer register (TTnCNT)

The TTnCNT register is a read buffer register that can read the count value of the 16-bit counter.

If this register is read when the TTnCTL0.TTnCE bit = 1, the count value of the 16-bit timer can be read.

This register is read-only, in 16-bit units.

The value of the TTmCNT register is set to 0000H when the TTmCTL2.TTmECC and TTmCE bits = 0. If the TTmCNT register is read at this time, the value of the 16-bit counter (FFFFH) is not read, but 0000H is read.

The TTmCNT register is not set to 0000H but the previous value is read when the TTmECC bit = 1 and TTmCE bit = 0.

The TTmECC and TTmCE bits are set to 0 after reset, and the value of the TTmCNT register is set to 0000H.

After reset: 0000H R Address: TT0CNT FFFFF58EH, TT1CNT FFFFF5CEH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTnCNT																

(n = 0, 1)

8.5 Timer Output Operations

The following table shows the operations and output levels of the TOTm0 and TOTm1 pins.

Table 8-6. Timer Output Control in Each Mode

Operation Mode	TOTm1 Pin	TOTm0 Pin
Interval timer mode	PWM output	
External event count mode[Note]	None	
External trigger pulse output mode[Note]	External trigger pulse output	PWM output
One-shot pulse output mode[Note]	One-shot pulse output	
PWM output mode[Note]	PWM output	
Free-running timer mode	PWM output (only when compare function is used)	
Pulse width measurement mode[Note]	None	
Triangular-wave PWM output mode[Note]	Triangular-wave PWM output	
Encoder compare mode[Note]	None	

Note In the V850E/IF3, this mode can be set only in TMT1.

Remark V850E/IF3: m = 1, a = 0, 1
V850E/IG3: m = 0, 1, a = 0, 1

Table 8-7. Truth Table of TOTm0 and TOTm1 Pins Under Control of Timer Output Control Bits

TTmIOC0.TTmOLa Bit	TTmIOC0.TTmOEa Bit	TTmCTL0.TTmCE Bit	Level of TOTma Pin
0	0	×	Low-level output
	1	0	Low-level output
		1	Low level immediately before counting, high level after counting is started
1	0	×	High-level output
	1	0	High-level output
		1	High level immediately before counting, low level after counting is started

Remark V850E/IF3: m = 1, a = 0, 1
V850E/IG3: m = 0, 1, a = 0, 1

8.6 Operation

The functions of TMTn that can be implemented differ from one channel to another. The functions of each channel are shown below (n = 0, 1).

Table 8-8. TMTm Specifications in Each Mode

Operation	TTmCTL1.TTmEST Bit (Software Trigger Bit)	EVTTm Pin (External Trigger Input)	Capture/Compare Register Setting	Compare Register Write Method
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode	Invalid	Invalid	Compare only	Anytime write
External trigger pulse output mode	Valid	Valid	Compare only	Batch write
One-shot pulse output mode	Valid	Valid	Compare only	Anytime write
PWM output mode	Invalid	Invalid	Compare only	Batch write
Free-running timer mode	Invalid	Invalid	Switchable	Anytime write
Pulse width measurement mode	Invalid	Invalid	Capture only	Not applicable
Triangular-wave PWM output mode	Invalid	Invalid	Compare only	Batch write
Encoder compare mode	Invalid	Invalid	Compare only	Anytime write

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Table 8-9. TMT0 Specifications in Each Mode of V850E/IF3

Operation	Software Trigger Bit	External Trigger Input	Capture/Compare Register Setting	Compare Register Write Method
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode	None			
External trigger pulse output mode	None			
One-shot pulse output mode	None			
PWM output mode	None			
Free-running timer mode	Invalid	Invalid	Compare only	Anytime write
Pulse width measurement mode	None			
Triangular-wave PWM output mode	None			
Encoder compare mode	None			

Remark TMT0 of the V850E/IF3 does not have timer input pins (TIT00, TIT01, TECR0, TENC00, TENC01, EVTT0) and timer output pins (TOT00, TOT01). It has interrupt request signals (INTTTEQC00, INTTTEQC01) indicating a match between the value of the 16-bit counter and the values of the TT0CCR0 and TT0CCR1 registers.

(1) Counter basic operation

This section explains the basic operation of the 16-bit counter. For details, refer to the description of the operation in each mode.

Remark V850E/IF3: n = 0, 1, m = 1
V850E/IG3: n = 0, 1, m = 0, 1

(a) Counter start operation

- **Encoder compare mode**

  The count operation is controlled by TENCm0 and TENCm1 phases.

  When the 16-bit counter initial setting is performed by transferring the set value of the TTmTCW register to the 16-bit counter and the count operation is started. (When the TTmCTL2.TTmECC bit = 0, the TTmTCW register set value is transferred to the 16-bit counter at the timing when the TTmCTL0.TTmCE bit changes from 0 to 1.)

- **Triangular-wave PWM mode**

  The 16-bit counter starts counting from the initial value FFFFH.

  It counts up FFFFH, 0000H, 0001H, 0002H, 0003H, and so on.

  Following count up operation, the counter counts down upon a match between the 16-bit count value and the CCR0 buffer register.

- **Mode other than above**

  The 16-bit counter starts counting from the initial value FFFFH.

  It counts up FFFFH, 0000H, 0001H, 0002H, 0003H, and so on.

(b) Clear operation

The 16-bit counter is cleared to 0000H when its value matches the value of the compare register and cleared, when the value of the 16-bit counter is captured and cleared, when the edge of the encoder clear signal is detected and cleared, and when the clear level condition of the TENCm0, TENCm1, and TECRm pins is detected and cleared. The count operation from FFFFH to 0000H that takes place immediately after the counter has started counting or when the counter overflows is not a clearing operation. Therefore, the INTTTEQCn0 and INTTTEQCn1 interrupt signals are not generated.

(c) Overflow operation

The 16-bit counter overflows when the counter counts up from FFFFH to 0000H in the free-running mode, pulse width measurement mode, and encoder compare mode. If the counter overflows, the TTnOPT0.TTnOVF bit is set to 1 and an interrupt request signal (INTTTIOVn) is generated in the free-running mode and pulse width measurement mode.

If the counter overflows, the TTnOPT1.TTnEOF bit is set to 1 and an interrupt request signal (INTTTIOVn) is generated in the encoder compare mode.

Note that the INTTTIOVn signal is not generated under the following conditions.

- Immediately after a count operation has been started
- If the counter value matches the compare value FFFFH and is cleared
- When FFFFH is captured and cleared to 0000H in the pulse width measurement mode

Caution After the overflow interrupt request signal (INTTTIOVn) has been generated, be sure to check that the overflow flag (TTnOVF, TTmEOF bits) is set to 1.

(d) Count value holding operation

The value of the 16-bit counter is held by the TTmCTL2.TTmECC bit in the encoder compare mode. The value of the 16-bit counter is reset to FFFFH when the TTmECC bit = 0 and TTmCTL0.TTmCE bit = 0. When the TTmCE bit is set to 1 next time, the set value of the TTmTCW register is transferred to the 16-bit counter and the counter continues its count operation.

If the TTmECC bit = 1 and TTmCE bit = 0, the value of the 16-bit counter is held. When the TTmCE bit is set to 1 next time, the counter resumes the count operation from the held value.

(e) Counter read operation during count operation

The value of the 16-bit counter of TMTn can be read by using the TTnCNT register during the count operation. When the TTnCTL0.TTnCE bit = 1, the value of the 16-bit counter can be read by reading the TTnCNT register. If the TTmCNT register is read when the TTmCTL2.TTmECC bit = 0 and TTmCE bit = 0, however, it is 0000H. The held value of the TTmCNT register is read if the register is read when the TTmECC bit = 1 and TTmCE bit = 0.

(f) Underflow operation

The 16-bit counter underflow occurs at the timing when the 16-bit counter value changes from 0000H to FFFFH in the encoder compare mode. When underflow occurs, the TTmOPT1.TTmEUF bit is set to 1 and an interrupt request signal (INTTTIOVm) is generated.

(g) Interrupt operation

TMTn generates the following four types of interrupt request signals.

- INTTTEQCn0 interrupt: This signal functions as a match interrupt request signal of the CCR0 buffer register and as a capture interrupt request signal to the TTnCCR0 register.
- INTTTEQCn1 interrupt: This signal functions as a match interrupt request signal of the CCR1 buffer register and as a capture interrupt request signal to the TTnCCR1 register.
- INTTTIOVn interrupt: This signal functions as an overflow interrupt request signal.
- INTTIECm interrupt: This signal functions as a valid edge detection interrupt request signal of the encoder clear input (TECRm pin).

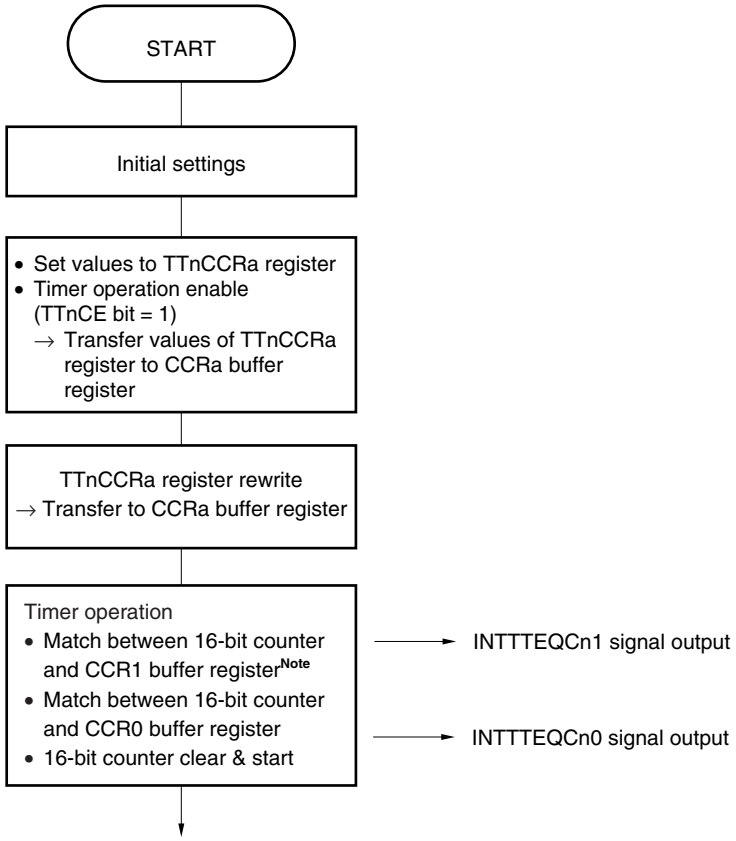
(2) Anytime write and batch write

The TTnCCR0 and TTnCCR1 registers in TMTn can be rewritten during timer operation (TTnCTL0.TTnCE bit = 1), but the write method (anytime write, batch write) of the CCR0 and CCR1 buffer registers differs depending on the mode.

(a) Anytime write

In this mode, data is transferred at any time from the TTnCCR0 and TTnCCR1 registers to the CCR0 and CCR1 buffer registers during timer operation (n = 0, 1).

Figure 8-3. Flowchart of Basic Operation for Anytime Write



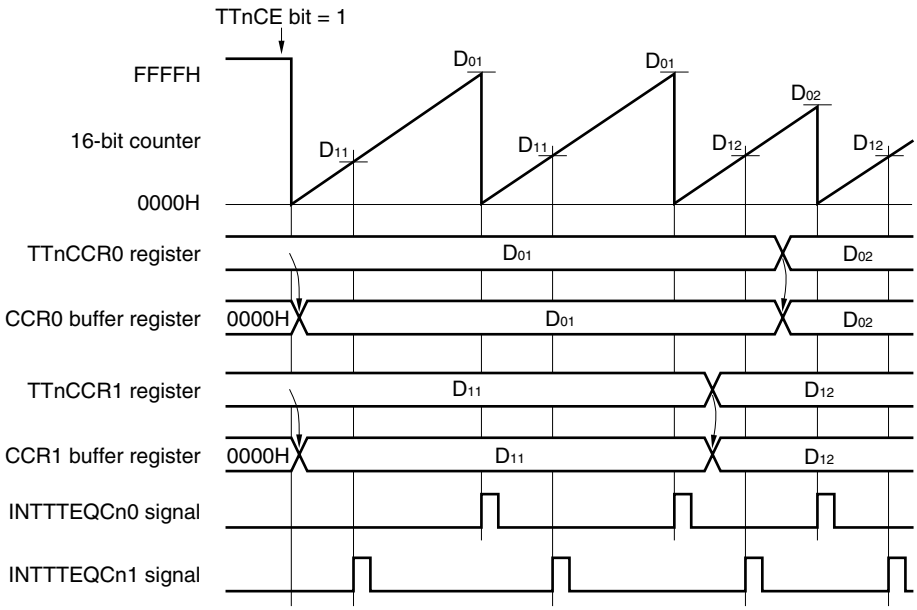
Note The 16-bit counter is not cleared upon a match between the 16-bit counter value and the CCR1 buffer register value. It is cleared upon a match between the 16-bit counter value and the CCR0 buffer register value.

Remarks 1. The above flowchart illustrates an example of the operation in the interval timer mode.

2. n = 0, 1

a = 0, 1

Figure 8-4. Timing of Anytime Write



Remarks 1. D_{01}, D_{02}: Set values of TTnCCR0 register

D_{11}, D_{12}: Set values of TTnCCR1 register

2. The above timing chart illustrates an example of the operation in the interval timer mode.

3. n = 0, 1

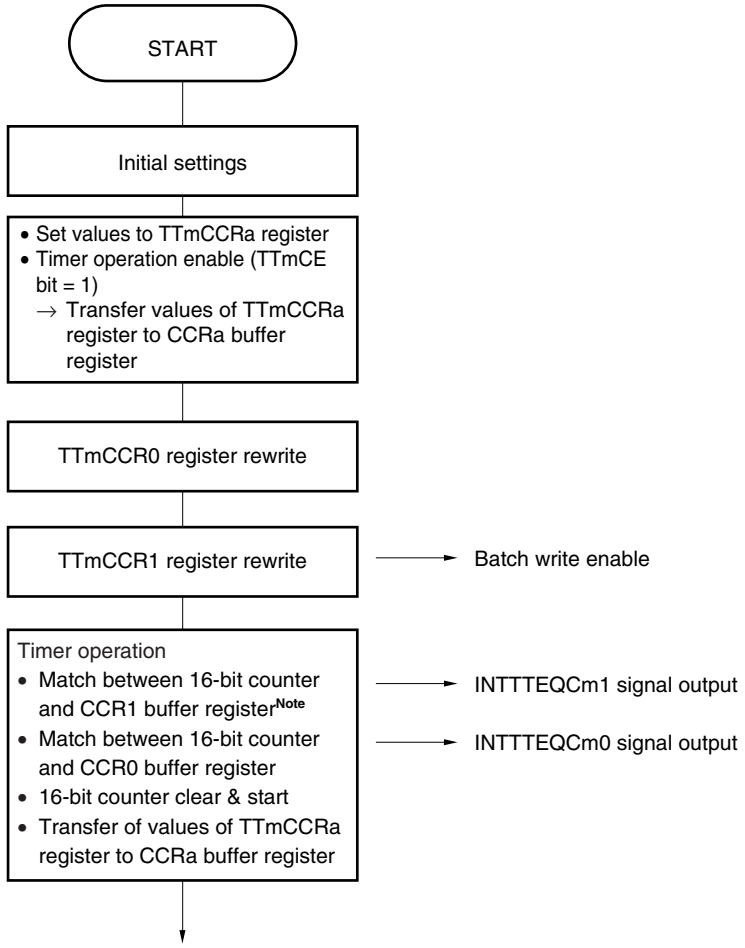
(b) Batch write

In this mode, data is transferred all at once from the TTmCCR0 and TTmCCR1 registers to the CCR0 and CCR1 buffer registers during timer operation. This data is transferred upon a match between the value of the CCR0 buffer register and the value of the 16-bit counter. Transfer is enabled by writing to the TTmCCR1 register. Whether to enable or disable the next transfer timing is controlled by writing or not writing to the TTmCCR1 register.

In order for the set value when the TTmCCR0 and TTmCCR1 registers are rewritten to become the 16-bit counter comparison value (in other words, in order for this value to be transferred to the CCR0 and CCR1 buffer registers), it is necessary to rewrite the TTmCCR0 register and then write to the TTmCCR1 register before the 16-bit counter value and the CCR0 buffer register value match. Therefore, the values of the TTmCCR0 and TTmCCR1 registers are transferred to the CCR0 and CCR1 buffer registers upon a match between the count value of the 16-bit counter and the value of the CCR0 buffer register. Thus even when wishing only to rewrite the value of the TTmCCR0 register, also write the same value (same as preset value of the TTmCCR1 register) to the TTmCCR1 register.

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Figure 8-5. Flowchart of Basic Operation for Batch Write



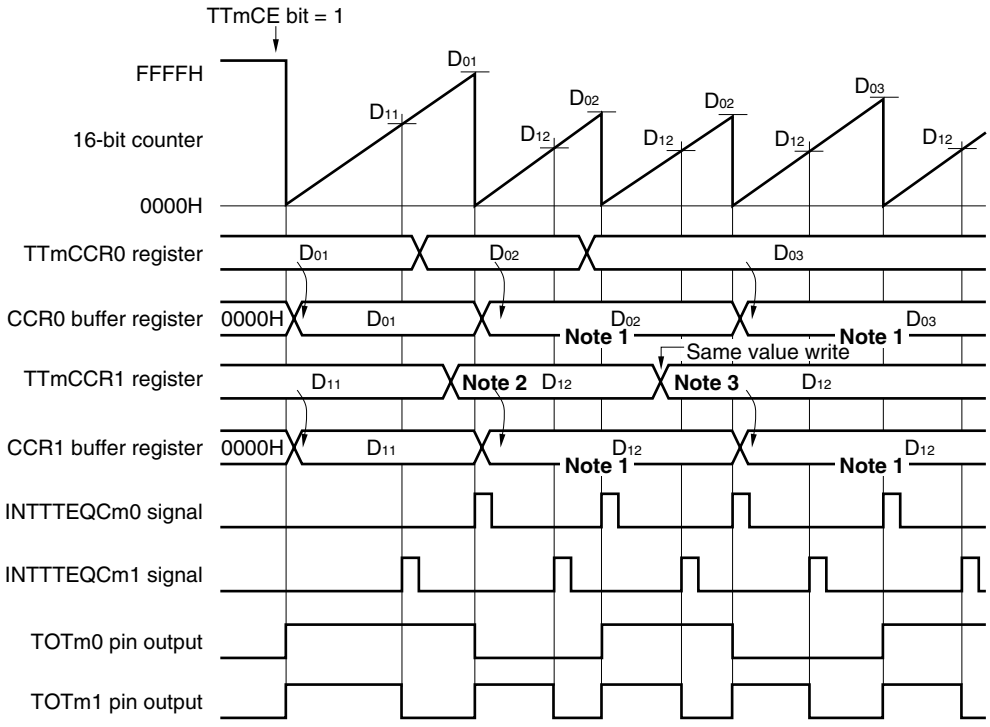
Note The 16-bit counter is not cleared upon a match between the 16-bit counter value and the CCR1 buffer register value. It is cleared upon a match between the 16-bit counter value and the CCR0 buffer register value.

Caution Writing to the TTmCCR1 register includes enabling of batch write. Thus, rewrite the TTmCCR1 register after rewriting the TTmCCR0 register.

Remarks 1. The above flowchart illustrates an example of the operation in the PWM output mode.

2. V850E/IF3: m = 1, a = 0, 1
V850E/IG3: m = 0, 1, a = 0, 1

Figure 8-6. Timing of Batch Write



Notes 1. Because the TTmCCR1 register was not rewritten, D_{03} is not transferred.

2. Because the TTmCCR1 register has been written (D_{12}), data is transferred to the CCR1 buffer register upon a match between the value of the 16-bit counter and the value of the TTmCCR0 register (D_{01}).

3. Because the TTmCCR1 register has been written (D_{12}), data is transferred to the CCR1 buffer register upon a match between the value of the 16-bit counter and the value of the TTmCCR0 register (D_{02}).

Remarks 1. D_{01}, D_{02}, D_{03}: Set values of TTmCCR0 register
D_{11}, D_{12}: Set values of TTmCCR1 register

2. The above timing chart illustrates the operation in the PWM output mode as an example.

3. V850E/IF3: m = 1
V850E/IG3: m = 0, 1

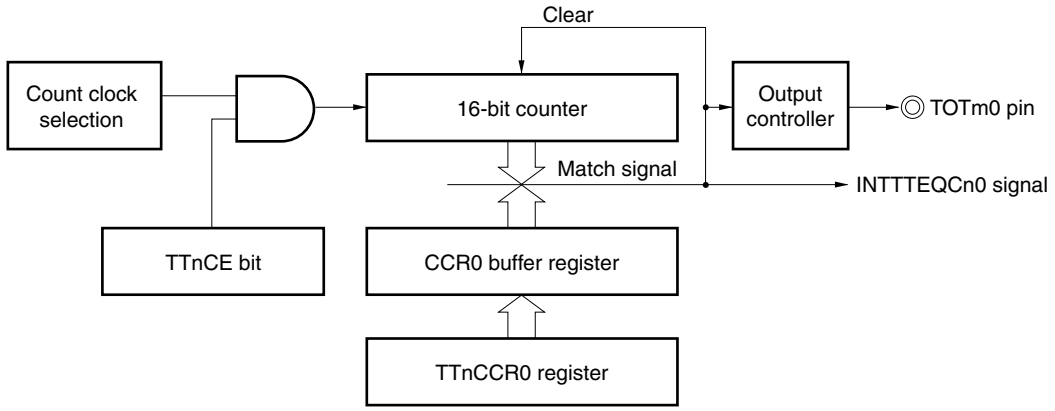
8.6.1 Interval timer mode (TTnMD3 to TTnMD0 bits = 0000)

In the interval timer mode, an interrupt request signal (INTTTEQCn0) is generated at the interval set by the TTnCCR0 register if the TTnCTL0.TTnCE bit is set to 1. A PWM waveform with a duty factor of 50% whose half cycle is equal to the interval can be output from the TOTm0 pin.

The TTnCCR1 register is not used in the interval timer mode. However, the set value of the TTnCCR1 register is transferred to the CCR1 buffer register, and when the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTTEQCn1) is generated. In addition, a PWM waveform with a duty factor of 50%, which is inverted when the INTTTEQCm1 signal is generated, can be output from the TOTm1 pin.

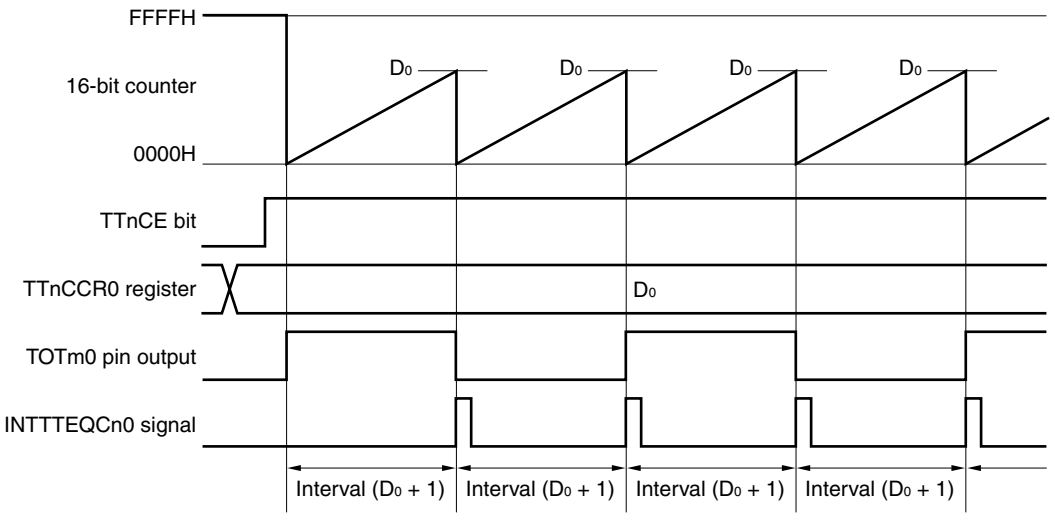
The value of the TTnCCR0 and TTnCCR1 registers can be rewritten even while the timer is operating.

Figure 8-7. Configuration of Interval Timer



Remark V850E/IF3: m = 1, n = 0, 1
V850E/IG3: m = 0, 1, n = 0, 1

Figure 8-8. Basic Timing of Operation in Interval Timer Mode



Remark V850E/IF3: m = 1, n = 0, 1
V850E/IG3: m = 0, 1, n = 0, 1

When the TTnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H in synchronization with the count clock, and the counter starts counting. At this time, the output of the TOTm0 pin is inverted. Additionally, the set value of the TTnCCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, the output of the TOTm0 pin is inverted, and a compare match interrupt request signal (INTTTEQCn0) is generated.

The interval can be calculated by the following expression.

Interval = (Set value of TTnCCR0 register + 1) × Count clock cycle

Remark V850E/IF3: n = 0, 1, m = 1
V850E/IG3: n = 0, 1, m = 0, 1

Figure 8-9. Register Setting for Interval Timer Mode Operation (1/2)

(a) TMTn control register 0 (TTnCTL0)

	TTnCE					TTnCKS2	TTnCKS1	TTnCKS0
TTnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock

0: Stop counting
1: Enable counting

(b) TMTn control register 1 (TTnCTL1)

		TTmEST	TTmEEE			TTnMD3	TTnMD2	TTnMD1	TTnMD0
TTnCTL1	0	0	0	0	0	0	0	0	0

0, 0, 0, 0:
Interval timer mode

(c) TMTm I/O control register 0 (TTmIOC0)

					TTmOL1	TTmOE1	TTmOL0	TTmOE0
TTmIOC0	0	0	0	0	0/1	0/1	0/1	0/1

0: Disable TOTm0 pin output
1: Enable TOTm0 pin output

Setting of TOTm0 pin output level before count operation
0: Low level
1: High level

0: Disable TOTm1 pin output
1: Enable TOTm1 pin output

Setting of TOTm1 pin output level before count operation
0: Low level
1: High level

Figure 8-9. Register Setting for Interval Timer Mode Operation (2/2)

(d) TMTn counter read buffer register (TTnCNT)

By reading the TTnCNT register, the count value of the 16-bit counter can be read.

(e) TMTn capture/compare register 0 (TTnCCR0)

If the TTnCCR0 register is set to D_0, the interval is as follows.

Interval = $(D_0 + 1) \times$ Count clock cycle

(f) TMTn capture/compare register 1 (TTnCCR1)

The TTnCCR1 register is not used in the interval timer mode. However, the set value of the TTnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, the TOTm1 pin output is inverted and a compare match interrupt request signal (INTTTEQCn1) is generated.

By setting this register to the same value as the value set in the TTnCCR0 register, a PWM waveform with a duty factor of 50% can be output from the TOTm1 pin.

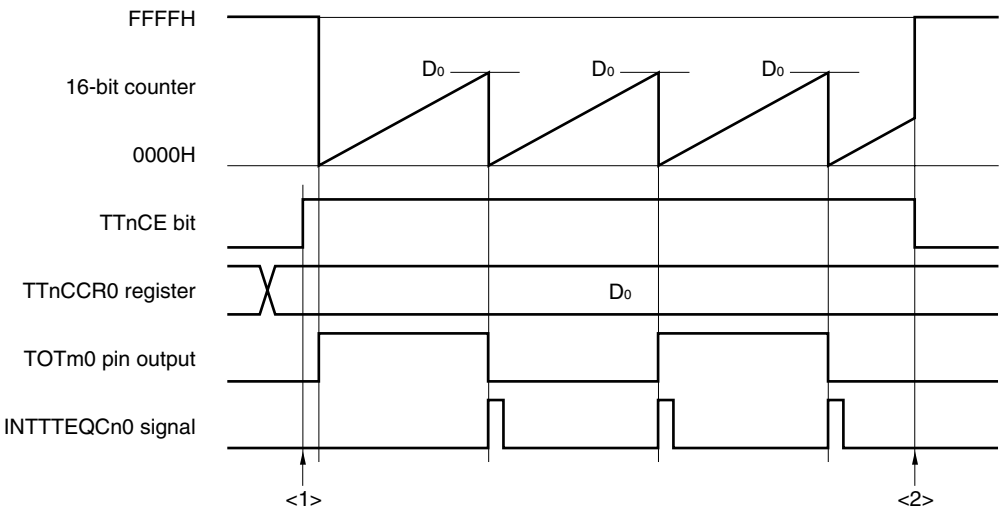
When the TTnCCR1 register is not used, it is recommended to set its value to FFFFH. Also mask the register by the interrupt mask flag (TTnCCIC1.TTnCCMK1).

Remarks 1. TMTm control register 2 (TTmCTL2), TMTm I/O control register 1 (TTmIOC1), TMTm I/O control register 2 (TTmIOC2), TMTm I/O control register 3 (TTmIOC3), TMTn option register 0 (TTnOPT0), TMTm option register 1 (TTmOPT1), TMTm capture input select register (TTISLm), and TMTm counter write register (TTmTCW) are not used in the interval timer mode.

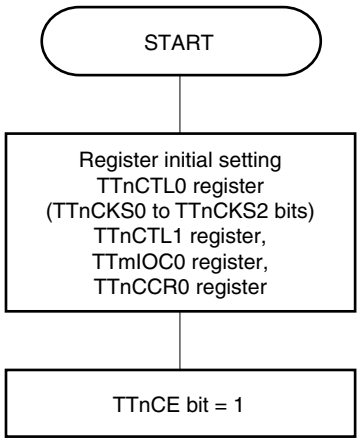
2. V850E/IF3: m = 1, n = 0, 1
V850E/IG3: m = 0, 1, n = 0, 1

(1) Interval timer mode operation flow

Figure 8-10. Software Processing Flow in Interval Timer Mode



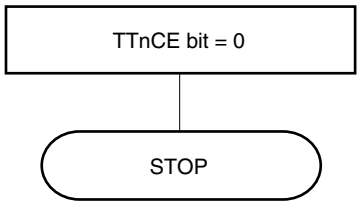
<1> Count operation start flow



Initial setting of these registers is performed before setting the TTnCE bit to 1.

The TTnCKS0 to TTnCKS2 bits can be set at the same time as when counting starts (TTnCE bit = 1).

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TTnCE bit to 0. The output level of the TOTm0 pin is as specified by the TTmIOC0 register.

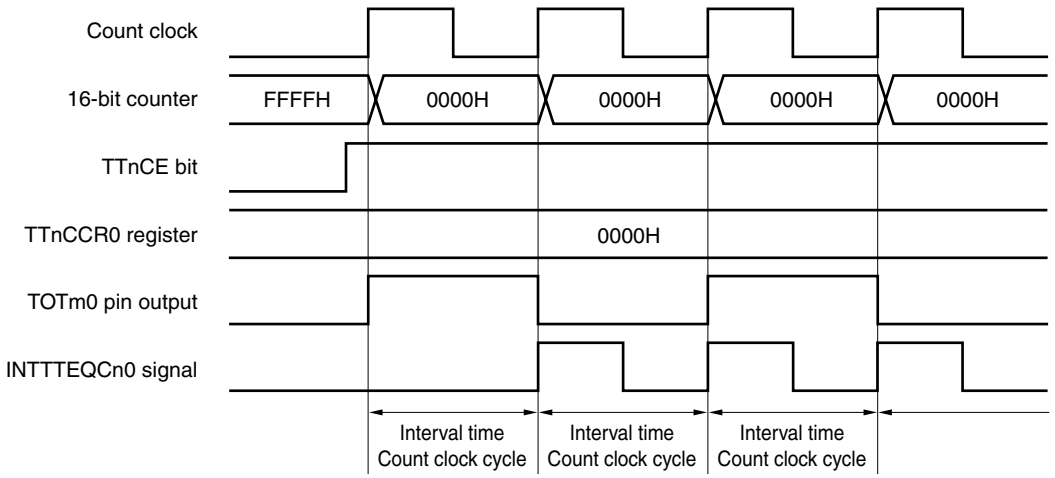
Remark V850E/IF3: n = 0, 1, m = 1
V850E/IG3: n = 0, 1, m = 0, 1

(2) Interval timer mode operation timing

(a) Operation if TTnCCR0 register is set to 0000H

If the TTnCCR0 register is set to 0000H, the INTTTEQCn0 signal is generated at each count clock, and the output of the TOTm0 pin is inverted.

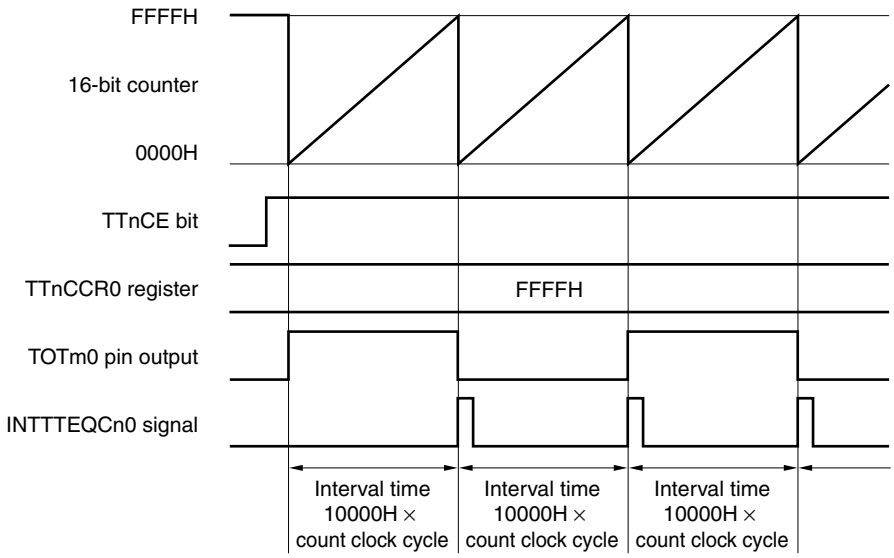
The value of the 16-bit counter is always 0000H.



Remark V850E/IF3: n = 0, 1, m = 1
V850E/IG3: n = 0, 1, m = 0, 1

(b) Operation if TTnCCR0 register is set to FFFFH

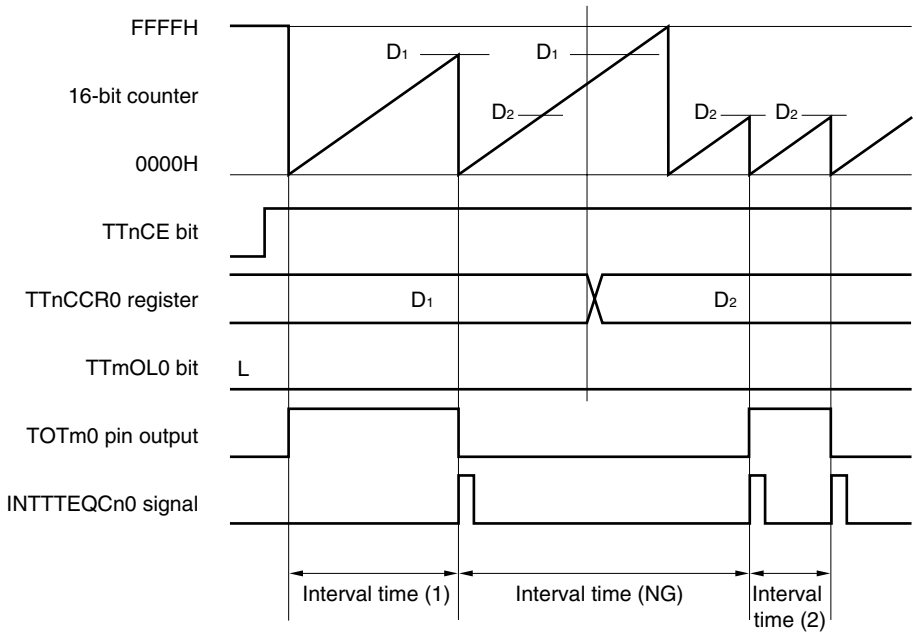
If the TTnCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTTEQCn0 signal is generated and the output of the TOTm0 pin is inverted. At this time, an overflow interrupt request signal (INTTTIOVn) is not generated, nor is the overflow flag (TTnOPT0.TTnOVF bit) set to 1.



Remark V850E/IF3: n = 0, 1, m = 1
V850E/IG3: n = 0, 1, m = 0, 1

(c) Notes on rewriting TTnCCR0 register

If the value of the TTnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When an overflow may occur, stop counting and then change the set value.



Remarks 1. Interval time (1): $(D_1 + 1) \times$ Count clock cycle
Interval time (NG): $(10000H + D_2 + 1) \times$ Count clock cycle
Interval time (2): $(D_2 + 1) \times$ Count clock cycle

2. V850E/IF3: n = 0, 1, m = 1
V850E/IG3: n = 0, 1, m = 0, 1

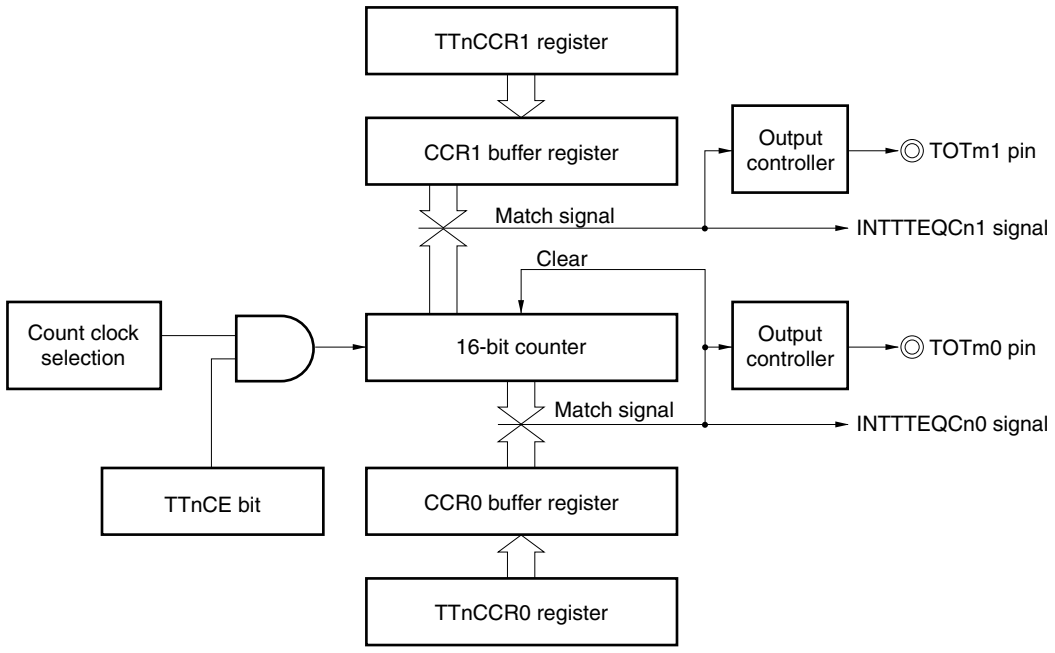
If the value of the TTnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TTnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTTEQCn0 signal is generated and the output of the TOTm0 pin is inverted.

Therefore, the INTTTEQCn0 signal may not be generated at the interval time "$(D_1 + 1) \times$ Count clock cycle" or "$(D_2 + 1) \times$ Count clock cycle" originally expected, but may be generated at an interval of "$(10000H + D_2 + 1) \times$ Count clock cycle".

(d) Operation of TTnCCR1 register

Figure 8-11. Configuration of TTnCCR1 Register



Remark V850E/IF3: n = 0, 1, m = 1
V850E/IG3: n = 0, 1, m = 0, 1

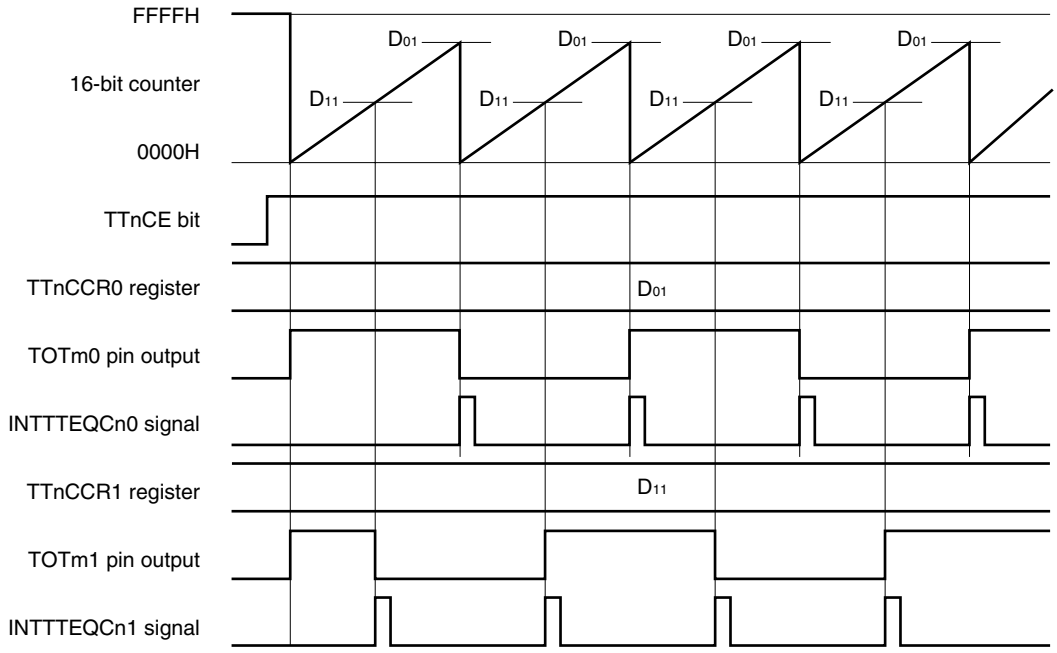
When the TTnCCR1 register is set to the same value as the TTnCCR0 register, the INTTTEQCn0 signal is generated at the same timing as the INTTTEQCn1 signal and the TOTm1 pin output is inverted. In other words, a PWM waveform with a duty factor of 50% can be output from the TOTm1 pin.

The following shows the operation when the TTnCCR1 register is set to other than the value set in the TTnCCR0 register.

If the set value of the TTnCCR1 register is less than the set value of the TTnCCR0 register, the INTTTEQCn1 signal is generated once per cycle. At the same time, the output of the TOTm1 pin is inverted.

The TOTm1 pin outputs a PWM waveform with a duty factor of 50% after outputting a short-width pulse.

Figure 8-12. Timing Chart When $D_{01} \geq D_{11}$

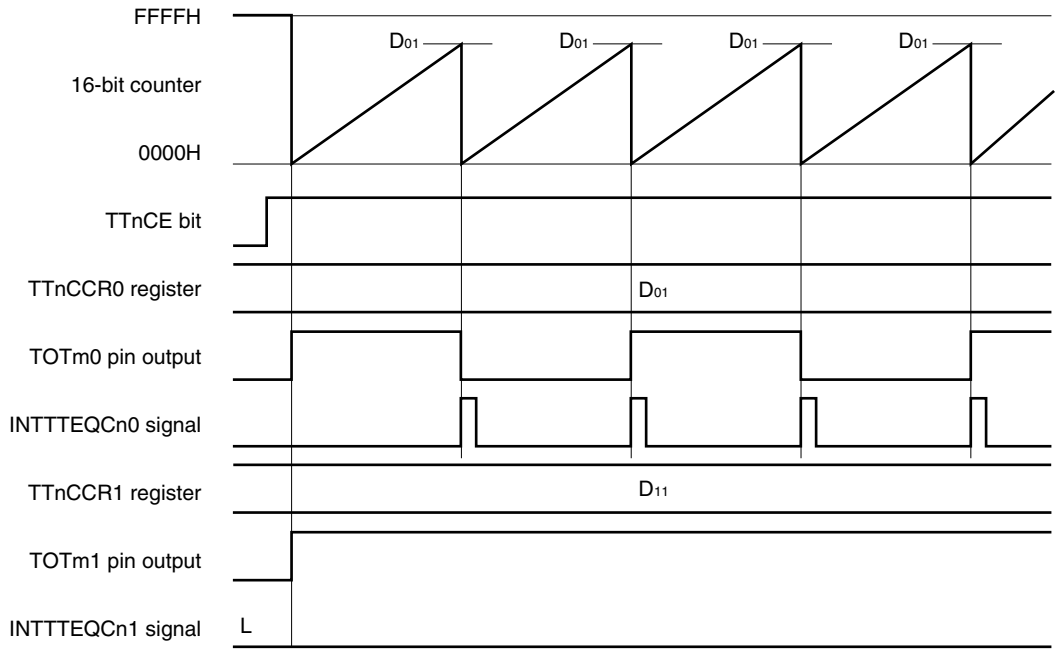


Remark V850E/IF3: n = 0, 1, m = 1
V850E/IG3: n = 0, 1, m = 0, 1

If the set value of the TTnCCR1 register is greater than the set value of the TTnCCR0 register, the count value of the 16-bit counter does not match the value of the TTnCCR1 register. Consequently, the INTTTEQCn1 signal is not generated, nor is the output of the TOTm1 pin changed.

When the TTnCCR1 register is not used, it is recommended to set its value to FFFFH.

Figure 8-13. Timing Chart When $D_{01} < D_{11}$



Remark V850E/IF3: n = 0, 1, m = 1
V850E/IG3: n = 0, 1, m = 0, 1

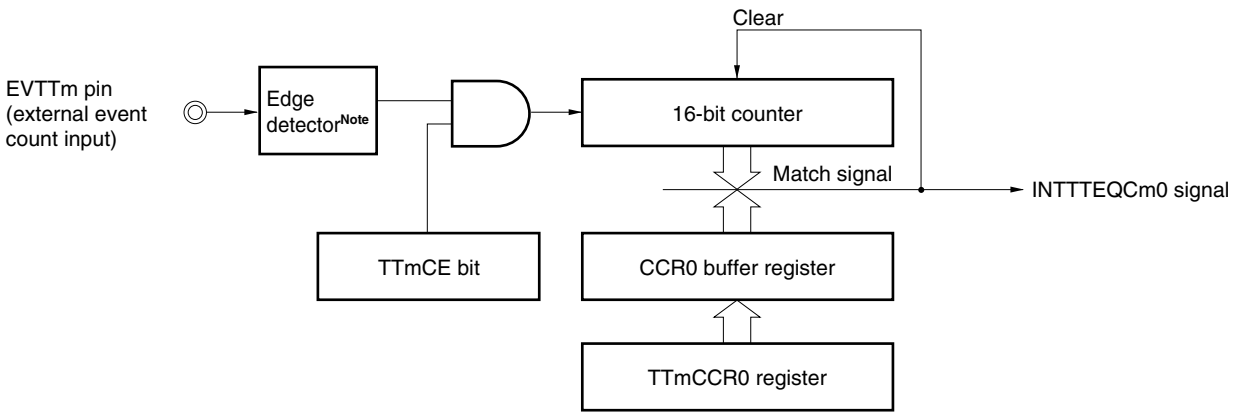
8.6.2 External event count mode (TTmMD3 to TTmMD0 bits = 0001)

This mode is valid only in TMT0 (V850E/IG3 only) and TMT1.

In the external event count mode, the valid edge of the external event count input (EVTTm) is counted when the TTmCTL0.TTmCE bit is set to 1, and an interrupt request signal (INTTTEQCm0) is generated each time the number of edges set by the TTmCCR0 register have been counted. The TOTm0 and TOTm1 pins cannot be used.

The TTmCCR1 register is not used in the external event count mode.

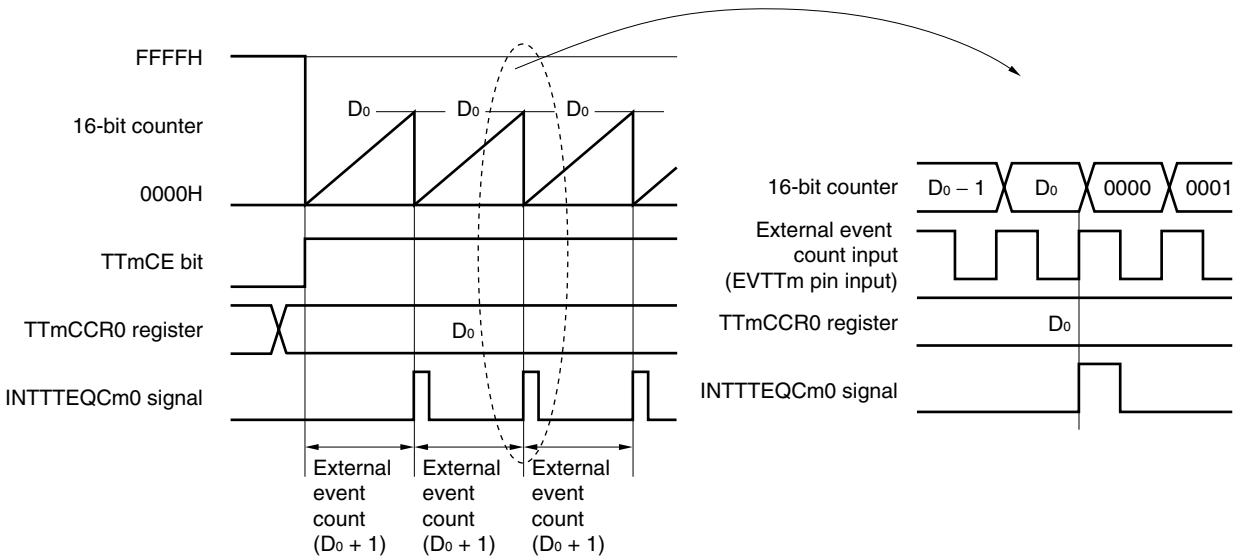
Figure 8-14. Configuration in External Event Count Mode



Note Set by the TTmIOC2.TTmEES1 and TTmIOC2.TTmEES0 bits.

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Figure 8-15. Basic Timing in External Event Count Mode



Remarks 1. This figure shows the basic timing when the rising edge is specified as the valid edge of the external event count input.

2. V850E/IF3: m = 1
V850E/IG3: m = 0, 1

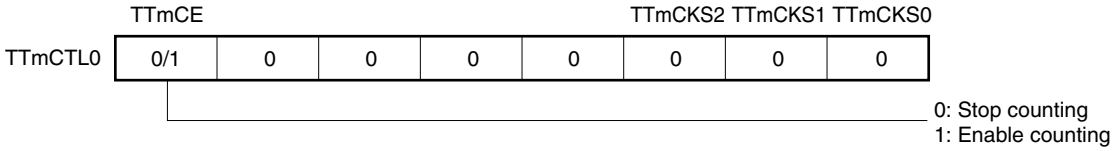
When the TTmCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H. The counter counts each time the valid edge of external event count input is detected. Additionally, the set value of the TTmCCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTTEQCm0) is generated.

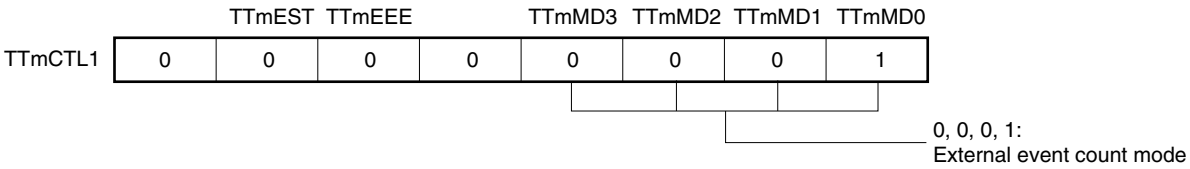
The INTTTEQCm0 signal is generated each time the valid edge of the external event count input has been detected "value set to TTmCCR0 register + 1" times.

Figure 8-16. Register Setting for Operation in External Event Count Mode (1/2)

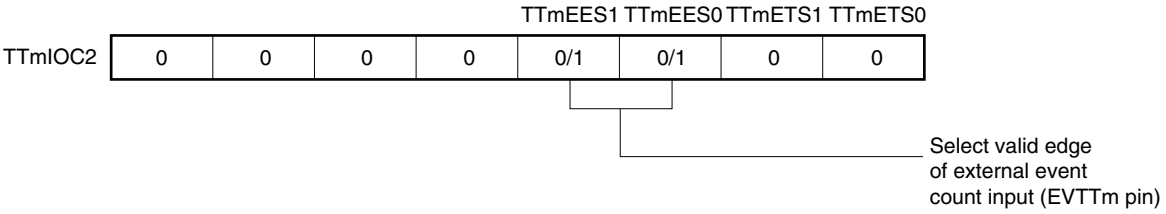
(a) TMTm control register 0 (TTmCTL0)



(b) TMTm control register 1 (TTmCTL1)



(c) TMTm I/O control register 2 (TTmIOC2)



(d) TMTm counter read buffer register (TTmCNT)

The count value of the 16-bit counter can be read by reading the TTmCNT register.

(e) TMTm capture/compare register 0 (TTmCCR0)

If the TTmCCR0 register is set to D_0, the count is cleared when the number of external events has reached (D_0 + 1) and the compare match interrupt request signal (INTTTEQCm0) is generated.

Figure 8-16. Register Setting for Operation in External Event Count Mode (2/2)

(f) TMTm capture/compare register 1 (TTmCCR1)

The TTmCCR1 register is not used in the external event count mode. However, the set value of the TTmCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTTEQCm1) is generated.

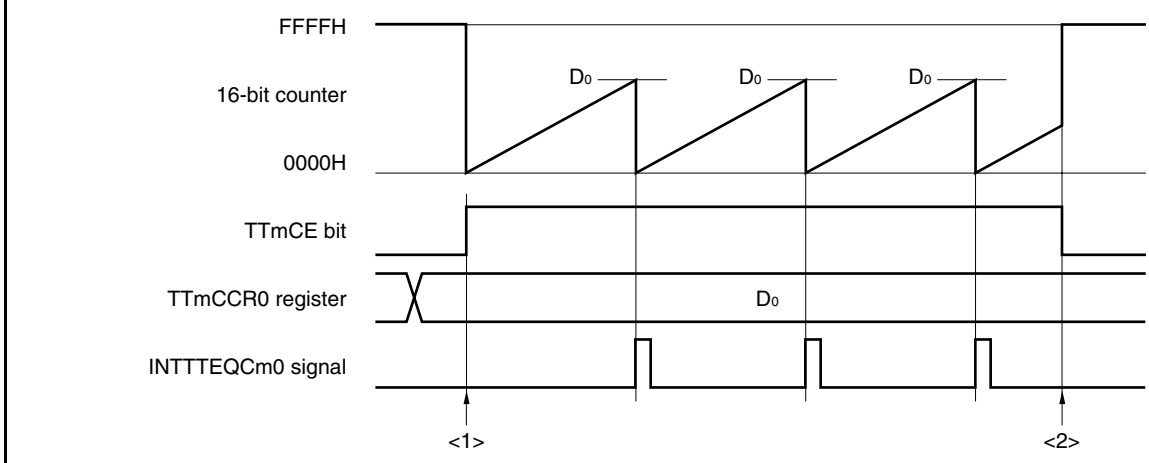
When the TTmCCR1 register is not used, it is recommended to set its value to FFFFH. Also mask the register by the interrupt mask flag (TTmCCIC1.TTmCCMK1).

Remarks 1. TMTm control register 2 (TTmCTL2), TMTm I/O control register 0 (TTmIOC0), TMTm I/O control register 1 (TTmIOC1), TMTm I/O control register 3 (TTmIOC3), TMTm option register 0 (TTmOPT0), TMTm option register 1 (TTmOPT1), TMTm capture input select register (TTISLm), and TMTm counter write register (TTmTCW) are not used in the external event count mode.

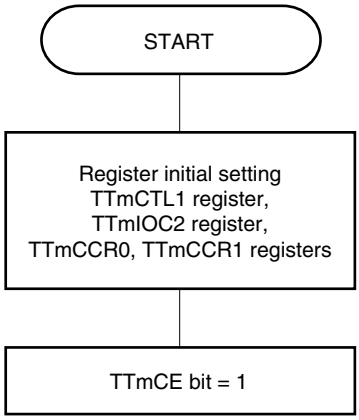
2. V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(1) External event count mode operation flow

Figure 8-17. Software Processing Flow in External Event Count Mode

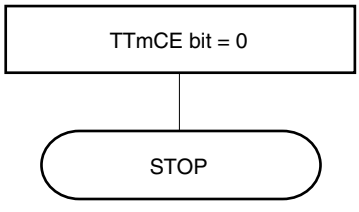


<1> Count operation start flow



Initial setting of these registers is performed before setting the TTmCE bit to 1.

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TTmCE bit to 0.

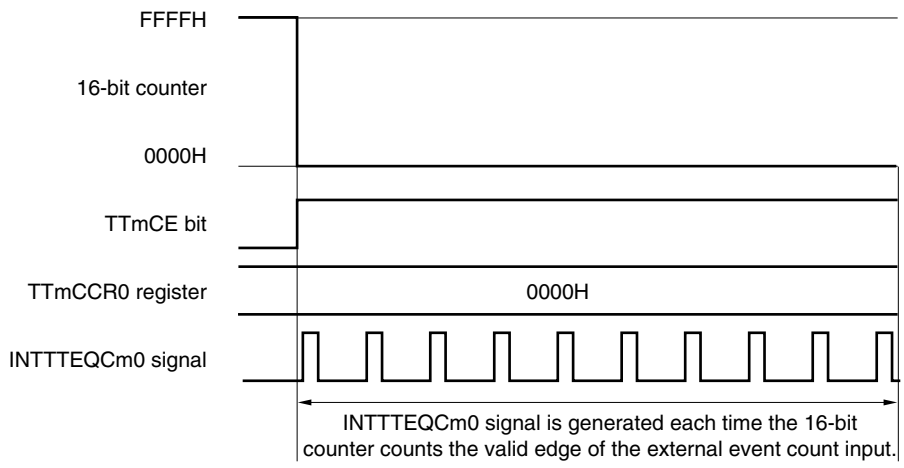
Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(2) Operation timing in external event count mode

(a) Operation if TTmCCR0 register is set to 0000H

When the TTmCCR0 register is set to 0000H, the 16-bit counter is repeatedly cleared to 0000H and generates the INTTTEQCm0 signal each time it has detected the valid edge of the external event count signal and its value has matched that of the CCR0 buffer register.

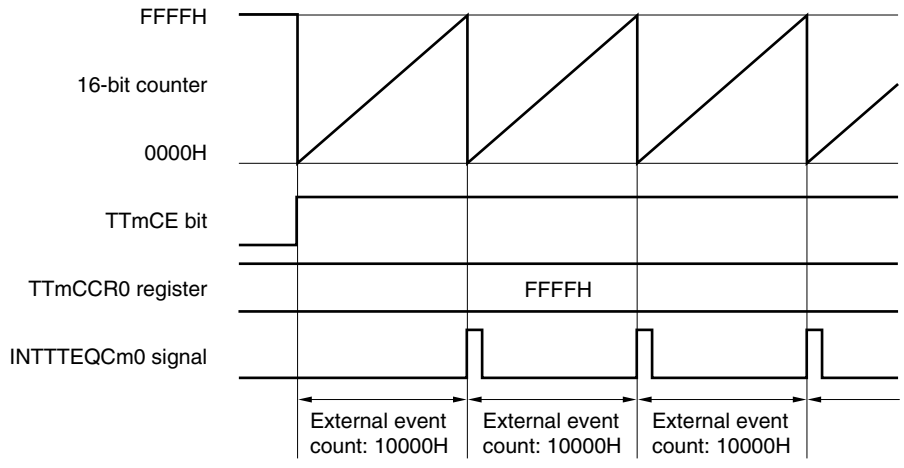
The value of the 16-bit counter is always 0000H.



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(b) Operation if TTmCCR0 register is set to FFFFH

If the TTmCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTTTEQCm0 signal is generated. At this time, the TTmOPT0.TTmOVF bit is not set.

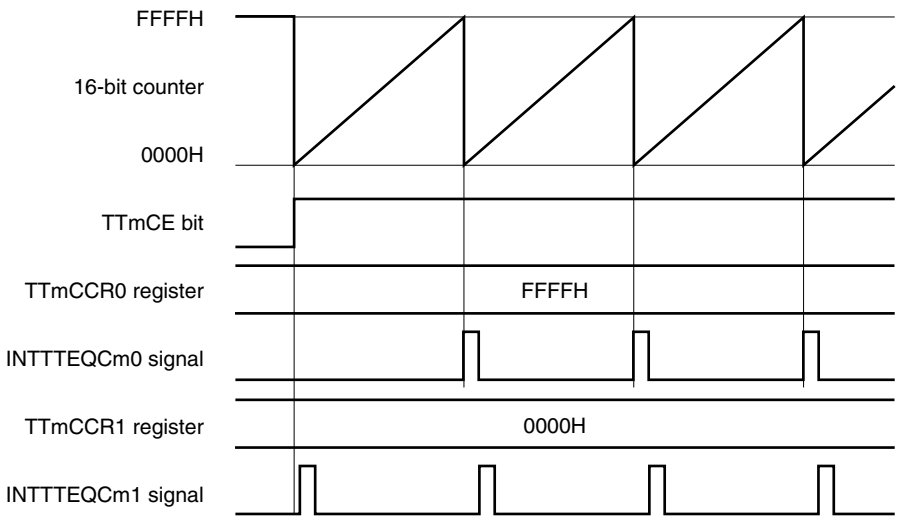


Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(c) Operation with TTmCCR0 set to FFFFH and TTmCCR1 register to 0000H

When the TTmCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH each time it has detected the valid edge of the external event count signal. The counter is then cleared to 0000H in synchronization with the next count-up timing and the INTTTEQCm0 signal is generated. At this time, the TTmOPT0.TTmOVF bit is not set.

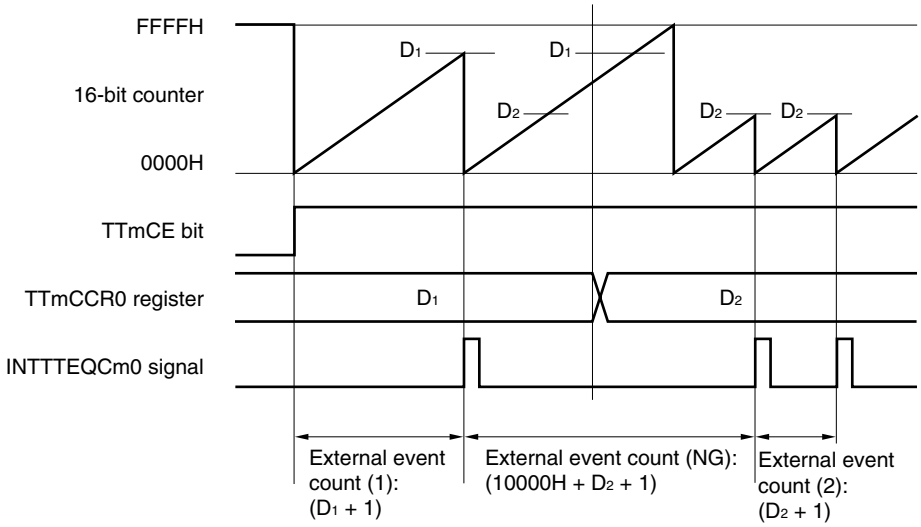
If the TTmCCR1 register is set to 0000H, the INTTTEQCm1 signal is generated when the 16-bit counter is cleared to 0000H.



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(d) Notes on rewriting the TTmCCR0 register

If the value of the TTmCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When the overflow may occur, stop counting once and then change the set value.



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

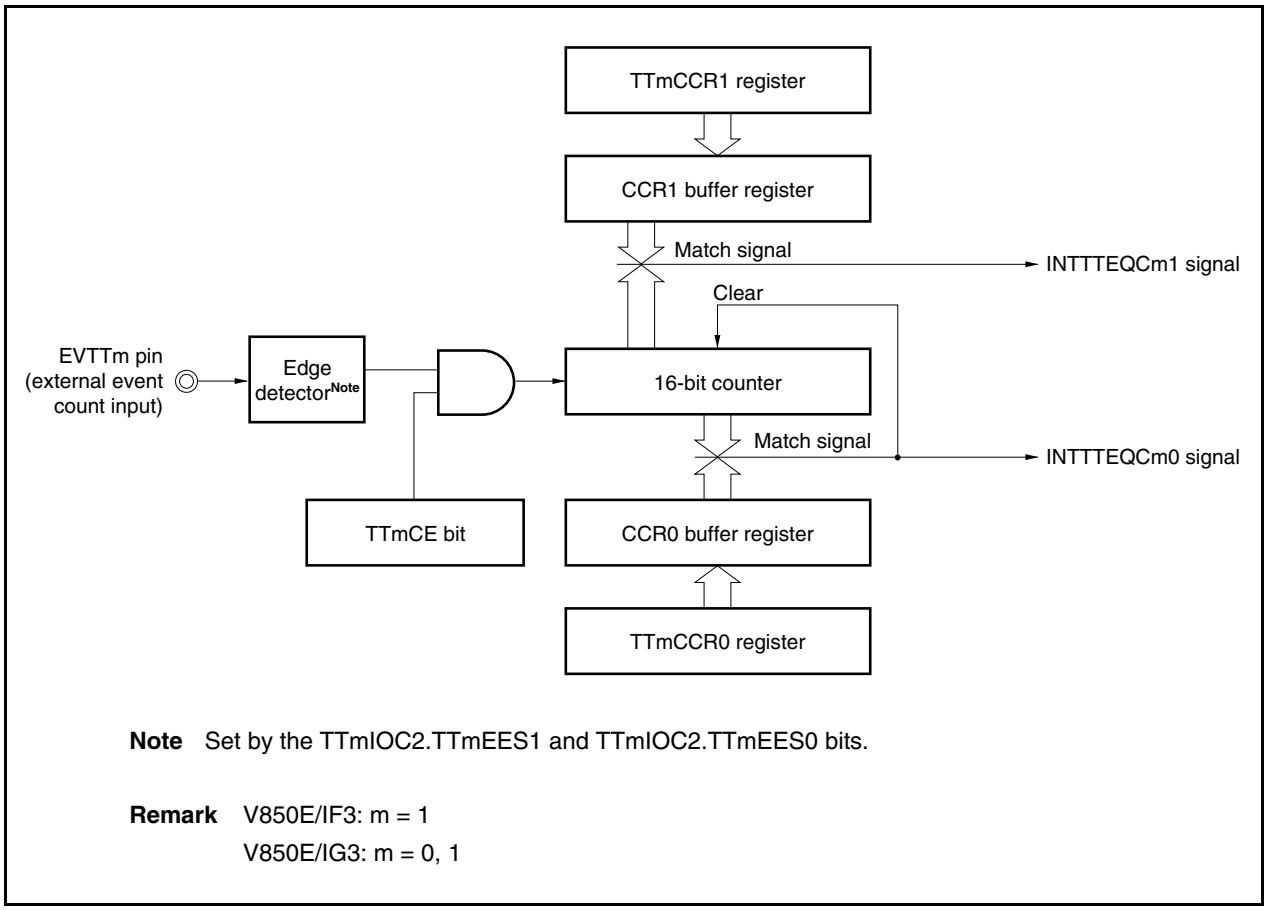
If the value of the TTmCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TTmCCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTTEQCm0 signal is generated.

Therefore, the INTTTEQCm0 signal may not be generated at the valid edge count of "(D_1 + 1) times" or "(D_2 + 1) times" originally expected, but may be generated at the valid edge count of "(10000H + D_2 + 1) times".

(e) Operation of TTmCCR1 register

Figure 8-18. Configuration of TTmCCR1 Register

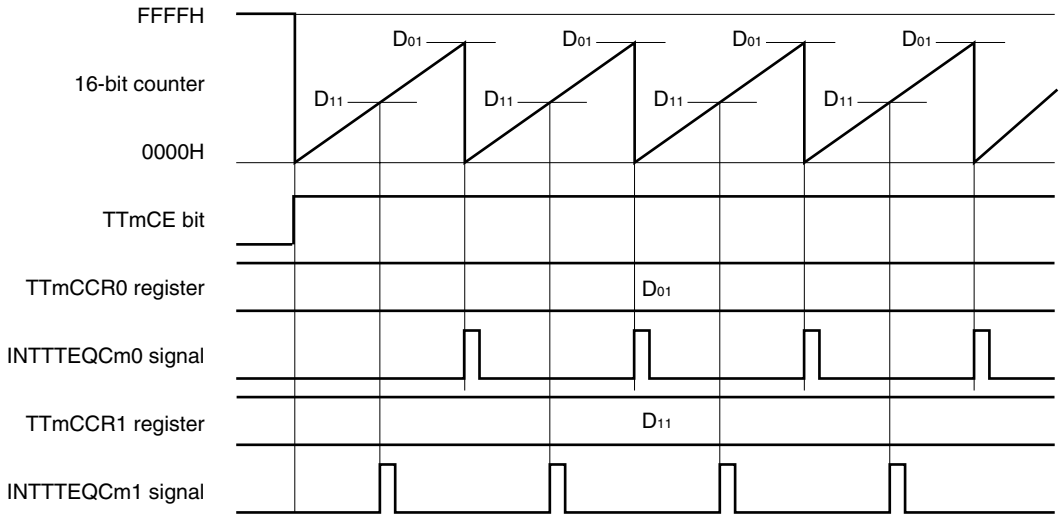


Note Set by the TTmIOC2.TTmEES1 and TTmIOC2.TTmEES0 bits.

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

If the set value of the TTmCCR1 register is smaller than the set value of the TTmCCR0 register, the INTTTEQCm1 signal is generated once per cycle.

Figure 8-19. Timing Chart When $D_{01} \geq D_{11}$

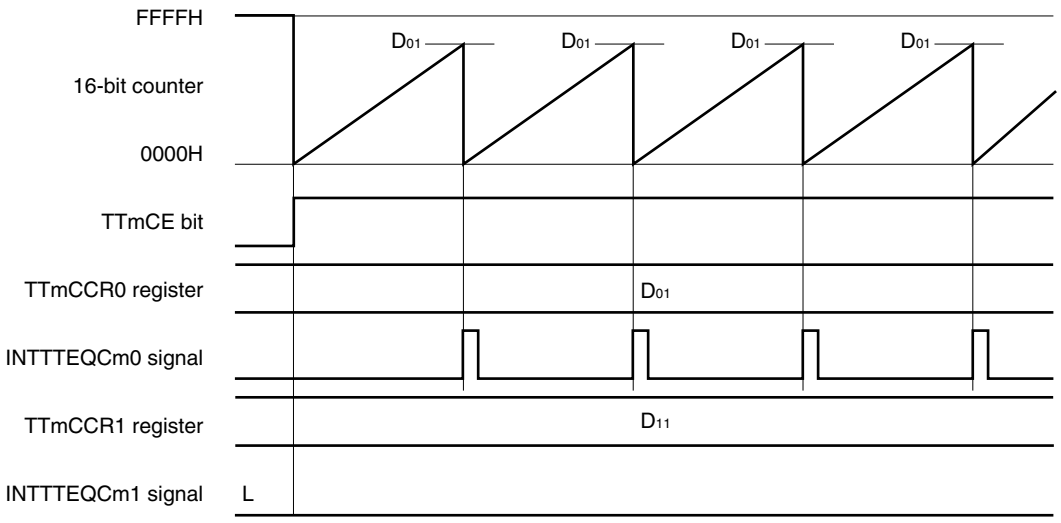


Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

If the set value of the TTmCCR1 register is greater than the set value of the TTmCCR0 register, the INTTTEQCm1 signal is not generated because the count value of the 16-bit counter and the value of the TTmCCR1 register do not match.

When the TTmCCR1 register is not used, it is recommended to set its value to FFFFH.

Figure 8-20. Timing Chart When $D_{01} < D_{11}$



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

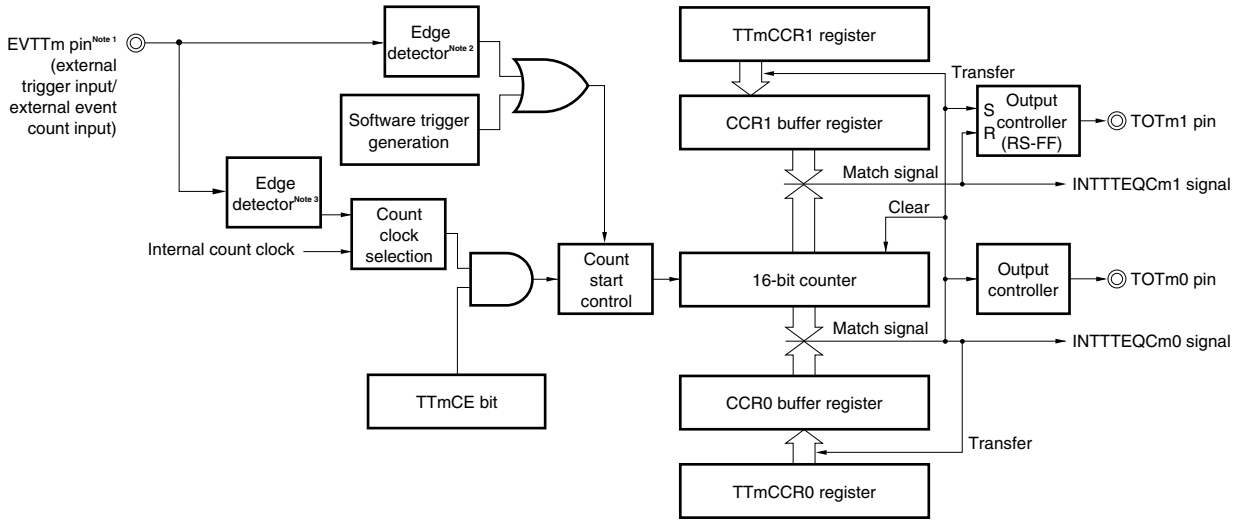
8.6.3 External trigger pulse output mode (TTmMD3 to TTmMD0 bits = 0010)

This mode is valid only in TMT0 (V850E/IG3 only) and TMT1.

In the external trigger pulse output mode, 16-bit timer/event counter T waits for a trigger when the TTmCTL0.TTmCE bit is set to 1. When the valid edge of an external trigger input (EVTTm) is detected, 16-bit timer/event counter T starts counting, and outputs a PWM waveform from the TOTm1 pin.

Pulses can also be output by generating a software trigger instead of using the external trigger. When using a software trigger, a PWM waveform with a duty factor of 50% that has the set value of the TTmCCR0 register + 1 as half its cycle can also be output from the TOTm0 pin.

Figure 8-21. Configuration in External Trigger Pulse Output Mode



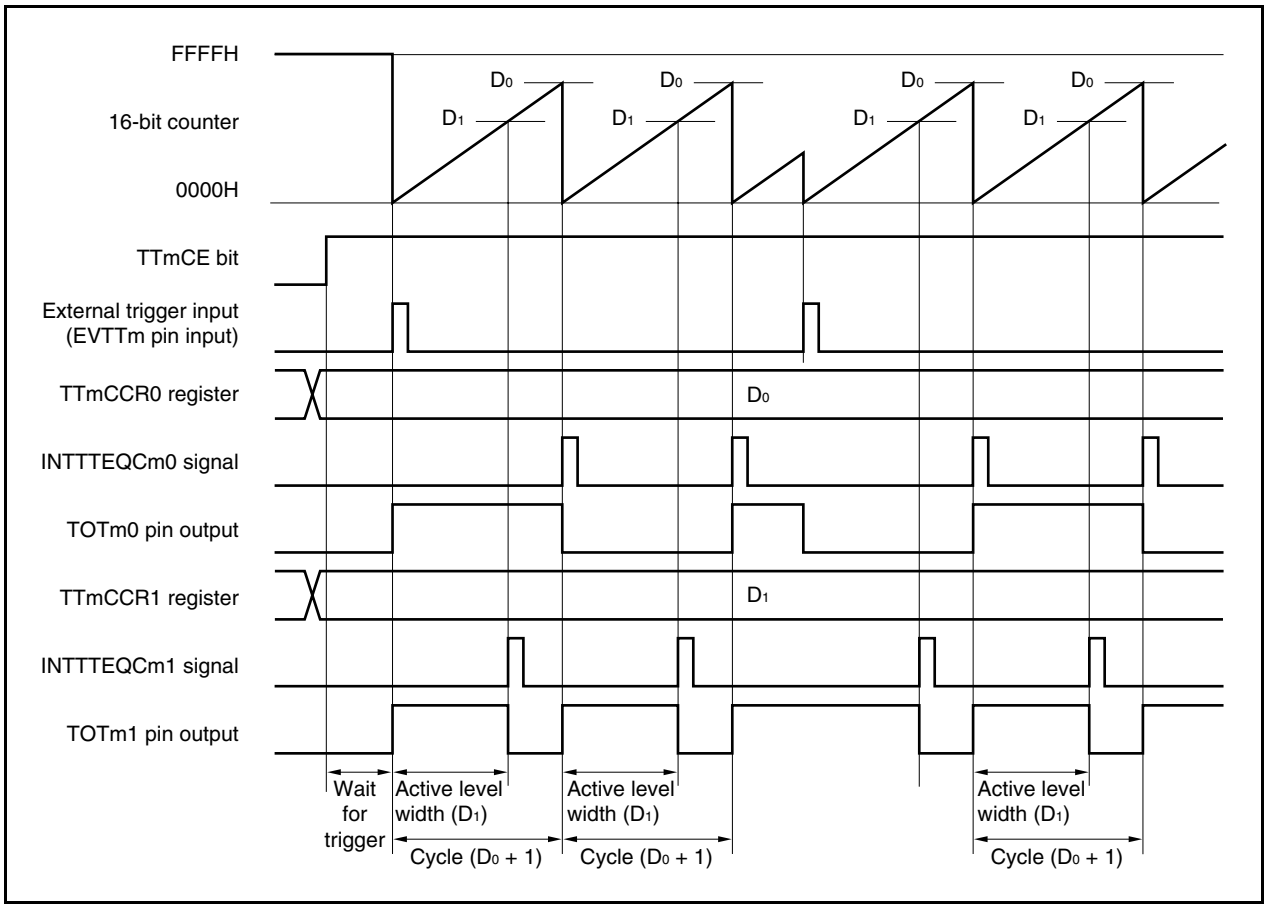
Notes 1. Since the external trigger input pin (EVTTm) and external event count input pin (EVTTm) are the same alternate-function pin, two or more functions cannot be used at the same time.

2. Edge detector for external trigger input.
Set by the TTmIOC2.TTmETS1 and TTmIOC2.TTmETS0 bits.

3. Edge detector for external event count input.
Set by the TTmIOC2.TTmEES1 and TTmIOC2.TTmEES0 bits.

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Figure 8-22. Basic Timing in External Trigger Pulse Output Mode



16-bit timer/event counter T waits for a trigger when the TTmCE bit is set to 1. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting at the same time, and outputs a PWM waveform from the TOTm1 pin. If the trigger is generated again while the counter is operating, the counter is cleared to 0000H and restarted. (The output of the TOTm0 pin is inverted. The TOTm1 pin outputs a high-level regardless of the status (high/low) when a trigger occurs.)

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TTmCCR1 register) × Count clock cycle
Cycle = (Set value of TTmCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TTmCCR1 register)/(Set value of TTmCCR0 register + 1)

The compare match request signal (INTTTEQCm0) is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal (INTTTEQCm1) is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

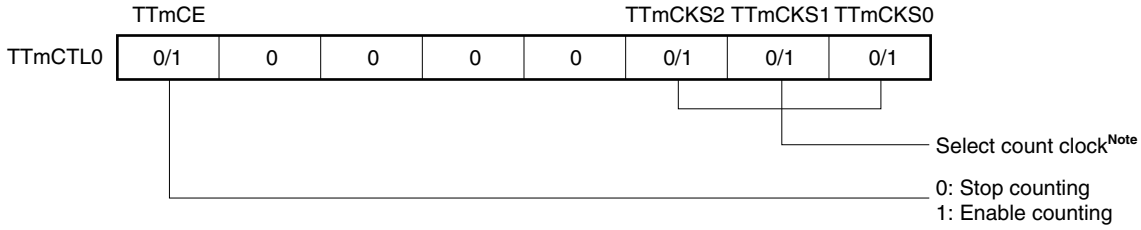
The value set to the TTmCCRa register is transferred to the CCRa buffer register when the count value of the 16-bit counter matches the value of the CCRa buffer register and the 16-bit counter is cleared to 0000H.

The valid edge of an external trigger input (EVTTm), or setting the software trigger (TTmCTL1.TTmEST bit) to 1 is used as the trigger.

Remark V850E/IF3: m = 1, a = 0, 1
V850E/IG3: m = 0, 1, a = 0, 1

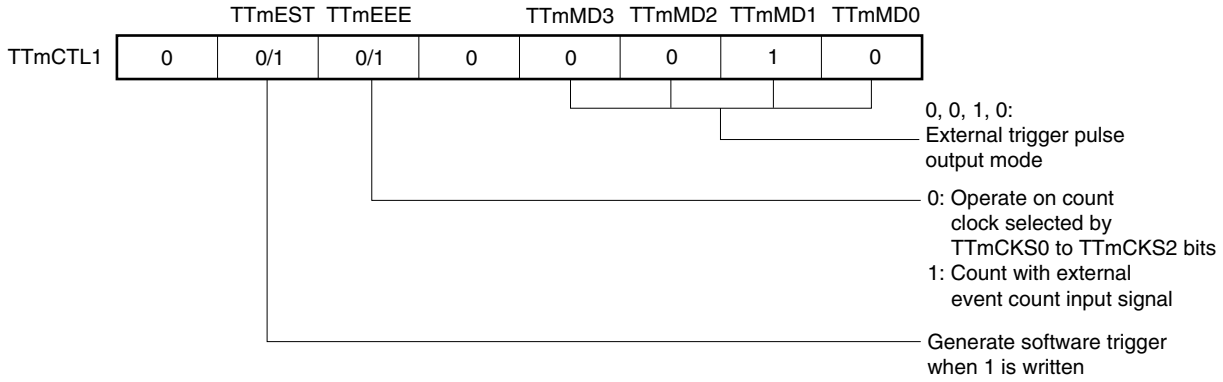
Figure 8-23. Setting of Registers in External Trigger Pulse Output Mode (1/2)

(a) TMTm control register 0 (TTmCTL0)

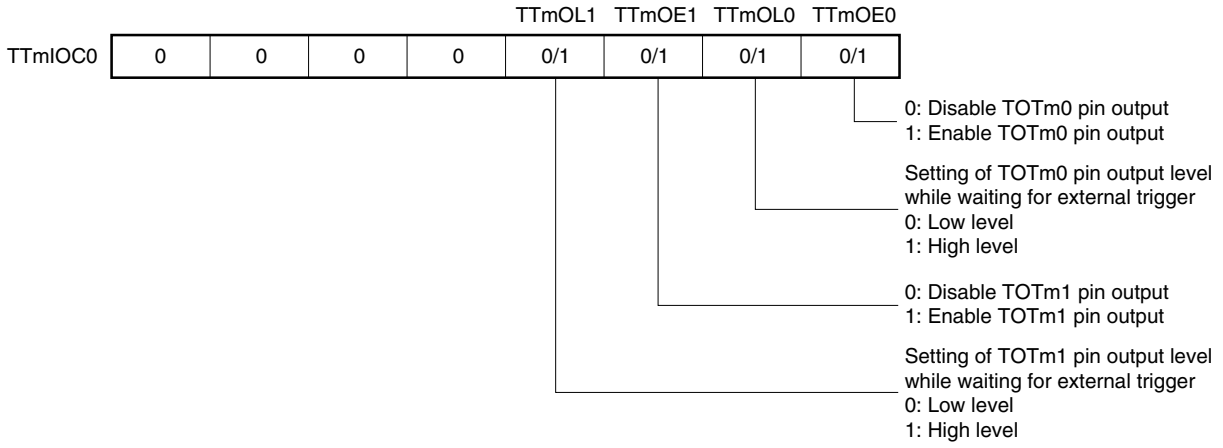


Note The setting is invalid when the TTmCTL1.TTmEEE bit = 1.

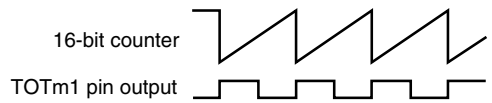
(b) TMTm control register 1 (TTmCTL1)



(c) TMTm I/O control register 0 (TTmIOC0)



- When TTmOL1 bit = 0



- When TTmOL1 bit = 1

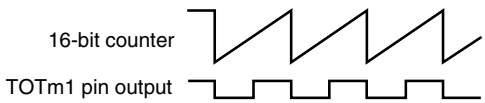
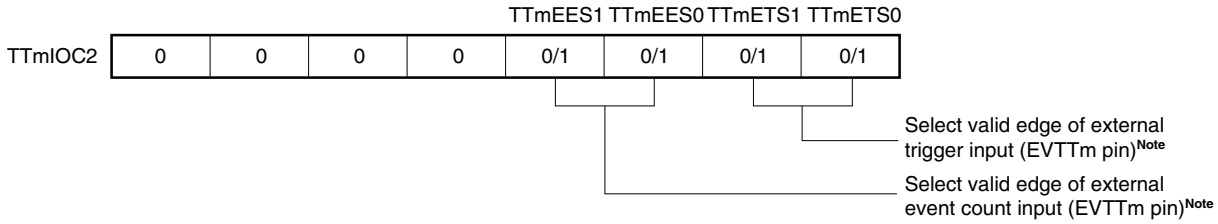


Figure 8-23. Setting of Registers in External Trigger Pulse Output Mode (2/2)

(d) TMTm I/O control register 2 (TTmIOC2)



Note Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

(e) TMTm counter read buffer register (TTmCNT)

The value of the 16-bit counter can be read by reading the TTmCNT register.

(f) TMTm capture/compare registers 0 and 1 (TTmCCR0 and TTmCCR1)

If D_0 is set to the TTmCCR0 register and D_1 to the TTmCCR1 register, the cycle and active level of the PWM waveform are as follows.

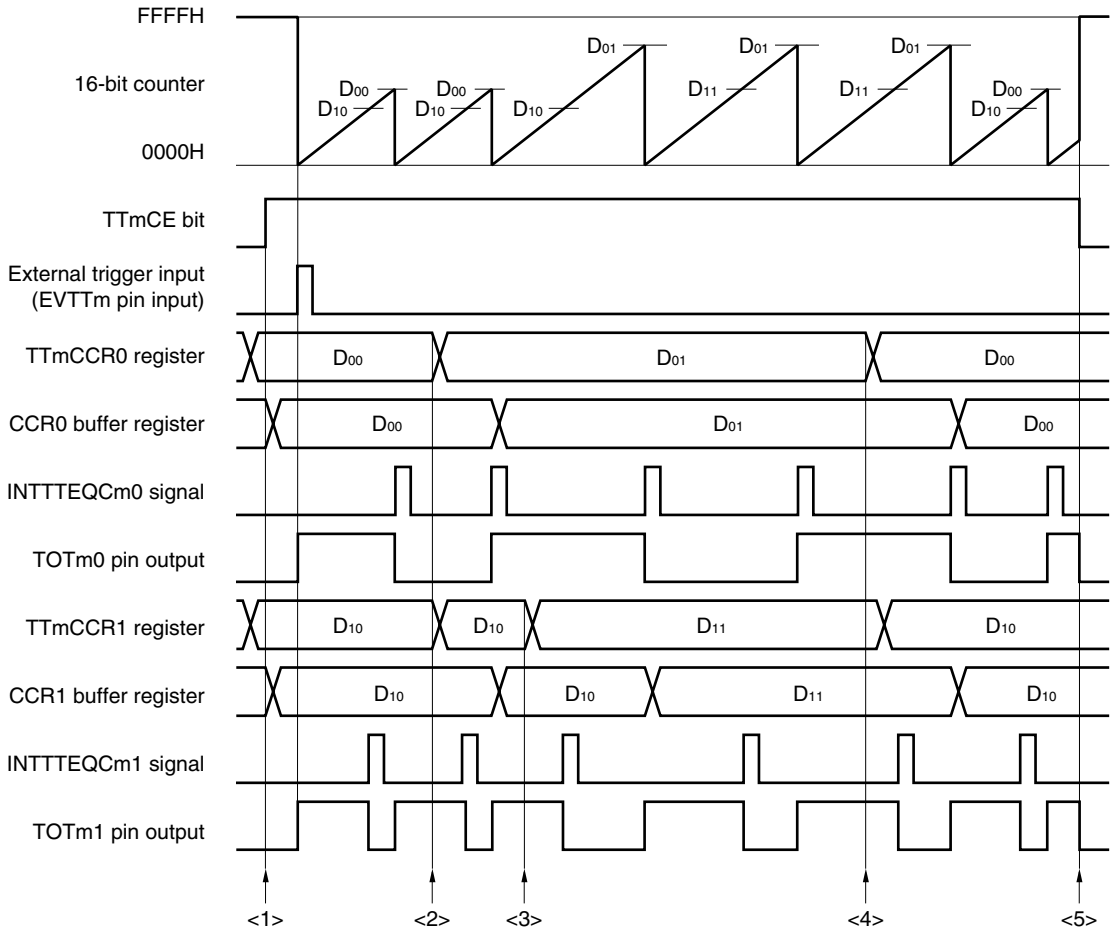
Cycle = $(D_0 + 1) \times$ Count clock cycle
Active level width = $D_1 \times$ Count clock cycle

Remarks 1. TMTm control register 2 (TTmCTL2), TMTm I/O control register 1 (TTmIOC1), TMTm I/O control register 3 (TTmIOC3), TMTm option register 0 (TTmOPT0), TMTm option register 1 (TTmOPT1), TMTm capture input select register (TTISLm), and TMTm counter write register (TTmTCW) are not used in the external trigger pulse output mode.

2. V850E/IF3: m = 1
V850E/IG3: m = 0, 1

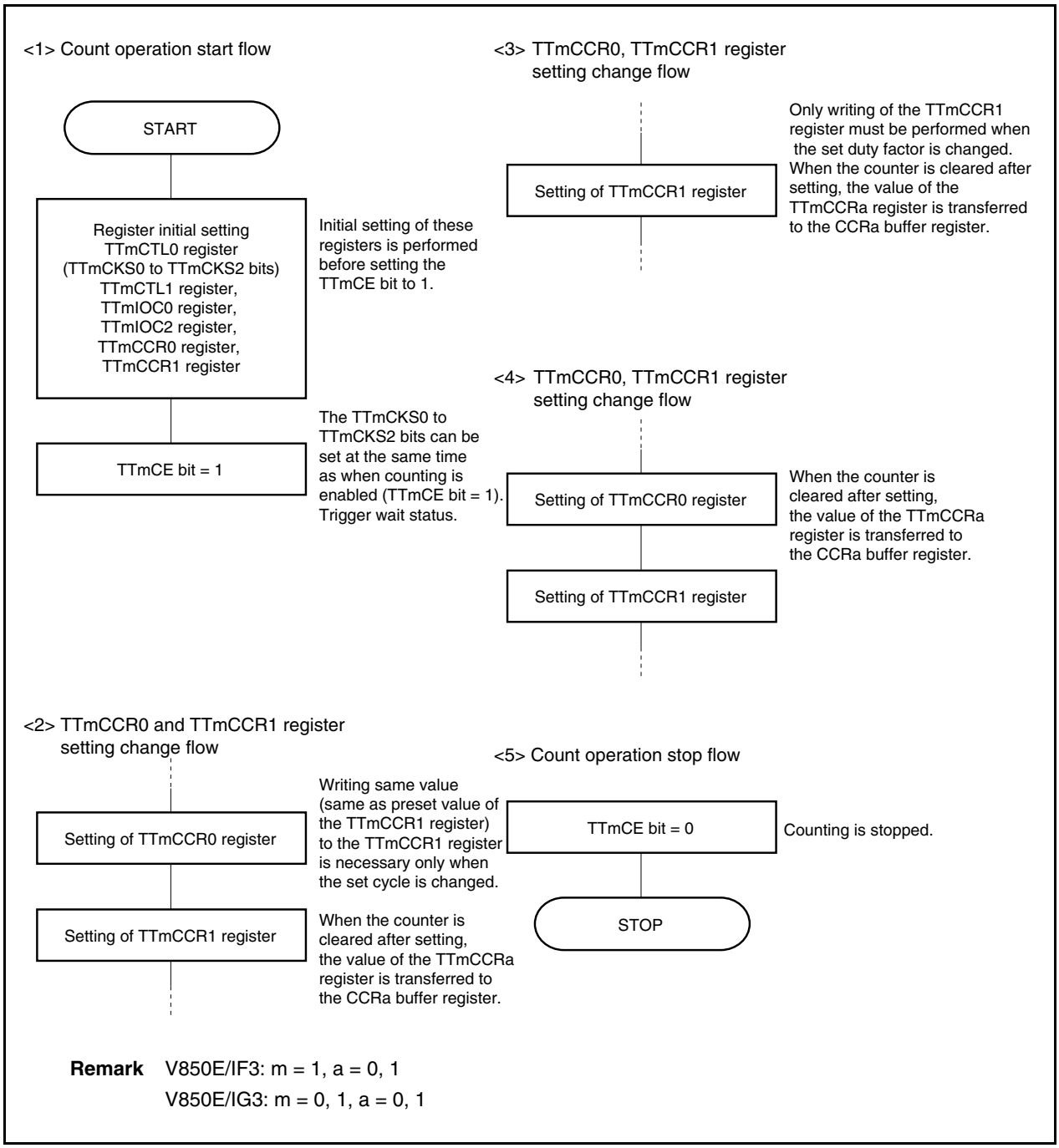
(1) Operation flow in external trigger pulse output mode

Figure 8-24. Software Processing Flow in External Trigger Pulse Output Mode (1/2)



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Figure 8-24. Software Processing Flow in External Trigger Pulse Output Mode (2/2)



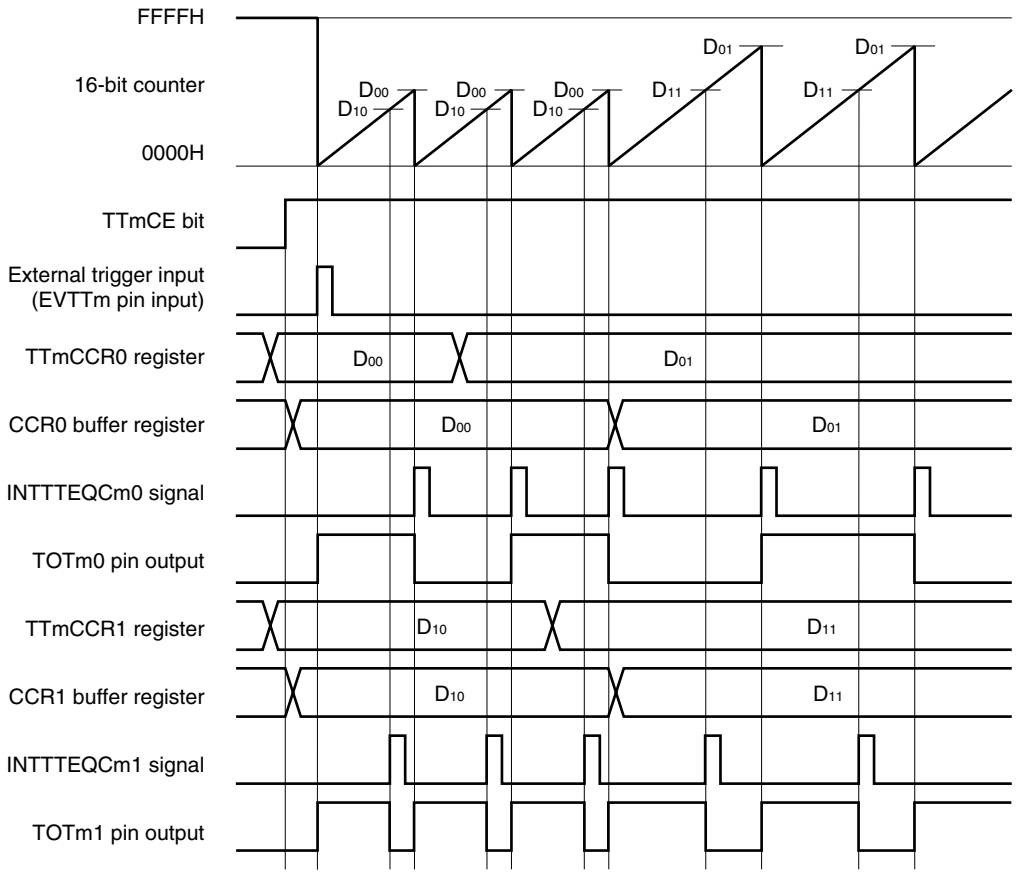
Remark V850E/IF3: m = 1, a = 0, 1
V850E/IG3: m = 0, 1, a = 0, 1

(2) External trigger pulse output mode operation timing

(a) Note on changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TTmCCR1 register last.

Rewrite the TTmCCRa register after writing the TTmCCR1 register after the INTTTEQCm0 signal is detected.



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

In order to transfer data from the TTmCCRa register to the CCRa buffer register, the TTmCCR1 register must be written.

To change both the cycle and active level width of the PWM waveform at this time, first set the cycle to the TTmCCR0 register and then set the active level width to the TTmCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TTmCCR0 register, and then write the same value (same as preset value of the TTmCCR1 register) to the TTmCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TTmCCR1 register has to be set.

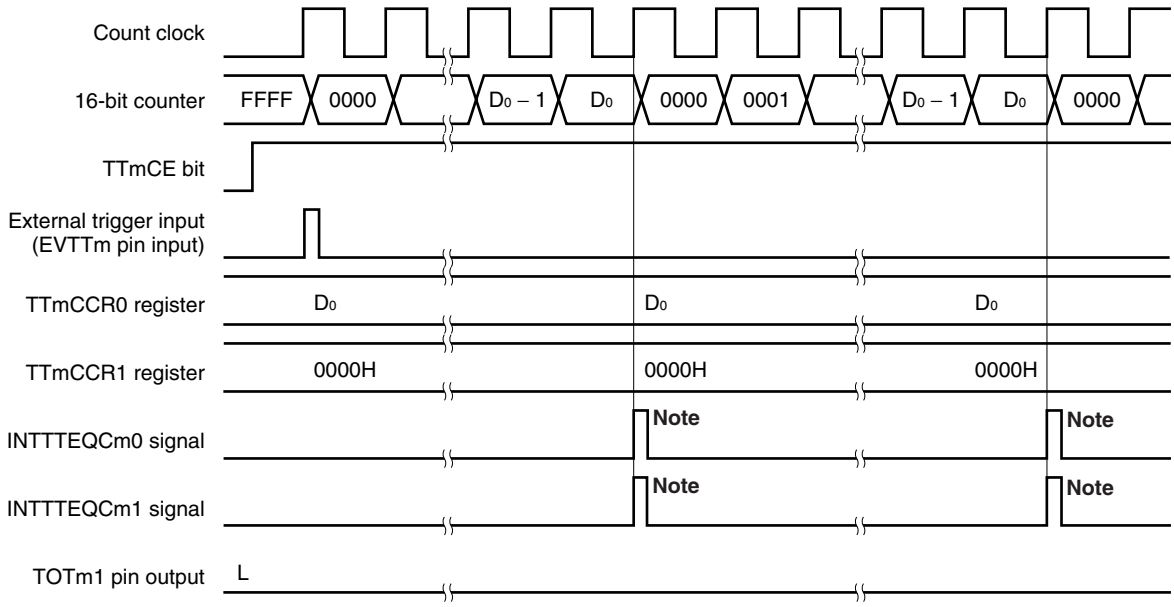
After data is written to the TTmCCR1 register, the value written to the TTmCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TTmCCR0 or TTmCCR1 register again after writing the TTmCCR1 register once, do so after the INTTTEQCm0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TTmCCRa register to the CCRa buffer register conflicts with writing the TTmCCRa register.

Remark V850E/IF3: m = 1, a = 0, 1
V850E/IG3: m = 0, 1, a = 0, 1

(b) 0%/100% output of PWM waveform

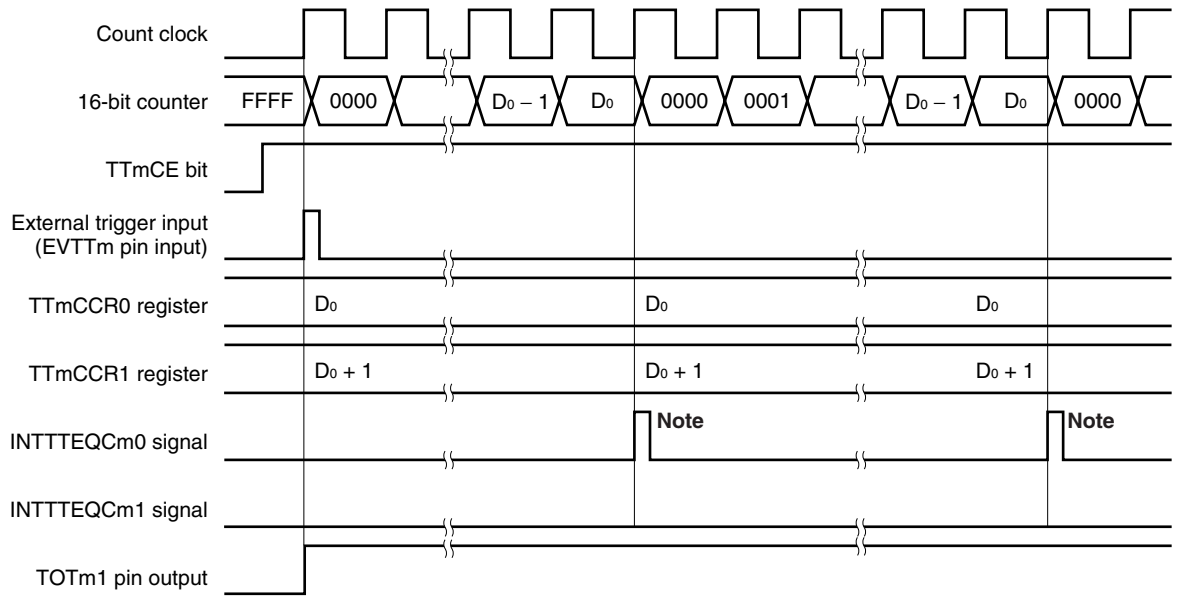
To output a 0% waveform, set the TTmCCR1 register to 0000H. The 16-bit counter is cleared to 0000H and the INTTTEQCm0 and INTTTEQCm1 signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.



Note The timing is actually delayed by one operating clock (fxx).

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

To output a 100% waveform, set a value of (set value of TTmCCR0 register + 1) to the TTmCCR1 register. If the set value of the TTmCCR0 register is FFFFH, 100% output cannot be produced.

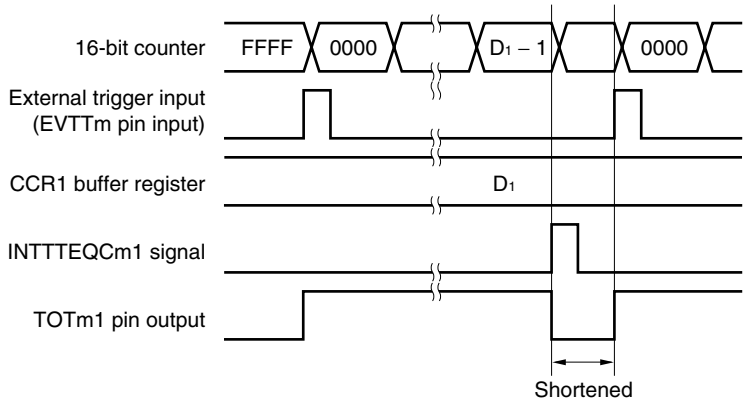


Note The timing is actually delayed by one operating clock (fxx).

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

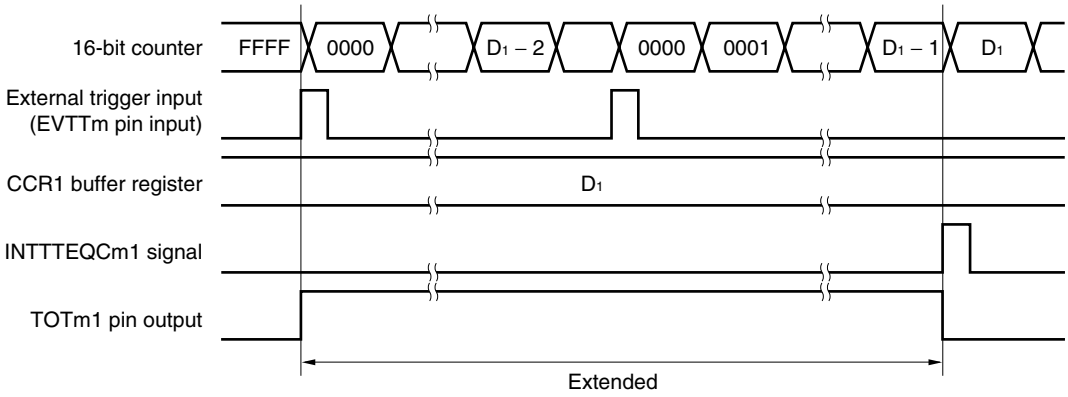
(c) Conflict between trigger detection and match with CCR1 buffer register

If the trigger is detected immediately after the INTTTEQCm1 signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TOTm1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

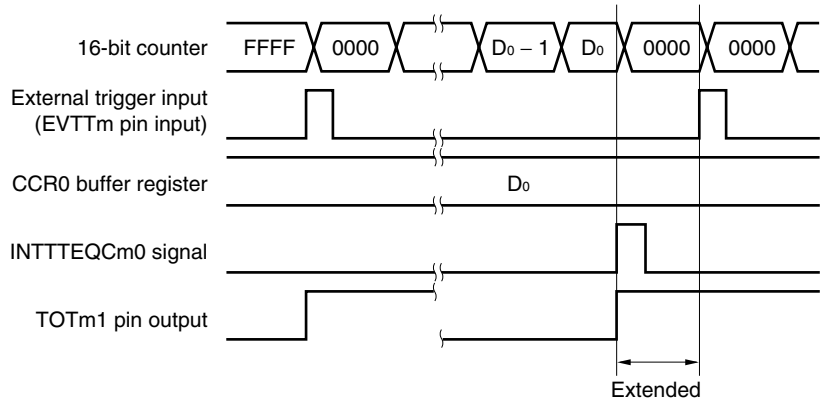
If the trigger is detected immediately before the INTTTEQCm1 signal is generated, the INTTTEQCm1 signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOTm1 pin remains active. Consequently, the active period of the PWM waveform is extended.



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

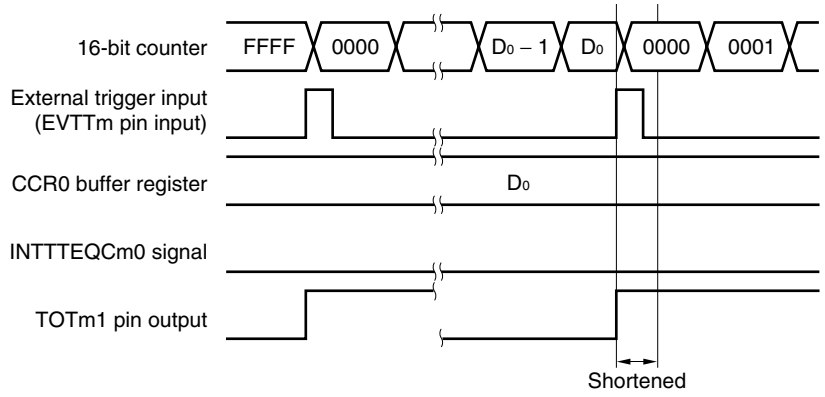
(d) Conflict between trigger detection and match with CCR0 buffer register

If the trigger is detected immediately after the INTTTEQCm0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOTm1 pin is extended by time from generation of the INTTTEQCm0 signal to trigger detection.



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

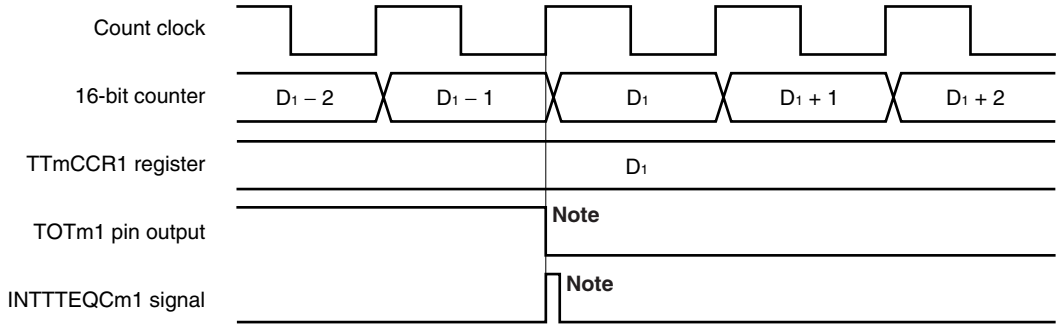
If the trigger is detected immediately before the INTTTEQCm0 signal is generated, the INTTTEQCm0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOTm1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(e) Generation timing of compare match interrupt request signal (INTTTEQCm1)

The timing of generation of the INTTTEQCm1 signal in the external trigger pulse output mode differs from the timing of INTTTEQCm1 signals in other mode; the INTTTEQCm1 signal is generated when the count value of the 16-bit counter matches the value of the TTmCCR1 register.



Note The timing is actually delayed by one operating clock (fxx).

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Usually, the INTTTEQCm1 signal is generated in synchronization with the next count-up, after the count value of the 16-bit counter matches the value of the TTmCCR1 register.

In the external trigger pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the timing of changing the output signal of the TOTm1 pin.

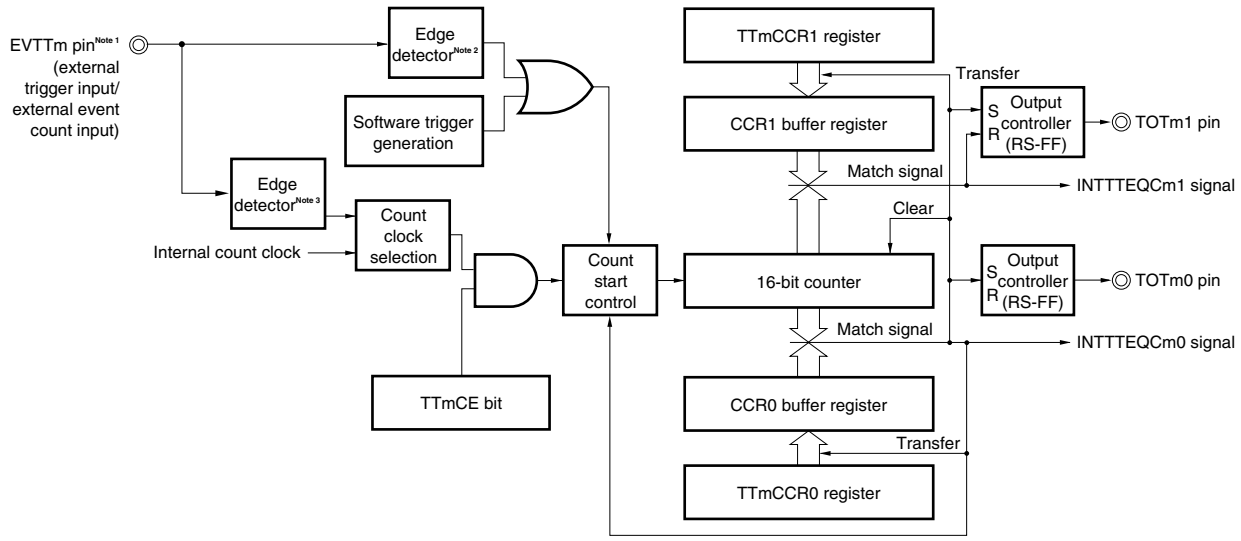
8.6.4 One-shot pulse output mode (TTmMD3 to TTmMD0 bits = 0011)

This mode is valid only in TMT0 (V850E/IG3 only) and TMT1.

In the one-shot pulse output mode, 16-bit timer/event counter T waits for a trigger when the TTmCTL0.TTmCE bit is set to 1. When the valid edge of an external trigger input (EVTTm) is detected, 16-bit timer/event counter T starts counting, and outputs a one-shot pulse from the TOTm1 pin.

Instead of the external trigger input (EVTTm), a software trigger can also be generated to output the pulse. When the software trigger is used, the TOTm0 pin outputs the active level while the 16-bit counter is counting, and the inactive level when the counter is stopped (waiting for a trigger).

Figure 8-25. Configuration in One-Shot Pulse Output Mode



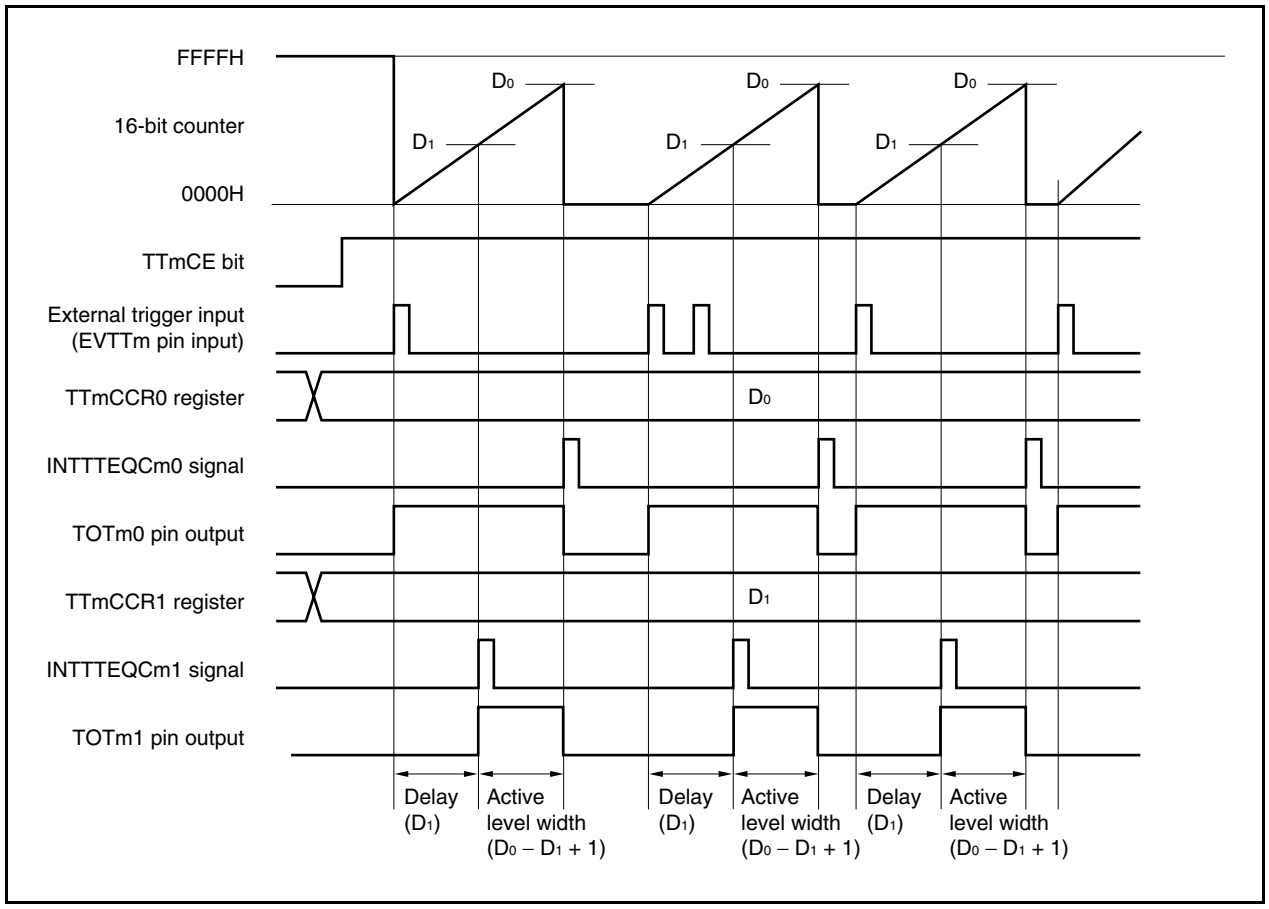
Notes 1. Because the external trigger input pin (EVTTm) and external event count input pin (EVTTm) are the same alternate-function pin, two functions cannot be used at the same time.

2. Edge detector for external trigger input.
Set by the TTmIOC2.TTmETS1 and TTmIOC2.TTmETS0 bits.

3. Edge detector for external event count input.
Set by the TTmIOC2.TTmEES1 and TTmIOC2.TTmEES0 bits.

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Figure 8-26. Basic Timing in One-Shot Pulse Output Mode



When the TTmCE bit is set to 1, 16-bit timer/event counter T waits for a trigger. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a one-shot pulse from the TOTm1 pin. After the one-shot pulse is output, the 16-bit counter is cleared to 0000H, stops counting, and waits for a trigger. When the trigger is generated again, the 16-bit counter starts counting from 0000H. If a trigger is generated again while the one-shot pulse is being output, it is ignored.

The output delay period and active level width of the one-shot pulse can be calculated as follows.

Output delay period = (Set value of TTmCCR1 register) × Count clock cycle

Active level width = (Set value of TTmCCR0 register − Set value of TTmCCR1 register + 1) × Count clock cycle

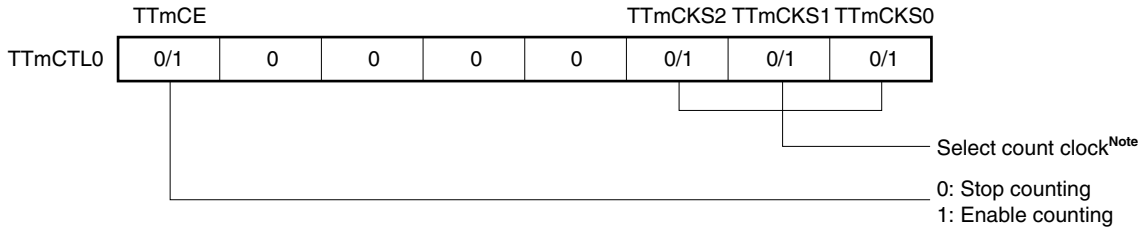
The compare match interrupt request signal (INTTTEQCm0) is generated when the 16-bit counter counts after its count value matches the value of the CCR0 buffer register. The compare match interrupt request signal (INTTTEQCm1) is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The valid edge of an external trigger input (EVTTm pin) or setting the software trigger (TTmCTL1.TTmEST bit) to 1 is used as the trigger.

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

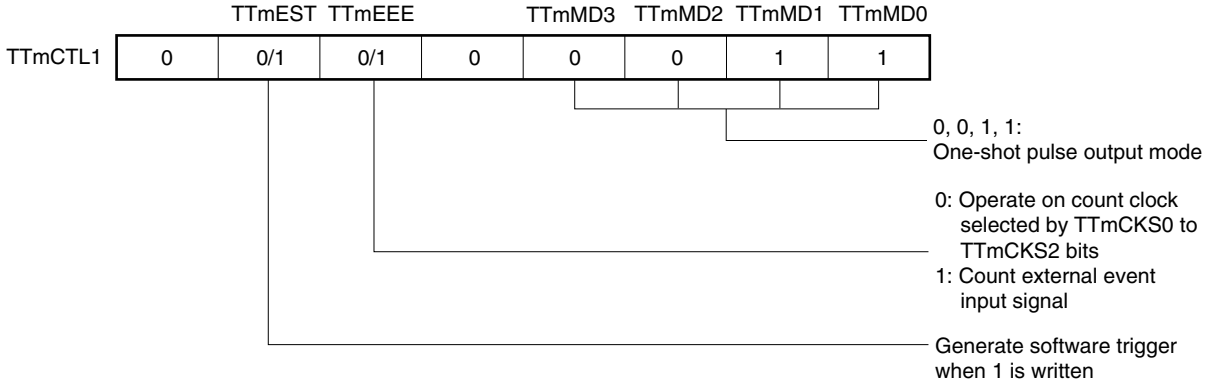
Figure 8-27. Setting of Registers in One-Shot Pulse Output Mode (1/2)

(a) TMTm control register 0 (TTmCTL0)



Note The setting is invalid when the TTmCTL1.TTmEEE bit = 1.

(b) TMTm control register 1 (TTmCTL1)



(c) TMTm I/O control register 0 (TTmIOC0)

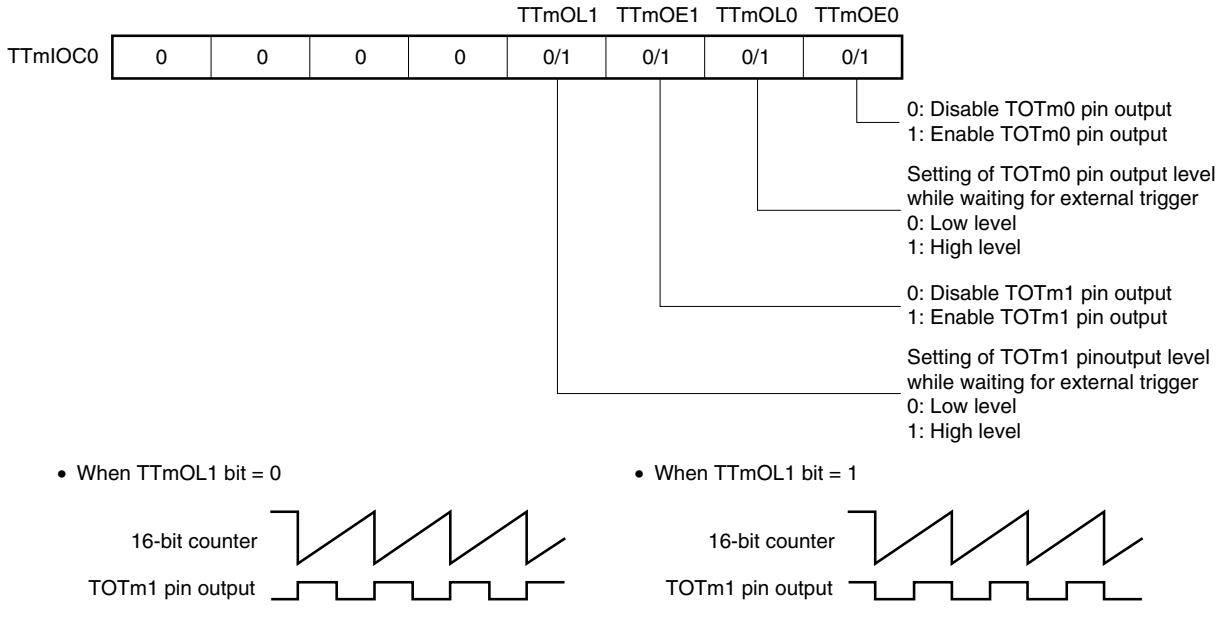
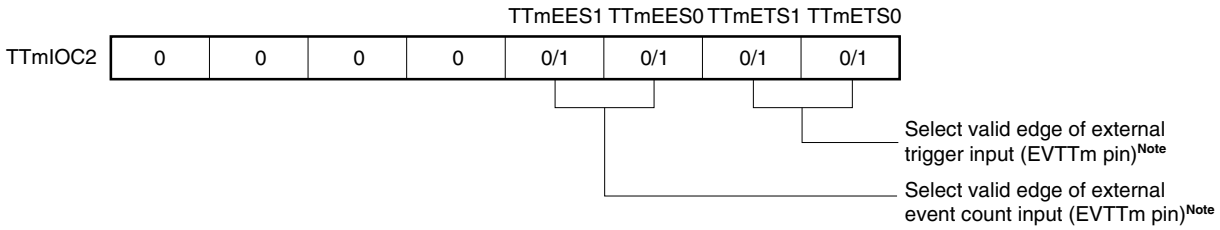


Figure 8-27. Setting of Registers in One-Shot Pulse Output Mode (2/2)

(d) TMTm I/O control register 2 (TTmIOC2)



Note Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

(e) TMTm counter read buffer register (TTmCNT)

The value of the 16-bit counter can be read by reading the TTmCNT register.

(f) TMTm capture/compare registers 0 and 1 (TTmCCR0 and TTmCCR1)

If D_0 is set to the TTmCCR0 register and D_1 to the TTmCCR1 register, the active level width and output delay period of the one-shot pulse are as follows.

Active level width = $(D_0 - D_1 + 1) \times$ Count clock cycle

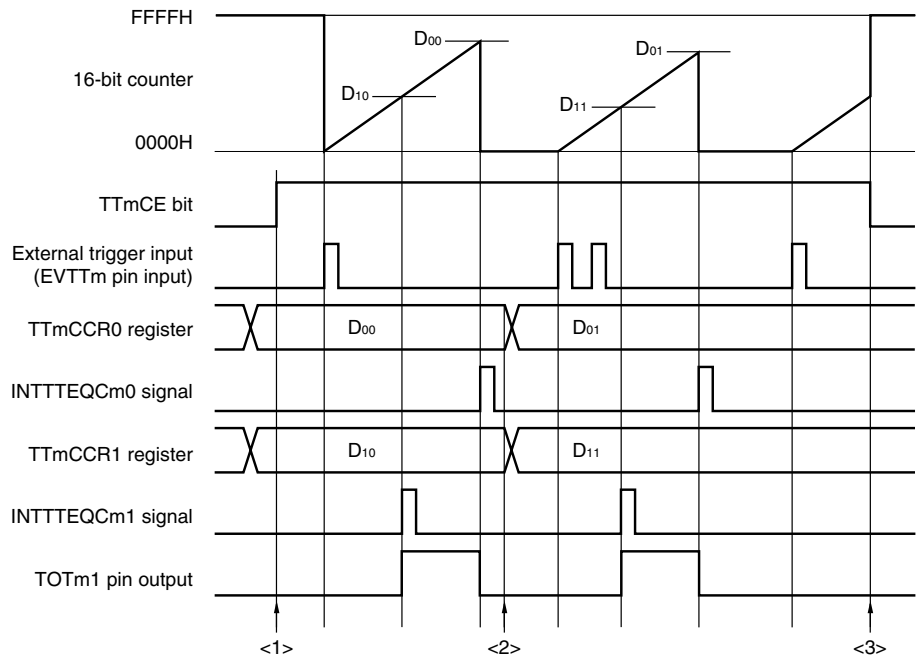
Output delay period = $D_1 \times$ Count clock cycle

Remarks 1. TMTm control register 2 (TTmCTL2), TMTm I/O control register 1 (TTmIOC1), TMTm I/O control register 3 (TTmIOC3), TMTm option register 0 (TTmOPT0), TMTm option register 1 (TTmOPT1), TMTm capture input select register (TTISLm), and TMTm counter write register (TTmTCW) are not used in the one-shot pulse output mode.

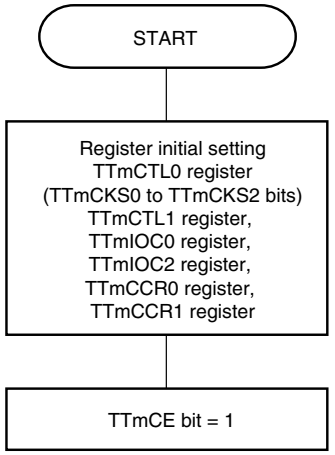
2. V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(1) Operation flow in one-shot pulse output mode

Figure 8-28. Software Processing Flow in One-Shot Pulse Output Mode



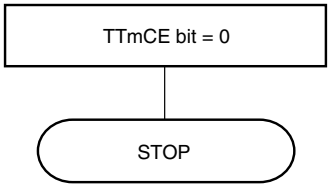
<1> Count operation start flow



Initial setting of these registers is performed before setting the TTmCE bit to 1.

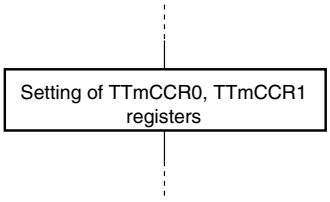
The TTmCKS0 to TTmCKS2 bits can be set at the same time as when counting starts (TTmCE bit = 1). Trigger wait status

<3> Count operation stop flow



Count operation is stopped

<2> TTmCCR0, TTmCCR1 register setting change flow



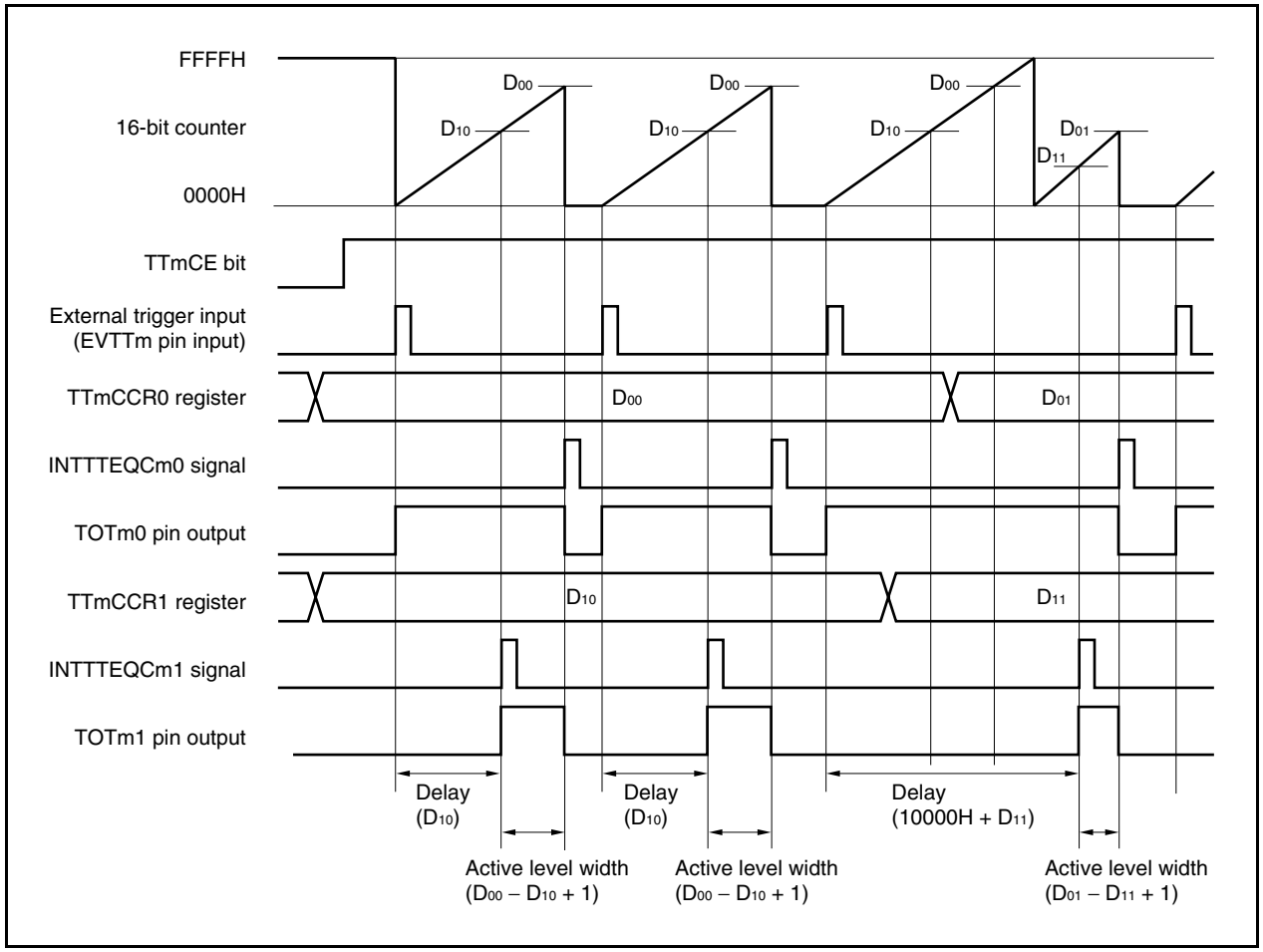
As rewriting the TTmCCRa register immediately forwards to the CCRa buffer register, rewriting immediately after the generation of the INTTTEQCm0 signal is recommended.

Remark V850E/IF3: m = 1, a = 0, 1
V850E/IG3: m = 0, 1, a = 0, 1

(2) Operation timing in one-shot pulse output mode

(a) Note on rewriting TTmCCRa register

If the value of the TTmCCRa register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When an overflow may occur, stop counting and then change the set value.



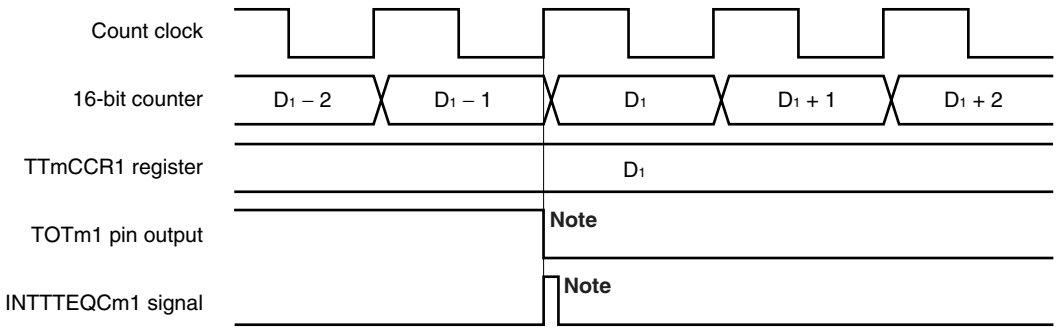
When the TTmCCR0 register is rewritten from D_{00} to D_{01} and the TTmCCR1 register from D_{10} to D_{11} where $D_{00} > D_{01}$ and $D_{10} > D_{11}$, if the TTmCCR1 register is rewritten when the count value of the 16-bit counter is greater than D_{11} and less than D_{10} and if the TTmCCR0 register is rewritten when the count value is greater than D_{01} and less than D_{00}, each set value is reflected as soon as the register has been rewritten and compared with the count value. The counter counts up to FFFFH and then counts up again from 0000H. When the count value matches D_{11}, the counter generates the INTTTEQCm1 signal and asserts the TOTm1 pin. When the count value matches D_{01}, the counter generates the INTTTEQCm0 signal, deasserts the TOTm1 pin, and stops counting.

Therefore, the counter may output a pulse with a delay period or active period different from that of the one-shot pulse that is originally expected.

Remark V850E/IF3: m = 1, a = 0, 1
V850E/IG3: m = 0, 1, a = 0, 1

(b) Generation timing of compare match interrupt request signal (INTTTEQCm1)

The generation timing of the INTTTEQCm1 signal in the one-shot pulse output mode is different from INTTTEQCm1 signals in other mode; the INTTTEQCm1 signal is generated when the count value of the 16-bit counter matches the value of the TTmCCR1 register.



Note The timing is actually delayed by one operating clock (fxx).

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Usually, the INTTTEQCm1 signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TTmCCR1 register.

In the one-shot pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the TOTm1 pin.

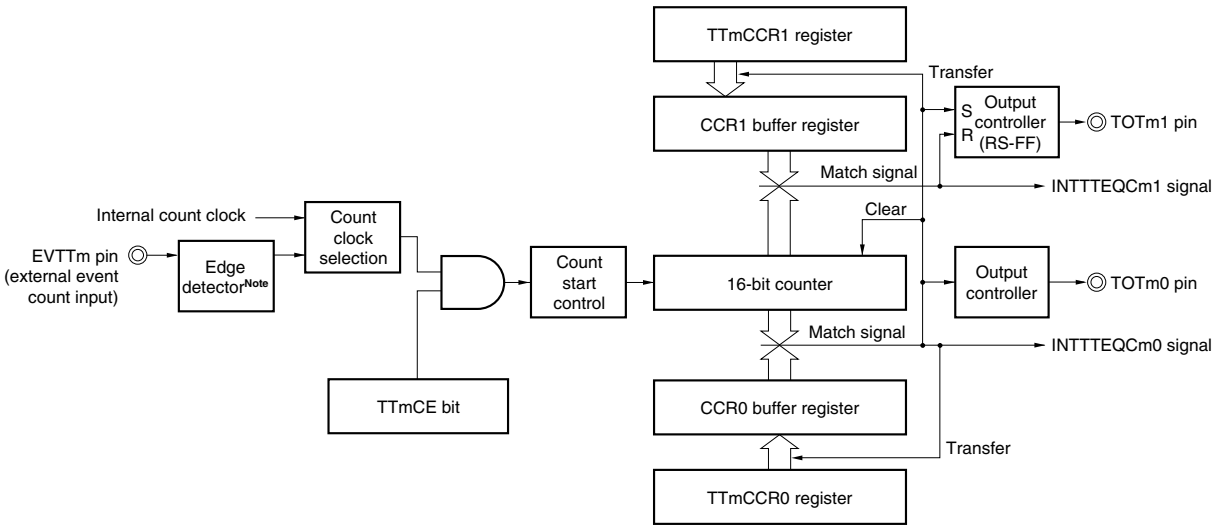
8.6.5 PWM output mode (TTmMD3 to TTmMD0 bits = 0100)

This mode is valid only in TMT0 (V850E/IG3 only) and TMT1.

In the PWM output mode, a PWM waveform is output from the TOTm1 pin when the TTmCTL0.TTmCE bit is set to 1.

In addition, a PWM waveform with a duty factor of 50% with the set value of the TTmCCR0 register + 1 as half its cycle is output from the TOTm0 pin.

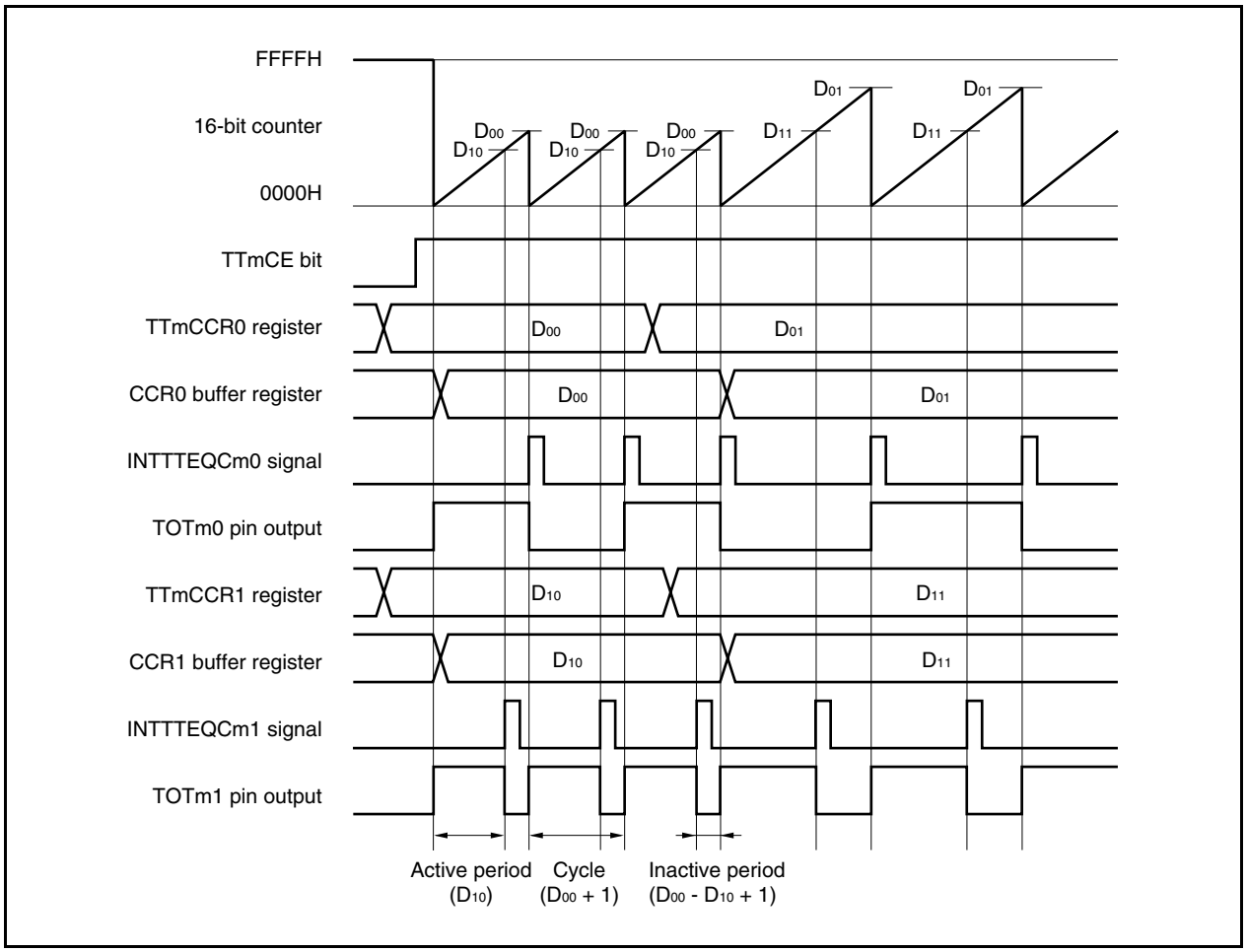
Figure 8-29. Configuration in PWM Output Mode



Note Set by the TTmIOC2.TTmEES1 and TTmIOC2.TTmEES0 bits.

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Figure 8-30. Basic Timing in PWM Output Mode



When the TTmCE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a PWM waveform from the TOTm1 pin.

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TTmCCR1 register) × Count clock cycle
Cycle = (Set value of TTmCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TTmCCR1 register)/(Set value of TTmCCR0 register + 1)

The PWM waveform can be changed by rewriting the TTmCCRa register while the counter is operating. The newly written value is reflected when the count value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

The compare match interrupt request signal (INTTTEQCm0) is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal (INTTTEQCm1) is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TTmCCRa register is transferred to the CCRa buffer register when the count value of the 16-bit counter matches the value of the CCRa buffer register and the 16-bit counter is cleared to 0000H.

Remark V850E/IF3: m = 1, a = 0, 1, V850E/IG3: m = 0, 1, a = 0, 1

Figure 8-31. Setting of Registers in PWM Output Mode (1/2)

(a) TMTm control register 0 (TTmCTL0)

	TTmCE					TTmCKS2	TTmCKS1	TTmCKS0
TTmCTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock[Note]

0: Stop counting
1: Enable counting

Note The setting is invalid when the TTmCTL1.TTmEEE bit = 1.

(b) TMTm control register 1 (TTmCTL1)

		TTmEST	TTmEEE		TTmMD3	TTmMD2	TTmMD1	TTmMD0
TTmCTL1	0	0	0/1	0	0	1	0	0

0, 1, 0, 0:
PWM output mode

0: Operate on count clock selected by TTmCKS0 to TTmCKS2 bits
1: Count with external event count input signal

(c) TMTm I/O control register 0 (TTmIOC0)

					TTmOL1	TTmOE1	TTmOL0	TTmOE0
TTmIOC0	0	0	0	0	0/1	0/1	0/1	0/1

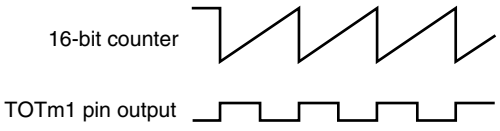
0: Disable TOTm0 pin output
1: Enable TOTm0 pin output

Setting of TOTm0 pin output level before count operation
0: Low level
1: High level

0: Disable TOTm1 pin output
1: Enable TOTm1 pin output

Setting of TOTm1 pin output level before count operation
0: Low level
1: High level

- When TTmOL1 bit = 0



- When TTmOL1 bit = 1

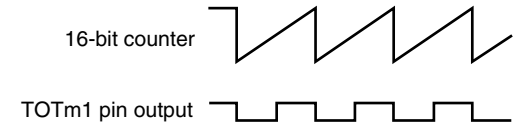
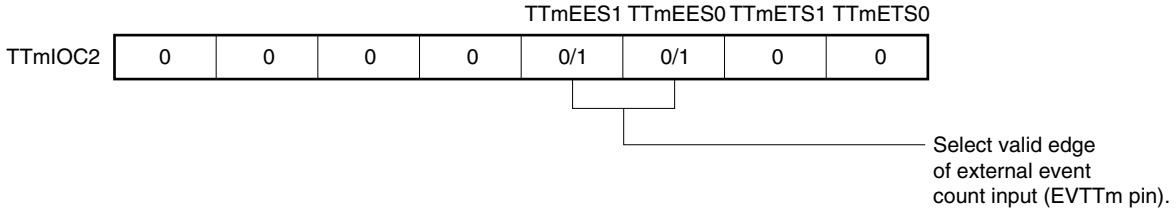


Figure 8-31. Register Setting in PWM Output Mode (2/2)

(d) TMTm I/O control register 2 (TTmIOC2)



(e) TMTm counter read buffer register (TTmCNT)

The value of the 16-bit counter can be read by reading the TTmCNT register.

(f) TMTm capture/compare registers 0 and 1 (TTmCCR0 and TTmCCR1)

If D_0 is set to the TTmCCR0 register and D_1 to the TTmCCR1 register, the cycle and active level of the PWM waveform are as follows.

Cycle = $(D_0 + 1) \times$ Count clock cycle

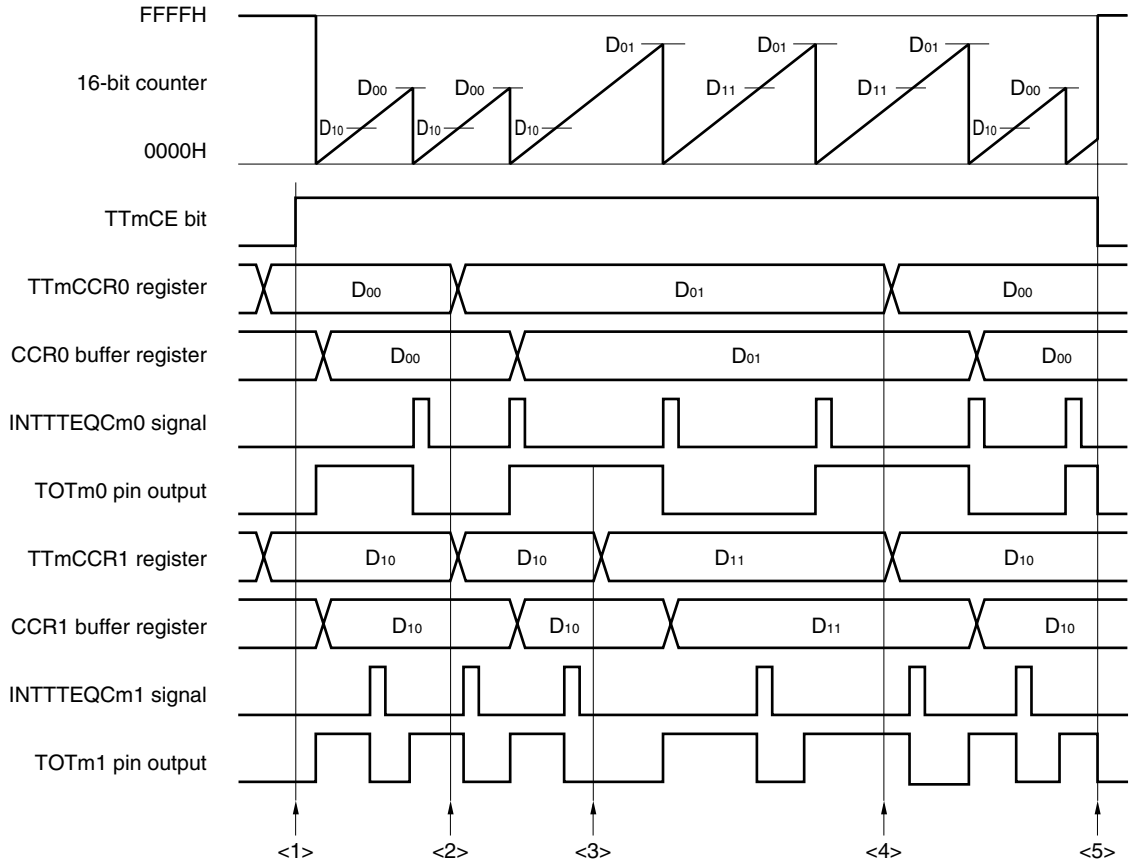
Active level width = $D_1 \times$ Count clock cycle

Remarks 1. TMTm control register 2 (TTmCTL2), TMTm I/O control register 1 (TTmIOC1), TMTm I/O control register 3 (TTmCTL3), TMTm option register 0 (TTmOPT0), TMTm option register 1 (TTmOPT1), TMTm capture input select register (TTISLm), and TMTm counter write register (TTmTCW) are not used in the PWM output mode.

2. V850E/IF3: m = 1
V850E/IG3: m = 0, 1

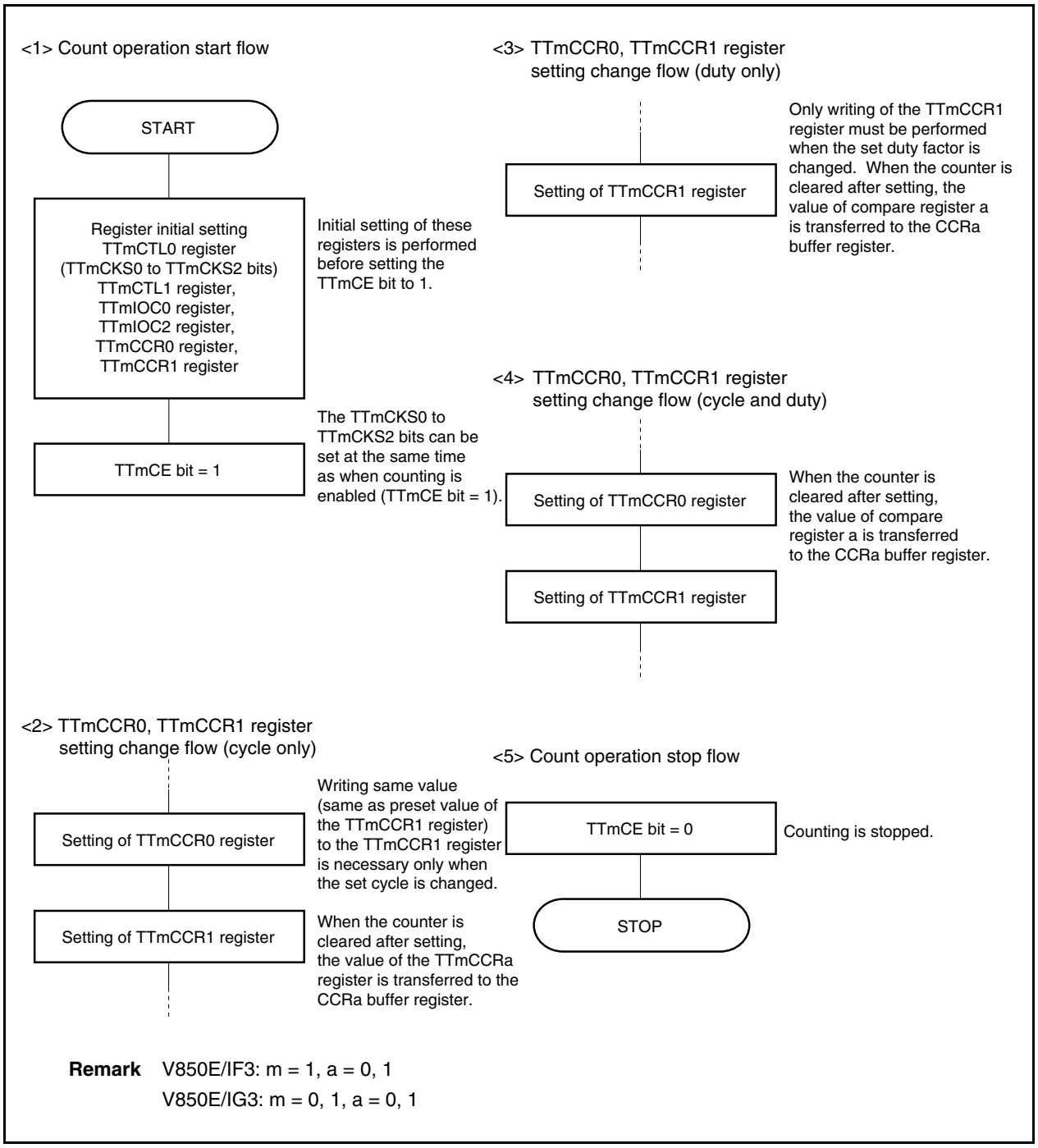
(1) Operation flow in PWM output mode

Figure 8-32. Software Processing Flow in PWM Output Mode (1/2)



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Figure 8-32. Software Processing Flow in PWM Output Mode (2/2)



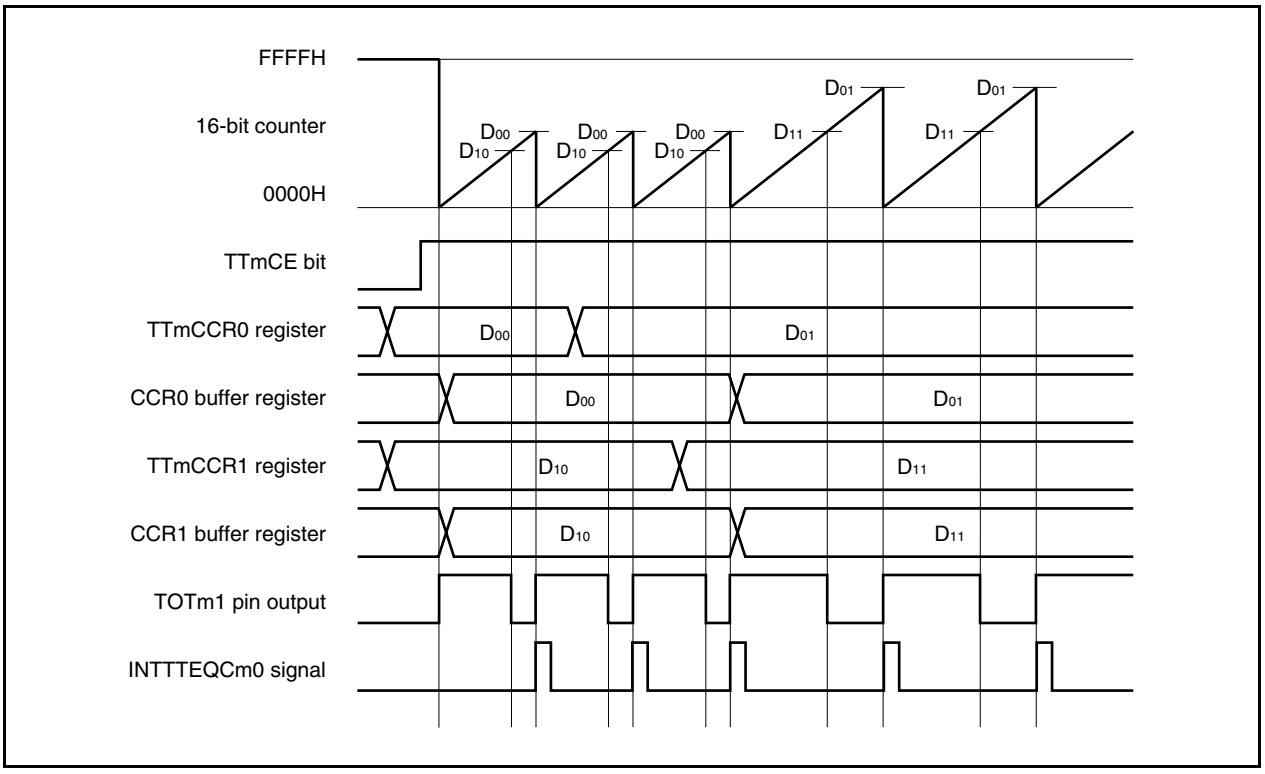
Remark V850E/IF3: m = 1, a = 0, 1
V850E/IG3: m = 0, 1, a = 0, 1

(2) PWM output mode operation timing

(a) Changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TTmCCR1 register last.

Rewrite the TTmCCRa register after writing the TTmCCR1 register after the INTTTEQCm1 signal is detected.



To transfer data from the TTmCCRa register to the CCRa buffer register, the TTmCCR1 register must be written.

To change both the cycle and active level of the PWM waveform at this time, first set the cycle to the TTmCCR0 register and then set the active level to the TTmCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TTmCCR0 register, and then write the same value (same as preset value of the TTmCCR1 register) to the TTmCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TTmCCR1 register has to be set.

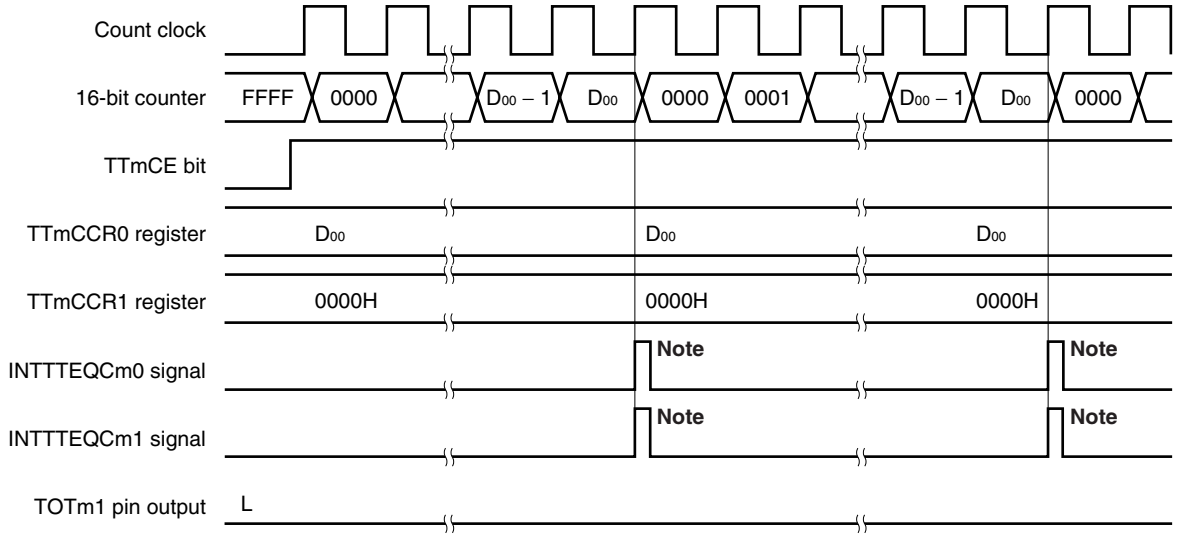
After data is written to the TTmCCR1 register, the value written to the TTmCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TTmCCR0 or TTmCCR1 register again after writing the TTmCCR1 register once, do so after the INTTTEQCm0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TTmCCRa register to the CCRa buffer register conflicts with writing the TTmCCRa register.

Remark V850E/IF3: m = 1, a = 0, 1, V850E/IG3: m = 0, 1, a = 0, 1

(b) 0%/100% output of PWM waveform

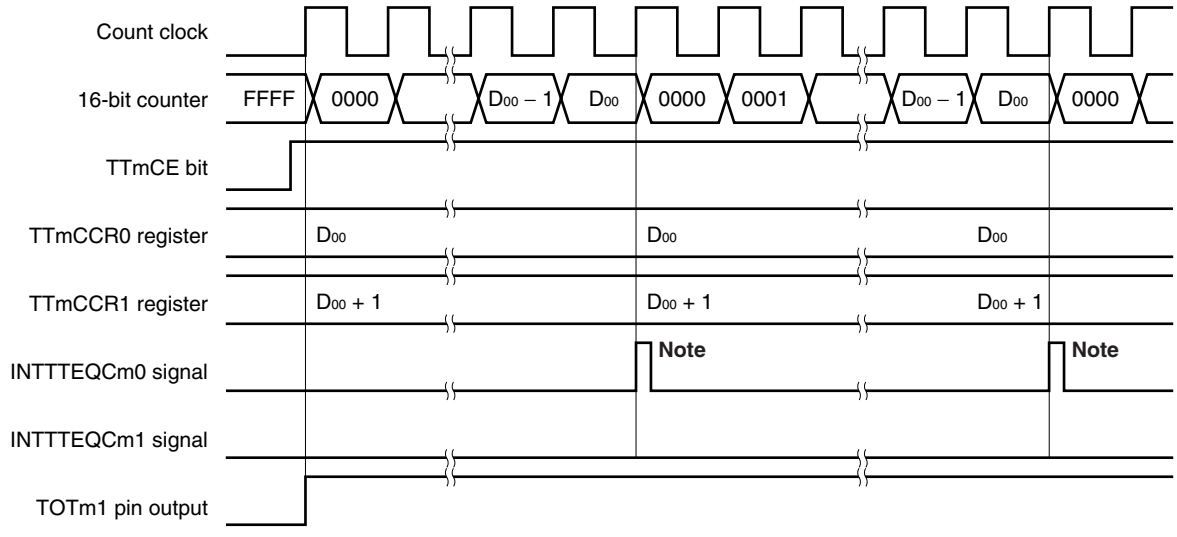
To output a 0% waveform, set the TTmCCR1 register to 0000H. The 16-bit counter is cleared to 0000H and the INTTTEQCm0 and INTTTEQCm1 signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.



Note The timing is actually delayed by one operating clock (fxx).

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

To output a 100% waveform, set a value of (set value of TTmCCR0 register + 1) to the TTmCCR1 register. If the set value of the TTmCCR0 register is FFFFH, 100% output cannot be produced.

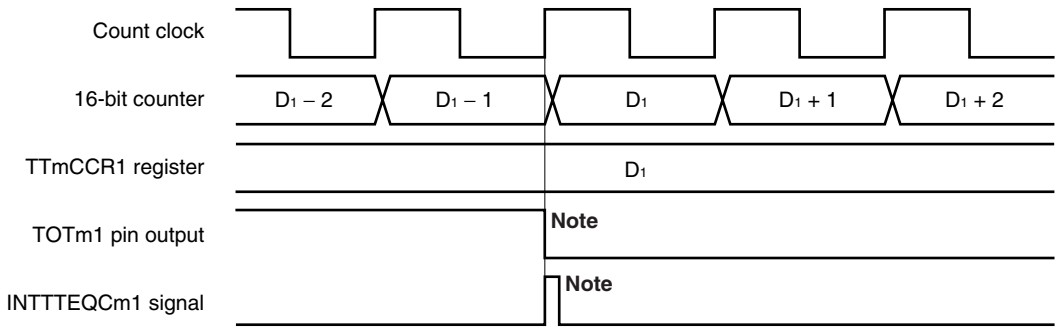


Note The timing is actually delayed by one operating clock (fxx).

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(c) Generation timing of compare match interrupt request signal (INTTTEQCm1)

The timing of generation of the INTTTEQCm1 signal in the PWM output mode differs from the timing of INTTTEQCm1 signals in other modes; the INTTTEQCm1 signal is generated when the count value of the 16-bit counter matches the value of the TTmCCR1 register.



Note The timing is actually delayed by one operating clock (fXX).

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Usually, the INTTTEQCm1 signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TTmCCR1 register.

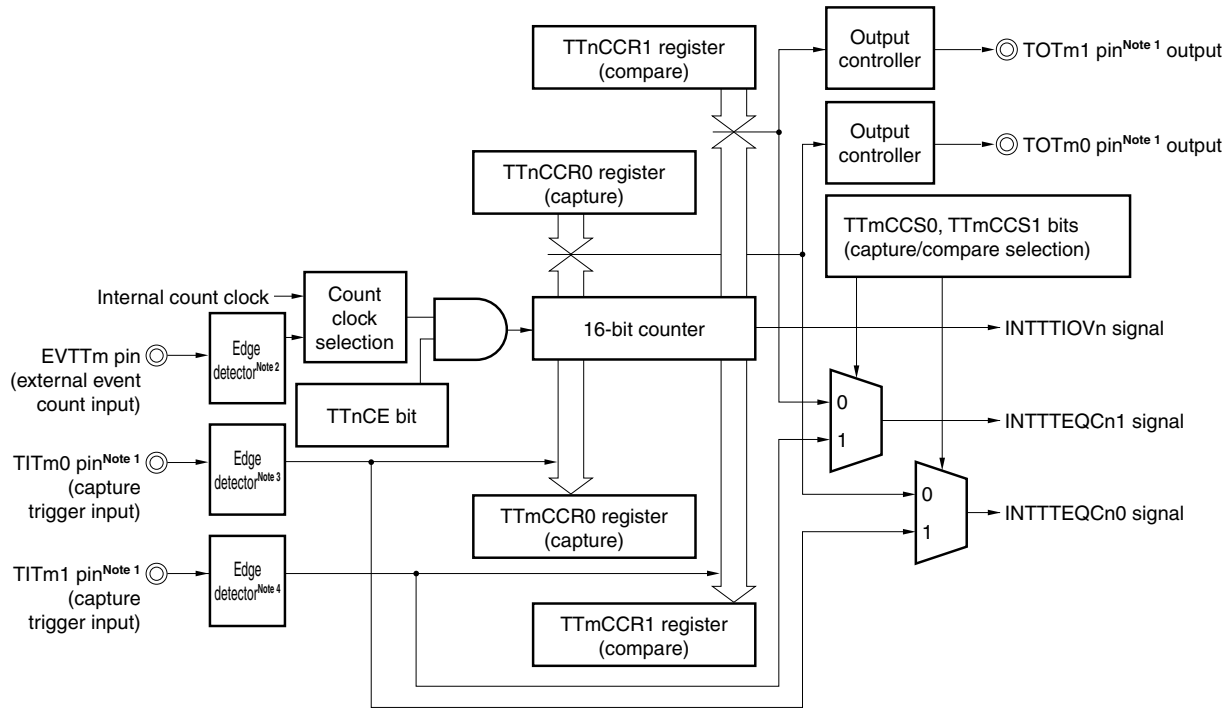
In the PWM output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the output signal of the TOTm1 pin.

8.6.6 Free-running timer mode (TTnMD3 to TTnMD0 bits = 0101)

The compare function is valid for all of TMT0 and TMT1. The capture function is valid only for TMT0 (V850E/IG3 only) and TMT1.

In the free-running timer mode, 16-bit timer/event counter T starts counting when the TTnCTL0.TTnCE bit is set to 1. At this time, the TTmCCR0 and TTmCCR1 registers can be used as compare registers or capture registers, depending on the setting of the TTmOPT0.TTmCCS0 and TTmOPT0.TTmCCS1 bits.

Figure 8-33. Configuration in Free-Running Timer Mode



Notes 1. Because the capture trigger input pins (TITm0, TITm1) and timer output pins (TOTm0, TOTm1) are the same alternate-function pins, two functions cannot be used at the same time.

2. Set by the TTmIOC2.TTmEES1 and TTmIOC2.TTmEES0 bits.

3. Set by the TTmIOC1.TTmIS1 and TTmIOC1.TTmIS0 bits.

4. Set by the TTmIOC1.TTmIS3 and TTmIOC1.TTmIS2 bits.

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

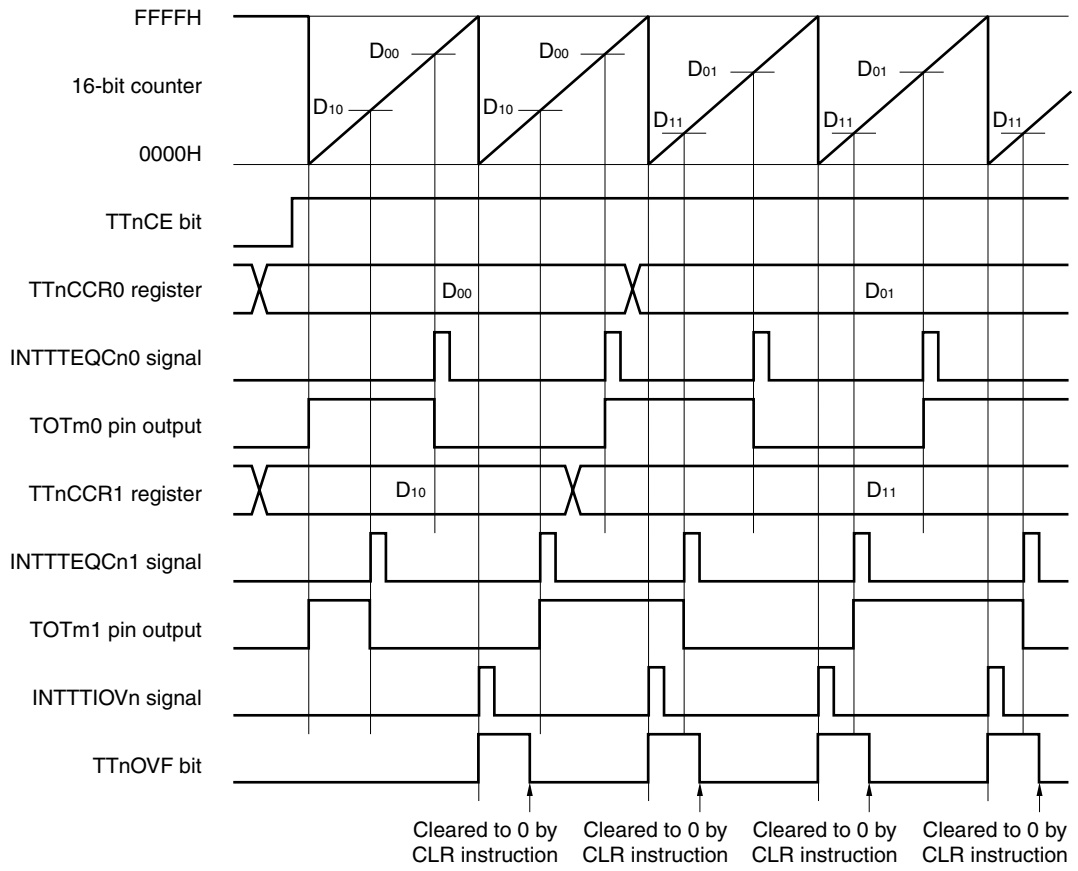
- Compare operation

  When the TTnCE bit is set to 1, 16-bit timer/event counter T starts counting, and the output signal of the TOTma pin is inverted. When the count value of the 16-bit counter later matches the set value of the TTnCCRa register, a compare match interrupt request signal (INTTTEQCna) is generated, and the output signal of the TOTna pin is inverted.

  The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTTIOVn) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TTnOPT0.TTnOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

  The TTnCCRa register can be rewritten while the counter is operating. If it is rewritten, the new value is reflected at that time by anytime write, and compared with the count value.

Figure 8-34. Basic Timing in Free-Running Timer Mode (Compare Function)



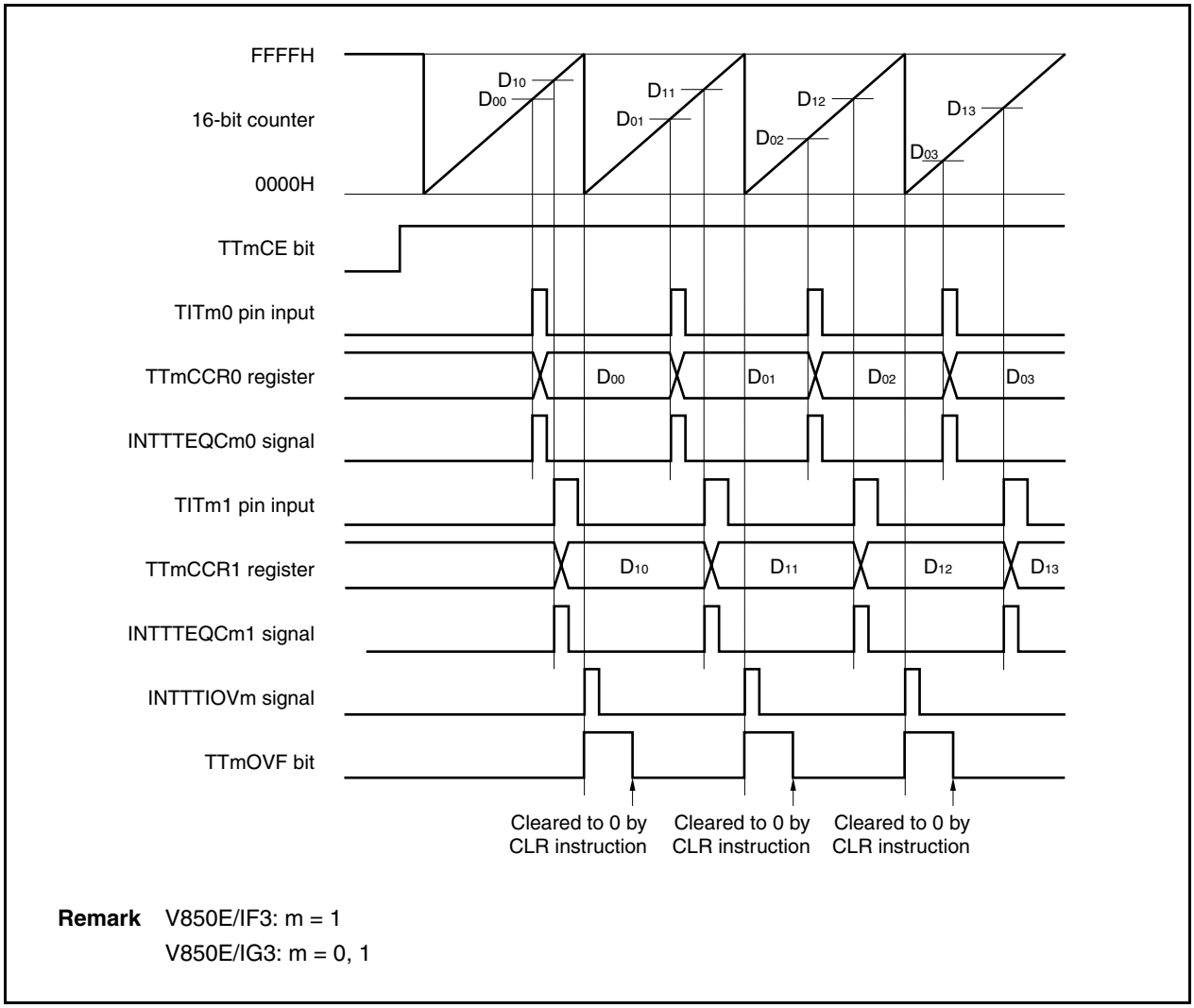
Remark V850E/IF3: n = 0, 1, m = 1, a = 0, 1
V850E/IG3: n = 0, 1, m = 0, 1, a = 0, 1

- Capture operation

  When the TTmCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TITma pin is detected, the count value of the 16-bit counter is stored in the TTmCCRa register, and a capture interrupt request signal (INTTTEQCma) is generated.

  The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTTIOVm) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TTmOPT0.TTmOVF bit) is also set to 1. Confirm that the overflow flag is set to 1 and then clear it to 0 by executing the CLR instruction via software.

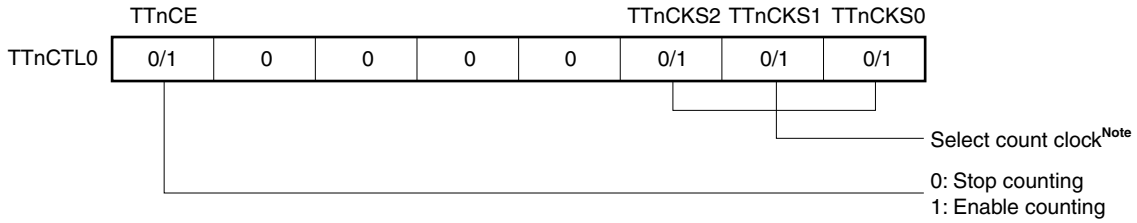
Figure 8-35. Basic Timing in Free-Running Timer Mode (Capture Function)



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

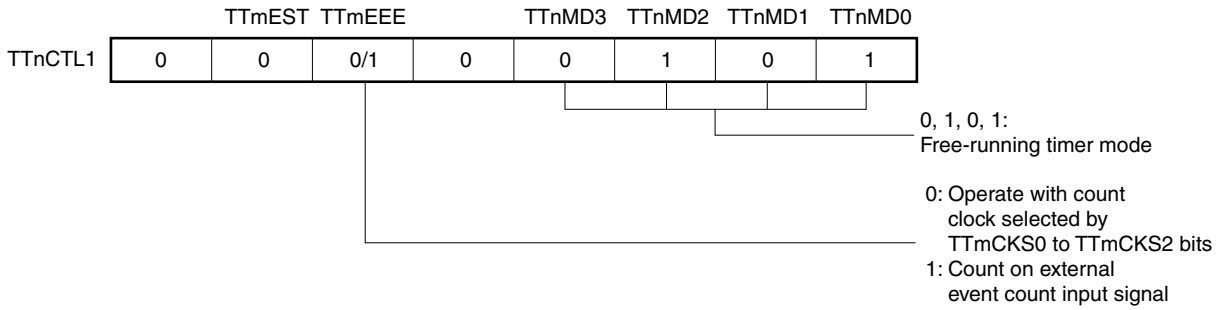
Figure 8-36. Register Setting in Free-Running Timer Mode (1/2)

(a) TMTn control register 0 (TTnCTL0)

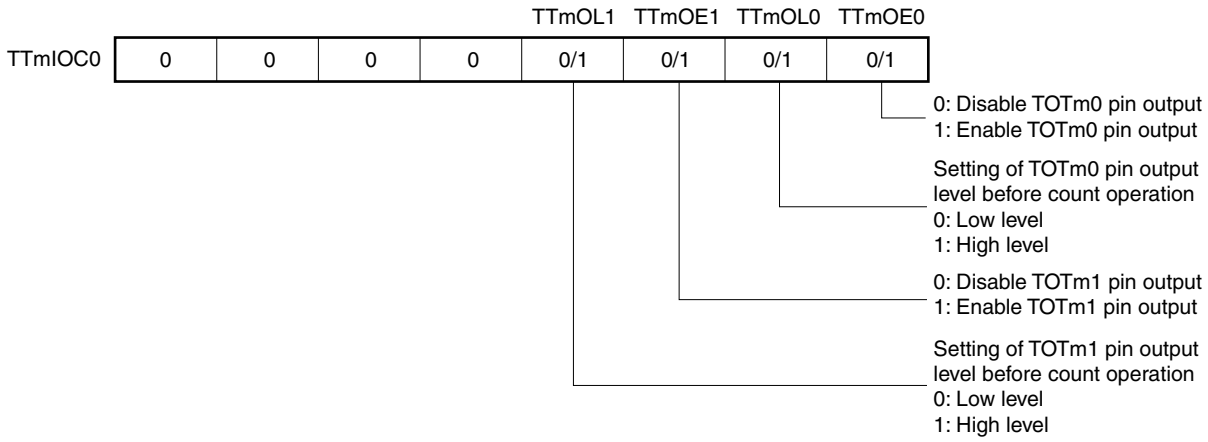


Note The setting is invalid when the TTmCTL1.TTmEEE bit = 1

(b) TMTn control register 1 (TTnCTL1)



(c) TMTm I/O control register 0 (TTmIOC0)



(d) TMTm I/O control register 1 (TTmIOC1)

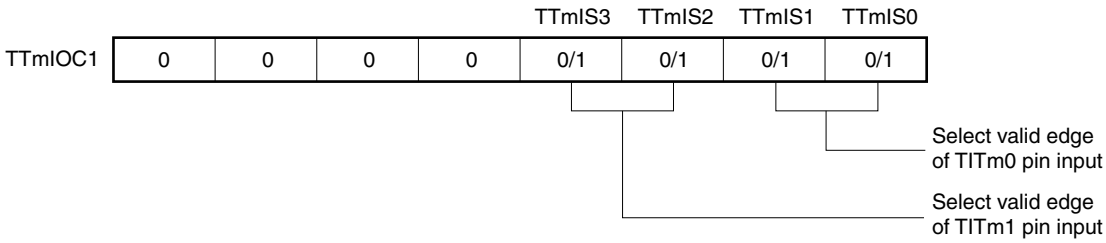
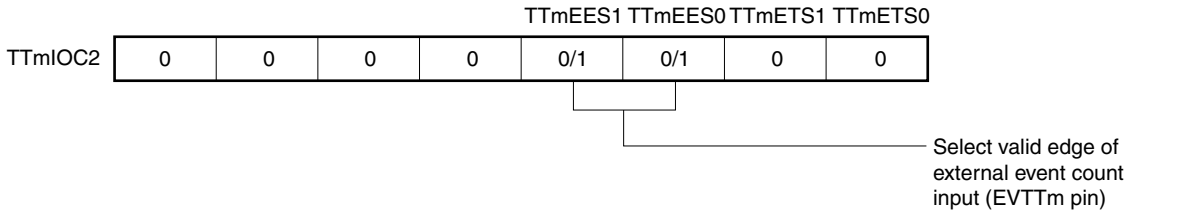
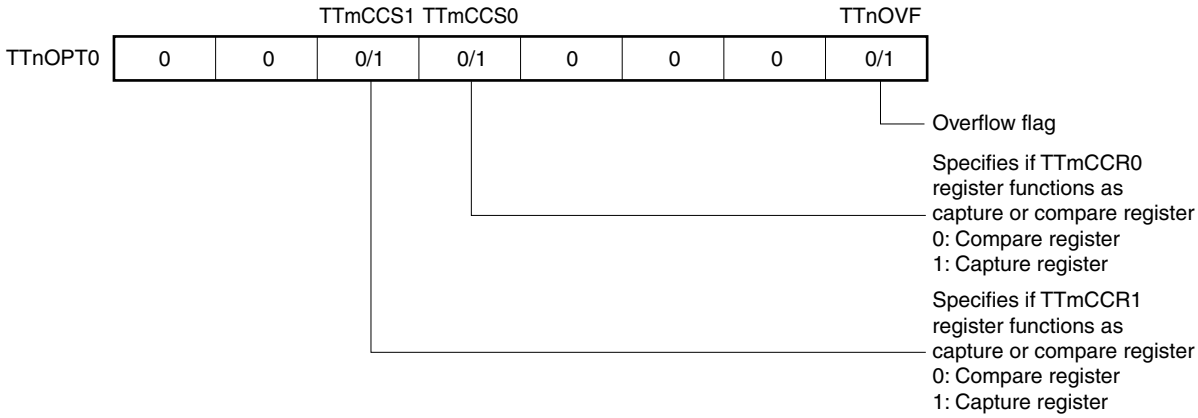


Figure 8-36. Register Setting in Free-Running Timer Mode (2/2)

(e) TMTm I/O control register 2 (TTmIOC2)



(f) TMTn option register 0 (TTnOPT0)



(g) TMTn counter read buffer register (TTnCNT)

The value of the 16-bit counter can be read by reading the TTnCNT register.

(h) TMTn capture/compare registers 0 and 1 (TTnCCR0 and TTnCCR1)

These registers function as capture registers or compare registers depending on the setting of the TTmOPT0.TTmCCSa bit.

When the registers function as capture registers, they store the count value of the 16-bit counter when the valid edge input to the TITma pin is detected.

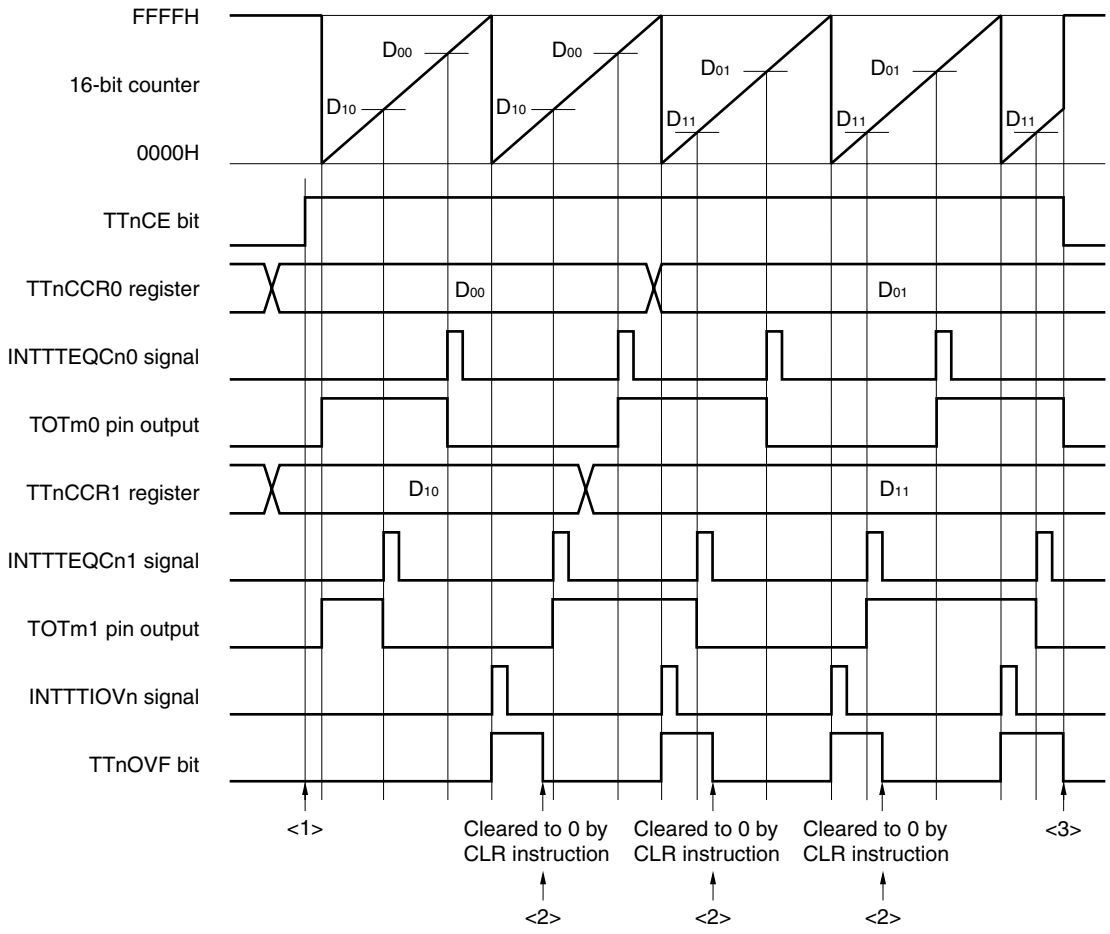
When the registers function as compare registers and when D_a is set to the TTnCCRa register, the INTTTEQCna signal is generated when the counter reaches (D_a + 1), and the output signals of the TOTm0 and TOTm1 pins are inverted.

Remark V850E/IF3: n = 0, 1, m = 1, a = 0, 1
V850E/IG3: n = 0, 1, m = 0, 1, a = 0, 1

(1) Operation flow in free-running timer mode

(a) When using capture/compare register as compare register

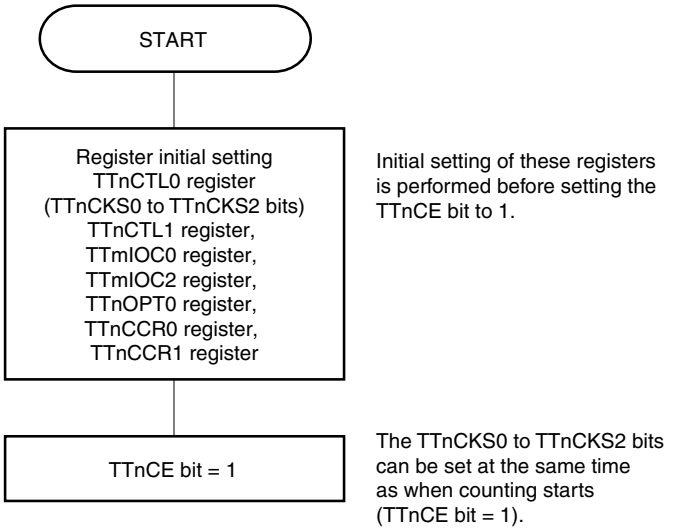
Figure 8-37. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)



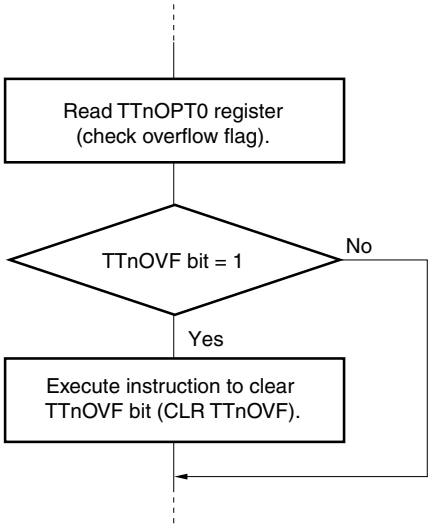
Remark V850E/IF3: n = 0, 1, m = 1
V850E/IG3: n = 0, 1, m = 0, 1

Figure 8-37. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)

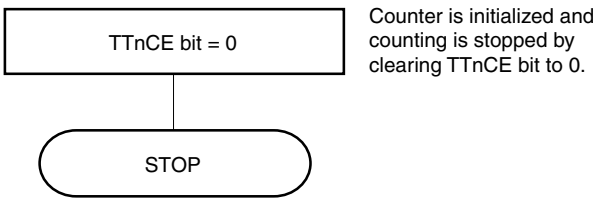
<1> Count operation start flow



<2> Overflow flag clear flow



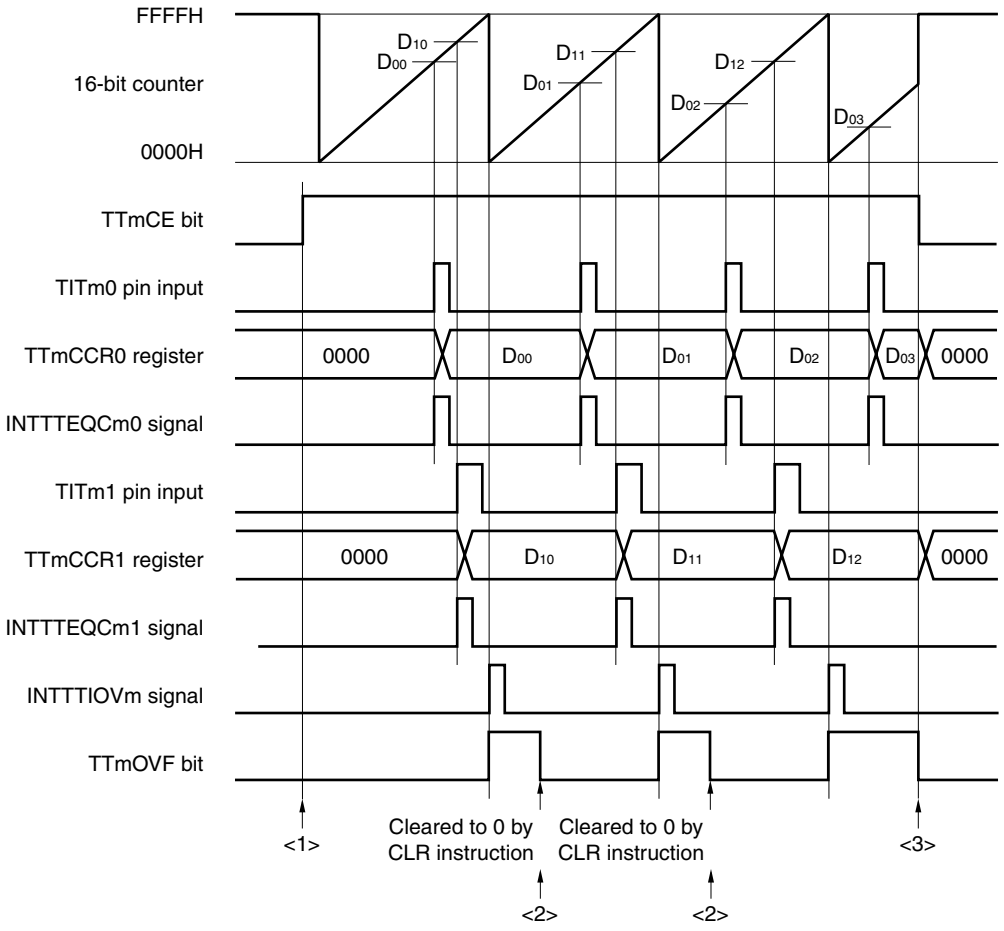
<3> Count operation stop flow



Remark V850E/IF3: n = 0, 1, m = 1
V850E/IG3: n = 0, 1, m = 0, 1

(b) When using capture/compare register as capture register

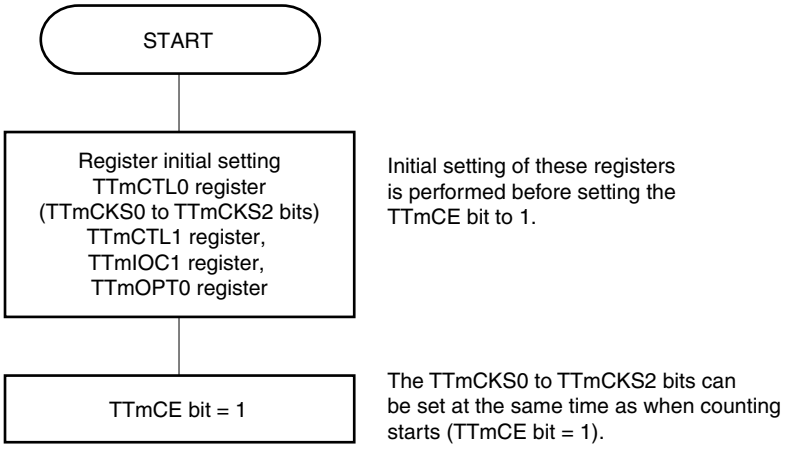
Figure 8-38. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)



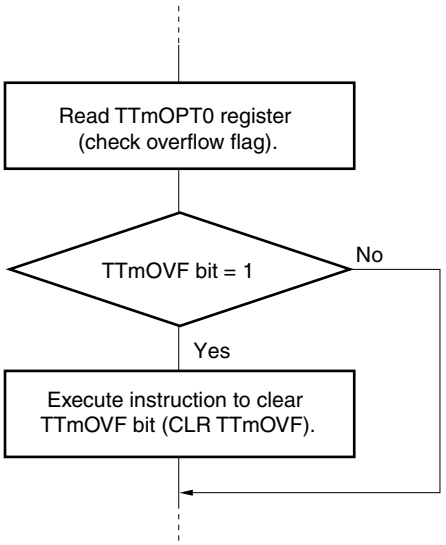
Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Figure 8-38. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)

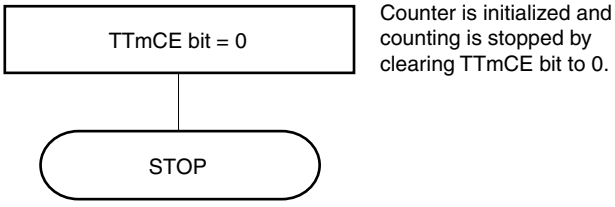
<1> Count operation start flow



<2> Overflow flag clear flow



<3> Count operation stop flow

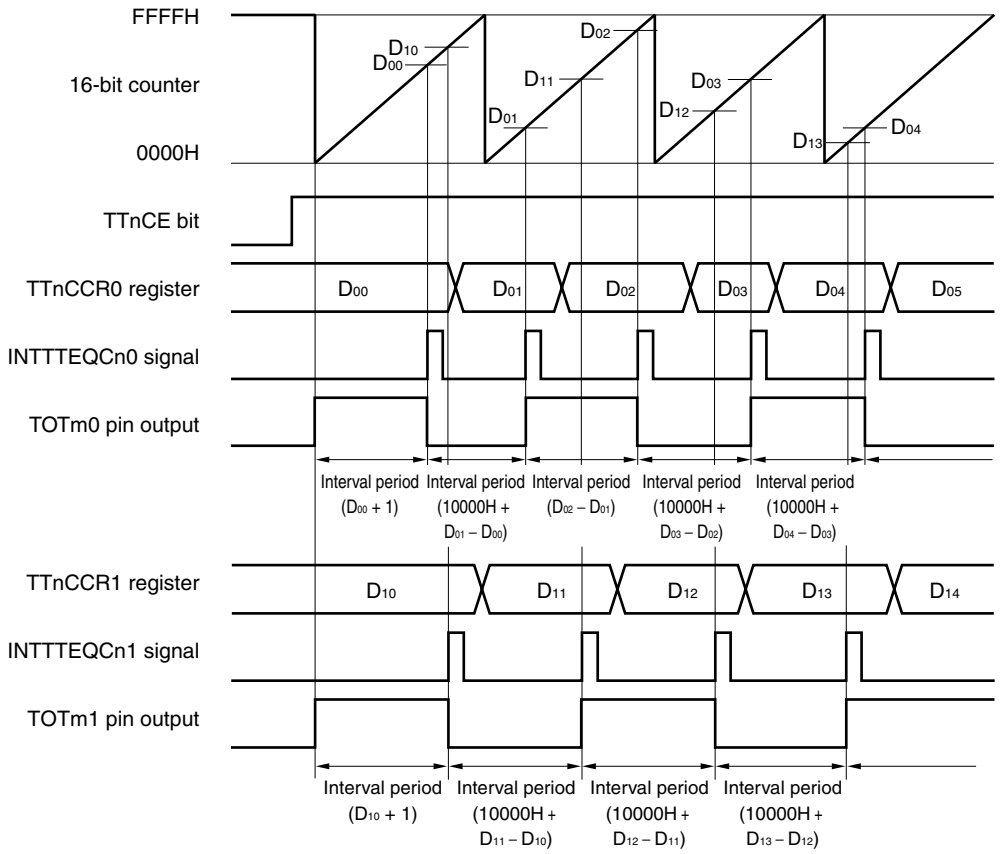


Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(2) Operation timing in free-running timer mode

(a) Interval operation with compare register

When 16-bit timer/event counter T is used as an interval timer with the TTnCCRa register used as a compare register, software processing is necessary for setting a comparison value to generate the next interrupt request signal each time the INTTTEQCna signal has been detected.



When performing an interval operation in the free-running timer mode, two intervals can be set with one channel.

To perform the interval operation, the value of the corresponding TTnCCRa register must be re-set in the interrupt servicing that is executed when the INTTTEQCna signal is detected.

The set value for re-setting the TTnCCRa register can be calculated by the following expression, where "D_a" is the interval period.

Compare register default value: $D_a - 1$

Value set to compare register second and subsequent time: Previous set value + D_a

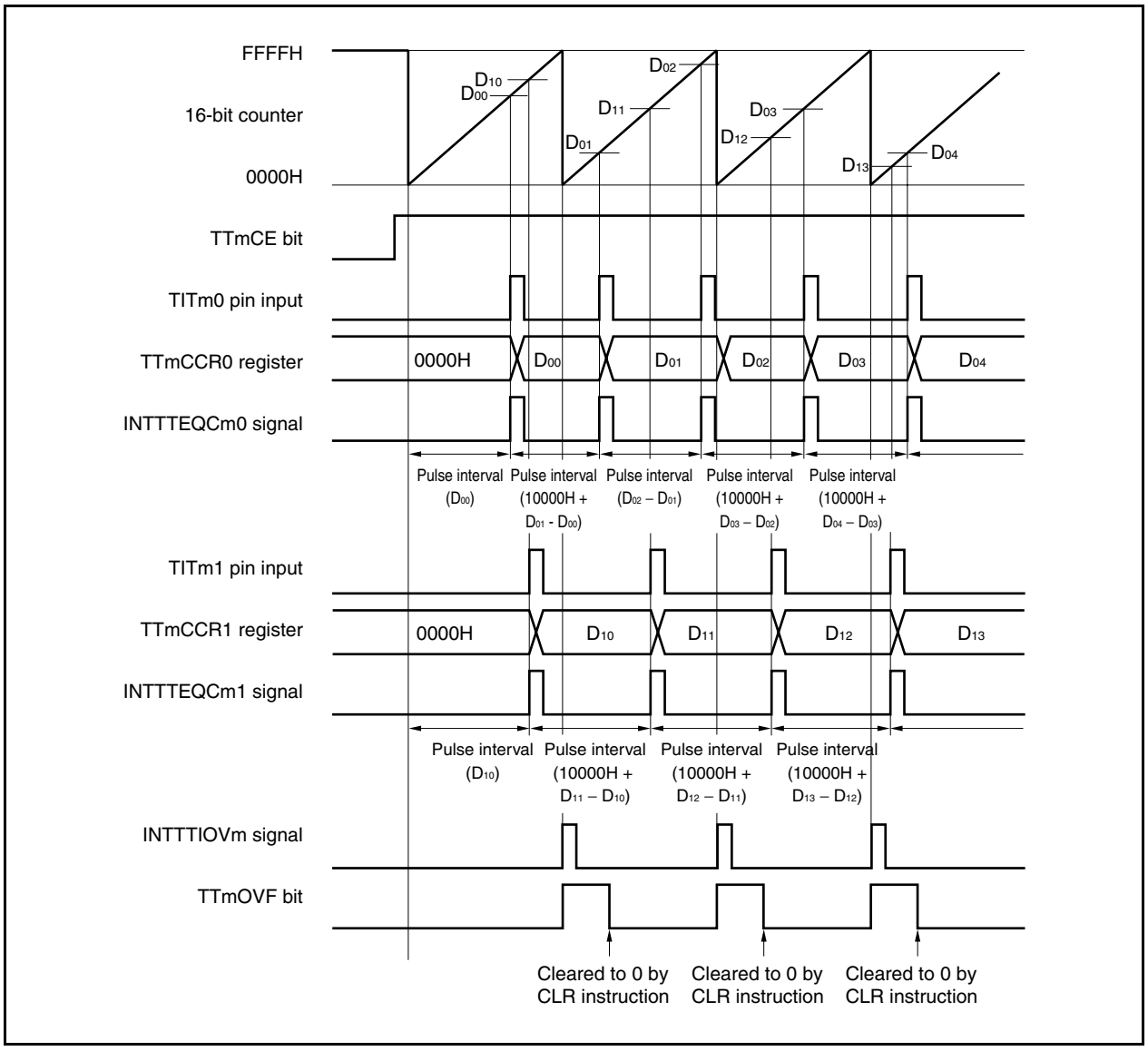
(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

Remark V850E/IF3: n = 0, 1, m = 1, a = 0, 1

V850E/IG3: n = 0, 1, m = 0, 1, a = 0, 1

(b) Pulse width measurement with capture register

When pulse width measurement is performed with the TTmCCRa register used as a capture register, software processing is necessary for reading the capture register each time the INTTTEQCma signal has been detected and for calculating an interval.



When executing pulse width measurement in the free-running timer mode, two pulse widths can be measured with one channel.

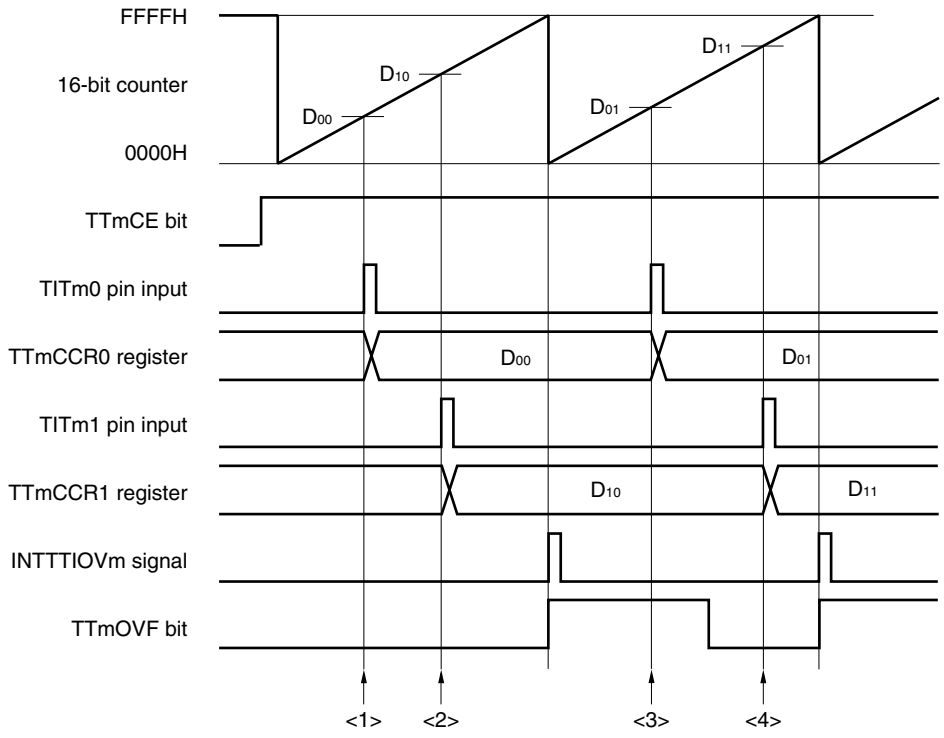
To measure a pulse width, the pulse width can be calculated by reading the value of the TTmCCRa register in synchronization with the INTTTEQCma signal, and calculating the difference between the read value and the previously read value.

Remark V850E/IF3: m = 1, a = 0, 1
V850E/IG3: m = 0, 1, a = 0, 1

(c) Processing of overflow when two capture registers are used

Care must be exercised in processing the overflow flag when two capture registers are used. First, an example of incorrect processing is shown below.

Example of incorrect processing when two capture registers are used



The following problem may occur when two pulse widths are measured in the free-running timer mode.

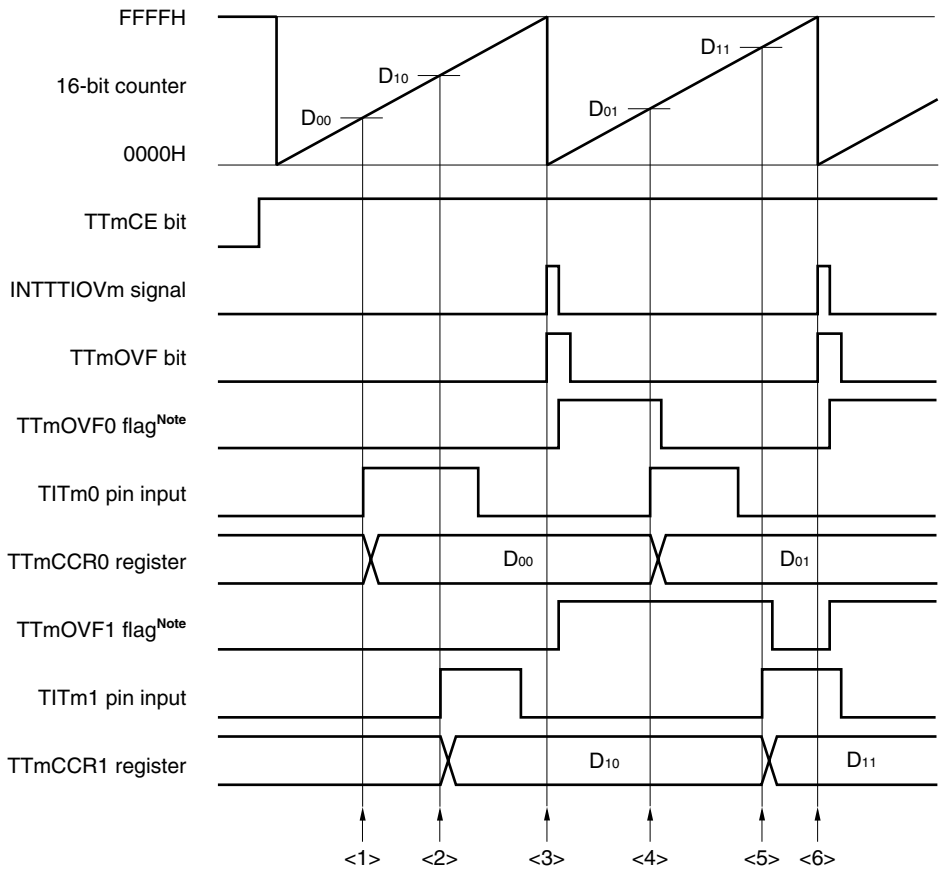
<1> Read the TTmCCR0 register (setting of the default value of the TITm0 pin input).
<2> Read the TTmCCR1 register (setting of the default value of the TITm1 pin input).
<3> Read the TTmCCR0 register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).
<4> Read the TTmCCR1 register.
Read the overflow flag. Because the flag is cleared in <3>, 0 is read.
Because the overflow flag is 0, the pulse width can be calculated by (D_{11} – D_{10}) (incorrect).

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

When two capture registers are used, and if the overflow flag is cleared to 0 by one capture register, the other capture register may not obtain the correct pulse width.
Use software when using two capture registers. An example of how to use software is shown below.

(1/2)

Example when two capture registers are used (using overflow interrupt)



Note The TTmOVF0 and TTmOVF1 flags are set on the internal RAM by software.

<1> Read the TTmCCR0 register (setting of the default value of the TITm0 pin input).

<2> Read the TTmCCR1 register (setting of the default value of the TITm1 pin input).

<3> An overflow occurs. Set the TTmOVF0 and TTmOVF1 flags to 1 in the overflow interrupt servicing, and clear the overflow flag to 0.

<4> Read the TTmCCR0 register.
Read the TTmOVF0 flag. If the TTmOVF0 flag is 1, clear it to 0.
Because the TTmOVF0 flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).

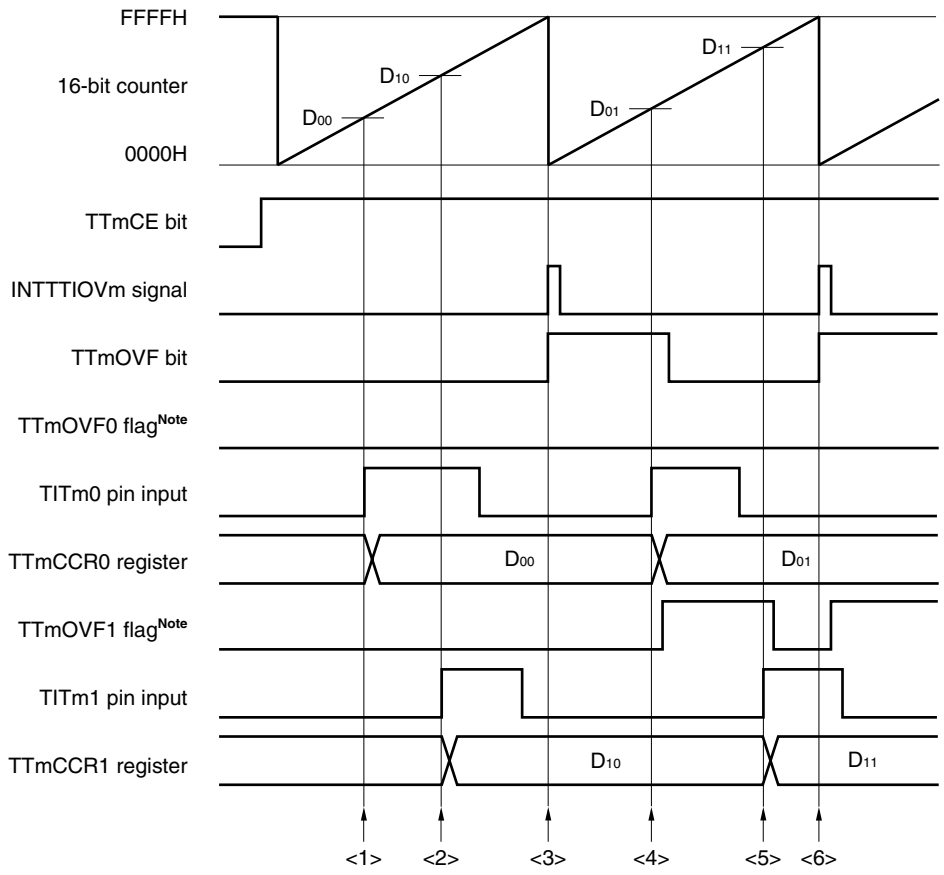
<5> Read the TTmCCR1 register.
Read the TTmOVF1 flag. If the TTmOVF1 flag is 1, clear it to 0 (the TTmOVF0 flag is cleared in <4>, and the TTmOVF1 flag remains 1).
Because the TTmOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} – D_{10}) (correct).

<6> Same as <3>

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(2/2)

Example when two capture registers are used (without using overflow interrupt)



Note The TTmOVF0 and TTmOVF1 flags are set on the internal RAM by software.

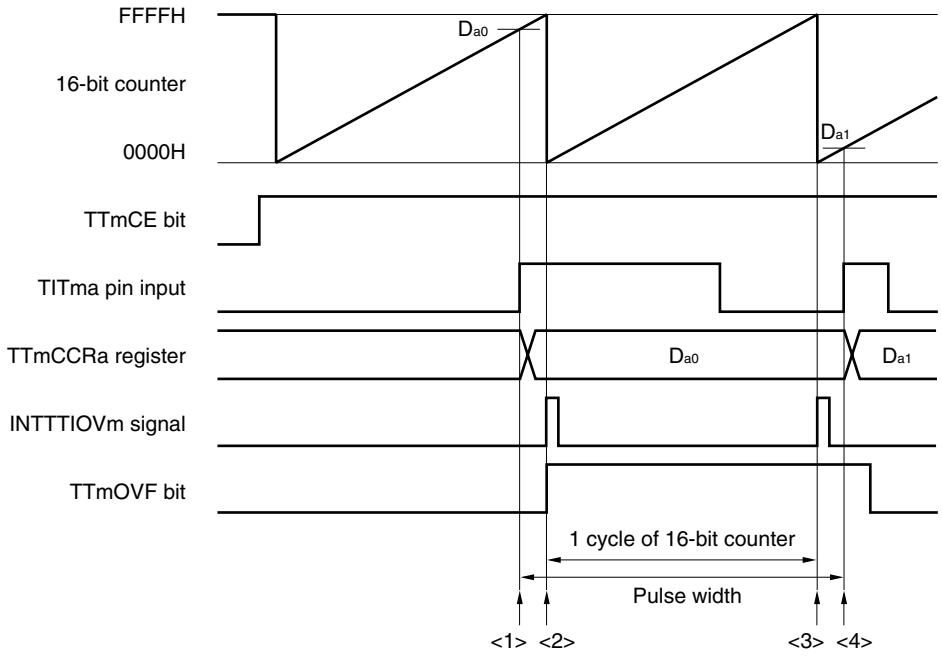
<1> Read the TTmCCR0 register (setting of the default value of the TITm0 pin input).
<2> Read the TTmCCR1 register (setting of the default value of the TITm1 pin input).
<3> An overflow occurs. Nothing is done by software.
<4> Read the TTmCCR0 register.
Read the overflow flag. If the overflow flag is 1, set only the TTmOVF1 flag to 1, and clear the overflow flag to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).
<5> Read the TTmCCR1 register.
Read the overflow flag. Because the overflow flag is cleared in <4>, 0 is read.
Read the TTmOVF1 flag. If the TTmOVF1 flag is 1, clear it to 0.
Because the TTmOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} – D_{10}) (correct).
<6> Same as <3>

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(d) Processing of overflow if capture trigger interval is long

If the pulse width is greater than one cycle of the 16-bit counter, care must be exercised because an overflow may occur more than once from the first capture trigger to the next. First, an example of incorrect processing is shown below.

Example of incorrect processing when capture trigger interval is long



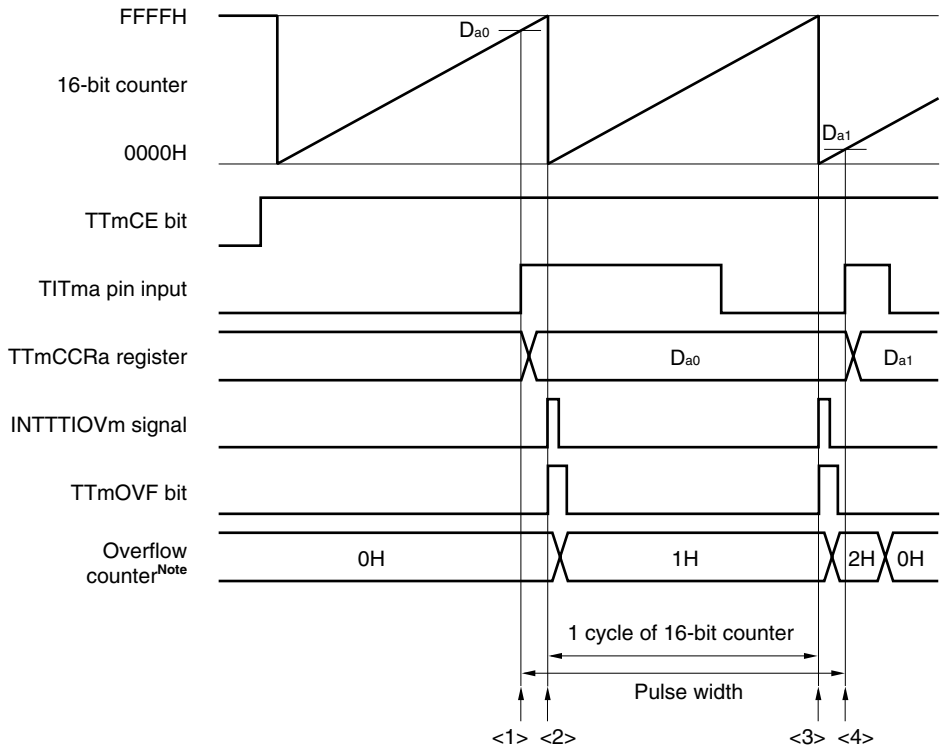
The following problem may occur when long pulse width is measured in the free-running timer mode.

<1> Read the TTmCCRa register (setting of the default value of the TITma pin input).
<2> An overflow occurs. Nothing is done by software.
<3> An overflow occurs a second time. Nothing is done by software.
<4> Read the TTmCCRa register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{a1} – D_{a0}) (incorrect).
Actually, the pulse width must be (20000H + D_{a1} – D_{a0}) because an overflow occurs twice.

Remark V850E/IF3: m = 1, a = 0, 1
V850E/IG3: m = 0, 1, a = 0, 1

If an overflow occurs twice or more when the capture trigger interval is long, the correct pulse width may not be obtained.
If the capture trigger interval is long, slow the count clock to lengthen one cycle of the 16-bit counter, or use software. An example of how to use software is shown next.

Example when capture trigger interval is long



Note The overflow counter is set arbitrarily by software on the internal RAM.

<1> Read the TTmCCRa register (setting of the default value of the TITma pin input).

<2> An overflow occurs. Increment the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.

<3> An overflow occurs a second time. Increment the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.

<4> Read the TTmCCRa register.

Read the overflow counter.

→ When the overflow counter is “N”, the pulse width can be calculated by ($N \times 10000H + D_{a1} - D_{a0}$).

In this example, the pulse width is ($20000H + D_{a1} - D_{a0}$) because an overflow occurs twice.

Clear the overflow counter (0H).

Remark V850E/IF3: m = 1, a = 0, 1
V850E/IG3: m = 0, 1, a = 0, 1

(e) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TTmOVF bit to 0 with the CLR instruction after reading the TTmOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TTmOPT0 register after reading the TTmOVF bit when it is 1.

8.6.7 Pulse width measurement mode (TTmMD3 to TTmMD0 bits = 0110)

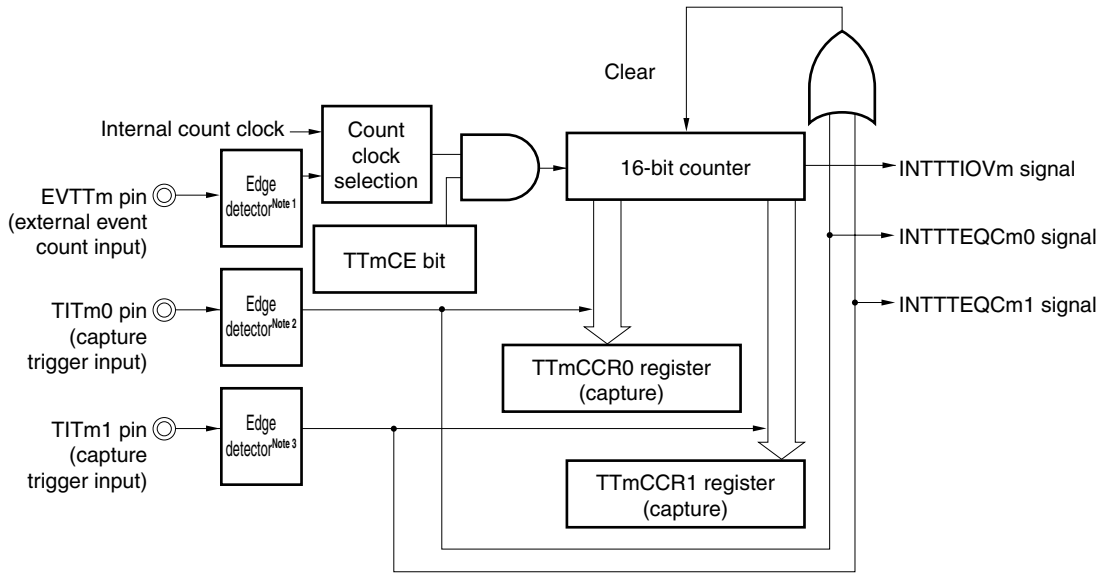
This mode is valid only in TMT0 (V850E/IG3 only) and TMT1.

In the pulse width measurement mode, 16-bit timer/event counter T starts counting when the TTmCTL0.TTmCE bit is set to 1. Each time the valid edge input to the TITma pin has been detected, the count value of the 16-bit counter is stored in the TTmCCRa register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TTmCCRa register after a capture interrupt request signal (INTTTEQCma) occurs.

As shown in Figure 8-40, select either the TITm0 or TITm1 pin as the capture trigger input pin and set the unused pins to “No edge detection” by using the TTmIOC1 register.

Figure 8-39. Configuration in Pulse Width Measurement Mode



Notes 1. Set by the TTmIOC2.TTmEES1 and TTmIOC2.TTmEES0 bits.

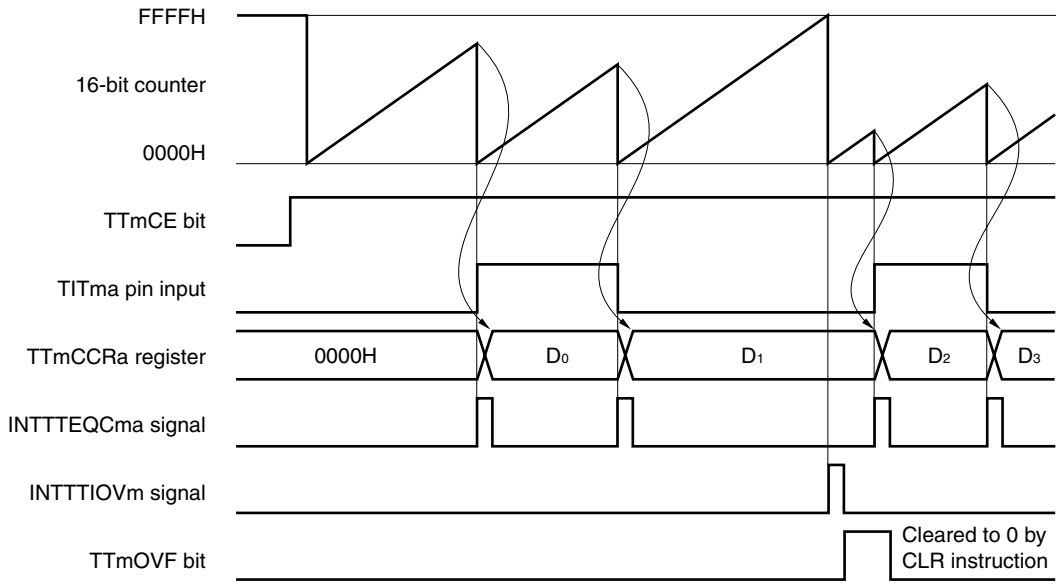
2. Set by the TTmIOC1.TTmIS1 and TTmIOC1.TTmIS0 bits.

3. Set by the TTmIOC1.TTmIS3 and TTmIOC1.TTmIS2 bits.

Remark V850E/IF3: m = 1, a = 0, 1

V850E/IG3: m = 0, 1, a = 0, 1

Figure 8-40. Basic Timing in Pulse Width Measurement Mode



Remark V850E/IF3: m = 1, a = 0, 1
V850E/IG3: m = 0, 1, a = 0, 1

When the TTmCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TITma pin is later detected, the count value of the 16-bit counter is stored in the TTmCCRa register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTTTEQCma) is generated.

The pulse width is calculated as follows.

Pulse width = Captured value × Count clock cycle

If the valid edge is not input to the TITma pin even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTTIOVm) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TTmOPT0.TTmOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction via software.

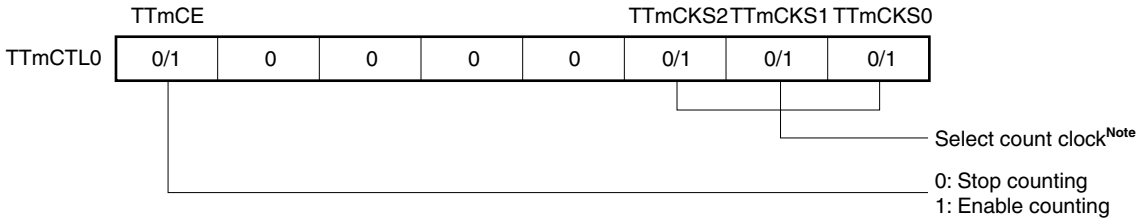
If the overflow flag is set to 1, the pulse width can be calculated as follows.

Pulse width = (10000H × TTmOVF bit set (1) count + Captured value) × Count clock cycle

Remark V850E/IF3: m = 1, a = 0, 1
V850E/IG3: m = 0, 1, a = 0, 1

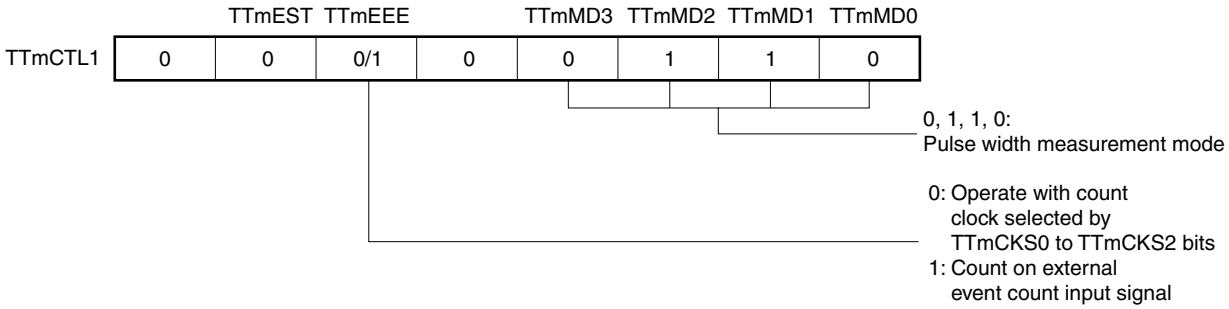
Figure 8-41. Register Setting in Pulse Width Measurement Mode (1/2)

(a) TMTm control register 0 (TTmCTL0)

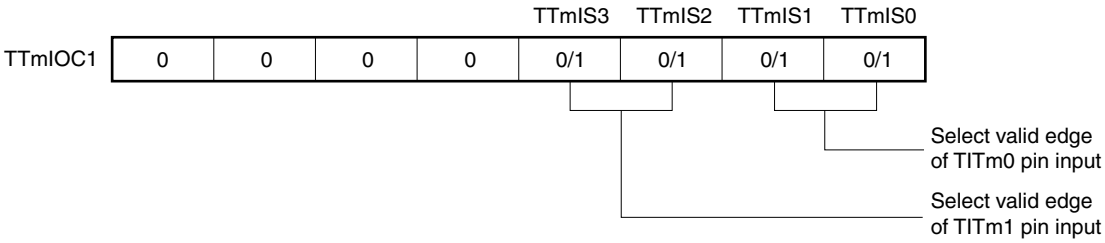


Note Setting is invalid when the TTmCTL1.TTmEEE bit = 1.

(b) TMTm control register 1 (TTmCTL1)



(c) TMTm I/O control register 1 (TTmIOC1)



(d) TMTm I/O control register 2 (TTmIOC2)

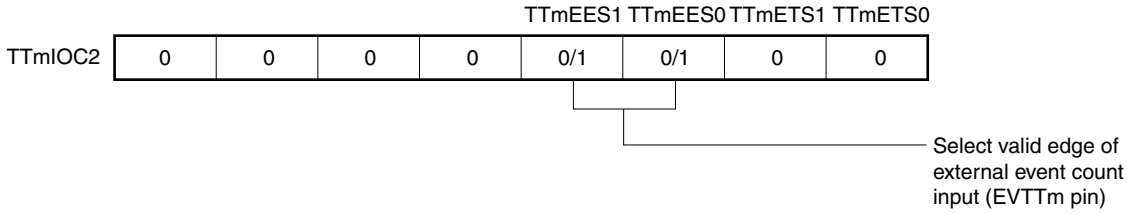


Figure 8-41. Register Setting in Pulse Width Measurement Mode (2/2)

(e) TMTm option register 0 (TTmOPT0)

				TTmCCS1	TTmCCS0			TTmOVF
TTmOPT0	0	0	0	0	0	0	0	0/1

Overflow flag

(f) TMTm counter read buffer register (TTmCNT)

The value of the 16-bit counter can be read by reading the TTmCNT register.

(g) TMTm capture/compare registers 0 and 1 (TTmCCR0 and TTmCCR1)

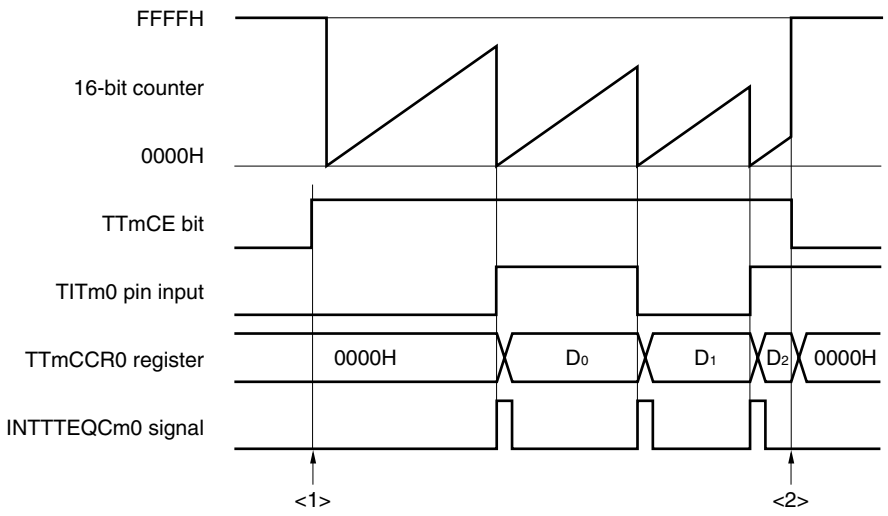
These registers store the count value of the 16-bit counter when the valid edge input to the TITm0 and TITm1 pins is detected.

Remarks 1. TMTm control register 2 (TTmCTL2), TMTm I/O control register 0 (TTmIOC0), TMTm I/O control register 3 (TTmIOC3), TMTm option register 1 (TTmOPT1), TMTm capture input select register (TTISLm), and TMTm counter write register (TTmTCW) are not used in the pulse width measurement mode.

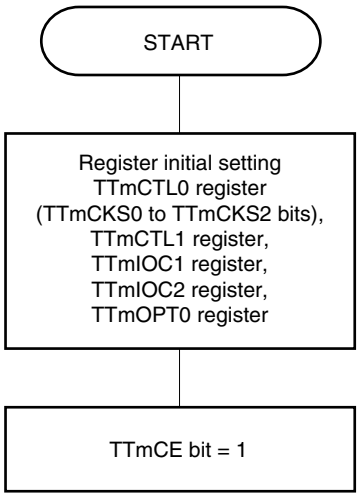
2. V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(1) Operation flow in pulse width measurement mode

Figure 8-42. Software Processing Flow in Pulse Width Measurement Mode



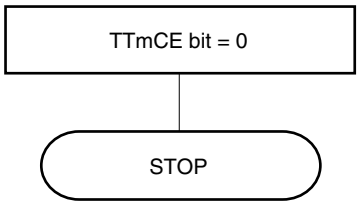
<1> Count operation start flow



Initial setting of these registers is performed before setting the TTmCE bit to 1.

The TTmCKS0 to TTmCKS2 bits can be set at the same time as when counting starts (TTmCE bit = 1).

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TTmCE bit to 0.

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(2) Operation timing in pulse width measurement mode

(a) Clearing overflow flag

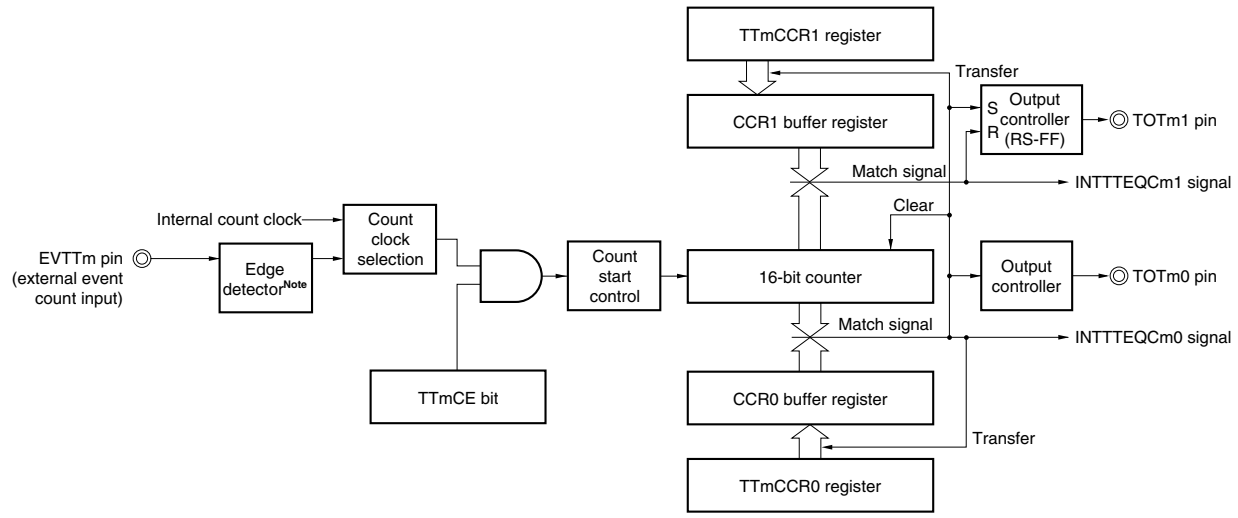
The overflow flag can be cleared to 0 by clearing the TTmOVF bit to 0 with the CLR instruction after reading the TTmOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TTmOPT0 register after reading the TTmOVF bit when it is 1.

8.6.8 Triangular-wave PWM output mode (TTmMD3 to TTmMD0 bits = 0111)

In the triangular-wave PWM output mode, a triangular-wave PWM waveform is output from the TOTm1 pin when the TTmCTL0.TTmCE bit is set to 1.

An inverted PWM waveform is output from the TOTm0 pin when the count value of the 16-bit counter matches the value of the CCR0 buffer register and when the 16-bit counter is set to 0000H.

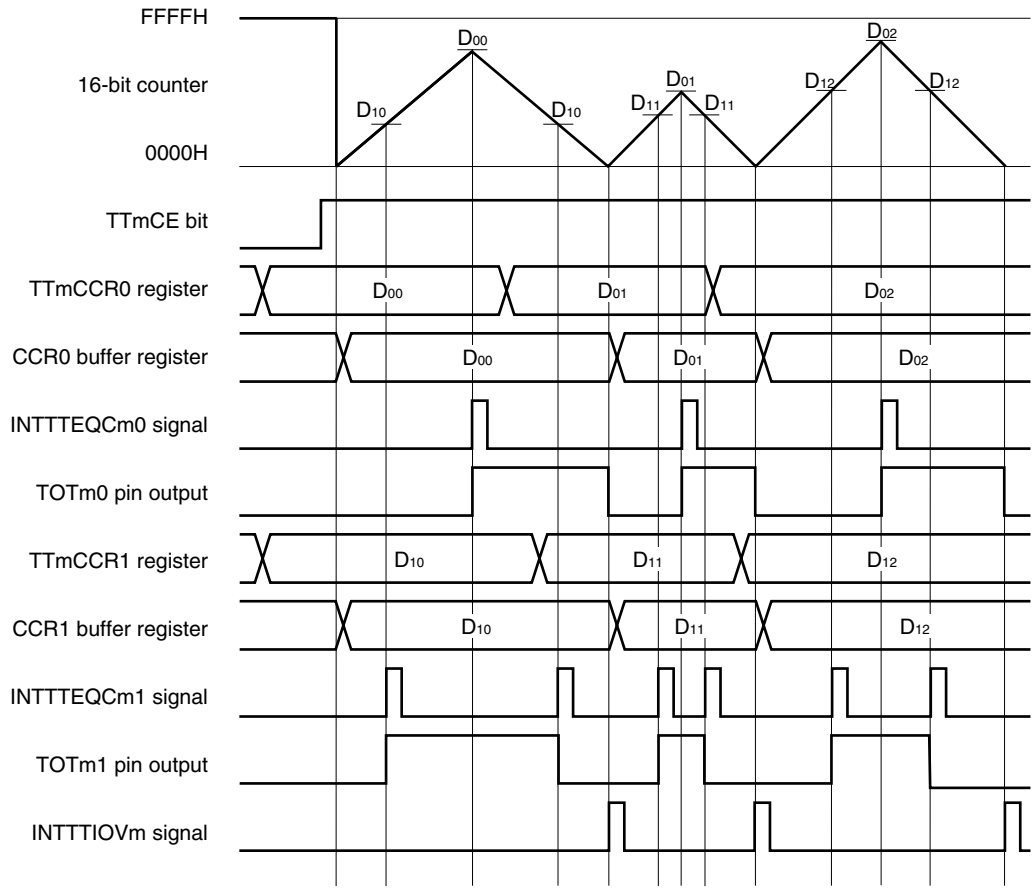
Figure 8-43. Configuration in Triangular-Wave PWM Output Mode



Note Set by the TTmIOC2.TTmEES1 and TTmIOC2.TTmEES0 bits.

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Figure 8-44. Basic Timing in Triangular-Wave PWM Output Mode



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

The 16-bit counter is cleared from FFFFH and 0000H and starts counting when the TTmCE bit is set to 1. The triangular PWM waveform is output from the TOTm1 pin.

In the triangular-wave PWM output mode, the counter counts up or down. When the 16-bit counter reaches 0000H while it is counting down, an overflow interrupt request signal (INTTTIOVm) is generated. At this time, the TTmOPT0.TTmOVF bit is not set to 1. If the count value of the 16-bit counter matches the value of the CCR0 buffer register while the counter is counting up, a compare match interrupt request signal (INTTTEQCm0) is generated.

The counting direction is changed from up to down when the value of the 16-bit counter matches that of the CCR0 buffer register, and from down to up when the counter is cleared to 0000H.

The PWM waveform can be changed by rewriting the TTmCCRa register during operation. To change the PWM waveform during operation, write the TTmCCR1 register last.

The cycle of the triangular PWM waveform is set by the TTmCCR0 register and its duty factor is set by the TTmCCR1 register. Set a value to the TTmCCR0 register in a range of "0 ≤ TTmCCR0 ≤ FFFEH". The rewritten value is reflected when the 16-bit counter reaches 0000H while it is counting down.

Even when changing only the cycle of the PWM waveform, first set a period to the TTmCCR0 register, and then write the same value (value same as that set to the TTmCCR1 register) to the TTmCCR1 register.

To transfer data from the TTmCCRa register to the CCRa buffer register, the data must be written to the TTmCCR1 register (a = 0, 1).

8.6.9 Encoder count function

The encoder count function includes an encoder compare mode (see **8.6.10 Encoder compare mode (TTmMD3 to TTmMD0 bits = 1000)**).

Mode	TTmCCR0 Register	TTmCCR1 Register
Encoder compare mode	Compare only	Compare only

(1) Count-up/-down control

Counting up or down by the 16-bit counter is controlled by the phase of input encoder signals (TENCm0 and TENCm1) and setting of the TTmCTL2.TTmUDS1 and TTmCTL2.TTmUDS0 bits.

When the encoder count function is used, the internal count clock and external event count input (EVTTm) cannot be used. Set the TTmCTL0.TTmCKS2 to TTmCTL0.TTmCKS0 bits to 000 and the TTmCTL1.TTmEEE bit to 0.

(2) Setting initial value of 16-bit counter

The initial count value set to the TTmTCW register when the TTmCTL2.TTmECC bit = 0 is transferred to the 16-bit counter immediately after the counter starts its operation (TTmCTL0.TTmCE bit = 0 → 1), and the counter starts the operation after it detects the valid edge of the encoder input signal (TENCm0 or TENCm1).

(3) Basic operation

The TTmCCRa register generates a compare match interrupt request signal (INTTTEQCma) when the count value of the 16-bit counter matches the value of the CCRa buffer register.

(4) Clear operation

The 16-bit counter is cleared when the following conditions are satisfied in the encoder compare mode.

- When the value of the 16-bit counter matches the value of the compare register (the TTmCTL2.TTmECM1 and TTmCTL2.TTmECM0 bits are set)
- When the edge of the encoder clear input signal (TECRm) is detected and cleared (the TTmECS1 and TTmECS0 bits are set when the TTmIOC3.TTmSCE bit = 0)
- When the clear level condition of the TENCm0, TENCm1, and TECRm pins is detected (the TTmZCL, TTmBCL, and TTmACL bits are set when the TTmSCE bit = 1)

Remark V850E/IF3: m = 1, a = 0, 1
V850E/IG3: m = 0, 1, a = 0, 1

(5) Controlling bits of TTmCTL2 register

The setting of the TTmCTL2 register in the encoder compare mode is shown below.

Table 8-10. Setting of TTmCTL2 Register

Mode	TTmUDS1, TTmUDS0 Bits (<1>)	TTmECM1 Bit (<2>)	TTmECM0 Bit (<2>)	TTmLDE Bit (<3>)	Counter Clear (Target Compare Register)	Transfer to Counter
Encoder compare mode	Can be set to 00, 01, 10, or 11.	0	0	0	–	–
				1		Possible
			1	0	TTmCCR0	–
				1		Possible[Note]
		1	0	Invalid	TTmCCR1	–
			1	Invalid	TTmCCR0, TTmCCR1	–

Note The counter can operate in a range from 0000H to the set value of the TTmCCR0 register.

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(a) Outline of each bit

<1> The TTmUDS1 and TTmUDS0 bits identify the counting direction (up or down) of the 16-bit counter by the phase input from the encoder input pin (TENCm0 or TENCm1).

<2> The TTmECM1 and TTmECM0 bits control clearing of the 16-bit counter when its count value matches the value of the CCR0 or CCR1 buffer register.

<3> The TTmLDE bit controls a function to transfer the set value of the TTmCCR0 register to the 16-bit counter when the counter underflows. The TTmLDE bit is valid only when the TTmECM1 and TTmECM0 bits are 00 or 01. It is invalid when these bits are set to any other value.

(b) Detailed explanation of each bit

<1> TTmUDS1 and TTmUDS0 bits: Count-up/-down selection

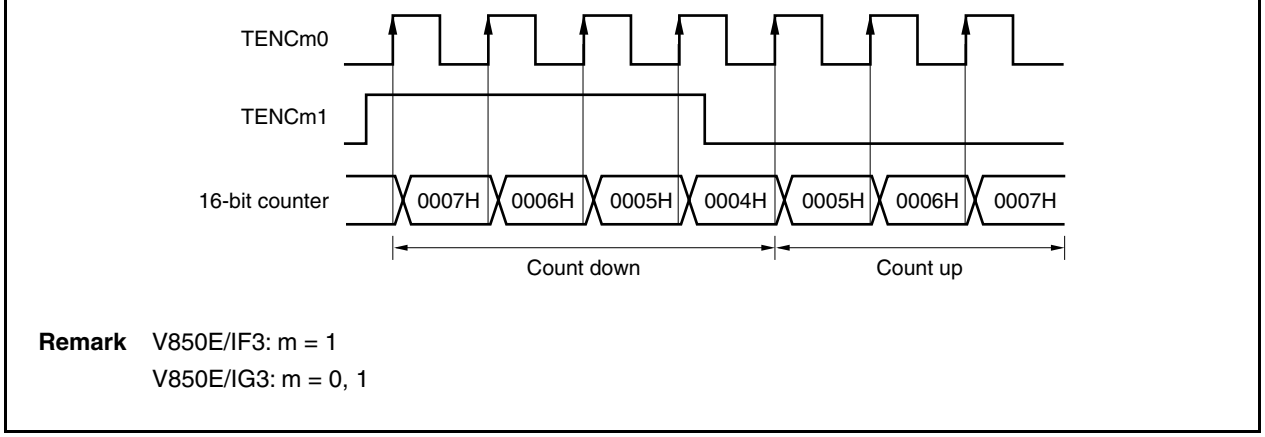
Whether the 16-bit counter is counting up or down is identified by the phase input from the TENCm0 or TENCm1 pin and depending on the setting of the TTmUDS1 and TTmUDS0 bits. These bits are valid only in the encoder compare mode.

- When TTmUDS1 and TTmUDS0 bits = 00

TENCm0 Pin	TENCm1 Pin	Count Operation
Rising edge	High level	Count down
Falling edge		
Both edges		
Rising edge	Low level	Count up
Falling edge		
Both edges		

Remark Detecting the edge of the TENCm0 pin is specified by the TTmIOC3.TTmEIS1 and TTmEIS0 bits.

Figure 8-45. Operation Example (When Valid Edge of TENCm0 Pin Is Specified to Be Rising Edge and No Edge Is Specified as Valid Edge of TENCm1 Pin)



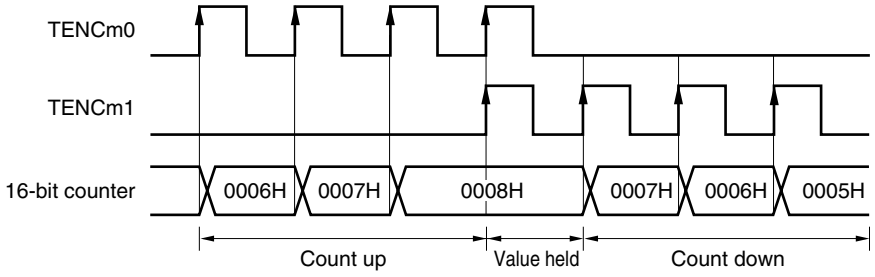
Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

- When TTmUDS1 and TTmUDS0 bits = 01

TENCm0 Pin	TENCm1 Pin	Count Operation
Low level	Rising edge	Count down
	Falling edge	
	Both edges	
High level	Rising edge	
	Falling edge	
	Both edges	
Rising edge	High level	Count up
Falling edge		
Both edges		
Rising edge	Low level	
Falling edge		
Both edges		
Simultaneous input to TENCm0 and TENCm1 pins		Counter does not perform count operation but holds value immediately before.

Remark Detecting the edge of the TENCm0 pin is specified by the TTmIOC3.TTmEIS1 and TTmIOC3.TTmEIS0 bits.

Figure 8-46. Operation Example (When Rising Edge Is Specified as Valid Edge of TENCm0 and TENCm1 Pins)



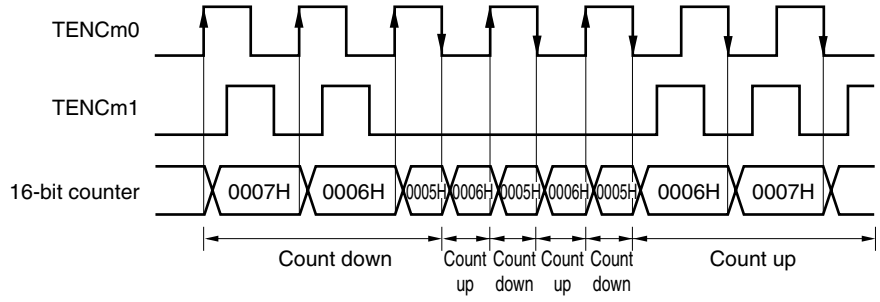
Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

- When TTmUDS1 and TTmUDS0 bits = 10

TENCm0 Pin	TENCm1 Pin	Count Operation
Low level	Falling edge	Counter does not perform count operation but holds value immediately before.
Rising edge	Low level	Count down
High level	Rising edge	Counter does not perform count operation but holds value immediately before.
Falling edge	High level	
Rising edge		
High level	Falling edge	
Falling edge	Low level	Count up
Low level	Rising edge	Counter does not perform count operation but holds value immediately before.
Rising edge		
Falling edge		
Rising edge	Falling edge	Count down
Falling edge		Count up

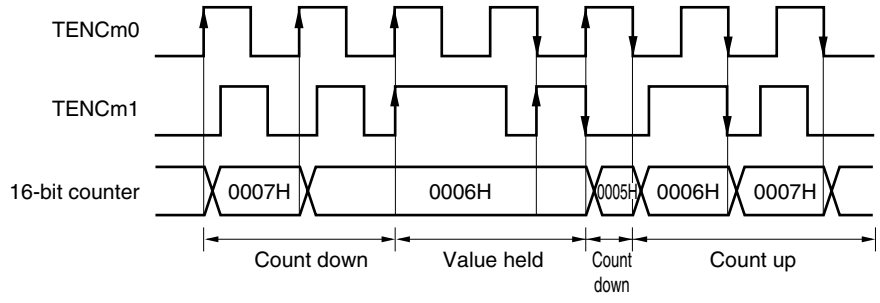
Caution Specification of the valid edge of the TENCm0 and TENCm1 pins is invalid.

Figure 8-47. Operation Example (Count Operation When Valid Edges of TENCm0 and TENCm1 Pins do not Overlap)



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Figure 8-48. Operation Example (Count Operation When Valid Edges of TENCm0 and TENCm1 Pins Overlap)



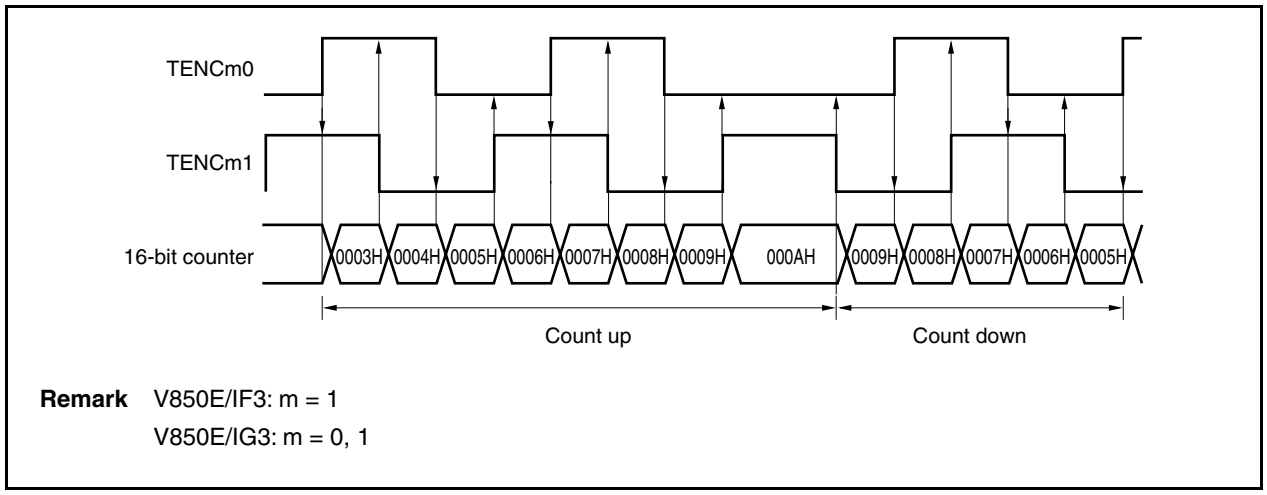
Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

- When TTmUDS1 and TTmUDS0 bits = 11

TENCm0 Pin	TENCm1 Pin	Count Operation
Low level	Falling edge	Count down
Rising edge	Low level	
High level	Rising edge	
Falling edge	High level	
Rising edge		Count up
High level	Falling edge	
Falling edge	Low level	
Low level	Rising edge	
Simultaneous input to TENCm0 and TENCm1 pins		Counter does not perform count operation but holds value immediately before.

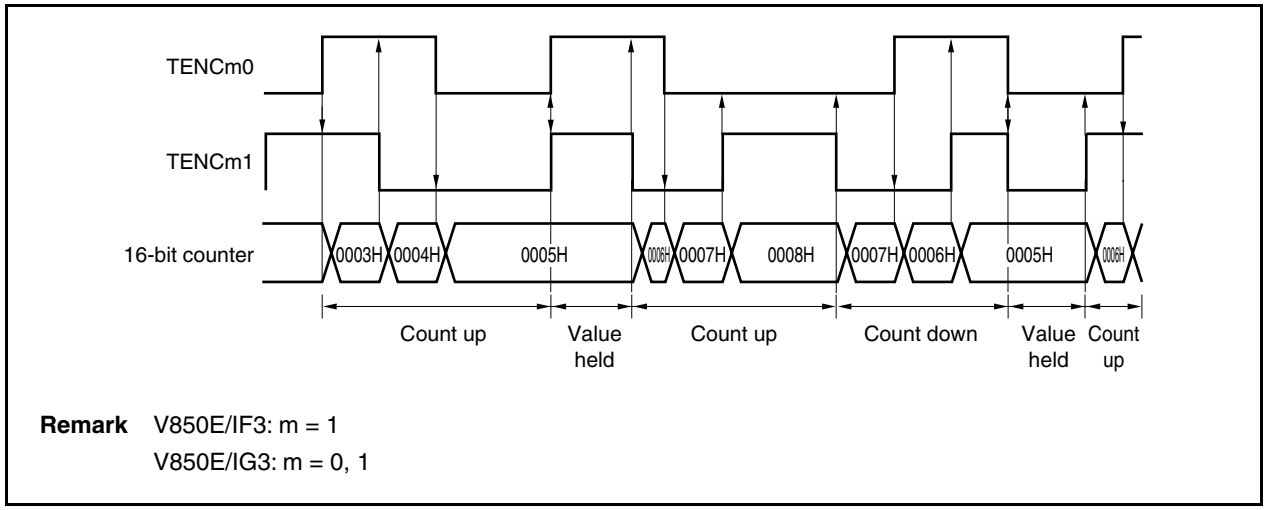
Caution Specification of the valid edge of the TENCm0 and TENCm1 pins is invalid.

Figure 8-49. Operation Example (Count Operation When Valid Edges of TENCm0 and TENCm1 Pins do not Overlap)



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Figure 8-50. Operation Example (Count Operation When Valid Edges of TENCm0 and TENCm1 Pins Overlap)



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

<2> TTmECM1 and TTmECM0 bits: Timer/counter clear function upon match of the compare register

The 16-bit counter performs its count operation in accordance with the set value of the TTmECM1 and TTmECM0 bits when the count value of the counter matches the value of the CCRa buffer register.

- When TTmECM1 and TTmECM0 bits = 00

  The 16-bit counter is not cleared when its count value matches the value of the CCRa buffer register.

- When TTmECM1 and TTmECM0 bits = 01

  The 16-bit counter performs a count operation under the following condition when its count value matches the value of the CCR0 buffer register.

Next Count Operation	Description
Count up	16-bit counter is cleared to 0000H.
Count down	Count value of 16-bit counter is counted down.

- When TTmECM1 and TTmECM0 bits = 10

  The 16-bit counter performs a count operation under the following condition when its count value matches the value of the CCR1 buffer register.

Next Count Operation	Description
Count up	Count value of 16-bit counter is counted up.
Count down	16-bit counter is cleared to 0000H.

- When TTmECM1 and TTmECM0 bits = 11

  The 16-bit counter performs a count operation under the following condition when its count value matches the value of the CCR0 buffer register.

Next Count Operation	Description
Count up	16-bit counter is cleared to 0000H.
Count down	Count value of 16-bit counter is counted down.

  The 16-bit counter performs a count operation under the following condition when its count value matches the value of the CCR1 buffer register.

Next Count Operation	Description
Count up	Count value of 16-bit counter is counted up.
Count down	16-bit counter is cleared to 0000H.

<3> TTmLDE bit: Transfer function of the set value of the TTmCCR0 register to the 16-bit counter when the counter underflows

When the TTmLDE bit = 1, the set value of the TTmCCR0 register can be transferred to the 16-bit counter when the counter underflows.

The TTmLDE bit is valid only in the encoder compare mode.

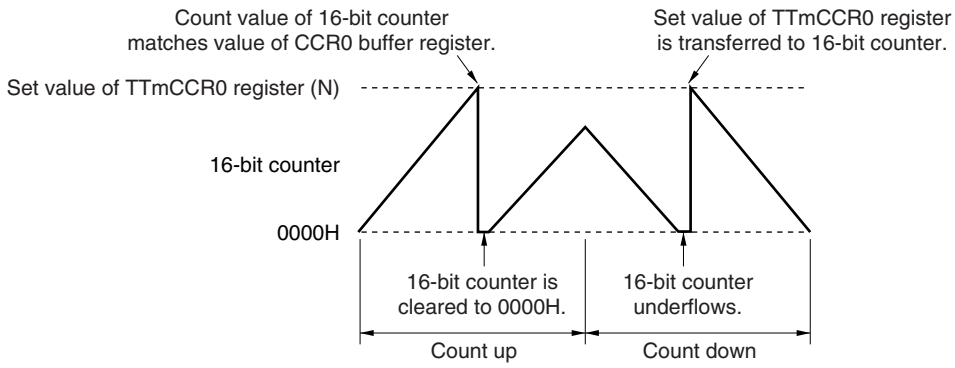
- Count operation in range from 0000H to set value of the TTmCCR0 register

  If the 16-bit counter performs a count operation when the TTmLDE bit = 1 and TTmECM1 and TTmECM0 bits = 01, and when the count value of the counter matches the set value of the CCR0 buffer register when the TTmECM0 bit = 1, the 16-bit counter is cleared to 0000H if the next count operation is counting up.

  If the 16-bit counter underflows when the TTmLDE bit = 1, the set value of the TTmCCR0 register is transferred to the counter.

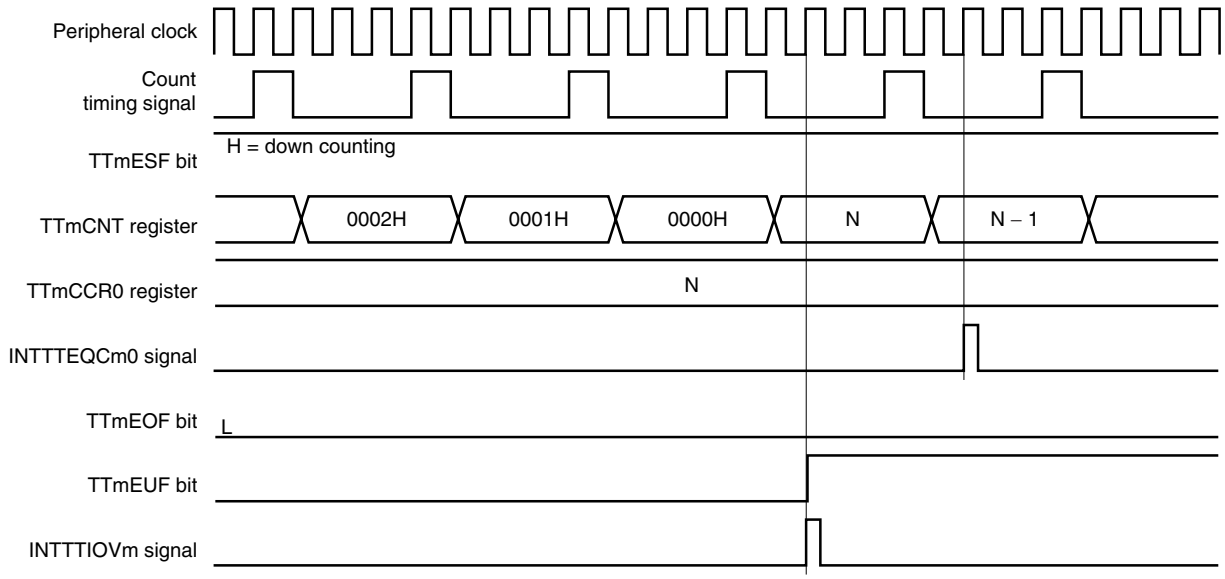
  Therefore, the counter can operate in a range from 0000H to the set value of the TTmCCR0 register in which the upper-limit count value is the set value of the TTmCCR0 register and the lower-limit value is 0000H.

Figure 8-51. Operation Example (Count Operation in Range from 0000H to Set Value of TTmCCR0 Register)



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Figure 8-52. Operation Timing (Count Operation in Range from 0000H to Set Value of TTmCCR0 Register)



Remarks 1. TTmESF bit: Bit 0 of TMTm option register 1 (TTmOPT1)
TTmEOF bit: Bit 1 of TMTm option register 1 (TTmOPT1)
TTmEUF bit: Bit 2 of TMTm option register 1 (TTmOPT1)

2. V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(6) Clearing counter to 0000H by encoder clear signal (TECRm pin)

The 16-bit counter can be cleared to 0000H by the input signal of the TECRm pin in two ways which are selected by the TTmIOC3.TTmSCE bit. The TTmSCE bit also controls, depending its setting, the TTmIOC3.TTmZCL, TTmIOC3.TTmBCL, TTmIOC3.TTmACL, TTmIOC3.TTmESC1, and TTmIOC3.TTmECS0 bits.

The counter can be cleared by the methods described below only in the encoder compare mode.

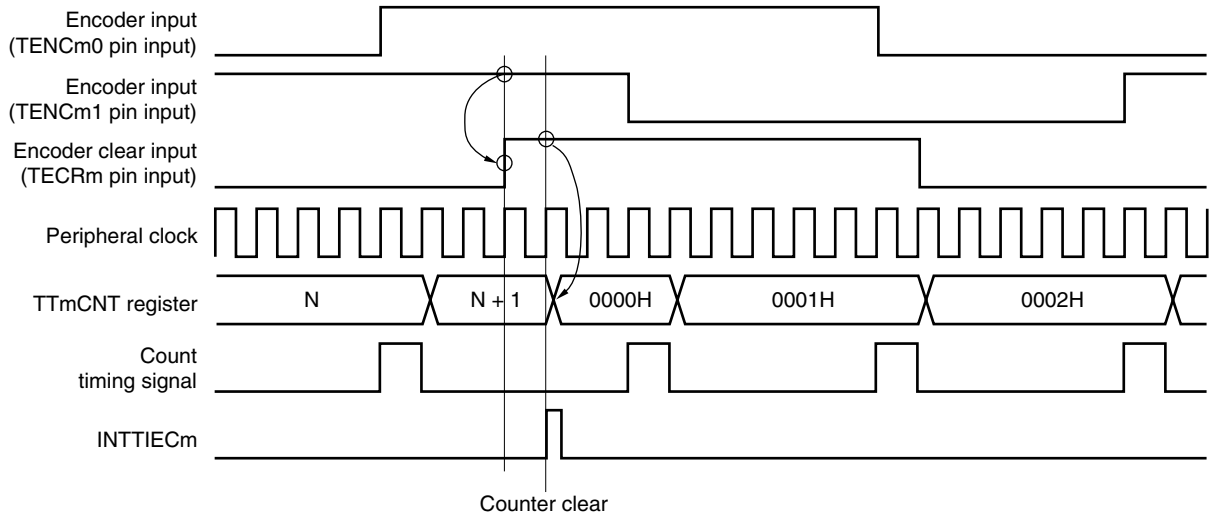
Table 8-11. Relationship Between TTmSCE Bit and TTmZCL, TTmBCL, TTmACL, TTmECS1, and TTmECS0 Bits

Clearing Method	TTmSCE Bit	TTmZCL Bit	TTmBCL Bit	TTmACL Bit	TTmECS1, TTmECS0 Bits
<1>	0	Invalid	Invalid	Invalid	Valid
<2>	1	Valid	Valid	Valid	Invalid

(a) Clearing method <1>: By detecting edge of encoder clear signal (TECRm pin) (TTmSCE bit = 0)

When the TTmSCE bit = 0, the 16-bit counter is cleared to 0000H in synchronization with the peripheral clock if the valid edge of the TECRm pin specified by the TTmECS1 and TTmECS0 bits is detected. At this time, an encoder clear interrupt request signal (INTTIECm) is generated. When the TTmSCE bit = 0, setting of the TTmZCL, TTmBCL, and TTmACL bits is invalid.

Figure 8-53. Operation Example (When TTmSCE Bit = 0, TTmECS1 and TTmECS0 Bits = 01, and TTmUDS1 and TTmUDS0 Bits = 11)



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(b) Clearing method <2>: By detecting clear level condition of the TENCm0, TENCm1, and TECRm pins (TTmSCE bit = 1)

When the TTmSCE bit = 1, the 16-bit counter is cleared to 0000H if the clear level condition of the TECRm, TENCm0, or TENCm1 pin specified by the TTmZCL, TTmBCL, and TTmACL bits is detected. At this time, the encoder clear interrupt request signal (INTTIECm) is not generated. Setting of the TTmECS1 and TTmECS0 bits is invalid when the TTmSCE bit = 1.

Table 8-12. 16-bit Counter Clearing Condition When TTmSCE Bit = 1

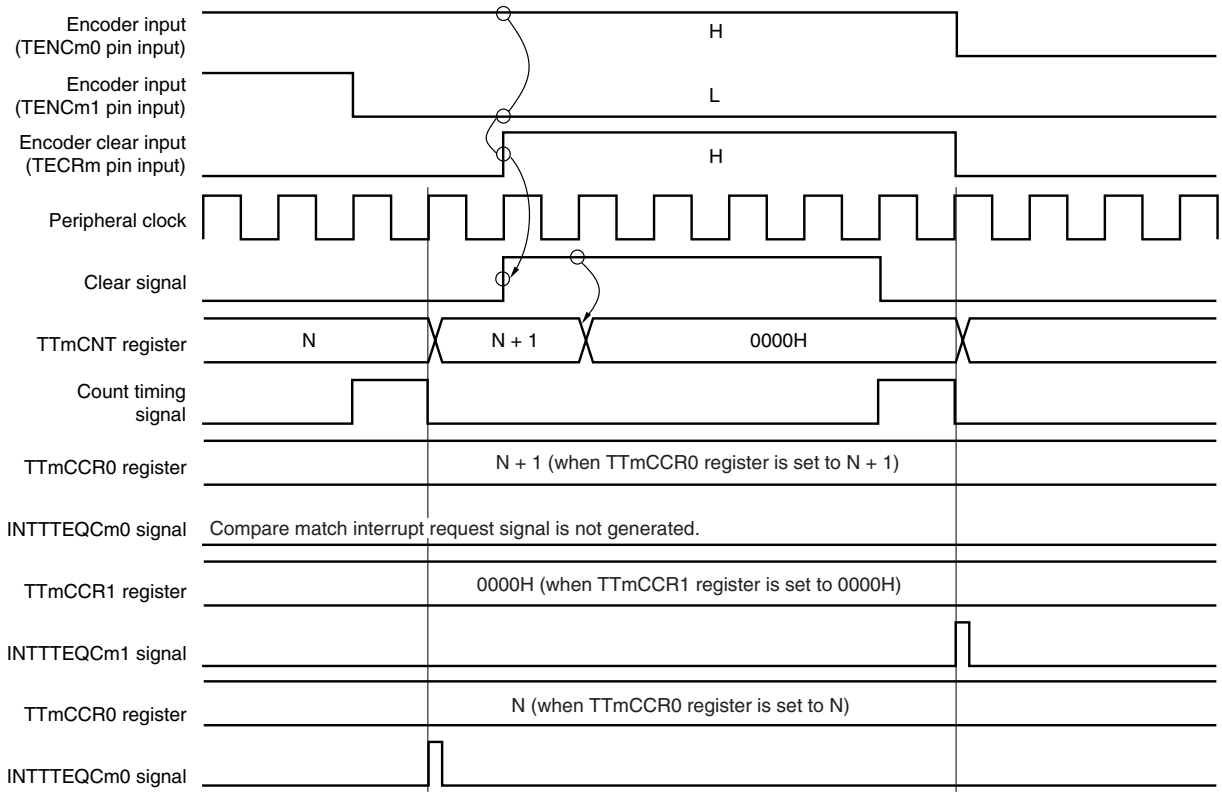
Clear Level Condition Setting			Input Level of Encoder Pin		
TTmZCL Bit	TTmBCL Bit	TTmACL Bit	TECRm Pin	TENCm1 Pin	TENCm0 Pin
0	0	0	L	L	L
0	0	1	L	L	H
0	1	0	L	H	L
0	1	1	L	H	H
1	0	0	H	L	L
1	0	1	H	L	H
1	1	0	H	H	L
1	1	1	H	H	H

Caution The 16-bit counter is cleared to 0000H when the clear level condition of the TTmZCL, TTmBCL, and TTmACL bits match the input level of the TECRm, TENCm1, or TENCm0 pin.

Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Figure 8-54. Operation Example (When TTmSCE Bit = 1, TTmZCL Bit = 1, TTmBCL Bit = 0, TTmACL Bit = 1, TTmUDS1 and TTmUDS0 Bits = 11, TECRm = High Level, TENCm1 = Low Level, and TENCm0 = High Level) (1/3)

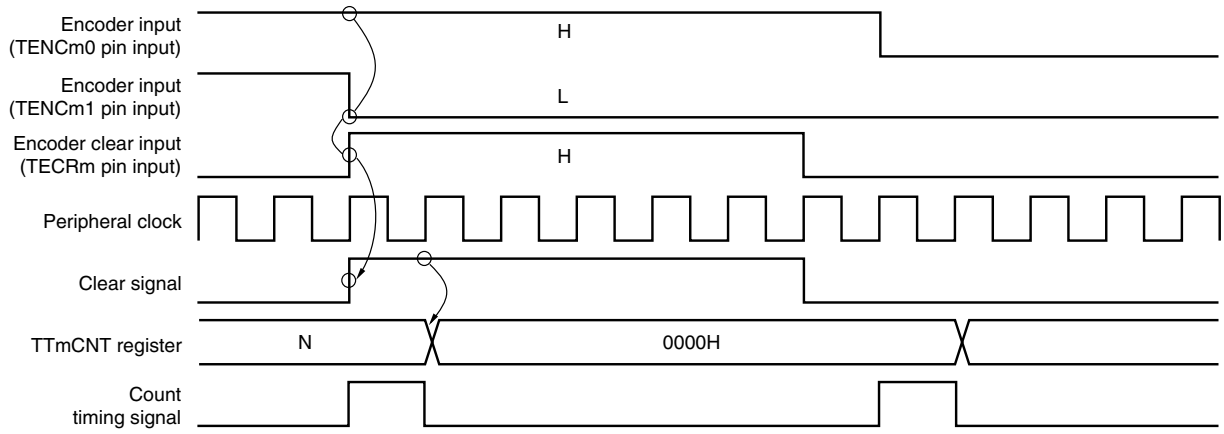
(i) If inputting the high level to the TECRm pin lags behind inputting the low level to the TENCm1 pin while the counter is counting up, the counter is cleared after it counts up.



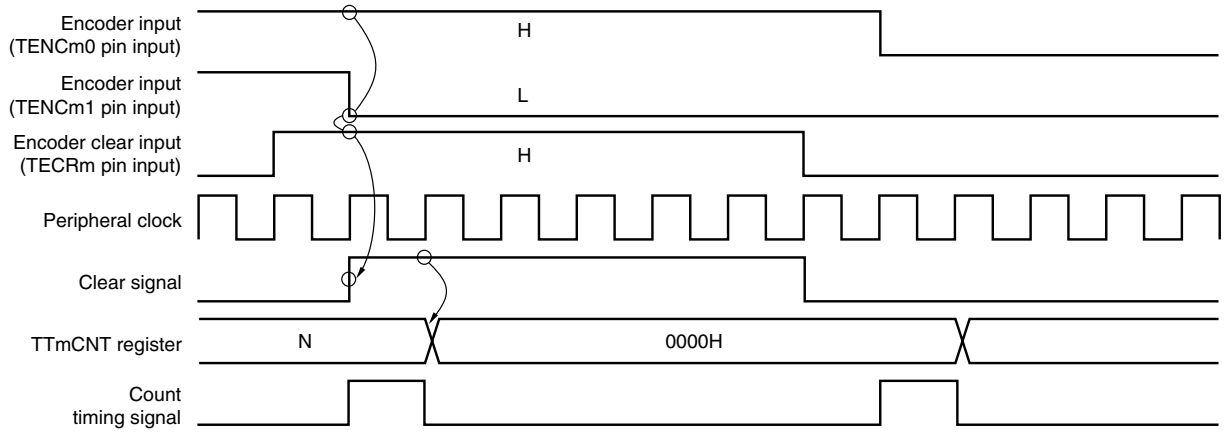
Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Figure 8-54. Operation Example (When TTmSCE Bit = 1, TTmZCL Bit = 1, TTmBCL Bit = 0, TTmACL Bit = 1, TTmUDS1 and TTmUDS0 Bits = 11, TECRm = High Level, TENCm1 = Low Level, and TENCm0 = High Level) (2/3)

(ii) If the high level is input to the TECRm pin at the same time as the low level is input to the TECNm1 pin while the counter is counting up, the counter is cleared without counting up.



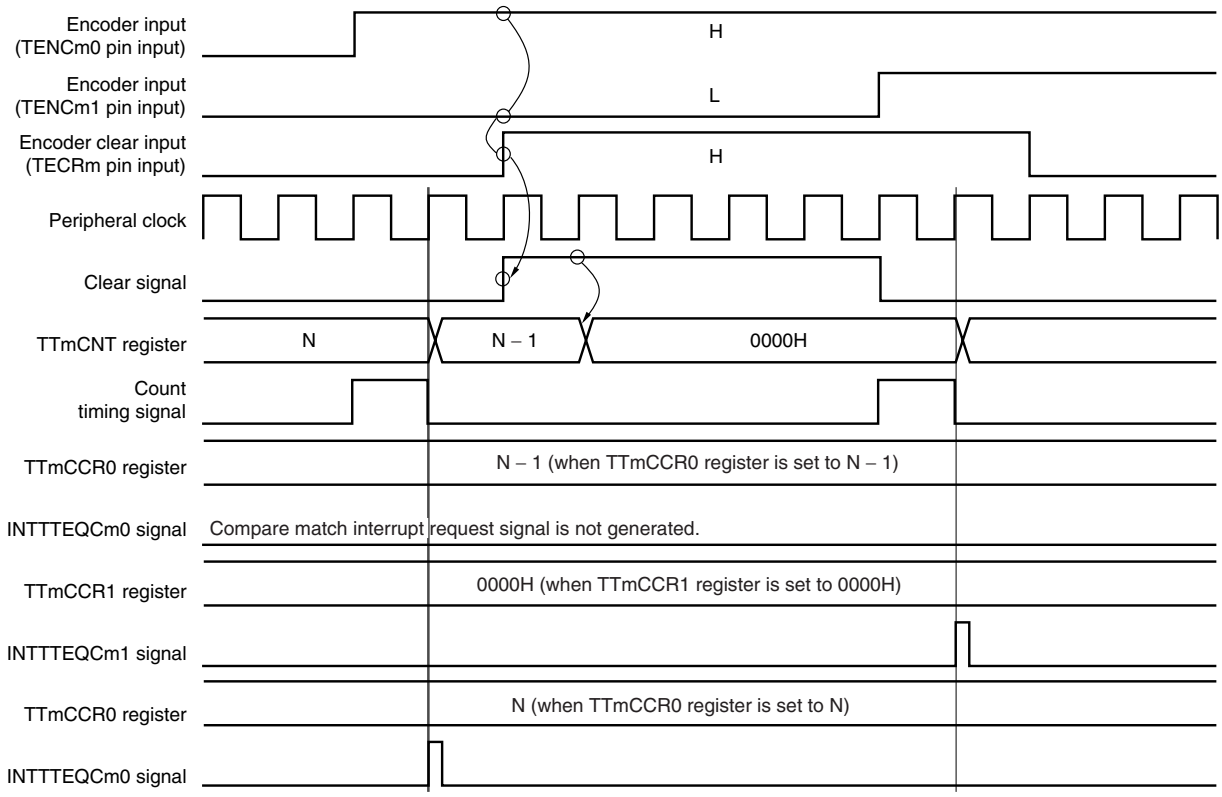
(iii) If the high level is input to the TECRm pin earlier than the low level is input to the TENCm1 pin while the counter is counting up, the counter is cleared without counting up.



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Figure 8-54. Operation Example (When TTmSCE Bit = 1, TTmZCL Bit = 1, TTmBCL Bit = 0, TTmACL Bit = 1, TTmUDS1 and TTmUDS0 Bits = 11, TECRm = High Level, TENCm1 = Low Level, and TENCm0 = High Level) (3/3)

(iv) If the high level is input to the TECRm pin later than the low level is input to the TENCm1 pin while the counter is counting up, the counter is cleared after it counts up.



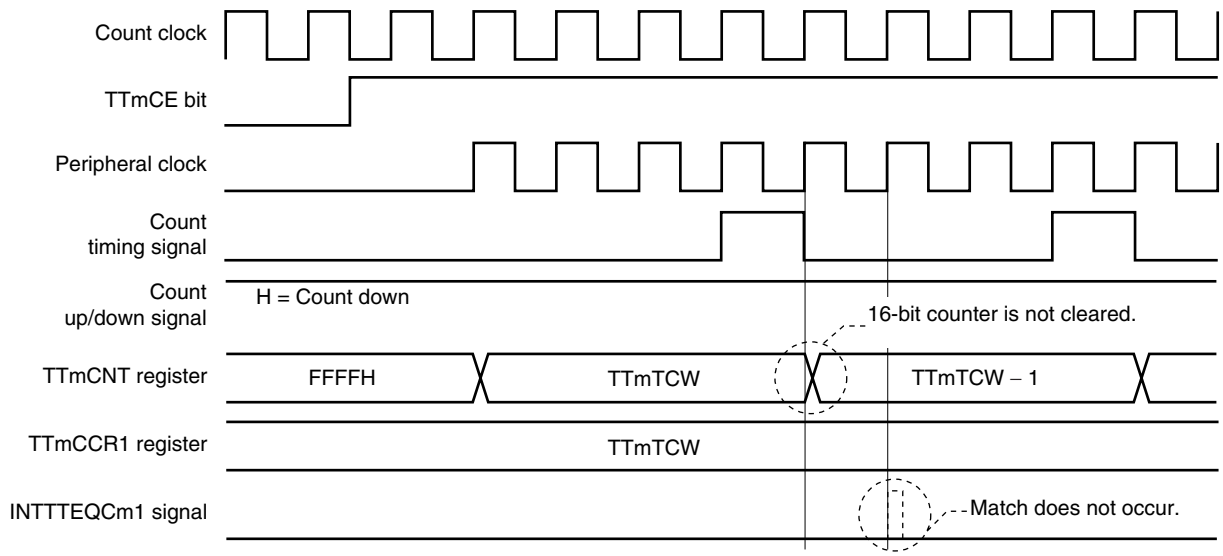
Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

If the counter is cleared in this way, a miscount does not occur even if inputting the signal to the TECRm pin is late, because the clear level condition of the TECRm, TENCm1, and TENCm0 pins is set and the 16-bit counter is cleared to 0000H when the clear level condition is detected.

(7) Notes on using encoder count function

(a) If compare match interrupt is not generated immediately after operation is started

If a value which is the same as that of the TTmTCW register is set to the TTmCCR0 or TTmCCR1 register and the counter operation is started when the TTmCTL2.TTmECC bit = 0, and if the count value (TTmTCW) of the 16-bit counter matches the value of the CCRa buffer register immediately after the start of the operation, the match is masked and the compare match interrupt request signal (INTTTEQCma) is not generated (a = 0, 1). In addition, the 16-bit counter is not cleared to 0000H by setting the TTmCTL2.TTmECM1 and TTmCTL2.TTmECM0 bits.

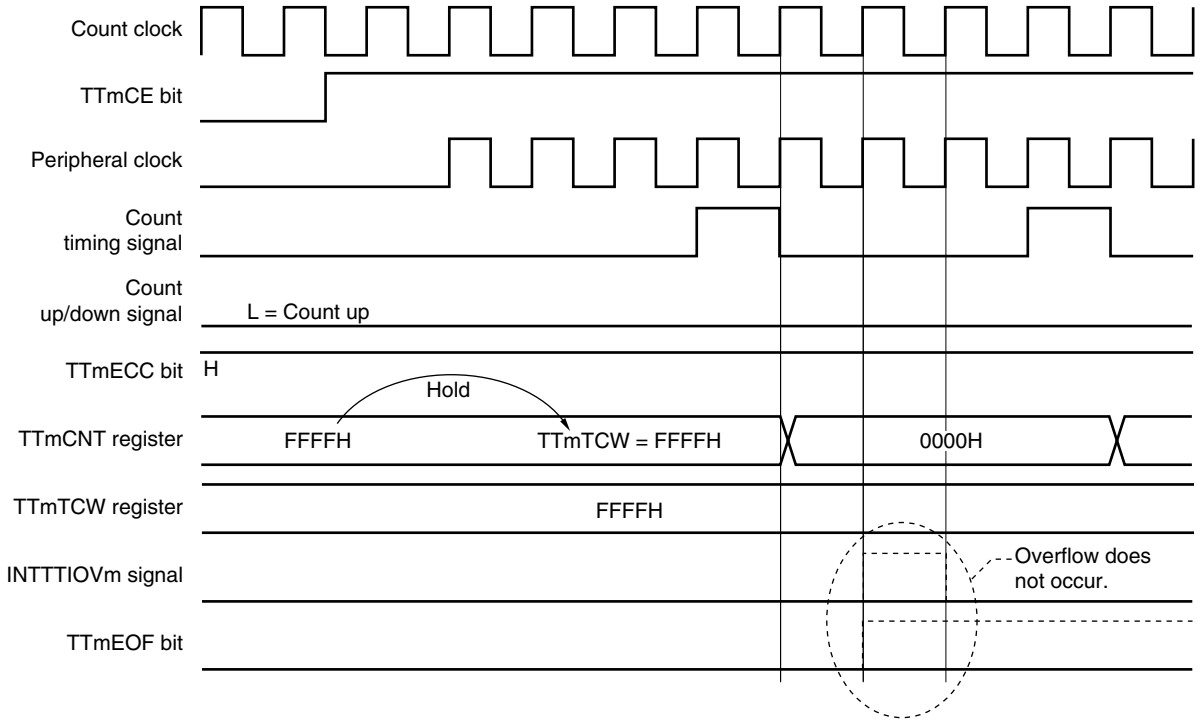


Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(b) If overflow does not occur immediately after start of operation

If the count operation is resumed when the TTmCTL2.TTmECC bit = 1, the 16-bit counter does not overflow if its count value that has been held is FFFFH and if the next count operation is counting up.

After the counter starts operating and counts up from a count value (value of TTmTCW register = FFFFH), the counter overflows from FFFFH to 0000H. However, detection of the overflow is masked, the overflow flag (TTmEOF) is not set, and the overflow interrupt request signal (INTTTIOVm) is not generated.



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

8.6.10 Encoder compare mode (TTmMD3 to TTmMD0 bits = 1000)

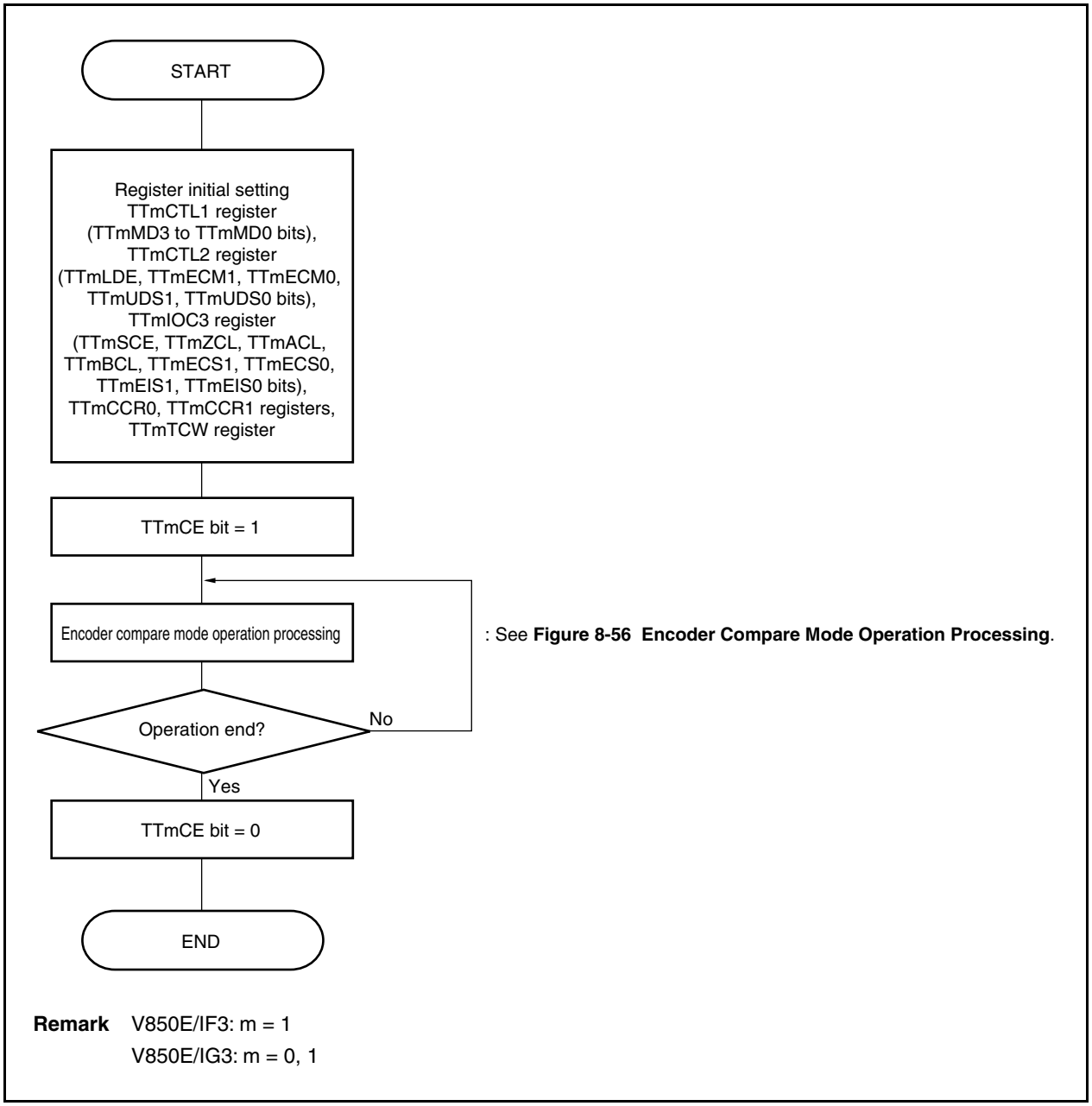
In the encoder compare mode, the encoder is controlled by using both the TTmCCR0 and TTmCCR1 registers as compare registers and the input pins for encoder count function (TENCm0, TENCm1, and TECRm).

In this mode, the 16-bit counter can be cleared to 0000H in three ways: when the count value of the counter matches the value of the CCRa buffer register (compare match interrupt request signal (INTTTEQCma) is generated), when the edge of the encoder clear input (TECRm pin) is detected and cleared, and when the clear level condition of TENCm0, TENCm1, and TECRm pins is detected and cleared.

When the 16-bit counter underflows, the set value of the TTmCCR0 register can be transferred to the counter.

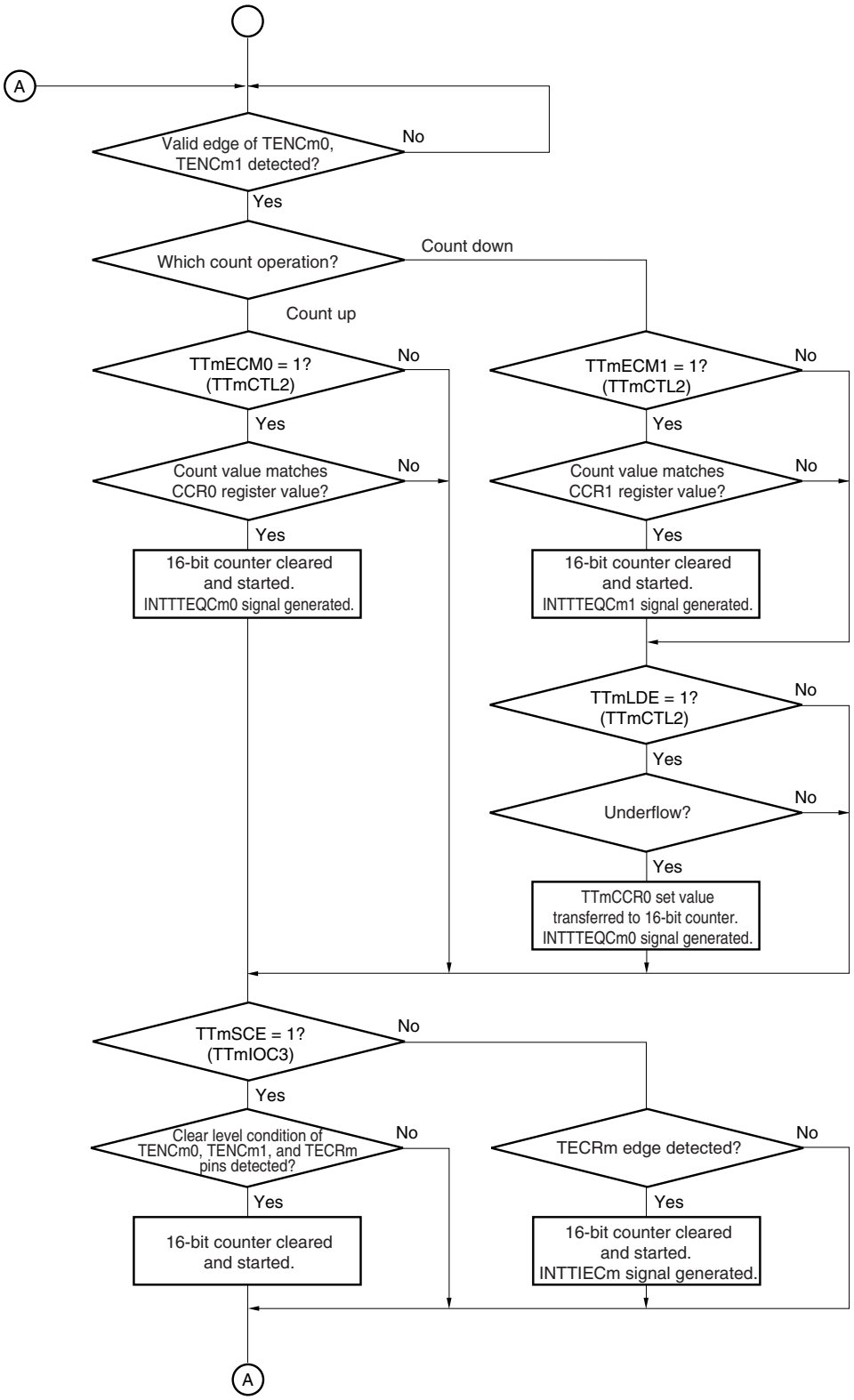
(1) Encoder compare mode operation flow

Figure 8-55. Encoder Compare Mode Operation Flow



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

Figure 8-56. Encoder Compare Mode Operation Processing



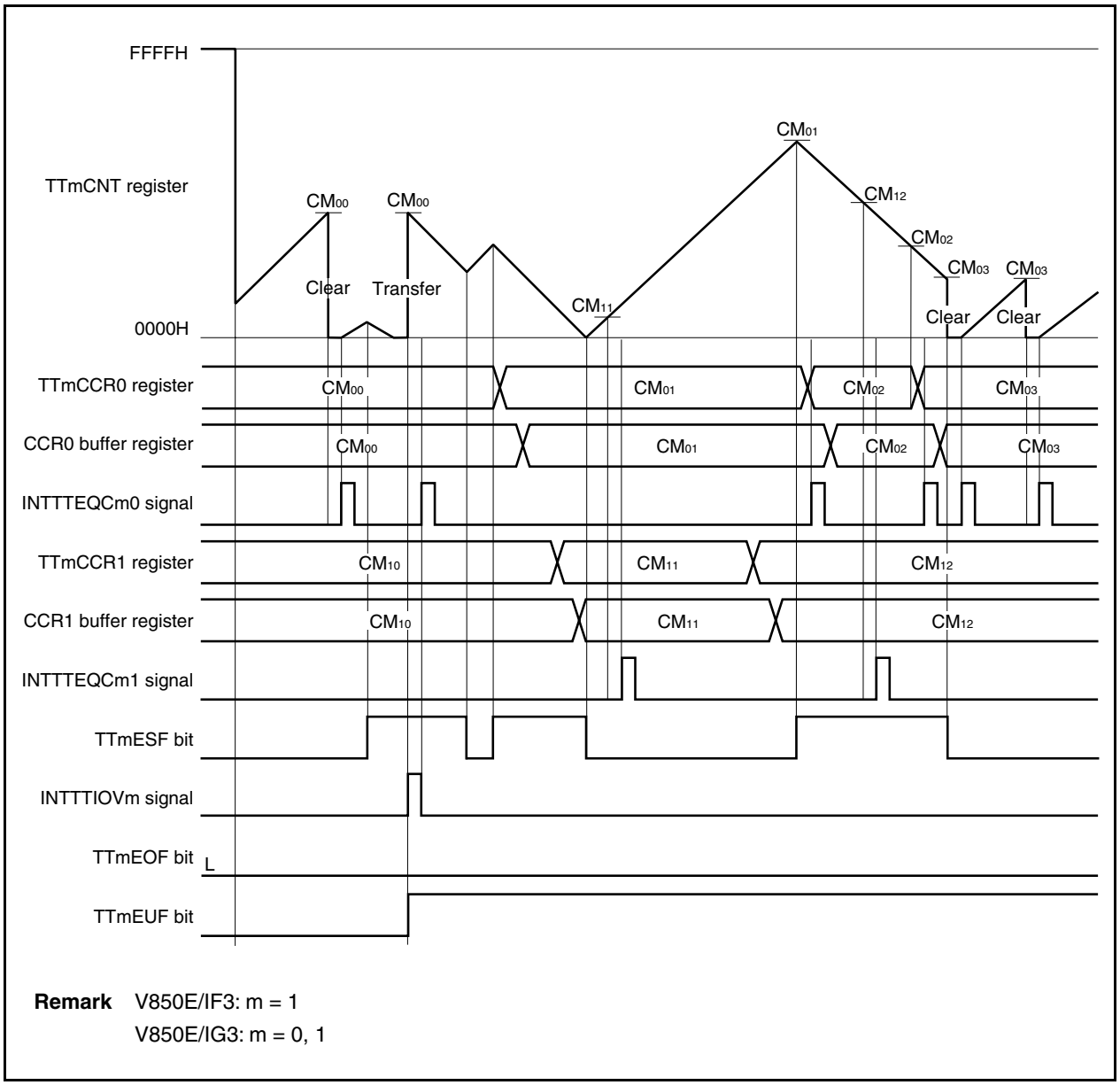
Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

(2) Encoder compare mode operation timing

(a) Basic timing 1

[Register setting conditions]

- TTmCTL2.TTmECM1 and TTmCTL2.TTmECM0 bits = 01
  The 16-bit counter is cleared to 0000H when its count value matches the value of the CCR0 buffer register.
- TTmCTL2.TTmLDE bit = 1
  The set value of the TTmCCR0 register is transferred to the 16-bit counter when it overflows.
- TTmIOC3.TTmSCE bit = 0, and TTmIOC3.TTmECS1 and TTmIOC3.TTmECS0 bits = 00
  Specification of the edge of encoder clear input signal (TECRm pin) to be detected and cleared (no edge specified)



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

When the 16-bit counter starts operating (TTmCE bit = 0 → 1), the set value of the TTmTCW register is transferred to the counter and the 16-bit counter starts operating.

When the count value of the counter matches the value of the CCR0 buffer register, the compare match interrupt request signal (INTTTEQCm0) is generated. Because the TTmECM0 bit = 1, the 16-bit counter is cleared to 0000H if the next count operation is counting up.

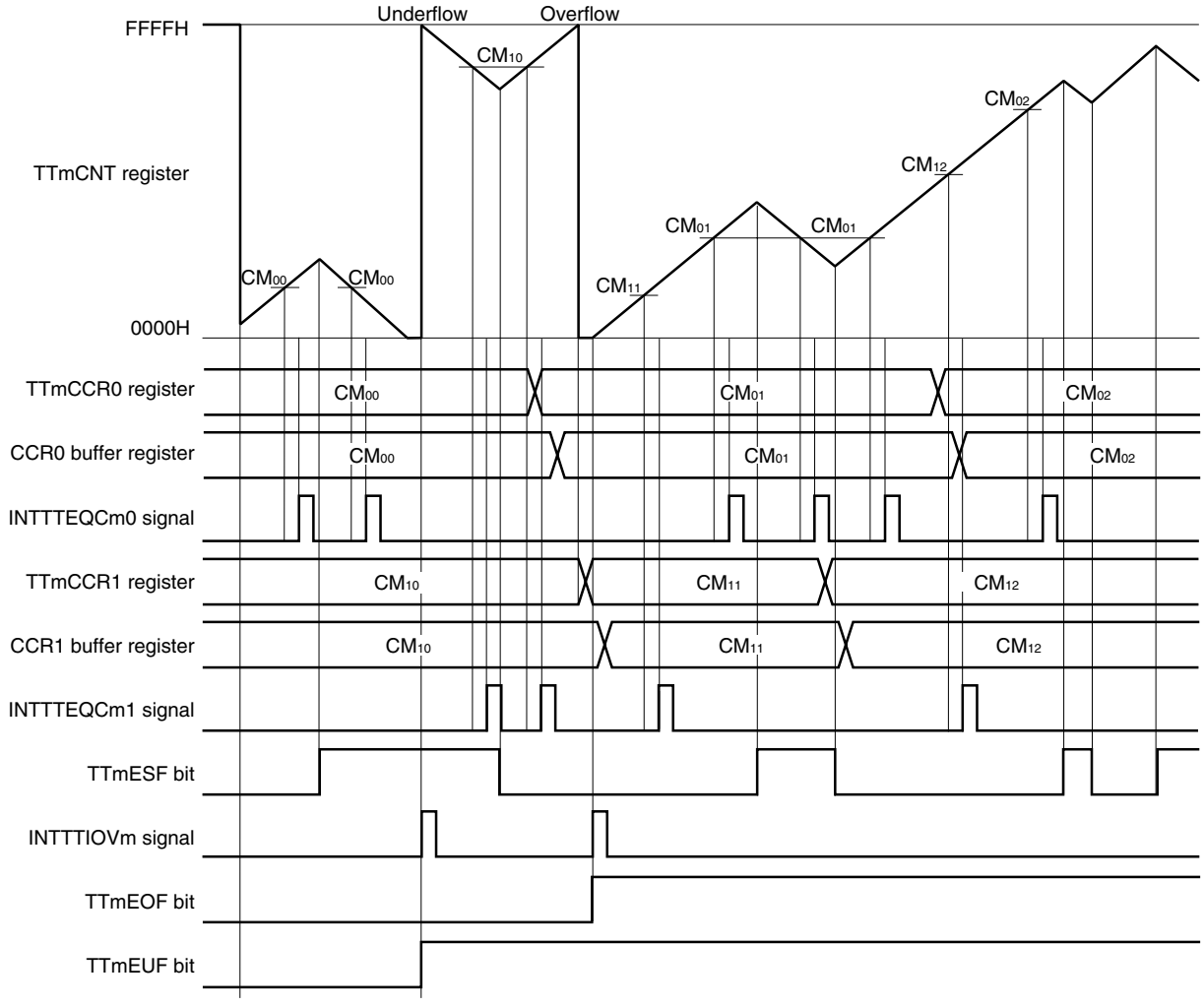
When the count value of the 16-bit counter matches the value of the CCR1 buffer register, the compare match interrupt request signal (INTTTEQCm1) is generated. Because the TTmECM1 bit = 0, the 16-bit counter is not cleared to 0000H when its value matches that of the CCR1 buffer register.

When the TTmLDE bit = 1 and TTmECM0 bit = 1, the counter can operate in a range from 0000H to the set value of the TTmCCR0 register.

(b) Basic timing 2

[Register setting condition]

- TTmCTL2.TTmECM1 and TTmCTL2.TTmECM0 bits = 00
  The 16-bit counter is not cleared even when its count value matches the value of the CCRa buffer register (a = 0, 1).
- TTmCTL2.TTmLDE bit = 0
  The set value of the TTmCCR0 register is not transferred to the 16-bit counter after the counter underflows.
- TTmIOC3.TTmSCE bit = 0, and TTmIOC3.TTmECS1 and TTmIOC3.TTmECS0 bits = 00
  Specification of the edge of the encoder clear input signal (TECRm pin) to be detected and cleared (no edge specified)



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

When the 16-bit counter starts operating (TTmCE bit = 0 → 1), the set value of the TTmTCW register is transferred to the 16-bit counter and the counter starts operating.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTTEQCm0) is generated.

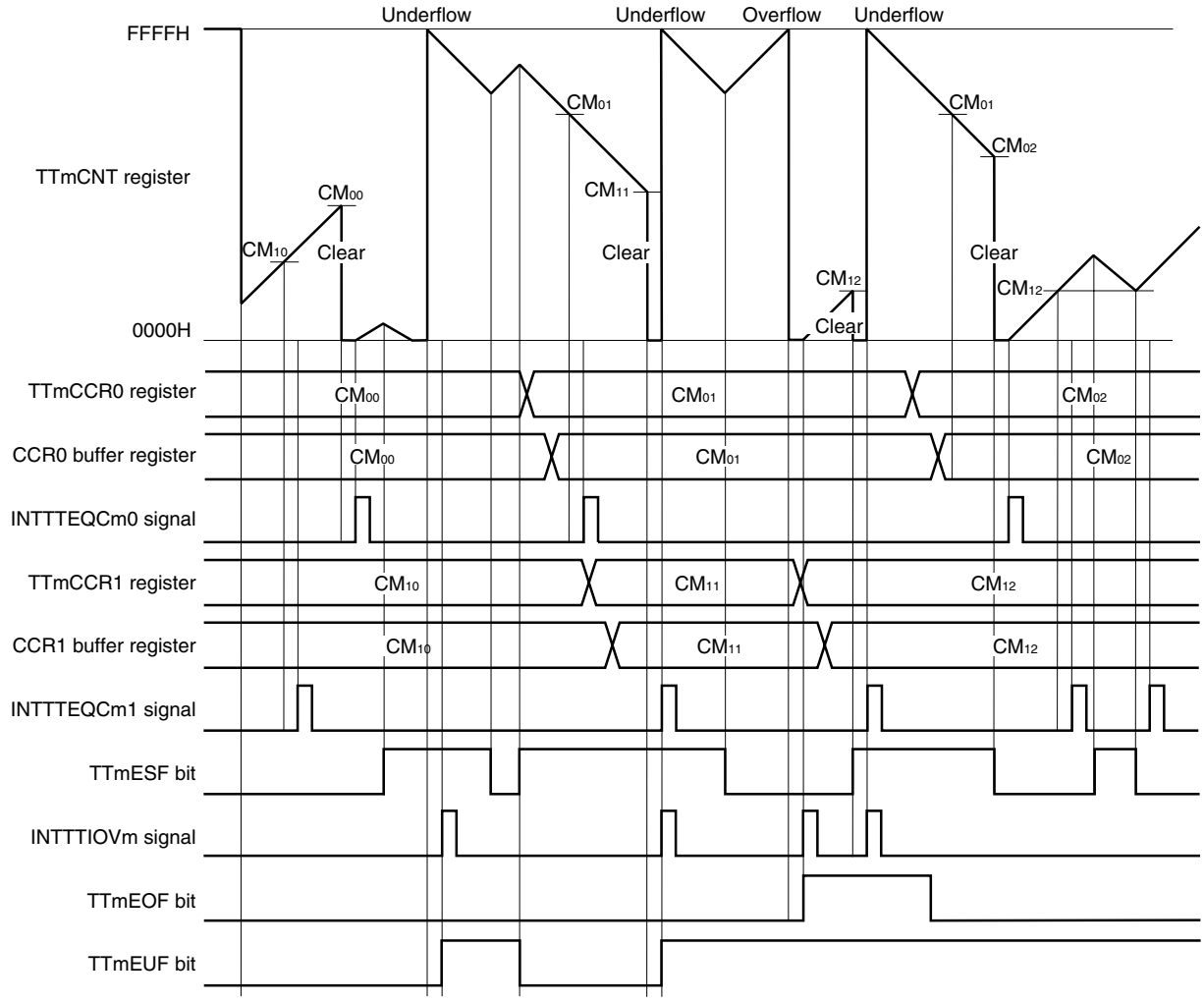
When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTTEQCm1) is generated.

The 16-bit counter is not cleared to 0000H even when its count value matches the value of the CCRa buffer register because the TTmECM1 and TTmECM0 bits = 00 (a = 0, 1).

(c) Basic timing 3

[Register setting condition]

- TTmCTL2.TTmECM1 and TTmCTL2.TTmECM0 bits = 11
  The count value of the 16-bit counter is cleared to 0000H when its value matches the value of the CCR0 buffer register.
  The count value of the 16-bit counter is cleared to 0000H when its value matches the value of the CCR1 buffer register.
- Setting of the TTmCTL2.TTmLDE bit is invalid.
- TTmIOC3.TTmSCE bit = 0, and TTmIOC3.TTmECS1 and TTmIOC3.TTmECS0 bits = 00
  Specification of the edge of the encoder clear input signal (TECRm pin) to be detected and cleared (no edge specified)



Remark V850E/IF3: m = 1
V850E/IG3: m = 0, 1

When the 16-bit counter starts operating (TTmCE bit = 0 → 1), the set value of the TTmTCW register is transferred to the 16-bit counter and the counter starts operating.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTTEQCm0) is generated. At this time, the 16-bit counter is cleared to 0000H if the next count operation is counting up.

When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTTEQCm1) is generated. At this time, the 16-bit counter is cleared to 0000H if the next count operation is counting down.

CHAPTER 9 16-BIT INTERVAL TIMER M (TMM)

Timer M (TMM) is a 16-bit interval timer.
The V850E/IF3 and V850E/IG3 incorporate TMM0 to TMM3.

9.1 Overview

An outline of TMMn is shown below (n = 0 to 3).

- Interval function
- 8 clocks selectable
- 16-bit counter × 1 (The 16-bit counter cannot be read during timer count operation.)
- Compare register × 1 (The compare register cannot be written during timer count operation.)
- Compare match interrupt × 1

Timer M supports only the clear & start mode. The free-running timer mode is not supported.

9.2 Configuration

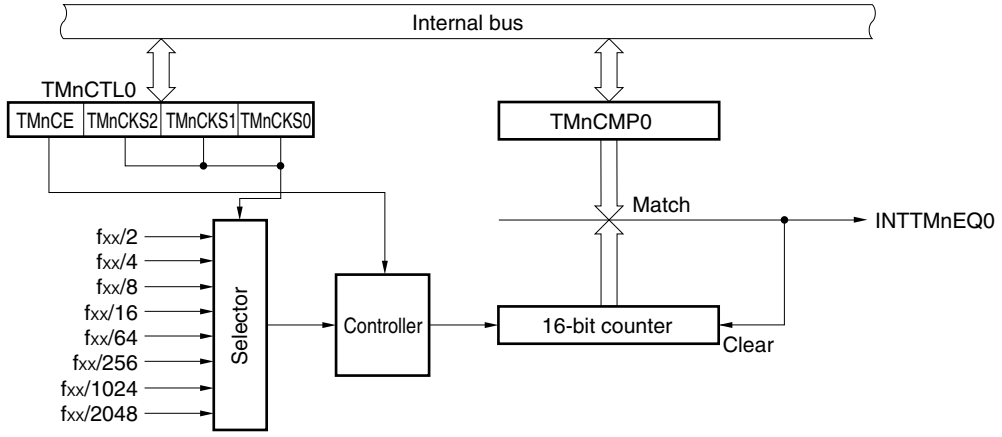
TMMn includes the following hardware (n = 0 to 3).

Table 9-1. Configuration of TMMn

Item	Configuration
Timer register	16-bit counter × 1
Register	TMMn compare register 0 (TMnCMP0)
Control register	TMMn control register 0 (TMnCTL0)

Remark n = 0 to 3

Figure 9-1. Block Diagram of TMMn



Remarks 1. fxx: Peripheral clock frequency

2. n = 0 to 3

(1) 16-bit counter

This is a 16-bit counter that counts the internal clock.

The 16-bit counter cannot be read or written.

(2) TMMn compare register 0 (TMnCMP0)

The TMnCMP0 register is a 16-bit compare register.

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

Reset sets this register to 0000H.

The same value can always be written to the TMnCMP0 register by software.

Rewriting the TMnCMP0 register is prohibited during TMMn operation (TMnCTL0.TMnCE bit = 1).

After reset: 0000H R/W Address: TM0CMP0 FFFFF544H, TM1CMP0 FFFFF554H, TM2CMP0 FFFFF564H, TM3CMP0 FFFFF574H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMnCMP0																
(n = 0 to 3)																

9.3 Control Register

(1) TMMn control register 0 (TMnCTL0)

The TMnCTL0 register is an 8-bit register that controls the TMMn operation.

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

Reset sets this register to 00H.

The same value can always be written to the TMnCTL0 register by software.

After reset: 00H R/W Address: TM0CTL0 FFFFF540H, TM1CTL0 FFFFF550H, TM2CTL0 FFFFF560H, TM3CTL0 FFFFF570H

	<7>	6	5	4	3	2	1	0
TMnCTL0 (n = 0 to 3)	TMnCE	0	0	0	0	TMnCKS2	TMnCKS1	TMnCKS0

TMnCE	Internal clock operation enable/disable specification
0	TMMn operation disabled (16-bit counter reset asynchronously)
1	TMMn operation enabled. Start operation clock supply. Start TMMn operation.
The internal clock control and internal circuit reset for TMMn are performed asynchronously with the TMnCE bit. When the TMnCE bit is cleared to 0, the internal clock of TMMn is stopped (fixed to low level) and 16-bit counter is reset asynchronously.	

TMnCKS2	TMnCKS1	TMnCKS0	Count clock selection
0	0	0	$f_{XX}/2$
0	0	1	$f_{XX}/4$
0	1	0	$f_{XX}/8$
0	1	1	$f_{XX}/16$
1	0	0	$f_{XX}/64$
1	0	1	$f_{XX}/256$
1	1	0	$f_{XX}/1024$
1	1	1	$f_{XX}/2048$

Cautions 1. Set the TMnCKS2 to TMnCKS0 bits when the TMnCE bit = 0.
However, when changing the value of the TMnCE bit from 0 to 1, it is impossible to set the value of the TMnCKS2 to TMnCKS0 bits simultaneously.

2. Be sure to clear bits 3 to 6 to "0".

Remark f_{XX}: Peripheral clock frequency

9.4 Operation

9.4.1 Interval timer mode

In the interval timer mode, an interrupt request signal (INTTMnEQ0) is generated at the interval set by the TMnCMP0 register if the TMnCTL0.TMnCE bit is set to 1.

Figure 9-2. Configuration of Interval Timer

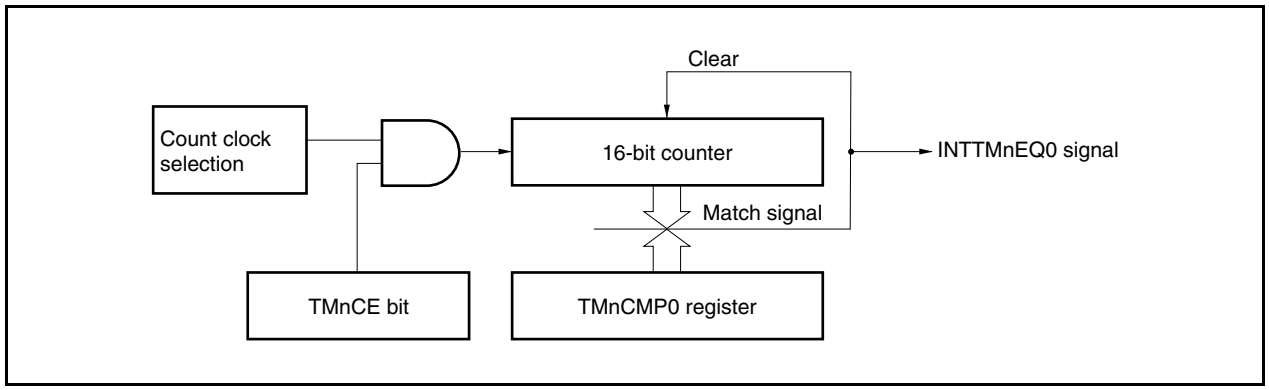
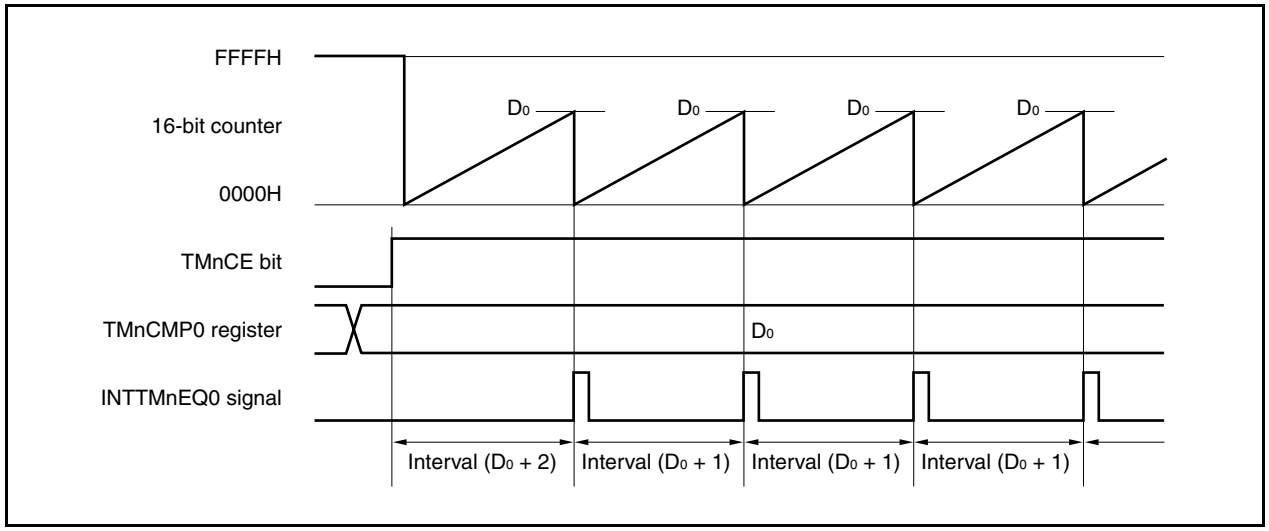


Figure 9-3. Basic Timing of Operation in Interval Timer Mode



When the TMnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H in synchronization with the count clock, and the counter starts counting.

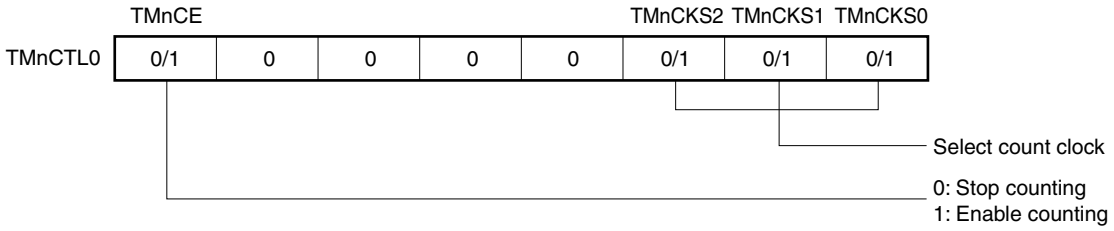
When the count value of the 16-bit counter matches the value of the TMnCMP0 register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTMnEQ0) is generated.

The interval can be calculated by the following expression.

Interval = (Set value of TMnCMP0 register + 1) × Count clock cycle

Figure 9-4. Register Setting for Interval Timer Mode Operation

(a) TMMn control register 0 (TMnCTL0)



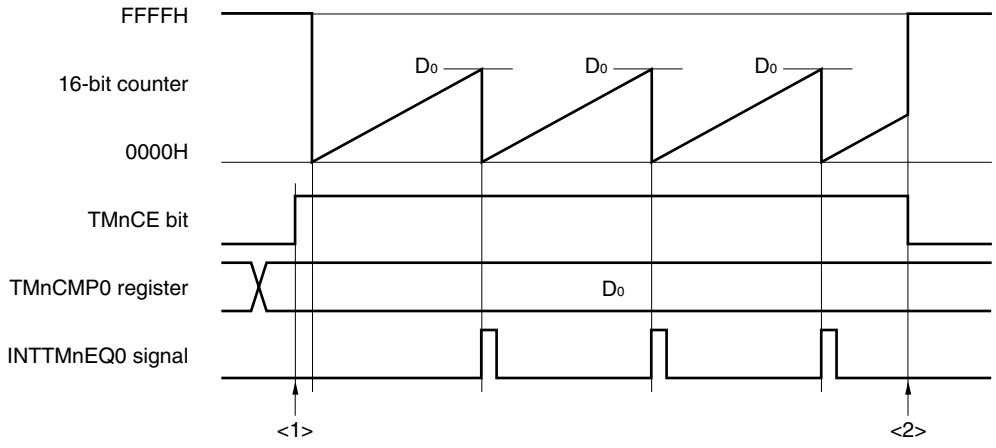
(b) TMMn compare register 0 (TMnCMP0)

If the TMnCMP0 register is set to D_0, the interval is as follows.

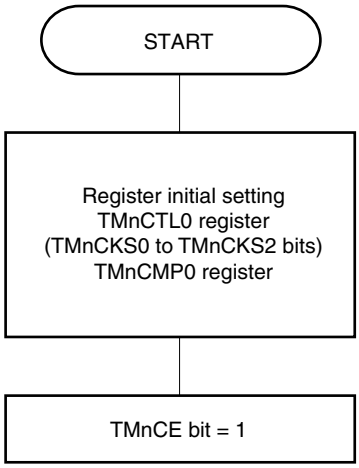
Interval = (D_0 + 1) × Count clock cycle

(1) Interval timer mode operation flow

Figure 9-5. Software Processing Flow in Interval Timer Mode



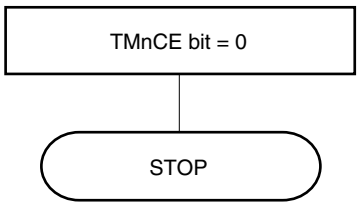
<1> Count operation start flow



Initial setting of these registers is performed before setting the TMnCE bit to 1.

The TMnCKS0 to TMnCKS2 bits cannot be set at the same time as when counting starts (TMnCE bit = 1).

<2> Count operation stop flow

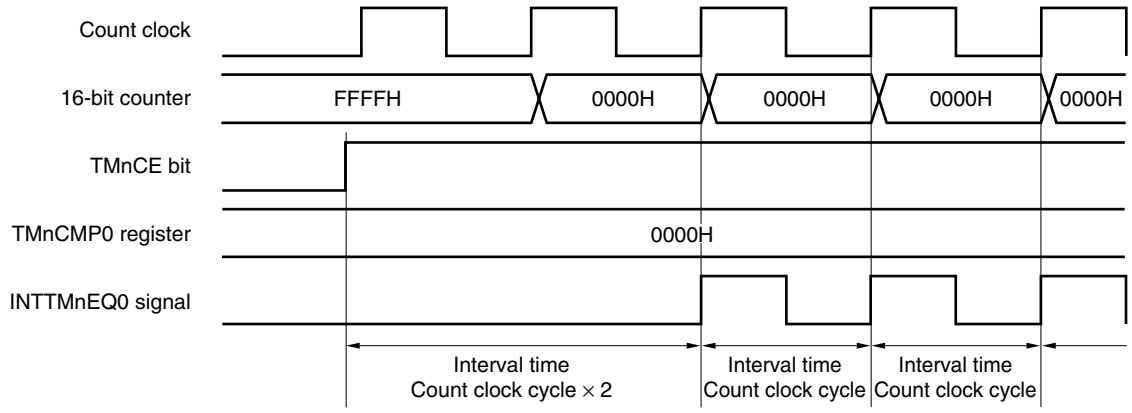


The counter is initialized and counting is stopped by clearing the TMnCE bit to 0.

(2) Interval timer mode operation timing

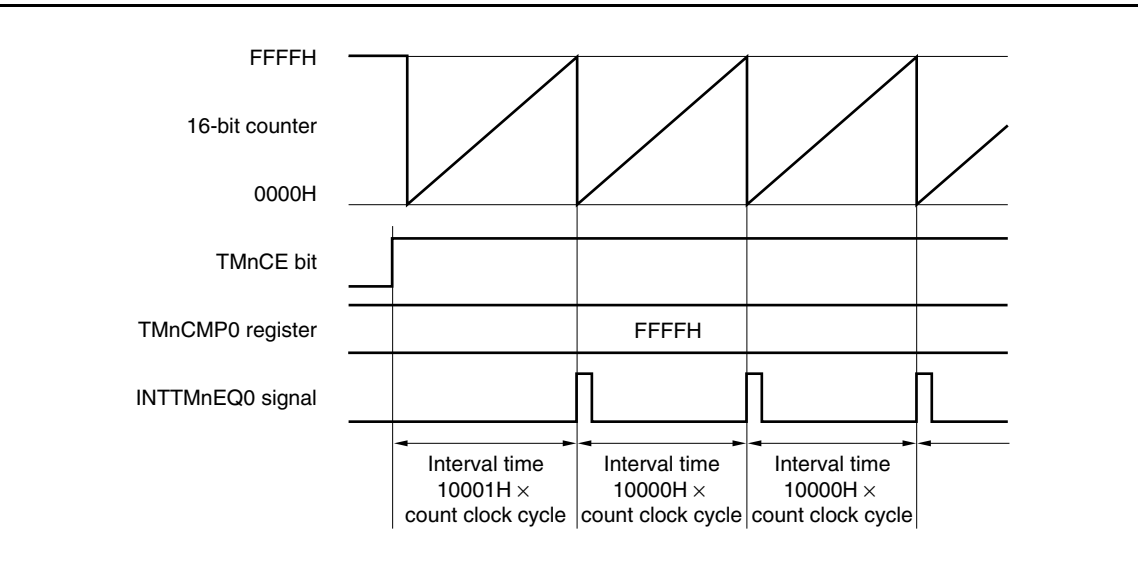
(a) Operation if TMnCMP0 register is set to 0000H

If the TMnCMP0 register is set to 0000H, the INTTMnEQ0 signal is generated at each count clock. The value of the 16-bit counter is always 0000H.



(b) Operation if TMnCMP0 register is set to FFFFH

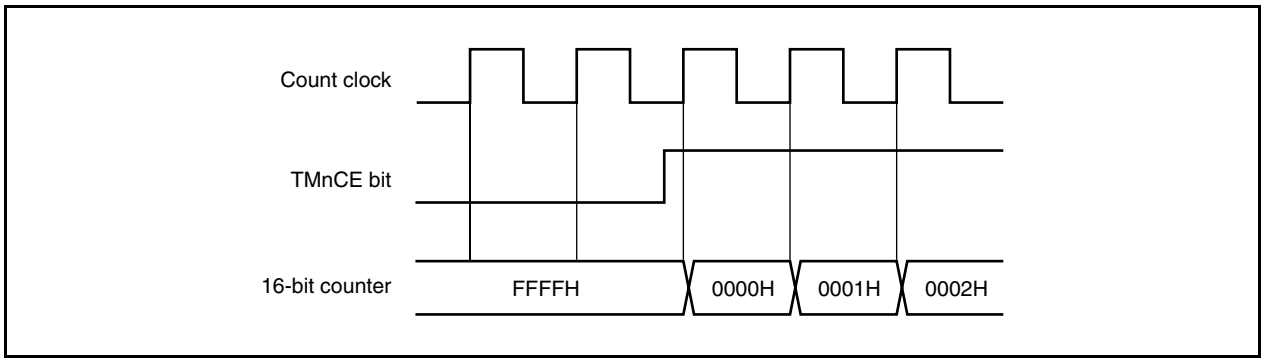
If the TMnCMP0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTMnEQ0 signal is generated.



9.5 Cautions

(1) Error on starting timer

It takes one clock to generate the first compare match interrupt request signal (INTTMnEQ0) after the TMnCTL0.TMnCE bit is set to 1 and TMMn is started. This is because the value of the 16-bit counter is FFFFH when the TMnCE bit = 0 and TMMn is started asynchronously to the count clock.



(2) Rewriting the TMnCMP0 and TMnCTL0 registers is prohibited while TMMn is operating.

If these registers are rewritten while the TMnCTL0.TMnCE bit is 1, the operation cannot be guaranteed.
If they are rewritten by mistake, clear the TMnCE bit to 0, and re-set the registers.

CHAPTER 10 MOTOR CONTROL FUNCTION

10.1 Functional Overview

Timer ABn (TABn) and the TMQn option (TMQOPn) can be used as an inverter function that controls a motor. It performs a tuning operation with timer AAn (TAAn) and A/D conversion of A/D converters 0 and 1 can be started when the value of TABn matches the value of TAAn. The following operations can be performed as motor control functions.

- 6-phase PWM output function with 16-bit accuracy (with dead-timer, for upper and lower arms)
- Timer tuning operation function (tunable with TAAn)
- Period setting function (period can be changed during operation of crest or valley interrupt)
- Compare register rewriting: Anytime rewrite, batch write, or intermittent rewrite (selectable during TABn operation)
- Interrupt and transfer culling functions
- Dead-time setting function
- A/D trigger timing function of A/D converters 0 and 1 (four types of timing can be generated)
- 0% output and 100% output available
- 0% output and 100% output selectable by crest interrupt and valley interrupt
- Forced output stop function
  - At valid edge detection by external pin input (TOBnOFF, TOAmOFF)
  - At overvoltage detection by comparator function of A/D converter
  - At main clock oscillation stop detection by clock monitor function

Remark V850E/IF3: n = 0, 1, m = 2
V850E/IG3: n = 0, 1, m = 2, 3

10.2 Configuration

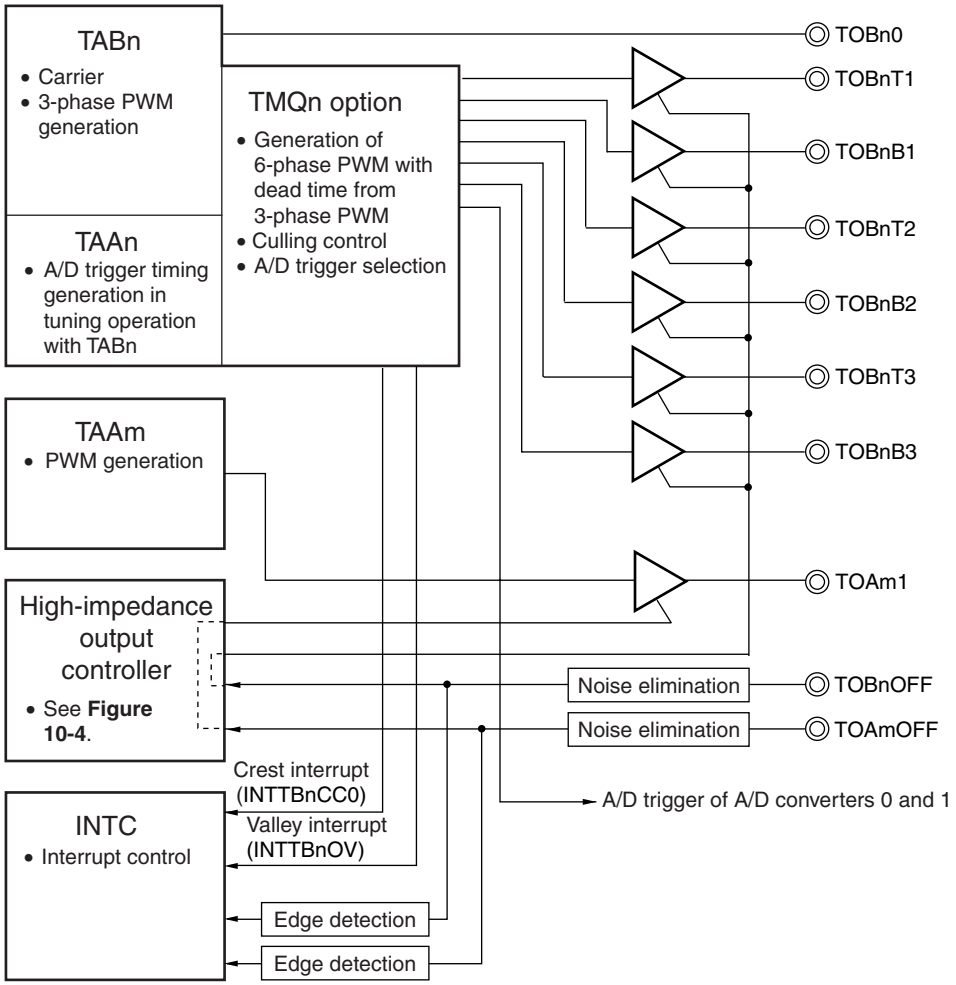
The motor control function consists of the following hardware.

Item	Configuration
Timer register	Dead-time counter m
Compare register	TABn dead-time compare register (TABnDTC register)
Control registers	TABn option register 0 (TABnOPT0) TABn option register 1 (TABnOPT1) TABn option register 2 (TABnOPT2) TABn option register 3 (TABnOPT3) TABn I/O control register 3 (TABnIOC3) High-impedance output control registers 0, 1 (HZAyCTLa)

Remark V850E/IF3: m = 0 to 3, n = 0, 1, y = 0, 2, 3, a = 0 when y = 1, a = 0, 1
V850E/IG3: m = 0 to 3, n = 0, 1, y = 0 to 3, a = 0, 1

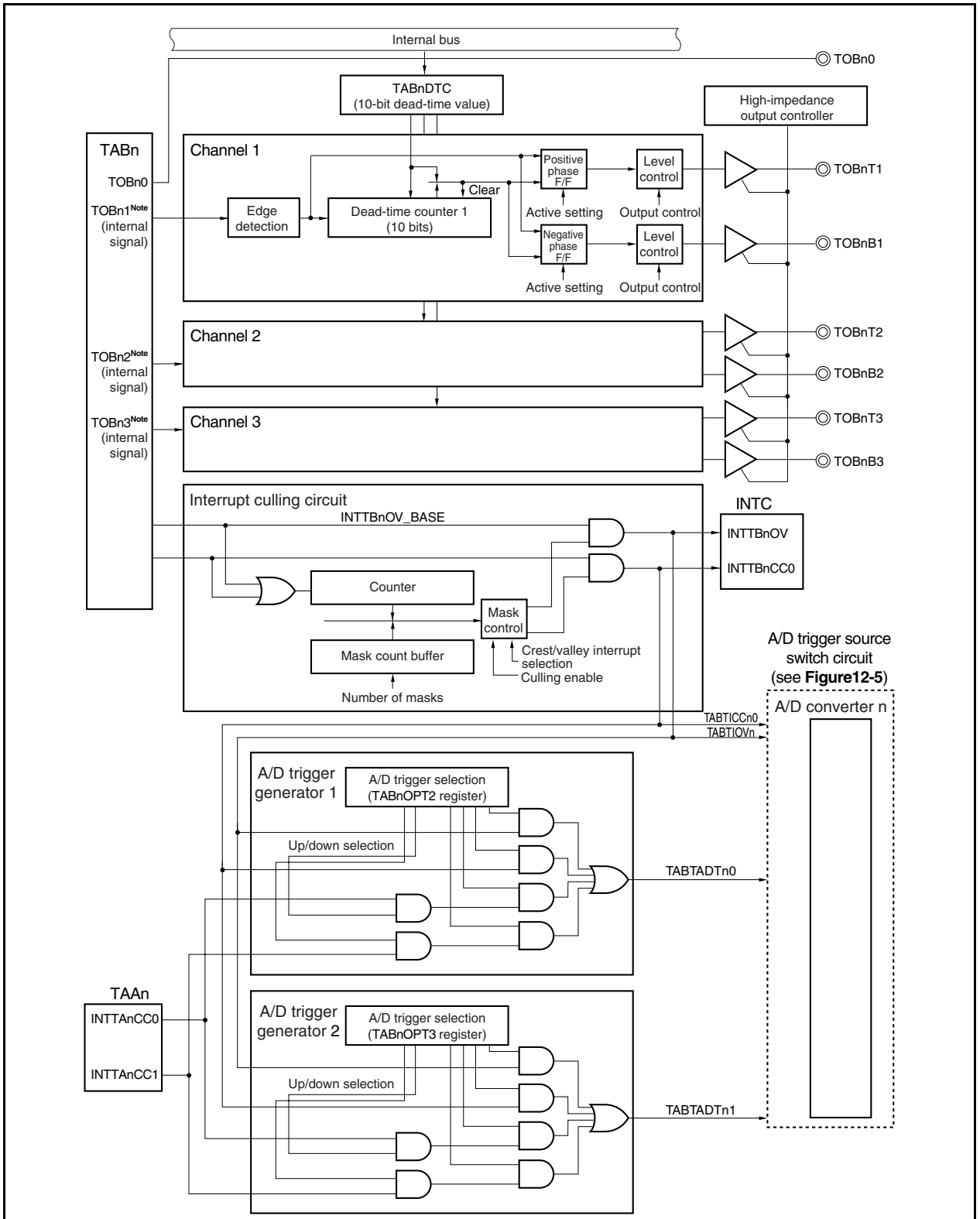
- 6-phase PWM output can be produced with dead time by using the output of TABn (TOBn1, TOBn2, TOBn3)
- The output level of the 6-phase PWM output can be set individually.
- The 16-bit timer/counter of TABn counts up/down triangular waves. When the timer/counter underflows and when a period match occurs, an interrupt is generated. Interrupt generation, however, can be suppressed up to 31 times.
- TAAn can execute counting at the same time as TABn (timer tuning operation function). TAAn can be set in four ways as it can generate two types of A/D trigger sources (INTTAnCC0 and INTTAnCC1), and two types of interrupts: on underflow interrupt of TABn (INTTBnOV) and period match interrupt (INTTBnCC0).

Figure 10-1. Block Diagram of Motor Control



Remark V850E/IF3: n = 0, 1, m = 2
V850E/IG3: n = 0, 1, m = 2, 3

Figure 10-2. TMQn Option



Note TOBn1, TOBn2, and TOBn3 function alternately as output pins.

Remark V850E/IF3: n = 0, 1, m = 2
V850E/IG3: n = 0, 1, m = 2, 3

(1) TABn dead-time compare register (TABnDTC)

The TABnDTC register is a 10-bit compare register that specifies a dead-time value.

Rewriting this register is prohibited when the TABnCTL0.TABnCE bit = 1.

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

Reset sets this register to 0000H.

Caution To generate a dead time period, set a value of 1 or greater to the TABnDTC register.
While the operation is stopped (TABnCTL0.TABnCE bit = 0), the dead time period is not generated and the output levels of the TOBnT1 to TOBnT3 and TOBnB1 to TOBnB3 pins are in the initial status. To protect the system, therefore, allow the TOBnT1 to TOBnT3 and TOBnB1 to TOBnB3 pins to go into a high-impedance state or select the port mode with setting the output levels of the pins, before stopping the operation.
If the dead time period is not necessary, set the TABnDTC register to 0.

After reset: 0000H R/W Address: TAB0DTC FFFFF604H, TAB1DTC FFFFF644H

	15	10	9	0
TABnDTC (n = 0, 1)	000000		TABnDTC9 to TABnDTC0	

(2) Dead-time counters 1 to 3

The dead-time counters are 10-bit counters that count dead time.

These counters are cleared or count up at the rising or falling edge of the TOBnm output signal by TABn, and are cleared and stopped when their count value matches the value of the TABnDTC register. The count clock of these counters is the same as that set by the TABnCTL0.TABnCKS2 to TABnCTL0.TABnCKS0 bits of TABn.

Remarks 1. The operation differs when the TABnOPT2.TABnDTM bit = 1. For details, see **10.4.2 (4) Automatic dead-time width narrowing function (TABnOPT2.TABnDTM bit = 1)**.

2. n = 0, 1, m = 1 to 3

10.3 Control Registers

(1) TABn option register 0 (TABnOPT0)

The TABnOPT0 register is an 8-bit register that controls the timer Q option function.

This register can be read or written in 8-bit or 1-bit units. However, the TABnCUF bit is read-only.

Reset sets this register to 00H.

Caution The TABnCMS and TABnCUF bits can be set only in the 6-phase PWM output mode. Be sure to clear these bits to 0 when TABn is used alone.

After reset: 00H R/W Address: TAB0OPT0 FFFFF5E5H, TAB1OPT0 FFFFF625H

TABnOPT0 (n = 0, 1)

<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
TABnCCS3Note 1	TABnCCS2Note 1	TABnCCS1Note 1	TABnCCS0Note 1	0	TABnCMS	TABnCUF	TABnOVFNote 2

TABnCMS	Compare register rewrite mode selection
0	Batch write mode (transfer operation)
1	Anytime write mode

- The TABnCMS bit is valid only when the 6-phase PWM output mode is set (when the TABnCTL1.TABnMD2 to TABnCTL1.TABnMD0 bits = 111). Clear the TABnCMS bit to 0 in any other mode.
- The TABnCMS bit can be rewritten while the timer is operating (when the TABnCTL0.TABnCE bit = 1).
- The following compare registers are rewritten in the batch write mode. TABnCCR0 to TABnCCR3, TAnCCR0, TAnCCR1, TABnOPT1, and TABnDTC registers

TABnCUF	Up-count/down-count flag of timer ABn
0	Timer ABn is counting up.
1	Timer ABn is counting down.

The TABnCUF bit is valid only when the 6-phase PWM output mode is set (when the TABnCTL1.TABnMD2 to TABnCTL1.TABnMD0 bits = 111).

Notes 1. Be sure to clear the TABnCCS3 to TABnCCS0 bits to 0 in the 6-phase PWM output mode.

2. For details of the TABnOVF bit, see **CHAPTER 7 16-BIT TIMER/EVENT COUNTER AB (TAB)**.

(2) TABn option register 1 (TABnOPT1)

The TABnOPT1 register is an 8-bit register that controls the interrupt request signal generated by the timer Qn option function.

This register can be rewritten when the TABnCTL0.TABnCE bit is 1.

Two rewriting modes (batch write mode and anytime write mode) can be selected, depending on the setting of the TABnOPT0.TABnCMS bit.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0OPT1 FFFFF600H, TAB1OPT1 FFFFF640H

	<7>	<6>	5	4	3	2	1	0
TABnOPT1 (n = 0, 1)	TABnICE	TABnIOE	0	TABnID4	TABnID3	TABnID2	TABnID1	TABnID0

TABnICE	Crest interrupt (INTTBnCC0 signal) enable
0	Do not use INTTBnCC0 signal (do not use it as count signal for interrupt culling).
1	Use INTTBnCC0 signal (use it as count signal for interrupt culling).

TABnIOE	Valley interrupt (INTTBnOV signal) enable
0	Do not use INTTBnOV signal (do not use it as count signal for interrupt culling).
1	Use INTTBnOV signal (use it as count signal for interrupt culling).

TABnID4	TABnID3	TABnID2	TABnID1	TABnID0	Number of times of interrupt
0	0	0	0	0	Not culled (all interrupts are output)
0	0	0	0	1	1 masked (one of two interrupts is output)
0	0	0	1	0	2 masked (one of three interrupts is output)
0	0	0	1	1	3 masked (one of four interrupts is output)
:	:	:	:	:	:
1	1	1	0	0	28 masked (one of 29 interrupts is output)
1	1	1	0	1	29 masked (one of 30 interrupts is output)
1	1	1	1	0	30 masked (one of 31 interrupts is output)
1	1	1	1	1	31 masked (one of 32 interrupts is output)

(3) TABn option register 2 (TABnOPT2)

The TABnOPT2 register is an 8-bit register that controls the timer Q option function.

This register can be rewritten when the TABnCTL0.TABnCE bit is 1. However, rewriting the TABnDTM bit is prohibited when the TABnCE bit is 1. The same value can be rewritten.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: TAB0OPT2 FFFFF601H, TAB1OPT2 FFFFF641H

TABnOPT2 (n = 0, 1; m = 1 to 3)

<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TABnRDE	TABnDTM	TABnATM3	TABnATM2	TABnAT3	TABnAT2	TABnAT1	TABnAT0

TABnRDE	Transfer culling enable
0	Do not cull transfer (transfer timing is generated every time at crest and valley).
1	Cull transfer at the same interval as interrupt culling set by the TABnOPT1 register.

TABnDTM	Dead-time counter operation mode selection
0	Dead-time counter counts up normally and, if TOBnm output of TABn is at a narrow interval (TOBnm output width < dead-time width), the dead-time counter is cleared and counts up again.
1	Dead-time counter counts up normally and, if TOBnm output of TABn is at a narrow interval (TOBnm output width < dead-time width), the dead-time counter counts down and the dead-time control width is automatically narrowed.
Rewriting the TABnDTM bit is disabled during timer operation. If it is rewritten by mistake, stop the timer operation by clearing the TABnCE bit to 0, and re-set the TABnDTM bit.	

Cautions 1. When using interrupt culling (the TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits are set to other than 00000), be sure to set the TABnRDE bit to 1.
Therefore, the interrupt and transfer are generated at the same timing. The interrupt and transfer cannot be set separately. If the interrupt and transfer are set separately (TABnRDE bit = 0), transfer is not performed normally.

2. To generate a dead time period, set a value 1 or greater to the TABnDTC register.
While the operation is stopped (TABnCTL0.TABnCE bit = 0), the dead time period is not generated and the output levels of the TOBnT1 to TOBnT3 and TOBnB1 to TOBnB3 pins are in the initial status. To protect the system, therefore, allow the TOBnT1 to TOBnT3 and TOBnB1 to TOBnB3 pins go into a high-impedance state or select the port mode with setting the output levels of the pins, before stopping the operation.
If the dead time period is not necessary, set the TABnDTC register to 0.

(2/2)

TABnATM3	TABnATM3 mode selection
0	Output A/D trigger signal (TABTADTn0) for INTTAnCC1 interrupt while dead-time counter is counting up.
1	Output A/D trigger signal (TABTADTn0) for INTTAnCC1 interrupt while dead-time counter is counting down.

TABnATM2	TABnATM2 mode selection
0	Output A/D trigger signal (TABTADTn0) for INTTAnCC0 interrupt while dead-time counter is counting up.
1	Output A/D trigger signal (TABTADTn0) for INTTAnCC0 interrupt while dead-time counter is counting down.

TABnAT3Note	A/D trigger output control 3
0	Disable output of A/D trigger signal (TABTADTn0) for INTTAnCC1 interrupt.
1	Enable output of A/D trigger signal (TABTADTn0) for INTTAnCC1 interrupt.

TABnAT2Note	A/D trigger output control 2
0	Disable output of A/D trigger signal (TABTADTn0) for INTTAnCC0 interrupt.
1	Enable output of A/D trigger signal (TABTADTn0) for INTTAnCC0 interrupt.

TABnAT1Note	A/D trigger output control 1
0	Disable output of A/D trigger signal (TABTADTn0) for INTTBnCC0 (crest interrupt).
1	Enable output of A/D trigger signal (TABTADTn0) for INTTBnCC0 (crest interrupt).

TABnAT0Note	A/D trigger output control 0
0	Disable output of A/D trigger signal (TABTADTn0) for INTTBnOV (valley interrupt).
1	Enable output of A/D trigger signal (TABTADTn0) for INTTBnOV (valley interrupt).

Note For the setting of the TABnAT3 to TABnAT0 bits, see **CHAPTER 12 A/D CONVERTERS 0 AND 1**.

(4) TABn option register 3 (TABnOPT3)

The TABnOPT3 register is an 8-bit register that controls the timer Qn option function.

This register can be rewritten when the TABnCTL0.TABnCE bit is 1.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0OPT3 FFFFF603H, TAB1OPT3 FFFFF643H

	7	6	<5>	<4>	<3>	<2>	<1>	<0>
TABnOPT3	0	0	TABnATM7	TABnATM6	TABnAT7	TABnAT6	TABnAT5	TABnAT4

(n = 0, 1)

TABnATM7	TABnATM7 mode selection
0	Output A/D trigger signal (TABTADTn1) of INTTAnCC1 interrupt while dead-time counter is counting up.
1	Output A/D trigger signal (TABTADTn1) of INTTAnCC1 interrupt while dead-time counter is counting down.

TABnATM6	TABnATM6 mode selection
0	Output A/D trigger signal (TABTADTn1) of INTTAnCC0 interrupt while dead-time counter is counting up.
1	Output A/D trigger signal (TABTADTn1) of INTTAnCC0 interrupt while dead-time counter is counting down.

TABnAT7Note	A/D trigger output control 3
0	Disable output of A/D trigger signal (TABTADTn1) for INTTAnCC1 interrupt.
1	Enable output of A/D trigger signal (TABTADTn1) for INTTAnCC1 interrupt.

TABnAT6Note	A/D trigger output control 2
0	Disable output of A/D trigger signal (TABTADTn1) for INTTAnCC0 interrupt.
1	Enable output of A/D trigger signal (TABTADTn1) for INTTAnCC0 interrupt.

TABnAT5Note	A/D trigger output control 1
0	Disable output of A/D trigger signal (TABTADTn1) for INTTBnCC0 interrupt (crest interrupt).
1	Enable output of A/D trigger signal (TABTADTn1) for INTTBnCC0 interrupt (crest interrupt).

TABnAT4Note	A/D trigger output control 0
0	Disable output of A/D trigger signal (TABTADTn1) for INTTBnOV interrupt (valley interrupt).
1	Enable output of A/D trigger signal (TABTADTn1) for INTTBnOV interrupt (valley interrupt).

Note For the setting of the TABnAT7 to TABnAT4 bits, see **CHAPTER 12 A/D CONVERTERS 0 AND 1**.

(5) TABn I/O control register 3 (TABnIOC3)

The TABnIOC3 register is an 8-bit register that controls the output of the timer Qn option function.

To output from the TOBnTm pin, set the TABnIOC0.TABnOEm bit to 1 and then set the TABnIOC3 register.

The TABnIOC3 register can be rewritten only when the TABnCTL0.TABnCE bit is 0.

Rewriting each bit of the TABnIOC3 register is prohibited when the TABnCTL0.TABnCE bit is 1; however the same value can be rewritten to each bit of the TABnIOC3 register when the TABnCTL0.TABnCE bit is 1.

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

Reset sets this register to A8H.

Caution Set the TABnIOC3 register to the default value (A8H) when the timer is used in a mode other than the 6-phase PWM output mode.

Remark Set the output level of the TOBnTm pin by the TABnIOC0 register.

After reset: A8H R/W Address: TAB0IOC3 FFFFF602H, TAB1IOC3 FFFFF642H

TABnIOC3 (n = 0, 1; m = 1 to 3)

<7>	<6>	<5>	<4>	<3>	<2>	1	0
TABnOLB3	TABnOEB3	TABnOLB2	TABnOEB2	TABnOLB1	TABnOEB1	0	0

TABnOLBm	Setting of TOBnBm pin output level
0	Disable inversion of output of TOBnBm pin
1	Enable inversion of output of TOBnBm pin

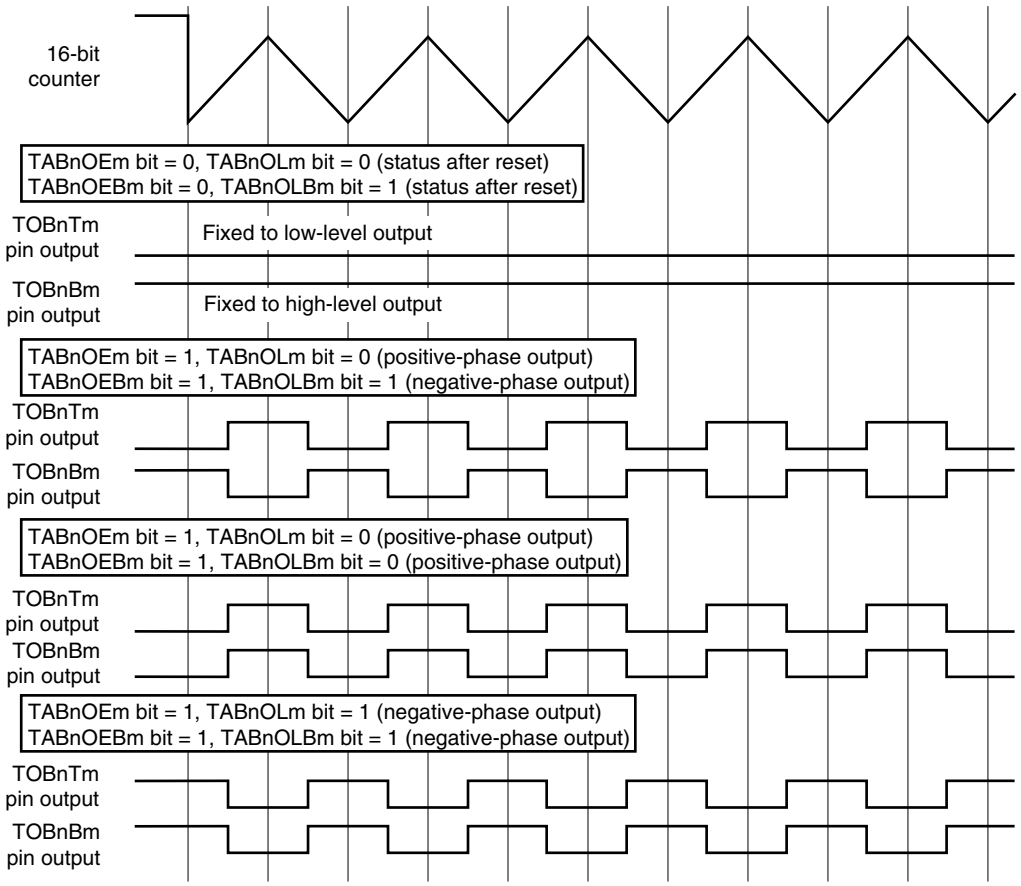
TABnOEBm	Setting of TOBnBm pin output
0	Disable TOBnBm pin output. • When TABnOLBm bit = 0, low level is output from TOBnBm pin. • When TABnOLBm bit = 1, high level is output from TOBnBm pin.
1	Enable TOBnBm pin output.

(a) Output from TOBnTm and TOBnBm pins

The TOBnTm pin output is controlled by the TABnIOC0.TABnOLm and TABnIOC0.TABnOEm bits. The TOBnBm pin output is controlled by the TABnIOC3.TABnOLBm and TABnIOC3.TABnOEBm bits.

A timer output with each setting in the 6-phase PWM output mode is shown below.

Figure 10-3. TOBnTm and TOBnBm Pin Output Control (Without Dead Time)



Remark n = 0, 1

m = 1 to 3

Table 10-1. TOBnTm Pin Output

TABnOLm Bit	TABnOEm Bit	TABnCE Bit	TOBnTm Pin Output
0	0	x	Low-level output
	1	0	Low-level output
		1	TOBnTm positive-phase output
1	0	x	High-level output
	1	0	High-level output
		1	TOBnTm negative-phase output

Remark n = 0, 1
m = 1 to 3

Table 10-2. TOBnBm Pin Output

TABnOLBm Bit	TABnOEBm Bit	TABnCE Bit	TOBnBm Pin Output
0	0	x	Low-level output
	1	0	Low-level output
		1	TOBnBm positive-phase output
1	0	x	High-level output
	1	0	High-level output
		1	TOBnBm negative-phase output

Remark n = 0, 1
m = 1 to 3

(6) High-impedance output control registers 00, 01, 10, 11, 20, 21, 30, 31 (HZAyCTL0, HZAyCTL1)

The HZAyCTL0 and HZAyCTL1 registers are 8-bit registers that control the high-impedance state of the output buffer.

These registers can be read or written in 8-bit or 1-bit units. However, the HZAyDCFn bit is a read-only bit and cannot be written.

16-bit access is not possible.

Reset sets these registers to 00H.

The same value can be always rewritten to the HZAyCTLn register by software.

The relationship between detection factor and the control registers is shown below.

Pins Subject to High-Impedance Control	High-Impedance Control Factor		Control Register
	External Pin	A/D Unit (Comparator)	
When TOB0T1 to TOB0T3 are output When TOB0B1 to TOB0B3 are output	TOB0OFF	–	HZA0CTL0
	–	When low range reference voltage of ANI00/ANI05 input is exceeded (rising edge) or not attained (falling edge)	HZA2CTL0
	–	When full range reference voltage of ANI00/ANI05 input is exceeded (rising edge) or not attained (falling edge)	HZA2CTL1
When TOA21 is output	TOA2OFF	–	HZA0CTL1
When TOB1T1 to TOB1T3 are output When TOB1B1 to TOB1B3 are output	TOB1OFF	–	HZA1CTL0[Note]
	–	When low range reference voltage of ANI10 to ANI12 and ANI15 to ANI17 inputs is exceeded (rising edge) or not attained (falling edge)	HZA3CTL0
	–	When full range reference voltage of ANI10 to ANI12 and ANI15 to ANI17 inputs is exceeded (rising edge) or not attained (falling edge)	HZA3CTL1
When TOA31[Note] is output	TOA3OFF[Note]	–	HZA1CTL1[Note]

Note V850E/IG3 only

Caution High-impedance control is performed only when a port pin is set to function as indicated in the above table.

(1/3)

After reset: 00H R/W Address: HZA0CTL0 FFFFF610H, HZA0CTL1 FFFFF611H,
HZA1CTL0 FFFFF618H, HZA1CTL1 FFFFF619H[Note 1],
HZA2CTL0 FFFFF650H, HZA2CTL1 FFFFF651H,
HZA3CTL0 FFFFF658H, HZA3CTL1 FFFFF659H

HZAyCTLn

<7>	<6>	5	4	<3>	<2>	1	<0>
HZAyDCEn	HZAyDCMn	HZAyDCNn	HZAyDCPn	HZAyDCTn	HZAyDCCn	0	HZAyDCFn

(V850E/IF3: n = 0, 1; y = 0, 2, 3; n = 0 when y = 1)

(V850E/IG3: n = 0, 1; y = 0 to 3)

HZAyDCEn	High-impedance output control
0	Disable high-impedance output control operation. Pins can function as output pins.
1	Enable high-impedance output control operation.

HZAyDCMn	Condition of clearing high-impedance state by HZAyDCCn bit
0	Setting of the HZAyDCCn bit is valid regardless of the external pin[Note 2] input.
1	Setting of the HZAyDCCn bit is invalid while the external pin[Note 2] input holds a level detected as abnormal (active level).
Rewrite the HZAyDCMn bit when the HZAyDCEn bit = 0.	

Notes 1. V850E/IG3 only

2. • V850E/IF3

HZA0CTL0: TOB0OFF pin, HZA0CTL1: TOA2OFF pin,
HZA1CTL0: TOB1OFF pin,
HZA2CTL0: ANI00/ANI05 pin, HZA2CTL1: ANI00/ANI05 pin,
HZA3CTL0: ANI10 to ANI12, ANI15 to ANI17 pins,
HZA3CTL1: ANI10 to ANI12, ANI15 to ANI17 pins

• V850E/IG3

HZA0CTL0: TOB0OFF pin, HZA0CTL1: TOA2OFF pin,
HZA1CTL0: TOB1OFF pin, HZA1CTL1: TOA3OFF pin,
HZA2CTL0: ANI00/ANI05 pin, HZA2CTL1: ANI00/ANI05 pin,
HZA3CTL0: ANI10 to ANI12, ANI15 to ANI17 pins,
HZA3CTL1: ANI10 to ANI12, ANI15 to ANI17 pins

(2/3)

HZAyDCNn	HZAyDCPn	External pin[Note 1] input edge specification
0	0	No valid edge (setting the HZAyDCFn bit by external pin[Note 1] input is prohibited).
0	1	Rising edge of the external pin[Note 1] input is valid (abnormality is detected by rising edge input)[Note 2].
1	0	Falling edge of the external pin[Note 1] input is valid (abnormality is detected by falling edge input)[Note 2].
1	1	Setting prohibited

- Rewrite the HZAyDCNn and HZAyDCPn bits when the HZAyDCEn bit is 0.
- For the edge specification of the INTP00, INTP02, INTP08, and INTP10 pins, see **20.4.2 (1) External interrupt rising edge specification register 0 (INTR0), external interrupt falling edge specification register 0 (INTF0)** and **(2) External interrupt rising edge specification register 1 (INTR1), external interrupt falling edge specification register 1 (INTF1)**.
- The edge of the external pins must be specified starting from the TOB0OFF, TOB1OFF, TOA2OFF, and TOA3OFF[Note 3] pins. Then the edge of the external pins other than the TOB0OFF, TOB1OFF, TOA2OFF, and TOA3OFF[Note 3] pins must be specified. Otherwise, the undefined edge may be detected when the edges of the TOB0OFF, TOB1OFF, TOA2OFF, and TOA3OFF[Note 3] pins are specified.
- High-impedance output control is performed when the valid edge is input after the operation is enabled (by setting HZAyDCEn bit to 1). If the external pin[Note 1] is at the active level when the operation is enabled, therefore, high-impedance output control is not performed.

Notes 1. • V850E/IF3

HZA0CTL0: TOB0OFF pin, HZA0CTL1: TOA2OFF pin,
HZA1CTL0: TOB1OFF pin,
HZA2CTL0: ANI00/ANI05 pin, HZA2CTL1: ANI00/ANI05 pin,
HZA3CTL0: ANI10 to ANI12, ANI15 to ANI17 pins,
HZA3CTL1: ANI10 to ANI12, ANI15 to ANI17 pins

- V850E/IG3

HZA0CTL0: TOB0OFF pin, HZA0CTL1: TOA2OFF pin,
HZA1CTL0: TOB1OFF pin, HZA1CTL1: TOA3OFF pin,
HZA2CTL0: ANI00/ANI05 pin, HZA2CTL1: ANI00/ANI05 pin,
HZA3CTL0: ANI10 to ANI12, ANI15 to ANI17 pins,
HZA3CTL1: ANI10 to ANI12, ANI15 to ANI17 pins

2. For detecting the voltage of a comparator exceeding the reference voltage, set the rising edge input. For detecting the voltage of a comparator which has not attained the reference voltage, set the falling edge input.

(3/3)

HZAyDCTn	High-impedance output trigger bit
0	No operation
1	Pins are made to go into a high-impedance state by software and the HZAyDCFn bit is set to 1.

- If an edge indicating abnormality is input to the external pin[Note 2] (which is detected according to the setting of the HZAyDCNn and HZAyDCPn bits), the HZAyDCTn bit is invalid even if it is set to 1.
- The HZAyDCTn bit is always 0 when it is read because it is a software-triggered bit.
- The HZAyDCTn bit is invalid even if it is set to 1 when the HZAyDCEn bit = 0.
- Simultaneously setting the HZAyDCTn and HZAyDCCn bits to 1 is prohibited.

HZAyDCCn	High-impedance output control clear bit
0	No operation
1	Pins that have gone into a high-impedance state are output-enabled by software and the HZAyDCFn bit is cleared to 0.

- Pins can function as output pins when the HZAyDCM bit = 0, regardless of the status of the external pin[Note].
- If an edge indicating abnormality is input to the external pin[Note] (which is set by the HZAyDCNn and HZAyDCPn bits) when the HZAyDCM bit = 1, the HZAyDCCn bit is invalid even if it is set to 1.
- The HZAyDCCn bit is always 0 when it is read.
- The HZAyDCCn bit is invalid even if it is set to 1 when the HZAyDCEn bit = 0.
- Simultaneously setting the HZAyDCTn and HZAyDCCn bits to 1 is prohibited.

HZAyDCFn	High-impedance output status flag
0	Indicates that output of the pin is enabled. • This bit is cleared to 0 when the HZAyDCEn bit = 0. • This bit is cleared to 0 when the HZAyDCCn bit = 1.
1	Indicates that the pin goes into a high-impedance state. • This bit is set to 1 when the HZAyDCTn bit = 1. • This bit is set to 1 when an edge indicating abnormality is input to the external pin[Note] (which is detected according to the setting of the HZAyDCNn and HZAyDCPn bits).

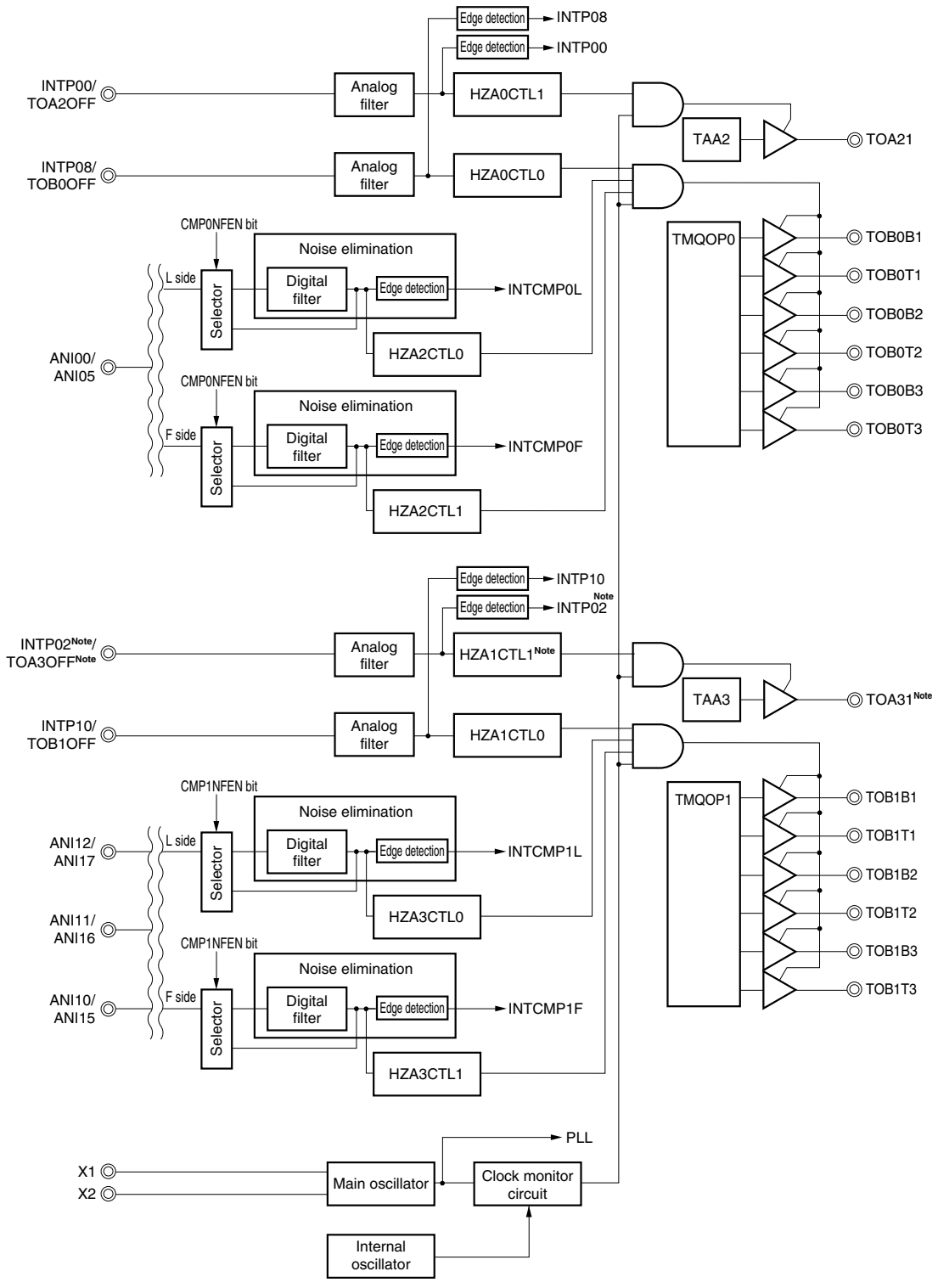
Note • V850E/IF3

HZA0CTL0: TOB0OFF pin, HZA0CTL1: TOA2OFF pin,
HZA1CTL0: TOB1OFF pin,
HZA2CTL0: ANI00/ANI05 pin, HZA2CTL1: ANI00/ANI05 pin,
HZA3CTL0: ANI10 to ANI12, ANI15 to ANI17 pins,
HZA3CTL1: ANI10 to ANI12, ANI15 to ANI17 pins

• V850E/IG3

HZA0CTL0: TOB0OFF pin, HZA0CTL1: TOA2OFF pin,
HZA1CTL0: TOB1OFF pin, HZA1CTL1: TOA3OFF pin,
HZA2CTL0: ANI00/ANI05 pin, HZA2CTL1: ANI00/ANI05 pin,
HZA3CTL0: ANI10 to ANI12, ANI15 to ANI17 pins,
HZA3CTL1: ANI10 to ANI12, ANI15 to ANI17 pins

Figure 10-4. High-Impedance Output Controller Configuration



Note V850E/IG3 only

Remark When referring to **Figure 10-4**, also refer to **Figures 12-3** and **12-4**.

(a) Setting procedure

(i) Setting of high-impedance control operation

<1> Set the HZAyDCMn, HZAyDCNn, and HZAyDCPn bits.

<2> Set the HZAyDCEn bit to 1 (enable high-impedance control).

(ii) Changing setting after enabling high-impedance control operation

<1> Clear the HZAyDCEn bit to 0 (to stop the high-impedance control operation).

<2> Change the setting of the HZAyDCMn, HZAyDCNn, and HZAyDCPn bits.

<3> Set the HZAyDCEn bit to 1 (to enable the high-impedance control operation again).

(iii) Resuming output when pins are in high-impedance state

If the HZAyDCMn bit is 1, set the HZAyDCCn bit to 1 to clear the high-impedance state after the valid edge of the external pin[Note] is detected. However, the high-impedance state cannot be cleared unless this bit is set while the input level of the external pin[Note] is inactive.

<1> Set the HZAyDCCn bit to 1 (command signal to clear the high-impedance state).

<2> Read the HZAyDCFn bit and check the flag status.

<3> Return to <1> if the HZAyDCFn bit is 1. The input level of the external pin[Note] must be checked. The pin can function as an output pin if the HZAyDCFn bit is 0.

(iv) To make the pin to go into a high-impedance state by software

The HZAyDCTn bit must be set to 1 by software to make the pin to go into a high-impedance state while the input level of the external pin[Note] is inactive. The following procedure is an example in which the setting is not dependent upon the setting of the HZAyDCMn bit.

<1> Set the HZAyDCTn bit to 1 (high-impedance output command).

<2> Read the HZAyDCFn bit to check the flag status.

<3> Return to <1> if the HZAyDCFn bit is 0. The input level of the external pin[Note] must be checked. The pin is in a high-impedance state if the HZAyDCFn bit is 1.

However, if the external pin[Note] is not used with the HZAyDCPn bit and HZAyDCNn bit cleared to 0, the pin goes into a high-impedance state when the HZAyDCTn bit is set to 1.

Note • V850E/IF3

HZA0CTL0: TOB0OFF pin, HZA0CTL1: TOA2OFF pin,
HZA1CTL0: TOB1OFF pin,
HZA2CTL0: ANI00/ANI05 pin, HZA2CTL1: ANI00/ANI05 pin,
HZA3CTL0: ANI10 to ANI12, ANI15 to ANI17 pins,
HZA3CTL1: ANI10 to ANI12, ANI15 to ANI17 pins

• V850E/IG3

HZA0CTL0: TOB0OFF pin, HZA0CTL1: TOA2OFF pin,
HZA1CTL0: TOB1OFF pin, HZA1CTL1: TOA3OFF pin,
HZA2CTL0: ANI00/ANI05 pin, HZA2CTL1: ANI00/ANI05 pin,
HZA3CTL0: ANI10 to ANI12, ANI15 to ANI17 pins,
HZA3CTL1: ANI10 to ANI12, ANI15 to ANI17 pins

10.4 Operation

10.4.1 System outline

(1) Outline of 6-phase PWM output

The 6-phase PWM output mode is used to generate a 6-phase PWM output waveform, by using TABn and the TABn option in combination.

The 6-phase PWM output mode is enabled by setting the TABnCTL1.TABnMD2 to TABnCTL1.TABnMD0 bits of TABn to “111”.

One 16-bit counter and four 16-bit compare registers of TABn are used to generate a basic 3-phase wave.

The functions of the compare registers are as follows.

TAAn can perform a tuning operation with TABn to start a conversion trigger source for A/D converters 0 and 1.

Remark n = 0, 1

Compare Register	Function	Settable Range
TABnCCR0 register	Setting of cycle	0002H ≤ m ≤ FFFEH
TABnCCR1 register	Specifying output width of phase U	0000H ≤ i ≤ m + 1
TABnCCR2 register	Specifying output width of phase V	0000H ≤ j ≤ m + 1
TABnCCR3 register	Specifying output width of phase W	0000H ≤ k ≤ m + 1

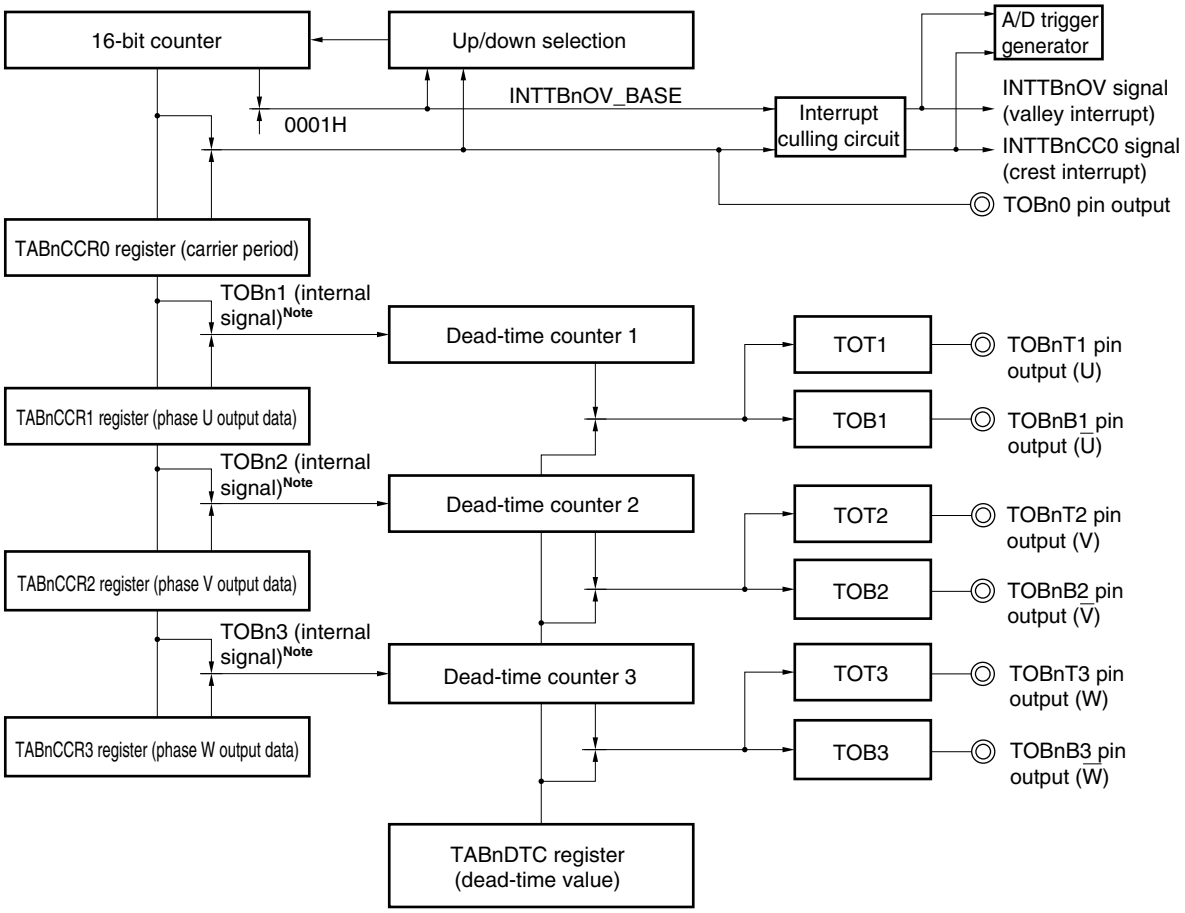
Remark m = Set value of TABnCCR0 register
i = Set value of TABnCCR1 register
j = Set value of TABnCCR2 register
k = Set value of TABnCCR3 register

A dead-time interval is generated from the basic 3-phase wave generated by using three 10-bit dead-time counters and one compare register to create a wave with a reverse phase to that of the basic 3-phase wave. Then a 6-phase PWM output waveform (U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$) is generated.

The 16-bit counter for generating the basic 3-phase wave counts up or down. After the operation has been started, this counter counts up. When its count value matches the cycle set to the TABnCCR0 register, the counter starts counting down. When the count value matches 0001H, the counter counts up again. This means that a value two times higher than the value set to the TABnCCR0 register +1 is the carrier cycle.

10-bit dead-time counters 1 to 3 that generate the dead-time interval count up. Therefore, the value set to the TABn dead-time compare register (TABnDTC) is used as a dead-time value as is. Because three counters are used, dead time can be generated independently in phases U, V, and W. However, because there is only one register that specifies a dead-time value (TABnDTC), the same dead-time value is used in the three phases.

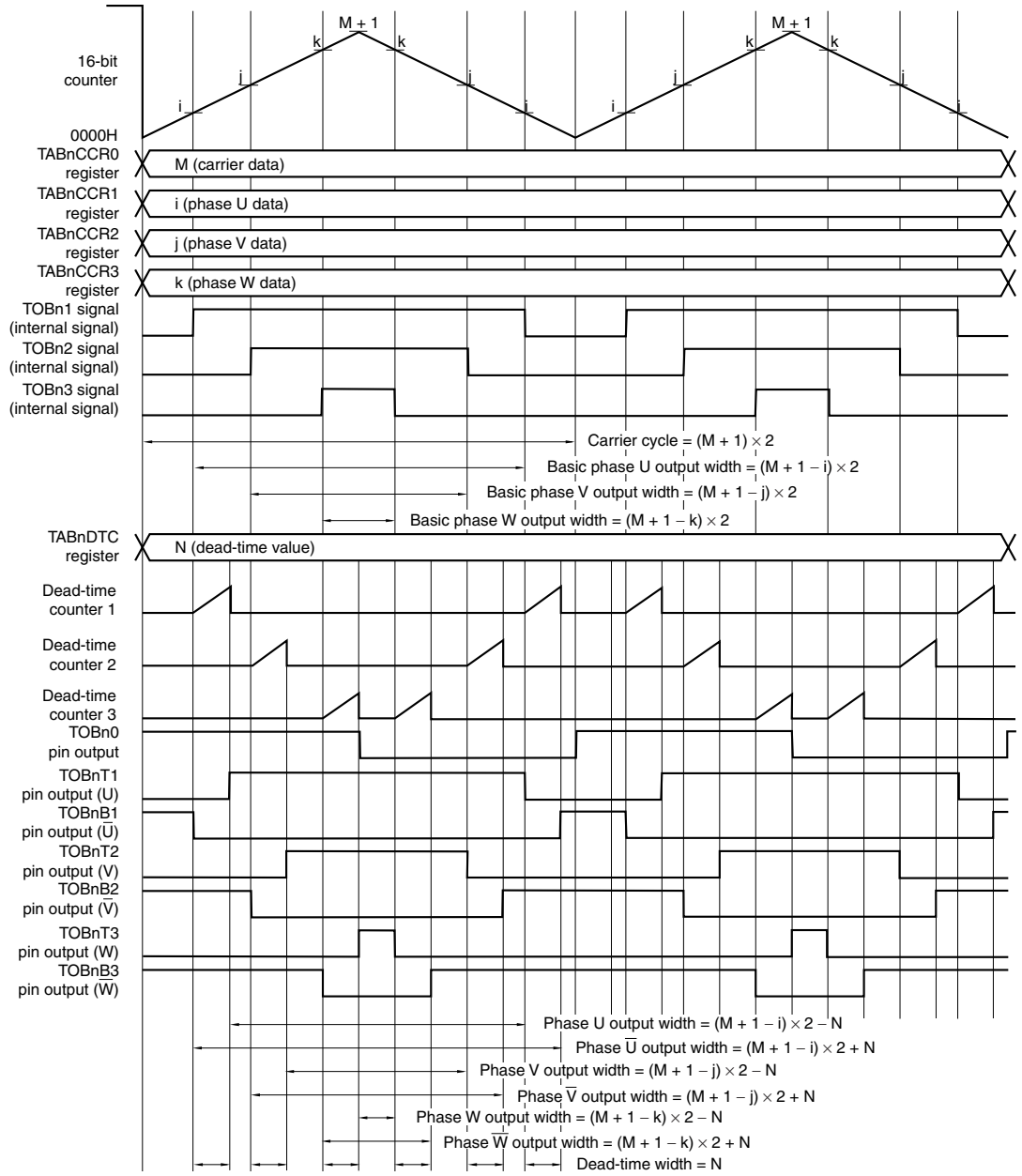
Figure 10-5. Outline of 6-Phase PWM Output Mode



Note TOBn1, TOBn2, and TOBn3 function alternately as output pins.

Remark n = 0, 1

Figure 10-6. Timing Chart of 6-Phase PWM Output Mode



Cautions 1. Set the value "M" of the TABnCCR0 register in a range of 0002H ≤ M ≤ FFFEH in the 6-phase PWM output mode.

2. Only a value of up to "M + 1" can be set to the TABnCCR1, TABnCCR2, and TABnCCR3 registers.

3. The output is 100% if "0000H" is set to the TABnCCR1, TABnCCR2, and TABnCCR3 registers. The output is 0% if "M + 1" is set to the TABnCCR1, TABnCCR2, and TABnCCR3 registers.
The output (duty 50%) rises at the crest (M + 1) of the 16-bit counter and falls at the valley (0000H) if "M + 2" or higher is set to the TABnCCR1, TABnCCR2, and TABnCCR3 registers.

4. If the operation value of an equation (such as (M + 1 − i) × 2 − N) of the output width of phases U, V, and W is 0 or lower, it is converged to 0 (100% output). If the operation value is higher than "(M + 1) × 2", it is converged to (M + 1) × 2 (0% output).

Remark n = 0, 1

(2) Interrupt requests

Two types of interrupt requests are available: the INTTBnCC0 (crest interrupt) signal and INTTBnOV (valley interrupt) signal.

The INTTBnCC0 and INTTBnOV signals can be culled by using the TABnOPT1 register.

For details of culling interrupts, see **10.4.3 Interrupt culling function**.

- INTTBnCC0 (crest interrupt) signal: Interrupt signal indicating matching between the value of the 16-bit counter that counts up and the value of the TABnCCR0 register
- INTTBnOV (valley interrupt) signal: Interrupt signal indicating matching between the value of the 16-bit counter that counts down and the value 0001H

(3) Rewriting registers during timer operation

The following registers have a buffer register and can be rewritten in the anytime rewriting mode, batch rewrite mode, or intermittent batch rewrite mode.

Related Unit	Register
Timer AAn	TAAn capture/compare register 0 (TAAnCCR0) TAAn capture/compare register 1 (TAAnCCR1)
Timer ABn	TABn capture/compare register 0 (TABnCCR0) TABn capture/compare register 1 (TABnCCR1) TABn capture/compare register 2 (TABnCCR2) TABn capture/compare register 3 (TABnCCR3)
Timer Qn option	TABn option register 1 (TABnOPT1)

Remark n = 0, 1

For details of the transfer function of the compare register, see **10.4.4 Operation to rewrite register with transfer function**.

(4) Counting-up/-down operation of 16-bit counter

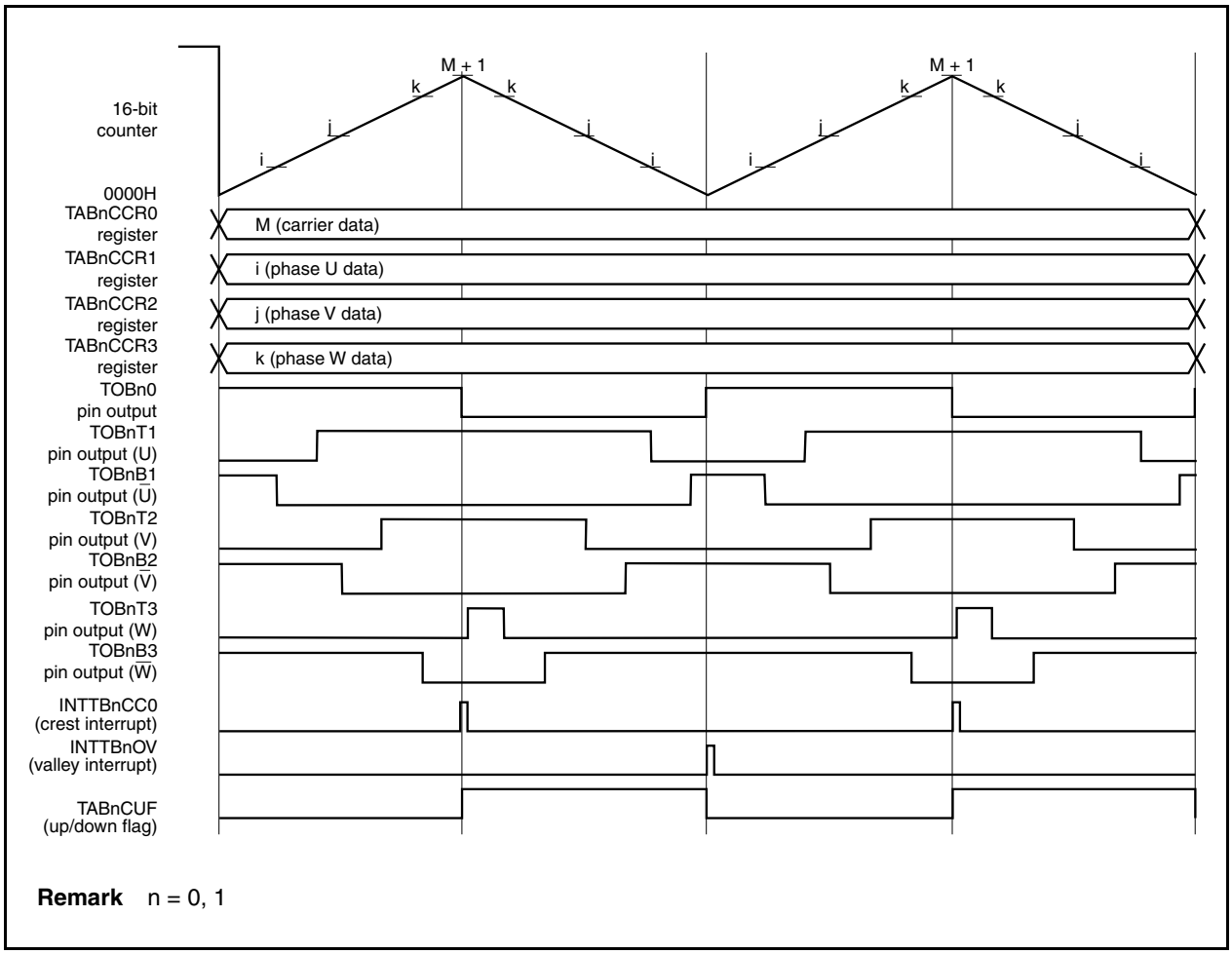
The operation status of the 16-bit counter can be checked by using the TABnCUF bit of TABn option register 0 (TABnOPT0).

Status of TABnCUF Bit	Status of 16-Bit Counter	Range of 16-Bit Counter Value
TABnCUF bit = 0	Counting up	0000H – m
TABnCUF bit = 1	Counting down	(m+1) – 0001H

Remarks 1. m = Set value of TABnCCR0 register

2. n = 0, 1

Figure 10-7. Interrupt and Up/Down Flag



Remark n = 0, 1

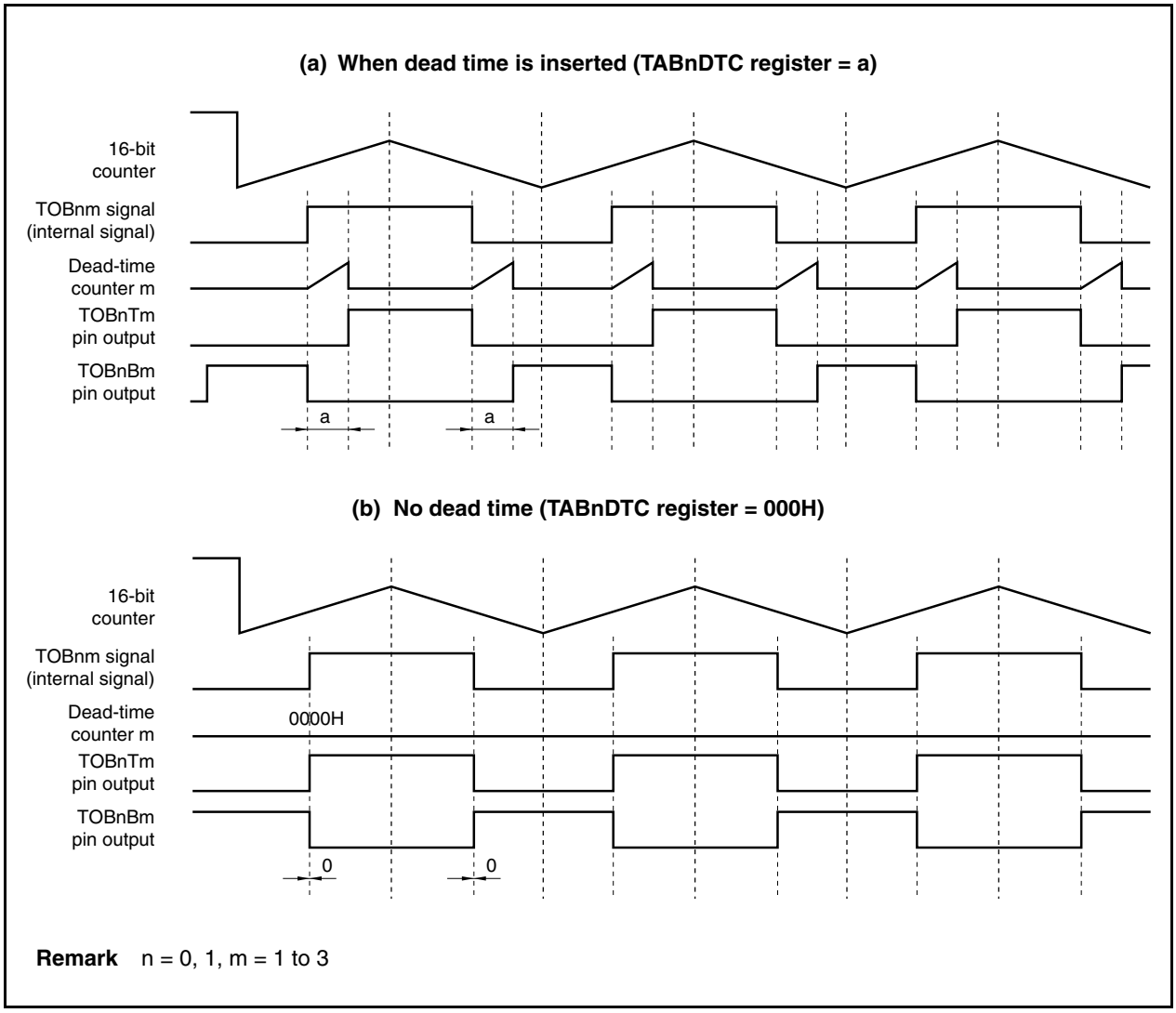
10.4.2 Dead-time control (generation of negative-phase wave signal)

(1) Dead-time control mechanism

In the 6-phase PWM output mode, compare registers 1 to 3 (TABnCCR1, TABnCCR2, and TABnCCR3) are used to set the duty factor, and compare register 0 (TABnCCR0) is used to set the cycle. By setting these four registers and by starting the operation of TAB, three types of PWM output waves (basic 3-phase waves) with a variable duty factor are generated. These three PWM output waves are input to the timer Q option unit (TMQOPn) and their inverted signal with dead-time is created to generate three sets of (six) PWM waves.

The TMQOPn unit consists of three 10-bit counters (dead-time counters 1 to 3) that operate in synchronization with the count clock of TABn, and a TABn dead-time compare register (TABnDTC) that specifies dead time. If "a" is set to the TABnDTC register, the dead-time value is "a", and interval "a" is created between a positive-phase wave and a negative-phase wave.

Figure 10-8. PWM Output Waveform with Dead Time (1)



Remark n = 0, 1, m = 1 to 3

(2) PWM output of 0%/100%

The V850E/IF3 and V850E/IG3 are capable of 0% waveform output and 100% waveform output for PWM output.

A low level is continuously output from TOBnTm pin as the 0% waveform output. A high level is continuously output from TOBnTm pin as the 100% waveform output.

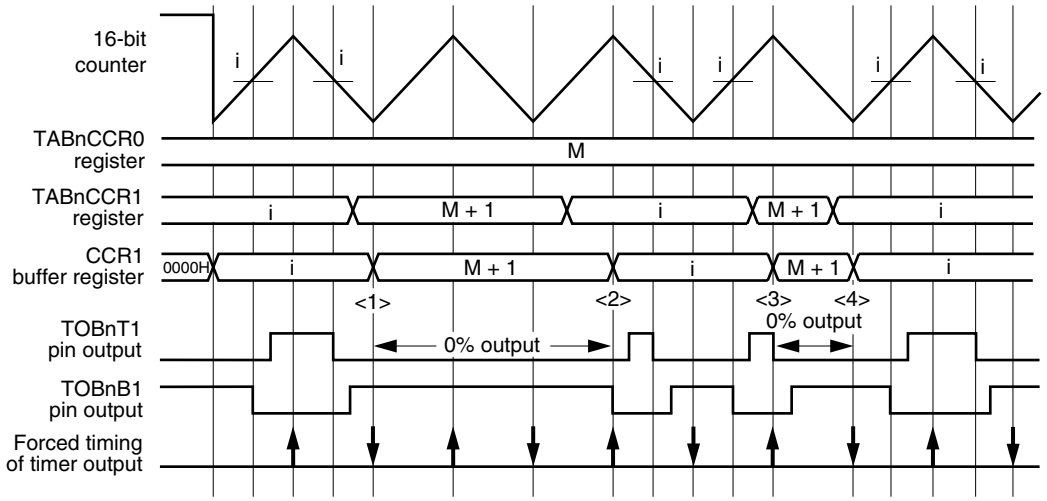
The 0% waveform is output by setting the TABnCCRm register to "M + 1" when the TABnCCR0 register = M.

The 100% waveform is output by setting the TABnCCRm register to "0000H".

Rewriting the TABnCCRm register is enabled while the timer is operating, and 0% waveform output or 100% waveform output can be selected at the point of the crest interrupt (INTTBnCC0) and valley interrupt (INTTBnOV).

Remark n = 0, 1, m = 1 to 3

Figure 10-9. 0% PWM Output Waveform (With Dead Time)



<1> 0% output is selected by the valley interrupt (without a match with the 16-bit counter).
The valley interrupt forcibly lowers the timer output. This produces the 0% output.

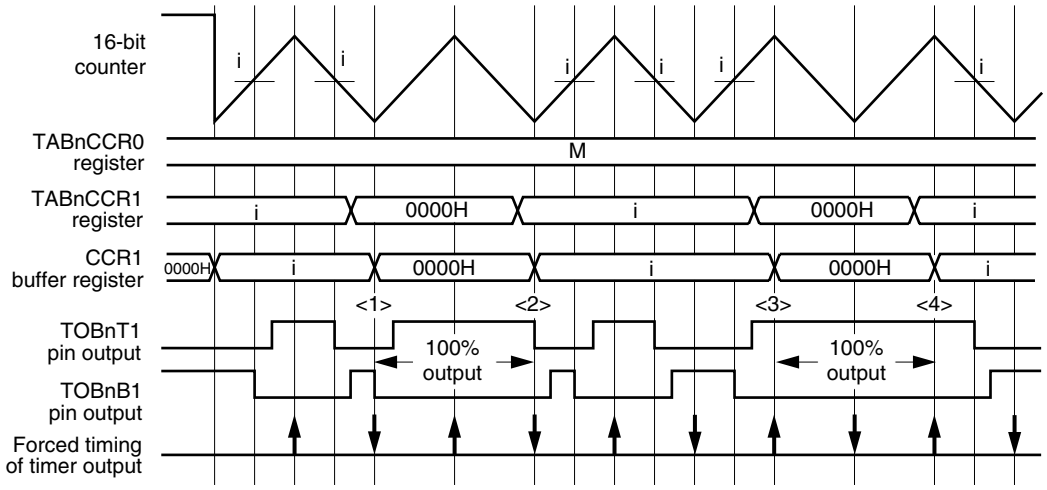
<2> 0% output is canceled by the crest interrupt (without a match with the 16-bit counter).
The crest interrupt forcibly raises the timer output. This cancels the 0% output.

<3> 0% output is selected by the crest interrupt (with a match with the 16-bit counter).
The crest interrupt forcibly raises the timer output, but lowering the timer output takes precedence when the value of the TABnCCRm register matches the value of the 16-bit counter. As a result, the 0% wave is output.

<4> 0% output is canceled by the valley interrupt (without a match with the 16-bit counter).
The valley interrupt forcibly lowers the timer output. This cancels the 0% output.

Remark ↑ means forced raising and ↓ means forced lowering.

Figure 10-10. 100% PWM Output Waveform (With Dead Time)



<1> 100% output is selected by the valley interrupt (with a match with the 16-bit counter).
The valley interrupt forcibly lowers the timer output, but raising the timer output takes precedence when the value of the TABnCCRm register matches the value of the 16-bit counter. As a result, the 100% output is produced.

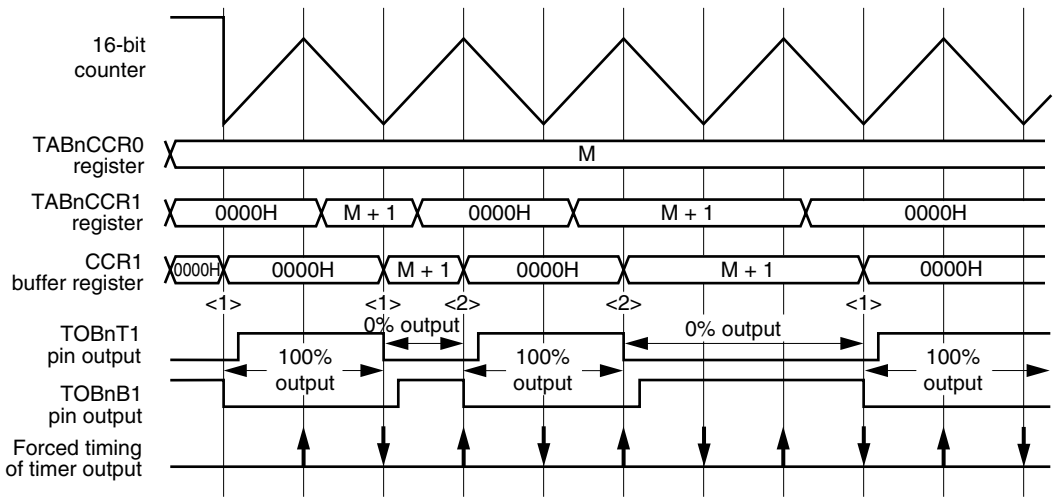
<2> 100% output is canceled by the valley interrupt (without a match with the 16-bit counter).
The valley interrupt forcibly lowers the timer output. This cancels the 100% output.

<3> 100% output is selected by the crest interrupt (without a match with the 16-bit counter).
The crest interrupt forcibly raises the timer output. This produces the 100% output.

<4> 100% output is canceled by the crest interrupt (without a match with the 16-bit counter).
The crest interrupt forcibly raises the timer output. This cancels the 100% output.

Remark ↑ means forced raising and ↓ means forced lowering.

Figure 10-11. PWM Output Waveform from 0% to 100% and from 100% to 0% (With Dead Time)



<1> The valley interrupt selects 100% ←→ 0% or 0% ←→ 100% output.
Output can be selected from 100% ←→ 0% or 0% ←→ 100% immediately after the timer has been started.

<2> The crest interrupt selects 100% ←→ 0% output.
The crest interrupt selects 100% → 0% output by using the timer output forced raising function and by a match between the 16-bit counter value and the TABnCCR0 register value.

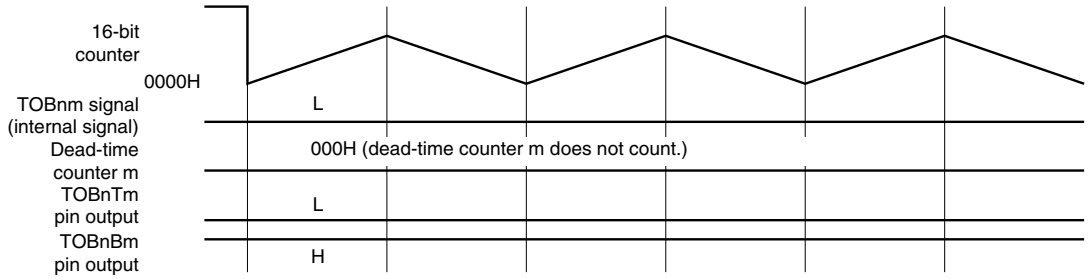
(3) Output wave in vicinity of 0% and 100% output

If an interrupt is generated because the value of the 16-bit counter matches the value of the compare register while dead time is being counted, the dead-time counter is cleared and starts its count operation again.

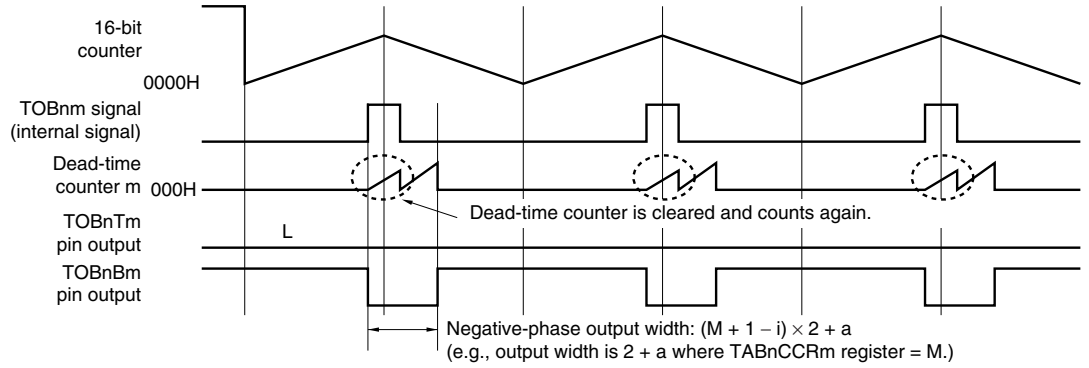
The output waveform of dead-time control in the vicinity of 0% and 100% output is shown below.

Figure 10-12. PWM Output Waveform with Dead Time (2)

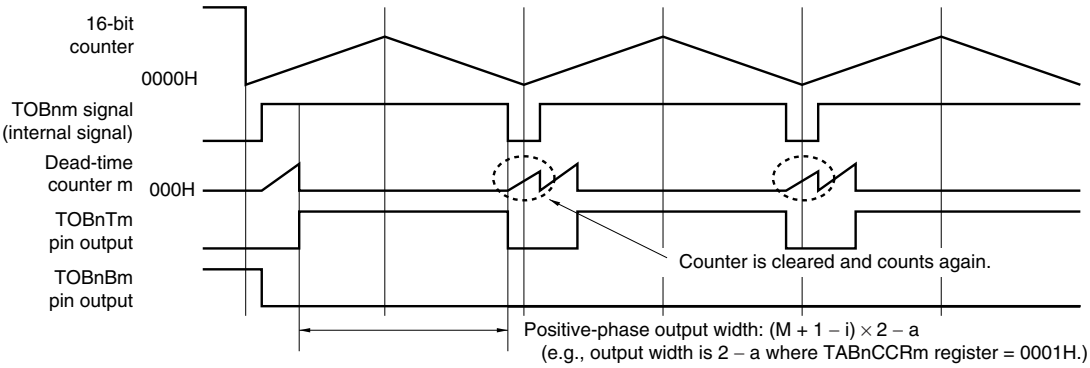
(a) 0% output (TABnCCRm register = M + 1, TABnCCR0 register = M, TABnDTC register = a)



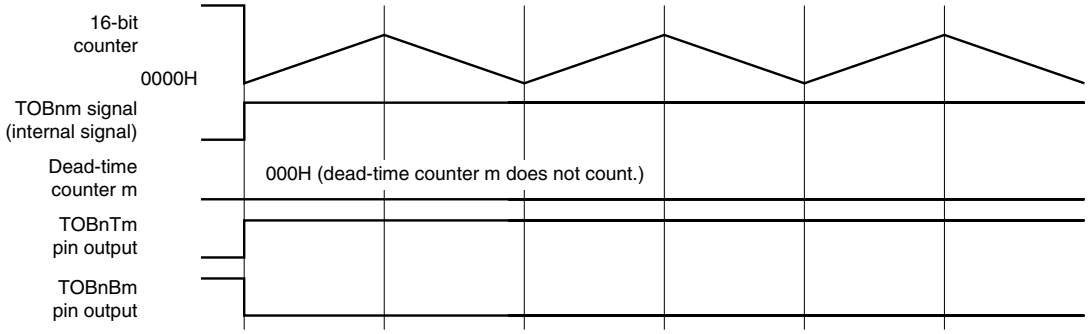
(b) In vicinity of 0% output (TABnCCRm register = i ≥ M + 1 – a/2, TABnCCR0 register = M, TABnDTC register = a)



(c) In vicinity of 100% output (TABnCCRm register = i ≤ a/2, TABnCCR0 register = M, TABnDTC register = a)



(d) 100% output (TABnCCRm register = 0000H, TABnCCR0 register = M, TABnDTC register = a)



Remark n = 0, 1, m = 1 to 3

(4) Automatic dead-time width narrowing function (TABnOPT2.TABnDTM bit = 1)

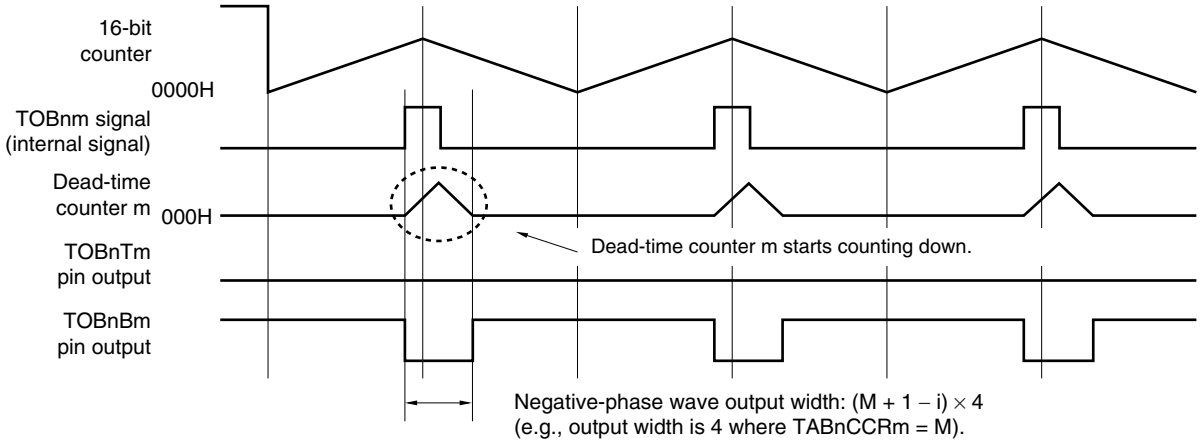
The dead-time width can be automatically narrowed in the vicinity of 0% output or 100% output by setting the TABnOPT2.TABnDTM bit to 1.

By setting the TABnDTM bit to 1, the dead-time counter is not cleared, but starts down counting if the TOBnm (internal signal) output of timer AB changes during dead-time counting.

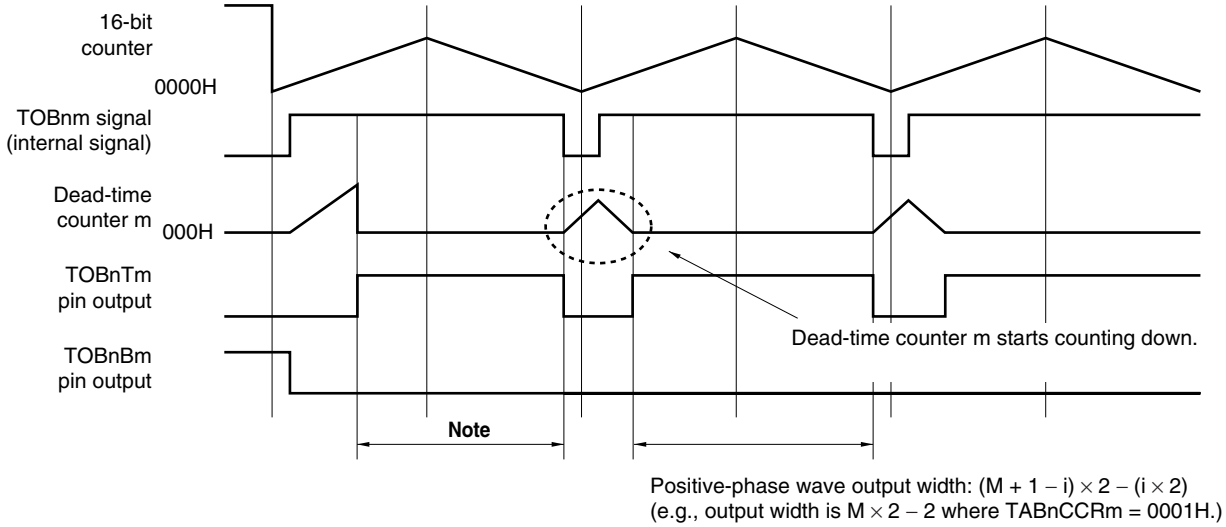
The following timing chart shows the operation of the dead-time counter when the TABnDTM bit is set to 1.

Figure 10-13. Operation of Dead-Time Counter m (1)

(a) In vicinity of 0% output (TABnCCRm register = i ≥ M + 1 – a/2, TABnCCR0 register = M, TABnDTC register = a)



(b) In vicinity of 100% output (TABnCCRm register = i ≤ a/2, TABnCCR0 register = M, TABnDTC register = a)



Note The output width of the first wave differs from that of the second and subsequent waves immediately after the TABnCTL0.TABnCE bit has been set. The first wave is shorter than the second wave because the dead time is fully counted.

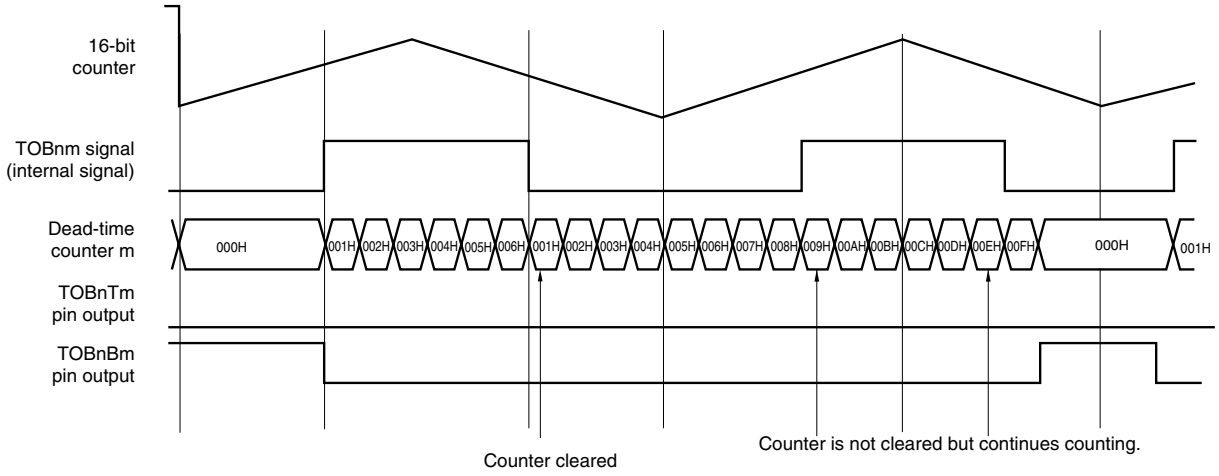
Remark n = 0, 1, m = 1 to 3

(5) Dead-time control in case of incorrect setting

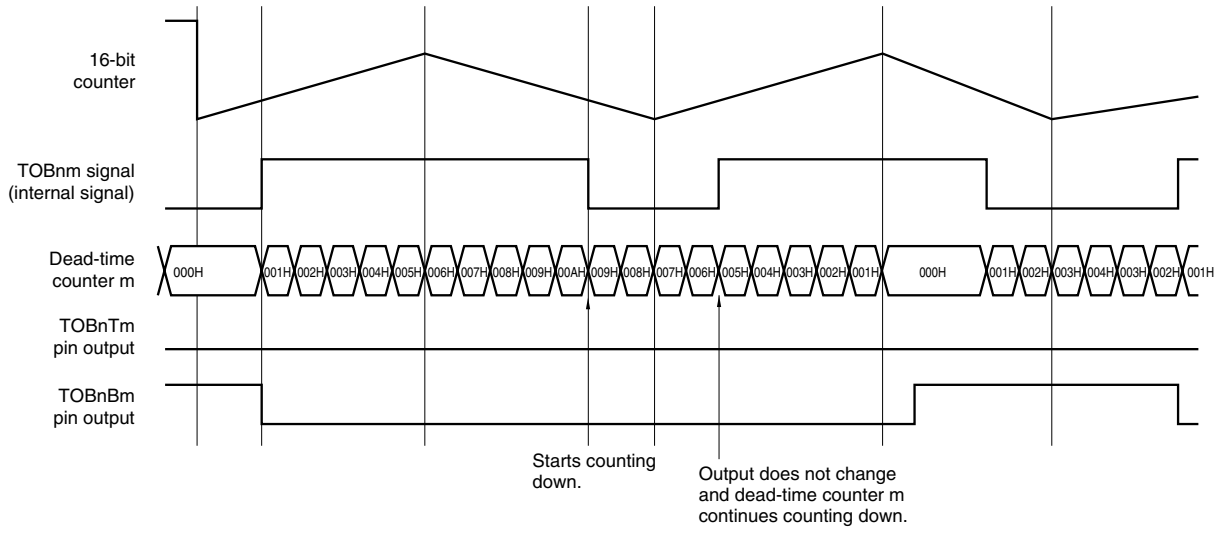
Usually, the TOBnm (internal signal) output of TABn changes only once during dead-time counting, only in the vicinity of 0% and 100% output. This section shows an example where the TABnCCR0 register (carrier cycle) and TABnDTC register (dead-time value) are incorrectly set. If these registers are incorrectly set, the TOBnm (internal signal) output of TABn changes more than once during dead-time counting. The following flowchart shows the 6-phase PWM output waveform in this case.

Figure 10-14. Operation of Dead-Time Counter m (2)

(a) When TABnOPT2.TABnDTM bit = 0, TABnCCR0 register = 0006H, TABnDTC register = 000FH, TABnCCRm register = 0004H



(b) When TABnOPT2.TABnDTM bit = 1, TABnCCR0 register = 0006H, TABnDTC register = 000FH, TABnCCRm register = 0002H



Remark n = 0, 1, m = 1 to 3

10.4.3 Interrupt culling function

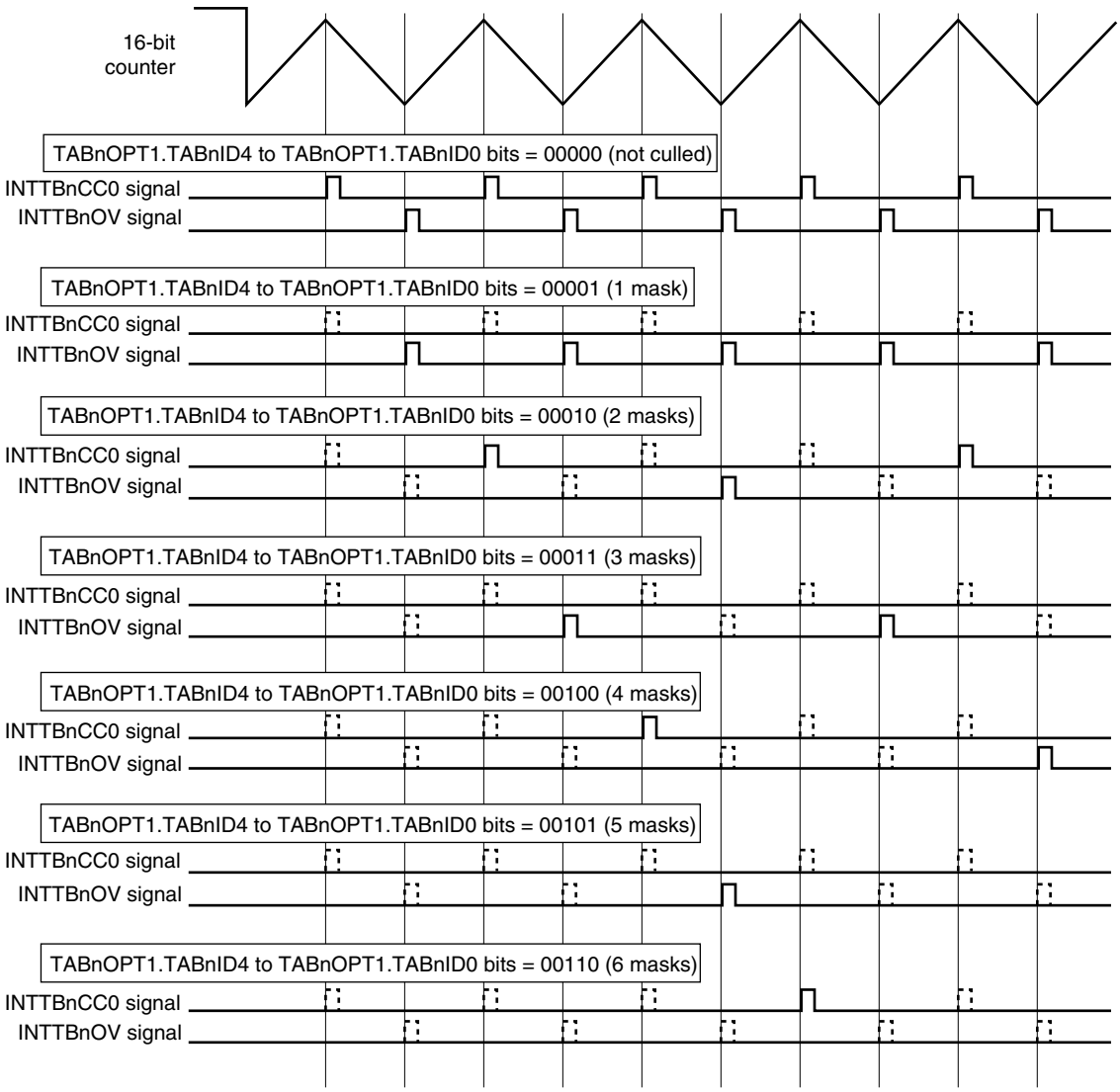
- The interrupts to be culled are INTTBnCC0 (crest interrupt) and INTTBnOV (valley interrupt).
- The TABnOPT1.TABnICE bit is used to enable output of the INTTBnCC0 interrupt and specify the count signal for interrupt culling.
- The TABnOPT1.TABnIOE bit is used to enable output of the INTTBnOV interrupt and specify the count signal for interrupt culling.
- The TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits are used to specify the number of interrupts to be culled, specified for the count signals for interrupt culling.
  The interrupts are masked for the specified number of culling counts and an interrupt occurs at the next interrupt timing.
- The TABnOPT2.TABnRDE bit is used to specify whether transfer is to be culled or not.
  If it is specified that transfer is to be culled, transfer is executed at the same timing as the interrupt output after culling. If it is specified that transfer is not to be culled, transfer is executed at the transfer timing after the TABnCCR1 register has been written.
- The TABnOPT0.TABnCMS bit is used to specify whether the registers with a transfer function are batch rewritten or anytime rewritten.
  The values of the registers are updated in synchronization with transferring when the TABnCMS bit is 0. When the TABnCMS bit is 1, the values of the registers are immediately updated when a new value is written to the registers.
  Transfer is performed from the TABnCCRm register to the CCRm buffer register in synchronization with interrupt culling timing.

Cautions 1. When using the interrupt culling function in the batch rewrite mode (transfer mode), execute the function in the intermittent batch rewrite mode (transfer culling mode).

2. An interrupt is generated at the timing after culling.

(1) Interrupt culling operation

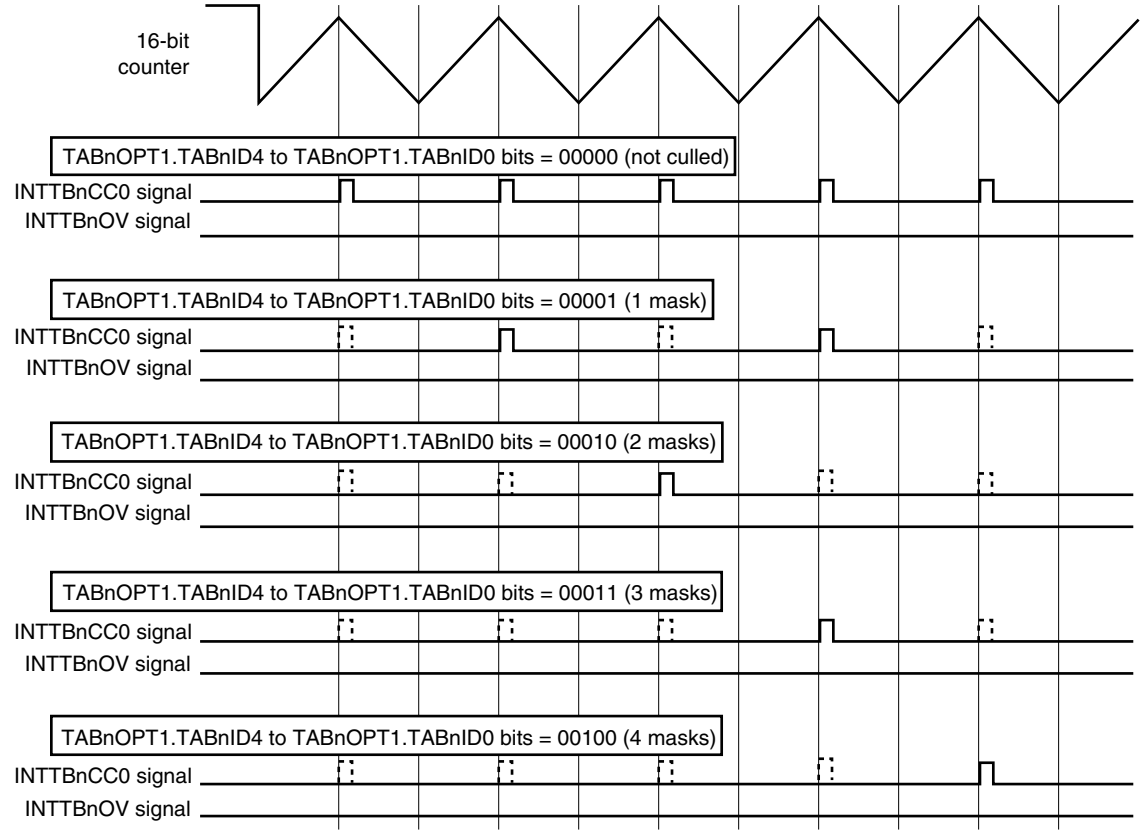
Figure 10-15. Interrupt Culling Operation When TABnOPT1.TABnICE Bit = 1, TABnOPT1.TABnIOE Bit = 1, TABnOPT2.TABnRDE Bit = 1 (Crest/Valley Interrupt Output)



Remarks 1. : Culled interrupt

2. n = 0, 1

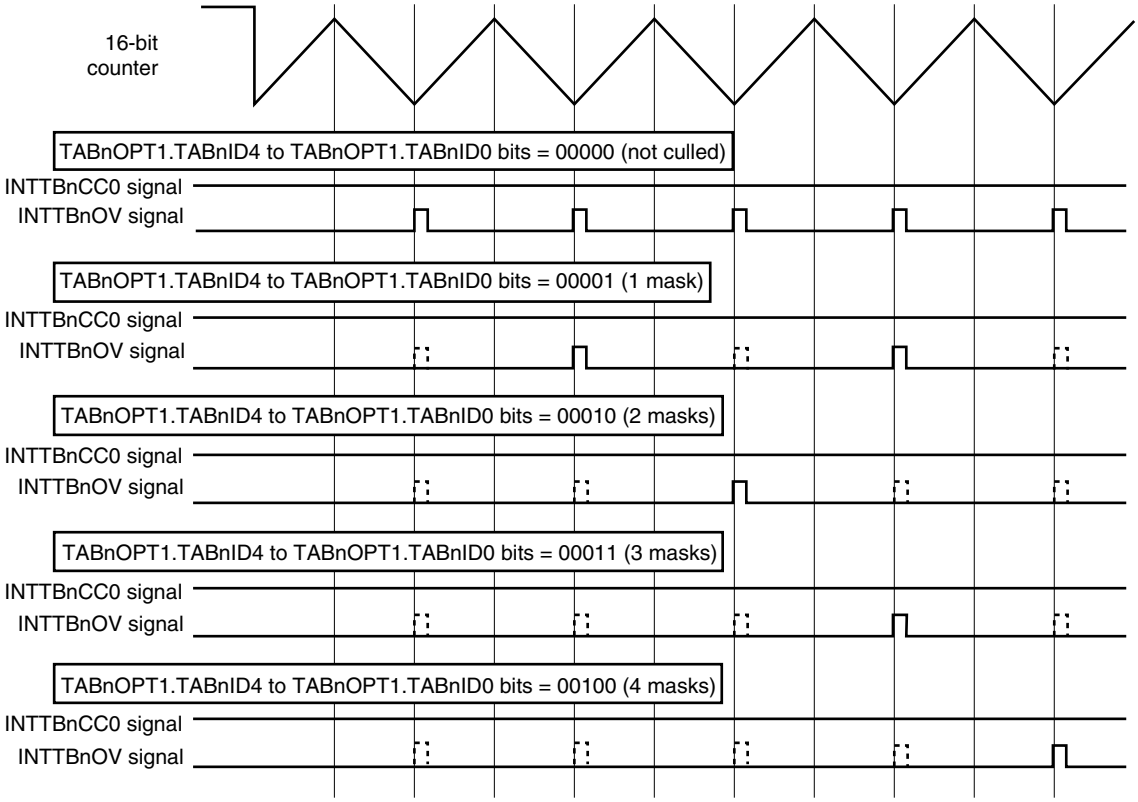
Figure 10-16. Interrupt Culling Operation When TABnOPT1.TABnICE Bit = 1, TABnOPT1.TABnIOE Bit = 0, TABnOPT2.TABnRDE Bit = 1 (Crest Interrupt Output)



Remarks 1. : Culled interrupt

2. n = 0, 1

Figure 10-17. Interrupt Culling Operation When TABnOPT1.TABnICE Bit = 0, TABnOPT1.TABnIOE Bit = 1, TABnOPT2.TABnRDE Bit = 1 (Valley Interrupt Output)



Remarks 1. : Culled interrupt

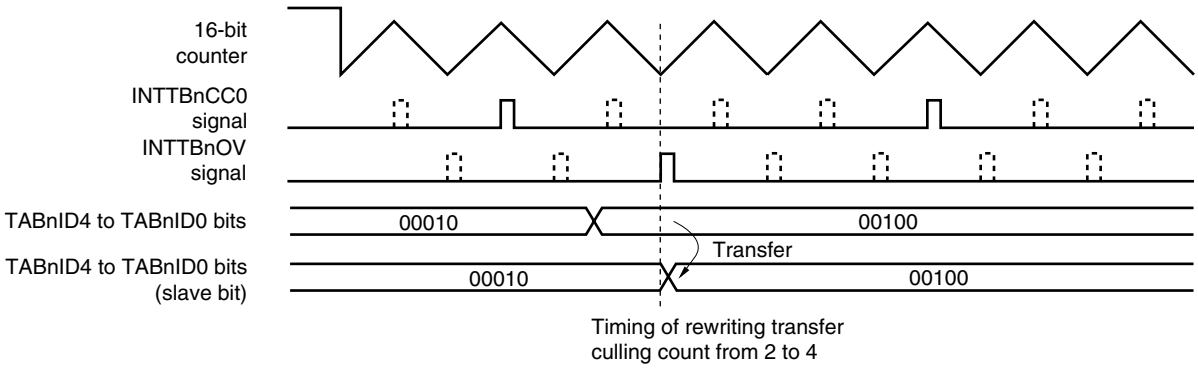
2. n = 0, 1

(2) To alternately output crest interrupt (INTTBnCC0) and valley interrupt (INTTBnOV)

To alternately output the crest and valley interrupts, set both the TABnOPT1.TABnICE and TABnOPT1.TABnIOE bits to 1.

Figure 10-18. Crest/Valley Interrupt Output

(a) TABnOPT0.TABnCMS bit = 0, TABnOPT2.TABnRDE bit = 1 (with transfer culling control)

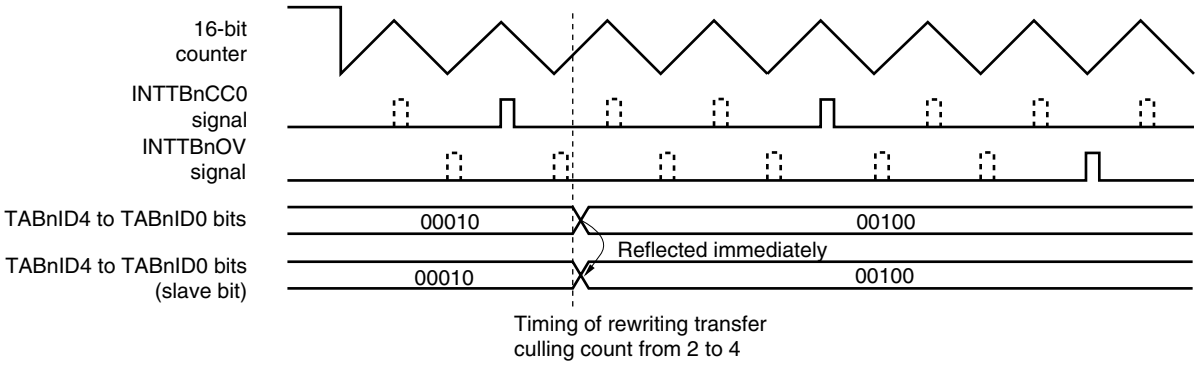


Remarks 1. Transfer is performed when the culled interrupt is output. The other transfer timing is ignored.

2. : Culled interrupt

3. n = 0, 1

(b) TABnCMS bit = 1, TABnRDE bit = 0 or 1 (without transfer control)



Remarks 1. Rewriting is reflected immediately. The transfer timing is ignored.

2. : Culled interrupt

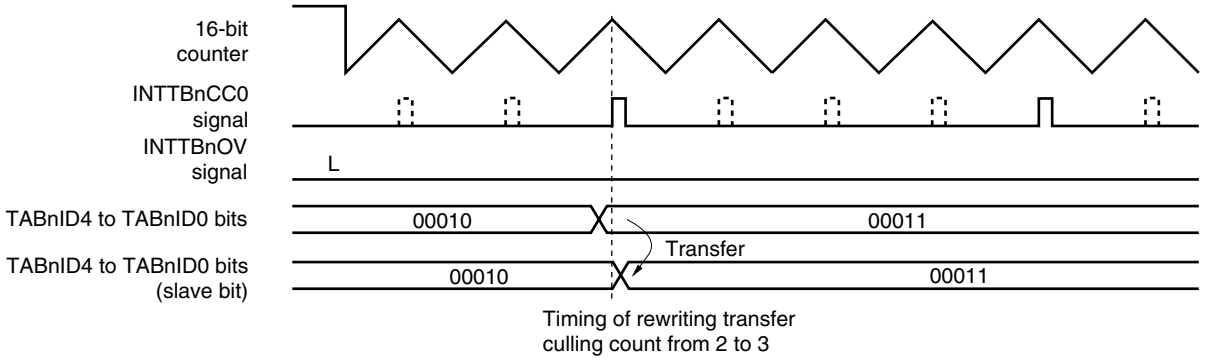
3. n = 0, 1

(3) To output only crest interrupt (INTTBnCC0)

Set the TABnOPT1.TABnICE bit to 1 and set the TABnOPT1.TABnIOE bit to 0.

Figure 10-19. Crest Interrupt Output

(a) TABnOPT0.TABnCMS bit = 0, TABnOPT2.TABnRDE bit = 1 (with transfer culling control)

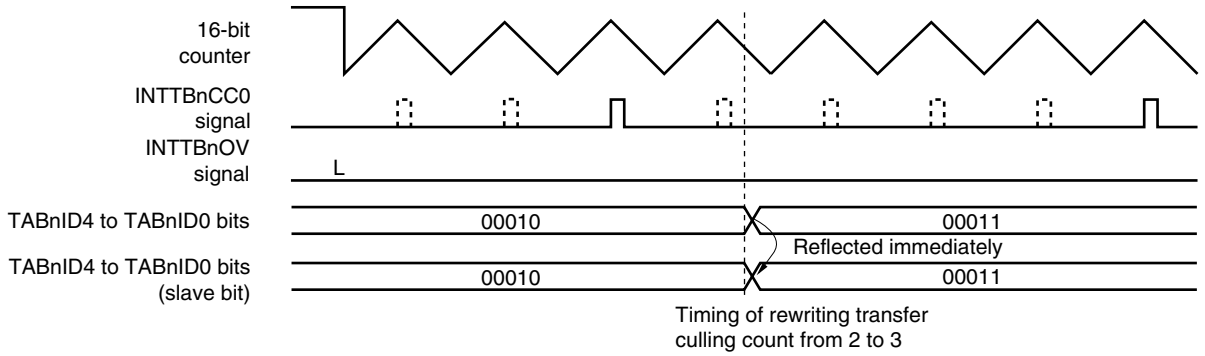


Remarks 1. Transfer is performed when the culled interrupt is output. The other transfer timing is ignored.

2. : Culled interrupt

3. n = 0, 1

(b) TABnOPT0.TABnCMS bit = 1, TABnOPT0.TABnRDE bit = 0 or 1 (without transfer control)



Remarks 1. Rewriting is reflected immediately. The transfer timing is ignored.

2. : Culled interrupt

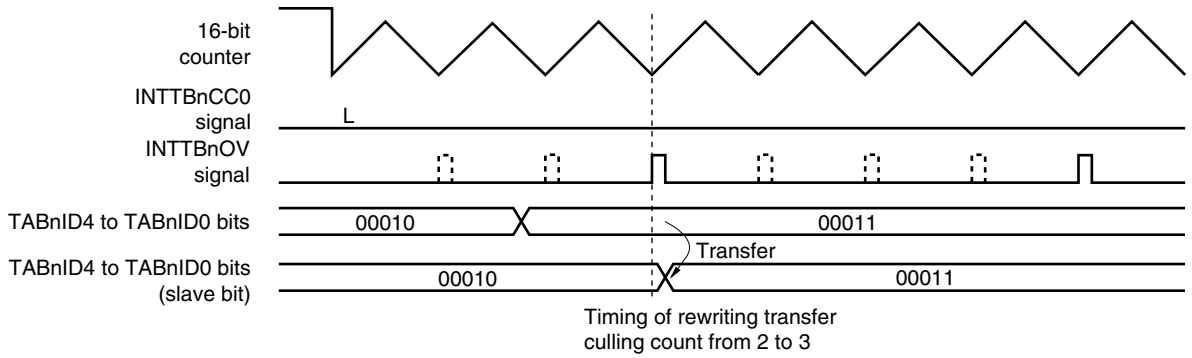
3. n = 0, 1

(4) To output only valley interrupt (INTTBnOV)

Set the TABnOPT1.TABnICE bit to 0 and set the TABnOPT1.TABnIOE bit to 1.

Figure 10-20. Valley Interrupt Output

(a) TABnOPT0.TABnCMS bit = 0, TABnOPT2.TABnRDE bit = 1 (with transfer culling control)

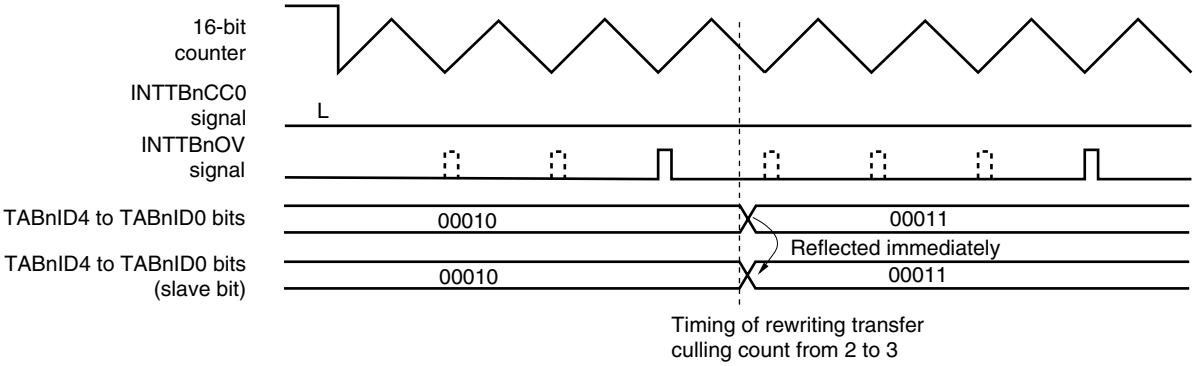


Remarks 1. Transfer is performed when the culled interrupt is output. The other transfer timing is ignored.

2. : Culled interrupt

3. n = 0, 1

(b) TABnOPT0.TABnCMS bit = 1, TABnOPT0.TABnRDE bit = 0 or 1 (without transfer control)



Remarks 1. Rewriting is reflected immediately. The transfer timing is ignored.

2. : Culled interrupt

3. n = 0, 1

10.4.4 Operation to rewrite register with transfer function

The following seven registers are provided with a transfer function and used to control a motor. Each of registers has a buffer register.

- TABnCCR0: Register that specifies the cycle of the 16-bit counter (TAB)
- TABnCCR1: Register that specifies the duty factor of TOBnT1 (U) and TOBnB1 ($\overline{U}$)
- TABnCCR2: Register that specifies the duty factor of TOBnT2 (V) and TOBnB2 ($\overline{V}$)
- TABnCCR3: Register that specifies the duty factor of TOBnT3 (W) and TOBnB3 ($\overline{W}$)
- TABnOPT1: Register that specifies the culling of interrupts
- TAAnCCR0: Register that specifies the A/D conversion start trigger generation timing (TAAn during tuning operation)
- TAAnCCR1: Register that specifies the A/D conversion start trigger generation timing (TAAn during tuning operation)

The following three rewrite modes are provided in the registers with a transfer function.

- Anytime rewriting mode

  This mode is specified by setting the TABnOPT0.TABnCMS bit to 1. The setting of the TABnOPT2.TABnRDE bit is ignored.

  In this mode, each compare register is updated independently, and the value of the compare register is updated as soon as a new value is written to it.

- Batch rewrite mode (transfer mode)

  This mode is specified by setting the TABnOPT0.TABnCMS bit to 0, the TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits to 00000, and the TABnOPT2.TABnRDE bit to 0.

  When data is written to the TABnCCR1 register, the seven registers are transferred to the buffer register all at once at the next transfer timing. Unless the TABnCCR1 register is rewritten, the transfer operation is not performed even if the other six registers are rewritten.

  The transfer timing is the timing of each crest (match between the 16-bit counter value and TABnCCR0 register value) and valley (match between the 16-bit counter value and 0001H) regardless of the interrupt.

- Intermittent batch rewrite mode (transfer culling mode)

  This mode is specified by setting the TABnOPT0.TABnCMS bit to 0 and the TABnOPT2.TABnRDE bit to 1.

  When data is written to the TABnCCR1 register, the seven registers are transferred to the buffer register all at once at the next transfer timing. Unless the TABnCCR1 register is rewritten, the transfer operation is not performed even if the other six registers are rewritten.

  If interrupt culling is specified by the TABnOPT1 register, the transfer timing is also culled as the interrupts are culled, and the seven registers are transferred all at once at the culled timing of crest interrupt (match between the 16-bit counter value and TABnCCR0 register value) or valley interrupt (match between the 16-bit counter value and 0001H).

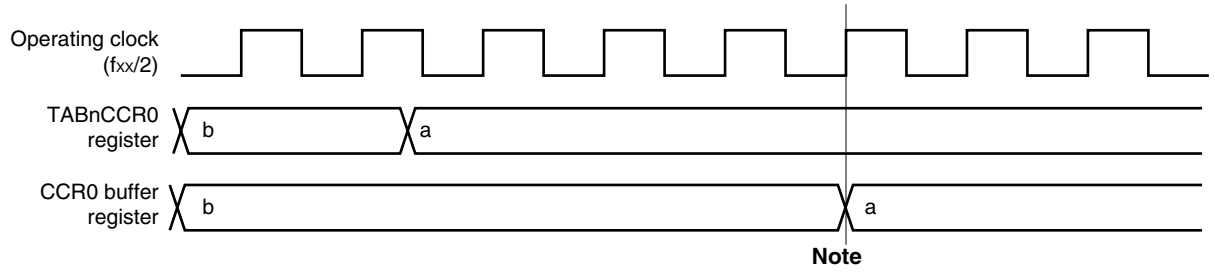
  For details of the interrupt culling function, see **10.4.3 Interrupt culling function**.

(1) Anytime rewriting mode

This mode is specified by setting the TABnOPT0.TABnCMS bit is 1. The setting of the TABnOPT2.TABnRDE bit is ignored.

In this mode, the value written to each register with a transfer function is immediately transferred to an internal buffer register and compared with the value of the counter. If a register with transfer function is rewritten in this mode after the count value of the 16-bit counter matches the value of the TABnCCRm register, the rewritten value is not reflected because the next match is ignored after the first match has occurred. If the register is rewritten during up counting, the new register value becomes valid after the counter has started counting down.

Figure 10-21. Timing of Reflecting Rewritten Value

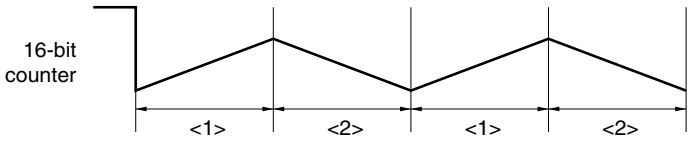


Note After the register (TABnCCR0, TABnCCR2, TABnCCR3, TABnOPT1, TAAnCCR0, or TAAnCCR1) has been written, the written value is transferred to an internal buffer register after four clocks of the operating clock. However, the value of only the TABnCCR1 register is transferred after 5 more clocks.

(a) Rewriting TABnCCR0 register

Even if the TABnCCR0 register is rewritten in the anytime rewriting mode, the new value may not be reflected in some cases.

Figure 10-22. Example of Rewriting TABnCCR0 Register



Rewriting during period <1> (rewriting during up counting)

If the newly rewritten value is greater than the value of the 16-bit counter, there is no problem because it will match the value of the 16-bit counter. If the new value is less than the value of the 16-bit counter, it will not match the value of the counter. As a result, the 16-bit counter overflows and continues counting up from 0000H until it matches the register value again, and the correct PWM waveform is not output.

Rewriting during period <2> (rewriting during down counting)

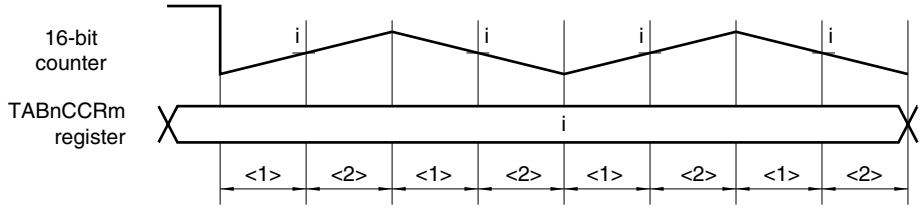
A match with the value of the 16-bit counter is ignored during counting down. Therefore, the rewritten period value is reflected starting from counting up in the next cycle as a match point.

(b) Rewriting TABnCCRm register

Figure 10-24 shows the timing of rewriting before the value of the 16-bit counter matches the value of the TABnCCRm register (<1> in Figure 10-23), and Figure 10-25 shows the timing of rewriting after the value of the 16-bit counter matches the value of the TABnCCRm register (<2> in Figure 10-23).

Figure 10-23. Basic Operation of 16-Bit Counter and TABnCCRm Register

(a) Basic figure



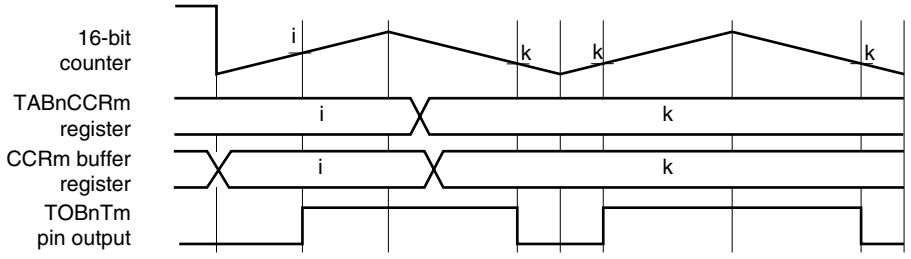
Remarks 1. i = Set value of TABnCCRm register

2. n = 0, 1, m = 1 to 3

Figure 10-24. Example of Rewriting TABnCCR1 to TABnCCR3 Registers (Rewriting Before Match Occurs)

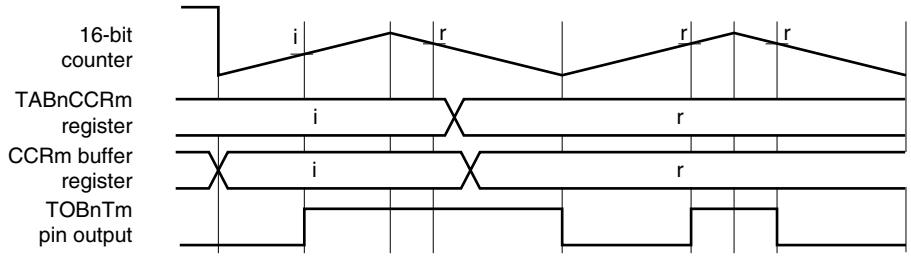
(a)

If the TABnCCRm register is rewritten before its value matches the value of the 16-bit counter, the register value will match the value of the 16-bit counter after the register has been rewritten. Consequently, the new register value is immediately reflected.



(b)

If a value less than the value of the 16-bit counter (greater if the counter is counting down) is written to the TABnCCRm register, the output waveform is as follows because the register value does not match the counter value.

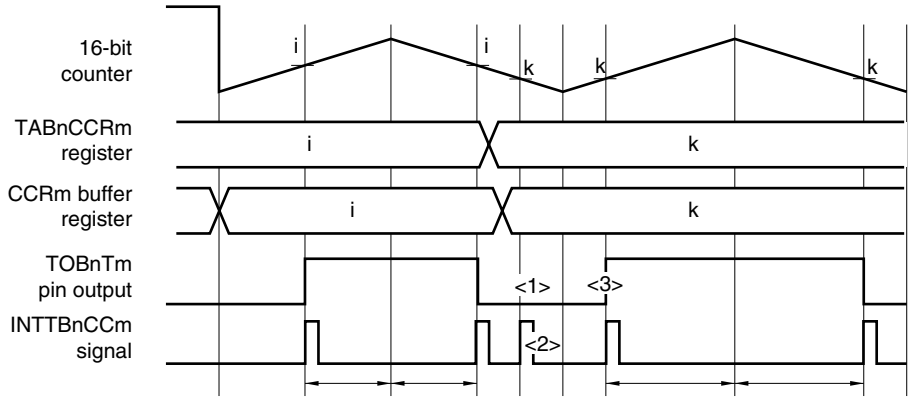


If the register value does not match the counter value, the TOBnTm pin output does not change. Even if the value of the 16-bit counter does not match the value of the TABnCCRm register, the TOBnTm pin output always changes to the high level if the crest interrupt occurs and to the low level if the valley interrupt occurs.
This is a function provided for 0% output and 100% output.
For details, see **10.4.2 (2) PWM output of 0%/100%**.

Remarks 1. i, r, k = Set values of TABnCCRm register

2. n = 0, 1, m = 1 to 3

Figure 10-25. Example of Rewriting TABnCCR1 to TABnCCR3 Registers (Rewriting After Match Occurs)



<1> Matching of the count value of the 16-bit counter and the value of the TABnCCRm register as a result of rewriting the register is ignored after a match signal has been generated, and the PWM output does not change.

<2> Even if the PWM output does not change, the interrupt generated upon a match between the 16-bit counter value and the TABnCCRm register value (INTTBnCCm) is output.

<3> The next match between the 16-bit counter and TABnCCRm register is valid after the counter has changed its counting direction to up or down, and the PWM output changes.

If the TABnCCRm register is rewritten after its value matches the value of the 16-bit counter, the next match is ignored after the first match occurs and the rewritten value is not reflected to the TOBnTm pin output. If the register is rewritten while the counter is counting down, the match that occurs after the counter starts counting down is valid (the match that occurs after the counter has started counting up is valid if the register is rewritten while the counter is counting up).

Remarks 1. i, r, k = Set value of TABnCCRm register

2. n = 0, 1, m = 1 to 3

(c) Rewriting TABnOPT1 register

The interrupt culling counter is cleared when the TABnOPT1 register is written. When the interrupt culling counter has been cleared, the measured number of times the interrupt has occurred is discarded. Consequently, the interrupt generation interval is temporarily extended.

To avoid this operation, rewrite the TABnOPT1 register in the intermittent batch rewriting mode (transfer culling mode).

For details of rewriting the TABnOPT1 register, see **10.4.3 Interrupt culling function**.

(2) Batch rewrite mode (transfer mode)

This mode is specified by setting the TABnOPT0.TABnCMS bit to 0, the TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits to 00000, and the TABnOPT2.TABnRDE bit to 0.

In this mode, the values written to each compare register are transferred to the internal buffer register all at once at the transfer timing and compared with the counter value.

(a) Rewriting procedure

If data is written to the TABnCCR1 register, the values set to the TABnCCR0 to TABnCCR3, TABnOPT1, TAAnCCR0, and TAAnCCR1 registers are transferred all at once to the internal buffer register at the next transfer timing. Therefore, write to the TABnCCR1 register last. Writing to the register is prohibited after the TABnCCR1 register has been written and before the transfer timing is generated (until the crest (match between the 16-bit counter value and TABnCCR0 register value) or the valley (match between the 16-bit counter value and 0001H)). The operation procedure is as follows.

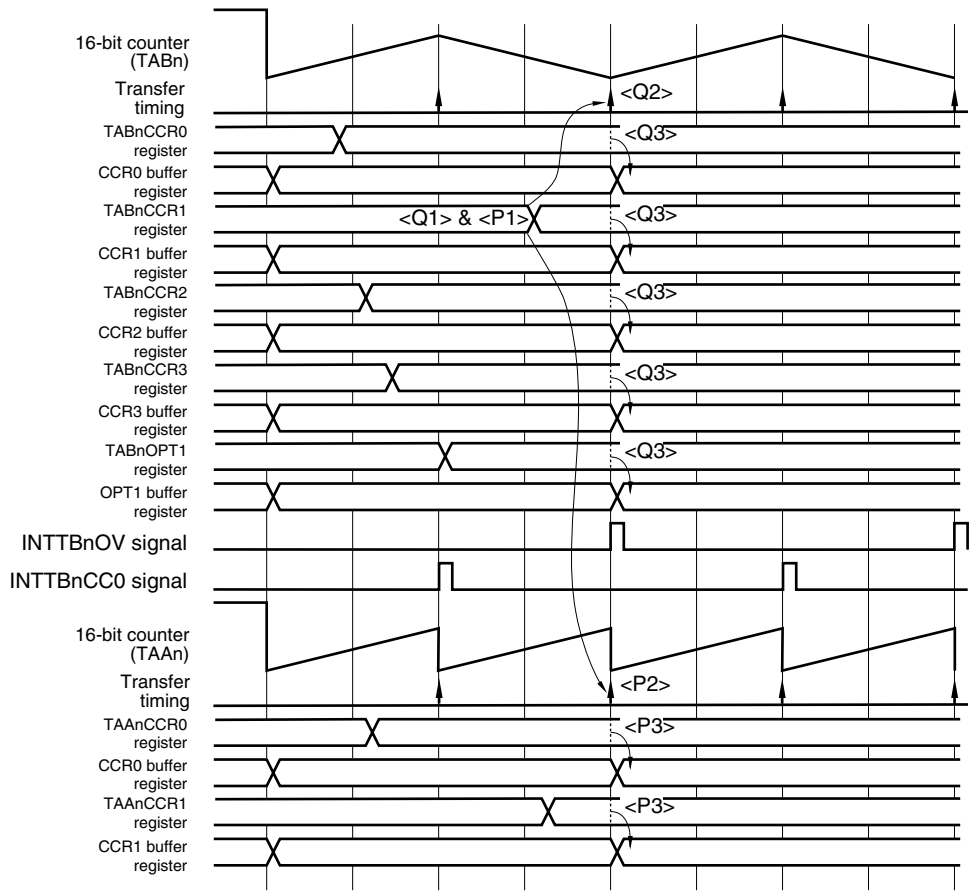
<1> Rewriting the TABnCCR0, TABnCCR2, TABnCCR3, TABnOPT1, TAAnCCR0, and TAAnCCR1 registers.
Do not rewrite registers that do not have to be rewritten.

<2> Rewriting the TABnCCR1 register.
Rewrite the same value to the register even when it is not necessary to rewrite the TABnCCR1 register.

<3> Holding the next rewriting pending until the transfer timing is generated.
Rewrite the register next time after the INTTBnOV or INTTBnCC0 interrupt has occurred.

<4> Return to <1>.

Figure 10-26. Basic Operation in Batch Mode



[Operation of TABn]

<Q1> Write the TABnCCR1 register

<Q2> The target timing is the first transfer timing after a write to the TABnCCR1 register.

<Q3> The values are transferred all at once at the transfer timing.

[Operation of TAAn]

<P1> Write the TABnCCR1 register

<P2> The target timing is the first transfer timing after a write to the TABnCCR1 register.

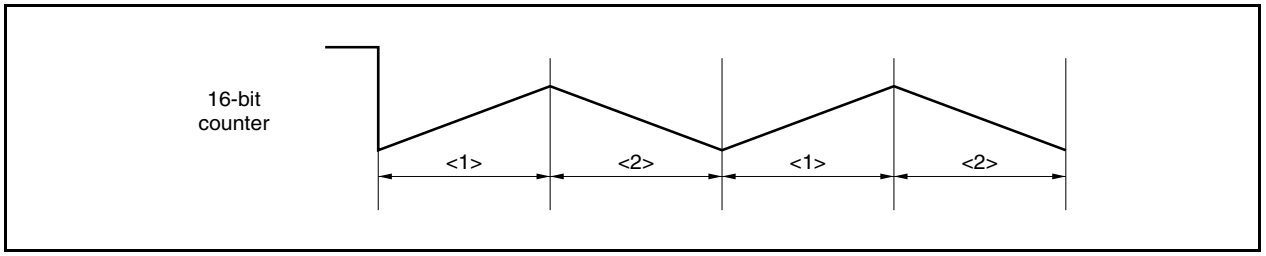
<P3> The values are transferred all at once at the transfer timing.

(b) Rewriting TABnCCR0 register

When rewriting the TABnCCR0 register in the batch rewrite mode, the output waveform differs depending on whether transfer occurs at the crest (match between the 16-bit counter value and TABnCCR0 register value) or at the valley (match between the 16-bit counter value and 0001H). Usually, it is recommended to rewrite the TABnCCR0 register while the 16-bit counter is counting down, and transfer the register value at the transfer timing of the crest timing.

Figure 10-28 shows an example of rewriting the TABnCCR0 register while the 16-bit counter is counting up (during period <1> in Figure 10-27). Figure 10-29 shows an example of rewriting the TABnCCR0 register while the counter is counting down (during period <2> in Figure 10-27).

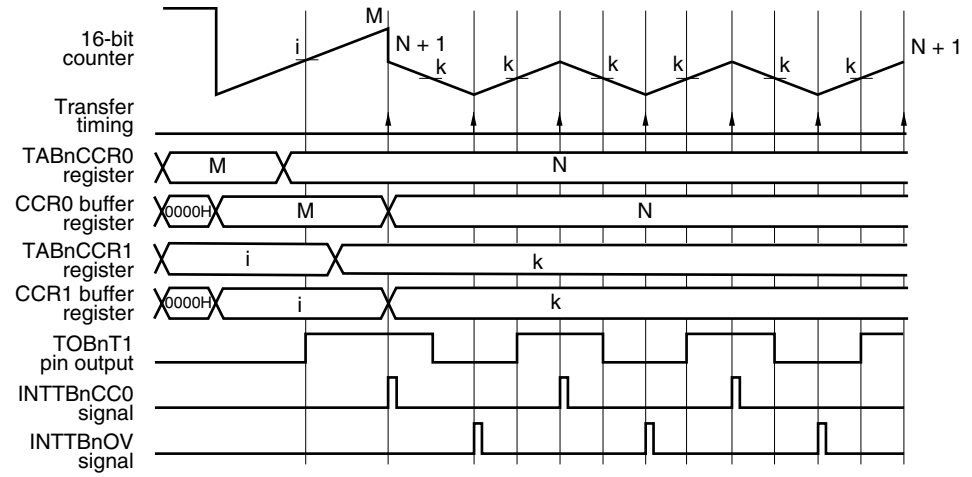
Figure 10-27. Basic Operation of 16-Bit Counter



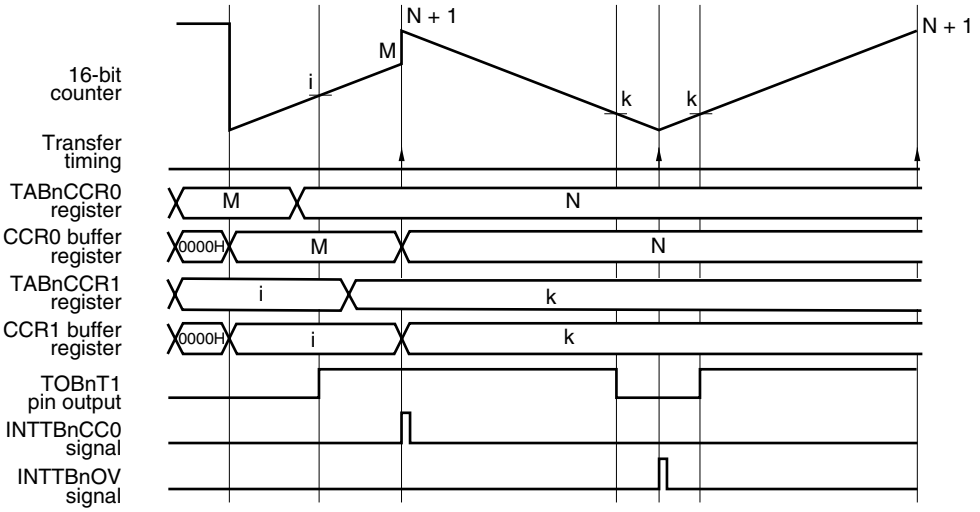
The transfer timing in Figure 10-28 is at the point where the crest timing occurs. While the 16-bit counter is counting down, the cycle changes and an asymmetrical triangular wave is output. Because the cycle changes, rewrite the duty factor (voltage data value).

Figure 10-28. Example of Rewriting TABnCCR0 Register (During Up Counting)

(a) M > N



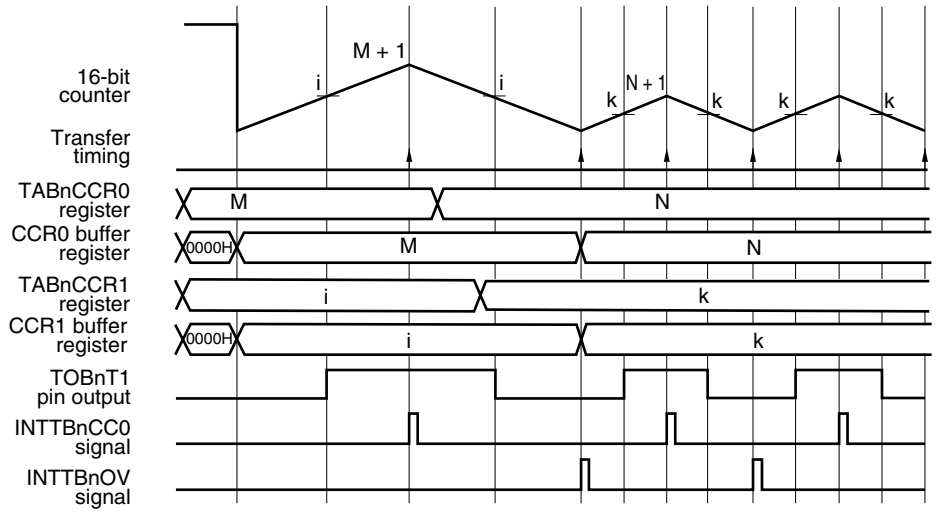
(b) M < N



Remarks 1. If transfer (match between the value of the 16-bit counter and the value of the CCR0 buffer register) occurs in the 6-phase PWM output mode, the value of the TABnCCR0 register plus 1 is loaded to the 16-bit counter. In this way, the expected wave can be output even if the cycle value is changed at the transfer timing of the crest (match between the 16-bit counter value and the TABnCCR0 register value) timing.

2. M: Value of CCR0 buffer register before rewriting
N: Value of CCR0 buffer register after rewriting

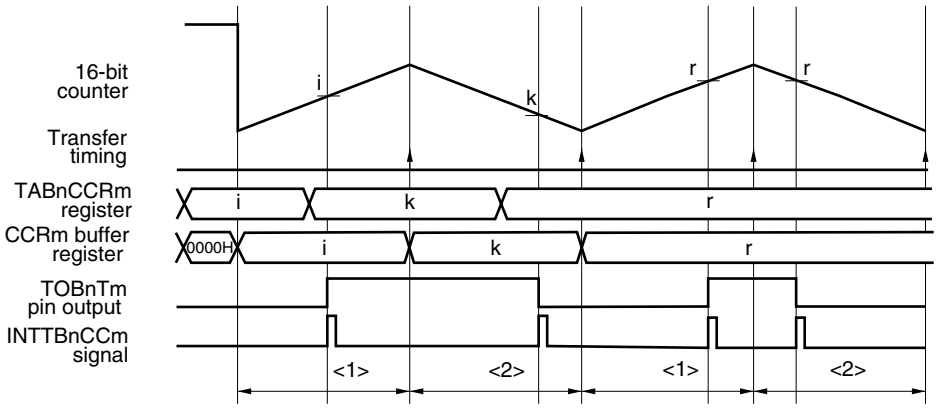
Figure 10-29. Example of Rewriting TABnCCR0 Register (During Down Counting)



Because the next transfer timing is at the point of the valley (match between the 16-bit counter value and 0001H), the cycle value changes from the next cycle and output of a symmetrical triangular wave is maintained. Because the cycle changes, rewrite the duty value (voltage data value) as required.

(c) Rewriting TABnCCRm register

Figure 10-30. Example of Rewriting TABnCCRm Register



Rewriting during period <1> (rewriting during counting up)

Because the TABnCCRm register value is transferred at the transfer timing of the crest (match between the 16-bit counter value and TABnCCR0 register value), an asymmetrical triangular wave is output.

Rewriting during period <2> (rewriting during counting down)

Because the TABnCCRm register value is transferred at the transfer timing of the valley (match between the 16-bit counter value and 0001H), a symmetrical triangular wave is output.

Remark m = 1 to 3

(d) Transferring TABnOPT1 register value

Do not set the TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits to other than 00000. When using the interrupt culling function, rewrite the TABnOPT1 register in the intermittent batch rewrite mode (transfer culling mode).

For details of rewriting the TABnOPT1 register, see **10.4.3 Interrupt culling function**.

(3) Intermittent batch rewriting mode (transfer culling mode)

This mode is specified by setting the TABnOPT0.TABnCMS bit is 0 and the TABnOPT2.TABnRDE bit is 1.

In this mode, the values written to each compare register are transferred to the internal buffer register all at once at the culled transfer timing and compared with the counter value.

The transfer timing is the timing at which an interrupt is generated (INTTBnCC0, INTTBnOV) by interrupt culling.

For details of the interrupt culling function, see **10.4.3 Interrupt culling function**.

(a) Rewriting procedure

If data is written to the TABnCCR1 register, the TABnCCR0 to TABnCCR3, TABnOPT1, TAAnCCR0, and TAAnCCR1 registers are transferred all at once to the internal buffer register at the next transfer timing. Therefore, write to the TABnCCR1 register last. Writing to the register is prohibited after the TABnCCR1 register has been written until the transfer timing is generated (until the INTTBnOV or INTTBnCC0 interrupt occurs). The operation procedure is as follows.

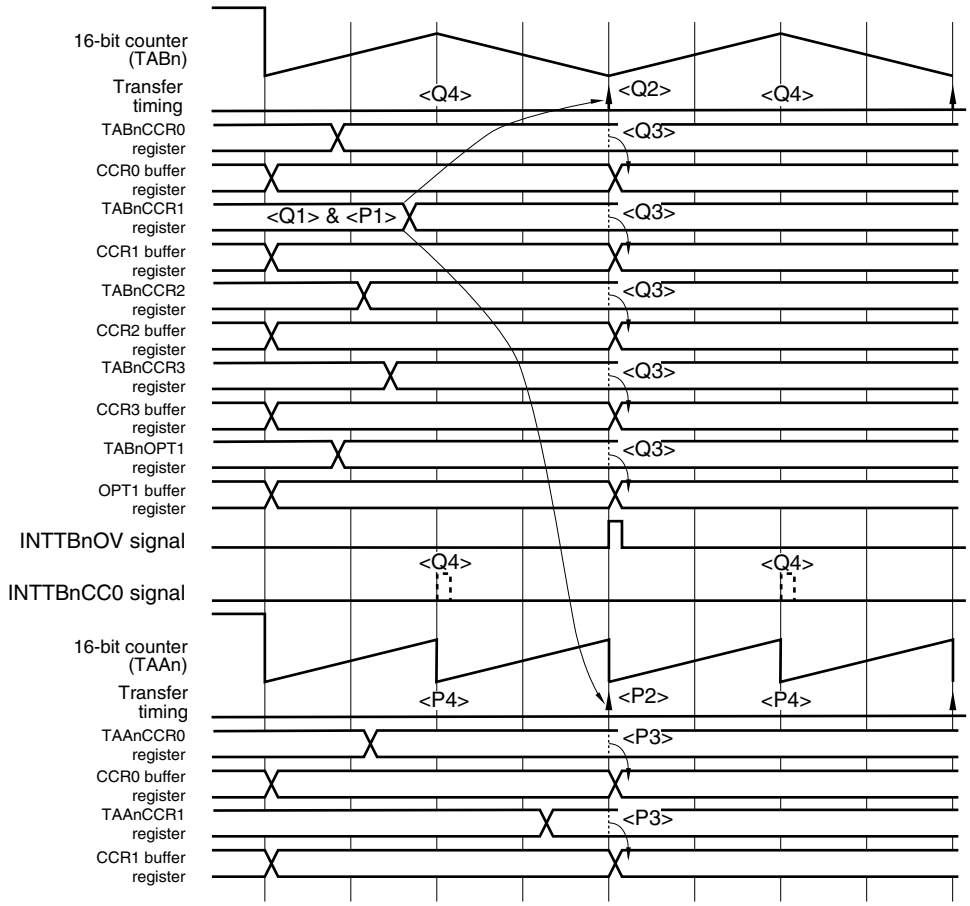
<1> Rewrite the TABnCCR0, TABnCCR2, TABnCCR3, TABnOPT1, TAAnCCR0, and TAAnCCR1 registers. Do not rewrite registers that do not have to be rewritten.

<2> Rewrite the TABnCCR1 register.
Rewrite the same value to the register even when it is not necessary to rewrite the TABnCCR1 register.

<3> Hold the next rewriting pending until the transfer timing is generated.
Perform the next rewrite after the INTTBnOV or INTTBnCC0 interrupt has occurred.

<4> Return to <1>.

Figure 10-31. Basic Operation in Intermittent Batch Rewriting Mode



[TABn operation]

<Q1> Write the TABnCCR1 register.

<Q2> Rewrite the register at the transfer timing that is generated after the TABnCCR1 register has been rewritten.

<Q3> The registers are transferred all at once at the transfer timing.

<Q4> The transfer timing is also culled as the interrupts are culled.

[TAAn operation]

<P1> Write the TABnCCR1 register.

<P2> Rewrite the register at the transfer timing that is generated after the TABnCCR1 register has been rewritten.

<P3> The registers are transferred all at once at the transfer timing.

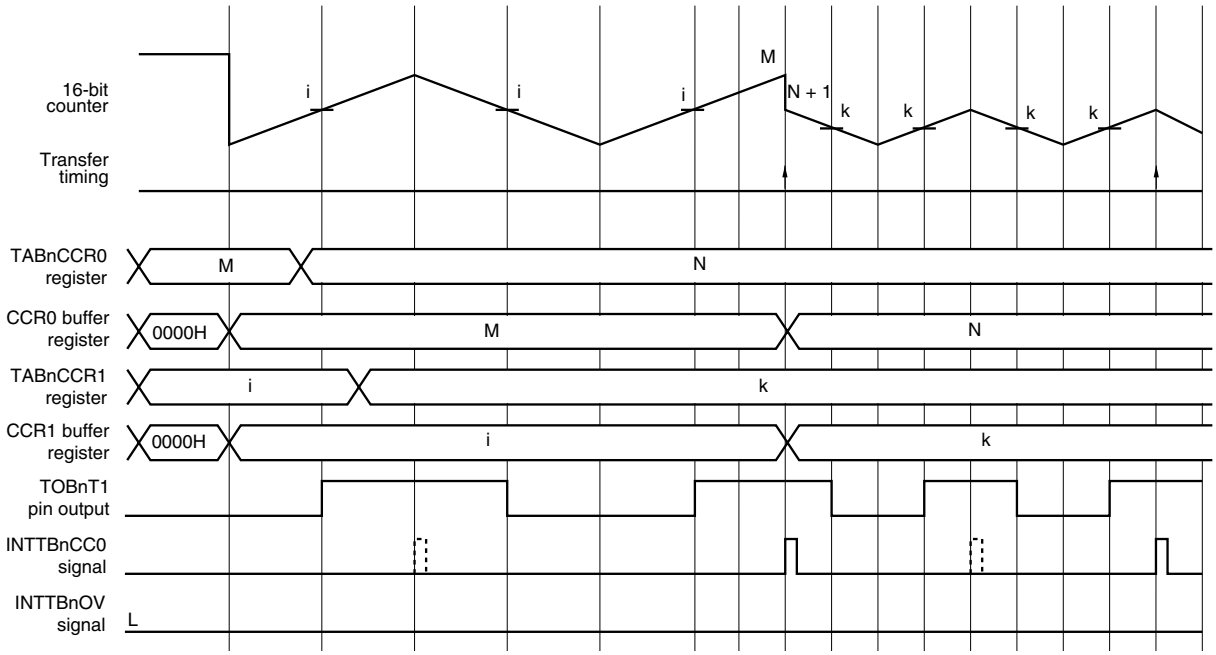
<P4> The transfer timing is also culled as the interrupts are culled.

Remark This is an example of the operation when the TABnOPT1.TABnICE bit = 1, TABnOPT1.TABnIOE bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00001.

(b) Rewriting TABnCCR0 register

When rewriting the TABnCCR0 register in the intermittent batch mode, the output waveform differs depending on where the occurrence of the crest or valley interrupt is specified by the interrupt culling setting. The following figure illustrates the change of the output waveform when interrupts are culled.

Figure 10-32. Rewriting TABnCCR0 Register (When Crest Interrupt Is Set)



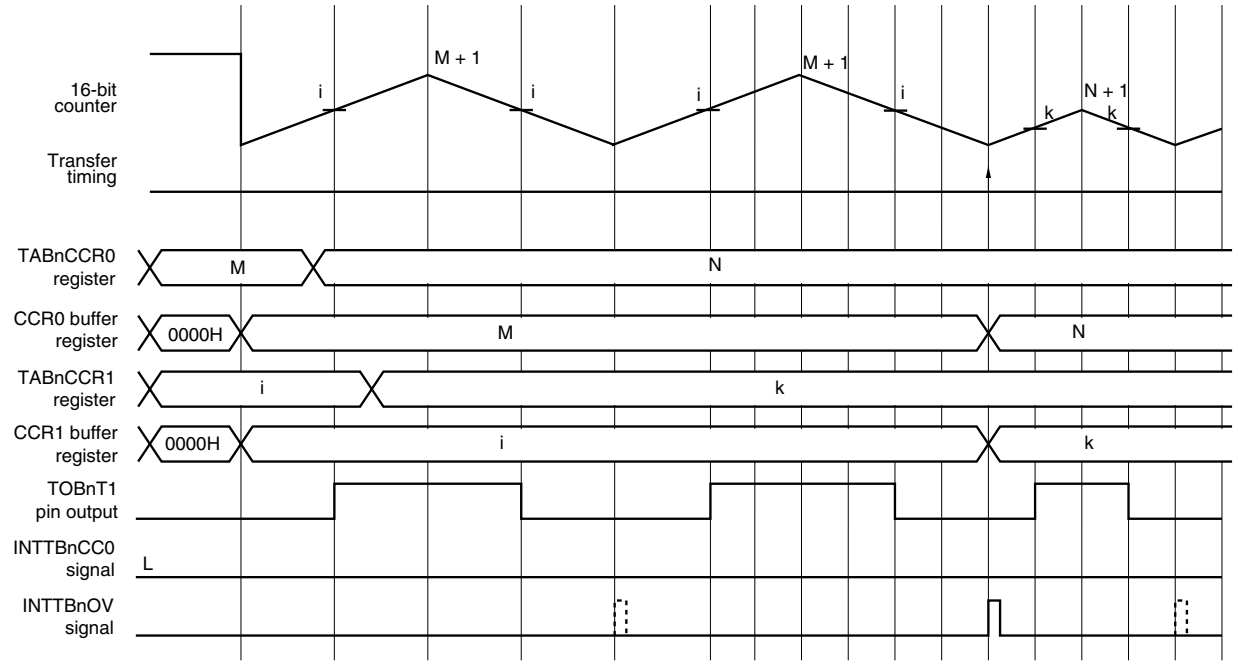
The transfer timing is generated when the crest interrupt occurs, the period of up counting and down counting changes, and an asymmetrical triangular wave is output.

Remarks 1. This is an example of the operation when the TABnOPT1.TABnICE bit = 1, TABnOPT1.TABnIOE bit = 0, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00001.

2. : Culled interrupt

3. n = 0, 1

Figure 10-33. Rewriting TABnCCR0 Register (When Valley Interrupt Is Set)



The transfer timing is generated when the valley interrupt occurs, the cycle of up counting and down counting becomes identical, and a symmetrical triangular wave is output.

Remarks 1. This is an example of the operation when the TABnOPT1.TABnICE bit = 0, TABnOPT1.TABnIOE bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 bits = 00001.

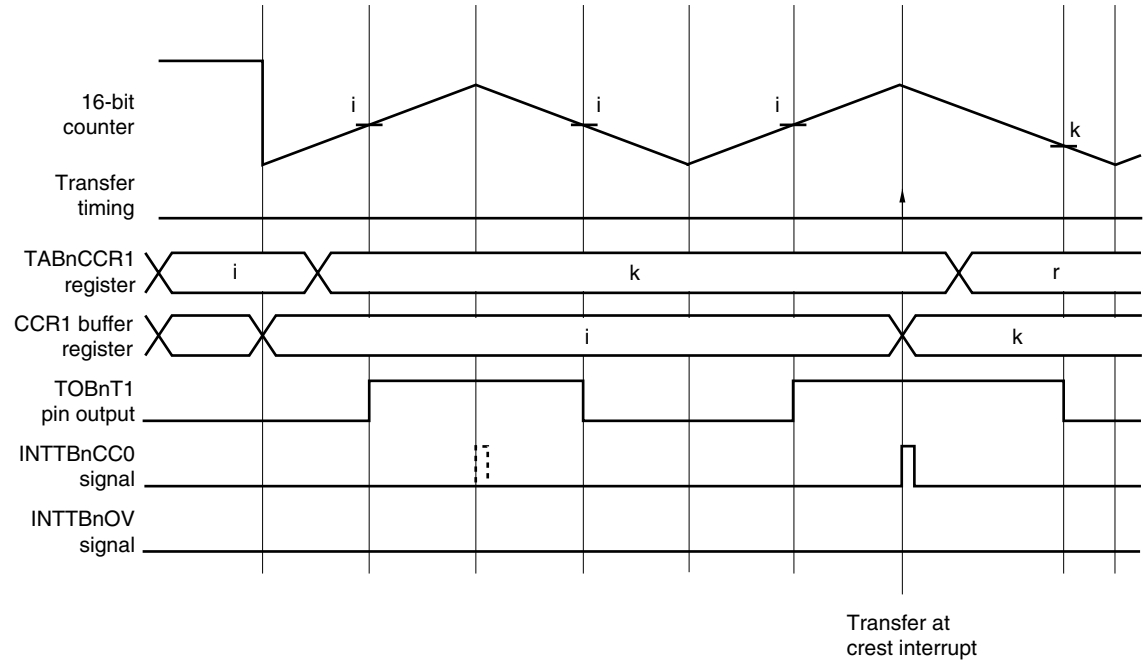
2. : Culled interrupt

3. n = 0, 1

(c) Rewriting TABnCCR1 to TABnCCR3 registers

- Transfer at crest when crest interrupt is set
  Because the register is transferred at the transfer timing of the crest interrupt, an asymmetrical triangular wave is output.

Figure 10-34. Rewriting TABnCCR1 Register
(TABnOPT1.TABnICE Bit = 1, TABnOPT1.TABnIOE Bit = 0, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 = 00001)

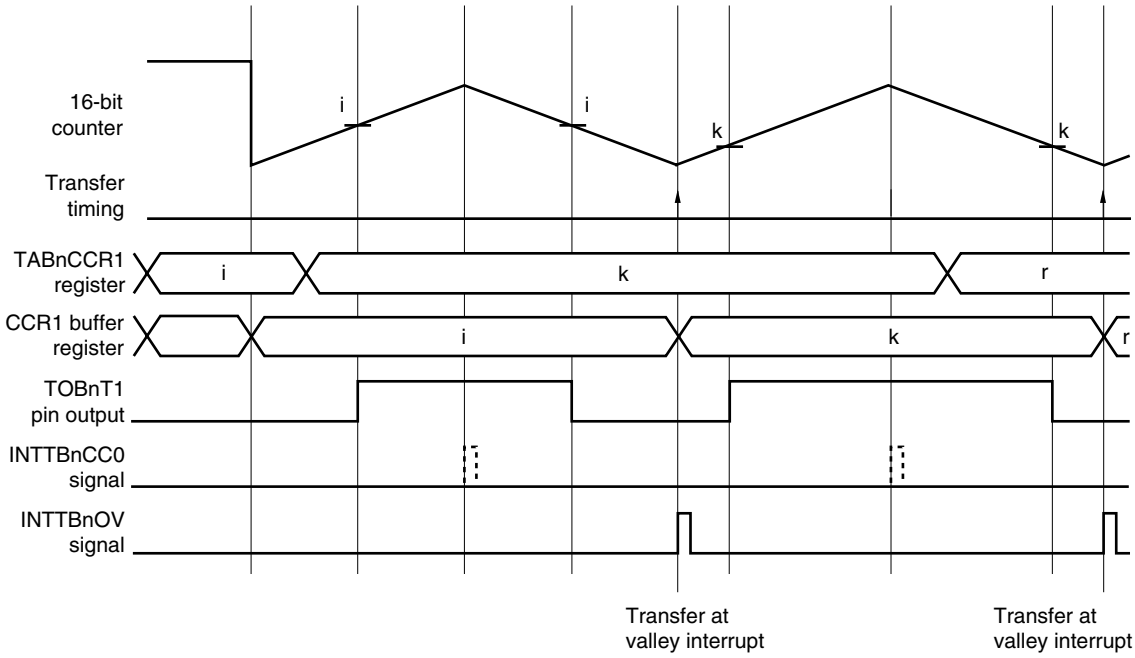


Remarks 1. : Culled interrupt

2. n = 0, 1

- Transfer at valley when valley interrupt is set
  Because the register is transferred at the transfer timing of the valley interrupt, a symmetrical triangular wave is output.

Figure 10-35. Rewriting TABnCCR1 Register
(TABnOPT1.TABnICE Bit = 1, TABnOPT1.TABnIOE Bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 = 00001)



Remarks 1. : Culled interrupt
2. n = 0, 1

(d) Rewriting TABnOPT1 register

Because a new interrupt culling value is transferred when the value of the interrupt culling counter matches the value of the 16-bit counter, the next interrupt and those that follow occur at the set interval.

For details of rewriting the TABnOPT1 register, see **10.4.3 Interrupt culling function**.

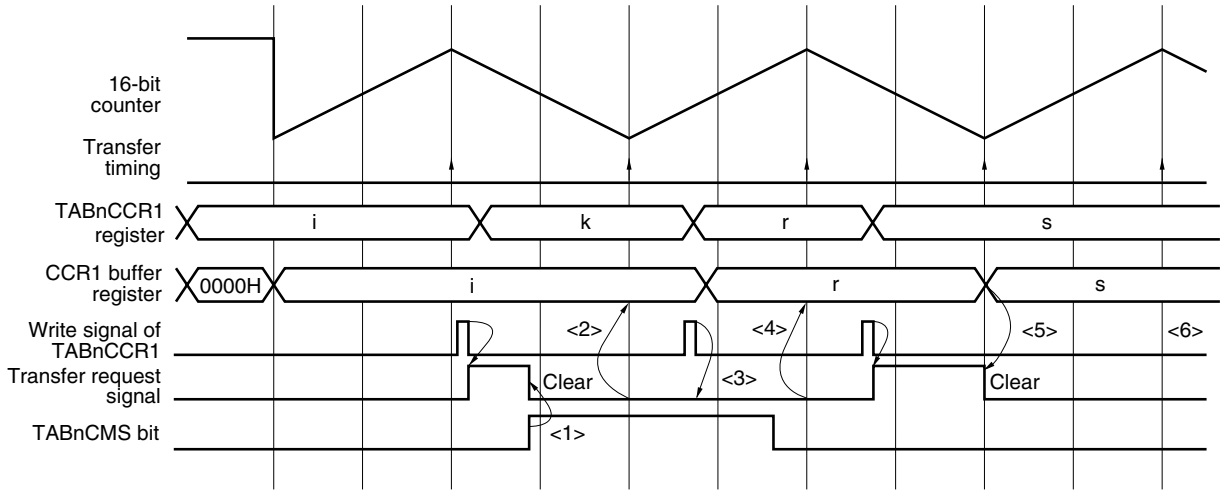
(4) Rewriting TABnOPT0.TABnCMS bit

The TABnCMS bit can select the anytime rewrite mode and batch rewrite mode. This bit can be rewritten during timer operation (when TABnCTL0.TABnCE bit = 1). However, the operation and caution illustrated in Figure 10-36 are necessary.

If the TABnCCR1 register is written when the TABnCMS bit is set to 0, a transfer request signal (internal signal) is set.

When the transfer request signal is set, the register is transferred at the next transfer timing, and the transfer request signal is cleared. This transfer request signal is also cleared when the TABnCMS bit is set to 1.

Figure 10-36. Rewriting TABnCMS Bit



<1> If the TABnCCR1 register is rewritten when the TABnCMS bit is 0, the transfer request signal is set.
If the TABnCMS bit is set to 1 in this status, the transfer request signal is cleared.

<2> The register is not transferred because the TABnCMS bit is set to 1 and the transfer request signal is cleared.

<3> The transfer request signal is not set even if the TABnCCR1 register is written when the TABnCMS bit is 1.

<4> The transfer request signal is not set even if the TABnCCR1 register is written when the TABnCMS bit is 1, so even if the TABnCMS bit is set to 0, transfer does not occur at the subsequent transfer timing.

<5> The transfer request signal is set if the TABnCCR1 register is written when the TABnCMS bit is 0.
Transfer is performed at the subsequent transfer timing and the transfer request signal is cleared.

<6> Once transfer has been performed, the transfer request signal is cleared. Therefore, transfer is not performed at the next transfer timing.

Remark n = 0, 1

10.4.5 TAAn tuning operation for A/D conversion start trigger signal output

This section explains the tuning operation of TAAn and TABn in the 6-phase PWM output mode.

In the 6-phase PWM output mode, the tuning operation is performed with TABn serving as the master and TAAn as a slave. The conversion start trigger signal of A/D converters 0 and 1 can be set as the A/D conversion start trigger source by the INTTAnCC0 and INTTAnCC1 signals of TAAn and the INTTBnOV and INTTBnCC0 signals of TABn.

Remark n = 0, 1

(1) Tuning operation starting procedure

The TAAn and TABn registers should be set using the following procedure to perform the tuning operation.

(a) Setting of TAAn register (stop the operations of TABn and TAAn (by setting the TABnCTL0.TABnCE bit and TAAnCTL0.TAAnCE bit to 0))

- Set the TAAnCTL1 register to 85H (set the tuning operation slave mode and free-running timer mode).
- Set the TAAnOPT0 register to 00H (select the compare register).
- Set an appropriate value to the TAAnCCR0 and TAAnCCR1 registers (set the default value for comparison for starting the operation).

(b) Setting of TABn register

- Set the TABnCTL1 register to 07H (set the master mode and 6-phase PWM output mode).
- Set an appropriate value to the TABnIOC0 register (set the output mode of TOBnT1 to TOBnT3). However, set the TABnOL0 bit to 0 and the TABnOE0 bit to 1 (enable positive phase output). Unless this setting is made, the crest interrupt (INTTBnCC0) and valley interrupt (INTTBnOV) do not occur. Consequently, the conversion start trigger signal of A/D converters 0 and 1 is not correctly generated.
- Clear the TABnIOC1 and TABnIOC2 registers to 00H (the TIBn0 to TIBn3, EVTBn, and TRGBn pins of TABn are not used).
- Clear the TABnOPT0 register to 00H (select the compare register).
- Set an appropriate value to the TABnCCR0 to TABnCCR3 registers (set the default value for comparison for starting the operation).
- Set the TABnCTL0 register to 0xH (set the TABnCE bit to 0 and the operating clock of TABn). The operating clock of TABn set by the TABnCTL0 register is also supplied to TAAn, and the count operation is performed at the same timing. The operating clock of TAAn set by the TAAnCTL0 register is ignored.

(c) Setting of TMQOPn (TMQn option) register

- Set an appropriate value to the TABnOPT1 and TABnOPT2 registers.
- Set an appropriate value to the TABnIOC3 register (set TOBnB1 to TOBnB3 in the output mode).
- Set an appropriate value to the TABnDTC register (set the default value for comparison for starting the operation).

(d) Setting of alternate function

- Select the alternate function of the port by setting the port to the port control mode.

(e) Set the TAAnCE bit to 1 and set the TABnCE bit to 1 immediately after that to start the 6-phase PWM output operation.

Rewriting the TABnCTL0, TABnCTL1, TABnIOC1, TABnIOC2, TAAnCTL0, and TAAnCTL1 registers is prohibited during operation. The operation and the PWM output waveform are not guaranteed if any of these registers is rewritten during operation. However, rewriting the TABnCTL0.TABnCE bit to clear it is permitted. Manipulating (reading/writing) the other TABn, TAAn, and TMQn option registers is prohibited until the TAAnCTL0.TAAnCE bit is set to 1 and then the TABnCE bit is set to 1.

(2) Tuning operation clearing procedure

To clear the tuning operation and exit the 6-phase PWM output mode, set the TAAn and TABn registers using the following procedure.

<1> Clear the TABnCTL0.TABnCE bit to 0 and stop the timer operation.
<2> Clear the TAAnCTL0.TAAnCE bit to 0 so that TAAn can be separated.
<3> Stop the timer output by using the TABnIOC0 register.
<4> Clear the TAAnCTL1.TAAnSYE bit to 0 to clear the tuning operation.

Caution Manipulating (reading/writing) the other TABn, TAAn, and TMQn option registers is prohibited until the TABnCE bit is set to 0 and then the TAAnCE bit is set to 0.

(3) When not tuning TAAn

When the match interrupt signal of TAAn is not necessary as the conversion trigger source that starts A/D converters 0 and 1, TAAn can be used independently as a separate timer without being tuned. In this case, the match interrupt signal of TAAn cannot be used as a trigger source to start A/D conversion in the 6-phase PWM output mode. Therefore, fix the TABnOPT2.TABnAT0 to TABnOPT2.TABnAT3 bits and the TABnOPT3.TABnAT4 to TABnOPT3.TABnAT7 bits to 0.

The other control bits can be used in the same manner as when TAAn is tuned.

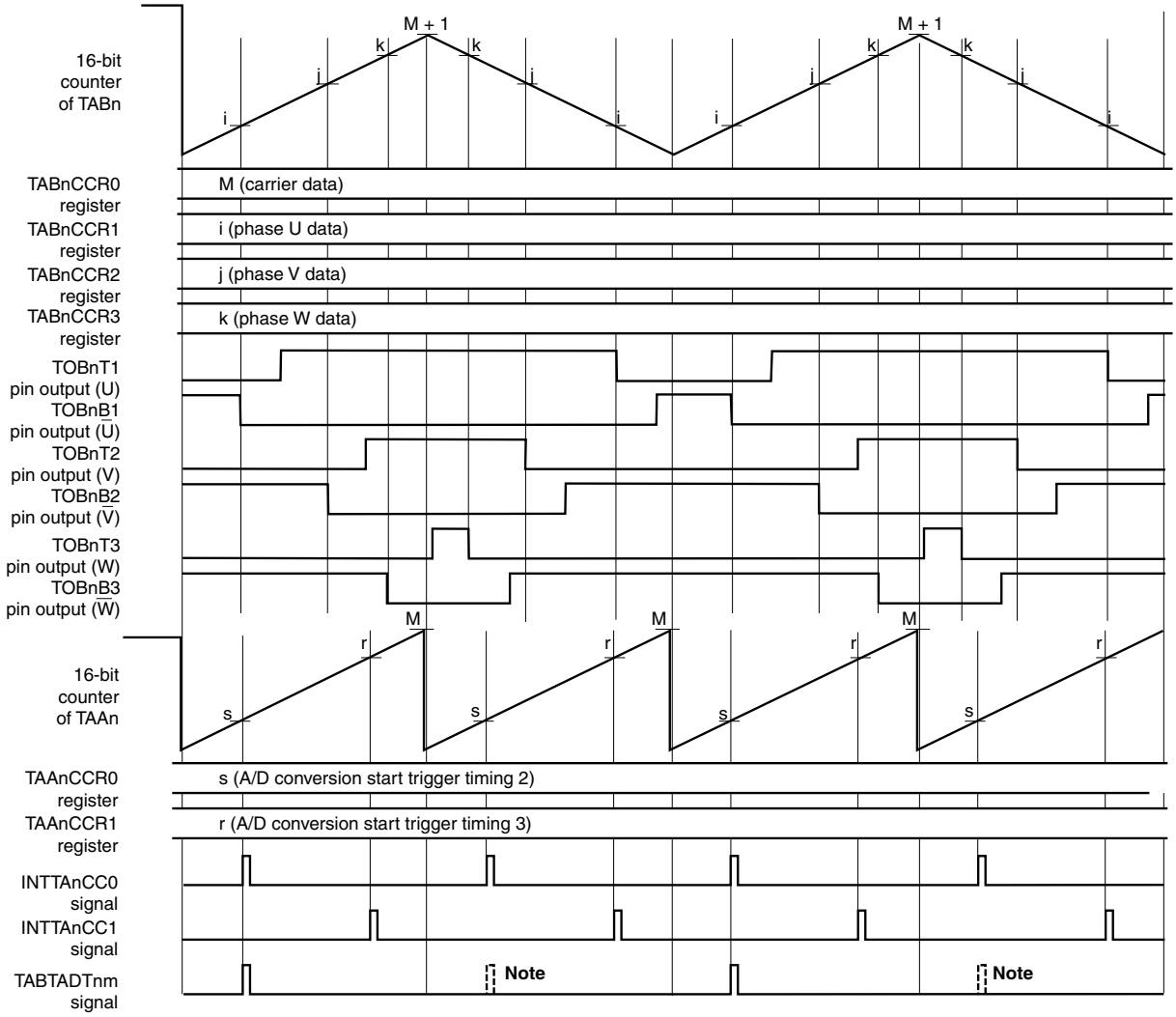
If TAAn is not tuned, the compare registers (TAAnCCR0 and TAAnCCR1) of TAAn are not affected by the settings of the TABnOPT0.TABnCMS and TABnOPT2.TABnRDE bits. For the initialization procedure when TAAn is not tuned, see (b) to (e) in **10.4.5 (1) Tuning operation starting procedure**. (a) is not necessary because it is a step used to set TAAn for the tuning operation.

(4) Basic operation of TAAn during tuning operation

The 16-bit counter of TAAn only counts up. The 16-bit counter is cleared by the set cycle value of the TABnCCR0 register and starts counting from 0000H again. The count value of this counter is the same as the value of the 16-bit counter of TAAn when it counts up. However, it is not the same when the 16-bit counter of TABn counts down.

- When TABn counts up (same value)
  16-bit counter of TABn: 0000H → M (up counting)
  16-bit counter of TAAn: 0000H → M (up counting)
- When TABn counts down (not same value)
  16-bit counter of TABn: M + 1 → 0001H (down counting)
  16-bit counter of TAAn: 0000H → M (up counting)

Figure 10-37. TAAn During Tuning Operation



Note The TABTADTn0 signal is masked by the TABnOPT2.TABnATM2 and TABnOPT2.TABnATM3 bits. The TABTADTn1 signal is masked by the TABnOPT3.TABnATM6 and TABnOPT3.TABnATM7 bits.

Remark n = 0, 1, m = 0, 1

10.4.6 A/D conversion start trigger output function

The V850E/IF3 and V850E/IG3 have a function to select four trigger sources (INTTBnOV, INTTBnCC0, INTTAnCC0, INTTAnCC1) to generate the A/D conversion start trigger signal (TABTADTn0, TABTADTn1) of A/D converters 0 and 1.

The trigger sources are specified by the TABnOPT2.TABnAT0 to TABnOPT2.TABnAT3 and TABnOPT3.TABnAT4 to TABnOPT3.TABnAT7 bits.

- TABnAT0, TABnAT4 bits = 1:
  A/D conversion start trigger signal generated when INTTBnOV (counter underflow) occurs.
- TABnAT1, TABnAT5 bits = 1:
  A/D conversion start trigger signal generated when INTTBnCC0 (cycle match) occurs.
- TABnAT2, TABnAT6 bits = 1:
  A/D conversion start trigger signal generated when INTTAnCC0 (match of TAAnCCR0 register of TAAn during tuning operation) occurs.
- TABnAT3, TABnAT7 bits = 1:
  A/D conversion start trigger signal generated when INTTAnCC1 (match of TAAnCCR1 register of TAAn during tuning operation) occurs.

The A/D conversion start trigger signals selected by the TABnAT0 to TABnAT3 and TABnAT4 to TABnAT7 bits are ORed and output. Therefore, two or more trigger sources can be specified at the same time.

The INTTBnOV and INTTBnCC0 signals selected by the TABnAT0, TABnAT1, TABnAT4, and TABnAT5 bits are culled interrupt signals.

Therefore, these signals are output after the interrupts have been culled and, unless interrupt output is enabled (TABnOPT1.TABnICE, TABnOPT1.TABnIOE bits), the A/D conversion start trigger is not output.

The trigger sources (INTTAnCC0 and INTTAnCC1) from TAAn have a function to mask the A/D conversion start trigger signal depending on the status of the up-count/down-count of the 16-bit counter, if so set by the TABnAT2, TABnAT3, TABnAT6, and TABnAT7 bits.

- TABnATM2, TABnATM6 bits:
  Correspond to the TABnAT2 and TABnAT6 bits and control INTTAnCC0 (match interrupt signal) of TAAn.
  - TABnATM2, TABnATM6 bits = 0
    The A/D conversion start trigger signal is output when the 16-bit counter counts up (TABnOPT0.TABnCUF bit = 0), and the A/D conversion start trigger signal is not output when the 16-bit counter counts down (TABnOPT0.TABnCUF bit = 1).
  - TABnATM2, TABnATM6 bits = 1
    The A/D conversion start trigger signal is output when the 16-bit counter counts down (TABnOPT0.TABnCUF bit = 1), and the A/D conversion start trigger signal is not output when the 16-bit counter counts up (TABnOPT0.TABnCUF bit = 0).

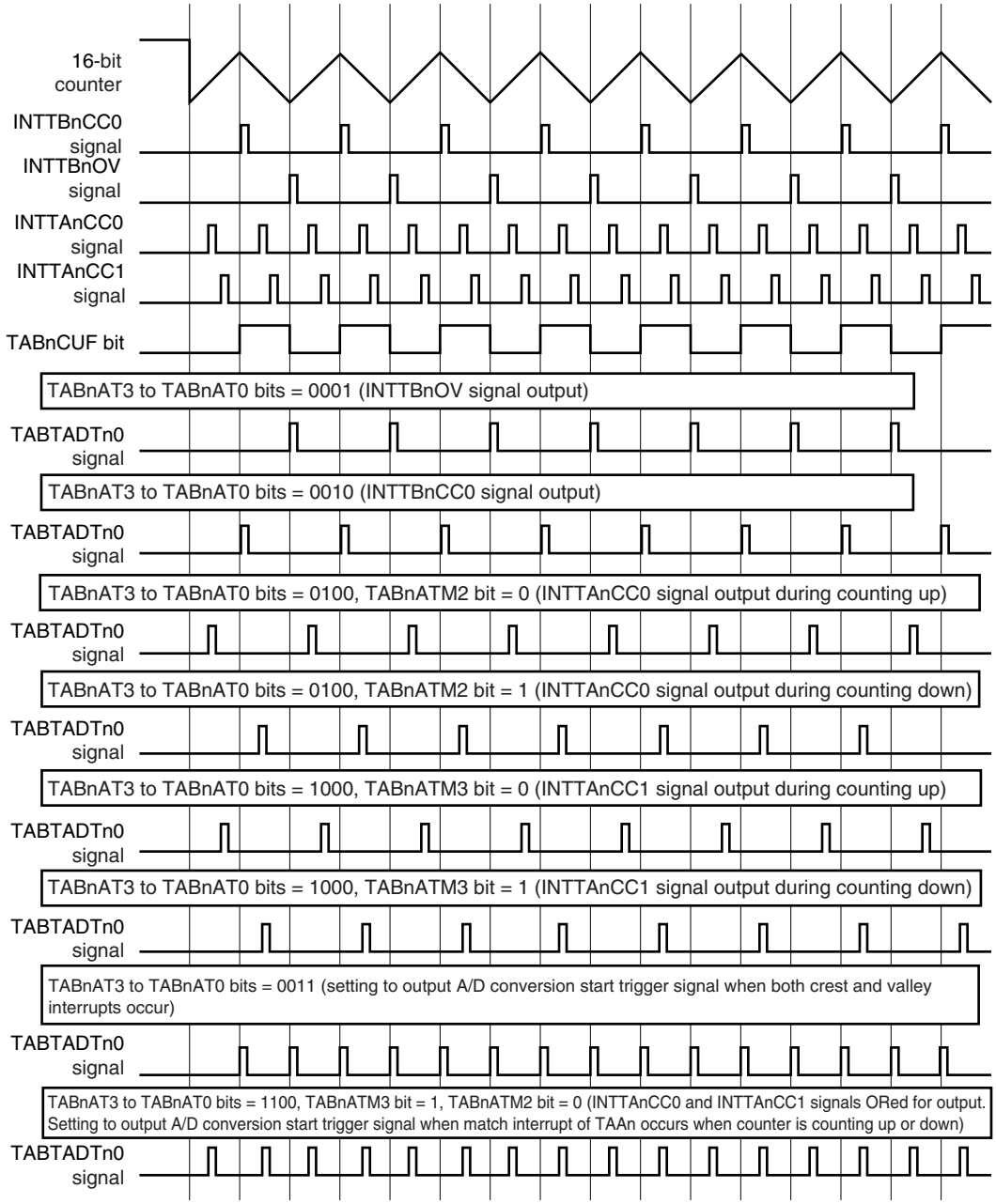
- TABnATM3, TABnATM7 bits:
  Correspond to the TABnAT3 and TABnAT7 bits and control INTTAnCC1 (match interrupt signal) of TAAn.
  - TABnATM3, TABnATM7 bits = 0
    The A/D conversion start trigger signal is output when the 16-bit counter counts up (TABnCUF bit = 0), and the A/D conversion start trigger signal is not output when the 16-bit counter counts down (TABnCUF bit = 1).
  - TABnATM3, TABnATM7 bits = 1
    The A/D conversion start trigger signal is output when the 16-bit counter counts down (TABnCUF bit = 1), and the A/D conversion start trigger signal is not output when the 16-bit counter counts up (TABnCUF bit = 0).

The TABnATM3, TABnATM2, TABnAT3 to TABnAT0, TABnATM7, TABnATM6, and TABnAT7 to TABnAT4 bits can be rewritten while the timer is operating. If the bit that sets the A/D conversion start trigger signal is rewritten while the timer is operating, the new setting is immediately reflected on the output status of the A/D conversion start trigger. These control bits do not have a transfer function and can be used only in the anytime rewriting mode.

Cautions 1. The A/D conversion start trigger signal output that is set by the TABnAT2, TABnAT3, TABnAT6, and TABnAT7 bits can be used only when TAAn is performing a tuning operation as the slave timer of TABn. If TABn and TAAn are not performing a tuning operation, or if a mode other than the 6-phase PWM output mode is used, the output cannot be guaranteed.

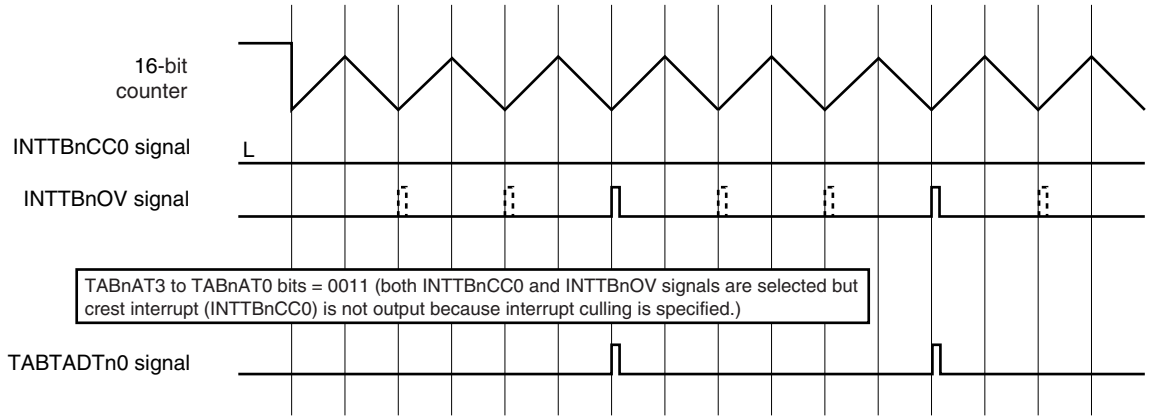
2. The TOBn0 signal output is internally used to identify whether the 16-bit counter is counting up or down. Therefore, enable TOBn0 pin output by setting the TABnIOC0.TABnOL0 bit to 0 and the TABnIOC0.TABnOE0 bit to 1.

Figure 10-38. Example of A/D Conversion Start Trigger (TABTADTn0) Signal Output (TABnOPT1.TABnICE Bit = 1, TABnOPT1.TABnIOE Bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 Bits = 00000: Without Interrupt Culling)



Remark n = 0, 1

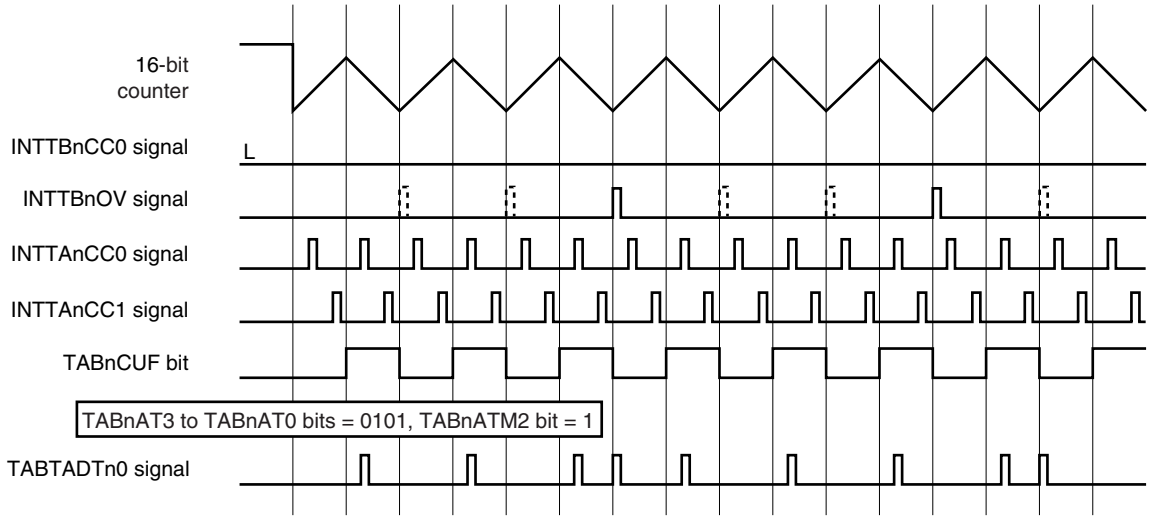
Figure 10-39. Example of A/D Conversion Start Trigger (TABTADTn0) Signal Output (TABnOPT1.TABnICE Bit = 0, TABnOPT1.TABnIOE Bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 Bits = 00010: With Interrupt Culling) (1)



Remarks 1. : Culled interrupt

2. n = 0, 1

Figure 10-40. Example of A/D Conversion Start Trigger (TABTADTn0) Signal Output (TABnOPT1.TABnICE Bit = 0, TABnOPT1.TABnIOE Bit = 1, TABnOPT1.TABnID4 to TABnOPT1.TABnID0 Bits = 00010: With Interrupt Culling) (2)



Caution The INTTBnCC0 signal is culled but the INTTAnCC0 signal is not.

Remarks 1. : Culled interrupt

2. n = 0, 1

(1) Operation under boundary condition (operation when 16-bit counter matches INTTAnCC0 signal)

Table 10-3. Operation When TABnCCR0 Register = M, TABnAT2 Bit = 1, TABnAT6 Bit = 1, TABnATM2 Bit = 0, TABnATM6 Bit = 0 (Up Counting Period Selected)

Value of TAAnCCR0 Register	Value of 16-bit Counter of TABn	Value of 16-bit Counter of TAAn	Status of 16-bit Counter of TABn	Output of INTTAnCC0 Signal from TABTADTnm Signal
0000H	0000H	0000H	–	Output
0000H	M + 1	0000H	–	Not output
0001H	0001H	0001H	Up count	Output
0001H	M	0001H	Down count	Not output
M	M	M	Up count	Output
M	0001H	M	Down count	Not output

Table 10-4. Operation When TABnCCR0 Register = M, TABnAT2 Bit = 1, TABnAT6 Bit = 1, TABnATM2 Bit = 1, TABnATM6 Bit = 1 (Down Counting Period Selected)

Value of TAAnCCR0 Register	Value of 16-bit Counter of TABn	Value of 16-bit Counter of TAAn	Status of 16-bit Counter of TABn	Output of INTTAnCC0 Signal from TABTADTnm Signal
0000H	0000H	0000H	–	Not output
0000H	M + 1	0000H	–	Output
0001H	0001H	0001H	Up count	Not output
0001H	M	0001H	Down count	Output
M	M	M	Up count	Not output
M	0001H	M	Down count	Output

Caution The TAAnCCRm register enables setting of "0" to "M" when the TABnCCR0 register = M. Setting of a value of "M + 1" or higher is prohibited.
If a value higher than "M + 1" is set, the 16-bit counter of TAAn is cleared by "M". Therefore, the TABTADTnm signal is not output.

Remark n = 0, 1, m = 0, 1

CHAPTER 11 WATCHDOG TIMER FUNCTIONS

11.1 Functions

The watchdog timer has the following functions.

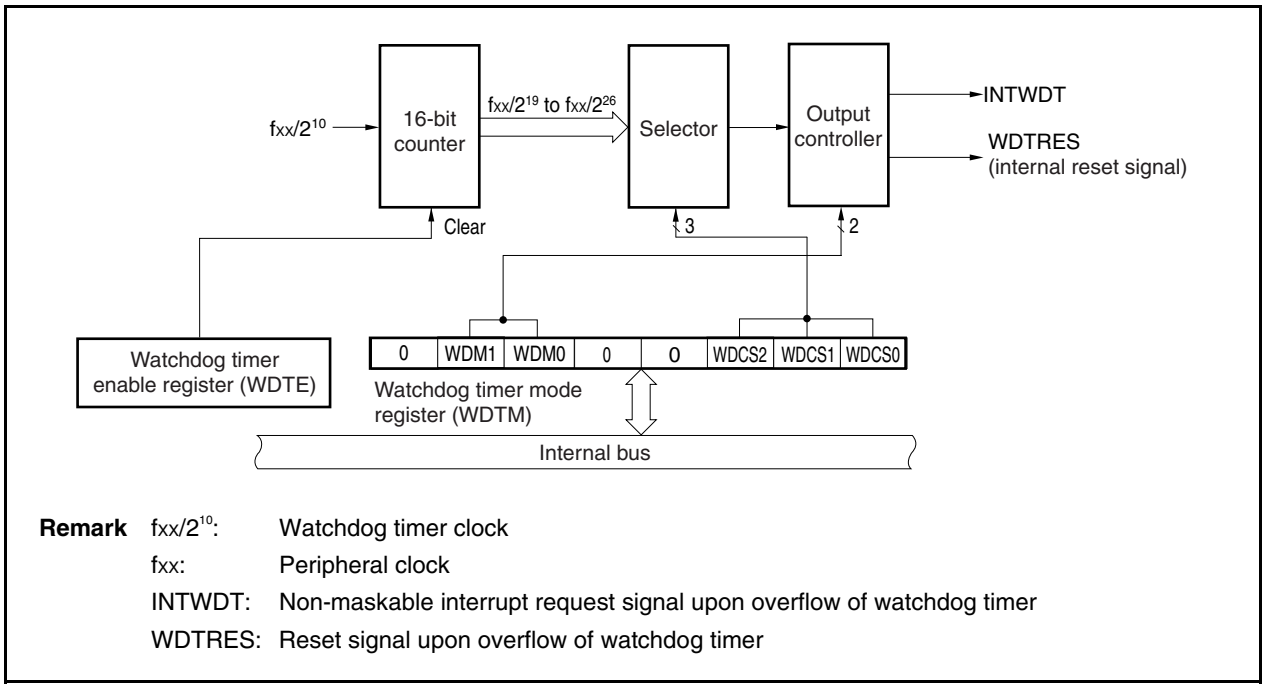
- Reset mode: Reset operation upon overflow of the watchdog timer (generation of WDTRES signal)
- Non-maskable interrupt request mode:
  Non-maskable interrupt operation upon overflow of the watchdog timer (generation of INTWDT signal)

Caution The watchdog timer is stopped after reset is released.
It starts operating when "ACH" is written to the WDTE register. Also, write to the WDTM register for verification purposes only once, even if the default settings (reset mode, interval time: $2^{26}/f_{XX}$) do not need to be changed.

11.2 Configuration

The block diagram of the watchdog timer is shown below.

Figure 11-1. Block Diagram of Watchdog Timer



Remark $f_{XX}/2^{10}$: Watchdog timer clock
f_{XX}: Peripheral clock
INTWDT: Non-maskable interrupt request signal upon overflow of watchdog timer
WDTRES: Reset signal upon overflow of watchdog timer

The watchdog timer consists of the following hardware.

Table 11-1. Configuration of Watchdog Timer

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

11.3 Control Registers

(1) Watchdog timer mode register (WDTM)

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

This register can be read or written in 8-bit units. This register can be read any number of times, but can be written only once following reset release; it cannot then be written a second or subsequent time.

Reset sets this register to 67H.

After reset: 67H R/W Address: FFFFF6D0H

	7	6	5	4	3	2	1	0
WDTM	0	WDM1	WDM0	0	0	WDCS2	WDCS1	WDCS0

WDM1	WDM0	Selection of operation mode of watchdog timer
0	0	Stop operation
0	1	Non-maskable interrupt request mode (generation of INTWDT signal)
1	×	Reset mode (generation of WDTRES signal)

Cautions 1. For details of the WDCS2 to WDCS0 bits, see Table 11-2 Overflow Time.

2. Be sure to clear bits 3, 4, and 7 to "0".

Table 11-2. Overflow Time

WDCS2	WDCS1	WDCS0	Overflow Time	f_{XX} = 64 MHz	f_{XX} = 32 MHz
0	0	0	$2^{19}/f_{XX}$	8.2 ms	16.4 ms
0	0	1	$2^{20}/f_{XX}$	16.4 ms	32.8 ms
0	1	0	$2^{21}/f_{XX}$	32.8 ms	65.5 ms
0	1	1	$2^{22}/f_{XX}$	65.5 ms	131.1 ms
1	0	0	$2^{23}/f_{XX}$	131.1 ms	262.1 ms
1	0	1	$2^{24}/f_{XX}$	262.1 ms	524.3 ms
1	1	0	$2^{25}/f_{XX}$	524.3 ms	1,048.5 ms
1	1	1	$2^{26}/f_{XX}$	1,048.5 ms	2,097.1 ms

(2) Watchdog timer enable register (WDTE)

The counter of the watchdog timer is cleared and counting restarted by writing “ACH” to the WDTE register.

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

Reset sets this register to 1AH.

After reset: 1AH R/W Address: FFFFF6D1H

	7	6	5	4	3	2	1	0
WDTE								

Cautions 1. If “ACH” is written to the WDTE register to enable the watchdog timer operation and then a value other than “ACH” is written to the WDTE register, a non-maskable interrupt request signal (INTWDT) or a reset signal (WDTRES) is generated due to watchdog timer overflow, depending on the specification of the WDTM.WDM1 and WDTM.WDM0 bits.

2. When the WDTE register is read or written in 1-bit units, an internal reset signal is output.

3. The read value of the WDTE register is “1AH” before the watchdog timer operates, and “9AH” after it operates. The value read from this register is different from the written value (ACH).

11.4 Operation

The watchdog timer is stopped after reset is released.

The WDTM register can be written only once after reset is released. If the register is written a second time after the watchdog timer has started operating, a non-maskable interrupt request signal (INTWDT) or a reset signal (WDTRES) is generated due to watchdog timer overflow, depending on the specification of the WDTM.WDM1 and WDTM.WDM0 bits. The INTWDT or WDTRES signal is also generated if the same value is written to the register. The operation is not guaranteed if the register is written three or more times.

To use the watchdog timer, write the operation mode and the interval time to the WDTM register in 8-bit units. After this, the operation of the watchdog timer cannot be stopped.

To not use the watchdog timer, write 00H to the WDTM register.

11.5 Caution

The cycle of the non-maskable interrupt request signal (INTWDT) that is generated due to watchdog timer overflow can be calculated from “Interval time set to WDTM register + 2^7 peripheral clock pulse width”, if INTWDT occurs successively without the watchdog timer being cleared.

Note that the pulse width until generation of the first interrupt request signal after the watchdog timer has been started is not included.

CHAPTER 12 A/D CONVERTERS 0 AND 1

12.1 Features

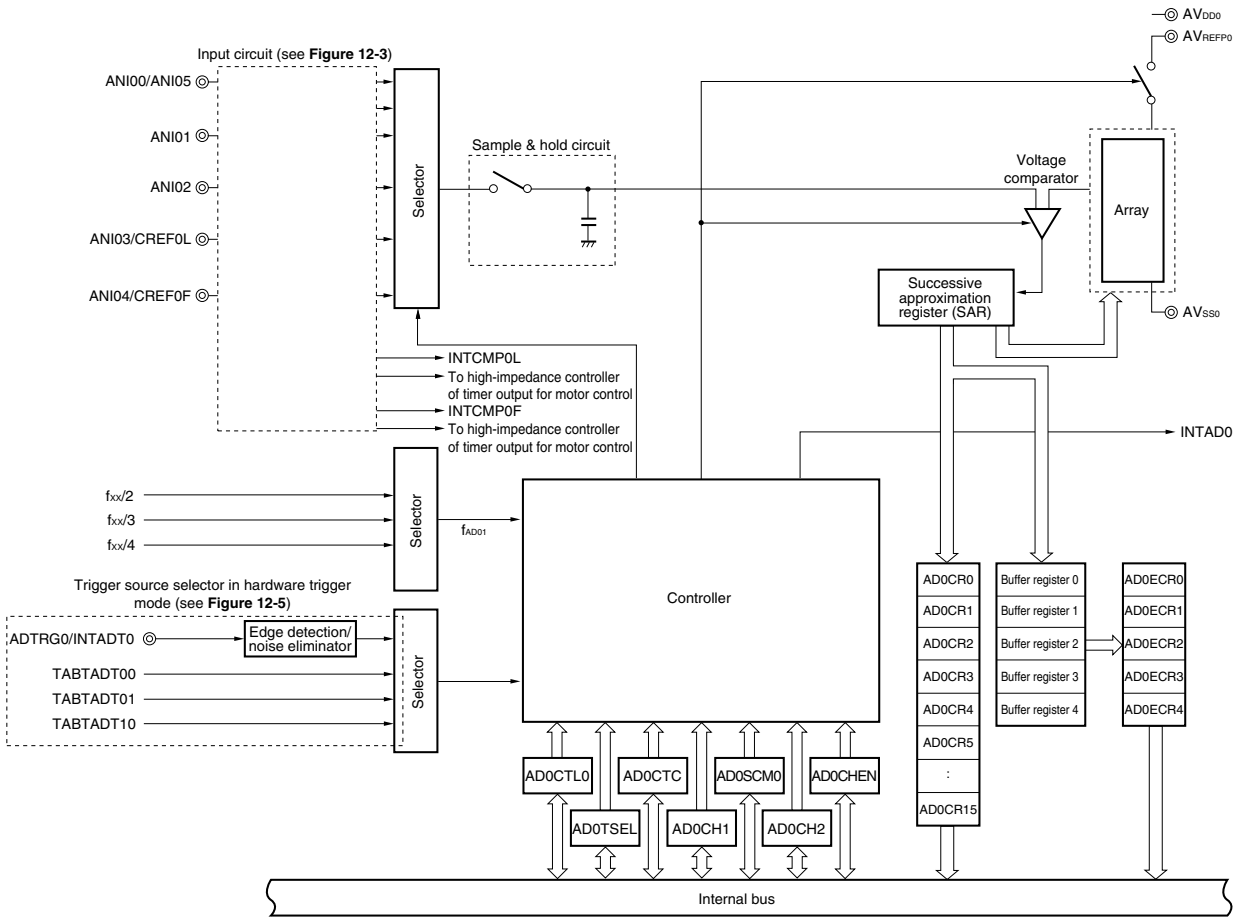
- Two 12-bit resolution A/D converter circuits (A/D converters 0 and 1)
  Simultaneous sampling of two circuits possible
- Analog input
  - When comparator is not used
    Total of 10 channels in two circuits
    A/D converter 0: ANI00/ANI05, ANI01 to ANI04 (5 channels)
    A/D converter 1: ANI10/ANI15, ANI11/ANI16, ANI12/ANI17, ANI13, ANI14 (5 channels)
  - When comparator is used
    Total of 6/8 channels in two circuits
    [6 channels (when comparators of low-range and full-range are used)]
    A/D converter 0: ANI00/ANI05, ANI01, ANI02 (3 channels)
    A/D converter 1: ANI10/ANI15, ANI11/ANI16, ANI12/ANI17 (3 channels)
    [8 channels (when comparator of low-range or full-range is used)]
    A/D converter 0: ANI00/ANI05, ANI01, ANI02, ANI03 or ANI04 (4 channels)
    A/D converter 1: ANI10/ANI15, ANI11/ANI16, ANI12/ANI17, ANI13 or ANI14 (4 channels)
- A/D conversion result registers
  12 bits × 16 + 12 bits × 16
  A/D converter 0: AD0CR0 to AD0CR15
  A/D converter 1: AD1CR0 to AD1CR15
- A/D conversion result extension registers
  Can be used only in the extension buffer mode
  12 bits × 5 + 12 bits × 5
  A/D converter 0: AD0ECR0 to AD0ECR4
  A/D converter 1: AD1ECR0 to AD1ECR4
- Operation modes
  - Normal operation modes
    A/D trigger mode
    A/D trigger polling mode
    Hardware trigger mode
  - Extension operation modes
    Conversion channel specification mode
    Extension buffer mode
- Operational amplifiers for input level amplification (×2.5 to ×10)
  These channels can be used only when the operational amplifier for input level amplification is used.
  Total of 4 units in two circuit
  A/D converter 0: ANI05 (1 unit)
  A/D converter 1: ANI15 to ANI17 (3 units)

- Overvoltage detection comparator
  - These channels can be used only when the overvoltage detection comparator is used.
  - Total of 4 units in two circuit
    A/D converter 0: 1 unit
    A/D converter 1: 3 units
  - Reference voltage
    Input voltage range from CREF0L, CREF1L pins (low-range side) = $0.02AV_{DD} + 0.1$ to $0.5AV_{DD} - 0.1$ V
    Input voltage range from CREF0F, CREF1F pins (full-range side) = $0.02AV_{DD} + 0.1$ to $0.92AV_{DD} - 0.1$ V
  - An interrupt occurs when an overvoltage is detected. Interrupt requests are output by two overvoltage detection signal pins ANI00/ANI05 (full-range side and low-range side) and as the logical sum (OR) of the overvoltage detection signals of three channels, ANI10/ANI15, ANI11/ANI16, and ANI12/ANI17, or two output signals (full-range side and low-range side) of a logical product (AND).
  - The output of a timer for motor control can be set to a high-impedance state when an overvoltage is detected.
- Successive approximation method
- Operating voltage range
  $EV_{DD0} = EV_{DD1} = EV_{DD2}$ (V850E/IG3 only) = $AV_{DD0} = AV_{DD1} = AV_{REFP0} = AV_{REFP1}$ = 4.0 to 5.5 V (target)

12.2 Configuration

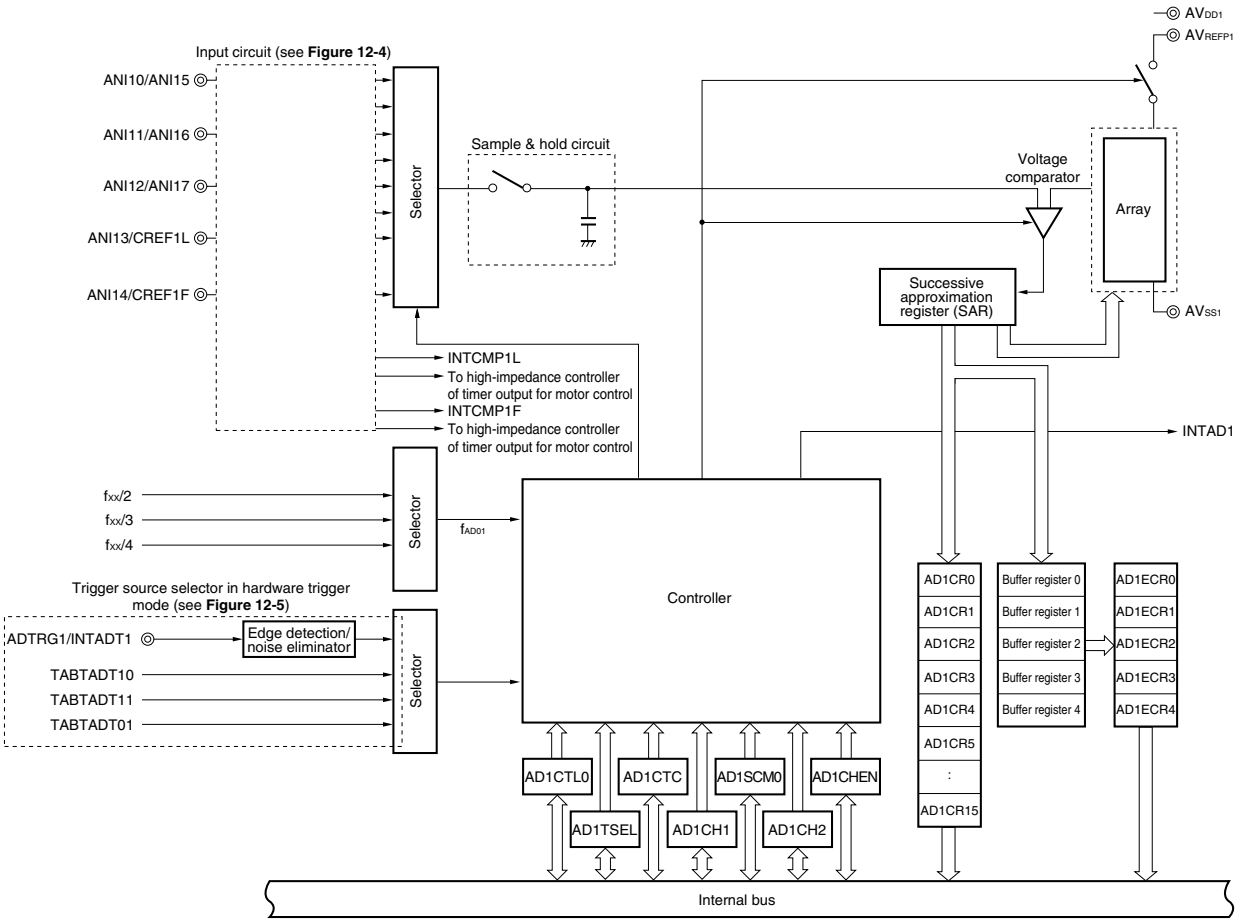
The block diagram is shown below.

Figure 12-1. Block Diagram of A/D Converter 0



Remark fXX: Peripheral clock
fAD01: Basic clock
Buffer registers 0 to 4: A/D0 conversion result extension buffer registers 0 to 4

Figure 12-2. Block Diagram of A/D Converter 1



Remark fxx: Peripheral clock
f_{AD01}: Basic clock
Buffer registers 0 to 4: A/D1 conversion result extension buffer registers 0 to 4

Cautions 1. If there is noise at the analog input pins (ANI00 to ANI05, ANI10 to ANI17) or at the A/D converter reference voltage input pins (AV_{REFP0}, AV_{REFP1}), that noise may generate an illegal conversion result.

Software processing will be needed to avoid a negative effect on the system from this illegal conversion result.

An example of this software processing is shown below.

- **Take the average result of a number of A/D conversions and use that as the A/D conversion result.**
- **Execute a number of A/D conversions consecutively and use those results, omitting any exceptional results that may have been obtained.**
- **If an A/D conversion result that is judged to have generated a system malfunction is obtained, be sure to recheck the system malfunction before performing malfunction processing.**

2. Do not apply a voltage outside the AV_{SSn} to AV_{REFPn} range to the pins that are used as input pins of A/D converters 0 and 1 (n = 0, 1).

Figure 12-3. Block Diagram of Operational Amplifier for Input Level Amplification and Overvoltage Detection Comparato

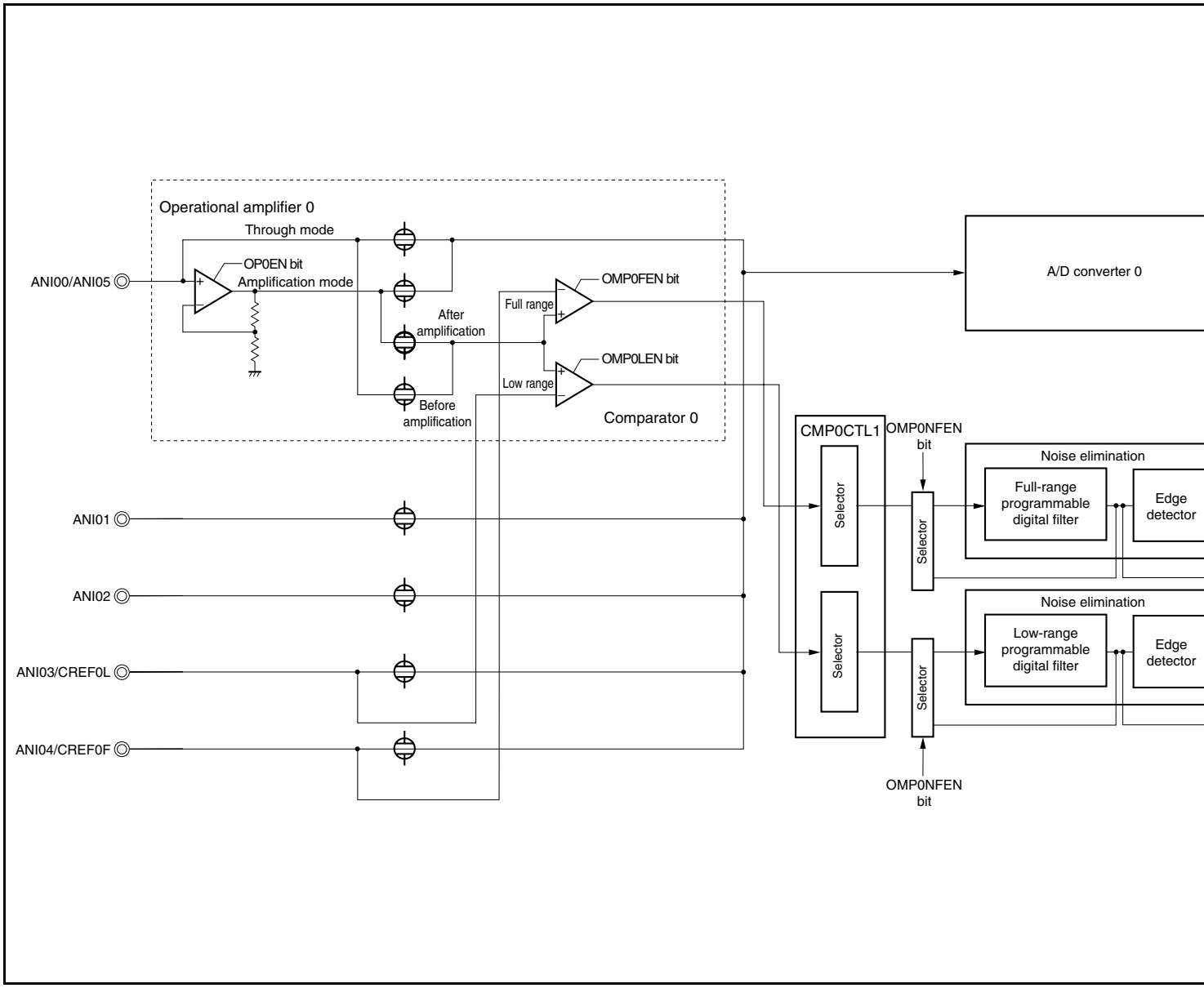


Figure 12-4. Block Diagram of Operational Amplifier for Input Level Amplification and Overvoltage Detection Comparator

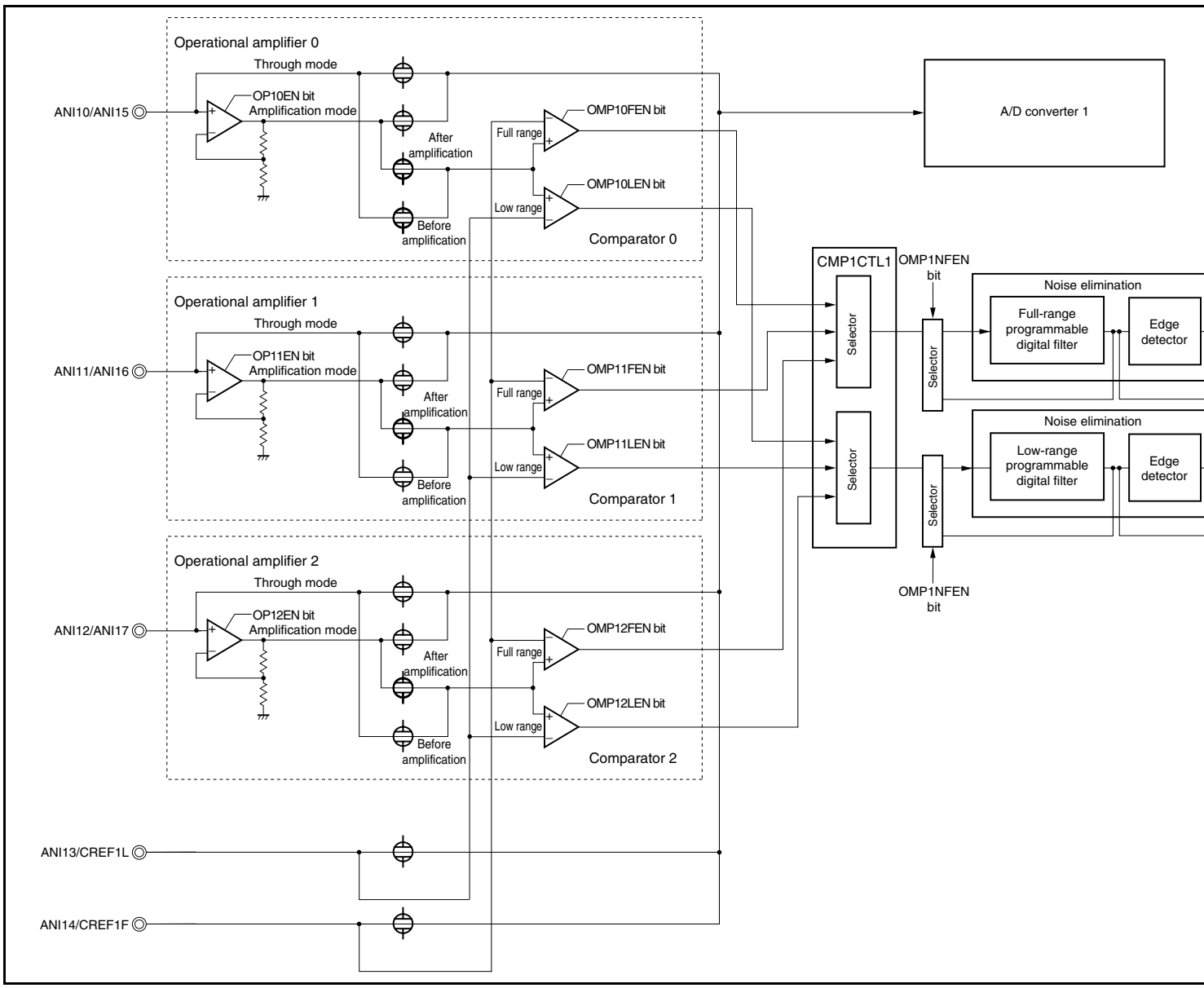
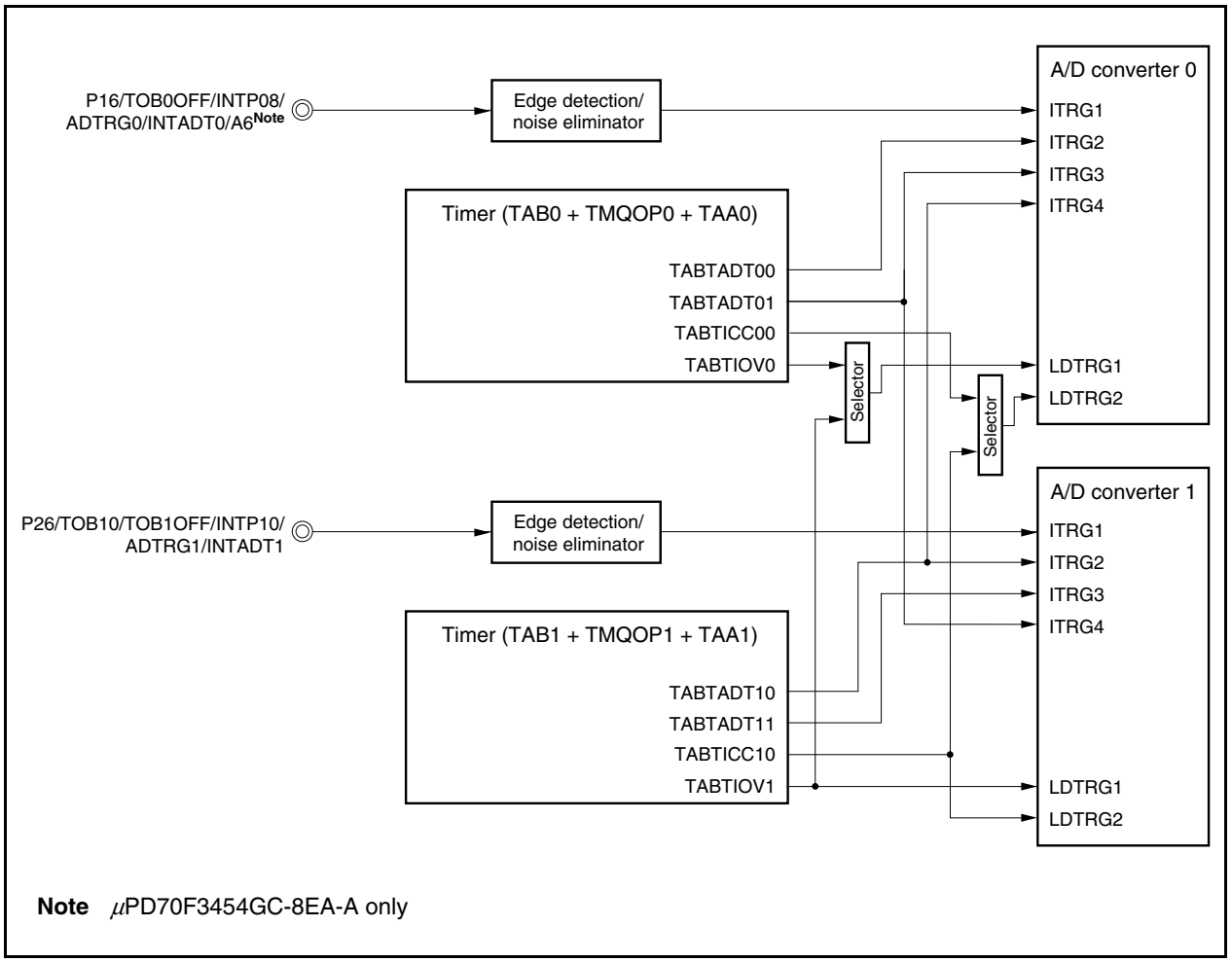


Figure 12-5. Block Diagram of Trigger Source Selector in Hardware Trigger Mode



Note μPD70F3454GC-8EA-A only

A/D converters 0 and 1 consist of the following hardware.

Table 12-1. Configuration of A/D Converters 0 and 1

Item	Configuration
Analog input	When comparator is not used: ANI00/ANI05, ANI01 to ANI04, ANI10/ANI15 to ANI12/ANI17, ANI13, ANI14 (total of 10 channels in two circuits) When comparator is used (when comparators of low-range and full-range are used): ANI00/ANI05, ANI01, ANI02, ANI10/ANI15 to ANI12/ANI17 (total of 6 channels in two circuits) When comparator is used (when comparator of low-range or full-range is used): ANI00/ANI05, ANI01, ANI02, ANI03 or ANI04, ANI10/ANI15 to ANI12/ANI17, ANI13 or ANI14 (total of 8 channels in two circuits)
Registers	Successive approximation register (SAR) A/Dn conversion result registers 0 to 15 (ADnCR0 to ADnCR15) A/Dn conversion result registers 0H to 15H (ADnCR0H to ADnCR15H) A/Dn conversion result extension registers 0 to 4 (ADnECR0 to ADnECR4) (only in extension operation mode (extension buffer mode)) A/Dn conversion result extension registers 0H to 4H (ADnECR0H to ADnECR4H) (only in extension operation mode (extension buffer mode))
Control registers	A/D converter n scan mode register (ADnSCM) A/D converter n scan mode register L (ADnSCML) A/D converter n scan mode register H (ADnSCMH) A/D converter n conversion time control register (ADnCTC) A/D converter n conversion channel specification register (ADnCHEN) A/D converter n conversion channel specification register L (ADnCHENL) A/D converter n conversion channel specification register H (ADnCHENH) A/D converter n control register (ADnCTL0) A/D converter n trigger select register (ADnTSEL) A/D converter n channel specification register 1 (ADnCH1) A/D converter n channel specification register 2 (ADnCH2) A/D converter n flag register (ADnFLG) A/D converter n flag buffer register (ADnFLGB) A/DLDTRG1 input select register (ADLTS1) A/DLDTRG2 input select register (ADLTS2) Operational amplifier n control register 0 (OPnCTL0) Comparator n control register 0 (CMPnCTL0) Comparator n control register 1 (CMPnCTL1) Comparator n control register 2 (CMPnCTL2) Comparator n control register 3 (CMPnCTL3) A/D converter n clock select register (ADnOCKS) A/D trigger falling edge specification register (ADTF) A/D trigger rising edge specification register (ADTR) Comparator output interrupt rising edge specification register (CMPOR) Comparator output interrupt falling edge specification register (CMPOF) Comparator output digital noise elimination register nL (CMPNFCnL) Comparator output digital noise elimination register nF (CMPNFCnF)

Remark n = 0, 1

(1) Selector

The selector selects the analog input pin according to the mode set by the ADnSCM, ADnCTC, ADnCHEN, ADnCTL0, ADnTSEL, ADnCH1, ADnCH2, ADLTS1, ADLTS2, and ADnOCKS registers and sends the input to the sample & hold circuit (n = 0, 1).

ANI05, ANI15 to ANI17 are provided with an operational amplifier for input level amplification and an overvoltage detection comparator. The operational amplifier and comparator of each analog input pin can be specified to be on or off. The amplification (gain) of the operational amplifier can be selected from 2.5 to 10 times for ANI05, ANI15 to ANI17.

(2) Sample & hold circuit

The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the voltage comparator. When the operational amplifier for input level amplification is used, the gain specified by the OPnCTL0.OPnGA3 to OPnCTL0.OPnGA0 bits × the input voltage is sampled. This circuit also holds the sampled analog input voltage during A/D conversion.

(3) Voltage comparator

This comparator compares the voltage generated from the voltage tap of the array with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 AV_{REFPn}) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage (1/2 AV_{REFPn}), the MSB of the SAR is reset.

After that, bit 10 of the SAR is automatically set, and the next comparison is made. The voltage tap of the array is selected by the value of bit 11, to which the result has been already set.

Bit 11 = 0: (1/4 AV_{REFPn})
Bit 11 = 1: (3/4 AV_{REFPn})

The voltage tap of the array and the analog input voltage are compared and bit 10 of the SAR is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of array: Bit 10 = 1
Analog input voltage ≤ Voltage tap of array: Bit 10 = 0

Comparison is continued like this to bit 0 of the SAR.

(4) Array

The array generates the comparison voltage input from an analog input pin.

(5) Successive approximation register (SAR)

The SAR is a 12-bit register that sets voltage tap data whose values from the array match the voltage values of the analog input pins, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR (conversion results) are held in A/Dn conversion result registers 0 to 15 (ADnCR0 to ADnCR15) (n = 0, 1). In the extension buffer mode, however, the conversion result is stored in A/Dn conversion result extension buffer registers 0 to 4 and, when selection load trigger x is generated, shifted to and stored in the ADnECR0 to ADnECR4 registers (x = 1, 2). When all the specified A/D conversion operations have ended, an A/Dn conversion end interrupt request signal (INTADn) is generated.

(6) A/Dn conversion result registers 0 to 15 (ADnCR0 to ADnCR15), A/Dn conversion result registers 0H to 15H (ADnCR0H to ADnCR15H) (n = 0, 1)

The ADnCR0 to ADnCR15 and ADnCR0H to ADnCR15H registers are registers that hold the A/D conversion results. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR) and stored in the higher 12 bits of the ADnCR0 to ADnCR15 registers. The lower 4 bits of these registers are always 0 when read.

The higher 8 bits of the result of A/D conversion are read from the ADnCR0H to ADnCR15H registers.

To read the result of A/D conversion in 16-bit units, specify the ADnCR0 to ADnCR15 registers. To read the higher 8 bits, specify the ADnCR0H to ADnCR15H registers.

(7) A/Dn conversion result extension registers 0 to 4 (ADnECR0 to ADnECR4), A/Dn conversion result extension registers 0H to 4H (ADnECR0H to ADnECR4H) (n = 0, 1)

The ADnECR0 to ADnECR4 and ADnECR0H to ADnECR4H registers are registers that hold the A/D conversion results. These registers can be used only in extension buffer mode. When A/D conversion is completed, the A/D conversion result is stored in the A/Dn conversion result extension buffer register. If selection load trigger 1 is generated after that, the A/D conversion result is shifted from A/Dn conversion result extension buffer registers 0 to 2 to the higher 12 bits of the ADnECR0 to ADnECR2 registers for storage. Bits 1 to 3 are always 0 when read. If selection load trigger 2 is generated, the A/D conversion result is shifted from A/Dn conversion result extension buffer registers 3 and 4 to the higher 12 bits of the ADnECR3 and ADnECR4 registers. Bits 1 to 3 are always 0 when read.

The higher 8 bits of the result of A/D conversion are read from the ADnECR0H to ADnECR4H registers.

To read the result of A/D conversion in 16-bit units, specify the ADnECR0 to ADnECR4 registers. To read the higher 8 bits, specify the ADnECR0H to ADnECR4H registers.

(8) ANI00 to ANI05, ANI10 to ANI17 pins (n = 0, 1)

The ANI00 to ANI05 and ANI10 to ANI17 pins are analog input pins for A/D converters 0 and 1. They input the analog signals to be A/D converted.

Caution Make sure that the voltages input to the ANI00 to ANI05 and ANI10 to ANI17 pins do not exceed the rated values. If a voltage higher than or equal to AV_{REFPn} or lower than or equal to AV_{SSn} (even within the range of the absolute maximum ratings) is input to a channel, the conversion value of the channel is undefined, and the conversion values of the other channels may also be affected.

(9) AV_{REFPn} pin (n = 0, 1)

This pin is used for inputting the reference voltage of A/D converters 0 and 1. It converts signals input to the analog input pin to digital signals based on the voltage applied between AV_{REFPn} and AV_{SSn} (n = 0, 1).

Always make the potential at this pin the same as that at the EV_{DD0}, EV_{DD1}, and EV_{DD2} (V850E/IG3 only) pins even when A/D converters 0 and 1 are not used.

The operating voltage range of the AV_{REFPn} pin is $EV_{DD0} = EV_{DD1} = EV_{DD2}$ (V850E/IG3 only) = $AV_{DDn} = AV_{REFPn}$ = 4.0 to 5.5 V (target).

(10) AV_{SSn} pin (n = 0, 1)

This is the ground pin of A/D converters 0 and 1. Always make the potential at this pin the same as that at the EV_{SS0}, EV_{SS1}, and EV_{SS2} (V850E/IG3 only) pins even when A/D converters 0 and 1 are not used.

(11) AV_{DDn} pin (n = 0, 1)

This pin is the analog power supply pin of A/D converters 0 and 1.

Supply the same potential to the AV_{DD0} and AV_{DD1} pins.

Always make the potential at this pin the same as that at the EV_{DD0}, EV_{DD1}, and EV_{DD2} (V850E/IG3 only) pins even when A/D converters 0 and 1 are not used.

The operating voltage range of the AV_{DDn} pin is $EV_{DD0} = EV_{DD1} = EV_{DD2}$ (V850E/IG3 only) = $AV_{REFPn} = AV_{DDn}$ = 4.0 to 5.5 V (target).

(12) CREFnL, CREFnF pins (n = 0, 1)

The CREFnL pin supplies the low range of the reference voltage of the comparator for overvoltage detection, and the CREFnF pin supplies the full range of the reference voltage (input voltage range of CREF0L and CREF1L pins = $0.02AV_{DD}$ + 0.1 to $0.5AV_{DD}$ − 0.1 V, input range of CREF0F and CREF1F pins = $0.02AV_{DD}$ + 0.1 to $0.92AV_{DD}$ − 0.1 V).

12.3 Control Registers

A/D converters 0 and 1 are controlled by the following registers.

- A/D converter n scan mode register (ADnSCM)
- A/D converter n scan mode register L (ADnSCML)
- A/D converter n scan mode register H (ADnSCMH)
- A/D converter n conversion time control register (ADnCTC)
- A/D converter n conversion channel specification register (ADnCHEN)
- A/D converter n conversion channel specification register L (ADnCHENL)
- A/D converter n conversion channel specification register H (ADnCHENH)
- A/D converter n control register (ADnCTL0)
- A/D converter n trigger select register (ADnTSEL)
- A/D converter n channel specification registers 1 and 2 (ADnCH1, ADnCH2)
- A/D converter n flag register (ADnFLG)
- A/D converter n flag buffer register (ADnFLGB)
- A/DLDTRG1 input select register (ADLTS1)
- A/DLDTRG2 input select register (ADLTS2)
- Operational amplifier n control register 0 (OPnCTL0)
- Comparator n control registers 0 to 3 (CMPnCTL0 to CMPnCTL3)
- A/D converter n clock select register (ADnOCKS)
- A/D trigger falling edge specification register (ADTF)
- A/D trigger rising edge specification register (ADTR)
- Comparator output interrupt rising edge specification register (CMPOR)
- Comparator output interrupt falling edge specification register (CMPOF)
- Comparator output digital noise elimination registers nL, nF (CMPNFCnL, CMPNFCnF)

The following registers are also used.

- A/Dn conversion result registers 0 to 15 (ADnCR0 to ADnCR15)
- A/Dn conversion result registers 0H to 15H (ADnCR0H to ADnCR15H)
- A/Dn conversion result extension registers 0 to 4 (ADnECR0 to ADnECR4)
- A/Dn conversion result extension registers 0H to 4H (ADnECR0H to ADnECR4H)

(1) A/D converter n scan mode register (ADnSCM)

The ADnSCM register is a register that specifies the normal operation mode and controls conversion operations.

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

When the higher 8 bits of the ADnSCM register are used as the ADnSCMH register and the lower 8 bits, as the ADnSCML register, these registers can be read or written in 1-bit or 8-bit units. However, bit 14 is read-only.

Reset sets this register to 0000H.

(1/2)

After reset: 0000H R/W Address: AD0SCM FFFFF220H, AD1SCM FFFFF2A0H

	<15>	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADnSCM (n = 0, 1)	ADnCE	ADnCS	0	0	0	ADnPLM	ADnTRG1	ADnTRG0	ADnPS	0	0	0	0	0	0Note 1	0

Note 1. When using A/D converters 1 and 0, be sure to set bit 1 to "1".
This setting can be performed at the same time as other ADnSCM register bits.

ADnCE	A/D conversion operation control
0	Stop conversion operation
1	Start conversion operation

ADnCS	Status of A/D converter nNote 2
0	A/D conversion stopped
1	A/D conversion operating (remains “1” even when the channel is changed during successive conversion)

ADnPLM	ADnTRG1	ADnTRG0	Normal operation mode specification
0	0	0	A/D trigger mode
0	0	1	Hardware trigger modeNote 3
1	0	0	A/D trigger polling mode
Other than above			Setting prohibited

ADnPS	A/D power save mode specification
0	A/D power save mode
1	A/D operational mode

Notes 2. The ADnCS bit is set to 1 five basic clocks (f_{AD01}) after the ADnCE bit has been set to 1 and A/D conversion has been started.

A/D conversion is started when a trigger signal, such as one from a timer, is input in the hardware trigger mode, conversion channel specification mode, or extension buffer mode. In the A/D trigger mode and A/D trigger polling mode, it is started when the ADnCE bit is 1.

3. In the extended operation mode (conversion channel specification mode or extension buffer mode), be sure to set the hardware trigger mode.

(2/2)

Cautions 1. In the A/D trigger mode or the A/D trigger polling mode, conversion is triggered when 1 is written to the ADnCE bit.

In the hardware trigger mode, the conversion channel specification mode, or the extension buffer mode, the trigger signal wait state starts when 1 is written to the ADnCE bit.

The ADnCE bit is not cleared to 0 even after the A/Dn conversion end interrupt request signal (INTADn) is generated in all modes. To stop the A/D conversion operation, therefore, write 0 to the ADnCE bit.

2. If the ADnSCM0 register is written during A/D conversion operation (ADnCS bit = 1), the operation is performed as follows in each mode. The corresponding conversion result register is undefined during A/D conversion operation.

- **In A/D trigger mode, A/D trigger polling mode**
  A/D conversion is stopped and executed again from the beginning.
- **In hardware trigger mode, conversion channel specification mode, extension buffer mode**
  A/D conversion is stopped and the trigger standby state is restored again.

3. Make sure that time of at least five basic clocks (f_{AD01}) passes before successively writing data to the ADnSCM0 register when the conversion operation is enabled (ADnCE bit = 1). Otherwise, the register may not be set correctly.

The register can be successively written if the ADnCE bit is set to 1 after the ADSCMn register is written when ADnCE bit = 0.

4. The ADnCS bit remains set (1) when the conversion channel is changed during successive conversion.

5. It is recommended to set the A/D power save mode (ADnPS bit = 0) when the A/D converter is not used.

6. The setting procedure is as follows when an A/D conversion operation is started (after reset release and after recovery from the A/D power save mode (ADnPS bit = 0)).

<1> Select an input clock (f_{AD01}) by using the ADnOCKS register and set the ADnOCKSEN bit to 1 (enable supplying the operating clock to A/D converter n).
<2> Set the ADnPS bit to 1 (A/D operation mode).
<3> Wait for 1 μs or longer after <2>.
<4> Initialize A/D converters 0 and 1.
<5> Set the ADnCE bit to 1 (enable conversion operation).

7. The setting procedure is as follows when an A/D conversion operation is stopped.

<1> Set the ADnCE bit to 0 (stop conversion operation).
<2> Set the ADnPS bit to 0 (A/D power save mode).
<3> Set the ADnOCKS.ADnOCKSEN bit to 0 (stop supplying the operating clock to A/D converter n).

8. It is recommended to set the A/D power save mode even in the IDLE and STOP modes. Follow the setting procedure in Caution 6 above when releasing the IDLE or STOP mode by using the reset signal.

9. Be sure to set bits 0 to 6 and 11 to 13 to "0".

(2) A/D converter n conversion time control register (ADnCTC)

The ADnCTC register is a register that specifies the number of A/D conversion clocks and A/D conversion time.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: AD0CTC FFFFF222H, AD1CTC FFFFF2A2H

	7	6	5	4	3	2	1	0
ADnCTC (n = 0, 1)	0	0	0	0	ADnFR3	ADnFR2	ADnFR1	ADnFR0

Cautions 1. See Table 12-2 Number of A/D Conversion Clocks and A/D Conversion Time for the ADnFR3 to ADnFR0 bits.

2. Set the ADnFR3 to ADnFR0 bits when the ADnSCM.ADnCE bit = 0 (conversion operation is stopped).

3. Be sure to set bits 4 to 7 to "0".

Table 12-2. Number of A/D Conversion Clocks and A/D Conversion Time

ADnFR3	ADnFR2	ADnFR1	ADnFR0	Number of A/D Conversion Clocks[Note 1]	A/D Conversion Time (μs)[Note 2]	
					f_{AD01} = 16 MHz (f_{XX} = 64 MHz)	f_{AD01} = 12 MHz (f_{XX} = 48 MHz)
0	0	0	0	89	5.56	7.42
0	0	0	1	88	5.50	7.33
0	0	1	0	57	3.56	4.75
0	0	1	1	56	3.50	4.67
0	1	0	0	41	2.56	3.42
0	1	0	1	40	2.50	3.33
0	1	1	0	35	2.19	2.92
0	1	1	1	34	2.13	2.83
1	0	0	0	34	2.13	2.83
1	0	0	1	33	2.06	2.75
1	0	1	0	33	2.06	2.75
1	0	1	1	32	2.00	2.67
1	1	0	0	32	2.00	2.67
1	1	0	1	31	Setting prohibited	2.58
1	1	1	0	31	Setting prohibited	2.58
1	1	1	1	30	Setting prohibited	2.50

Notes 1. The number of clocks (f_{AD01}) from the start to the end of A/D conversion.

The number of clocks (f_{AD01}) per conversion during successive conversion (1-channel conversion (repeat), multiple channel conversion, or multiple channel conversion (repeat)) is the same.

2. Set the A/D conversion time in a range of 2 to 7.42 μs.

A/D Conversion time = $1/f_{AD01} \times$ Number of A/D conversion clocks

(3) A/D converter n conversion channel specification register (ADnCHEN)

The ADnCHEN register is a register that specifies the analog input pin, number of conversion times, and conversion result register.

This register is used to specify an analog input pin in the A/D trigger mode, A/D trigger polling mode, and hardware trigger mode. The ADnCRm register corresponds to an analog input pin on a one-to-one basis. Use the bits (AD0CHEN00 to AD0CHEN05 and AD1CHEN00 to AD1CHEN07) corresponding to the ANI00 to ANI05 and ANI10 to ANI17 pins. If two or more analog input pins are specified, they are sequentially selected, starting from the one with the lowest number, for conversion (when AD1CHEN register = 004DH: ANI10 → ANI12 → ANI13 → ANI16). If an analog input pin that is not specified is skipped during successive conversion.

In the conversion channel specification mode, specify the number of times of conversion and a conversion result register. Specify an analog input pin by using the ADnCH1 register. A value set to the lower bits of the ADnCHEN register, justified to the lowest bit, is the number of times of conversion. These bits correspond to the ADnCRm and ADnCHmH registers on a one-to-one basis.

Because the ADnCHEN register is of master/slave configuration, a new analog input pin can be set to the master register during A/D conversion operation. The set value of the master register is transferred to a slave register after completion of A/D conversion (after the A/Dn conversion end interrupt request signal (INTADn) is generated).

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

When the higher 8 bits of the ADnCHEN register are used as the ADnCHENH register and the lower 8 bits, as the ADnCHENL register, these registers can be read or written in 1-bit or 8-bit units.

Reset sets this register to 0000H.

After reset: 0000H R/W Address: AD0CHEN FFFFF224H, AD1CHEN FFFFF2A4H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADnCHEN (n = 0, 1)	ADn CHEN 15	ADn CHEN 14	ADn CHEN 13	ADn CHEN 12	ADn CHEN 11	ADn CHEN 10	ADn CHEN 9	ADn CHEN 8	ADn CHEN 7	ADn CHEN 6	ADn CHEN 5	ADn CHEN 4	ADn CHEN 3	ADn CHEN 2	ADn CHEN 1	ADn CHEN 0

Remark See **Table 12-3 Specifying Analog Input Pin in A/D Trigger Mode, A/D Trigger Polling Mode, and Hardware Trigger Mode** for how to specify an analog input pin in the A/D trigger mode, A/D trigger polling mode, and hardware trigger mode. For how to specify the number of times of conversion and the A/D conversion result register in the conversion channel specification mode, see **Table 12-4 Correspondence Among Set Value of ADnCHEN Register, Number of Times of Conversion, and A/D Conversion Result Register in Conversion Channel Specification Mode**.

Cautions 1. The A/D conversion operation is prohibited when the ADnCHEN register = 0000H. If the ADnCHEN register = 0000H, the operation is the same as when the ADnCHEN register = 0001H.

2. Do not write the ADnCHEN register when the ADSCMn.ADnPS bit = 0. If it is written, the CPU deadlocks.

3. To change the setting of the ADnCHEN register when the ADnSCM.ADnCE bit = 1 in the hardware trigger mode, be sure to set the ADnCE bit to 0.

Table 12-3. Specifying Analog Input Pin in A/D Trigger Mode, A/D Trigger Polling Mode, and Hardware Trigger Mode

ADnCHENm Bit	Specification of Analog Input Pin
0	Specifying ANInk pin is prohibited.
1	Specifying ANInk pin is enabled.

Remark A/D converter 0: n = 0, k = 0 to 5, m = 0 to 15
A/D converter 1: n = 1, k = 0 to 7, m = 0 to 15

Table 12-4. Correspondence Among Set Value of ADnCHEN Register, Number of Times of Conversion, and A/D Conversion Result Register in Conversion Channel Specification Mode

ADnCHEN Register Value	Number of Times of Conversion	A/D Conversion Result Register	
0001H	1	ADnCR0	ADnCR0H
0003H	2	ADnCR0, ADnCR1	ADnCR0H, ADnCR1H
0007H	3	ADnCR0 to ADnCR2	ADnCR0H to ADnCR2H
000FH	4	ADnCR0 to ADnCR3	ADnCR0H to ADnCR3H
001FH	5	ADnCR0 to ADnCR4	ADnCR0H to ADnCR4H
003FH	6	ADnCR0 to ADnCR5	ADnCR0H to ADnCR5H
007FH	7	ADnCR0 to ADnCR6	ADnCR0H to ADnCR6H
00FFH	8	ADnCR0 to ADnCR7	ADnCR0H to ADnCR7H
01FFH	9	ADnCR0 to ADnCR8	ADnCR0H to ADnCR8H
03FFH	10	ADnCR0 to ADnCR9	ADnCR0H to ADnCR9H
07FFH	11	ADnCR0 to ADnCR10	ADnCR0H to ADnCR10H
0FFFH	12	ADnCR0 to ADnCR11	ADnCR0H to ADnCR11H
1FFFH	13	ADnCR0 to ADnCR12	ADnCR0H to ADnCR12H
3FFFH	14	ADnCR0 to ADnCR13	ADnCR0H to ADnCR13H
7FFFH	15	ADnCR0 to ADnCR14	ADnCR0H to ADnCR14H
FFFFH	16	ADnCR0 to ADnCR15	ADnCR0H to ADnCR15H
Others	Setting prohibited		

Caution An analog input pin is specified by the ADnCH1 register in the conversion channel specification mode.

Remark n = 0, 1

(4) A/Dn conversion result registers 0 to 15, 0H to 15H (ADnCR0 to ADnCR15, ADnCR0H to ADnCR15H)

The ADnCRm and ADnCRmH registers are registers that hold the A/D conversion results in the A/D trigger mode, A/D trigger polling mode, hardware trigger mode, or conversion channel specification mode. Sixteen of these registers are provided per circuit, and two circuits are available. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR) and stored in the higher 12 bits of the ADnCRm register. The lower 4 bits of these registers are always 0 when read.

The higher 8 bits of A/D conversion result are read to the ADnCRmH register.

These registers can only be read in 16-bit or 8-bit units. When the A/D conversion results are read in 16-bit units, the ADnCRm register is specified, and when the higher 8 bits are read, the ADnCRmH register is specified.

Reset sets these registers to 0000H.

Remark While the result of A/D conversion is stored in the ADnCRm register, a read access to the same register is held pending. The pending read access is executed after the A/D conversion result is stored. Similarly, storing the result of A/D conversion in the ADnCRm register is held pending while a read access to that register is made. The pending A/D conversion result storing processing is executed after completion of the read access.

After reset: 0000H R Address: AD0CR0 FFFFF200H, AD0CR1 FFFFF202H,
AD0CR2 FFFFF204H, AD0CR3 FFFFF206H,
AD0CR4 FFFFF208H, AD0CR5 FFFFF20AH,
AD0CR6 FFFFF20CH, AD0CR7 FFFFF20EH,
AD0CR8 FFFFF210H, AD0CR9 FFFFF212H,
AD0CR10 FFFFF214H, AD0CR11 FFFFF216H,
AD0CR12 FFFFF218H, AD0CR13 FFFFF21AH,
AD0CR14 FFFFF21CH, AD0CR15 FFFFF21EH,
AD1CR0 FFFFF280H, AD1CR1 FFFFF282H,
AD1CR2 FFFFF284H, AD1CR3 FFFFF286H,
AD1CR4 FFFFF288H, AD1CR5 FFFFF28AH,
AD1CR6 FFFFF28CH, AD1CR7 FFFFF28EH,
AD1CR8 FFFFF290H, AD1CR9 FFFFF292H,
AD1CR10 FFFFF294H, AD1CR11 FFFFF296H,
AD1CR12 FFFFF298H, AD1CR13 FFFFF29AH,
AD1CR14 FFFFF29CH, AD1CR15 FFFFF29EH

ADnCRm (n = 0, 1) (m = 0 to 15)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADnCRm11	ADnCRm10	ADnCRm9	ADnCRm8	ADnCRm7	ADnCRm6	ADnCRm5	ADnCRm4	ADnCRm3	ADnCRm2	ADnCRm1	ADnCRm0	0	0	0	0

After reset: 0000H R Address: AD0CR0H FFFFF201H, AD0CR1H FFFFF203H,
AD0CR2H FFFFF205H, AD0CR3H FFFFF207H,
AD0CR4H FFFFF209H, AD0CR5H FFFFF20BH,
AD0CR6H FFFFF20DH, AD0CR7H FFFFF20FH,
AD0CR8H FFFFF211H, AD0CR9H FFFFF213H,
AD0CR10H FFFFF215H, AD0CR11H FFFFF217H,
AD0CR12H FFFFF219H, AD0CR13H FFFFF21BH,
AD0CR14H FFFFF21DH, AD0CR15H FFFFF21FH,
AD1CR0H FFFFF281H, AD1CR1H FFFFF283H,
AD1CR2H FFFFF285H, AD1CR3H FFFFF287H,
AD1CR4H FFFFF289H, AD1CR5H FFFFF28BH,
AD1CR6H FFFFF28DH, AD1CR7H FFFFF28FH,
AD1CR8H FFFFF291H, AD1CR9H FFFFF293H,
AD1CR10H FFFFF295H, AD1CR11H FFFFF297H,
AD1CR12H FFFFF299H, AD1CR13H FFFFF29BH,
AD1CR14H FFFFF29DH, AD1CR15H FFFFF29FH

ADnCRmH (n = 0, 1) (m = 0 to 15)

7	6	5	4	3	2	1	0
ADnCRm11	ADnCRm10	ADnCRm9	ADnCRm8	ADnCRm7	ADnCRm6	ADnCRm5	ADnCRm4

The correspondence between the analog input pins and the A/D conversion result registers in the A/D trigger mode, A/D trigger polling mode, hardware trigger mode, and conversion channel specification mode is shown below.

Table 12-5. Correspondence Between Analog Input Pins and A/D Conversion Result Registers in A/D Trigger Mode, A/D Trigger Polling Mode, Hardware Trigger Mode

A/D Converter	Analog Input Pin	A/D Conversion Result Register
A/D converter 0	ANI00	AD0CR0, AD0CR0H
	ANI01	AD0CR1, AD0CR1H
	ANI02	AD0CR2, AD0CR2H
	ANI03	AD0CR3, AD0CR3H
	ANI04	AD0CR4, AD0CR4H
	ANI05	AD0CR5, AD0CR5H
A/D converter 1	ANI10	AD1CR0, AD1CR0H
	ANI11	AD1CR1, AD1CR1H
	ANI12	AD1CR2, AD1CR2H
	ANI13	AD1CR3, AD1CR3H
	ANI14	AD1CR4, AD1CR4H
	ANI15	AD1CR5, AD1CR5H
	ANI16	AD1CR6, AD1CR6H
	ANI17	AD1CR7, AD1CR7H

Table 12-6. Correspondence Between Analog Input Pins and A/D Conversion Result Registers in Conversion Channel Specification Mode

ADnCHEN Register Set Value	Analog Input Pin	A/D Conversion Result Register	
0001H	Set by ADnCH1.ADnTRGCH12 to ADnCH1.ADnTRGCH10 bits	ADnCR0	ADnCR0H
0003H		ADnCR0, ADnCR1	ADnCR0H, ADnCR1H
0007H		ADnCR0 to ADnCR2	ADnCR0H to ADnCR2H
000FH		ADnCR0 to ADnCR3	ADnCR0H to ADnCR3H
001FH		ADnCR0 to ADnCR4	ADnCR0H to ADnCR4H
003FH		ADnCR0 to ADnCR5	ADnCR0H to ADnCR5H
007FH		ADnCR0 to ADnCR6	ADnCR0H to ADnCR6H
00FFH		ADnCR0 to ADnCR7	ADnCR0H to ADnCR7H
01FFH		ADnCR0 to ADnCR8	ADnCR0H to ADnCR8H
03FFH		ADnCR0 to ADnCR9	ADnCR0H to ADnCR9H
07FFH		ADnCR0 to ADnCR10	ADnCR0H to ADnCR10H
0FFFH		ADnCR0 to ADnCR11	ADnCR0H to ADnCR11H
1FFFH		ADnCR0 to ADnCR12	ADnCR0H to ADnCR12H
3FFFH		ADnCR0 to ADnCR13	ADnCR0H to ADnCR13H
7FFFH		ADnCR0 to ADnCR14	ADnCR0H to ADnCR14H
FFFFH		ADnCR0 to ADnCR15	ADnCR0H to ADnCR15H
0001H	Set by ADnCH1.ADnTRGCH16 to ADnCH1.ADnTRGCH14 bits	ADnCR0	ADnCR0H
0003H		ADnCR0, ADnCR1	ADnCR0H, ADnCR1H
0007H		ADnCR0 to ADnCR2	ADnCR0H to ADnCR2H
000FH		ADnCR0 to ADnCR3	ADnCR0H to ADnCR3H
001FH		ADnCR0 to ADnCR4	ADnCR0H to ADnCR4H
003FH		ADnCR0 to ADnCR5	ADnCR0H to ADnCR5H
007FH		ADnCR0 to ADnCR6	ADnCR0H to ADnCR6H
00FFH		ADnCR0 to ADnCR7	ADnCR0H to ADnCR7H
01FFH		ADnCR0 to ADnCR8	ADnCR0H to ADnCR8H
03FFH		ADnCR0 to ADnCR9	ADnCR0H to ADnCR9H
07FFH		ADnCR0 to ADnCR10	ADnCR0H to ADnCR10H
0FFFH		ADnCR0 to ADnCR11	ADnCR0H to ADnCR11H
1FFFH		ADnCR0 to ADnCR12	ADnCR0H to ADnCR12H
3FFFH		ADnCR0 to ADnCR13	ADnCR0H to ADnCR13H
7FFFH		ADnCR0 to ADnCR14	ADnCR0H to ADnCR14H
FFFFH		ADnCR0 to ADnCR15	ADnCR0H to ADnCR15H
Others	Setting prohibited		

Remark n = 0, 1

(5) A/D converter n control register (ADnCTL0)

The ADnCTL0 register is a register that specifies the operation mode.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: AD0CTL0 FFFFF230H, AD1CTL0 FFFFF2B0H

	7	6	5	4	3	2	1	0
ADnCTL0 (n = 0, 1)	0	0	0	0	0	0	ADnMD1	ADnMD0

ADnMD1	ADnMD0	Extended operating mode specification
0	0	Normal operating mode
0	1	Setting prohibited
1	0	Conversion channel specification mode
1	1	Extension buffer mode

Cautions 1. Set the ADnMD1 and ADnMD0 bits when the ADnSCM.ADnCE bit = 0 (conversion operation is stopped) (the same value can be written to these bits when the ADnCE bit = 1 (conversion operation is enabled)).

2. In the conversion channel specification mode and extension buffer mode, start of A/D conversion is delayed up to 1.5 basic clocks (f_{AD01}) as compared with the normal operating mode.

3. Be sure to set the hardware trigger mode in the conversion channel specification mode and extension buffer mode.

(6) A/D converter n trigger select register (ADnTSEL)

The ADnTSEL register is a register that specifies trigger in the hardware trigger mode and conversion channel specification mode, and trigger (selection trigger 1, selection trigger 2, selection load trigger 1, and selection load trigger 2) in the extension buffer mode.

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

Reset sets this register to 10H.

After reset: 10H R/W Address: AD0TSEL FFFFF231H, AD1TSEL FFFFF2B1H

ADnTSEL (n = 0, 1)

7	6	5	4	3	2	1	0
ADn LDTSEL2 Note	0	ADn TRGSEL21 Note	ADn TRGSEL20 Note	ADn LDTSEL1 Note	0	ADn TRGSEL11	ADn TRGSEL10

ADnLDTSEL2 Note	Specification of selection load trigger 2 for ADnECR3, ADnECR4 registers
0	LDTRG1
1	LDTRG2

ADnTRGSEL21 Note	ADnTRGSEL20 Note	Specification of selection trigger 2 for ADnECR3, ADnECR4 registers
0	0	ITRG1
0	1	ITRG2
1	0	ITRG3
1	1	ITRG4

ADnLDTSEL1 Note	Specification of selection load trigger 1 for ADnECR0 to ADnECR2 registers
0	LDTRG1
1	LDTRG2

ADnTRGSEL11	ADnTRGSEL10	• In hardware trigger mode or conversion channel specification mode: Trigger specification • In expansion buffer mode: Specification of selection trigger 1 for ADnECR0 to ADnECR2 registers
0	0	ITRG1
0	1	ITRG2
1	0	ITRG3
1	1	ITRG4

Note Be sure to set bits 3, 5, and 7 to “0” and set bit 4 to “1” in the hardware trigger mode and conversion channel specification mode.

Caution Set the ADnTSEL register when the ADnSCM.ADnCE bit = 0 (conversion operation is stopped) (the same value can be written to the register when the ADnCE bit = 1 (conversion operation is enabled)).

(7) A/D converter n channel specification register 1 (ADnCH1)

The ADnCH1 register is a register that specifies the analog input pin for selection trigger 1 in the conversion channel specification mode and extension buffer mode.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: AD0CH1 FFFFF232H, AD1CH1 FFFFF2B2H

	7	6	5	4	3	2	1	0
ADnCH1 (n = 0, 1)	0	ADn TRGCH16	ADn TRGCH15	ADn TRGCH14	0	ADn TRGCH12	ADn TRGCH11	ADn TRGCH10

ADnTRGCH16	ADnTRGCH15	ADnTRGCH14	Specification of analog input pin for selection trigger 1
0	0	0	ANIn0
0	0	1	ANIn1
0	1	0	ANIn2
0	1	1	ANIn3
1	0	0	ANIn4
1	0	1	ANIn5
1	1	0	ANI16
1	1	1	ANI17

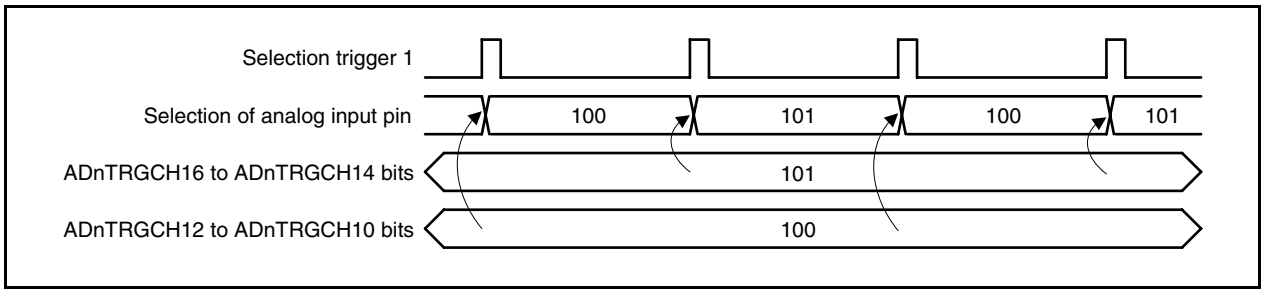
ADnTRGCH12	ADnTRGCH11	ADnTRGCH10	Specification of analog input pin for selection trigger 1
0	0	0	ANIn0
0	0	1	ANIn1
0	1	0	ANIn2
0	1	1	ANIn3
1	0	0	ANIn4
1	0	1	ANIn5
1	1	0	ANI16
1	1	1	ANI17

Cautions 1. Set the ADnCH1 register when the ADnSCM.ADnCE bit = 0 (conversion operation is stopped) (the same value can be written to the register when the ADnCE bit = 1 (conversion operation is enabled)).

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

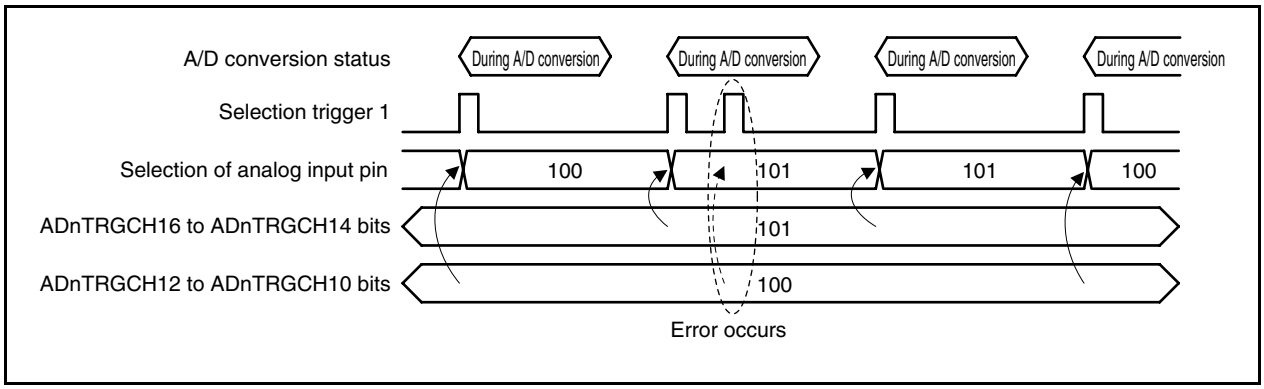
Setting the ADnCH1 register is enabled when a conversion operation is enabled (ADnSCM.ADnCE bit = 1) in the conversion channel specification mode or extension buffer mode. When the first selection trigger 1 is generated after the conversion operation is enabled (ADnCE bit = 1), the analog input pin specified by the ADnTRGCH12 to ADnTRGCH10 bits is selected and A/D conversion is executed. When the next selection trigger 1 is later generated, the analog input pin specified by the ADnTRGCH16 to ADnTRGCH14 bits is selected and A/D conversion is executed. After that, the analog input pins are alternately selected for output each time selection trigger 1 is generated.

Figure 12-6. ADnCH1 Register Operation



If an error occurs (when selection trigger 1 is generated during A/D conversion), the analog input pin specified by the ADnTRGCH12 to ADnTRGCH10 bits and the analog input pin specified by the ADnTRGCH16 to ADnTRGCH14 bits are alternately selected, but the selected analog input pin is not changed because A/D conversion is in progress.

Figure 12-7. ADnCH1 Register Operation In Case of Error



(8) A/D converter n channel specification register 2 (ADnCH2)

The ADnCH2 register is a register that specifies the analog input pin for selection trigger 2 in the extension buffer mode.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: AD0CH2 FFFFF233H, AD1CH2 FFFFF2B3H

	7	6	5	4	3	2	1	0
ADnCH2 (n = 0, 1)	0	ADn TRGCH26	ADn TRGCH25	ADn TRGCH24	0	ADn TRGCH22	ADn TRGCH21	ADn TRGCH20

ADnTRGCH26	ADnTRGCH25	ADnTRGCH24	Specification of analog input pin for selection trigger 2
0	0	0	ANIn0
0	0	1	ANIn1
0	1	0	ANIn2
0	1	1	ANIn3
1	0	0	ANIn4
1	0	1	ANIn5
1	1	0	ANI16
1	1	1	ANI17

ADnTRGCH22	ADnTRGCH21	ADnTRGCH20	Specification of analog input pin for selection trigger 2
0	0	0	ANIn0
0	0	1	ANIn1
0	1	0	ANIn2
0	1	1	ANIn3
1	0	0	ANIn4
1	0	1	ANIn5
1	1	0	ANI16
1	1	1	ANI17

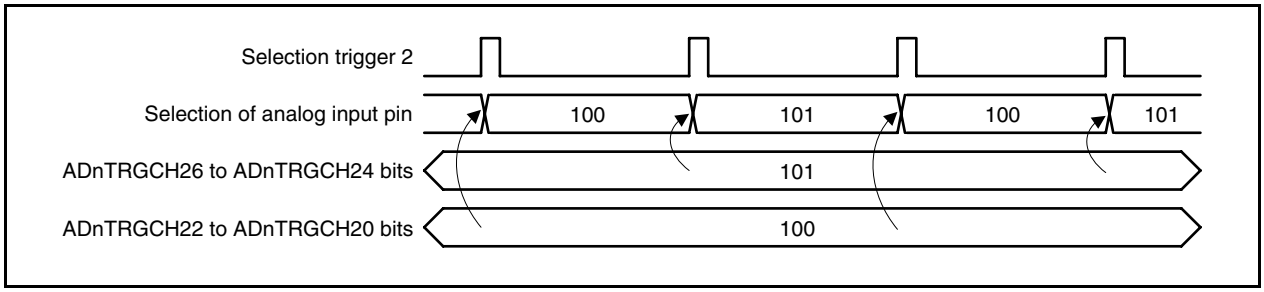
Cautions 1. Set the ADnCH2 register when the ADnSCM.ADnCE bit = 0 (conversion operation is stopped) (the same value can be written to the register when the ADnCE bit = 1 (conversion operation is enabled)).

2. The ADnCH2 register is valid only in the extension buffer mode; it is invalid in any other mode.

3. Be sure to set bits 3 and 7 to "0".

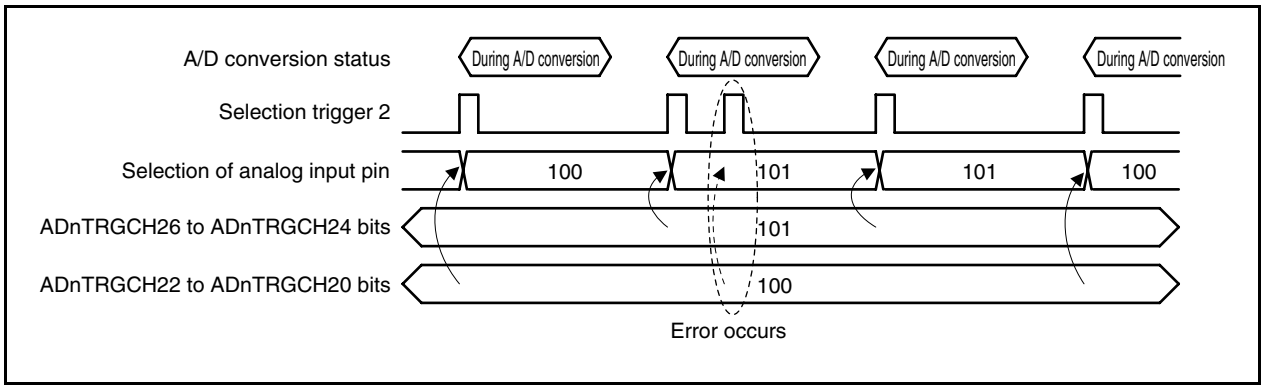
Setting the ADnCH2 register is enabled when a conversion operation is enabled (ADnSCM.ADnCE bit = 1) in the extension buffer mode. When the first selection trigger 2 is generated after the conversion operation is enabled (ADnCE bit = 1), the analog input pin specified by the ADnTRGCH22 to ADnTRGCH20 bits is selected and A/D conversion is executed. When the next selection trigger 2 is later generated, the analog input pin specified by the ADnTRGCH26 to ADnTRGCH24 bits is selected and A/D conversion is executed. After that, the analog input pins are alternately selected for output each time selection trigger 2 is generated.

Figure 12-8. ADnCH2 Register Operation



If an error occurs (when selection trigger 2 is generated during A/D conversion), the analog input pin specified by the ADnTRGCH22 to ADnTRGCH20 bits and the analog input pin specified by the ADnTRGCH26 to ADnTRGCH24 bits are alternately selected, but the selected analog input pin is not changed because A/D conversion is in progress.

Figure 12-9. ADnCH2 Register Operation In Case of Error



(9) A/Dn conversion result extension registers 0 to 4, 0H to 4H (ADnECR0 to ADnECR4, ADnECR0H to ADnECR4H)

The ADnECRa and ADnECRaH registers hold the result of A/D conversion in their higher 12 bits and indicate the status (information on the A/D conversion result of the analog input pin specified by the ADnCHx.ADnTRGCHx2 to ADnTRGCHx0 bits or ADnTRGCHx6 to ADnTRGCHx4 bits) of the A/D conversion result with the lower 1 bit in the extension buffer mode. Five of these registers are provided per circuit and two circuits are available. When A/D conversion is completed, the A/D conversion result is stored in A/Dn conversion result extension buffer register a. When selection load trigger 1 is later generated, the A/D conversion result is shifted from A/Dn conversion result extension buffer registers 0 to 2 to the higher 12 bits of the ADnECR0 to ADnECR2 registers and stored. Bits 1 to 3 are always 0 when read. When selection load trigger 2 is generated, the A/D conversion result is shifted from the A/Dn conversion result extension buffer registers 3 and 4 to the higher 12 bits of the ADnECR3 and ADnECR4 registers and stored. Bits 1 to 3 are always 0 when read.

The higher 8 bits of the A/D conversion result are read from the ADnECRaH register.

These registers are read-only in 16-bit or 8-bit units. To read the A/D conversion result in 16-bit units, specify the ADnECRa register. Specify the ADnECRaH register to read the higher 8 bits of the A/D conversion result.

Reset sets these registers to 0000H.

Remark While the result of A/D conversion is stored in the ADnECRa register, a read access to that register is held pending. The pending read access is executed when storing the A/D conversion result is completed. Similarly, storing the A/D conversion result in the ADnECRa register is held pending while a read access is made to that register. The pending A/D conversion result is stored in the register after the read access is completed.

After reset: 0000H R Address: AD0ECR0 FFFFF240H, AD0ECR1 FFFFF242H,
AD0ECR2 FFFFF244H, AD0ECR3 FFFFF246H,
AD0ECR4 FFFFF248H,
AD1ECR0 FFFFF2C0H, AD1ECR1 FFFFF2C2H,
AD1ECR2 FFFFF2C4H, AD1ECR3 FFFFF2C6H,
AD1ECR4 FFFFF2C8H

ADnECRa (n = 0, 1) (a = 0 to 4)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADnECRa11	ADnECRa10	ADnECRa9	ADnECRa8	ADnECRa7	ADnECRa6	ADnECRa5	ADnECRa4	ADnECRa3	ADnECRa2	ADnECRa1	ADnECRa0	0	0	0	ADnCHFLGa

ADnCHFLGa	Status of A/D conversion result (x = 1, 2)
0	A/D conversion result for analog input pin set by ADnCHx.ADnTRGCHx2 to ADnCHx.ADnTRGCHx0 bits
1	A/D conversion result for analog input pin set by ADnCHx.ADnTRGCHx6 to ADnCHx.ADnTRGCHx4 bits

After reset: 00H R Address: AD0ECR0H FFFFF241H, AD0ECR1H FFFFF243H,
AD0ECR2H FFFFF245H, AD0ECR3H FFFFF247H,
AD0ECR4H FFFFF249H,
AD1ECR0H FFFFF2C1H, AD1ECR1H FFFFF2C3H,
AD1ECR2H FFFFF2C5H, AD1ECR3H FFFFF2C7H,
AD1ECR4H FFFFF2C9H

ADnECRaH (n = 0, 1) (a = 0 to 4)

7	6	5	4	3	2	1	0
ADnECRa11	ADnECRa10	ADnECRa9	ADnECRa8	ADnECRa7	ADnECRa6	ADnECRa5	ADnECRa4

Caution The ADnECRa and ADnECRaH registers are valid only in the extension buffer mode; they are invalid in any other mode.

The correspondence between the analog input pins and the A/Dn conversion result extension registers is shown below.

Table 12-7. Correspondence Between Analog Input Pins and A/D Conversion Result Extension Registers

Analog Input Pin	A/Dn Conversion Result Register
Set with ADnCH1 register's ADnTRGCH12 to ADnTRGCH10, ADnTRGCH16 to ADnTRGCH14 bits	ADnECR0, ADnECR0H
	ADnECR1, ADnECR1H
	ADnECR2, ADnECR2H
Set with ADnCH2 register's ADnTRGCH22 to ADnTRGCH20, ADnTRGCH26 to ADnTRGCH24 bits	ADnECR3, ADnECR3H
	ADnECR4, ADnECR4H

Remark n = 0, 1

(10) A/D converter n flag register (ADnFLG)

The ADnFLG register indicates that an error has occurred when selection load trigger x is generated in the extension buffer mode (x = 1 or 2). The ADnTERR2 and ADnTERR1 flags can only be read and cleared when the conversion operation is stopped (ADnSCM.ADnCE bit = 0).

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

Reset sets this register to 00H.

After reset: 00H R Address: AD0FLG FFFFF254H, AD1FLG FFFFF2D4H

	7	6	5	4	3	2	1	0
ADnFLG (n = 0, 1)	0	0	0	0	0	0	ADn TERR2[Note]	ADn TERR1[Note]

ADnTERR2[Note]	Occurrence timing error flag of selection load trigger 2
0	Occurrence timing error of selection load trigger 2 has not occurred
1	Occurrence timing error of selection load trigger 2 has occurred

ADnTERR1[Note]	Occurrence timing error flag of selection load trigger 1
0	Occurrence timing error of selection load trigger 1 has not occurred
1	Occurrence timing error of selection load trigger 1 has occurred

Note The ADnTERR2 and ADnTERR1 flags are valid only in the extension buffer mode; they are fixed to 0 in any other mode.

(11) A/D converter n flag buffer register (ADnFLGB)

The ADnFLGB register indicates that an error has occurred when selection trigger x is generated in the extension buffer mode (x = 1 or 2). The ADnTERRB2 and ADnTERRB1 flags can only be read and cleared when the conversion operation is stopped (ADnSCM.ADnCE bit = 0).

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

Reset sets this register to 00H.

After reset: 00H R Address: AD0FLGB FFFFF255H, AD1FLGB FFFFF2D5H

	7	6	5	4	3	2	1	0
ADnFLGB (n = 0, 1)	0	0	0	0	0	0	ADn TERRB2[Note]	ADn TERRB1[Note]

ADnTERRB2[Note]	Occurrence timing error flag of selection trigger 2
0	Occurrence timing error of selection trigger 2 has not occurred
1	Occurrence timing error of selection trigger 2 has occurred

ADnTERRB1[Note]	Occurrence timing error flag of selection trigger 1
0	Occurrence timing error of selection trigger 1 has not occurred
1	Occurrence timing error of selection trigger 1 has occurred

Note The ADnTERRB2 and ADnTERRB1 flags are valid only in the extension buffer mode; they are fixed to 0 in any other mode.

(12) A/D LDTRG1 input select register (ADLTS1)

The ADLTS1 register is a register that specifies the input signal for selection load trigger (LDTRG1) in the extension buffer mode.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF2F8H

	7	6	5	4	3	2	1	0
ADLTS1	0	0	0	0	0	0	0	ADLTS10

ADLTS10	Specification of input signal for LDTRG1
0	TABTIOV0 signal
1	TABTIOV1 signal

Note The ADLTS1 register is valid only in the extension buffer mode; it is invalid in any other mode.

(13) A/D LDTRG2 input select register (ADLTS2)

The ADLTS2 register is a register that specifies the input signal for selection load trigger (LDTRG2) in the extension buffer mode.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF2FAH

	7	6	5	4	3	2	1	0
ADLTS2	0	0	0	0	0	0	0	ADLTS20

ADLTS20	Specification of input signal for LDTRG2
0	TABTICC00 signal
1	TABTICC10 signal

Note The ADLTS2 register is valid only in the extension buffer mode; it is invalid in any other mode.

(14) Operational amplifier n control register 0 (OPnCTL0)

The OPnCTL0 register is used to control the operation of an operational amplifier that amplifies the input level, and specify its gain.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF260H

	7	6	5	4	3	2	1	0
OP0CTL0	0	0	0	OP0EN	OP0GA3	OP0GA2	OP0GA1	OP0GA0

After reset: 00H R/W Address: FFFFF2E0H

	7	6	5	4	3	2	1	0
OP1CTL0	0	OP12EN	OP11EN	OP10EN	OP1GA3	OP1GA2	OP1GA1	OP1GA0

OP12EN	Operation control of operational amplifier 2
0	Operation disabled (not used)
1	Operation enabled (used)

OP11EN	Operation control of operational amplifier 1
0	Operation disabled (not used)
1	Operation enabled (used)

OP0EN	Operation control of operational amplifier 0
0	Operation disabled (not used)
1	Operation enabled (used)

OP10EN	Operation control of operational amplifier 0
0	Operation disabled (not used)
1	Operation enabled (used)

OPnGA3	OPnGA2	OPnGA1	OPnGA0	Gain specification of operational amplifier
0	0	0	0	×2.500
0	0	0	1	×2.667
0	0	1	0	×2.857
0	0	1	1	×3.077
0	1	0	0	×3.333
0	1	0	1	×3.636
0	1	1	0	×4.000
0	1	1	1	×4.444
1	0	0	0	×5.000
1	0	0	1	×5.714
1	0	1	0	×6.667
1	0	1	1	×8.000
1	1	0	0	×10.00
Others				Setting prohibited

Cautions 1. Be sure to set bits 5 to 7 of the OP0CTL0 register and bit 7 of the OP1CTL0 register to "0".

2. After enabling the operation of the operational amplifier, stabilization time of 10 μs is required.

(15) Comparator n control register 0 (CMPnCTL0)

The CMPnCTL0 register is a register that controls the operation of the overvoltage detection comparator.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF261H

	7	6	5	4	3	2	1	0
CMP0CTL0	0	0	0	CMP0FEN	0	0	0	CMP0LEN

After reset: 00H R/W Address: FFFFF2E1H

	7	6	5	4	3	2	1	0
CMP1CTL0	0	CMP12FEN	CMP11FEN	CMP10FEN	0	CMP12LEN	CMP11LEN	CMP10LEN

CMP12FEN	Operation control of comparator 2 (full range)
0	Operation disabled (not used)
1	Operation enabled (used)

CMP11FEN	Operation control of comparator 1 (full range)
0	Operation disabled (not used)
1	Operation enabled (used)

CMP0FEN	Operation control of comparator 0 (full range)
0	Operation disabled (not used)
1	Operation enabled (used)

CMP10FEN	Operation control of comparator 0 (full range)
0	Operation disabled (not used)
1	Operation enabled (used)

CMP12LEN	Operation control of comparator 2 (low range)
0	Operation disabled (not used)
1	Operation enabled (used)

CMP11LEN	Operation control of comparator 1 (low range)
0	Operation disabled (not used)
1	Operation enabled (used)

CMP0LEN	Operation control of comparator 0 (low range)
0	Operation disabled (not used)
1	Operation enabled (used)

CMP10LEN	Operation control of comparator 0 (low range)
0	Operation disabled (not used)
1	Operation enabled (used)

(2/2)

Cautions 1. Be sure to set bits 1 to 3, 5 to 7 of the CMP0CTL0 register and bits 3, 7 of the CMP1CTL0 register to "0".

2. After enabling the operation of the comparator, stabilization time of TBD μs is required.

3. The input voltage range of the comparator is as follows, regardless of whether it is amplified by the operational amplifier or not.

Input voltage range of CREF0L and CREF1L pins (low-range side): $0.02AV_{DD}$ + 0.1 to $0.5AV_{DD}$ – 0.1 V

Input voltage range of CREF0F and CREF1F pins (full-range side): $0.02AV_{DD}$ + 0.1 to $0.92AV_{DD}$ – 0.1 V

For details, see CHAPTER 28 ELECTRICAL SPECIFICATIONS (TARGET).

(16) Comparator n control register 1 (CMPnCTL1)

The CMPnCTL1 register is a register that monitors the output of the overvoltage detection comparator.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R Address: FFFFF262H

	7	6	5	4	3	2	1	0
CMP0CTL1	0	0	0	CMP0FOUT	0	0	0	CMP0LOUT

After reset: 00H R Address: FFFFF2E2H

	7	6	5	4	3	2	1	0
CMP1CTL1	0	CMP12FOUT	CMP11FOUT	CMP10FOUT	0	CMP12LOUT	CMP11LOUT	CMP10LOUT

CMP12FOUT	Output level status of comparator 2 (full range)
0	Comparator output = 0 (without overvoltage detection)
1	Comparator output = 1 (with overvoltage detection)

CMP11FOUT	Output level status of comparator 1 (full range)
0	Comparator output = 0 (without overvoltage detection)
1	Comparator output = 1 (with overvoltage detection)

CMP0FOUT	Output level status of comparator 0 (full range)
0	Comparator output = 0 (without overvoltage detection)
1	Comparator output = 1 (with overvoltage detection)

CMP10FOUT	Output level status of comparator 0 (full range)
0	Comparator output = 0 (without overvoltage detection)
1	Comparator output = 1 (with overvoltage detection)

CMP12LOUT	Output level status of comparator 2 (low range)
0	Comparator output = 0 (without overvoltage detection)
1	Comparator output = 1 (with overvoltage detection)

CMP11LOUT	Output level status of comparator 1 (low range)
0	Comparator output = 0 (without overvoltage detection)
1	Comparator output = 1 (with overvoltage detection)

CMP0LOUT	Output level status of comparator 0 (low range)
0	Comparator output = 0 (without overvoltage detection)
1	Comparator output = 1 (with overvoltage detection)

CMP10LOUT	Output level status of comparator 0 (low range)
0	Comparator output = 0 (without overvoltage detection)
1	Comparator output = 1 (with overvoltage detection)

(2/2)

Cautions 1. The CMP12FOUT, CMP11FOUT, CMP0FOUT, CMP10FOUT, CMP12LOUT, CMP11LOUT, CMP0LOUT, and CMP10LOUT bits are set to 0 when the input voltage falls to a level at which an overvoltage is not detected.

2. Be sure to set bits 1 to 3, 5 to 7 of the CMP0CTL1 register and bits 3, 7 of the CMP1CTL1 register to "0".

(17) Comparator n control register 2 (CMPnCTL2)

The CMPnCTL2 register is a register that specifies the compare signal of the overvoltage detection comparator.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF263H

	7	6	5	4	3	2	1	0
CMP0CTL2	0	0	0	0	0	0	0	CMP0SEL

After reset: 00H R/W Address: FFFFF2E3H

	7	6	5	4	3	2	1	0
CMP1CTL2	0	0	0	0	0	CMP12SEL	CMP11SEL	CMP10SEL

CMP12SEL	Specification of compare signal of comparator 2
0	Before operational amplifier 2 amplification
1	After operational amplifier 2 amplification

CMP11SEL	Specification of compare signal of comparator 1
0	Before operational amplifier 1 amplification
1	After operational amplifier 1 amplification

CMP0SEL	Specification of compare signal of comparator 0
0	Before operational amplifier 0 amplification
1	After operational amplifier 0 amplification

CMP10SEL	Specification of compare signal of comparator 0
0	Before operational amplifier 0 amplification
1	After operational amplifier 0 amplification

Caution Be sure to set bits 1 to 7 of the CMP0CTL2 register and bits 3 to 7 of the CMP1CTL2 register to “0”.

(18) Comparator n control register 3 (CMPnCTL3)

The CMPnCTL3 register is a register that specifies the output generation of the overvoltage detection comparator and controls the edge detection.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF264H

	7	6	5	4	3	2	1	0
CMP0CTL3	CMP0FDS	0	0	CMP0FDE	CMP0LDS	0	0	CMP0LDE

After reset: 00H R/W Address: FFFFF2E4H

	7	6	5	4	3	2	1	0
CMP1CTL3	CMP1FDS	CMP12FDE	CMP11FDE	CMP10FDE	CMP1LDS	CMP12LDE	CMP11LDE	CMP10LDE

CMPnFDS	Specification of output generation of comparator (full range)
0	Logical product (AND) detection
1	Logical sum (OR) detection

CMP12FDE	Edge detection control of comparator 2 (full range)
0	Edge detection disabled
1	Edge detection enabled

CMP11FDE	Edge detection control of comparator 1 (full range)
0	Edge detection disabled
1	Edge detection enabled

CMP0FDE	Edge detection control of comparator 0 (full range)
0	Edge detection disabled
1	Edge detection enabled

CMP10FDE	Edge detection control of comparator 0 (full range)
0	Edge detection disabled
1	Edge detection enabled

CMPnLDS	Specification of output generation of comparator (low range)
0	Logical product (AND) detection
1	Logical sum (OR) detection

CMP12LDE	Edge detection control of comparator 2 (low range)
0	Edge detection disabled
1	Edge detection enabled

(2/2)

CMP11LDE	Edge detection control of comparator 1 (low range)
0	Edge detection disabled
1	Edge detection enabled

CMP0LDE	Edge detection control of comparator 0 (low range)
0	Edge detection disabled
1	Edge detection enabled

CMP10LDE	Edge detection control of comparator 0 (low range)
0	Edge detection disabled
1	Edge detection enabled

(19) A/D converter n clock select register (ADnOCKS)

The ADnOCKS register is a register that selects the clock (f_{AD01}) to be input to the A/D converter n.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: AD0OCKS FFFFF270H, AD1OCKS FFFFF274H

	7	6	5	4	3	2	1	0
ADnOCKS (n = 0, 1)	0	0	0	ADnOCKSEN	0	0	ADnOCKS1	ADnOCKS0

ADnOCKSEN	Clock operation control
0	Stop operation clock supply of A/D converter n
1	Enable operation clock supply of A/D converter n

ADnOCKS1	ADnOCKS0	Input clock selection of A/D converter n (f_{AD01})
0	0	$f_{XX}/2$
0	1	$f_{XX}/3$
1	0	$f_{XX}/4$ (when f_{XX} = 64 MHz)
1	1	Setting prohibited

Cautions 1. Set f_{AD01} to 16 MHz or lower.

2. When A/D converter n is used, be sure to set the ADnOCKS register and set the ADnSCM.ADnPS bit to 1, as well as to read the A/D conversion result register.

3. Be sure to set bits 2, 3, and 5 to 7 to "0".

(20) Comparator output digital noise elimination register nL, nF (CMPNFCnL, CMPNFCnF)

The CMPNFCnL and CMPNFCnF registers are control the digital noise elimination of the overvoltage detection comparator output.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: CMPNFC0L FFFFF278H, CMPNFC1L FFFFF27CH

CMPNFCnL (n = 0, 1)

7	6	5	4	3	2	1	0
CMPnNFEN	0	0	0	0	CMPnNFC2	CMPnNFC1	CMPnNFC0

After reset: 00H R/W Address: CMPNFC0F FFFFF27AH, CMPNFC1F FFFFF27EH

CMPNFCnF (n = 0, 1)

7	6	5	4	3	2	1	0
CMPnNFEN	0	0	0	0	CMPnNFC2	CMPnNFC1	CMPnNFC0

CMPnNFEN	Setting of digital noise elimination
0	Does not perform digital noise elimination (through)
1	Performs digital noise elimination

CMPnNFC2	CMPnNFC1	CMPnNFC0	Sampling clock selection
0	0	0	f_{XX}/32
0	0	1	f_{XX}/64
0	1	0	f_{XX}/128
0	1	1	f_{XX}/256
1	0	0	f_{XX}/1024
1	0	1	f_{XX}/4096
Others			Setting prohibited

Cautions 1. The basic operation of digital noise elimination is the same as in Figure 4-4 Example of Noise Elimination Timing. The differences are that sampling is performed two times and that the sampling clock is different.

2. Be sure to set bits 3 to 6 to "0".

(21) A/D trigger rising edge, falling edge specification registers (ADTR, ADTF)

The ADTR and ADTF registers are registers that specify the trigger mode of the ADTRG0 and ADTRG1 pins and can specify the valid edge independently for each pin (rising edge, falling edge, or both rising and falling edges).

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

Reset sets these registers to 00H.

Caution When the function is changed from the external trigger input of the A/D converter n (alternate function) to the port mode, an edge may be detected. Therefore, be sure to set the ADTFn and ADTRn bits to 00, and then set the port mode.

After reset: 00H R/W Address: FFFFF2F2H

	7	6	5	4	3	2	1	0
ADTR	0	0	0	0	0	0	ADTR1	ADTR0

After reset: 00H R/W Address: FFFFF2F0H

	7	6	5	4	3	2	1	0
ADTF	0	0	0	0	0	0	ADTF1	ADTF0

Remark For the valid edge specification, see **Table 12-8**.

Table 12-8. Valid Edge Specification of ADTRG0 and ADTRG1 Pins

ADTFn	ADTRn	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution When not using these pins as the ADTRGn pins, be sure to set the ADTFn and ADTRn bits to 00.

Remark n = 0, 1

(22) Comparator output interrupt rising edge, falling edge specification registers (CMPOR, CMPOF)

The CMPOR and CMPOF registers are registers that specify the trigger mode of the INTCMP0L, INTCMP0F, INTCMP1L, and INTCMP1F signals and can specify the valid edge independently for each interrupt request signal (rising edge, falling edge, or both rising and falling edges).

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

Reset sets these registers to 00H.

After reset: 00H R/W Address: FFFFF2F6H

	7	6	5	4	3	2	1	0
CMPOR	0	0	0	0	CMPOR1F	CMPOR1L	CMPOR0F	CMPOR0L

After reset: 00H R/W Address: FFFFF2F4H

	7	6	5	4	3	2	1	0
CMPOF	0	0	0	0	CMPOF1F	CMPOF1L	CMPOF0F	CMPOF0L

Remark For the valid edge specification, see **Tables 12-9** and **12-10**.

Table 12-9. Valid Edge Specification of INTCMP0F and INTCMP1F Signals

CMPOFnF	CMPORnF	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Remark n = 0, 1

Table 12-10. Valid Edge Specification of INTCMP0L and INTCMP1L Signals

CMPOFnL	CMPORnL	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Remark n = 0, 1

12.4 Operation

Cautions 1. A/D converters 0 and 1 are capable of simultaneous sampling of two circuits.

2. For details of operation setting, see 12.3 (1) A/D converter n scan mode register (ADnSCM).

12.4.1 Basic operation

A/D conversion is executed by the following procedure.

(1) Select an input clock (f_{AD01}) by using the ADnOCKS register and set the ADnOCKSEN bit to 1 (enable supply of the operating clock to A/D converter n).

(2) Set ADnSCM.ADnPS bit = 1.

(3) Wait for 1 μs or more after <2>.

(4) Select an analog input pin and operation mode, by using the ADnSCM[Note], ADnCTC, ADnCHEN, ADnCTL0, ADnTSEL, ADnCH1, ADnCH2, ADLTS1, and ADLTS2 registers (n = 0, 1). Number of A/D conversion clocks and A/D conversion time are determined by the specification of the ADnCTC.ADnFR3 to ADnCTC.ADnFR0 bits.

Note Be sure to set bit 1 of the ADnSCM register to “1”.
This setting can be performed at the same time as other ADnSCM register bits.

(5) In the A/D trigger mode and the A/D trigger polling mode, setting the ADnSCM.ADnCE bit to 1 starts A/D conversion (n = 0, 1). If the ADnCE bit is set to 1 in the hardware trigger mode, conversion channel specification mode, and extension buffer mode, the A/D converter enters the trigger wait status.

(6) When A/D conversion is started, the voltage input to the selected analog input channel is sampled by the sample & hold circuit. When the operational amplifier for input level amplification is used, the gain specified by the OPnCTL0.OPnGA3 to OPnCTL0.OPnGA0 bits $\times$ the input voltage is sampled.

(7) When sampling has been performed for a specific time, the sample & hold circuit enters the hold status, and holds the input analog voltage until A/D conversion ends.

(8) Set bit 11 of the successive approximation register (SAR). The tap selector changes the level of the voltage tap of the array to the reference voltage ($1/2AV_{REFPn}$).

(9) The voltage generated by the voltage tap of the array is compared with the analog input voltage by a comparator. If the analog input voltage is found to be greater than the reference voltage ($1/2AV_{REFPn}$) as a result of comparison, the most significant bit (MSB) of the successive approximation register (SAR) remains set. If the analog input voltage is less than the reference voltage ($1/2AV_{REFPn}$), the MSB of the SAR is reset.

(10) Next, bit 10 of the successive approximation register (SAR) is automatically set, and the next comparison is started. The voltage tap of the array is selected according to the value of bit 11, to which the result has been already set.

Bit 11 = 0: ($1/4AV_{REFPn}$)
Bit 11 = 1: ($3/4AV_{REFPn}$)

The voltage tap of the array and the analog input voltage are compared and bit 10 of the SAR is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of array: Bit 10 = 1
Analog input voltage ≤ Voltage tap of array: Bit 10 = 0

Comparison is continued like this to bit 0 of the SAR.

(11) When comparison of 12 bits has been completed, the valid digital value result remains in the successive approximation register (SAR). This value is transferred to A/Dn conversion result register m (ADnCRm) and the conversion result is stored in this register in the A/D trigger mode, A/D trigger polling mode, hardware trigger mode, and conversion channel specification mode (n = 0, 1, m = 0 to 15). The valid digital value is stored in the A/Dn conversion result extension buffer register a in the extension buffer mode, and is shifted to A/Dn conversion result extension register a when selection load trigger x is generated and stored (x = 1, 2, a = 0 to 4). When A/D conversion has ended the specified number of times, an A/Dn conversion end interrupt request signal (INTADn) is generated.

12.4.2 Input voltage and conversion result

The relationship between the analog voltage input to the analog input pin (ANInk) and the A/D conversion result (of A/Dn conversion result register m (ADnCRm) or A/Dn conversion result extension register a (ADnECRa)) is as follows:

$$ADCR = INT\left(\frac{V_{IN}}{AV_{REFP}} \times 4{,}096 + 0.5\right)$$

or,

$$(ADCR - 0.5) \times \frac{AV_{REFP}}{4{,}096} \leq V_{IN} < (ADCR + 0.5) \times \frac{AV_{REFP}}{4{,}096}$$

INT( ): Function that returns the integer of the value in ( )

V_{IN}: Analog input voltage

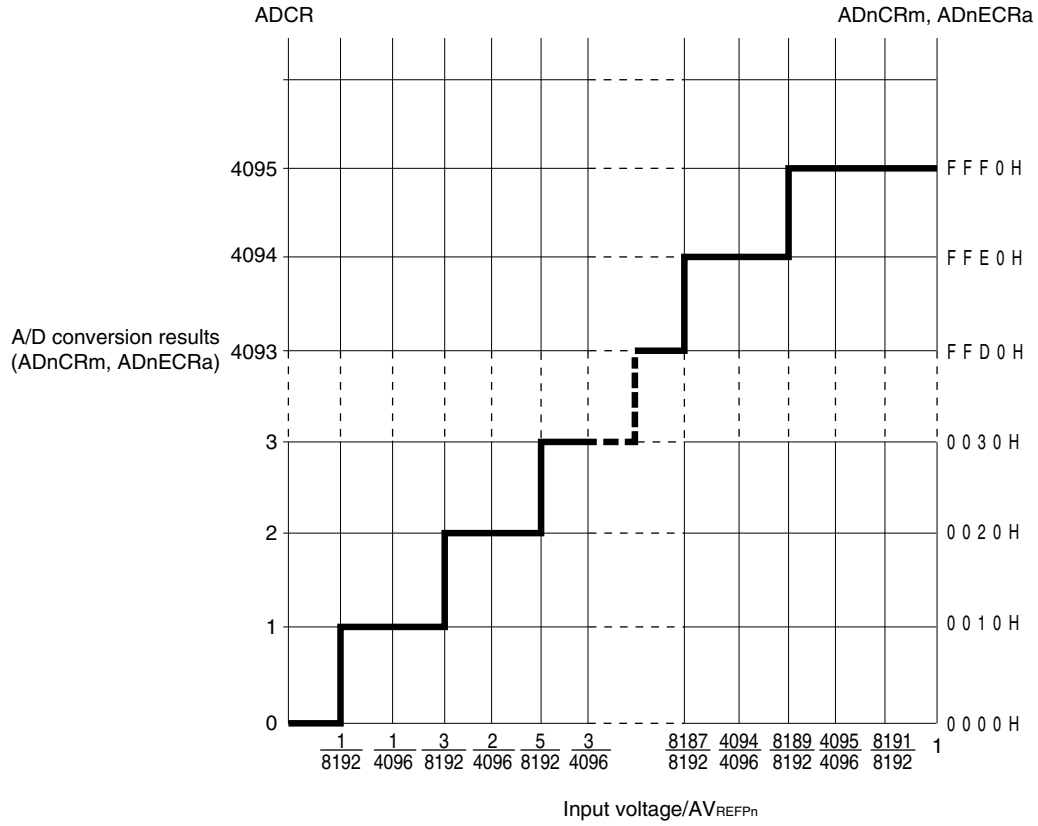
AV_{REFP}: AV_{REFPn} pin voltage

ADCR: Value of A/Dn conversion result register m (ADnCRm) or A/Dn conversion result extension register a (ADnECRa)

The relationship between the analog input voltage and the A/D conversion result is shown below.

Remark A/D converter 0: n = 0, m = 0 to 15, k = 0 to 5, a = 0 to 4
A/D converter 1: n = 1, m = 0 to 15, k = 0 to 7, a = 0 to 4

Figure 12-10. Relationship Between Analog Input Voltage and A/D Conversion Results



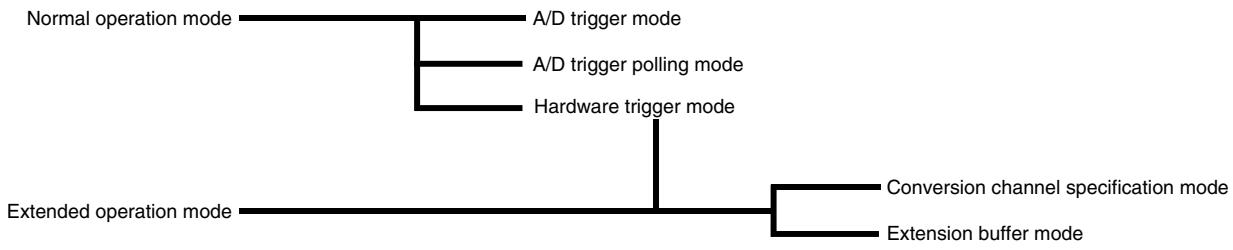
Remark A/D converter 0: n = 0, m = 0 to 15, k = 0 to 5, a = 0 to 4
A/D converter 1: n = 1, m = 0 to 15, k = 0 to 7, a = 0 to 4

12.4.3 Operation mode

Various conversion operations can be specified for the A/D converters 0 and 1 by specifying the operation mode. The operation mode is set by the ADnSCM, ADnCTC, ADnCHEN, ADnCTL0, ADnTSEL, ADnCH1, ADnCH2, ADLTS1, ADLTS2, and ADnOCKS registers.

The following shows the relationship between the operation modes.

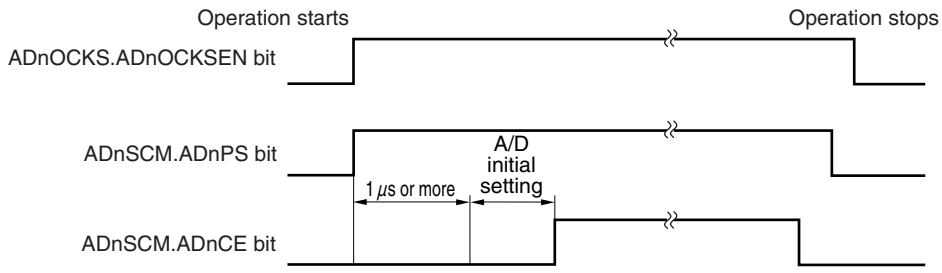
Remark n = 0, 1



Caution Be sure to set the hardware trigger mode when the conversion channel specification mode or extension buffer mode is used.

12.4.4 Operation setting

Start or stop the operation of A/D converters 0 and 1 in the following procedure.



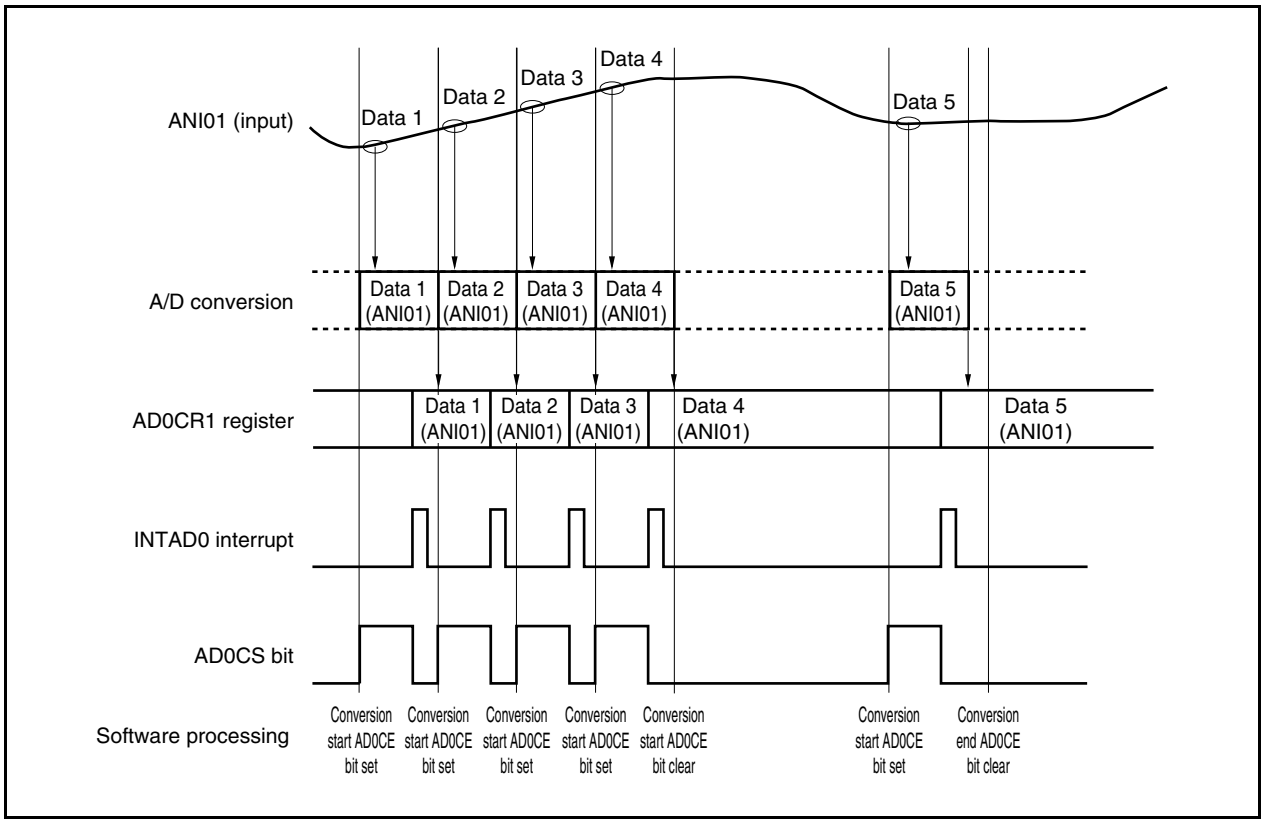
12.4.5 Operation of 1-channel conversion

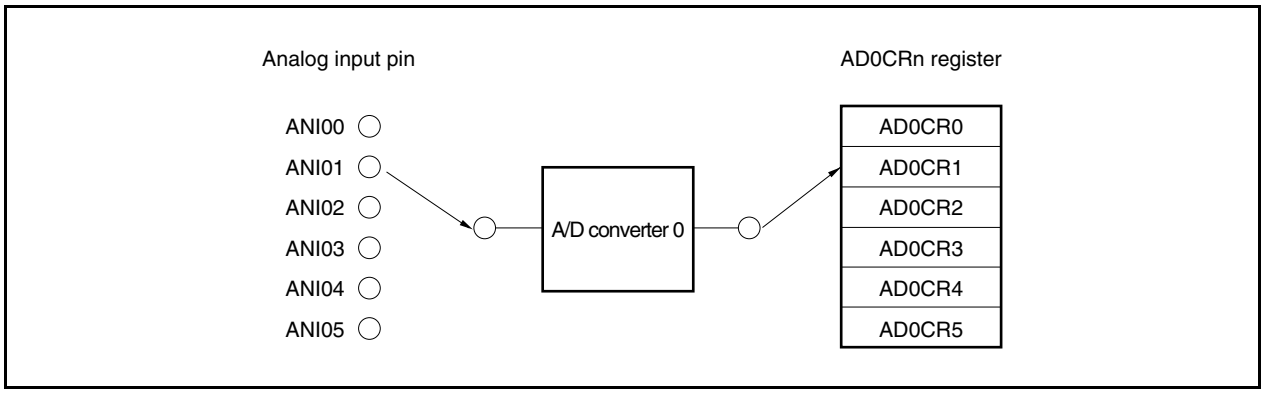
The signal of one analog input pin (ANInk) specified by the ADnCHEN register is converted. The result of conversion is stored in the ADnCRk register corresponding to the ANInk pin. The ANInk pin and ADnCRk register correspond to each other on a one-to-one basis, and an A/Dn conversion end interrupt request signal (INTADn) is generated each time conversion has been completed.

After completion of A/D conversion, the conversion operation is stopped in the A/D trigger mode or A/D trigger polling mode. In the hardware trigger mode, the A/D converter waits for a trigger.

Remark A/D converter 0: n = 0, k = 0 to 5
A/D converter 1: n = 1, k = 0 to 7

Figure 12-11. Operation of 1-Channel Conversion (in A/D Trigger Mode): A/D Converter 0





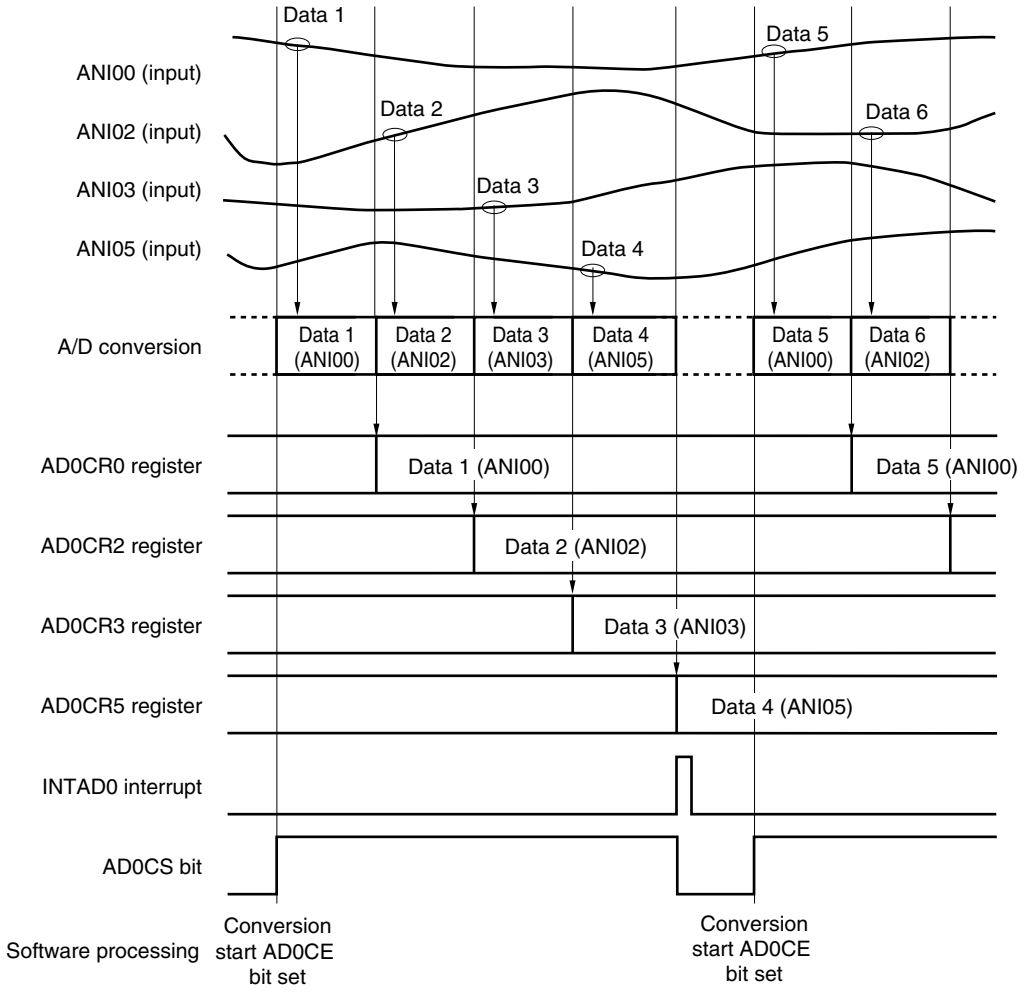
12.4.6 Operation of multiple channel conversion

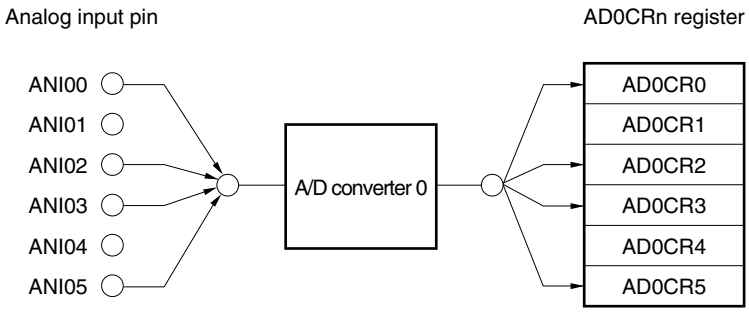
The signals of two or more analog input pins (ANInk) specified by the ADnCHEN register are converted. The signals are sequentially converted starting from the pin with the lowest number (in the example in Figure 12-12, ANI00 → ANI02 → ANI03 → ANI05). An analog input pin that is not specified is skipped. The result of conversion is stored in the ADnCRk register corresponding to the ANInk pin. The ANInk pin and ADnCRk register correspond to each other on a one-to-one basis. When conversion of the signal of the specified analog input pins is completed, an A/Dn conversion end interrupt request signal (INTADn) is generated.

After completion of A/D conversion, the conversion operation is stopped in the A/D trigger mode or A/D trigger polling mode. In the hardware trigger mode, the A/D converter waits for a trigger.

Remark A/D converter 0: n = 0, k = 0 to 5
A/D converter 1: n = 1, k = 0 to 7

Figure 12-12. Operation of Multiple Channel Conversion (in A/D Trigger Mode): A/D Converter 0





12.4.7 A/D trigger mode (normal operation mode)

A/D conversion is started when the ADnSCM.ADnCE bit is set to 1.

When conversion is started, the ADnSCM.ADnCS bit is set to 1 (conversion is in progress).

If the ADnSCM register is written during A/D conversion, the conversion is stopped and started again from the beginning.

(1) Operation of 1-channel conversion

The signal of one analog input pin (ANInk) is converted once and the result is stored in one ADnCRk register. The ANInk pin and ADnCRk register correspond to each other on a one-to-one basis.

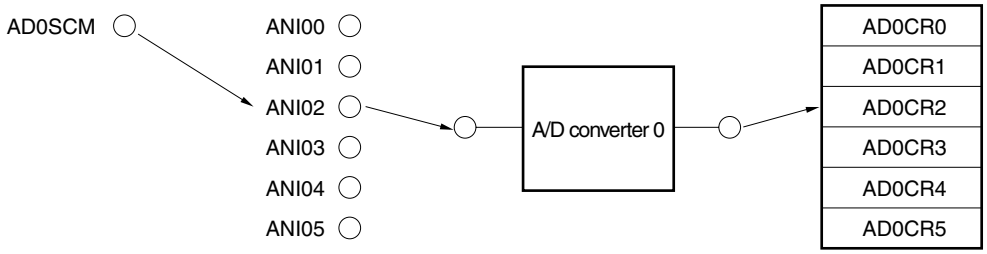
Each time conversion has been completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. After A/D conversion is completed, The A/D converter stops conversion operation with the ADnSCM.ADnCE bit remaining set to 1. The A/D conversion can be restarted by setting the ADnCE bit to 1.

This operation is suitable for an application where the result of A/D conversion should be read each time conversion has been completed once.

Analog Input Pin	A/D Conversion Result Register
ANInk	ADnCRk

Remark A/D converter 0: n = 0, k = 0 to 5
A/D converter 1: n = 1, k = 0 to 7

Figure 12-13. Example of 1-Channel Conversion Operation (A/D Trigger Mode): A/D Converter 0



(1) AD0CE bit = 1 (enable)
(2) Signal of ANI02 pin is A/D converted
(3) Conversion result is stored in AD0CR2 register
(4) AD0SCM.AD0CS bit = 0
(5) INTAD0 interrupt request signal is generated

Remark This is an operation example when the AD0SCM.AD0PLM, AD0TRG1, and AD0TRG0 bits = 000, AD0CTL0.AD0MD1 and AD0CTL0.AD0MD0 bits = 00, and AD0CHEN register = 0004H.

(2) Operation of multiple channel conversion

The signals of two or more analog input pins specified by the ADnCHEN register are converted sequentially starting from the pin with the lowest number. The result of conversion is stored in the ADnCRk register corresponding to the analog input pin.

When conversion of the signals of all the specified analog input pins is completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. After A/D conversion is completed, the A/D converter stops conversion operation with the ADnSCM.ADnCE bit remaining set to 1. The A/D conversion can be restarted by setting the ADnCE bit to 1.

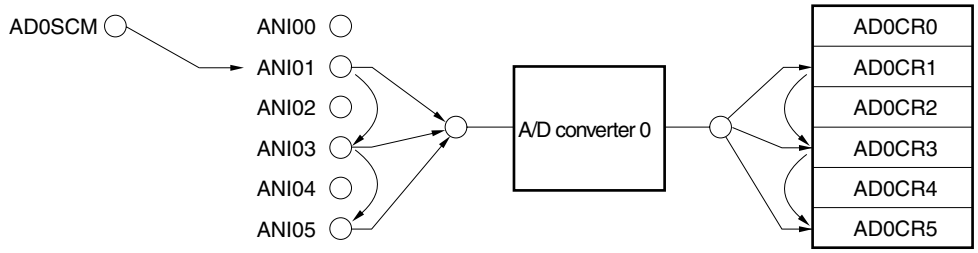
This operation is suitable for an application where two or more analog input signals should be monitored.

Analog Input Pin	A/D Conversion Result Register
ANInk[Note]	ADnCRk
|	|
ANInk[Note]	ADnCRk

Note Two or more can be specified by the ADnCHEN register.
However, A/D conversion is sequentially executed starting from the pin with the lowest number.

Remark A/D converter 0: n = 0, k = 0 to 5
A/D converter 1: n = 1, k = 0 to 7

Figure 12-14. Example of Multiple Channel Conversion Operation (A/D Trigger Mode): A/D Converter 0



(1) AD0CE bit = 1 (enable)
(2) Signal of ANI01 pin is A/D-converted
(3) Conversion result is stored in AD0CR1 register
(4) Signal of ANI03 pin is A/D-converted
(5) Conversion result is stored in AD0CR3 register
(6) Signal of ANI05 pin is A/D-converted
(7) Conversion result is stored in AD0CR5 register
(8) AD0SCM.AD0CS bit = 0
(9) INTAD0 interrupt request signal is generated

Remark This is an operation example when the AD0SCM.AD0PLM, AD0TRG1, and AD0TRG0 bits = 000, AD0CTL0.AD0MD1 and AD0CTL0.AD0MD0 bits = 00, and AD0CHEN register = 002AH.

12.4.8 A/D trigger polling mode (normal operation mode)

A/D conversion is started when the ADnSCM.ADnCE bit is set to 1.

When conversion is started, the ADnSCM.ADnCS bit is set to 1 (conversion is in progress).

In the A/D trigger polling mode, it is not necessary to write 1 to the ADnCE bit to restart A/D conversion after the A/Dn conversion end interrupt request signal (INTADn) is generated.

If the ADnSCM register is written during A/D conversion, the conversion is stopped and started again from the beginning.

(1) Operation of 1-channel conversion

The signal of one analog input pin (ANInk) is converted once and the result is stored one ADnCRk register. The ANInk pin and ADnCRk register correspond to each other on a one-to-one basis.

Each time conversion has been completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. A/D conversion is repeated until the ADnSCM.ADnCE bit is set to 0. The conversion operation is stopped when the ADnCE bit is cleared to 0.

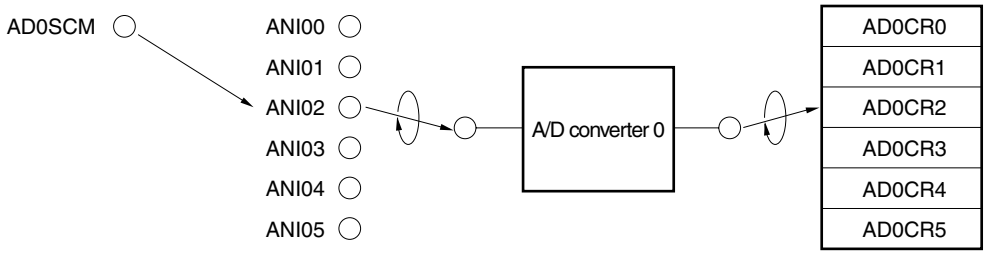
It is not necessary to set the ADnCE bit to restart the conversion operation in the A/D trigger polling mode.

This operation is suitable for an application where the A/D conversion value is always read.

Analog Input Pin	A/D Conversion Result Register
ANInk	ADnCRk

Remark A/D converter 0: n = 0, k = 0 to 5
A/D converter 1: n = 1, k = 0 to 7

Figure 12-15. Example of 1-Channel Conversion Operation (A/D Trigger Polling Mode): A/D Converter 0



(1) AD0CE bit = 1 (enable)
(2) Signal of ANI02 pin is A/D-converted
(3) Conversion result is stored in AD0CR2 register
(4) AD0SCM.AD0CS bit = 0
(5) INTAD0 interrupt request signal is generated
(6) Return to (2)
(7) Set AD0CE bit to 0 to end (stop)

Remark This is an operation example when the AD0SCM.AD0PLM, AD0TRG1, and AD0TRG0 bits = 100, AD0CTL0.AD0MD1 and AD0CTL0.AD0MD0 bits = 00, and AD0CHEN register = 0004H.

(2) Operation of multiple channel conversion

The signals of two or more analog input pins specified by the ADnCHEN register are converted sequentially starting from the pin with the lowest number. The result of conversion is stored in the ADnCRk register corresponding to the analog input pin.

When conversion of the signals of all the specified analog input pins is completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. A/D conversion is repeated until the ADnSCM.ADnCE bit is set to 0. The conversion operation is stopped when the ADnCE bit is cleared to 0.

It is not necessary to set the ADnCE bit to restart the conversion operation in the A/D trigger polling mode.

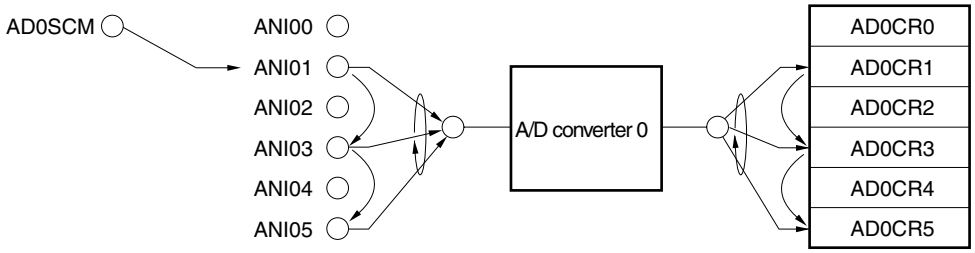
This operation is suitable for an application where the A/D conversion value is always read.

Analog Input Pin	A/D Conversion Result Register
ANIn*k*[Note]	ADnCRk
ANIn*k*[Note]	ADnCRk

Note Two or more can be specified by the ADnCHEN register.
However, A/D conversion is sequentially executed starting from the pin with the lowest number.

Remark A/D converter 0: n = 0, x = 0 to 5
A/D converter 1: n = 1, x = 0 to 7

Figure 12-16. Example of Multiple Channel Conversion Operation (A/D Trigger Polling Mode): A/D Converter 0



(1) AD0CE bit = 1 (enable)
(2) Signal of ANI01 pin is A/D-converted
(3) Conversion result is stored in AD0CR1 register
(4) Signal of ANI03 pin is A/D-converted
(5) Conversion result is stored in AD0CR3 register
(6) Signal of ANI05 pin is A/D-converted
(7) Conversion result is stored in AD0CR5 register
(8) AD0SCM.AD0CS bit = 0
(9) INTAD0 interrupt request signal is generated
(10) Return to (2)
(11) Set AD0CE bit to 0 to end (stop)

Remark This is an operation example when the AD0SCM.AD0PLM, AD0TRG1, and AD0TRG0 bits = 100, AD0CTL0.AD0MD1 and AD0CTL0.AD0MD0 bits = 00, and AD0CHEN register = 002AH.

12.4.9 Hardware trigger mode (normal operation mode)

The A/D converter waits for a trigger when the ADnSCM.ADnCE bit is set to 1, and starts A/D conversion when a trigger specified by the ADnTSEL.ADnTRGSEL11 and ADnTSEL.ADnTRGSEL10 bits is generated.

When conversion is started, the ADnSCM.ADnCS bit is set to 1 (conversion is in progress).

If the ADnSCM register is written during A/D conversion, the conversion is stopped and becomes trigger wait status again.

(1) Operation of 1-channel conversion

The signal of one analog input pin (ANInk) is converted once, using a signal specified by the ADnTSEL.ADnTRGSEL11 and ADnTSEL.ADnTRGSEL10 bits as a trigger, and the result of conversion is stored in one ADnCRk register. The ANInk pin and ADnCRk register correspond to each other on a one-to-one basis.

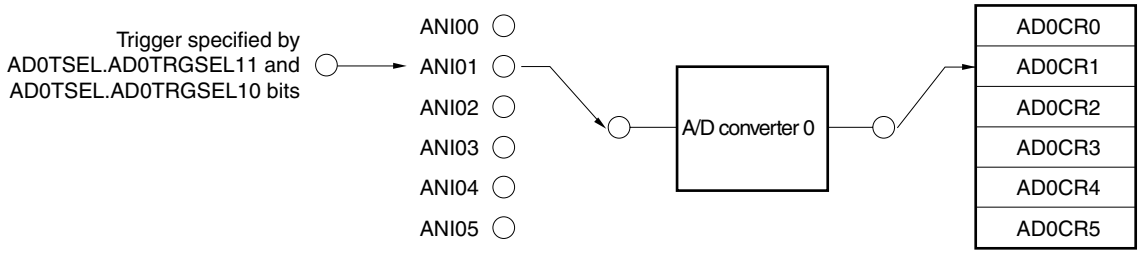
Each time conversion has been completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. After completing the conversion, the converter waits for the trigger with the ADnSCM.ADnCE bit set to 1.

This operation is suitable for an application where the result of A/D conversion should be read each time conversion by one trigger has been completed.

Analog Input Pin	A/D Conversion Result Register
ANInk	ADnCRk

Remark A/D converter 0: n = 0, k = 0 to 5
A/D converter 1: n = 1, k = 0 to 7

Figure 12-17. Example of 1-Channel Conversion Operation (Hardware Trigger Mode): A/D Converter 0



(1) AD0CE bit = 1 (enable)
(2) Trigger specified by AD0TSEL.AD0TRGSEL11 and AD0TSEL.AD0TRGSEL10 bits is generated
(3) Signal of ANI01 pin is A/D-converted
(4) Conversion result is stored in AD0CR1 register
(5) AD0SCM.AD0CS bit = 0
(6) INTAD0 interrupt request signal is generated
(7) Returns to (3) when the next trigger is input
(8) Set AD0CE bit to 0 to end (stop)

Remark This is an operation example when the AD0SCM.AD0PLM, AD0TRG1, and AD0TRG0 bits = 001, AD0CTL0.AD0MD1 and AD0CTL0.AD0MD0 bits = 00, and AD0CHEN register = 0002H.

(2) Operation of multiple channel conversion

The signals of two or more analog input pins specified by the ADnCHEN register are sequentially converted, starting from the pin with the lowest number, using a signal specified by the ADnTSEL.ADnTRGSEL11 and ADnTSEL.ADnTRGSEL10 bits as a trigger. The result of conversion is stored in the ADnCRk register corresponding to the analog input pin.

When conversion of the signals of all the specified analog input pins is completed, an A/Dn conversion end interrupt request signal (INTADn) is generated. After completion of conversion, the A/D converter waits for the trigger with the ADnSCM.ADnCE bit remaining set to 1.

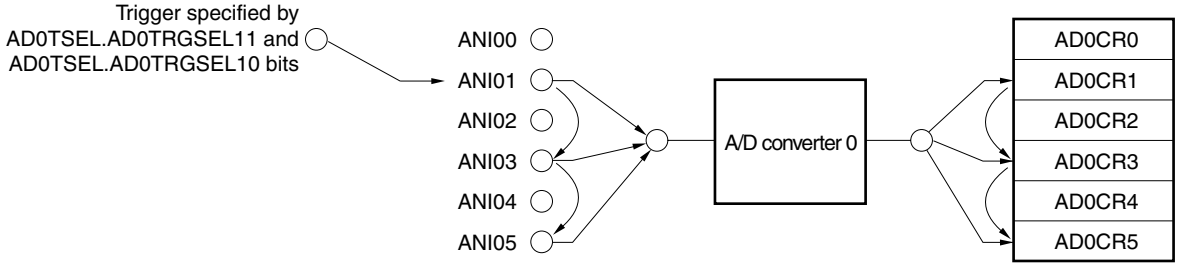
This operation is suitable for an application where two or more analog input signals should be monitored when the trigger is generated.

Analog Input Pin	A/D Conversion Result Register
ANInk[Note]	ADnCRk
|	|
ANInk[Note]	ADnCRk

Note Two or more can be specified by the ADnCHEN register.
However, A/D conversion is sequentially executed starting from the pin with the lowest number.

Remark A/D converter 0: n = 0, k = 0 to 5
A/D converter 1: n = 1, k = 0 to 7

Figure 12-18. Example of Multiple Channel Conversion Operation (Hardware Trigger Mode): A/D Converter 0



(1) AD0CE bit = 1 (enable)
(2) Trigger specified by AD0TSEL.AD0TRGSEL11 and AD0TSEL.AD0TRGSEL10 bits is generated
(3) Signal of ANI01 pin is A/D-converted
(4) Conversion result is stored in AD0CR1 register
(5) Signal of ANI03 pin is A/D-converted
(6) Conversion result is stored in AD0CR3 register
(7) Signal of ANI05 pin is A/D-converted
(8) Conversion result is stored in AD0CR5 register
(9) AD0SCM.AD0CS bit = 0
(10) INTAD0 interrupt request signal is generated
(11) Returns to (3) when the next trigger is input
(12) Set AD0CE bit to 0 to end (stop)

Remark This is an operation example when the AD0SCM.AD0PLM, AD0TRG1, and AD0TRG0 bits = 001, AD0CTL0.AD0MD1 and AD0CTL0.AD0MD0 bits = 00, and AD0CHEN register = 002AH.

12.4.10 Conversion channel specification mode (extension operation mode)

When the ADnSCM.ADnCE bit is set to 1, the A/D converter waits for a trigger. When selection trigger 1 specified by the ADnTSEL.ADnTRGSEL11 and ADnTSEL.ADnTRGSEL10 bits is generated, the converter starts A/D conversion.

When conversion is started, the ADnSCM.ADnCS bit is set to 1 (conversion is in progress).

If the ADnSCM register is written during A/D conversion operation, the conversion is stopped and the converter waits for the trigger again.

The analog input pin is specified by the ADnCH1.ADnTRGCH12 to ADnCH1.ADnTRGCH10 and ADnCH1.ADnTRGCH16 to ADnCH1.ADnTRGCH14 bits. Each time selection trigger 1 is generated, the analog input pins specified by the ADnCH1.ADnTRGCH12 to ADnCH1.ADnTRGCH10 and ADnCH1.ADnTRGCH16 to ADnCH1.ADnTRGCH14 bits are sequentially selected.

The signal of a specified analog input pin is converted the number of times specified by the ADnCHEN register (up to 16 times), using selection trigger 1 as the trigger, and the result is stored in the ADnCRm register specified by the ADnCHEN register. The conversion results are sequentially stored from ADnCR0.

When the signal of the specified analog input pin has been converted the number of times (up to 16 times) specified by the ADnCHEN register, an A/Dn conversion end interrupt request signal (INTADn) is generated. After A/D conversion is completed, the A/D converter waits for the trigger with the ADnSCM.ADnCE bit remaining set to 1.

This operation is suitable for an application where two or more analog input signals should be monitored.

Selection Trigger	Analog Input Pin	A/D Conversion Result Extension Register
Selection trigger 1	ANInx[Note 1]	ADnCR0[Note 3]
	ANInx[Note 1]	|
	ANInx[Note 1]	ADnCRm[Note 3]
Selection trigger 2	ANIny[Note 2]	ADnCR0[Note 3]
	ANIny[Note 2]	|
	ANIny[Note 2]	ADnCRm[Note 3]

Notes 1. Set by ADnCH1.ADnTRGCH12 to ADnCH1.ADnTRGCH10 bits

2. Set by ADnCH1.ADnTRGCH16 to ADnCH1.ADnTRGCH14 bits

3. Two or more times can be set by the ADnCHEN register.

Cautions 1. Be sure to set the hardware trigger mode as the conversion channel specification mode.

2. Be sure to set the ADnCHEN register using the lower bits, justifying to the bottom. Any other setting is prohibited.

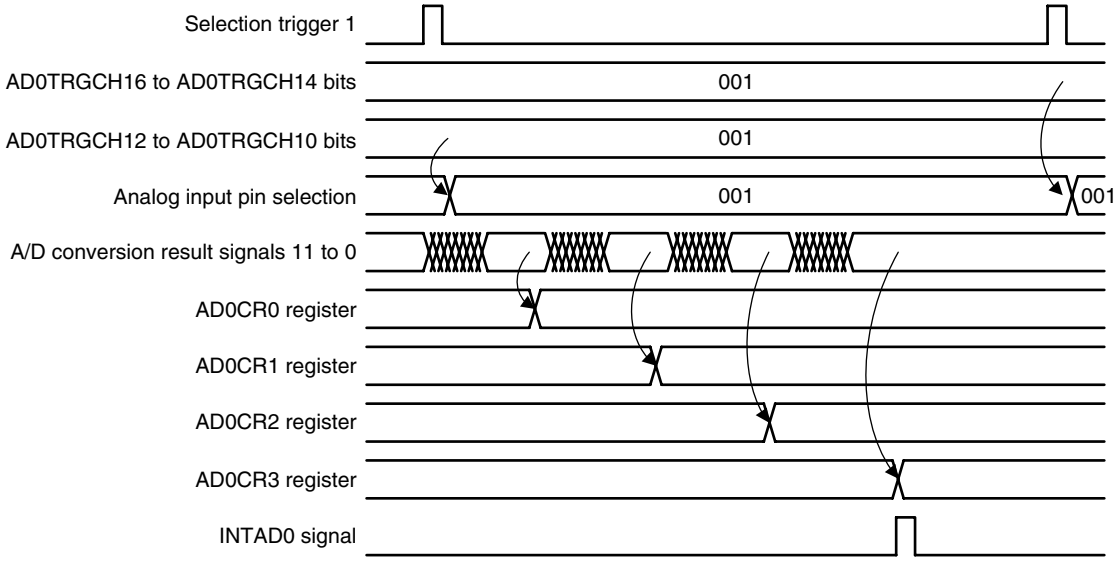
3. Setting of the ADnCH2 register is invalid.

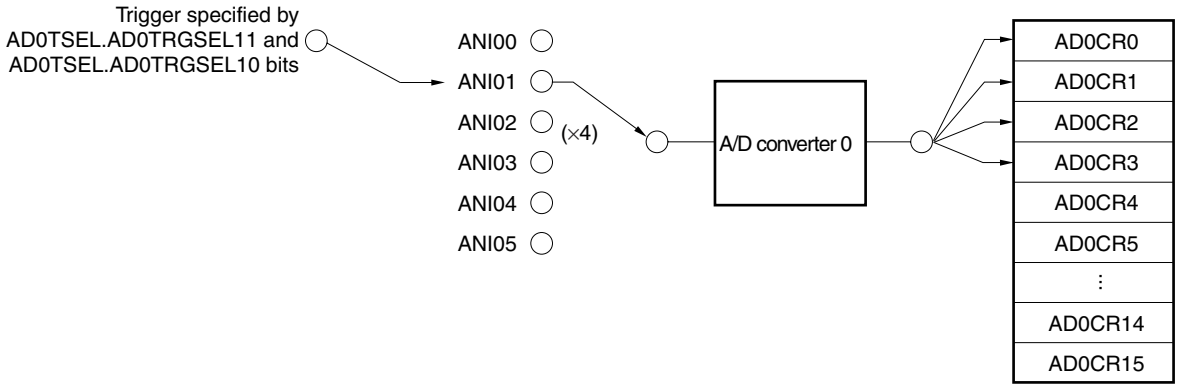
4. The ADnECRa, ADnECRaH, ADnFLG, and ADnFLGB registers are not used. If these registers are read, 0000H and 00H are read.

5. Selection trigger 1 is ignored if it is generated during A/D conversion operation. The next selection trigger 1 is accepted when a trigger is generated after completion of A/D conversion (after generation of the INTADn signal).

Remark A/D converter 0: n = 0, k = 0 to 5, m = 0 to 15
A/D converter 1: n = 1, k = 0 to 7, m = 0 to 15

Figure 12-19. Example of Operation in Conversion Channel Specification Mode: A/D Converter 0





(1) AD0CE bit = 1 (enable)
(2) Trigger specified by AD0TSEL.AD0TRGSEL11 and AD0TSEL.AD0TRGSEL10 bits is generated
(3) Signal of ANI01 pin is A/D-converted
(4) Conversion result is stored in AD0CR0 register
(5) Signal of ANI01 pin is A/D-converted
(6) Conversion result is stored in AD0CR1 register
(7) Signal of ANI01 pin is A/D-converted
(8) Conversion result is stored in AD0CR2 register
(9) Signal of ANI01 pin is A/D-converted
(10) Conversion result is stored in AD0CR3 register
(11) AD0SCM.AD0CS bit = 0
(12) INTAD0 interrupt request signal is generated
(13) Returns to (3) when the next trigger is input
(14) Set AD0CE bit to 0 to end (stop)

Remark This is an operation example when the AD0SCM.AD0PLM, AD0TRG1, and AD0TRG0 bits = 001, AD0CTL0.AD0MD1 and AD0CTL0.AD0MD0 bits = 10, AD0CHEN register = 000FH, AD0CH1.AD0TRGCH12 to AD0CH1.AD0TRGCH10 bits = 001, and AD0CH1.AD0TRGCH16 to AD0CH1.AD0TRGCH14 bits = 001.

12.4.11 Extension buffer mode (extension operation mode)

When the ADnSCM.ADnCE bit is set to 1, the A/D converter waits for a trigger. When selection trigger 1 specified by the ADnTSEL.ADnTRGSEL11 and ADnTSEL.ADnTRGSEL10 bits or selection trigger 2 specified by the ADnTSEL.ADnTRGSEL21 and ADnTSEL.ADnTRGSEL20 bits is generated, the converter starts A/D conversion.

When conversion is started, the ADnSCM.ADnCS bit is set to 1 (conversion is in progress).

If the ADnSCM register is written during A/D conversion operation, the conversion is stopped and the converter waits for the trigger again.

The analog input pin for selection trigger x is specified by the ADnCHx.ADnTRGCHx2 to ADnCHx.ADnTRGCHx0 and ADnCHx.ADnTRGCHx6 to ADnCHx.ADnTRGCHx4 bits. Each time selection trigger x is generated, the analog input pins specified by the ADnCHx.ADnTRGCHx2 to ADnCHx.ADnTRGCHx0 and ADnCHx.ADnTRGCHx6 to ADnCHx.ADnTRGCHx4 bits are sequentially selected.

When selection trigger 1 is used, the signal of the analog input pin specified by the ADnTRGCH12 to ADnTRGCH10 bits is converted when the trigger is generated for the first time. The result is stored in the A/Dn conversion result extension buffer register 0 and an A/Dn conversion end interrupt request signal (INTADn) is generated. When the trigger is generated the second time, the signal of the analog input pin specified by the ADnTRGCH16 to ADnTRGCH14 bits is converted. The result is stored in the A/Dn conversion result extension buffer register 0 and, at the same time, the first value stored in the A/Dn conversion result extension buffer register 0 is stored in the A/Dn conversion result extension buffer register 1. Then the INTADn interrupt request signal is generated. For A/D conversion using selection trigger 1, up to three A/Dn conversion result extension buffer registers, 0 to 2, can be used. When selection load trigger 1 is later generated, the values of the A/Dn conversion result extension buffer registers 0 to 2 are transferred to the ADnECR0 to ADnECR2 registers. After A/D conversion is competed, the converter waits for the trigger with the ADnSCM.ADnCE bit remaining set to 1.

When selection trigger 2 is used, the signal of the analog input pin specified by the ADnTRGCH22 to ADnTRGCH20 bits is converted when the trigger is generated for the first time, and the result is stored in the A/Dn conversion end extension buffer register 3. Then an A/Dn conversion end interrupt request signal (INTADn) is generated. When the trigger is generated the second time, the signal of the analog input pin specified by the ADnTRGCH26 to ADnTRGCH24 bits is converted and the result is stored in the A/Dn conversion result extension buffer register 4. At the same time, the value stored first in the A/Dn conversion result extension buffer register 3 is stored in the A/Dn conversion result extension buffer register 4, and the INTADn interrupt request signal is generated. When selection trigger 2 is used for A/D conversion, up to two A/Dn conversion result extension buffer registers, 3 and 4, can be used. When selection load trigger 2 is generated again, the values of the A/Dn conversion result extension buffer registers 3 and 4 are transferred to and stored in the ADnECR3 and ADnECR4 registers. After A/D conversion is completed, the converter waits for the trigger with the ADnCE bit remaining set to 1.

Therefore, the contents of the ADnECR0 to ADnECR4 registers can be saved to RAM all at once.

This operation is suitable for an application where there is little time to save the conversion result and two or more analog input signals should be monitored when a trigger is generated.

Selection Trigger	Analog Input Pin	A/D Conversion Result Extension Register
Selection trigger 1	ANInx[Note 1]	ADnECR0 to ADnECR2
Selection trigger 1	ANIny[Note 2]	ADnECR0, ADnECR1
Selection trigger 1	ANInx[Note 1]	ADnECR0
Selection trigger 2	ANIns[Note 3]	ADnECR3, ADnECR4
Selection trigger 2	ANInt[Note 4]	ADnECR3

Notes 1. Set by ADnCH1.ADnTRGCH12 to ADnCH1.ADnTRGCH10 bits

2. Set by ADnCH1.ADnTRGCH16 to ADnCH1.ADnTRGCH14 bits

3. Set by ADnCH2.ADnTRGCH22 to ADnCH2.ADnTRGCH20 bits

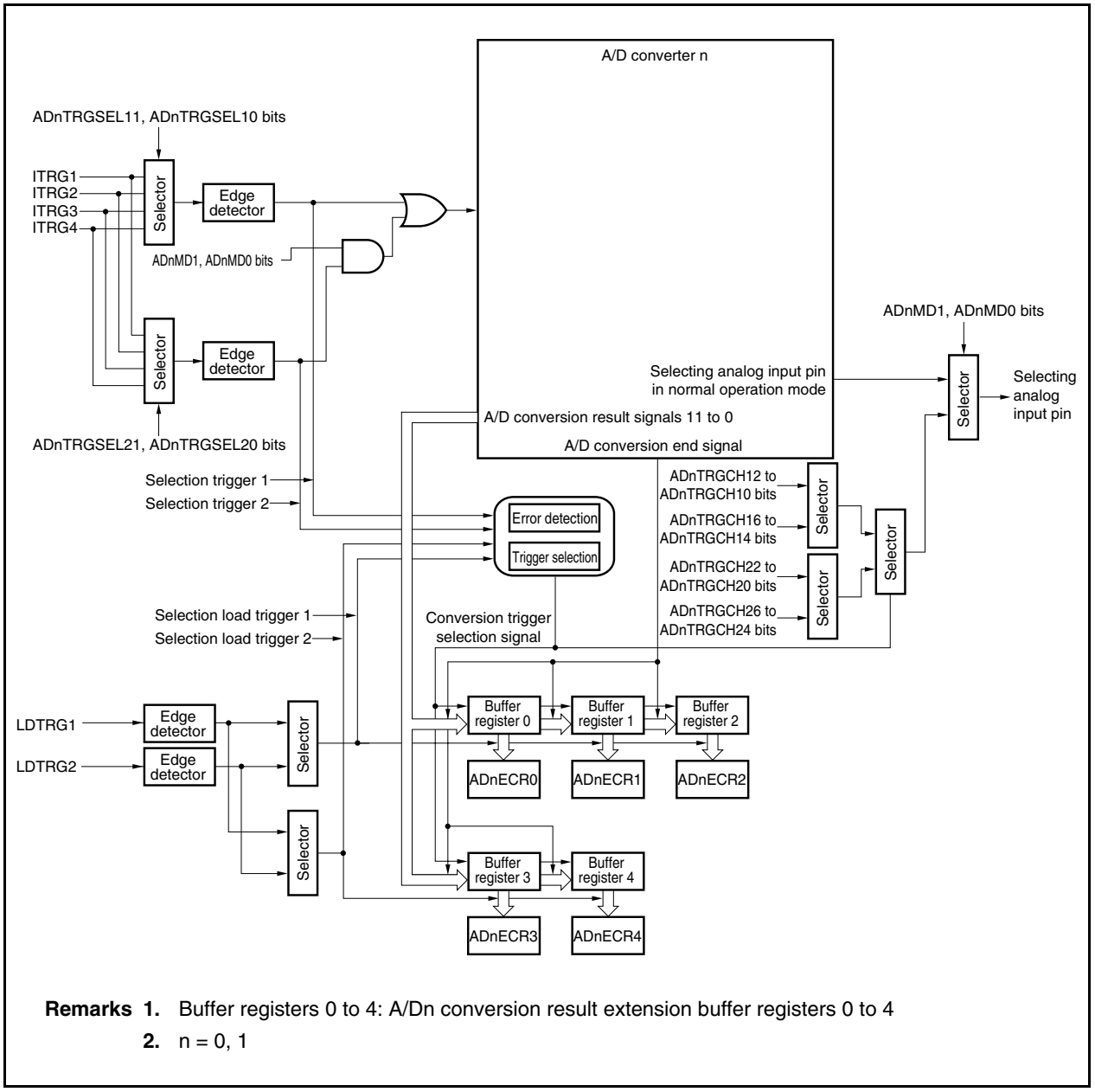
4. Set by ADnCH2.ADnTRGCH26 to ADnCH2.ADnTRGCH24 bits

Cautions 1. In the extension buffer mode, be sure to set the hardware trigger mode and the ADnCHEN register to 0001H.

2. The conversion result is stored in the ADnECRa register. The value of the ADnCRm register is undefined.

Remark n = 0, 1

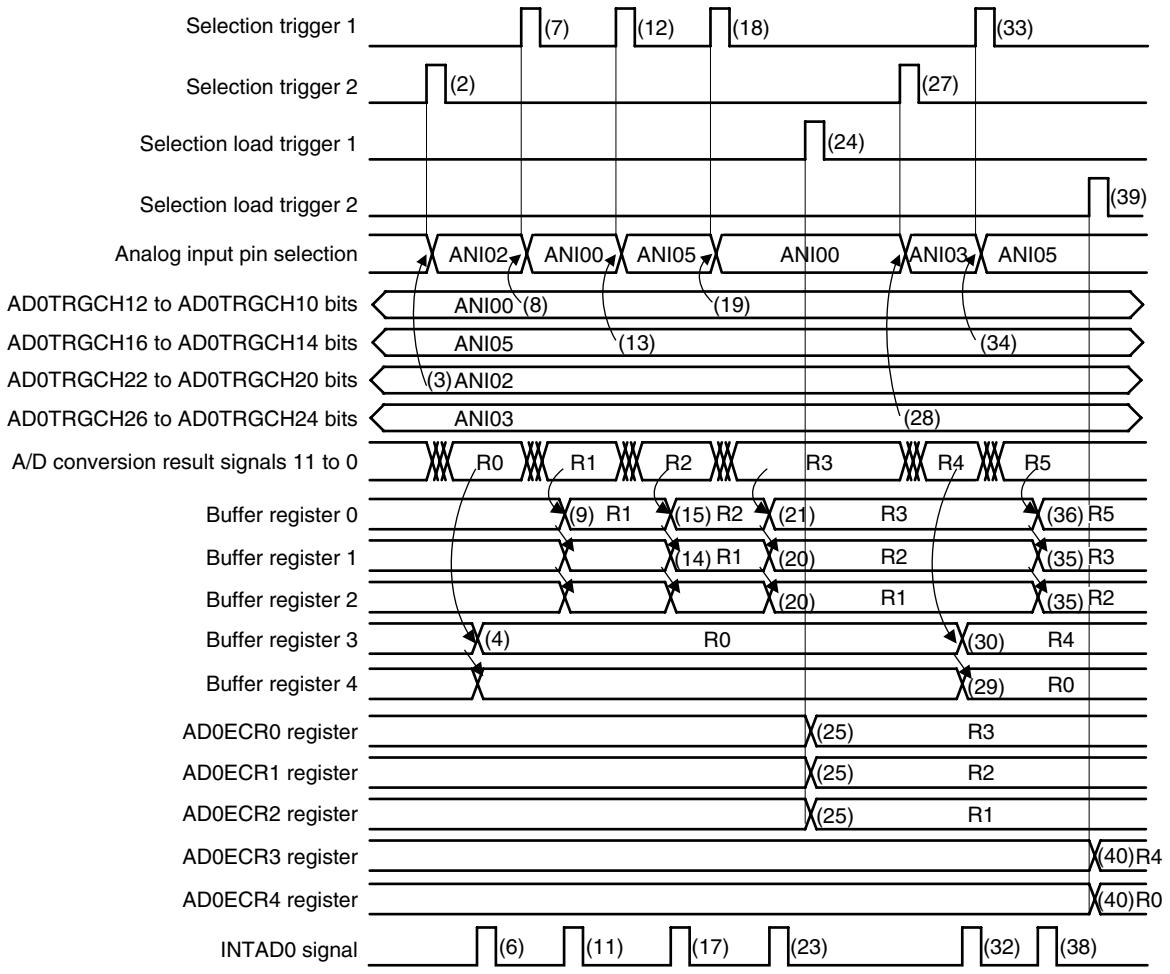
Figure 12-20. Block Diagram in Extension Buffer Mode



Remarks 1. Buffer registers 0 to 4: A/Dn conversion result extension buffer registers 0 to 4

2. n = 0, 1

Figure 12-21. Example of Operation in Extension Buffer Mode: A/D Converter 0 (1/2)



Remarks 1. Buffer registers 0 to 4: A/Dn conversion result extension buffer registers 0 to 4

2. R0 to R6: Conversion result

3. n = 0, 1

Figure 12-21. Example of Operation in Extension Buffer Mode: A/D Converter 0 (2/2)

(1) AD0CE bit = 1 (enable)
(2) Selection trigger 2 is generated
(3) Signal of ANI02 pin is A/D-converted
(4) Conversion result is stored in buffer register 3
(5) AD0SCM.AD0CS bit = 0
(6) INTAD0 interrupt request signal is generated
(7) Selection trigger 1 is generated
(8) Signal of ANI00 pin is A/D-converted
(9) Conversion result is stored in buffer register 0
(10) AD0SCM.AD0CS bit = 0
(11) INTAD0 interrupt request signal is generated
(12) Selection trigger 1 is generated
(13) Signal of ANI05 pin is A/D-converted
(14) Shifted from buffer register 0 to buffer register 1
(15) Conversion result is stored in buffer register 0
(16) AD0SCM.AD0CS bit = 0
(17) INTAD0 interrupt request signal is generated
(18) Selection trigger 1 is generated
(19) Signal of ANI00 pin is A/D-converted
(20) Shifted from buffer register 0 to buffer register 1 to buffer register 2
(21) Conversion result is stored in buffer register 0
(22) AD0SCM.AD0CS bit = 0
(23) INTAD0 interrupt request signal is generated
(24) Selection load trigger 1 is generated
(25) Shifted from buffer registers 0 to 2, to AD0ECR0 to AD0ECR2
(26) AD0SCM.AD0CS bit = 0
(27) Selection trigger 2 is generated
(28) Signal of ANI03 pin is A/D-converted
(29) Shifted from buffer register 3 to buffer register 4
(30) Conversion result is stored in buffer register 3
(31) AD0SCM.AD0CS bit = 0
(32) INTAD0 interrupt request signal is generated
(33) Selection trigger 1 is generated
(34) Signal of ANI05 pin is A/D-converted
(35) Shifted from buffer register 0 to buffer register 1 to buffer register 2
(36) Conversion result is stored in buffer register 0
(37) AD0SCM.AD0CS bit = 0
(38) INTAD0 interrupt request signal is generated
(39) Selection load trigger 2 is generated
(40) Shifted from buffer registers 3 and 4 to AD0ECR3 and AD0ECR4 registers
(41) AD0SCM.AD0CS bit = 0
(42) When the next trigger is input, the operation is performed in accordance with that trigger.
(43) Set ADnCE bit to 0 to end (stop)

(1) Error detection function

The extension buffer mode has an error detection function. If a trigger (selection trigger 1, selection trigger 2, selection load trigger 1, or selection load trigger 2) is generated during A/D conversion, an error occurs. The error is detected by the ADnFLG.ADnTERR2 and ADnFLG.ADnTERR1 flags, and ADnFLGB.ADnTERRB2 and ADnFLGB.ADnTERRB1 flags.

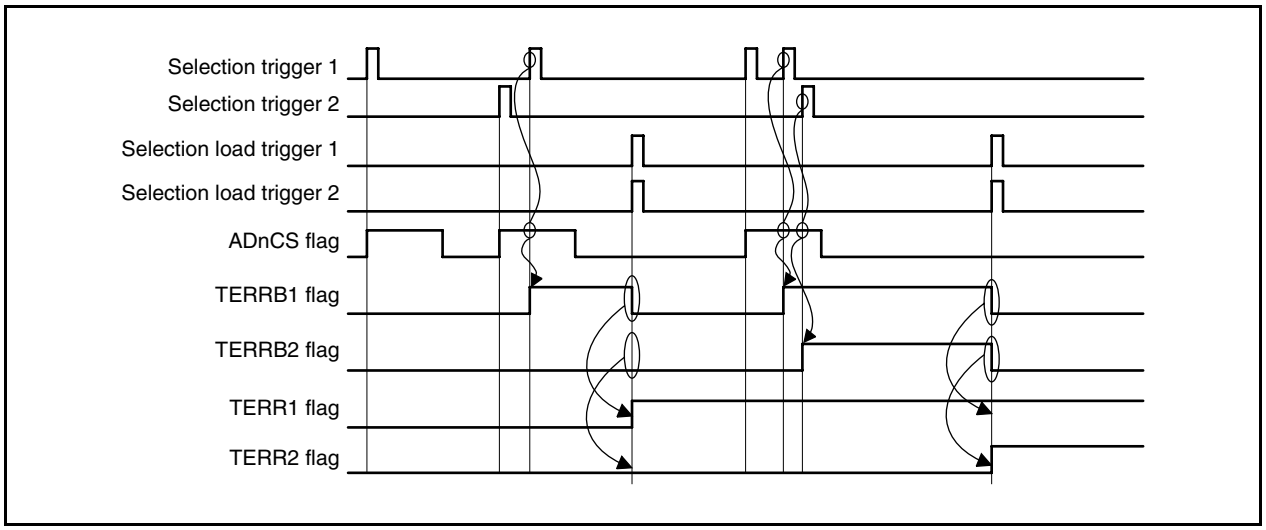
Cautions 1. Selection trigger 1, selection trigger 2, selection load trigger 1, and selection load trigger 2 are generated when asynchronous signals ITRG1 to ITRG4, LDTRG1, and LDTRG2 signals are synchronized. Although the timing of inputting these triggers seems to be the same, their simultaneous operation is not guaranteed because the asynchronous signals are synchronized.

2. Selection trigger 1 or 2 is ignored, even if it is generated again, during a period of up to 2.5 basic clocks (f_{AD01}) after the trigger is once generated (no error occurs).

(a) Error detection by generation of selection trigger 1 or 2 during A/D conversion

If selection trigger 1 is generated during A/D conversion, the ADnFLGB.ADnTERRB1 flag is set to 1 and A/D conversion by selection trigger 1 is ignored. If selection load trigger 1 is generated next, the value of the ADnTERRB1 flag is stored in the ADnFLG.ADnTERR1 flag.

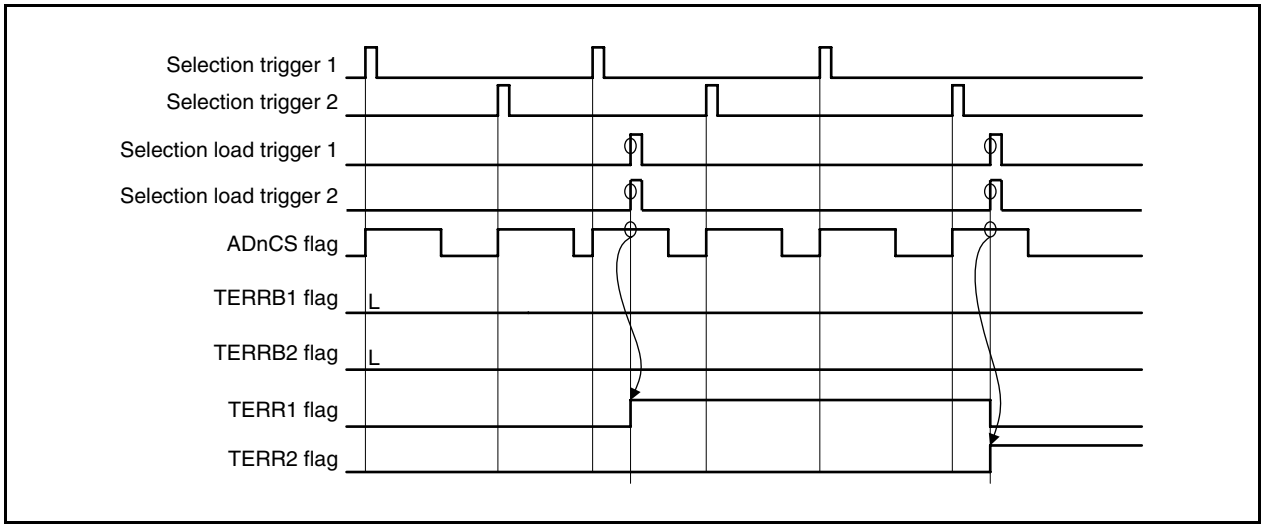
Similarly, if selection trigger 2 is generated during A/D conversion, the ADnFLGB.ADnTERRB2 flag is set to 1 and A/D conversion by selection trigger 2 is ignored. When selection load trigger 2 is generated next, the value of the ADnTERRB2 flag is stored in the ADnFLG.ADnTERR2 flag.



(b) Error detection by generation of selection load trigger 1 or 2 during A/D conversion

If selection load trigger 1 is generated during A/D conversion that uses selection trigger 1, the ADnFLG.ADnTERR1 flag is set to 1. A/D conversion and load operation are performed normally.

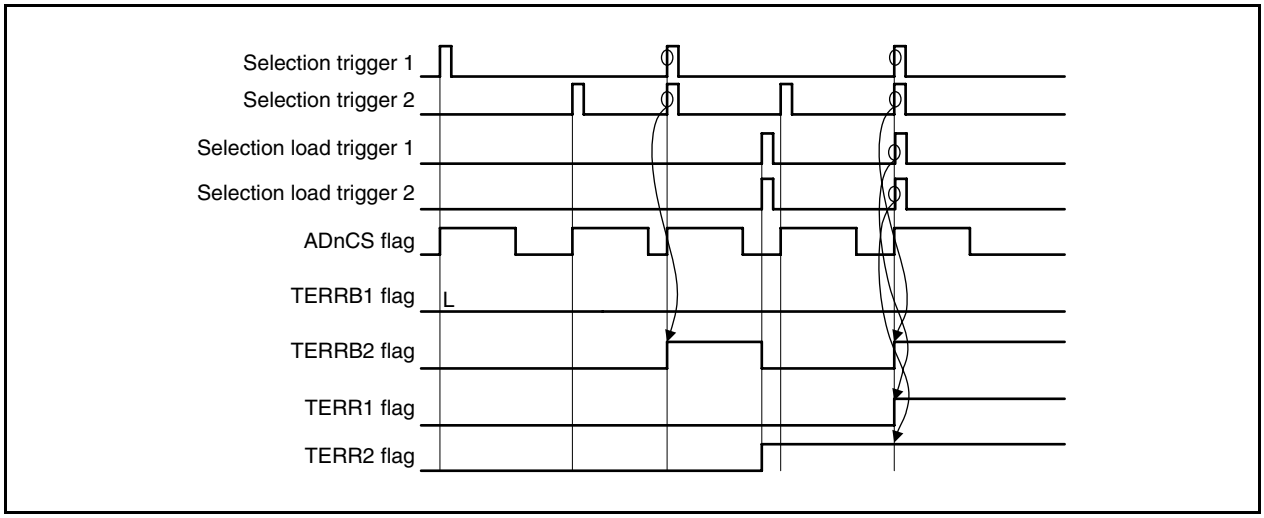
Similarly, if selection load trigger 2 is generated during A/D conversion that uses selection trigger 2, the ADnTERR2 flag is set to 1. A/D conversion and load operation are performed normally.



(c) Error detection by simultaneous generation of selection triggers 1 and 2, and of selection triggers 1 and 2, and selection load triggers 1 and 2

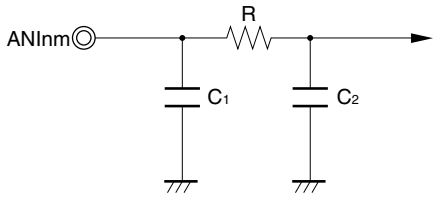
If selection triggers 1 and 2 are simultaneously generated, A/D conversion that uses selection trigger 1 is started and selection trigger 2 is ignored. Therefore, the ADnFLGB.ADnTERRB2 flag is set to 1.

If selection triggers 1 and 2, and selection load triggers 1 and 2 are simultaneously generated, the ADnFLGB.ADnTERRB2, ADnFLG.ADnTERR1, and ADnFLG.ADnTERR2 flags are set to 1. A/D conversion by selection trigger 1 and load operation of selection load triggers 1 and 2 are performed normally. Selection trigger 2 is ignored.



12.5 Internal Equivalent Circuit

The following figure shows the equivalent circuit of the analog input block.



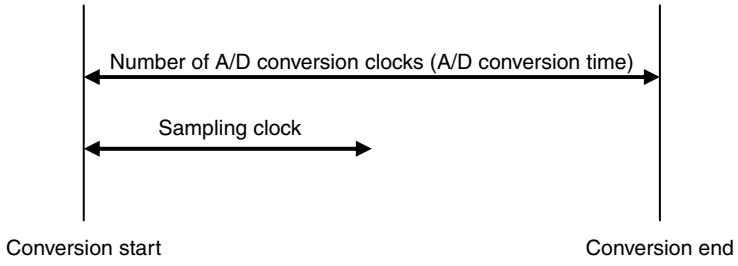
R	C_1	C_2
5.1 kΩ	15 pF	3.9 pF

Remarks 1. The maximum values are shown (reference values).

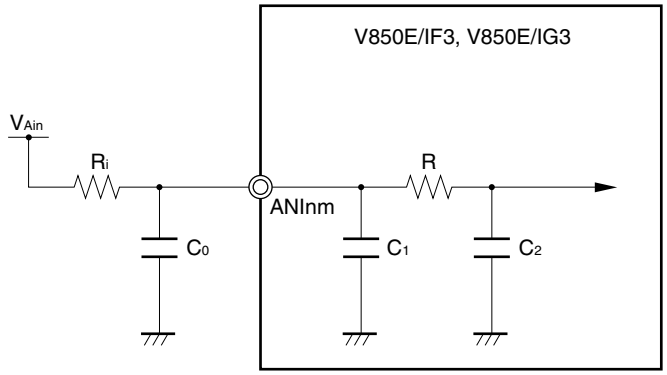
2. m = 0 to 5 when n = 0
m = 0 to 7 when n = 1

ADnCTC register				Number of A/D conversion clocks (f_{AD01})	Number of sampling clocks
ADnFR3 bit	ADnFR2 bit	ADnFR1 bit	ADnFR0 bit		
0	0	0	0	89	69.5
0	0	0	1	88	68.5
0	0	1	0	57	37.5
0	0	1	1	56	36.5
0	1	0	0	41	21.5
0	1	0	1	40	20.5
0	1	1	0	35	15.5
0	1	1	1	34	14.5
1	0	0	0	34	14.5
1	0	0	1	33	13.5
1	0	1	0	33	13.5
1	0	1	1	32	12.5
1	1	0	0	32	12.5
1	1	0	1	31	11.5
1	1	1	0	31	11.5
1	1	1	1	30	10.5

Caution Number of sampling clocks is included in number of A/D conversion clocks.



An example of calculating an overall error of A/D converters 0 and 1 is shown below.



fXX (MHz)	A/D conversion time (μs)	Sampling (μs)	R (kΩ)	C1 (pF)	C2 (pF)	C0 (pF)	Ri (kΩ)	Sampling error (LSB)[Note]
64	2.00 (31/fAD01)	0.78 (12.5/fAD01)	5.1	15	3.9	100	1.0	364.8
						100	0.5	30.4
						100	0.25	0.1 or lower
						100	0.125	0.1 or lower
						50	1.0	62.4
						50	0.5	0.8
						50	0.25	0.1 or lower
						50	0.125	0.1 or lower

Note The error when considering the signal source impedance is “sampling error + overall error”.

Remarks 1. These values are reference values calculated by simulating what happens to C_2 voltage by R_i and C_0 when V_{Ain} is applied from 0 V to 5 V at the same time as sampling start.

2. m = 0 to 5 when n = 0
m = 0 to 7 when n = 1

3. f_{XX}: System clock frequency
f_{AD01}: Basic clock frequency

12.6 Cautions

12.6.1 Stopping conversion operation

The ongoing conversion operation is stopped when 0 is written to the ADnSCM.ADnCE bit. At this time, the conversion result in the A/Dn conversion result register m (ADnCRm) and A/Dn conversion result extension register a (ADnECRa) is undefined. Therefore, read the A/D conversion result after A/D conversion has been completed (after the A/Dn conversion end interrupt request signal (INTADn) has been issued), and then write 0 to the ADnCE bit as necessary.

Note that the ADnCE bit is not cleared to 0 in all the modes even after the INTADn signal is generated.

Remark n = 0, 1
m = 0 to 15

12.6.2 Interval of trigger during conversion operation in hardware trigger mode, conversion channel specification mode, and extension buffer mode

Inputting a trigger during conversion operation is ignored in the hardware trigger mode, conversion channel specification mode, and extension buffer mode. Therefore, the interval of the trigger (input time) in the hardware trigger mode, conversion channel specification mode, and extension buffer mode must be longer than the A/D conversion time specified by the ADnCTC.ADnFR3 to ADnCTC.ADnFR0 bits (see **Table 12-2 Number of A/D Conversion Clocks and A/D Conversion Time**).

Remark n = 0, 1

12.6.3 Writing to ADnSCM register

(1) Restarting A/D conversion

To restart A/D conversion, write the same value to the ADnSCM register. To change the ADnPLM, ADnTRG1, and ADnTRG0 bits, be sure to set the ADnCE bit to 0.

(2) Contention between end of A/D conversion and writing to ADnSCM register

If completion of A/D conversion contends with writing to the ADnSCM register during A/D conversion operation, the conversion result is correctly stored in the ADnCRm and ADnECRa registers, if the A/Dn conversion end interrupt request signal (INTADn) is generated. If the INTADn signal is not generated, the A/D conversion operation is aborted. Therefore, the previous conversion result is held by the ADnCRm and ADnECRa registers.

(3) Successive writing to ADnSCM register

To successively write the ADnSCM0 register when the conversion operation is enabled (ADnCE bit = 1), be sure to wait for time of at least 5 basic clocks (f_{AD01}).

The ADnSCM0 register can be successively written when the ADnCE bit is set to 1 after the ADSCMn register is written while the ADnCE bit = 0.

Remark n = 0, 1

12.6.4 A/D conversion start timing

In the conversion channel specification mode and extension buffer mode, starting A/D conversion is delayed up to 1.5 basic clocks (f_{AD01}) as compared with the normal operation mode.

12.6.5 Operation in standby mode

(1) HALT mode

The A/D conversion operation continues. If the HALT mode is released by a maskable interrupt request signal that is not masked, the values of the ADnSCM, ADnCRm, and ADnECRa registers are held.

(2) IDLE mode and STOP mode

No conversion operation is performed because clock supply to A/D converters 0 and 1 is stopped.

Be sure to set the ADnSCM.ADnCE bit to 0 when the IDLE or STOP mode is set. At this time, setting the A/D power save mode (ADnSCM.ADnPS bit = 0) is recommended.

Remark n = 0, 1
m = 0 to 15

12.6.6 Timing of accepting trigger in conversion channel specification mode and extension buffer mode

In the conversion channel specification mode and extension buffer mode, selection trigger 1 or 2 is ignored, even if it is generated again, until the A/Dn conversion end interrupt signal (INTADn) is generated after A/D conversion is started by the first generation of selection trigger 1 or 2. In the extension buffer mode, the error flag is set to 1 in accordance with a specified error condition if selection trigger 1 or 2, or selection load trigger 1 or 2 is generated during this period (except, however, the case in **Caution 2** in **12.4.11 (1) Error detection function**).

Remark n = 0, 1

12.6.7 Variation of A/D conversion results

The results of the A/D conversion may vary depending on the fluctuation of the supply voltage, or may be affected by noise. To reduce the variation, take counteractive measures with the program, such as by averaging the A/D conversion results.

12.6.8 A/D conversion result hysteresis characteristics

Successive comparison type A/D converters hold an analog input voltage in an internal sample & hold capacitor and then perform A/D conversion. After the A/D conversion has finished, the analog input voltage remains in the internal sample & hold capacitor. As a result, the following phenomena may occur.

- When the same channel is used for A/D conversions, if the voltage is higher or lower than the previous A/D conversion, then hysteresis characteristics may appear where the conversion result is affected by the previous value. Even if the conversion were to be performed at the same potential, the results may thus vary.
- When switching the analog input channel, hysteresis characteristics may appear where the conversion result is affected by the previous channel value. This is because one A/D converter is used for the A/D conversions. Even if the conversion were to be performed at the same potential, the results may thus vary.

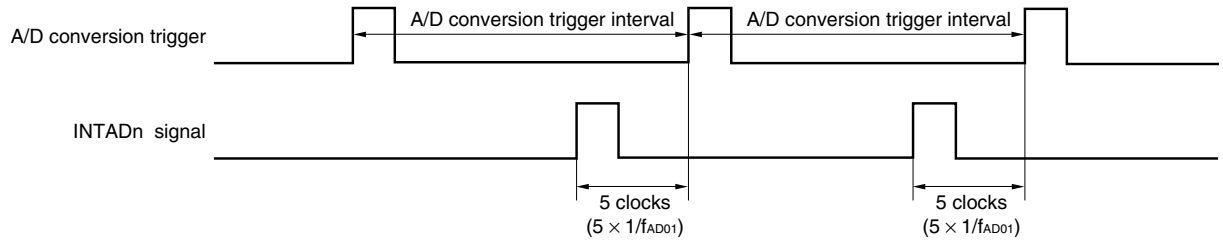
To obtain more accurate conversion results, execute A/D conversion twice consecutively on the same channel, and discard the first conversion result.

12.6.9 A/D conversion trigger interval for continuous conversion

For the A/D conversion trigger interval for continuous conversion, secure at least the minimum trigger interval shown below before inputting the next trigger. Otherwise, the trigger will be invalid (not retained).

Minimum trigger interval clock count = A/D conversion clock count + 5 clocks
Minimum trigger interval time = Minimum trigger interval clock count × $1/f_{AD01}$

Example f_{AD01} = 16 MHz, A/D conversion time = 2 μs, A/D conversion clock count = 32 clocks
Minimum trigger interval clock count = 32 + 5 = 37
Minimum trigger interval timer = 37 × 1/16 = 2.3125 [μs]



Remark n = 0, 1

12.7 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 1 LSB (Least Significant Bit). The percentage of 1 LSB 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_{REFPn} – 0)/100
= AV_{REFPn}/100

1 LSB is as follows when the resolution is 12 bits.

1 LSB = $1/2^{12}$ = 1/4,096
= 0.024%FSR

Accuracy has no relation to resolution, but is determined by overall error.

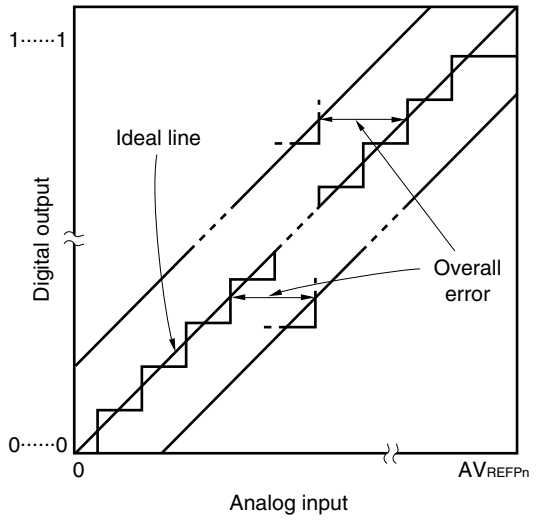
(2) Overall error

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 12-22. Overall Error

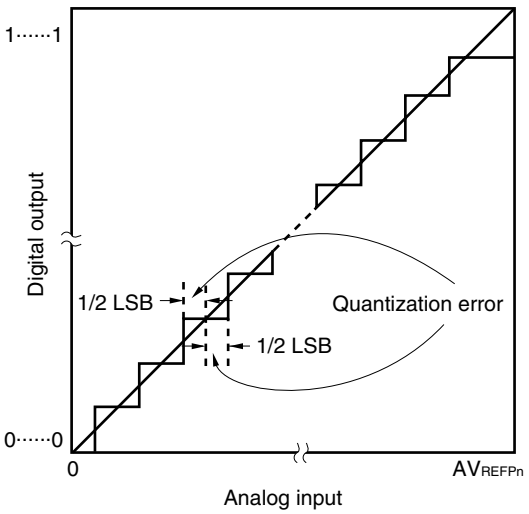


(3) Quantization error

When analog values are converted to digital values, a ±1/2 LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2 LSB 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.

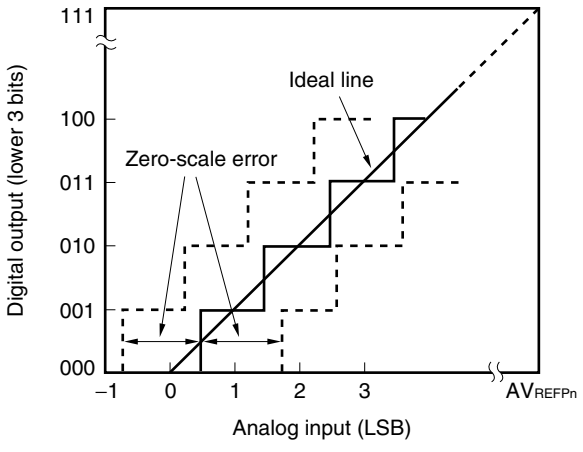
Figure 12-23. Quantization Error



(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.

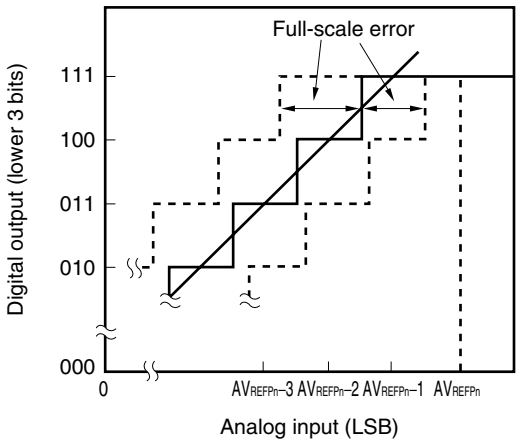
Figure 12-24. Zero-Scale Error



(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (full-scale value − 3/2 LSB) when the digital output changes from 1……110 to 1……111.

Figure 12-25. Full-Scale Error

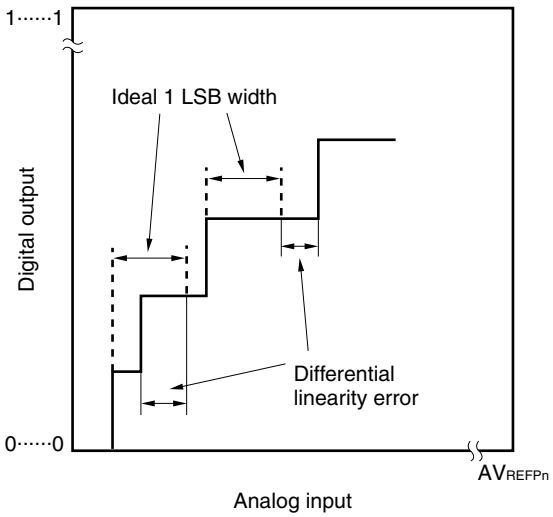


(6) Differential linearity error

While the ideal width of code output is 1 LSB, this indicates the difference between the actual measurement value and the ideal value.

This indicates the basic characteristics of the A/D conversion when the voltage applied to the analog input pins of the same channel is consistently increased bit by bit from AV_{SSn} to AV_{REFPn}. See **12.7 (2) Overall error** for when the input voltage is increased or decreased, or when two or more channels are used.

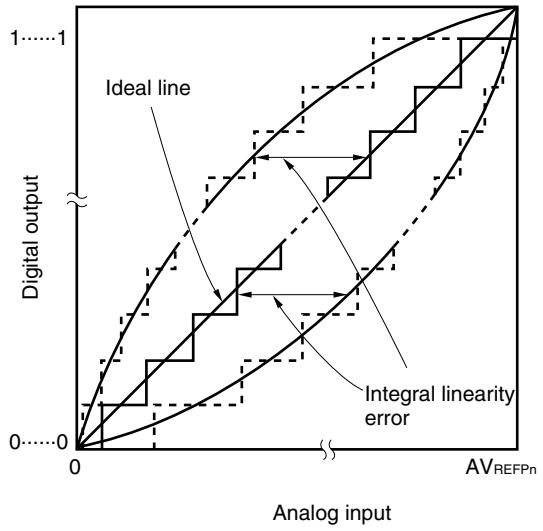
Figure 12-26. Differential Linearity Error



(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 12-27. Integral Linearity Error



(8) Conversion time

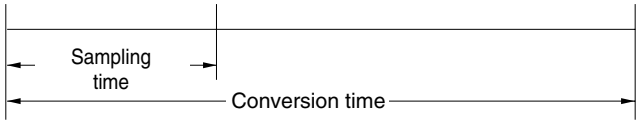
This expresses the time from when the trigger is generated to when the digital output is obtained.

The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.

Figure 12-28. Sampling Time



CHAPTER 13 A/D CONVERTER 2

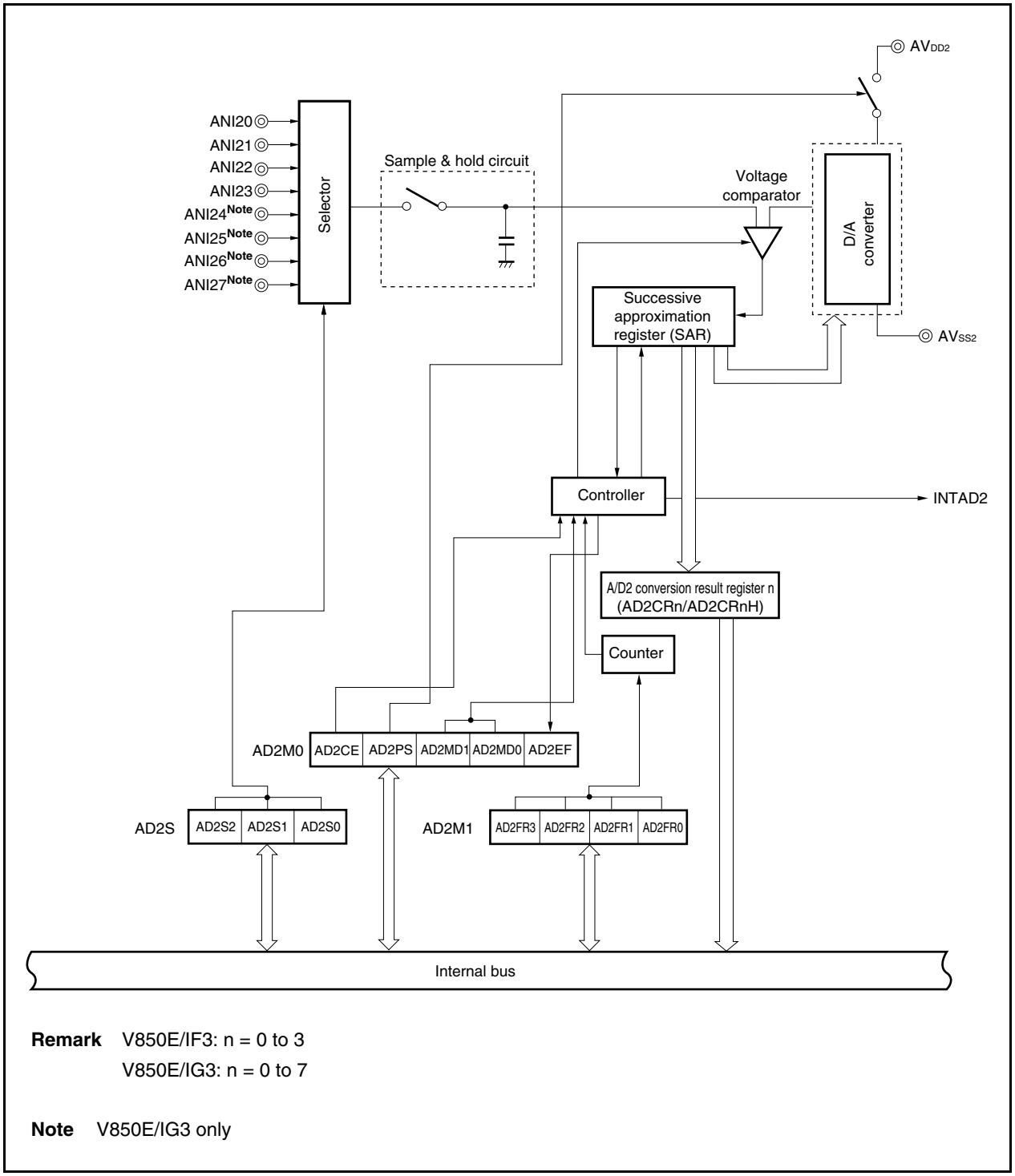
13.1 Features

- On-chip 10-bit resolution A/D converter
- Analog input
  V850E/IF3: ANI20 to ANI23 (4 channels)
  V850E/IG3: ANI20 to ANI27 (8 channels)
- A/D conversion result register
  V850E/IF3: AD2CR0 to AD2CR3 (10 bits × 4)
  V850E/IG3: AD2CR0 to AD2CR7 (10 bits × 8)
- A/D conversion trigger mode
  Software trigger mode
- A/D conversion operation mode
  Continuous select mode
  Continuous scan mode
  One-shot select mode
  One-shot scan mode
- Successive comparison approximation method
- Operating voltage: $EV_{DD0} = EV_{DD1} = EV_{DD2}$ (V850E/IG3 only) = AV_{DD2} = 4.0 to 5.5 V (target)

13.2 Configuration

The block diagram is shown below.

Figure 13-1. Block Diagram of A/D Converter 2



Remark V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

Note V850E/IG3 only

Cautions 1. If there is noise at the analog input pin (ANI2n) and at the A/D converter power supply voltage pin (AV_{DD2}), that noise may generate an illegal conversion result.

Software processing will be needed to avoid a negative effect on the system from this illegal conversion result.

An example of this software processing is shown below.

- **Take the average result of a number of A/D conversions and use that as the A/D conversion result.**
- **Execute a number of A/D conversions successively and use those results, omitting any exceptional results that may have been obtained.**
- **If an A/D conversion result that is judged to have generated a system malfunction is obtained, be sure to recheck the system malfunction before performing malfunction processing.**

2. Do not apply a voltage outside the AV_{SS2} to AV_{DD2} range to the pins that are used as input pins of A/D converter 2.

A/D converter 2 consists of the following hardware.

Table 13-1. Configuration of A/D Converter 2

Item	Configuration
Analog input	V850E/IF3: ANI20 to ANI23 (4 channels) V850E/IG3: ANI20 to ANI27 (8 channels)
Registers	Successive approximation register (SAR) V850E/IF3: A/D2 conversion result registers 0 to 3 (AD2CR0 to AD2CR3) A/D2 conversion result registers 0H to 3H (AD2CR0H to AD2CR3H): Only the higher 8 bits can be read V850E/IG3: A/D2 conversion result registers 0 to 7 (AD2CR0 to AD2CR7) A/D2 conversion result registers 0H to 7H (AD2CR0H to AD2CR7H): Only the higher 8 bits can be read
Control registers	A/D converter 2 mode registers 0, 1 (AD2M0, AD2M1) A/D converter 2 channel specification register (AD2S)

(1) Successive approximation register (SAR)

The SAR register is a register that compares the voltage value of an analog input pin with the value of the voltage tap of the D/A converter and holds the result, starting from the most significant bit (MSB).

If data is held in the SAR all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register are transferred in AD2CRn register.

When all the specified A/D conversion operations have ended, an A/D2 conversion end interrupt request signal (INTAD2) is generated.

(2) A/D conversion result register n (AD2CRn), A/D conversion result register nH (AD2CRnH)

The AD2CRn register is a register that holds the A/D conversion results. The conversion result is stored in the higher 10 bits of the AD2CRn register corresponding to the analog input. The lower 6 bits of these registers are always 0 when read.

The higher 8 bits of the result of A/D conversion are read from the AD2CRn register.

To read the result of A/D conversion in 16-bit units, specify the AD2CRn register. To read the higher 8 bits, specify the AD2CRnH register.

Caution The contents of the AD2CRn register may become undefined depending on the operation to write the AD2M0, AD2M1, and AD2S registers. Read the result of conversion from the AD2CRn register after conversion and before writing the AD2M0, AD2M1, and AD2S registers. The correct conversion result cannot be read from the AD2CRn register if any other procedure is used.

(3) Sample & hold circuit

The sample & hold circuit samples the analog input signals selected by the input circuit and sends the sampled data to the voltage comparator. This circuit holds the sampled analog input voltage during A/D conversion.

(4) Voltage comparator

The voltage comparator compares the value that is sampled and held with the voltage generated from the voltage tap of the D/A converter.

(5) D/A converter

The D/A converter is connected between AV_{DD2} and AV_{SS2} and generates a voltage to be compared with an input analog signal.

(6) ANI2n pin

The ANI2n pin is an analog input pin for A/D converter 2. This pin inputs the analog signals to be A/D converted. Pins other than the one that is selected by the AD2S register as analog signal input pins can be used as input port pins.

Cautions 1. Make sure that the voltages input to the ANI2n pin do not exceed the rated values. If a voltage higher than or equal to AV_{DD2} or lower than or equal to AV_{SS2} is input to a channel, the conversion value of the channel is undefined, and the conversion values of the other channels may also be affected.

2. The analog input pin (ANI2n) is alternately used as input port pin (P7n). If an instruction to input a signal to port 7 is executed during conversion when one of ANI2n is selected for A/D conversion, the resolution for conversion may drop.

(7) AV_{DD2} pin

The AV_{DD2} pin alternately functions as the pin for inputting the positive power supply and reference voltage of A/D converter 2. This pin converts signals input to the ANI2n pin to digital signals based on the voltage applied between AV_{DD2} and AV_{SS2}.

Always make the potential at this pin the same as that at the EV_{DD0}, EV_{DD1}, and EV_{DD2} pins (V850E/IG3 only) even when A/D converter 2 is not used.

The operating voltage range of the AV_{DD2} pin is $EV_{DD0} = EV_{DD1} = EV_{DD2}$ (V850E/IG3 only) = AV_{DD2} = 4.0 to 5.5 V (target).

(8) AV_{SS2} pin

This is the ground pin of A/D converter 2. Always make the potential at this pin the same as that at the EV_{SS0}, EV_{SS1}, and EV_{SS2} (V850E/IG3 only) pins even when A/D converter 2 is not used.

Remark V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

13.3 Control Registers

A/D converter 2 is controlled by the following registers.

- A/D converter 2 mode registers 0, 1 (AD2M0 to AD2M1)
- A/D converter 2 channel specification register (AD2S)

The following registers are also used.

- A/D2 conversion result register n (AD2CRn)
- A/D2 conversion result register nH (AD2CRnH)

(1) A/D converter 2 mode register 0 (AD2M0)

The AD2M0 register is a register that specifies the operation mode and controls conversion operations.

This register can be read or written in 8-bit or 1-bit units. However, bit 0 is read-only.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFB80H

	<7>	6	5	4	3	2	1	0
AD2M0	AD2CE	AD2PS	AD2MD1	AD2MD0	0	0	0	AD2EF

AD2CE	Control of A/D conversion operation
0	Conversion operation stopped
1	Conversion operation enabled

AD2PS	A/D conversion control
0	A/D power on
1	A/D power off

- The first result of conversion by the A/D converter 2 becomes valid when the AD2CE bit is set to 1 (when conversion is enabled) at least 2 μs after the AD2PS bit is set to 1 (A/D power is turned on).
  If the AD2CE bit is set to 1 before 2 μs pass, the conversion operation is started and ends after the A/D conversion time, but the conversion result is undefined.
- When the A/D converter 2 is not used, set to 0 the AD2CE bit (stop conversion operation) and the AD2PS bit (turn off A/D power) to reduce the power consumption.
- Do not set the AD2PS2 bit during A/D conversion operation (AD2EF bit = 1). While the A/D conversion operation is not performed, the AD2CE and AD2PS bits can be simultaneously cleared to 0.

AD2MD1	AD2MD0	Specification of operation mode
0	0	Successive select mode
0	1	Successive scan mode
1	0	One-shot select mode
1	1	One-shot scan mode

AD2EF	Status of A/D converter 2 (status)
0	During A/D conversion stop
1	During A/D conversion operation

Cautions 1. Writing to bit 0 is ignored.

2. The conversion resolution of the pin to which an analog signal is input first immediately after A/D conversion is started may drop. For details, see 13.7 (6) About AV_{DD2} pin.

3. A/D conversion is stopped and started again from the beginning if the AD2M0 and AD2S registers are written during A/D conversion operation (AD2EF bit = 1).

4. Be sure to set bits 1 to 3 to "0".

(2) A/D converter 2 mode register 1 (AD2M1)

The AD2M1 register is a register that specifies the number of A/D conversion clocks and A/D conversion time.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFB81H

	7	6	5	4	3	2	1	0
AD2M1	0	0	0	0	AD2FR3	AD2FR2	AD2FR1	AD2FR0

Cautions 1. See Table 13-2 Setting Example During Conversion Mode for the AD2FR3 to AD2FR0 bits.

2. Changing the AD2FR3 to AD2FR0 bits is prohibited during conversion operation (AD2CE bit = 1).

3. Be sure to set bits 4 to 7 to "0".

Table 13-2. Setting Example During Conversion Mode

AD2FR3	AD2FR2	AD2FR1	AD2FR0	Number of A/D Conversion Clocks[Note]	A/D Conversion Time	f_{AD2} = 32 MHz (f_{XX} = 64 MHz)	$f_{AD}2$ = 24 MHz (f_{XX} = 48 MHz)
0	0	0	1	62	$62/f_{AD2}$	Setting prohibited	Setting prohibited
0	0	1	0	93	$93/f_{AD2}$	Setting prohibited	3.86 μs
0	0	1	1	124	$124/f_{AD2}$	3.88 μs	5.17 μs
0	1	0	0	155	$155/f_{AD2}$	4.84 μs	6.46 μs
0	1	0	1	186	$186/f_{AD2}$	5.81 μs	7.75 μs
0	1	1	0	217	$217/f_{AD2}$	6.78 μs	9.04 μs
0	1	1	1	248	$248/f_{AD2}$	7.75 μs	Setting prohibited
1	0	0	0	279	$279/f_{AD2}$	8.72 μs	Setting prohibited
1	0	0	1	310	$310/f_{AD2}$	9.69 μs	Setting prohibited
Other than above					Setting prohibited		

Note The number of clocks (f_{AD2}) from the start to the end of A/D conversion.

Caution Set the A/D conversion time to 3.8 μs or more.

Remark f_{AD2}: Operating clock of A/D converter 2

(3) A/D converter 2 channel specification register (AD2S)

The AD2S register is a register that specifies the analog input pin to be A/D-converted.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFB82H

	7	6	5	4	3	2	1	0
AD2S	0	0	0	0	0	AD2S2	AD2S1	AD2S0

AD2S2	AD2S1	AD2S0	Select mode	Scan mode
0	0	0	ANI20	ANI20
0	0	1	ANI21	ANI20, ANI21
0	1	0	ANI22	ANI20 to ANI22
0	1	1	ANI23	ANI20 to ANI23
1	0	0	ANI24Note	ANI20 to ANI24Note
1	0	1	ANI25Note	ANI20 to ANI25Note
1	1	0	ANI26Note	ANI20 to ANI26Note
1	1	1	ANI27Note	ANI20 to ANI27Note

Note V850E/IG3 only.

With the V850E/IF3, this setting is not available because the necessary pins are not provided.

Caution Be sure to set bits 3 to 7 to “0”.

(4) A/D2 conversion result registers n, nH (AD2CRn, AD2CRnH)

The AD2CRn and AD2CRnH registers are registers that hold the A/D conversion results. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR) and stored in the higher 10 bits of the AD2CRn register. The lower 6 bits of these registers are always 0 when read.

The higher 8 bits of A/D conversion result are read to the AD2CRnH register.

These registers can only be read in 16-bit or 8-bit units. When the A/D conversion results are read in 16-bit units, the AD2CRn register is specified, and when the higher 8 bits are read, the AD2CRnH register is specified.

Reset sets AD2CRn register to 0000H and AD2CRnH register to 00H.

Caution If a write operation is performed on the AD2M0, AD2M1, and AD2S registers, the contents of the AD2CRn register may become undefined. Read the conversion result after the conversion operation and before performing a write operation on the AD2M0, AD2M1, and AD2S registers. The correct conversion result may not be read if the timing is other than the above.

After reset: 0000H R Address: AD2CR0 FFFFFB90H, AD2CR1 FFFFFB92H, AD2CR2 FFFFFB94H, AD2CR3 FFFFFB96H, AD2CR4 FFFFFB98H[Note], AD2CR5 FFFFFB9AH[Note], AD2CR6 FFFFFB9CH[Note], AD2CR7 FFFFFB9EH[Note]

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AD2CRn	AD 29	AD 28	AD 27	AD 26	AD 25	AD 24	AD 23	AD 22	AD 21	AD 20	0	0	0	0	0	0

V850E/IF3 n = 0 to 3
V850E/IG3 n = 0 to 7

Note V850E/IG3 only.

After reset: 00H R Address: AD2CR0H FFFFFB91H, AD2CR1H FFFFFB93H, AD2CR2H FFFFFB95H, AD2CR3H FFFFFB97H, AD2CR4H FFFFFB99H[Note], AD2CR5H FFFFFB9BH[Note], AD2CR6H FFFFFB9DH[Note], AD2CR7H FFFFFB9FH[Note]

	7	6	5	4	3	2	1	0
AD2CRnH	AD29	AD28	AD27	AD26	AD25	AD24	AD23	AD22

V850E/IF3 n = 0 to 3
V850E/IG3 n = 0 to 7

Note V850E/IG3 only.

The correspondence between the analog input pins and the AD2CRn and AD2CRnH registers is shown below.

Table 13-3. Correspondence Between Analog Input Pins and AD2CRn and AD2CRnH Registers

Analog Input Pin	A/D Conversion Result Register
ANI20	AD2CR0, AD2CR0H
ANI21	AD2CR1, AD2CR1H
ANI22	AD2CR2, AD2CR2H
ANI23	AD2CR3, AD2CR3H
ANI24[Note]	AD2CR4[Note], AD2CR4H[Note]
ANI25[Note]	AD2CR5[Note], AD2CR5H[Note]
ANI26[Note]	AD2CR6[Note], AD2CR6H[Note]
ANI27[Note]	AD2CR7[Note], AD2CR7H[Note]

Note V850E/IG3 only

The relationship between the analog voltage input to the analog input pin (ANI2n) and the A/D conversion result (of A/D2 conversion result register n (AD2CRn)) is as follows:

$$SAR = INT\left(\frac{V_{IN}}{AV_{DD2}} \times 1{,}024 + 0.5\right)$$

$$ADCR^{Note} = SAR \times 64$$

or,

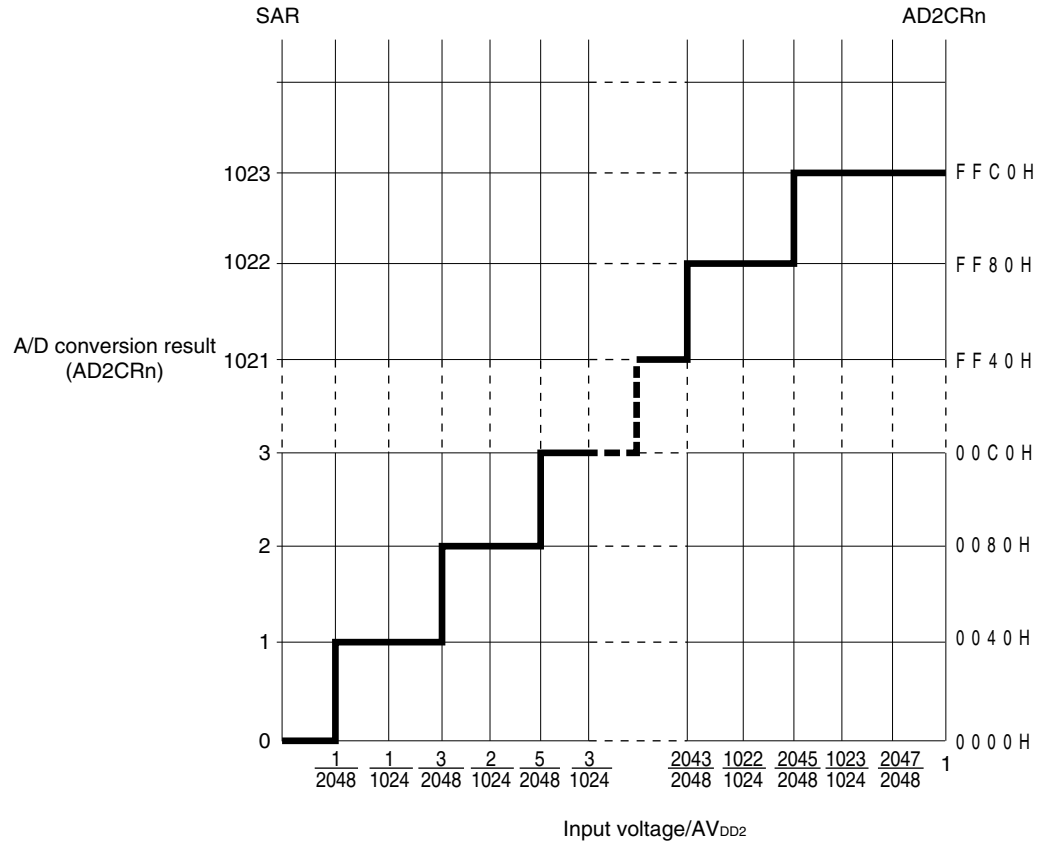
$$(SAR - 0.5) \times \frac{AV_{DD2}}{1{,}024} \leq V_{IN} < (SAR + 0.5) \times \frac{AV_{DD2}}{1{,}024}$$

INT( ): Function that returns the integer of the value in ( )
V_{IN}: Analog input voltage
AV_{DD2}: AV_{DD2} pin voltage
ADCR: Value of A/D2 conversion result register n (AD2CRn)

Note The lower 6 bits of the AD2CRn register are fixed to 0.

The relationship between the analog input voltage and the A/D conversion results is shown in Figure 13-2.

Figure 13-2. Relationship Between Analog Input Voltage and A/D Conversion Results



Remark V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

13.4 Operation

13.4.1 Basic operation

<1> Set the AD2M0.AD2PS bit to 1 to turn on A/D power while the AD2M0.AD2CE bit = 0.
At this time, bits other than the AD2M0.AD2CE bit can be simultaneously set.

<2> Select an operation mode of A/D conversion and A/D conversion time by using the AD2M0, AD2M1, and AD2S registers.

<3> Setting the AD2M0.AD2CE bit to 1 (enable conversion) at least 2 μs after turning on A/D power (AD2M0.AD2PS bit = 0 $\rightarrow$ 1) starts A/D conversion.
If the AD2CE bit is set to 1 before 2 μs passes, the conversion operation is started and ends after A/D conversion time, but the conversion result is undefined.

<4> When A/D conversion is started, the voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<5> When sampling has been performed for a specific time, the sample & hold circuit enters the hold status, and holds the input analog voltage until A/D conversion ends.

<6> Set bit 9 of the successive approximation register (SAR) and changes the level of the voltage tap of the D/A converter to the reference voltage ($1/2AV_{DD2}$).

<7> The voltage generated by the voltage tap of the D/A converter is compared with the analog input voltage by a voltage comparator. If the analog input voltage is found to be greater than ($1/2AV_{DD2}$) as a result of comparison, the MSB of the SAR register remains set. If the analog input voltage is less than ($1/2AV_{DD2}$), the MSB is reset.

<8> Next, bit 8 of the SAR register is automatically set, and the next comparison is started. The voltage tap of the D/A converter is selected according to the value of bit 9, to which the result has been already set as shown below.

Bit 9 = 1: ($3/4AV_{DD2}$)
Bit 9 = 0: ($1/4AV_{DD2}$)

The voltage tap of the D/A converter and the analog input voltage are compared and bit 8 of the SAR register is manipulated according to the result of the comparison as shown below.

Analog input voltage $\geq$ Voltage tap of D/A converter: Bit 8 = 1
Analog input voltage $\leq$ Voltage tap of D/A converter: Bit 8 = 0

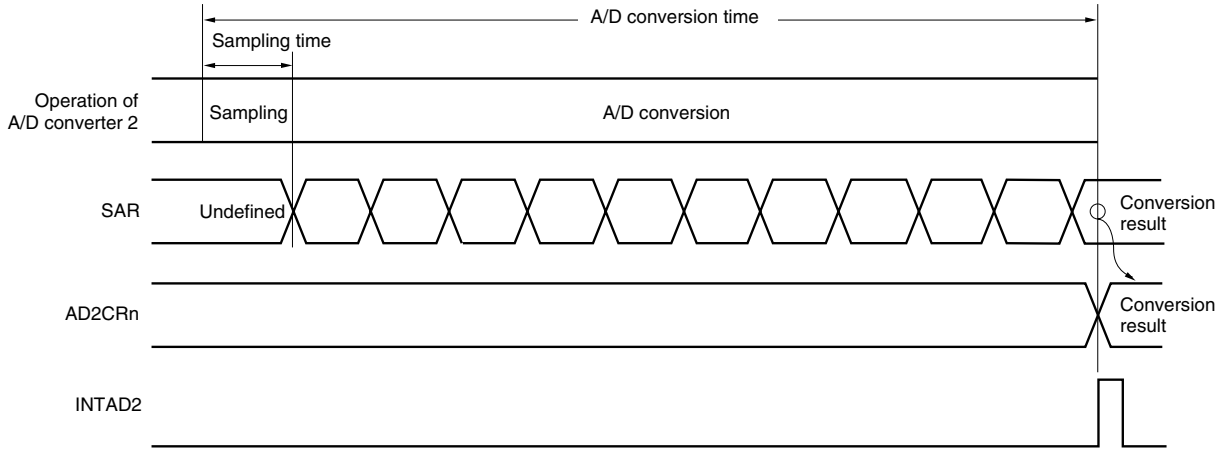
Comparison is continued like this to bit 0 of the SAR register.

<9> When comparison of 10 bits has been completed, the valid digital result remains in the SAR register. This value is transferred to the AD2CRn register and the conversion result is stored in this register (V850E/IF3: n = 0 to 3, V850E/IG3: n = 0 to 7). An A/D2 conversion end interrupt request signal (INTAD2) is generated simultaneously in the select mode and when all the specified A/D conversion operations are completed in the scan mode.

<10> In the continuous select mode or continuous scan mode, <4> to <9> are repeated unless the AD2CE bit is set to 0 after completion of A/D conversion.

In the one-shot select mode or one-shot scan mode, the conversion operation is stopped after it is completed (at this time, the AD2M0.AD2CE bit holds 1 and is not automatically cleared). Write 1 to the AD2CE bit to start conversion operation again.

Figure 13-3. Basic Operation of A/D Converter 2



Remark V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

13.4.2 Trigger mode

Trigger mode that serve as the start timing of an A/D conversion operation is software trigger mode.

This mode is set by the AD2M0 register.

(1) Software trigger mode

In this mode, the analog input pin (ANI2n) specified by the AD2S.AD2S2 to AD2S.AD2S0 bits is used for the A/D conversion start timing by setting the AD2M0.AD2CE bit to 1.

After A/D conversion ends, the conversion result is stored in A/D2 conversion result register n (AD2CRn). An A/D2 conversion end interrupt request signal (INTAD2) is generated simultaneously in the select mode and when all the specified A/D conversion operations are completed in the scan mode.

If the operation mode set by the AD2M0.AD2MD1 and AD2M0.AD2MD0 bits is the continuous select mode or continuous scan mode, the conversion operation is repeated unless the AD2M0.AD2CE bit is set to 0. In the one-shot select mode or one-shot scan mode, the conversion operation is stopped after A/D conversion ends.

The AD2M0.AD2EF bit is set to 1 (conversion in progress) when A/D conversion is started, and set to 0 (conversion stops) when it is completed.

If the AD2M0 and AD2S registers are written during A/D conversion, the conversion is stopped and executed again from the beginning.

Remark V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

13.4.3 Operation mode

There are four operation modes to which the ANI2n pin is set: continuous select mode, continuous scan mode, one-shot select mode, and one-shot scan mode. These modes are set by the AD2M0.AD2MD1 and AD2M0.AD2MD0 registers.

The relationship between the AD2M0, AD2M1, and AD2S registers and operation mode is shown below.

Trigger Mode	Operation Mode	Set Value		
		AD2M0	AD2M1	AD2S
Software trigger	Continuous select	X100000XB	0000XXXXB	00000XXXB
	Continuous scan	X101000XB	0000XXXXB	00000XXXB
	One-shot select	X110000XB	0000XXXXB	00000XXXB
	One-shot scan	X111000XB	0000XXXXB	00000XXXB

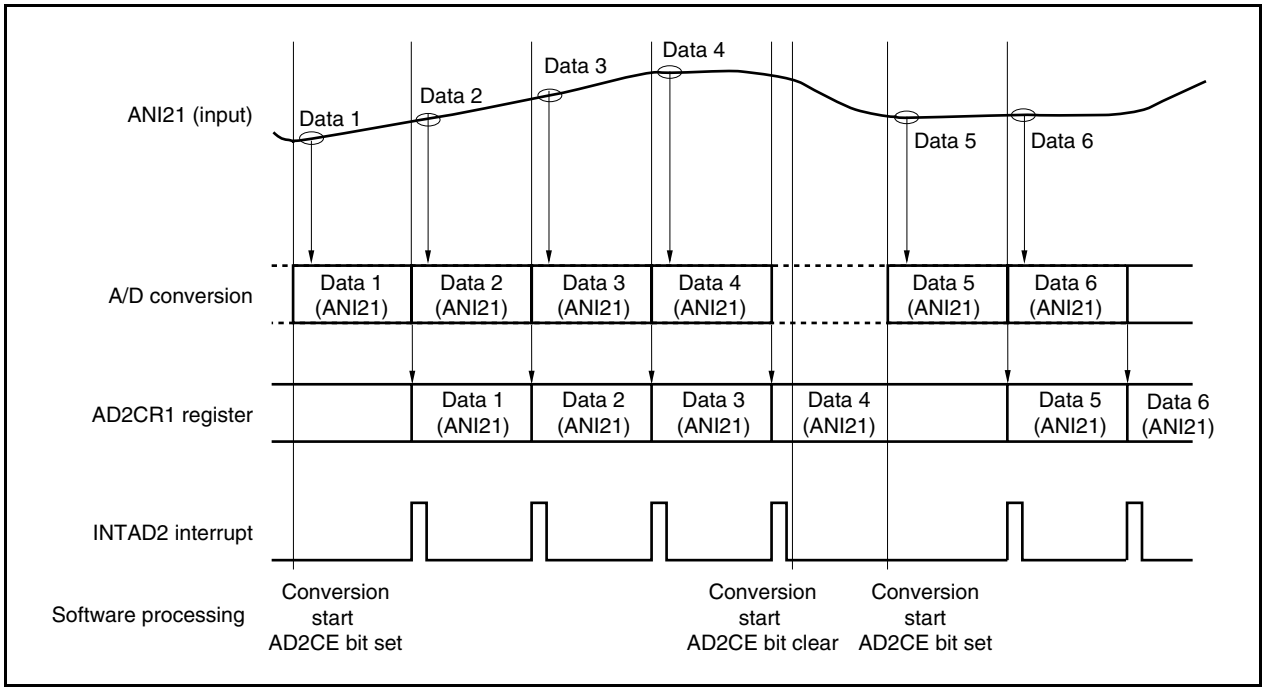
(1) Continuous select mode

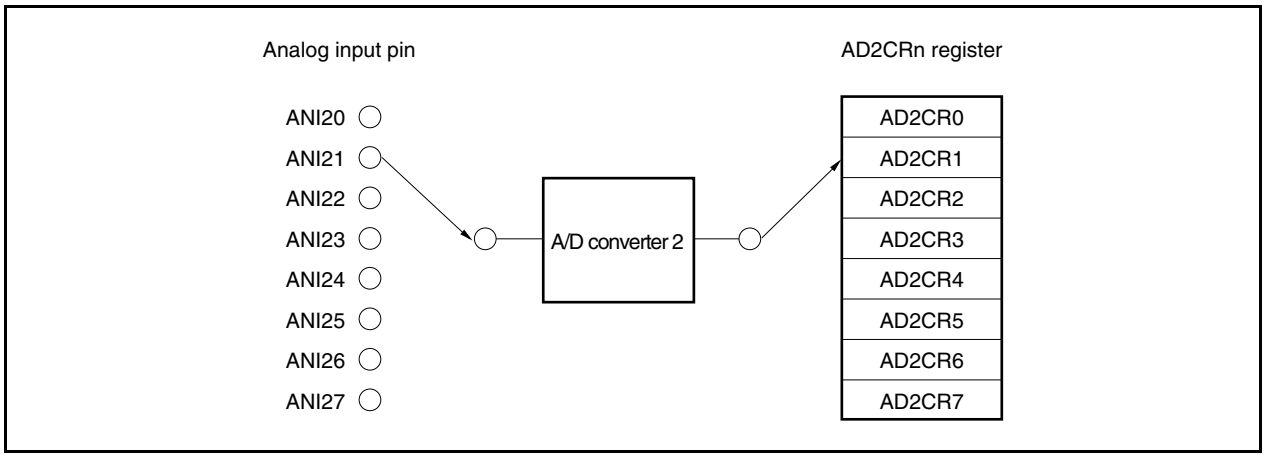
In this mode, the analog input pin (ANI2n) specified by the AD2S register is A/D-converted continuously. The conversion results are stored in the AD2CRn register corresponding to the ANI2n pin. The ANI2n pin and the AD2CRn register correspond one to one, and an A/D2 conversion end interrupt request signal (INTAD2) is generated each time one A/D conversion ends.

After A/D conversion ends, the conversion is repeated again unless the AD2M0.AD2CE bit is set to 0.

Remark V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

Figure 13-4. Continuous Select Mode Operation Timing
(When AD2M0.AD2MD1 and AD2M0.AD2MD0 Bits = 00, AD2S.AD2S2 to AD2S.AD2S0 Bits = 001): V850E/IG3



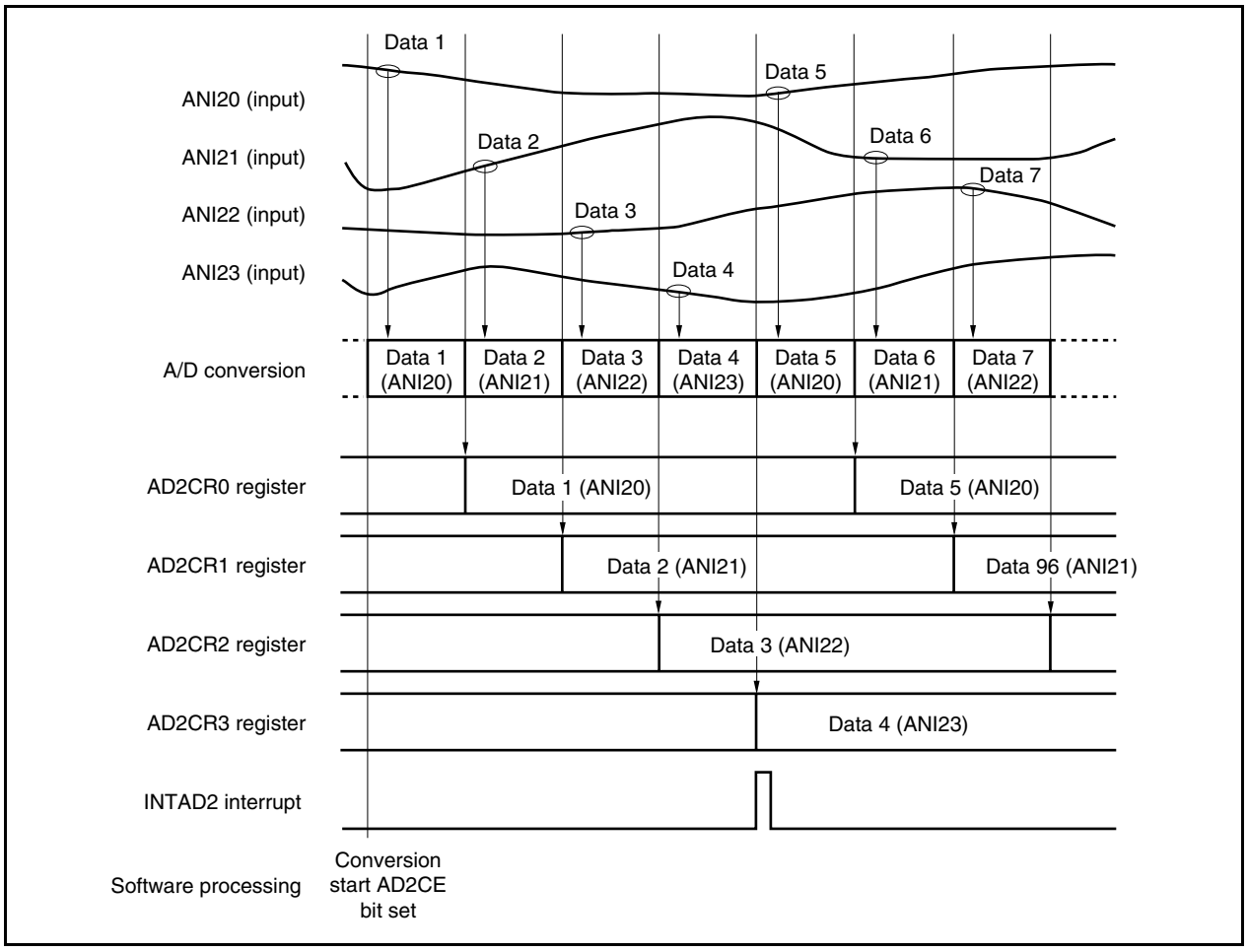


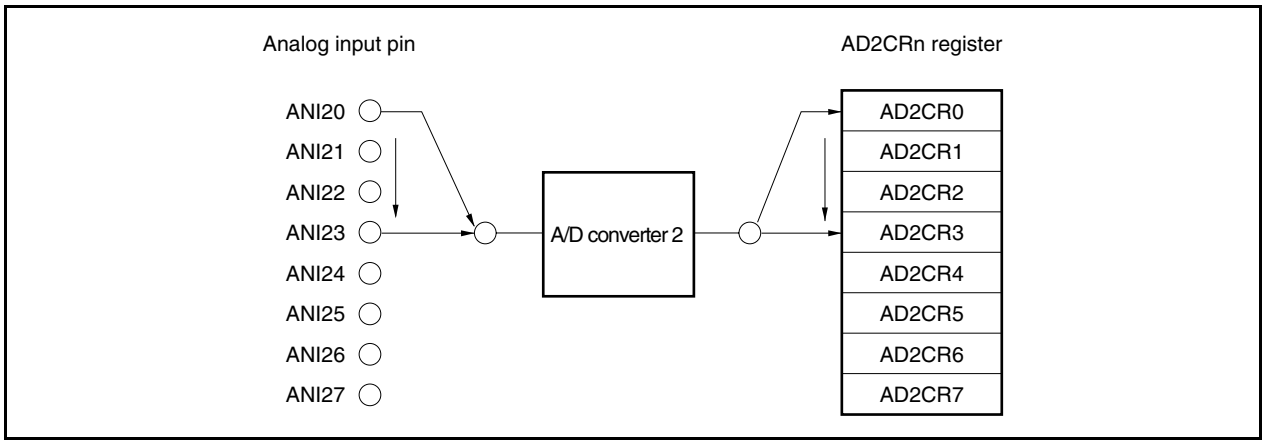
(2) Continuous scan mode

In this mode, the analog input pin (ANI2n) specified by the AD2S register is selected sequentially from the ANI20 pin, and A/D conversion is executed continuously. The A/D conversion results are stored in the AD2CRn register corresponding to the analog input pin. When conversion of all the specified analog input pin ends, the A/D2 conversion end interrupt request signal (INTAD2) is generated. After A/D conversion ends, the conversion is started again from the ANI20 pin, unless the AD2M0.AD2CE bit is set to 0.

Remark V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

Figure 13-5. Continuous Scan Mode Operation Timing
(When AD2M0.AD2MD1 and AD2M0.AD2MD0 Bits = 01, AD2S.AD2S2 to AD2S.AD2S0 Bits = 011): V850E/IG3





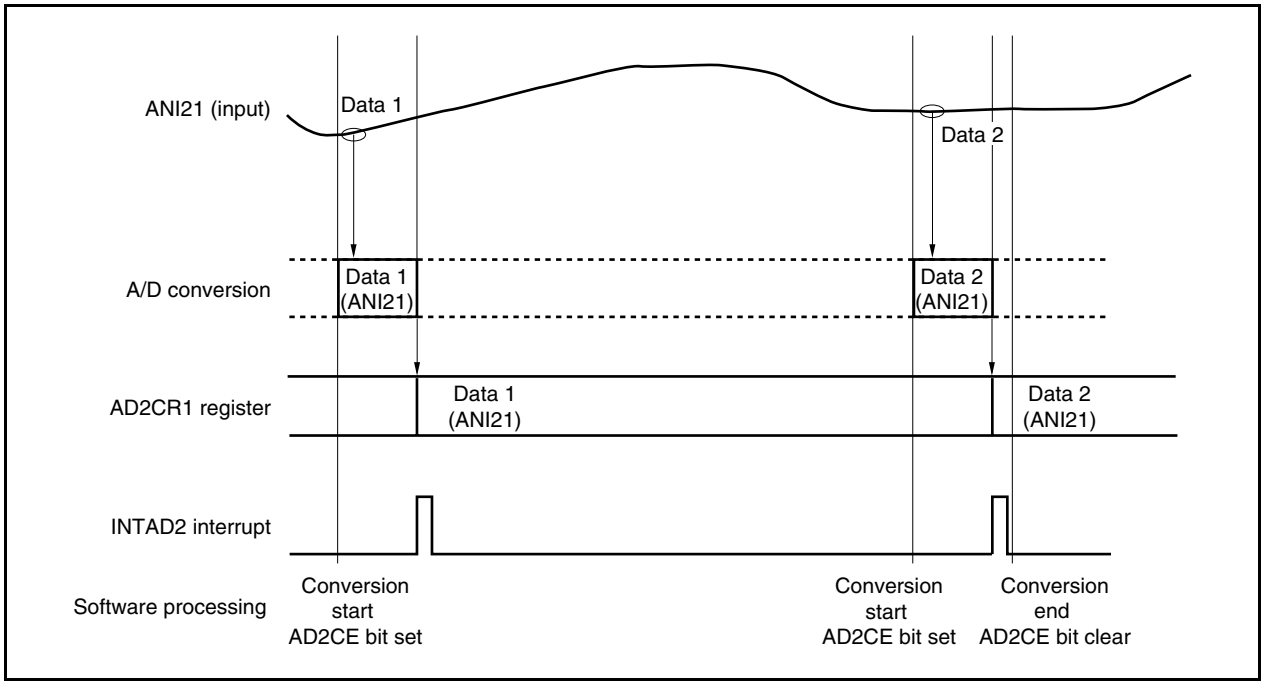
(3) One-shot select mode

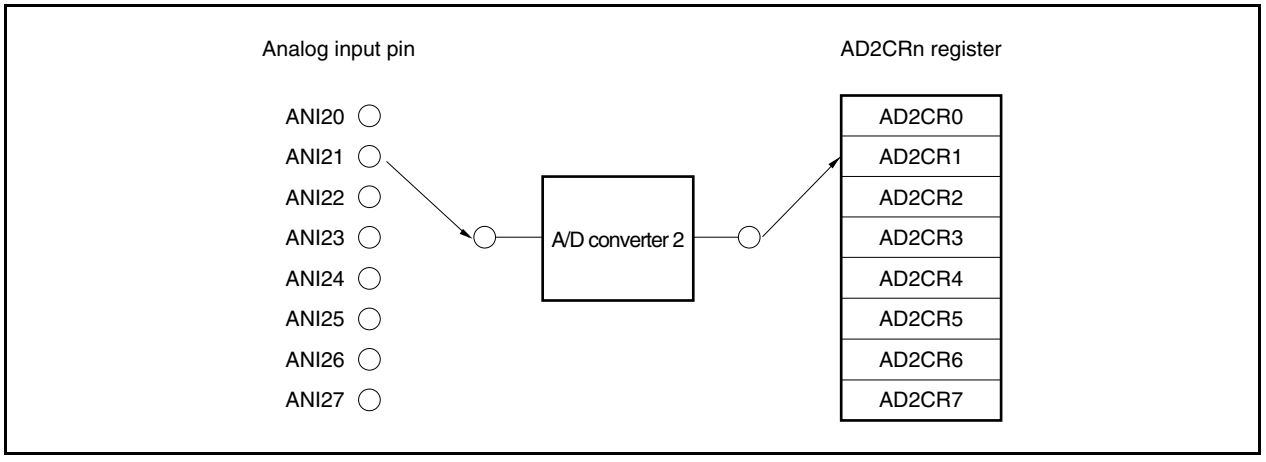
In this mode, the analog input pin (ANI2n) specified by the AD2S register is A/D-converted once. The conversion result is stored in the AD2CRn register corresponding to the ANI2n pin. The ANI2n pin and the AD2CRn register correspond one to one, and an A/D2 conversion end interrupt request signal (INTAD2) is generated each time one A/D conversion ends.

After A/D conversion ends, the conversion operation is stopped.

Remark V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

Figure 13-6. One-Shot Select Mode Operation Timing
(When AD2M0.AD2MD1 and AD2M0.AD2MD0 Bits = 10, AD2S.AD2S2 to AD2S.AD2S0 Bits = 001): V850E/IG3



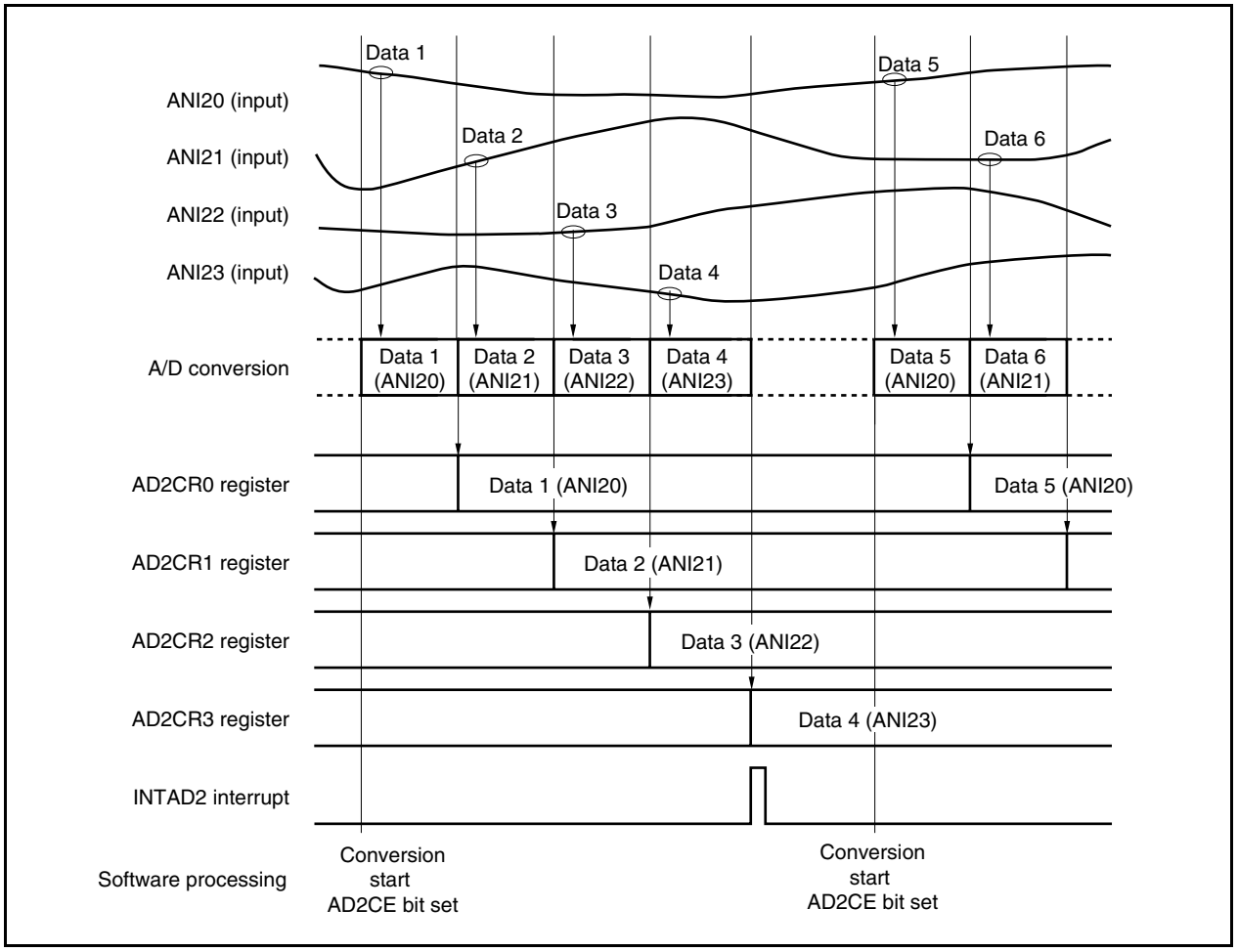


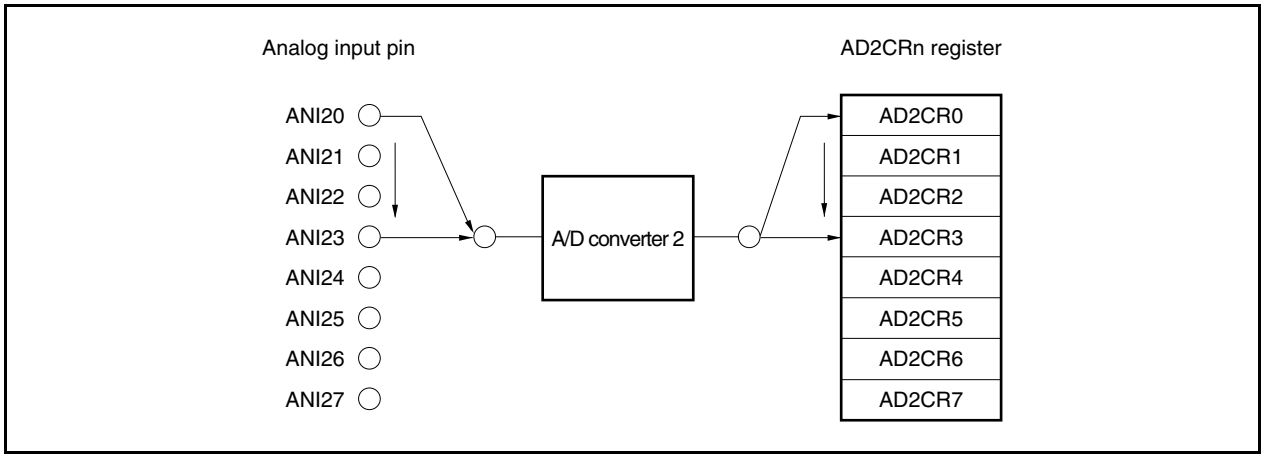
(4) One-shot scan mode

In this mode, pins up to the analog input pin (ANI2n) specified by the AD2S register from the ANI20 pin are selected sequentially, and A/D conversion is executed. The A/D conversion results are stored in the AD2CRn register corresponding to the analog input pin. When conversion of all the specified analog input pins ends, the A/D2 conversion end interrupt request signal (INTAD2) is generated. After A/D conversion ends, the conversion operation is stopped.

Remark V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

Figure 13-7. One-Shot Scan Mode Operation Timing
(When AD2M0.AD2MD1 and AD2M0.AD2MD0 Bits = 11, AD2S.AD2S2 to AD2S.AD2S0 Bits = 011): V850E/IG3





13.5 Operation in Software Trigger Mode

When the AD2M0.AD2CE bit is set to 1, A/D conversion is started.

When A/D conversion is started, the AD2M0.AD2EF bit = 1 (conversion in progress).

If the AD2M0 and AD2S registers are written during A/D conversion, the conversion is stopped and executed again from the beginning.

(1) Operation in software trigger continuous select mode

In this mode, one analog input pin (ANI2n) specified by the AD2S register is A/D-converted once. The conversion results are stored in one AD2CRn register. The ANI2n pin and AD2CRn register correspond one to one.

Each time an A/D conversion is executed, an A/D2 conversion end interrupt request signal (INTAD2) is generated and A/D conversion ends. After A/D conversion ends, the conversion is repeated again unless the AD2M0.AD2CE bit is set to 0.

It is not necessary to set (1) the AD2M0.AD2CE bit to restart A/D conversion[Note].

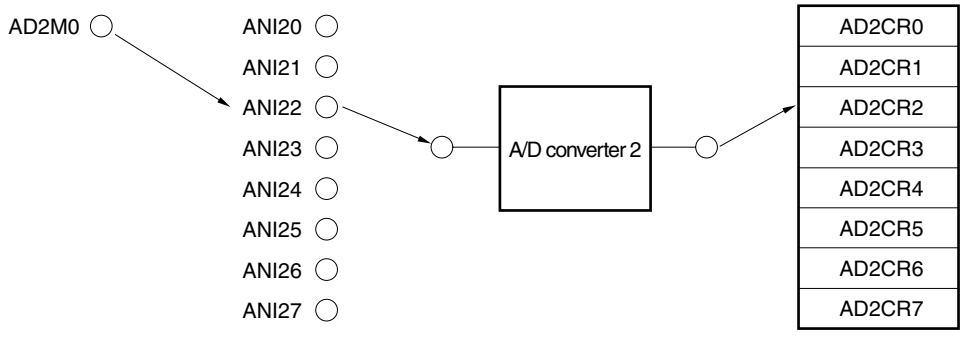
Note In the software trigger continuous select mode, the A/D conversion operation is not stopped unless the AD2M0.AD2CE bit is set to 0. If the AD2CRn register is not read before the next A/D conversion ends, it is overwritten.

This mode is suitable for applications in which the A/D conversion value of one analog input pin is read.

Analog Input Pin	A/D Conversion Result Register
ANI2n	AD2CRn

Remark V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

Figure 13-8. Operation Example of Software Trigger Continuous Select Mode: V850E/IG3



(1) The AD2CE bit = 1 (enable)
(2) The ANI22 pin is A/D-converted
(3) The conversion result is stored in the AD2CR2 register
(4) The AD2M0.AD2EF bit = 0
(5) The INTAD2 interrupt request signal is generated
(6) Return to (2)
(7) To end the conversion, the AD2CE bit = 0 (stop)

Remark This is an operation example with the following setting.
AD2M0.AD2MD1 and AD2M0.AD2MD0 bits = 00, AD2S.AD2S2 to AD2S.AD2S0 bits = 010

(2) Software trigger continuous scan mode operations

In this mode, pins up to the analog input pin (ANI2n) specified by the AD2S register from the ANI20 pin are selected sequentially, and A/D conversion is executed continuously. The A/D conversion results are stored in the AD2CRn register corresponding to the analog input pin.

When conversion of all the specified analog input pins ends, the A/D2 conversion end interrupt request signal (INTAD2) is generated. After A/D conversion ends, the conversion is started again from the ANI20 pin, unless the AD2M0.AD2CE bit is set to 0.

It is not necessary to set (1) the AD2M0.AD2CE bit to restart A/D conversion[Note].

Note In the software trigger continuous scan mode, the A/D conversion operation is not stopped unless the AD2M0.AD2CE bit is set to 0. If the AD2CRn register is not read before the next A/D conversion ends, it is overwritten.

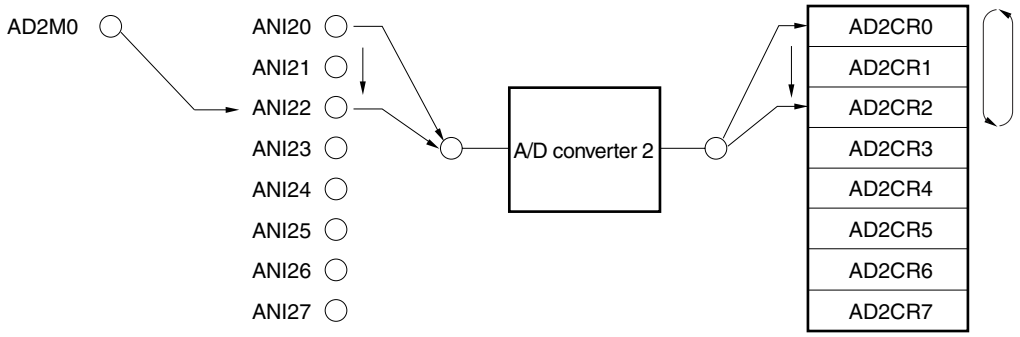
This mode is suitable for applications in which multiple analog inputs are constantly monitored.

Analog Input Pin	A/D Conversion Result Register
ANI20	AD2CR0
⋮	⋮
ANI2n[Note]	AD2CRn

Note Set by the AD2S.AD2S0 to AD2S.AD2S2 bits.

Remark V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

Figure 13-9. Operation Example of Software Trigger Continuous Scan Mode: V850E/IG3



(1) The AD2CE bit = 1 (enable)
(2) The ANI20 pin is A/D-converted
(3) The conversion result is stored in the AD2CR0 register
(4) The ANI21 pin is A/D-converted
(5) The conversion result is stored in the AD2CR1 register
(6) The ANI22 pin is A/D-converted
(7) The conversion result is stored in the AD2CR2 register
(8) The INTAD2 interrupt request signal is generated
(9) Return to (2)
(10) To end the conversion, the AD2CE bit = 0 (stop)

Remark This is an operation example with the following setting.
AD2M0.AD2MD1 and AD2M0.AD2MD0 bits = 01, AD2S.AD2S2 to AD2S.AD2S0 bits = 010

(3) Software trigger one-shot select mode

In this mode, the voltage of one analog input pin (ANI2n) specified by the AD2S register is A/D-converted once. The conversion result is stored in one AD2CRn register. The ANI2n pin and the AD2CRn register correspond one to one.

Each time an A/D conversion is executed, an A/D2 conversion end interrupt request signal (INTAD2) is generated and A/D conversion ends. After A/D conversion ends, the conversion operation is stopped.

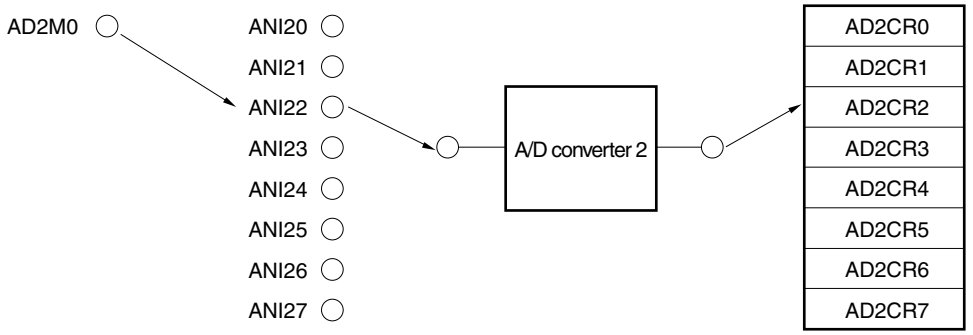
If the AD2M0.AD2CE bit is set to 1, A/D conversion can be restarted.

This mode is suitable for applications in which the results of each first-time A/D conversion are read.

Analog Input Pin	A/D Conversion Result Register
ANI2n	AD2CRn

Remark V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

Figure 13-10. Operation Example of Software Trigger One-Shot Select Mode: V850E/IG3



(1) The AD2CE bit = 1 (enable)
(2) The ANI22 pin is A/D-converted
(3) The conversion result is stored in the AD2CR2 register
(4) The AD2M0.AD2EF bit = 0
(5) The INTAD2 interrupt request signal is generated

Remark This is an operation example with the following setting.
AD2M0.AD2MD1 and AD2M0.AD2MD0 bits = 10, AD2S.AD2S2 to AD2S.AD2S0 bits = 010

(4) Software trigger one-shot scan mode operations

In this mode, pins up to the analog input pin (ANI2n) specified by the AD2S register from the ANI20 pin are selected sequentially, and A/D conversion is executed continuously. The A/D conversion results are stored in the AD2CRn register corresponding to the analog input pin.

When conversion of all the specified analog input pin ends, the A/D2 conversion end interrupt request signal (INTAD2) is generated. After A/D conversion ends, the conversion operation is stopped.

If the AD2M0.AD2CE bit is set to 1, A/D conversion can be restarted.

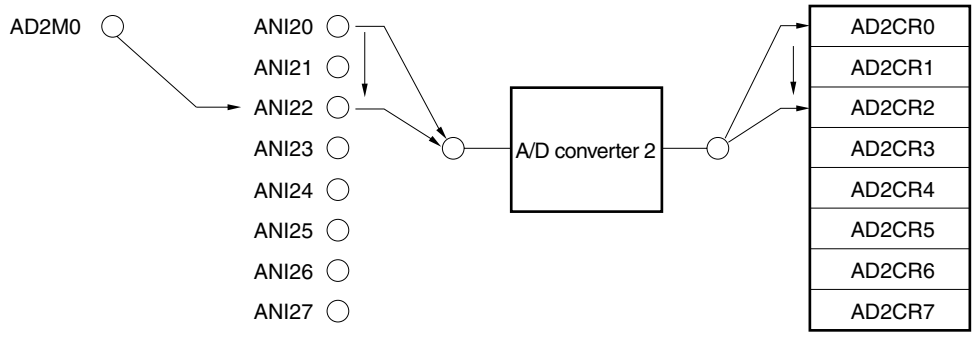
This mode is suitable for applications in which multiple analog inputs are constantly monitored.

Analog Input Pin	A/D Conversion Result Register
ANI20	AD2CR0
⋮	⋮
ANI2n[Note]	AD2CRn

Note Set by the AD2S.AD2S0 to AD2S.AD2S2 bits.

Remark V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

Figure 13-11. Operation Example of Software Trigger One-Shot Scan Mode: V850E/IG3

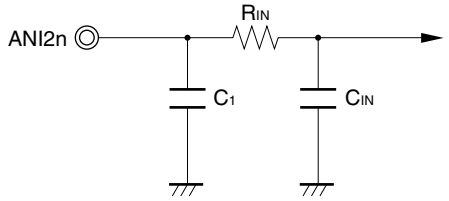


(1) The AD2CE bit = 1 (enable)
(2) The ANI20 pin is A/D-converted
(3) The conversion result is stored in the AD2CR0 register
(4) The ANI21 pin is A/D-converted
(5) The conversion result is stored in the AD2CR1 register
(6) The ANI22 pin is A/D-converted
(7) The conversion result is stored in the AD2CR2 register
(8) The AD2M0.AD2EF bit = 0
(9) The INTAD2 interrupt request signal is generated

Remark This is an operation example with the following setting.
AD2M0.AD2MD1 and AD2M0.AD2MD0 bits = 11, AD2S.AD2S2 to AD2S.AD2S0 bits = 010

13.6 Internal Equivalent Circuit

The following figure shows the equivalent circuit of the analog input block.



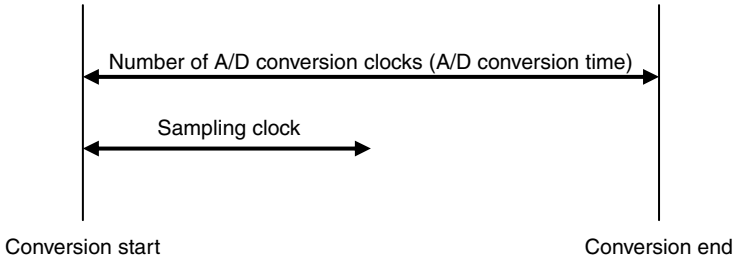
R	C_1	C_2
2.6 kΩ	15 pF	6.2 pF

Remarks 1. The maximum values are shown (reference values).

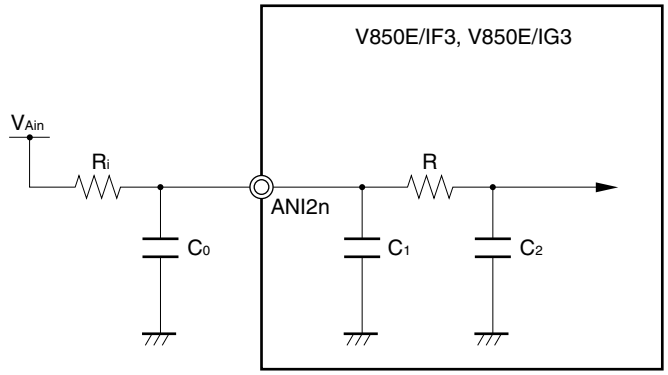
2. V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

AD2M1 register				Number of A/D conversion clocks (f_{AD2})	Number of sampling clocks
AD2FR3 bit	AD2FR2 bit	AD2FR1 bit	AD2FR0 bit		
0	0	0	1	62	33
0	0	1	0	93	49.5
0	0	1	1	124	66
0	1	0	0	155	82.5
0	1	0	1	186	99
0	1	1	0	217	115.5
0	1	1	1	248	132
1	0	0	0	279	148.5
1	0	0	1	310	165

Caution Number of sampling clocks is included in number of A/D conversion clocks.



An example of calculating an overall error of A/D converter 2 is shown below.



fXX (MHz)	A/D conversion time (μs)	Sampling (μs)	R (kΩ)	C1 (pF)	C2 (pF)	C0 (pF)	Ri (kΩ)	Sampling error (LSB)[Note]
64	3.88 (62/fAD2)	1.03 (33/fAD2)	2.6	15	6.2	100	1.0	0.1 or lower
						100	0.5	0.1 or lower
						100	0.25	0.1 or lower
						100	0.125	0.1 or lower
						50	1.0	0.1 or lower
						50	0.5	0.1 or lower
						50	0.25	0.1 or lower
						50	0.125	0.1 or lower

Note The error when considering the signal source impedance is “sampling error + overall error”.

Remarks 1. These values are reference values calculated by simulating what happens to C_2 voltage by R_i and C_0 when V_{Ain} is applied from 0 V to 5 V at the same time as sampling start.

2. V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

3. f_{XX}: System clock frequency
f_{AD2}: Operating clock frequency

13.7 Cautions

(1) When A/D converter is not used

When the A/D converter is not used, the power consumption can be reduced by clearing the AD2M0.AD2CE and AD2M0.AD2PS bits to 0.

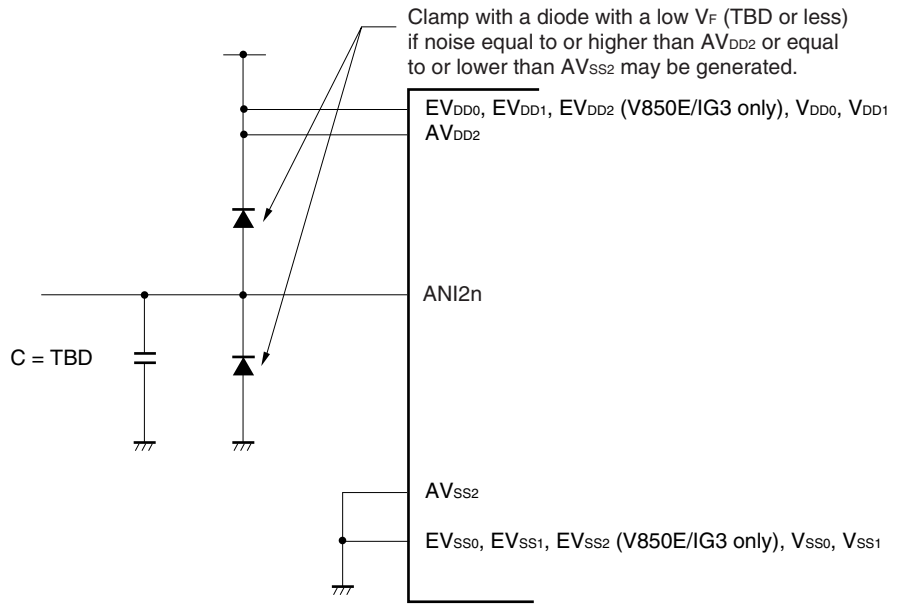
(2) Input range of ANI2n pin

Input the voltage within the specified range to the ANI2n pin. If a voltage equal to or higher than AV_{DD2} or equal to or lower than AV_{SS2} (even within the range of the absolute maximum ratings) is input to this pin, the conversion value of that channel is undefined, and the conversion value of the other channels may also be affected.

(3) Countermeasures against noise

To maintain the 10-bit resolution, the ANI2n pin must be effectively protected from noise. The influence of noise increases as the output impedance of the analog input source becomes higher. To lower the noise, connecting an external capacitor as shown in Figure 13-12 is recommended.

Figure 13-12. Processing of Analog Input Pin



Remark V850E/IF3: n = 0 to 3
V850E/IG3: n = 0 to 7

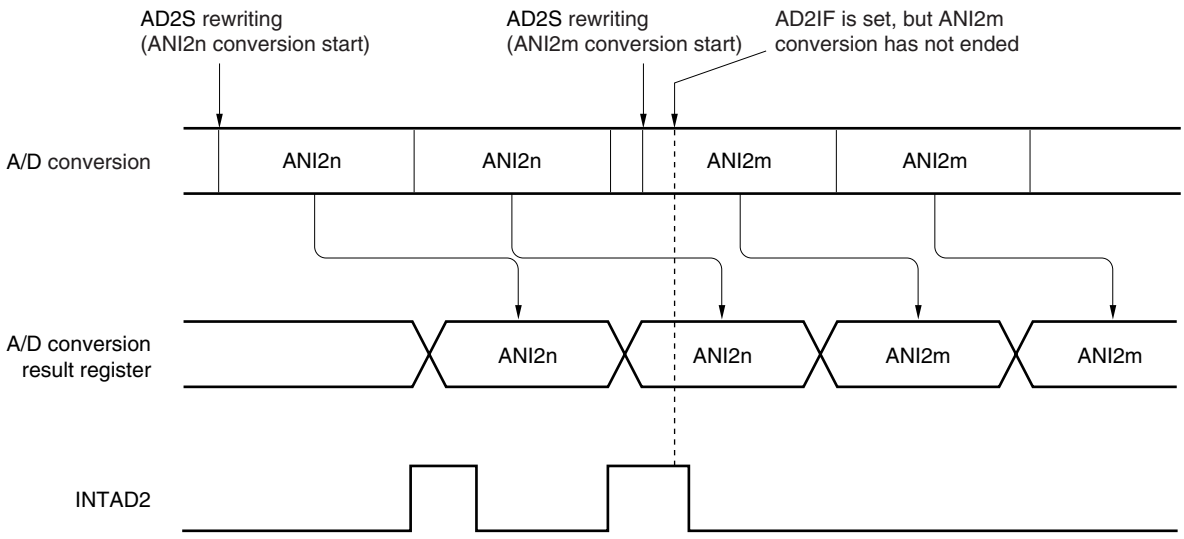
(4) Alternate input

The analog input pin (ANI2n) functions alternately as input port (P7n). When selecting one of the ANI2n pin to execute A/D conversion, do not execute an input instruction to port 7 during conversion as the conversion resolution may drop.

(5) Interrupt request flag (AD2IF)

The interrupt request flag (AD2IF) is not cleared even if the contents of the AD2S register are changed. If the analog input pin is changed during A/D conversion, therefore, the result of converting the previously selected analog input signal may be stored and the A/D2 conversion end interrupt request flag may be set immediately before the AD2S register is rewritten. If the AD2IF flag is read immediately after the AD2S register is rewritten, the AD2IF flag may be set even though the A/D conversion of the newly selected analog input pin has not been completed. When A/D conversion is stopped, clear the AD2IF flag before resuming conversion.

Figure 13-13. Generation Timing of A/D2 Conversion End Interrupt Request



Remark V850E/IF3: n = 0 to 3, m= 0 to 3
V850E/IG3: n = 0 to 7, m = 0 to 7

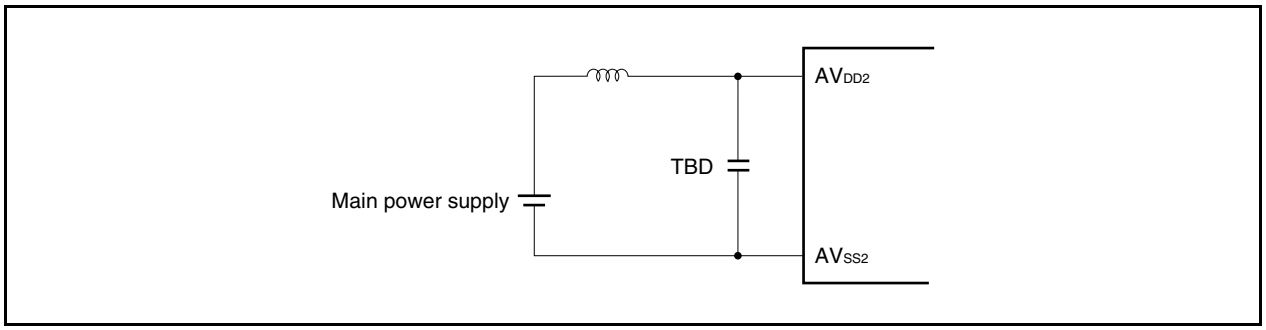
(6) AV_{DD2} pin

(a) The AV_{DD2} pin is used as the power supply pin of the A/D converter 2 and also supplies power to the alternate-function ports. In an application where a backup power supply is used, be sure to supply the same potential as EV_{DD0}, EV_{DD1}, and EV_{DD2} (V850E/IG3 only) to the AV_{DD2} pin as shown in Figure 13-12.

(b) The AV_{DD2} pin is also used as the reference voltage pin of the A/D converter 2. If the source supplying power to the AV_{DD2} pin has a high impedance or if the power supply has a low current supply capability, the reference voltage may fluctuate due to the current that flows during conversion (especially, immediately after the conversion operation enable (AD2CE bit = 1)). As a result, the conversion accuracy may drop. To avoid this, it is recommended to connect a capacitor across the AV_{DD2} and AV_{SS2} pins to suppress the reference voltage fluctuation as shown in Figure 13-14.

(c) If the source supplying power to the AV_{DD2} pin has a high DC resistance (for example, because of insertion of a diode), the voltage when conversion is enabled may be lower than the voltage when conversion is stopped, because of a voltage drop caused by the A/D conversion current.

Figure 13-14. AV_{DD2} Pin Connection Example



(7) Reading AD2CRn register

When the AD2M0, AD2M1, or AD2S register is written, the contents of the AD2CRn register may be undefined. Read the conversion result after completion of conversion and before writing to the AD2M0, AD2M1, and AD2S registers. The correct conversion result may not be read at a timing different from the above.

(8) A/D conversion result

If there is noise at the analog input pin (ANI2n) or at the power supply voltage pin (AV_{DD2}), that noise may generate an illegal conversion result.

Software processing will be needed to avoid a negative effect on the system from this illegal conversion result. An example of this software processing is shown below.

- Take the average result of a number of A/D conversions and use that as the A/D conversion result.
- Execute a number of A/D conversions successively and use those results, omitting any exceptional results that may have been obtained.
- If an A/D conversion result that is judged to have generated a system malfunction is obtained, be sure to recheck the system malfunction before performing counteractive measures.

(9) Standby mode

Because the A/D converter 2 stops operating in the IDLE and STOP modes, conversion results are invalid, so power consumption can be reduced. Operations are resumed after the IDLE and STOP modes are released, but the A/D conversion results after the IDLE and STOP modes are released are invalid. When using the A/D converter 2 after the IDLE and STOP modes are released, before setting the IDLE and STOP modes or releasing the IDLE and STOP modes, set the AD2M0.AD2CE bit to 0 then set the AD2CE bit to 1 after releasing the IDLE and STOP modes.

(10) Variation of A/D conversion results

The results of the A/D conversion may vary depending on the fluctuation of the supply voltage, or may be affected by noise. To reduce the variation, take counteractive measures with the program, such as by averaging the A/D conversion results.

(11) A/D conversion result hysteresis characteristics

Successive comparison type A/D converters hold an analog input voltage in an internal sample & hold capacitor and then perform A/D conversion. After the A/D conversion has finished, the analog input voltage remains in the internal sample & hold capacitor. As a result, the following phenomena may occur.

- When the same channel is used for A/D conversions, if the voltage is higher or lower than the previous A/D conversion, then hysteresis characteristics may appear where the conversion result is affected by the previous value. Even if the conversion were to be performed at the same potential, the results may thus vary.
- When switching the analog input channel, hysteresis characteristics may appear where the conversion result is affected by the previous channel value. This is because one A/D converter is used for the A/D conversions. Even if the conversion were to be performed at the same potential, the results may thus vary.

To obtain more accurate conversion results, execute A/D conversion twice successively on the same channel, and discard the first conversion result.

13.8 How to Read A/D Converter Characteristics Table

For details about the A/D converter characteristics table, see **12.7 How to Read A/D Converter Characteristics Table**.

14.1 Mode Switching Between UARTA and Other Serial Interface

14.1.1 Mode switching between UARTA0 and CSIB0

In the V850E/IF3 and V850E/IG3, UARTA0 and CSIB0 function alternately, and these pins cannot be used at the same time. To switch between UARTA0 and CSIB0, the PMC4, PFC4, and PFCE4 registers must be set in advance.

Caution The operations related to transmission and reception of UARTA0 or CSIB0 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 14-1. Mode Switch Settings of UARTA0 and CSIB0

After reset: 00H R/W Address: FFFFF448H

	7	6	5	4	3	2	1	0
PMC4	PMC47	PMC46	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40

After reset: 00H R/W Address: FFFFF468H

	7	6	5	4	3	2	1	0
PFC4	PFC47	PFC46	PFC45	PFC44	PFC43	PFC42	PFC41	PFC40

After reset: 00H R/W Address: FFFFF708H

	7	6	5	4	3	2	1	0
PFCE4	PFCE47	PFCE46	PFCE45	PFCE44	PFCE43	PFCE42	PFCE41	PFCE40

PMC42	PFCE42	PFC42	Specification of alternate function of P42 pin
0	×	×	I/O port
1	0	0	$\overline{\text{SCKB0}}$ I/O
1	0	1	INTP13 input
1	1	0	Setting prohibited
1	1	1	Setting prohibited

PMC41	PFCE41	PFC41	Specification of alternate function of P41 pin
0	×	×	I/O port
1	0	0	SOB0 output
1	0	1	TXDA0 output
1	1	0	Setting prohibited
1	1	1	Setting prohibited

PMC40	PFC40	Specification of alternate function of P40 pin
0	×	I/O port
1	0	SIB0 input
1	0	RXDA0 input

Remark × = don't care

14.1.2 Mode switching between UARTA1 and I^2C

In the V850E/IF3 and V850E/IG3, UARTA1 and I^2C function alternately, and their pins cannot be used at the same time. To switch between UARTA1 and I^2C, the PMC3, PFC3, and PFCE3 registers must be set in advance.

Caution The operations related to transmission and reception of UARTA1 or I^2C are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 14-2. Mode Switch Settings of UARTA1 and I^2C

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	PFCE37	PFCE36	PFCE35	PFCE34	0	PFCE32	PFCE31	PFCE30

PMC31	PFCE31	PFC31	Specification of alternate function of P31 pin
0	×	×	I/O port
1	0	0	TXDA1 output
1	0	1	SDA I/O
1	1	0	Setting prohibited
1	1	1	Setting prohibited

PMC30	PFCE30	PFC30	Specification of alternate function of P30 pin
0	×	×	I/O port
1	0	0	RXDA1 input
1	0	1	SCL I/O
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Remark x = don't care

14.1.3 Mode switching between UARTA2 and CSIB1

In the V850E/IF3 and V850E/IG3, UARTA2 and CSIB1 function alternately, and their pins cannot be used at the same time. To switch between UARTA2 and CSIB1, the PMC3, PFC3, and PFCE3 registers must be set in advance.

Caution The operations related to transmission and reception of UARTA2 or CSIB1 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 14-3. Mode Switch Settings of UARTA2 and CSIB1

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	PFCE37	PFCE36	PFCE35	PFCE34	0	PFCE32	PFCE31	PFCE30

PMC34	PFCE34	PFC34	Specification of alternate function of P34 pin
0	×	×	I/O port
1	0	0	$\overline{\text{SCKB1}}$ I/O
1	0	1	INTP11 input
1	1	0	$\overline{\text{CS0}}$[Note] output
1	1	1	Setting prohibited

PMC33	PFC33	Specification of alternate function of P33 pin
0	×	I/O port
1	0	SOB1 output
1	0	TXDA2 output

PMC32	PFCE32	PFC32	Specification of alternate function of P32 pin
0	×	×	I/O port
1	0	0	SIB1 input
1	0	1	RXDA2 input
1	1	0	$\overline{\text{CS1}}$[Note] output
1	1	1	Setting prohibited

Note μPD70F3454GC-8EA-A only

Remark x = don't care

14.2 Features

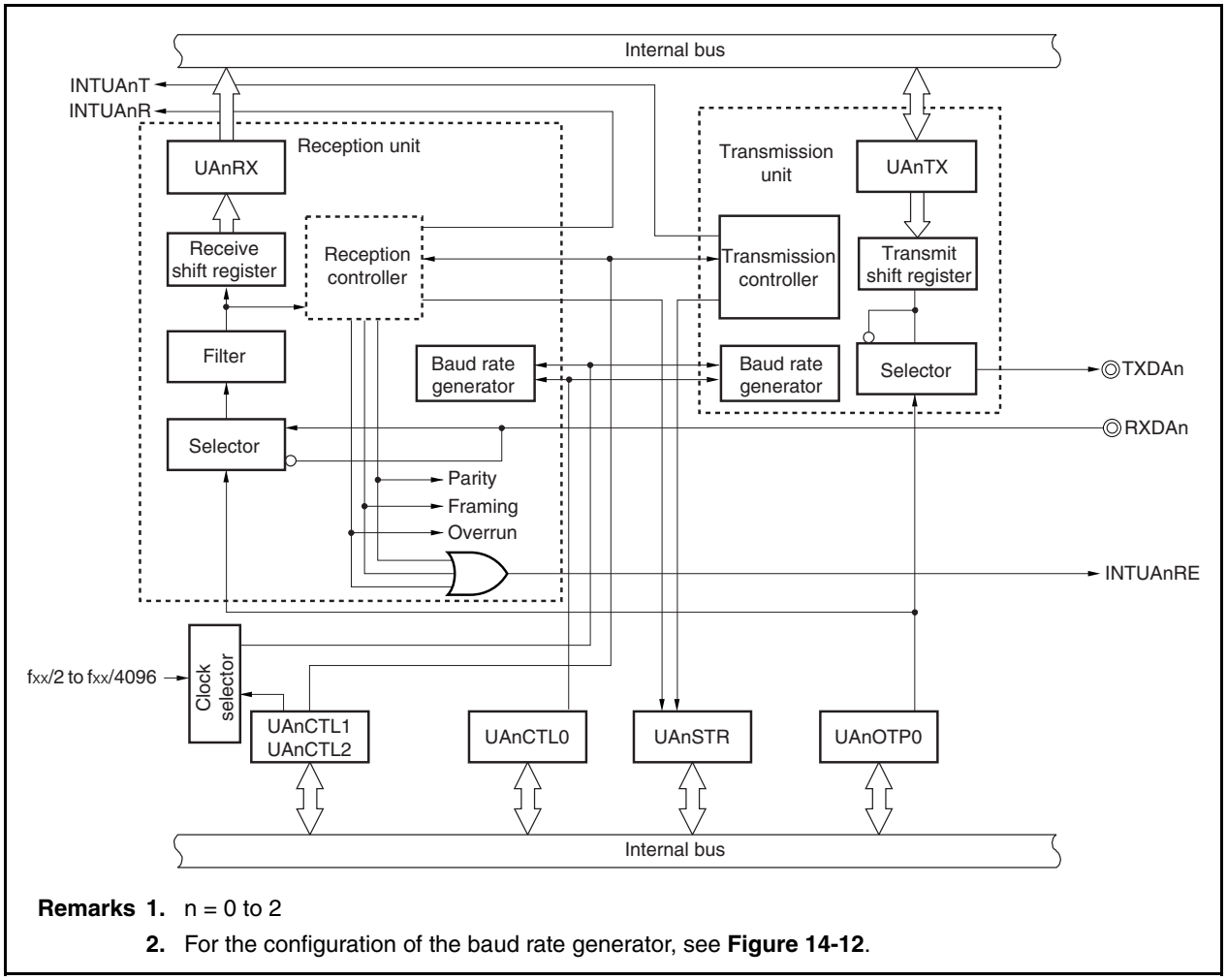
- Transfer rate: 300 bps to 1.25 Mbps (using peripheral clock (fxx) of 64 MHz and dedicated baud rate generator)
- Full-duplex communication: Internal UARTA receive data register n (UAnRX)
  Internal UARTA transmit data register n (UAnTX)
- 2-pin configuration: TXDAn: Transmit data output pin
  RXDAn: Receive data input pin
- Reception error output function
  - Parity error
  - Framing error
  - Overrun error
- Interrupt sources: 3
  - Reception error interrupt (INTUAnRE): This interrupt is generated by ORing the three types of reception errors
  - Reception end interrupt (INTUAnR): This interrupt occurs upon transfer of receive data from the shift register to the UAnRX register after serial transfer end, in the reception enabled status.
  - Transmission enable interrupt (INTUAnT): This interrupt occurs upon transfer of transmit data from the UAnTX register to the shift register in the transmission enabled status.
- Character length: 7, 8 bits
- Parity function: Odd, even, 0, none
- Transmission stop bit: 1, 2 bits
- On-chip dedicated baud rate generator
- MSB-/LSB-first transfer selectable
- Transmit/receive data inverted input/output possible

Remark n = 0 to 2

14.3 Configuration

The block diagram of the UARTAn is shown below.

Figure 14-4. Block Diagram of UARTAn



Remarks 1. n = 0 to 2

2. For the configuration of the baud rate generator, see **Figure 14-12**.

UARTAn consists of the following hardware units.

Table 14-1. Configuration of UARTAn

Item	Configuration
Registers	UARTAn control register 0 (UAnCTL0) UARTAn control register 1 (UAnCTL1) UARTAn control register 2 (UAnCTL2) UARTAn option control register 0 (UAnOPT0) UARTAn status register (UAnSTR) UARTAn receive shift register UARTAn receive data register (UAnRX) UARTAn transmit shift register UARTAn transmit data register (UAnTX)

(1) UARTAn control register 0 (UAnCTL0)

The UAnCTL0 register is an 8-bit register used to specify the UARTAn operation.

(2) UARTAn control register 1 (UAnCTL1)

The UAnCTL1 register is an 8-bit register used to select the base clock (f_{UCLK}) for the UARTAn.

(3) UARTAn control register 2 (UAnCTL2)

The UAnCTL2 register is an 8-bit register used to control the baud rate for the UARTAn.

(4) UARTAn option control register 0 (UAnOPT0)

The UAnOPT0 register is an 8-bit register used to control serial transfer for the UARTAn.

(5) UARTAn status register (UAnSTR)

The UAnSTR register consists of flags indicating the error contents when a reception error occurs. Each one of the reception error flags is set (to 1) upon occurrence of a reception error.

(6) UARTAn receive shift register

This is a shift register used to convert the serial data input to the RXDAn pin into parallel data. Upon reception of 1 byte of data and detection of the stop bit, the receive data is transferred to the UAnRX register.

This register cannot be manipulated directly.

(7) UARTAn receive data register (UAnRX)

The UAnRX register is an 8-bit register that holds receive data. When 7 characters are received, 0 is stored in the highest bit (when data is received LSB first).

In the reception enabled status, receive data is transferred from the UARTAn receive shift register to the UAnRX register in synchronization with the completion of shift-in processing of 1 frame.

Transfer to the UAnRX register also causes the reception end interrupt request signal (INTUAnR) to be output.

(8) UARTAn transmit shift register

The UARTAn transmit shift register is a shift register used to convert the parallel data transferred from the UAnTX register into serial data.

When 1 byte of data is transferred from the UAnTX register, the UARTAn transmit shift register data is output from the TXDAn pin.

This register cannot be manipulated directly.

(9) UARTAn transmit data register (UAnTX)

The UAnTX register is an 8-bit transmit data buffer. Transmission starts when transmit data is written to the UAnTX register. When data can be written to the UAnTX register (when data of one frame is transferred from the UAnTX register to the UARTAn transmit shift register), the transmission enable interrupt request signal (INTUAnT) is generated.

14.4 Control Registers

(1) UARTAn control register 0 (UAnCTL0)

The UAnCTL0 register is an 8-bit register that controls the UARTAn serial transfer operation.

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

Reset sets this register to 10H.

(1/2)

After reset: 10H R/W Address: UA0CTL0 FFFFFA00H, UA1CTL0 FFFFFA10H, UA2CTL0 FFFFFA20H

UAnCTL0 (n = 0 to 2)

<7>	<6>	<5>	<4>	3	2	1	0
UAnPWR	UAnTXE	UAnRXE	UAnDIR	UAnPS1	UAnPS0	UAnCL	UAnSL

UAnPWR	UARTAn operation control
0	Disable UARTAn operation (UARTAn reset asynchronously)
1	Enable UARTAn operation

The UARTAn operation is controlled by the UAnPWR bit. The TXDAn pin output is fixed to high level by clearing the UAnPWR bit to 0 (fixed to low level if UAnOPT0.UAnTDL bit = 1).

UAnTXE	Transmission operation enable
0	Disable transmission operation
1	Enable transmission operation

- To start transmission, set the UAnPWR bit to 1 and then set the UAnTXE bit to 1.
- To initialize the transmission unit, clear the UAnTXE bit to 0, wait for two cycles of the base clock (f_{UCLK}), and then set the UAnTXE bit to 1 again. Otherwise, initialization may not be executed (for the base clock, see **14.7 (1) (a) Base clock**).
- When the operation is enabled (UAnPWR bit = 1), the transmission operation is enabled after two or more cycles of the base clock (f_{UCLK}) have elapsed since UAnTXE = 1.
- When the UAnPWR bit is cleared to 0, the status of the internal circuit becomes the same status as UAnTXE bit = 0 by the UAnPWR bit even if the UAnTXE bit is 1. The transmission operation is enabled when the UAnPWR bit is set to 1 again.

UAnRXE	Reception operation enable
0	Disable reception operation
1	Enable reception operation

- To start reception, set the UAnPWR bit to 1 and then set the UAnRXE bit to 1.
- To initialize the reception unit, clear the UAnRXE bit to 0, wait for two cycles of the base clock, and then set the UAnRXE bit to 1 again. Otherwise, initialization may not be executed (for the base clock, see **14.7 (1) (a) Base clock**).
- When the operation is enabled (UAnPWR bit = 1), the reception operation is enabled after two or more cycles of the base clock (f_{UCLK}) have elapsed since UAnRXE = 1. The start bit is ignored if it is received before the reception operation is enabled.
- When the UAnPWR bit is cleared to 0, the status of the internal circuit becomes the same status as UAnRXE bit = 0 by the UAnPWR bit even if the UAnRXE bit is 1. The reception operation is enabled when the UAnPWR bit is set to 1 again.

(2/2)

UAnDIR[Note]	Transfer direction selection
0	MSB-first transfer
1	LSB-first transfer

UAnPS1[Note]	UAnPS0[Note]	Parity selection during transmission	Parity selection during reception
0	0	No parity output	Reception with no parity
0	1	0 parity output	Reception with 0 parity
1	0	Odd parity output	Odd parity check
1	1	Even parity output	Even parity check
If “reception with 0 parity” is selected during reception, a parity check is not performed. Therefore, since the UAnSTR.UAnPE bit is not set, no error interrupt due to a parity error is output.			

UAnCL[Note]	Specification of data character length of 1 frame of transmit/receive data
0	7 bits
1	8 bits

UAnSL[Note]	Specification of length of stop bit for transmit data
0	1 bit
1	2 bits
Only the first bit of the receive data stop bits is checked, regardless of the value of the UAnSL bit.	

Note This register can be rewritten only when the UAnPWR bit = 0 or the UAnTXE bit = UAnRXE bit = 0. However, setting any or all of the UAnPWR, UAnTXE, and UAnRXE bits to 1 at the same time is possible.

Remark For details of parity, see **14.6.6 Parity types and operations**.

(2) UARTAn control register 1 (UAnCTL1)

For details, see **14.7 (2) UARTAn control register 1 (UAnCTL1)**.

(3) UARTAn control register 2 (UAnCTL2)

For details, see **14.7 (3) UARTAn control register 2 (UAnCTL2)**.

(4) UARTAn option control register 0 (UAnOPT0)

The UAnOPT0 register is an 8-bit register that controls the serial transfer operation of UARTAn.

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

Reset sets this register to 14H.

After reset: 14H R/W Address: UA0OPT0 FFFFFA03H, UA1OPT0 FFFFFA13H, UA2OPT0 FFFFFA23H

	7	6	5	4	3	2	1	0
UAnOPT0 (n = 0 to 2)	0	0	0	1	0	1	UAnTDL	UAnRDL

UAnTDL	Transmit data level bit
0	Normal output of transfer data
1	Inverted output of transfer data

- The output level of the TXDAn pin can be inverted using the UAnTDL bit.
- This register can be set when the UAnCTL0.UAnPWR bit = 0 or when the UAnCTL0.UAnTXE bit = 0.

UAnRDL	Receive data level bit
0	Normal input of transfer data
1	Inverted input of transfer data

- The input level of the RXDAn pin can be inverted using the UAnRDL bit.
- This register can be set when the UAnPWR bit = 0 or the UAnCTL0.UAnRXE bit = 0.
- When the UAnRDL bit is set to 1 (inverted input of receive data), reception must be enabled (UAnCTL0.UAnRXE bit = 1) after setting the data reception pin to the UART reception pin (RXDAn) when reception is started. When the pin mode is changed after reception is enabled, the start bit will be mistakenly detected if the pin level is high.

Caution Be sure to set bits 3 and 5 to 7 to "0", and set bits 2 and 4 to "1". Operation with other settings is not guaranteed.

(5) UARTAn status register (UAnSTR)

The UAnSTR register is an 8-bit register that displays the UARTAn transfer status and reception error contents.

This register can be read or written in 8-bit or 1-bit units, but the UAnTSF bit is a read-only bit, while the UAnPE, UAnFE, and UAnOVE bits can both be read and written. However, these bits can only be cleared by writing 0; they cannot be set by writing 1 (even if 1 is written to them, the value is retained).

The initialization conditions are shown below.

Register/Bit	Initialization Conditions
UAnSTR register	• After reset • UAnCTL0.UAnPWR bit = 0
UAnTSF bit	• UAnCTL0.UAnTXE bit = 0
UAnPE, UAnFE, UAnOVE bits	• 0 write • UAnCTL0.UAnRXE bit = 0

Caution Be sure to read and check the error flags of the UAnPE, UAnFE, and UAnOVE bits, and clear the flags by writing "0" to them.

After reset: 00H R/W Address: UA0STR FFFFFA04H, UA1STR FFFFFA14H, UA2STR FFFFFA24H

	<7>	6	5	4	3	<2>	<1>	<0>
UAnSTR	UAnTSF	0	0	0	0	UAnPE	UAnFE	UAnOVE

(n = 0 to 2)

UAnTSF	Transfer status flag
0	• When the UAnPWR bit = 0 or the UAnTXE bit = 0 has been set. • When, following transfer end, there was no next data transfer from UAnTX register
1	Write to UAnTX register

The UAnTSF bit is always 1 when performing continuous transmission. When initializing the transmission unit, check that the UAnTSF bit = 0 before performing initialization. The transmit data is not guaranteed when initialization is performed while the UAnTSF bit = 1.

UAnPE	Parity error flag
0	• When the UAnPWR bit = 0 or the UAnRXE bit = 0 has been set. • When 0 has been written
1	When parity of data and parity bit do not match during reception.

- The operation of the UAnPE bit is controlled by the settings of the UAnCTL0.UAnPS1 and UAnCTL0.UAnPS0 bits.
- The UAnPE bit can be read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. When 1 is written to this bit, the value is retained.

UAnFE	Framing error flag
0	• When the UAnPWR bit = 0 or the UAnRXE bit = 0 has been set. • When 0 has been written
1	When no stop bit is detected during reception

- Only the first bit of the receive data stop bits is checked, regardless of the value of the UAnCTL0.UAnSL bit.
- The UAnFE bit can be both read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. When 1 is written to this bit, the value is retained.

UAnOVE	Overrun error flag
0	• When the UAnPWR bit = 0 or the UAnRXE bit = 0 has been set. • When 0 has been written
1	When receive data has been set to the UAnRX register and the next receive operation is ended before that receive data has been read.

- When an overrun error occurs, the data is discarded without the next receive data being written to the UAnRX register.
- The UAnOVE bit can be both read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. When 1 is written to this bit, the value is retained.

(6) UARTAn receive data register (UAnRX)

The UAnRX register is an 8-bit buffer register that stores parallel data converted by the UARTAn receive shift register.

The data stored in the UARTAn receive shift register is transferred to the UAnRX register upon end of reception of 1 byte of data. A reception end interrupt request signal (INTUAnR) is generated at this timing.

During LSB-first reception when the data length has been specified as 7 bits, the receive data is transferred to bits 6 to 0 of the UAnRX register and the MSB always becomes 0. During MSB-first reception, the receive data is transferred to bits 7 to 1 of the UAnRX register and the LSB always becomes 0.

When an overrun error occurs (UAnSTR.UAnOVE bit = 1), the receive data at this time is not transferred to the UAnRX register and is discarded.

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

In addition to reset, the UAnRX register can be set to FFH by clearing the UAnCTL0.UAnPWR bit to 0.

After reset: FFH R Address: UA0RX FFFFFA06H, UA1RX FFFFFA16H, UA2RX FFFFFA26H

	7	6	5	4	3	2	1	0
UAnRX								

(n = 0 to 2)

(7) UARTAn transmit data register (UAnTX)

The UAnTX register is an 8-bit register used to set transmit data.

Transmission starts when transmit data is written to the UAnTX register in the transmission enabled status (UAnCTL0.UAnTXE bit = 1). Upon end of the transfer of the data of the UAnTX register to the UARTAn transmit shift register, the transmission enable interrupt request signal (INTUAnT) is generated.

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

Reset sets this register to FFH.

After reset: FFH R/W Address: UA0TX FFFFFA07H, UA1TX FFFFFA17H, UA2TX FFFFFA27H

	7	6	5	4	3	2	1	0
UAnTX								

(n = 0 to 2)

14.5 Interrupt Request Signals

The following three interrupt request signals are generated from UARTAn.

- Reception error interrupt request signal (INTUAnRE)
- Reception end interrupt request signal (INTUAnR)
- Transmission enable interrupt request signal (INTUAnT)

Among these three interrupt signals, the reception error interrupt signal has the highest default priority, and the reception end interrupt request signal and transmission enable interrupt request signal follow in this order.

Table 14-2. Interrupts and Their Default Priorities

Interrupt	Priority
Reception error	High
Reception end	⇕
Transmission enable	Low

(1) Reception error interrupt request signal (INTUAnRE)

A reception error interrupt request signal is generated while reception is enabled by ORing the three types of reception errors (parity error, framing error, and overrun error) explained in the UAnSTR register section.

(2) Reception end interrupt request signal (INTUAnR)

A reception end interrupt request signal is output when data is shifted into the UARTAn receive shift register and transferred to the UAnRX register in the reception enabled status.

No reception end interrupt request signal is generated in the reception disabled status.

(3) Transmission enable interrupt request signal (INTUAnT)

If transmit data is transferred from the UAnTX register to the UARTAn transmit shift register with transmission enabled, the transmission enable interrupt request signal is generated.

14.6 Operation

14.6.1 Data format

Full-duplex serial data reception and transmission is performed.

As shown in Figure 14-5, one data frame of transmit/receive data consists of a start bit, character bits, parity bit, and stop bit(s).

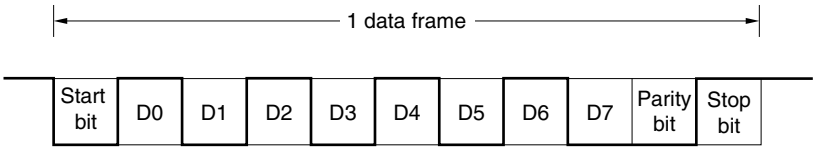
Specification of the character bit length within 1 data frame, parity selection, specification of the stop bit length, and specification of MSB-/LSB-first transfer are performed using the UAnCTL0 register.

Moreover, control of UARTAn output/inverted output for the TXDAn pin is performed using the UAnOPT0.UAnTDL bit.

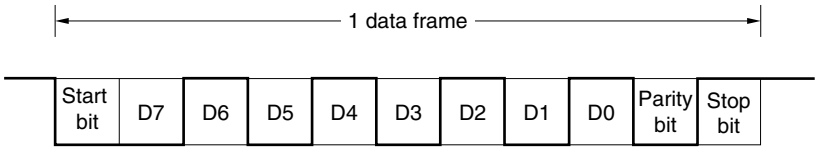
- Start bit ..... 1 bit
- Character bits..... 7 bits/8 bits
- Parity bit ..... Even parity/odd parity/0 parity/no parity
- Stop bit..... 1 bit/2 bits

Figure 14-5. UARTA Transmit/Receive Data Format

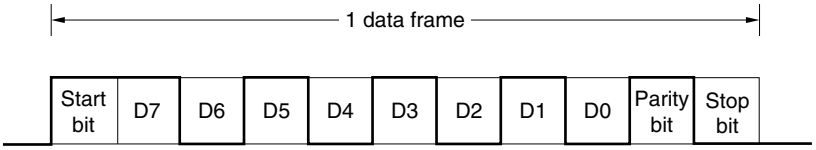
(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H



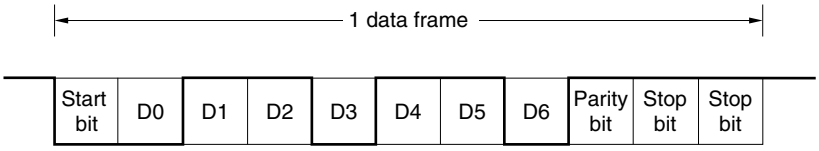
(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H



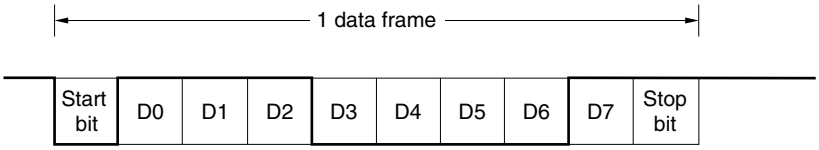
(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H, TXDAn inversion



(d) 7-bit data length, LSB first, odd parity, 2 stop bits, transfer data: 36H



(e) 8-bit data length, LSB first, no parity, 1 stop bit, transfer data: 87H



14.6.2 UART transmission

A high level is output to the TXDAn pin by setting the UAnCTL0.UAnPWR bit to 1.

Next, the transmission enabled status is set by setting the UAnCTL0.UAnTXE bit to 1, and transmission is started by writing transmit data to the UAnTX register. The start bit, parity bit, and stop bit are automatically added.

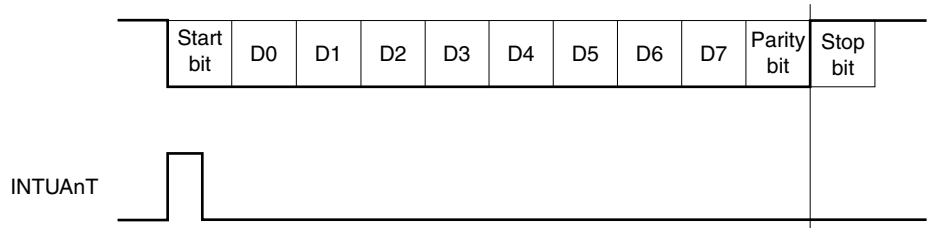
Since the CTS (transmit enable signal) input pin is not provided in UARTAn, use a port to check that reception is enabled at the transmit destination.

The data in the UAnTX register is transferred to the UARTAn transmit shift register upon the start of the transmit operation.

A transmission enable interrupt request signal (INTUAnT) is generated upon end of transmission of the data of the UAnTX register to the UARTAn transmit shift register, and thereafter the contents of the UARTAn transmit shift register are output to the TXDAn pin.

Write of the next transmit data to the UAnTX register is enabled by generating the INTUAnT signal.

Figure 14-6. UART Transmission



Remarks 1. LSB first

2. n = 0 to 2

14.6.3 Continuous transmission procedure

UARTAn can write the next transmit data to the UAnTX register when the UARTAn transmit shift register starts the shift operation. The transmit timing of the UARTAn transmit shift register can be judged from the transmission enable interrupt request signal (INTUAnT). An efficient communication rate is realized by writing the data to be transmitted next to the UAnTX register during transfer.

Caution During continuous transmission execution, perform initialization after checking that the UAnSTR.UAnTSF bit is 0. The transmit data cannot be guaranteed when initialization is performed while the UAnTSF bit is 1.

Remark n = 0 to 2

Figure 14-7. Continuous Transmission Processing Flow

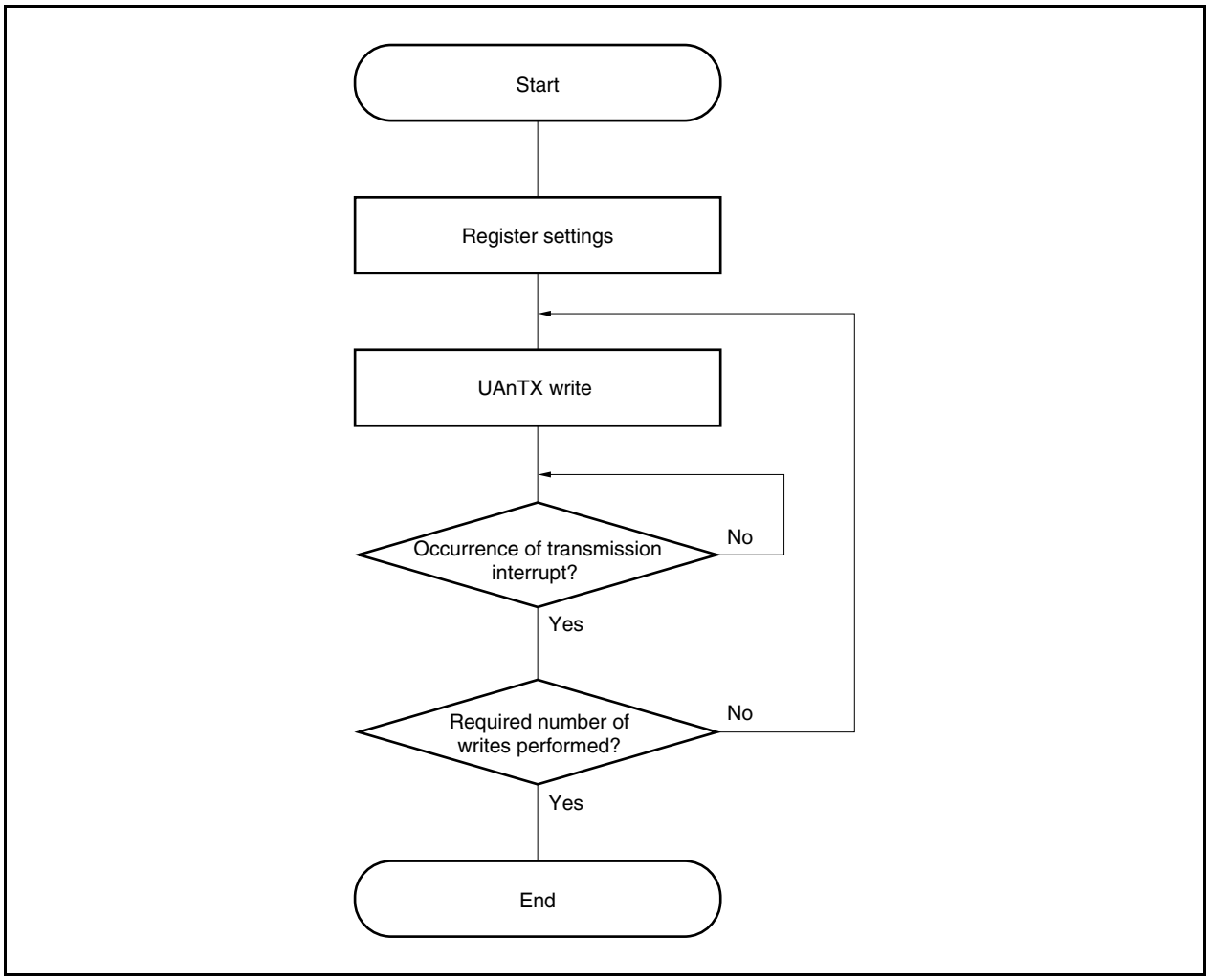
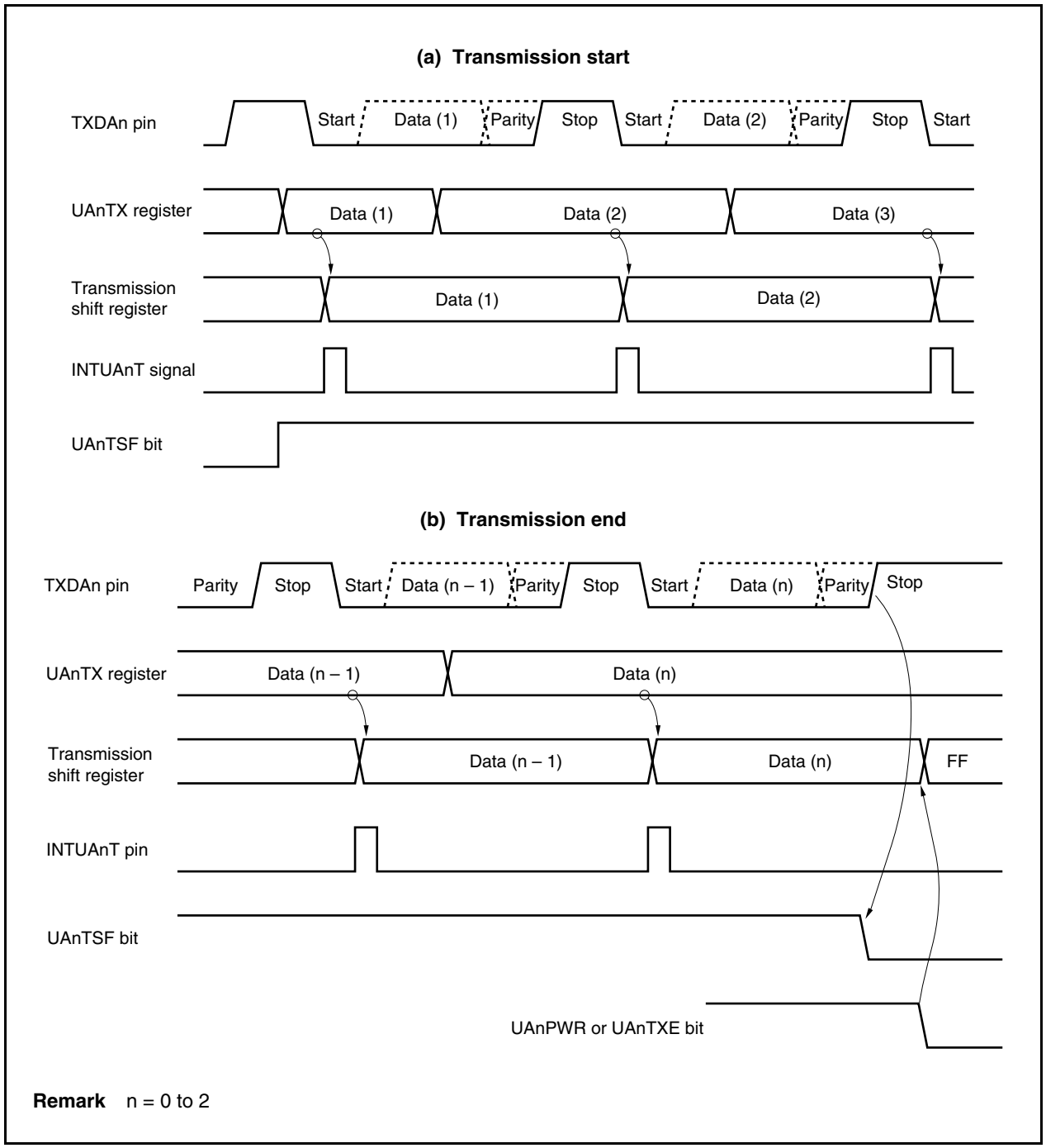


Figure 14-8. Continuous Transmission Operation Timing



Remark n = 0 to 2

14.6.4 UART reception

The reception wait status is set by setting the UAnCTL0.UAnPWR bit to 1 and then setting the UAnCTL0.UAnRXE bit to 1. In the reception wait status, the RXDAn pin is monitored and start bit detection is performed.

Start bit detection is performed using a two-step detection routine.

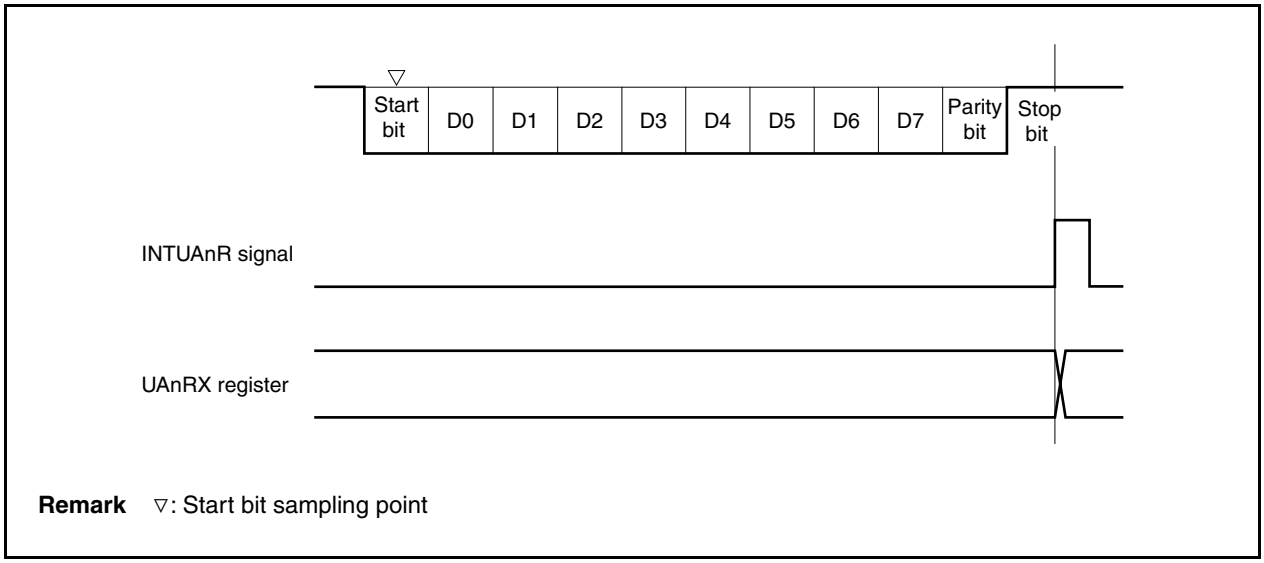
First the falling edge of the RXDAn pin is detected and sampling is started at the falling edge. The start bit is recognized if the RXDAn pin is low level at the start bit sampling point. After a start bit has been recognized, the receive operation starts, and serial data is saved to the UARTAn receive shift register according to the set baud rate.

When the reception end interrupt request signal (INTUAnR) is output upon reception of the stop bit, the data of the UARTAn receive shift register is written to the UAnRX register. However, if an overrun error occurs (UAnSTR.UAnOVE bit = 1), the receive data at this time is not written to the UAnRX register and is discarded.

Even if a parity error (UAnSTR.UAnPE bit = 1) or a framing error (UAnSTR.UAnFE bit = 1) occurs during reception, reception continues until the reception position of the first stop bit, and the INTUAnRE signal is output following reception end.

Remark n = 0 to 2

Figure 14-9. UART Reception



Remark ▽: Start bit sampling point

Cautions 1. Be sure to read the UAnRX register even when a reception error occurs. If the UAnRX register is not read, an overrun error occurs during reception of the next data, and reception errors continue occurring indefinitely.

2. The operation during reception is performed assuming that there is only one stop bit. A second stop bit is ignored.

3. When reception is completed, read the UAnRX register after the reception end interrupt request signal (INTUAnR) has been generated, and clear the UAnPWR or UAnRXE bit to 0. If the UAnPWR or UAnRXE bit is cleared to 0 before the INTUAnR signal is generated, the read value of the UAnRX register cannot be guaranteed.

4. If receive end processing (INTUAnR signal generation) of UARTAn and the UAnPWR bit = 0 or UAnRXE bit = 0 conflict, the INTUAnR signal may be generated in spite of these being no data stored in the UAnRX register. To end reception without waiting INTUAnR signal generation, be sure to clear (0) the interrupt request flag (UAnRIC.UAnRIF), after setting (1) the interrupt mask flag (UAnRIC.UAnRMK) and then set (1) the UAnPWR bit = 0 or UAnRXE bit = 0.

14.6.5 Reception errors

Errors during a receive operation are of three types: parity errors, framing errors, and overrun errors. Data reception result error flags are set in the UAnSTR register and a reception error interrupt request signal (INTUAnRE) is output when an error occurs.

It is possible to ascertain which error occurred during reception by reading the contents of the UAnSTR register.

Clear the reception error flag by writing 0 to it after reading it.

Caution The reception end interrupt request signal (INTUAnR) and reception error interrupt request signal (INTUAnRE) are not generated simultaneously. The INTUAnR signal is generated when a reception ends normally. The INTUAnRE signal is generated and the INTUAnR signal is not generated when a reception error occurs.

Remark n = 0 to 2

- Reception error causes

Error Flag	Reception Error	Cause
UAnPE	Parity error	Received parity bit does not match the setting
UAnFE	Framing error	Stop bit not detected
UAnOVE	Overrun error	Reception of next data ended before data was read from UAnRX register

14.6.6 Parity types and operations

The parity bit is used to detect bit errors in the communication data. Normally the same parity is used on the transmission side and the reception side.

In the case of even parity and odd parity, it is possible to detect odd-count bit errors. In the case of 0 parity and no parity, errors cannot be detected.

(a) Even parity

(i) During transmission

The number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so as to be an even number. The parity bit values are as follows.

- Odd number of bits whose value is “1” among transmit data: 1
- Even number of bits whose value is “1” among transmit data: 0

(ii) During reception

The number of bits whose value is “1” among the reception data, including the parity bit, is counted, and if it is an odd number, a parity error is output.

(b) Odd parity

(i) During transmission

Opposite to even parity, the number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so that it is an odd number. The parity bit values are as follows.

- Odd number of bits whose value is “1” among transmit data: 0
- Even number of bits whose value is “1” among transmit data: 1

(ii) During reception

The number of bits whose value is “1” among the receive data, including the parity bit, is counted, and if it is an even number, a parity error is output.

(c) 0 parity

During transmission, the parity bit is always made 0, regardless of the transmit data.

During reception, parity bit check is not performed. Therefore, no parity error occurs, regardless of whether the parity bit is 0 or 1.

(d) No parity

No parity bit is added to the transmit data.

Reception is performed assuming that there is no parity bit. No parity error occurs since there is no parity bit.

14.6.7 Receive data noise filter

This filter samples the RXDAn pin using the base clock (f_{UCLK}) of the prescaler output.

When the same sampling value is read twice, the match detector output changes and the RXDAn signal is sampled as the input data. Therefore, data not exceeding 2 clock width is judged to be noise and is not delivered to the internal circuit (see **Figure 14-11**). See **14.7 (1) (a) Base clock** regarding the base clock.

Moreover, since the circuit is as shown in Figure 14-10, the processing that goes on within the receive operation is delayed by 3 clocks in relation to the external signal status.

Remark n = 0 to 2

Figure 14-10. Noise Filter Circuit

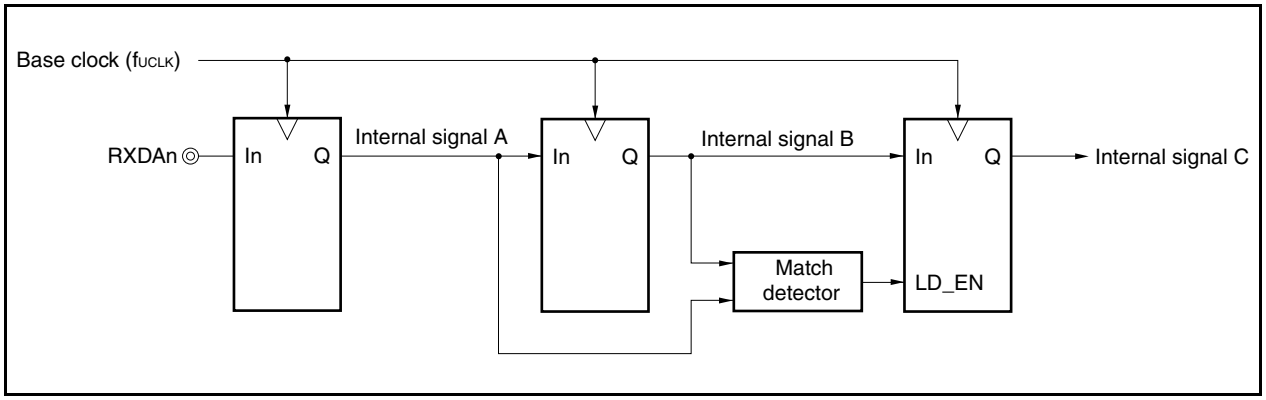
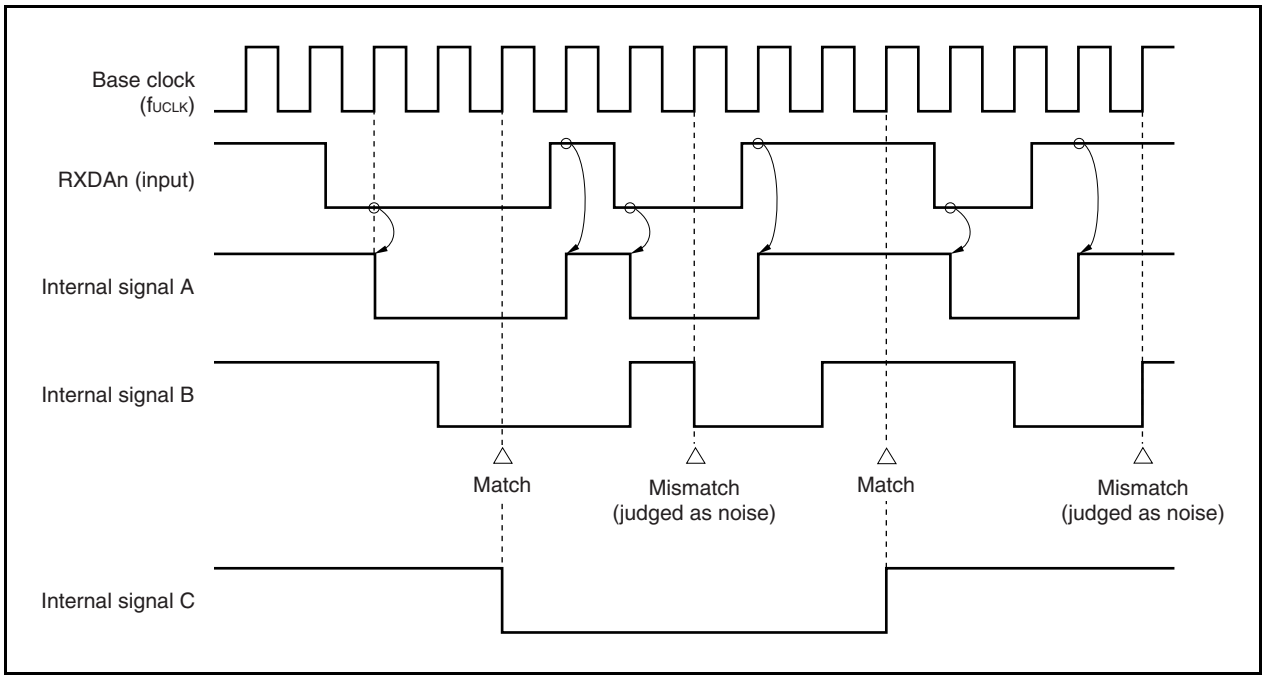


Figure 14-11. Timing of RXDAn Signal Judged as Noise



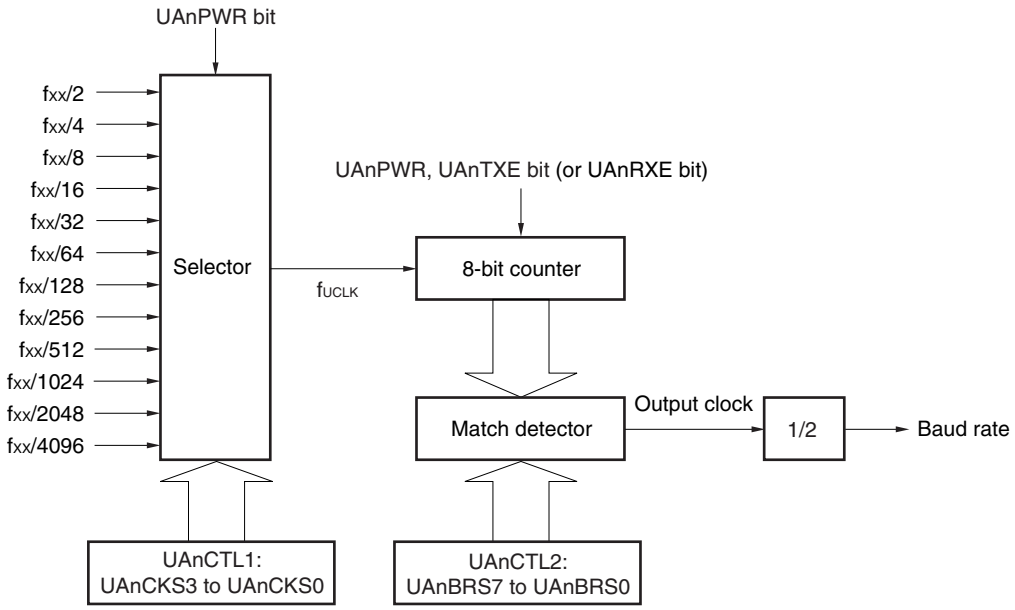
14.7 Dedicated Baud Rate Generator

The dedicated baud rate generator consists of a source clock selector block and an 8-bit programmable counter, and generates a serial clock during transmission and reception with UARTAn. Regarding the serial clock, a dedicated baud rate generator output can be selected for each channel.

There is an 8-bit counter for transmission and another one for reception.

(1) Baud rate generator configuration

Figure 14-12. Configuration of Baud Rate Generator



Caution If the CPU clock (f_{CPU}) is slower than f_{UCLK}, UARTAn cannot be used.

Remarks 1. n = 0 to 2

2. fxx: Peripheral clock frequency

(a) Base clock

When the UAnCTL0.UAnPWR bit is 1, the clock selected by the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits is supplied to the 8-bit counter. This clock is called the base clock (f_{UCLK}). When the UAnPWR bit = 0, f_{UCLK} is fixed to the low level.

(b) Serial clock generation

A serial clock can be generated by setting the UAnCTL1 register and the UAnCTL2 register.

The base clock (f_{UCLK}) is selected by the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits.

The frequency division value for the 8-bit counter can be set using the UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits.

(2) UARTAn control register 1 (UAnCTL1)

The UAnCTL1 register is an 8-bit register that selects the UARTAn base clock.

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

Reset sets this register to 00H.

Caution Clear the UAnCTL0.UAnPWR bit to 0 before rewriting the UAnCTL1 register.

After reset: 00H R/W Address: UA0CTL1 FFFFFA01H, UA1CTL1 FFFFFA11H, UA2CTL1 FFFFFA21H

	7	6	5	4	3	2	1	0
UAnCTL1 (n = 0 to 2)	0	0	0	0	UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0

UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0	Base clock (f_{UCLK}) selection
0	0	0	0	fxx/2
0	0	0	1	fxx/4
0	0	1	0	fxx/8
0	0	1	1	fxx/16
0	1	0	0	fxx/32
0	1	0	1	fxx/64
0	1	1	0	fxx/128
0	1	1	1	fxx/256
1	0	0	0	fxx/512
1	0	0	1	fxx/1,024
1	0	1	0	fxx/2,048
1	0	1	1	fxx/4,096
Other than above				Setting prohibited

Remark fxx: Peripheral clock frequency

(3) UARTAn control register 2 (UAnCTL2)

The UAnCTL2 register is an 8-bit register that selects the baud rate (serial transfer speed) clock of UARTAn.

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

Reset sets this register to FFH.

Caution Clear the UAnCTL0.UAnPWR bit to 0 or clear the UAnTXE and UAnRXE bits to 00 before rewriting the UAnCTL2 register.

After reset: FFH R/W Address: UA0CTL2 FFFFFA02H, UA1CTL2 FFFFFA12H, UA2CTL2 FFFFFA22H

	7	6	5	4	3	2	1	0
UAnCTL2	UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0

(n = 0 to 2)

UAn BRS7	UAn BRS6	UAn BRS5	UAn BRS4	UAn BRS3	UAn BRS2	UAn BRS1	UAn BRS0	Default (k)	Serial clock
0	0	0	0	0	0	×	×	×	Setting prohibited
0	0	0	0	0	1	0	0	4	$f_{UCLK}/4$
0	0	0	0	0	1	0	1	5	$f_{UCLK}/5$
0	0	0	0	0	1	1	0	6	$f_{UCLK}/6$
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	$f_{UCLK}/252$
1	1	1	1	1	1	0	1	253	$f_{UCLK}/253$
1	1	1	1	1	1	1	0	254	$f_{UCLK}/254$
1	1	1	1	1	1	1	1	255	$f_{UCLK}/255$

Remark f_{UCLK}: Frequency of base clock selected by the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits

(4) Baud rate

The baud rate is obtained by the following equation.

$$\text{Baud rate} = \frac{f_{UCLK}}{2 \times k} \text{ [bps]}$$

f_{UCLK}: Frequency of base clock selected by the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits

k: Value set using the UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (k = 4, 5, 6, ..., 255)

(5) Baud rate error

The baud rate error is obtained by the following equation.

$$\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Target baud rate (correct baud rate)}} - 1 \right) \times 100 \text{ [\%]}$$

Cautions 1. The baud rate error during transmission must be within the error tolerance on the receiving side.

2. The baud rate error during reception must satisfy the range indicated in section (7) Allowable baud rate range during reception.

Example Peripheral clock frequency = 32 MHz = 32,000,000 Hz

Set value of UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits = 0000B (f_{UCLK} = 16,000,000 Hz)

Set value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits = 00110100B (k = 52)

Target baud rate = 153,600

Baud rate = 16,000,000/ (2 × 52) = 153,846 [bps]

Error = (153,846/153,600 − 1) × 100

= 0.160 [%]

(6) Baud rate setting example

Table 14-3. Baud Rate Generator Setting Data

Baud Rate (bps)	f_{XX} = 64 MHz			f_{XX} = 32 MHz		
	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)
300	08H	D0H	0.16	07H	D0H	0.16
600	07H	D0H	0.16	06H	D0H	0.16
1,200	06H	D0H	0.16	05H	D0H	0.16
2,400	05H	D0H	0.16	04H	D0H	0.16
4,800	04H	D0H	0.16	03H	D0H	0.16
9,600	03H	D0H	0.16	02H	D0H	0.16
19,200	02H	D0H	0.16	01H	D0H	0.16
31,250	02H	80H	0	00H	80H	0
38,400	01H	D0H	0.16	00H	D0H	0.16
76,800	00H	D0H	0.16	00H	68H	0.16
153,600	00H	68H	0.16	00H	34H	0.16
312,500	00H	33H	0.39	00H	1AH	–1.54
625,000	00H	1AH	–1.54	00H	0DH	–1.54
1,250,000	00H	0DH	–1.54	00H	06H	6.67

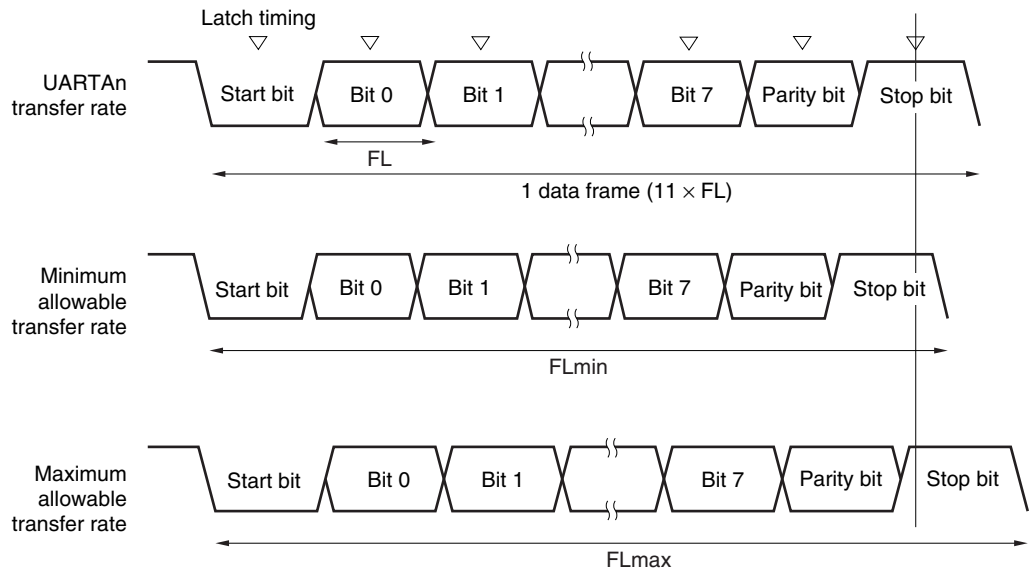
Remark f_{XX}: Peripheral clock frequency
ERR: Baud rate error (%)

(7) Allowable baud rate range during reception

The baud rate error range at the destination that is allowable during reception is shown below.

Caution The baud rate error during reception must be set within the allowable error range using the following equation.

Figure 14-13. Allowable Baud Rate Range During Reception



Remark n = 0 to 2

As shown in Figure 14-13, the receive data latch timing is determined by the counter set using the UAnCTL2 register following start bit detection. The transmit data can be normally received if up to the last data (stop bit) can be received in time for this latch timing.

When this is applied to 11-bit reception, the following is the theoretical result.

$FL = (Brate)^{-1}$

Brate: UARTAn baud rate (n = 0 to 2)

k: Set value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (n = 0 to 2)

FL: 1-bit data length

Latch timing margin: 2 clocks

Minimum allowable transfer rate: $FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$

Therefore, the maximum baud rate that can be received by the destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2} \text{ Brate}$$

Similarly, obtaining the following maximum allowable transfer rate yields the following.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FLmax = \frac{21k - 2}{20\ k} FL \times 11$$

Therefore, the minimum baud rate that can be received by the destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k - 2} \text{ Brate}$$

Obtaining the allowable baud rate error for UARTAn and the destination from the above-described equations for obtaining the minimum and maximum baud rate values yields the following.

Table 14-4. Maximum/Minimum Allowable Baud Rate Error

Division Ratio (k)	Maximum Allowable Baud Rate Error	Minimum Allowable Baud Rate Error
4	+2.32%	–2.43%
8	+3.52%	–3.61%
20	+4.26%	–4.30%
50	+4.56%	–4.58%
100	+4.66%	–4.67%
255	+4.72%	–4.72%

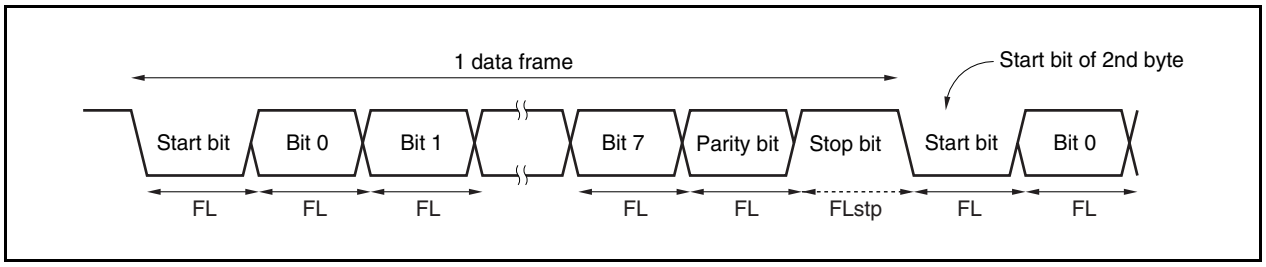
Remarks 1. The reception accuracy depends on the bit count in 1 frame, the input clock frequency, and the division ratio (k). The higher the input clock frequency and the larger the division ratio (k), the higher the accuracy.

2. k: Set value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (n = 0 to 2)

(8) Transfer rate during continuous transmission

During continuous transmission, the transfer rate from the stop bit to the next start bit is usually 2 base clocks longer. However, timing initialization is performed via start bit detection by the receiving side, so this has no influence on the transfer result.

Figure 14-14. Transfer Rate During Continuous Transmission



Assuming 1 bit data length: FL; stop bit length: FLstp; and base clock frequency: f_{UCLK}, we obtain the following equation.

$$FLstp = FL + 2/f_{UCLK}$$

Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times FL + (2/f_{UCLK})$$

14.8 Cautions

When the clock supply to UARTAn is stopped (for example, in IDLE or STOP mode), the operation stops with each register retaining the value it had immediately before the clock supply was stopped. The TXDAn pin output also holds and outputs the value it had immediately before the clock supply was stopped. However, the operation is not guaranteed after the clock supply is resumed. Therefore, after the clock supply is resumed, the circuits should be initialized by setting the UAnCTL0.UAnPWR, UAnCTL0.UAnRXE, and UAnCTL0.UAnTXE bits to 000.

Remark n = 0 to 2

15.1 Mode Switching Between UARTB and CSIB2

In the V850E/IF3 and V850E/IG3, UARTB and CSIB2 function alternately, and these pins cannot be used at the same time. To switch between UARTB and CSIB2, the PMC3, PFC3, and PFCE3 registers must be set in advance.

Caution The operations related to transmission and reception of UARTB or CSIB2 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 15-1. Mode Switch Settings of UARTB and CSIB2

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	PFCE37	PFCE36	PFCE35	PFCE34	0	PFCE32	PFCE31	PFCE30

PMC37	PFCE37	PFC37	Specification of alternate function of P37 pin
0	×	×	I/O port
1	0	0	$\overline{\text{SCKB2}}$ I/O
1	0	1	INTP12 input
1	1	0	ASTB[Note] output
1	1	1	Setting prohibited

PMC36	PFCE36	PFC36	Specification of alternate function of P36 pin
0	×	×	I/O port
1	0	0	SOB2 output
1	0	1	TXDB output
1	1	0	Setting prohibited
1	1	1	Setting prohibited

PMC35	PFCE35	PFC35	Specification of alternate function of P35 pin
0	×	×	I/O port
1	0	0	SIB2 input
1	0	1	RXDB input
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Note μPD70F3454GC-8EA-A only

Remark x = don't care

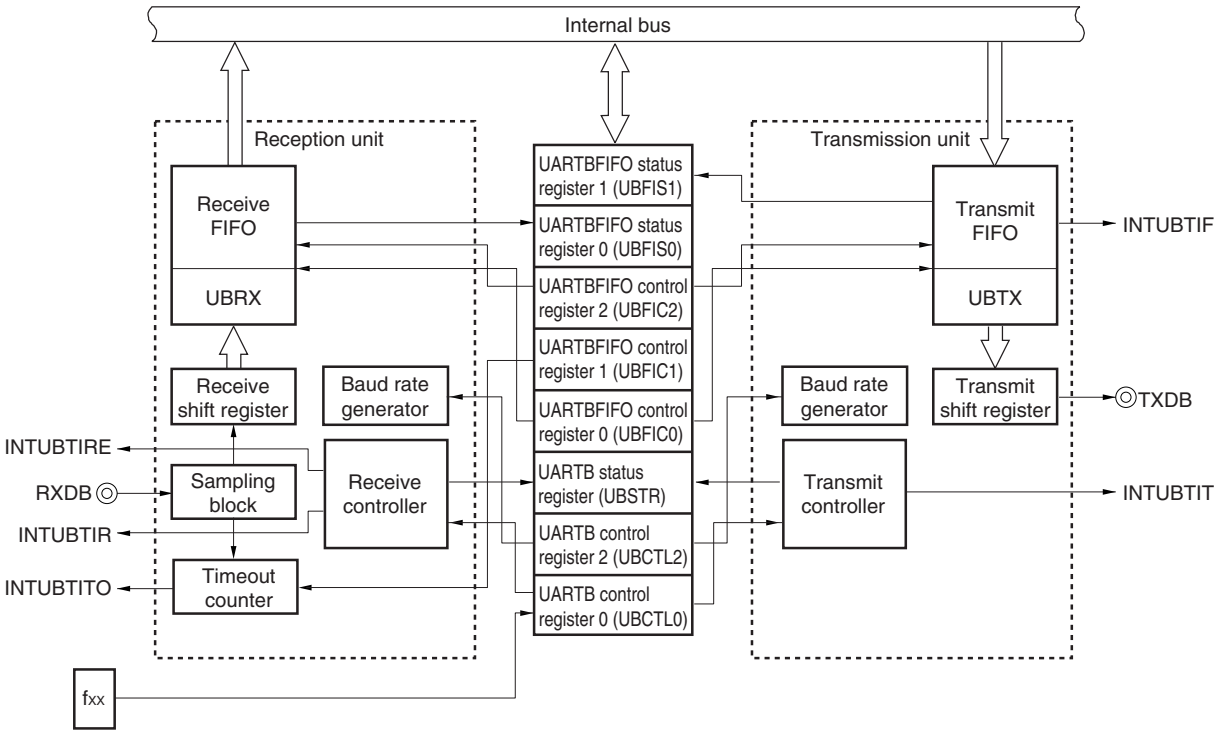
15.2 Features

- Transfer rate: Maximum 5.33 Mbps (using a dedicated baud rate generator)
- Full-duplex communications
- Single mode and FIFO mode selectable
  - Single mode: 8-bit × 1-stage data register (UBTX register or UBRX register) is used for each of transmission and reception.
  - FIFO mode
    Transmit FIFO: UBTX register (8 bits × 16 stages).
    Receive FIFO: UBRXAP register (16 bits × 16 stages)
    2 bits of the higher 8 bits of the UBRXAP register are for an error flag.
- Two-pin configuration
  TXDB: Transmit data output pin
  RXDB: Receive data input pin
- Reception error detection function
  - Overflow error (FIFO mode only)
  - Parity error
  - Framing error
  - Overrun error (single mode only)
- Interrupt sources: 5 types
  - Reception error interrupt request signal (INTUBTIRE)
  - Reception end interrupt request signal (INTUBTIR)
  - Transmission end interrupt request signal (INTUBTIT)
  - FIFO transmission end interrupt request signal (INTUBTIF) (FIFO mode only)
  - Reception timeout interrupt request signal (INTUBTITO) (FIFO mode only)
- The character length of transmit/receive data is specified according to the UBCTL0 register
- Character length: 7 or 8 bits
- Parity functions: Odd, even, 0, or none
- Transmission stop bits: 1 or 2 bits
- MSB first/LSB first selectable for transfer data
- On-chip dedicated baud rate generator

15.3 Configuration

The block diagram of the UARTB is shown below.

Figure 15-2. Block Diagram of UARTB



Remarks 1. fxx: Peripheral clock

2. For the configuration of the baud rate generator, see **Figure 15-9**.

UARTB consists of the following hardware units.

Table 15-1. Configuration of UARTB

Item	Configuration
Registers	UARTB control register 0 (UBCTL0) UARTB control register 2 (UBCTL2) UARTB status register (UBSTR) UARTB FIFO control register 0 (UBFIC0) UARTB FIFO control register 1 (UBFIC1) UARTB FIFO control register 2 (UBFIC2) UARTB FIFO status register 0 (UBFIS0) UARTB FIFO status register 1 (UBFIS1) Receive shift register UARTB receive data register AP (UBRXAP) UARTB receive data register (UBRX) Transmit shift register UARTB transmit data register (UBTX)

(1) UARTB control register 0 (UBCTL0)

This register controls the transfer operation of UARTB.

(2) UARTB status register (UBSTR)

This register indicates the transfer status during transmission and the contents of a reception error. The status flag of this register, which indicates the transfer status during transmission, indicates the data retention status of the transmit shift register and the transmit data register (the UBTX register in the single mode or transmit FIFO in the FIFO mode). Each reception error flag is set to 1 when a reception error occurs, and cleared to 0 when 0 is written to the UBSTR register.

(3) UARTB control register 2 (UBCTL2)

This register is used to specify the division ratio by which to control the baud rate (serial transfer speed) of UARTB.

(4) UARTB FIFO control register 0 (UBFIC0)

This register is used to select the operation mode of UARTB, clear the transmit FIFO/receive FIFO that becomes valid in the FIFO mode, and specify the timing mode in which the transmission end interrupt request signal (INTUBTIT)/reception end interrupt request signal (INTUBTIR) occurs.

(5) UARTB FIFO control register 1 (UBFIC1)

This register is valid in the FIFO mode. It generates a reception timeout interrupt request signal (INTUBTITO) if data is stored in the receive FIFO when the next data does not come (start bit is not detected) even after the reception wait time of the next data has elapsed after the stop bit has been received.

(6) UARTB FIFO control register 2 (UBFIC2)

This register is valid in the FIFO mode. It is used to set the timing to generate the transmission end interrupt request signal (INTUBTIT)/reception end interrupt request signal (INTUBTIR), using the number of data transmitted or received as a trigger.

(7) UARTB FIFO status register 0 (UBFIS0)

This register is valid in the FIFO mode. The number of bytes of data stored in the receive FIFO can be read from this register.

(8) UARTB FIFO status register 1 (UBFIS1)

This register is valid in the FIFO mode. The number of empty bytes of the transmit FIFO can be read from this register.

(9) Receive shift register

This is a shift register that converts the serial data that was input to the RXDB pin into parallel data. One byte of data is received, and if a stop bit is detected, the received data is transferred to the receive data register.

This register cannot be directly manipulated.

(10) UARTB receive data register AP (UBRXAP), UARTB receive data register (UBRX)

The receive data register holds receive data. In the single mode, the 8-bit × 1-stage UBRX register is used. The 16-bit × 16-stage receive FIFO (UBRXAP register) is used in the FIFO mode.

The receive data is stored in the lower 8 bits of the receive FIFO (UBRXAP register) and the error information of the received data is stored in the higher 8 bits (bit 8 and bit 9). If a reception error (such as a parity error or a framing error) occurs in the FIFO mode, the error data can be identified by reading the UBRXAP register in 16-bit (halfword) units (error information is appended as UBPEF bit = 1 or UBFEF bit = 1). When the lower 8 bits of the UBRXAP register are read in 8-bit (byte) units, the higher 8 bits are discarded. Therefore, if no error has occurred, only the receive data of the UBRXAP register can be read successively by being read in 8-bit (byte) units in the same way as the UBRX register.

When 7-bit length data is received with the LSB first, the received data is transferred to bits 6 to 0 of the receive data register from the LSB (bit 0), with the MSB (bit 7) always being 0. When data is received with the MSB first, the received data is transferred to bits 7 to 1 of the receive data register from the MSB (bit 7), with the LSB (bit 0) always being 0. If an overrun error occurs, the receive data at that time is not transferred to the receive data register.

While reception is enabled, the received data is transferred from the receive shift register to the receive data register, in synchronization with the shift-in processing of one frame.

A reception end interrupt request signal (INTUBTIR) is generated by transferring the data to the UBRX register in the single mode, or transferring the number of receive data set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits to receive FIFO in the FIFO mode. If data is stored in receive FIFO when the next data does not come (start bit is not detected) after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed in the FIFO mode, a reception timeout interrupt request signal (INTUBTITO) is generated.

(11) Transmit shift register

This is a shift register that converts the parallel data that was transferred from the transmit data register into serial data.

When one byte of data is transferred from the transmit data register, the transmit shift register data is output from the TXDB pin.

This register cannot be directly manipulated.

(12) UARTB transmit data register (UBTX)

The transmit data register is a buffer for transmit data. The 8-bit × 1-stage UBTX register is used as this buffer in the single mode. In the FIFO mode, the 8-bit × 16-stage transmit FIFO is used.

When 7-bit length data is transmitted with the LSB first, bits 6 to 0 of the transmit data register are transmitted as the transmit data from the LSB (bit 0) with the MSB (bit 7) always being 0. When data is transmitted with the MSB first, bits 7 to 1 of the transmit data register are transmitted as the transmit data from the MSB (bit 7) with the LSB (bit 0) always being 0.

In the single mode, transmission is started by writing transmit data to the UBTX register while transmission is enabled (UBCTL0.UBTXE bit = 1). When writing the transmit data to the UBTX register is enabled (when 1-byte data is transferred from the UBTX register to the transmit shift register), a transmission end interrupt request signal (INTUBTIT) is generated.

In the FIFO mode, transmission is started by writing at least the number of transmit data set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less to transmit FIFO and then enabling transmission (UBTXE bit = 1). When the number of transmit data set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits have been transferred from transmit FIFO to the transmit shift register (transmit data of the number set as the trigger can be written), a transmission end interrupt request signal (INTUBTIT) is generated. In the FIFO mode, a FIFO transmission end interrupt request signal (INTUBTIF) is generated when there is no more data in transmit FIFO and the transmit shift register (when FIFO and the register become empty).

(13) Timeout counter

This counter is used to recognize that data exists (remains) in receive FIFO when the number of received data does not reach the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits, and is valid only in the FIFO mode.

If data is stored in receive FIFO when the next data does not come (start bit is not detected) after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed after the stop bit has been received, a reception timeout interrupt request signal (INTUBTITO) is generated.

(14) Sampling block

This block samples the RXDB signal at the rising edge of the peripheral clock (f_{XX}). If the same sampling value is detected two times, output of the match detector changes, and the value is sampled as input data. Data of less than one clock width is judged as noise and is not transmitted to the internal circuitry.

15.4 Control Registers

(1) UARTB control register 0 (UBCTL0)

The UBCTL0 register controls the transfer operations of UARTB.

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

Reset sets this register to 10H.

Cautions 1. When using UARTB, set the external pins related to the UARTB function in the alternate-function mode, set UARTB control register 2 (UBCTL2). Then set the UBPWR bit to 1 before setting the other bits.

2. Be sure to input a high level to the RXDB pin when setting the external pins related to the UARTB function in the alternate-function mode. If a low level is input, it is judged that a falling edge is input after the UBRXE bit has been set to 1, and reception may be started.

Remark When reception is disabled, the receive shift register does not detect a start bit. No shift-in processing or transfer processing to the receive data register is performed, and the contents of the receive data register are retained.

When reception is enabled, the receive shift operation starts, in synchronization 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 receive data register.

A reception end interrupt request signal (INTUBTIR) is also generated, in synchronization with the transfer to the receive data register (in FIFO mode, transfer triggered by reaching set number of receive data).

If data is stored in receive FIFO when the next data does not come (start bit is not detected) after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed in the FIFO mode, a reception timeout interrupt request signal (INTUBTITO) is generated.

(1/2)

After reset: 10H R/W Address: FFFFFA40H

	<7>	<6>	<5>	<4>	3	2	1	0
UBCTL0	UBPWR	UBTXE	UBRXE	UBDIR	UBPS1	UBPS0	UBCL	UBSL

UBPWR	Operation clock control to UARTB
0	Stops supply of clocks to UARTB
1	Supplies clocks to UARTB

- When the UBPWR bit is cleared to 0, the UARTB can be asynchronously reset.
- When the UBPWR bit = 0, UARTB is in a reset state. Therefore, to operate UARTB, the UBPWR bit must be set to 1.
- When the UBPWR bit is changed from 1 to 0, all registers of UARTB are initialized. When the UBPWR bit is set to 1 again, the UARTB registers must be set again.
- The TXDB pin output is high level when the UBPWR bit is cleared to 0.

UBTXE	Transmission enable
0	Transmission is disabled
1	Transmission is enabled

- On startup, set the UBPWR bit to 1 and then set the UBTXE bit to 1. To stop transmission, clear the UBTXE bit to 0 and then the UBPWR bit to 0.
- When the transmission unit status is to be initialized, the transmission status may not be able to be initialized unless the UBTXE bit is set to 1 again after an interval of two cycles of f_{XX} has elapsed since the UBTXE bit was cleared to 0.

UBRXE	Reception enable
0	Reception is disabled
1	Reception is enabled

- On startup, set the UBPWR bit to 1 and then set the UBRXE bit to 1. To stop reception, clear the UBRXE bit to 0 and then the UBPWR bit to 0.
- When the reception unit status is to be initialized, the reception status may not be able to be initialized unless the UBRXE bit is set to 1 again after an interval of two cycles of f_{XX} has elapsed since the UBRXE bit was cleared to 0.

(2/2)

UBDIR	Specification of transfer direction mode (MSB/LSB)
0	MSB transfer first
1	LSB transfer first
• Clear the UBPWR bit or UBTXE and UBRXE bits to 0 before changing the setting of the UBDIR bit.	

UBPS1	UBPS0	Parity selection during transmission	Parity selection during reception
0	0	Do not output a parity bit	Receive with no parity
0	1	Output 0 parity	Receive as 0 parity
1	0	Output odd parity	Judge as odd parity
1	1	Output even parity	Judge as even parity
• Clear the UBTXE and UBRXE bits to 0 before overwriting the UBPS1 and UBPS0 bits. • If “0 parity” is selected for reception, no parity judgment is made. Therefore, no error interrupt is generated because the UBSTR.UBPE bit is not set to 1.			

UBCL	Specification of data character length of 1-frame transmit/receive data
0	7 bits
1	8 bits
Clear the UBTXE and UBRXE bits to 0 before overwriting the UBCL bit.	

UBSL	Specification of stop bit length of transmit data
0	1 bit
1	2 bits
• Clear the UBTXE bit to 0 before overwriting the UBSL bit. • Since reception always operates by using a single stop bit length, the UBSL bit setting does not affect receive operations.	

Remark For details of parity, see **15.7.6 Parity types and corresponding operation**.

(2) UARTB status register (UBSTR)

The UBSTR register indicates the transfer status and reception error contents while UARTB is transmitting data.

The status flag that indicates the transfer status during transmission indicates the data retention status of the transmit shift register and transmit data register (the UBTX register in the single mode or transmit FIFO in the FIFO mode). The status flag that indicates a reception error holds its status until it is cleared to 0.

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

Reset sets this register to 00H.

Caution When the UBCTL0.UBPWR bit or UBCTL0.UBRXE bit is set to 0, or when 0 is written to the UBSTR register, the UBSTR.UBOVF, UBSTR.UBPE, UBSTR.UBFE, and UBSTR.UBOVE bits are cleared to 0.

After reset: 00H R/W Address: FFFFFA44H

	<7>	6	5	4	3	<2>	<1>	<0>
UBSTR	UBTSF	0	0	0	UBOVF	UBPE	UBFE	UBOVE

UBTSF	Transfer status flag
0	• In single mode (UBFIC0.UBMOD bit = 0) Data to be transferred to the transmit shift register and UBTX register does not exist (cleared (0) when UBCTL0.UBPWR bit = 0 or UBCTL0.UBTXE bit = 0). • In FIFO mode (UBFIC0.UBMOD bit = 1) Data to be transferred to the transmit shift register and transmit FIFO does not exist (cleared (0) when UBCTL0.UBPWR bit = 0 or UBCTL0.UBTXE bit = 0).
1	• In single mode (UBFIC0.UBMOD bit = 0) Data to be transferred to the transmit shift register or UBTX register exists (transmission in progress). • In FIFO mode (UBFIC0.UBMOD bit = 1) Data to be transferred to the transmit shift register and transmit FIFO exists (transmission in progress).
The value of the UBTSF bit is reflected after two periods of f_{XX} have elapsed, after the transmit data is written to the UBTX register. Therefore, exercise care when referencing the UBTSF bit after transmit data has been written to the UBTX register.	

UBOVF	Overflow flag
0	Overflow did not occur.
1	Overflow occurred (during reception).
• The UBOVF bit is valid only in the FIFO mode (when UBFIC0.UBMOD bit = 1), and invalid in the single mode (when UBFIC0.UBMOD bit = 0). • If an overflow occurs, the received data is not written to receive FIFO but discarded.	

(2/2)

UBPE	Parity error flag
0	Parity error did not occur.
1	Parity error occurred (during reception).

- The UBPE bit is valid only in the single mode (when UBFIC0.UBMOD bit = 0), and invalid in the FIFO mode (when UBFIC0.UBMOD bit = 1).
- The operation of the UBPE bit differs according to the settings of the UBCTL0.UBPS1 and UBCTL0.UBPS0 bits.

UBFE	Framing error flag
0	Framing error did not occur.
1	Framing error occurred (during reception).

- The UBFE bit is valid only in the single mode (when UBFIC0.UBMOD bit = 0), and invalid in the FIFO mode (when UBFIC0.UBMOD bit = 1).
- Only the first bit of the stop bits of the receive data is checked, regardless of the stop bit length.

UBOVE	Overrun error flag
0	Overrun error did not occur.
1	Overrun error occurred (during reception).

- The UBOVE bit is valid only in the single mode (when UBFIC0.UBMOD bit = 0), and invalid in the FIFO mode (when UBFIC0.UBMOD bit = 1).
- When an overrun error occurs, the next receive data value is not written to the UBRX register and the data is discarded.

(3) UARTB control register 2 (UBCTL2)

The UBCTL2 register is used to specify the division ratio by which to control the baud rate (serial transfer speed) of UARTB.

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

Reset sets this register to FFFFH.

Caution When rewriting the UBBRS15 to UBBRS0 bits of this register, set the UBCTL0.UBTXE and UBCTL0.UBRXE bits to 0 or clear the UBCTL0.UBPWR bit to 0.

After reset: FFFFH R/W Address: FFFFFA42H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UBCTL2	UBBRS15	UBBRS14	UBBRS13	UBBRS12	UBBRS11	UBBRS10	UBBRS9	UBBRS8	UBBRS7	UBBRS6	UBBRS5	UBBRS4	UBBRS3	UBBRS2	UBBRS1	UBBRS0

Remark For the UBBRS15 to UBBRS0 bits, see **Table 15-2 Division Value of 16-bit Counter**.

Table 15-2. Division Value of 16-bit Counter

UBBRS15	UBBRS14	UBBRS13	UBBRS12	UBBRS11	UBBRS10	UBBRS9	UBBRS8	UBBRS7	UBBRS6	UBBRS5	UBBRS4	UBBRS3	UBBRS2	UBBRS1	UBBRS0	k	Output Clock Selected
0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	4	fxx/k
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4	fxx/k
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	5	fxx/k
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	6	fxx/k
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	65532	fxx/k
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	65533	fxx/k
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	65534	fxx/k
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	65535	fxx/k

Remarks 1. fxx: Peripheral clock

2. k: Value set by the UBCTL2.UBBRS15 to UBCTL2.UBBRS0 bits (k = 4, 5, 6, …, 65535)

3. x: Don't care

(4) UARTB transmit data register (UBTX)

The UBTX register is used to set transmit data. It functions as the 8-bit × 1-stage UBTX register, in the single mode (UBFIC0.UBMOD bit = 0), and as the 8-bit × 16-stage transmit FIFO in the FIFO mode (UBFIC0.UBMOD bit = 1).

In the single mode, transmission is started by writing transmit data to the UBTX register when transmission is enabled (UBCTL0.UBTXE bit = 1). When data can be written to the UBTX register (when 1 byte of data is transferred from the UBTX register to the transmit shift register), a transmission end interrupt request signal (INTUBTIT) is generated.

In the FIFO mode, transmission is started by enabling transmission (UBTXE bit = 1) after writing at least the number of transmit data set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less to transmit FIFO. When the number of transmit data set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits have been transferred from transmit FIFO to the transmit shift register (transmit data of the number set as the trigger can be written to transmit FIFO), a transmission end interrupt request signal (INTUBTIT) is generated. In the FIFO mode, a FIFO transmission end interrupt request signal (INTUBTIF) is generated when there is no more data in transmit FIFO and the transmit shift register (when the FIFO and register become empty).

For the generation timing of the interrupt, see **15.5 Interrupt Request Signals**.

When 7-bit length data is transmitted with the LSB first, bits 6 to 0 of the transmit data register are transmitted as the transmit data from the LSB (bit 0) with the MSB (bit 7) always being 0. When data is transmitted with the MSB first, bits 7 to 1 of the transmit data register are transmitted as the transmit data from the MSB (bit 7) with the LSB (bit 0) always being 0.

This register is write-only in 8-bit units. Data is written to the transmit data register.

Reset sets this register to FFH.

After reset: FFH W Address: FFFFFA48H

	7	6	5	4	3	2	1	0
UBTX	UBTD7	UBTD6	UBTD5	UBTD4	UBTD3	UBTD2	UBTD1	UBTD0

(5) UARTB receive data register AP (UBRXAP), UARTB receive data register (UBRX)

These registers store parallel data converted by the receive shift register. They function as the 8-bit × 1-stage UBRX register, in the single mode (UBFIC0.UBMOD bit = 0), and as the 16-bit × 16-stage receive FIFO (UBRXAP register) in the FIFO mode (UBFIC0.UBMOD bit = 1).

The receive data is stored in the lower 8 bits of the receive FIFO (UBRXAP register) and the error information of the received data is stored in the higher 8 bits (bit 8 and bit 9). If a reception error (such as a parity error or a framing error) occurs in the FIFO mode, the UBRXAP register is read in 16-bit (halfword) units. In this way, the flag of the data stored in receive FIFO can be checked (error information is appended as UBPEF bit = 1 or UBFEF bit = 1), so that the error data can be recognized (when the lower 8 bits of the UBRXAP register are read in 8-bit (byte) units, the higher 8 bits are discarded. Therefore, if no error has occurred, the receive data of the UBRXAP register can be read successively by being read in 8-bit (byte) units in the same way as the UBRX register).

If reception is enabled (UBCTL0.UBRXE bit = 1), the receive data is transferred from the receive shift register to the receive data register, in synchronization with the completion of the shift-in processing of one frame.

By transferring the receive data to the UBRX register in the single mode or by transferring the number of receive data set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits to the receive FIFO in the FIFO mode, a reception end interrupt request signal (INTUBTIR) is generated. If data is stored in receive FIFO when the next data does not come (start bit is not detected) even after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed in the FIFO mode, a reception timeout interrupt request signal (INTUBTITO) is generated.

For information about the timing for generating these interrupt requests, see **15.5 Interrupt Request Signals**.

If data is received with the LSB first when the data length is specified as 7 bits, the received data is transferred to bits 6 to 0 of the receive data register from the LSB (bit 0), with the MSB (bit 7) always being 0. If data is received with the MSB first, it is transferred to bits 7 to 1 of the receive data register from the MSB (bit 7) with the LSB (bit 0) always being 0. However, if an overrun error occurs, the receive data at that time is not transferred to the receive data register.

The UBRXAP register is read-only in 16-bit units. However, the lower 8 bits of the UBRXAP register are read-only in 8-bit units.

The UBRX register is read-only in 8-bit units.

In addition to reset input, the value of these registers can be set to FFH in the single mode or to 00FFH in the FIFO mode, by clearing the UBCTL0.UBPWR bit to 0.

Cautions 1. The UBPEF and UBFEF bits cannot be read because these registers serve as 8-bit registers in the single mode.

2. When no reception error has occurred in the FIFO mode, the receive data of the UBRXAP register can be read successively by reading the lower 8 bits of the UBRXAP register in 8-bit (byte) units. An 8-bit access to the higher 8 bits is prohibited. If they are accessed, the operation is not guaranteed.

Cautions 3. Do not perform the following operations when debugging a system that uses the single mode.

- **Setting a break for an instruction immediately after the UBRX register is read**
- **Setting a break before DMA transfer with the UBRX register specified as the transfer source is ended**
- **Setting a break before end of reception of the next data after reception of data and reading the UBRX register, and checking the UBRX register in the I/O register window of the debugger**

If any of these operations is performed, an overrun error may occur during the subsequent reception.

After reset: 00FFH R Address: FFFFFA46H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UBRXAP	0	0	0	0	0	0	UBPEF	UBFEF	UBRD7	UBRD6	UBRD5	UBRD4	UBRD3	UBRD2	UBRD1	UBRD0

After reset: FFH R Address: FFFFFA46H

	7	6	5	4	3	2	1	0
UBRX	UBRD7	UBRD6	UBRD5	UBRD4	UBRD3	UBRD2	UBRD1	UBRD0

UBPEF	Parity error flag
0	No parity error
1	Parity error occurs (during reception).

- The UBPEF bit is valid only in the FIFO mode (UBFIC0.UBMOD bit = 1), and is invalid in the single mode (UBFIC0.UBMOD bit = 0).
- The operation of the UBPEF bit differs depending on the set values of the UBCTL0.UBPS1 and UBCTL0.UBPS0 bits.

UBFEF	Framing error flag
0	No framing error
1	Framing error occurs (during reception).

- The UBFEF bit is valid only in the FIFO mode (UBFIC0.UBMOD bit = 1), and is invalid in the single mode (UBFIC0.UBMOD bit = 0).
- Only the first bit of the stop bits of the receive data is checked, regardless of the stop bit length.

UBRD7 to UBRD0	Stores receive data.

(6) UARTB FIFO control register 0 (UBFIC0)

The UBFIC0 register is used to select the operation mode of UARTB and the functions that become valid in the FIFO mode (UBMOD bit = 1). In the FIFO mode, it clears transmit FIFO/receive FIFO and specifies the timing mode in which the transmission end interrupt request signal (INTUBTIT)/reception end interrupt request signal (INTUBTIR) is generated.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFFA4AH

	7	6	5	4	3	2	1	0
UBFIC0	UBMOD	0	0	0	UBTFC	UBRFC	UBITM	UBIRM

UBMOD	Specification of UARTB operation mode
0	Single mode
1	FIFO mode

UBTFC	Transmit FIFO clear trigger bit
0	Normal status
1	Clear (This bit automatically returns to 0 after transmit FIFO is cleared.)

- The UBTFC bit is valid only in the FIFO mode (UBMOD bit = 1), and is invalid in the single mode (UBMOD bit = 0).
- When 1 is written to the UBTFC bit, the pointer to transmit FIFO is cleared to 0. In the pending mode (UBITM bit = 0), the interrupt request signal (INTUBTIT) held pending is cleared[Note]. However, bit 7 (UTIF) of the interrupt control register (UTIC) is not cleared to 0. Clear this bit to 0 as necessary.
  When 0 is written to the UBTFC bit, the status is retained. No operation, such as clearing or setting, is executed.
- When writing 1 to the UBTFC bit, be sure to clear the UBCTL0.UBTXE bit to 0 (disabling transmission). If 1 is written to the UBTFC bit when the UBTXE bit is 1 (transmission enabled), the operation is not guaranteed.

Note After transmit FIFO is cleared (UBTFC bit = 1), accessing the registers related to UARTB is prohibited for the duration of four cycles of fxx or until clearing the UBTFC bit (automatic recovery) is confirmed by reading the UBFIC0 register. If these registers are accessed, the operation is not guaranteed.

Remark fxx: Peripheral clock

(2/2)

UBRFC	Receive FIFO (UBRXAP) clear trigger bit
0	Normal status
1	Clear (This bit automatically returns to 0 after receive FIFO is cleared.)

- The UBRFC bit is valid only in the FIFO mode (UBMOD bit = 1), and is invalid in the single mode (UBMOD bit = 0).
- When 1 is written to the UBRFC bit, the pointer to receive FIFO is cleared to 0. In the pending mode (UBIRM bit = 0), the interrupt request signal (INTUBTIR) held pending is cleared[Note]. However, bit 7 (URIF) of the interrupt control register (URIC) is not cleared to 0. Clear this bit to 0 as necessary.
  When 0 is written to the UBRFC bit, the status is retained. No operation, such as clearing or setting, is executed.
- When writing 1 to the UBRFC bit, be sure to clear the UBCTL0.UBRXE bit to 0 (disabling reception). If 1 is written to the UBRFC bit when the UBRXE bit is 1 (reception enabled), the operation is not guaranteed.

UBITM	Specification of INTUBTIT interrupt generation timing in FIFO mode
0	Pending mode
1	Pointer mode

In the FIFO mode, the INTUBTIT signal is generated as soon as transmit data of the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits have been transferred from transmit FIFO to the transmit shift register. After the INTUBTIT signal request has been generated, specify the timing of actually generating the INTUBTIT signal as the pending mode or pointer mode. For details, see **15.6 (2) Pending mode/pointer mode**.

UBIRM	Specification of INTUBTIR interrupt generation timing in FIFO mode
0	Pending mode
1	Pointer mode

In the FIFO mode, the INTUBTIR signal is generated as soon as receive data of the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits have been transferred from the receive shift register to receive FIFO. After the INTUBTIR signal request has been generated, specify the timing of actually generating the INTUBTIR signal as the pending mode or pointer mode. For details, see **15.6 (2) Pending mode/pointer mode**.

Note After receive FIFO (UBRXAP) is cleared (UBRFC bit = 1), accessing the registers related to UARTB is prohibited for the duration of four cycles of fxx or until clearing the UBRFC bit (automatic recovery) is confirmed by reading the UBFIC0 register. If these registers are accessed, the operation is not guaranteed.

Remark fxx: Peripheral clock

(7) UARTB FIFO control register 1 (UBFIC1)

The UBFIC1 register is valid in the FIFO mode (UBFIC0.UBMOD bit = 1). It generates a reception timeout interrupt request signal (INTUBTITO) if data is stored in receive FIFO when the next data does not come (start bit is not detected) after the lapse of the time set by the UBTC4 to UBTC0 bits (next data reception wait time), after the stop bit has been received.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFA4BH

	7	6	5	4	3	2	1	0
UBFIC1	UBTCE	0	0	UBTC4	UBTC3	UBTC2	UBTC1	UBTC0

UBTCE	Specification of timeout counter function disable/enable
0	Disable use of timeout counter function.
1	Enable use of timeout counter function.

UBTC4	UBTC3	UBTC2	UBTC1	UBTC0	Next data reception wait time
0	0	0	0	0	32 bytes (32 × 8/baud rate)
0	0	0	0	1	31 bytes (31 × 8/baud rate)
0	0	0	1	0	30 bytes (30 × 8/baud rate)
0	0	0	1	1	29 bytes (29 × 8/baud rate)
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
1	1	1	0	0	4 bytes (4 × 8/baud rate)
1	1	1	0	1	3 bytes (3 × 8/baud rate)
1	1	1	1	0	2 bytes (2 × 8/baud rate)
1	1	1	1	1	1 byte (1 × 8/baud rate)

When counting up of the reception wait time, set by the UBTC4 to UBTC0 bits, is complete, the count value of the timeout counter is cleared to 0, regardless of the status of the data stored in receive FIFO. When the next start bit is later detected, counting is started again from the stop bit of that data.

(8) UARTB FIFO control register 2 (UBFIC2)

The UBFIC2 register is valid in the FIFO mode (UBFIC0.UBMOD bit = 1). It sets the timing of generating an interrupt, using the number of transmit/receive data as a trigger. When data is transmitted, the number of data transferred from transmit FIFO is specified as the condition of generating the interrupt. When data is received, the number of data stored in receive FIFO is specified as the interrupt generation condition.

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

When the higher 8 bits of the UBFIC2 register can be used as the UBFIC2H register and the lower 8 bits, as the UBFIC2L register, these registers can be read or written in 8-bit units.

Reset sets the UBFIC2 register to 0000H and the UBFIC2H and UBFIC2L registers to 00H.

Caution Be sure to set the UBCTL0.UBTXE bit (to disable transmission) and UBCTL0.UBRXE bit (to disable reception) to 0 before writing data to the UBFIC2 register. If data is written to the UBFIC2 register with the UBTXE or UBRXE bit set to 1, the operation is not guaranteed.

(1/2)

After reset: 0000H R/W Address: FFFFFA4CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UBFIC2	0	0	0	0	UBTT3	UBTT2	UBTT1	UBTT0	0	0	0	0	UBRT3	UBRT2	UBRT1	UBRT0

UBTT3	UBTT2	UBTT1	UBTT0	Number of data of transmit FIFO set as trigger	Pointer mode	Pending mode
0	0	0	0	1 byte	Settable	Settable
0	0	0	1	2 bytes	Setting prohibited	
0	0	1	0	3 bytes		
0	0	1	1	4 bytes		
0	1	0	0	5 bytes		
0	1	0	1	6 bytes		
0	1	1	0	7 bytes		
0	1	1	1	8 bytes		
1	0	0	0	9 bytes		
1	0	0	1	10 bytes		
1	0	1	0	11 bytes		
1	0	1	1	12 bytes		
1	1	0	0	13 bytes		
1	1	0	1	14 bytes		
1	1	1	0	15 bytes		
1	1	1	1	16 bytes		

- Set the number of transmit FIFO transmit data to be the trigger.
- Each time data of the specified number has shifted out from transmit FIFO to the transmit shift register, the INTUBTIT signal is generated.
  In the pending mode (UBFIC0.UBITM bit = 0), the INTUBTIT signal is generated under the conditions of the pending mode.
- In the pointer mode (UBFIC0.UBITM bit = 1), the number of transmit data set as the trigger can be only 1 byte (UBTT3 to UBTT0 bits = 0000), and other settings are prohibited. If a setting of other than 1 byte is made, the operation is not guaranteed.

(2/2)

UBRT3	UBRT2	UBRT1	UBRT0	Number of data of transmit FIFO set as trigger	Pointer mode	Pending mode
0	0	0	0	1 byte	Settable	Settable
0	0	0	1	2 bytes	Setting prohibited	
0	0	1	0	3 bytes		
0	0	1	1	4 bytes		
0	1	0	0	5 bytes		
0	1	0	1	6 bytes		
0	1	1	0	7 bytes		
0	1	1	1	8 bytes		
1	0	0	0	9 bytes		
1	0	0	1	10 bytes		
1	0	1	0	11 bytes		
1	0	1	1	12 bytes		
1	1	0	0	13 bytes		
1	1	0	1	14 bytes		
1	1	1	0	15 bytes		
1	1	1	1	16 bytes		

- Set the number of receive FIFO receive data to be the trigger.
- Each time data of the specified number has been stored from the receive shift register to receive FIFO, the INTUBTIR interrupt is generated.
  In the pending mode (UBFIC0.UBIRM bit = 0), the INTUBTIR signal is generated under the conditions of the pending mode.
- In the pointer mode (UBFIC0.UBIRM bit = 1), the number of receive data set as the trigger can be only 1 byte (UBRT3 to UBRT0 bits = 0000), and other settings are prohibited. If a setting of other than 1 byte is made, the operation is not guaranteed.

(9) UARTB FIFO status register 0 (UBFIS0)

The UBFIS0 register is valid in the FIFO mode (UBFIC0.UBMOD bit = 1). It is used to read the number of bytes of the data stored in receive FIFO.

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

Reset sets this register to 00H.

After reset: 00H R Address: FFFFFA4EH

	7	6	5	4	3	2	1	0
UBFIS0	0	0	0	UBRB4	UBRB3	UBRB2	UBRB1	UBRB0

UBRB4	UBRB3	UBRB2	UBRB1	UBRB0	Receive FIFO pointer
0	0	0	0	0	0 bytes
0	0	0	0	1	1 byte
0	0	0	1	0	2 bytes
0	0	0	1	1	3 bytes
0	0	1	0	0	4 bytes
0	0	1	0	1	5 bytes
0	0	1	1	0	6 bytes
0	0	1	1	1	7 bytes
0	1	0	0	0	8 bytes
0	1	0	0	1	9 bytes
0	1	0	1	0	10 bytes
0	1	0	1	1	11 bytes
0	1	1	0	0	12 bytes
0	1	1	0	1	13 bytes
0	1	1	1	0	14 bytes
0	1	1	1	1	15 bytes
1	0	0	0	0	16 bytes
Other than above					Invalid

Indicates the number of bytes (readable bytes) of the data stored in receive FIFO as a receive FIFO pointer.

(10) UARTB FIFO status register 1 (UBFIS1)

The UBFIS1 register is valid in the FIFO mode (UBFIC0.UBMOD bit = 1). This register can be used to read the number of empty bytes of transmit FIFO.

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

Reset sets this register to 10H.

Caution The values of the UBTB4 to UBTB0 bits are reflected after transmit data has been written to the UBTX register and then time of two cycles of the peripheral clock (f_{XX}) has passed. Therefore, care must be exercised when referencing the UBFIS1 register after transmit data has been written to the UBTX register.

After reset: 10H R Address: FFFFFA4FH

	7	6	5	4	3	2	1	0
UBFIS1	0	0	0	UBTB4	UBTB3	UBTB2	UBTB1	UBTB0

UBTB4	UBTB3	UBTB2	UBTB1	UBTB0	Transmit FIFO pointer
0	0	0	0	0	0 bytes
0	0	0	0	1	1 byte
0	0	0	1	0	2 bytes
0	0	0	1	1	3 bytes
0	0	1	0	0	4 bytes
0	0	1	0	1	5 bytes
0	0	1	1	0	6 bytes
0	0	1	1	1	7 bytes
0	1	0	0	0	8 bytes
0	1	0	0	1	9 bytes
0	1	0	1	0	10 bytes
0	1	0	1	1	11 bytes
0	1	1	0	0	12 bytes
0	1	1	0	1	13 bytes
0	1	1	1	0	14 bytes
0	1	1	1	1	15 bytes
1	0	0	0	0	16 bytes
Setting prohibited					Invalid

Indicates the number of empty bytes of transmit FIFO (bytes that can be written) as a transmit FIFO pointer.

15.5 Interrupt Request Signals

The following five types of interrupt requests are generated from UARTB.

- Reception error interrupt request signal (INTUBTIRE)
- Reception end interrupt request signal (INTUBTIR)
- Transmission end interrupt request signal (INTUBTIT)
- FIFO transmission end interrupt request signal (INTUBTIF)
- Reception timeout interrupt request signal (INTUBTITO)

The default priorities among these five types of interrupt requests is, from high to low, reception error interrupt request signal, reception end interrupt request signal, transmission end interrupt request signal, FIFO transmission end interrupt request signal, and reception timeout interrupt request signal.

Table 15-3. Generated Interrupts and Default Priorities

Interrupt	Priority
Reception error	1
Reception end	2
Transmission end	3
FIFO transmission end	4
Reception timeout	5

(1) Reception error interrupt request signal (INTUBTIRE)

(a) Single mode

When reception is enabled, a reception error interrupt request signal is generated according to the logical OR of the three types of reception errors (parity error, framing error, overrun error) explained for the UBSTR register.

When reception is disabled, no reception error interrupt request signal is generated.

(b) FIFO mode

When reception is enabled, a reception error interrupt request signal is generated according to the logical OR of the three types of reception errors (parity error, framing error, overflow error) explained for the UBSTR register.

When reception is disabled, no reception error interrupt request signal is generated.

(2) Reception end interrupt request signal (INTUBTIR)

(a) Single mode

When reception is enabled, a reception end interrupt request signal is generated if data is shifted into the receive shift register and stored in the UBRX register (if the receive data can be read).
When reception is disabled, no reception end interrupt request signal is generated.

(b) FIFO mode

When reception is enabled, a reception end interrupt request signal is generated if data is shifted into the receive shift register and receive data of the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits is transferred to receive FIFO (if receive data of the specified number can be read).
When reception is disabled, no reception end interrupt request signal is generated.

(3) Transmission end interrupt request signal (INTUBTIT)

(a) Single mode

The transmission end interrupt request signal is generated if transmit data of one frame, including 7 or 8 bits of characters, is shifted out from the transmit shift register and the UBTX register becomes empty (if transmit data can be written).

(b) FIFO mode

The transmission end interrupt request signal is generated if transmit data of the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits is transferred to the transmit shift register from transmit FIFO (if transmit data of the specified number can be written).

(4) FIFO transmission end interrupt request signal (INTUBTIF)

(a) Single mode

Cannot be used.

(b) FIFO mode

The FIFO transmission end interrupt request signal is generated when no more data is in transmit FIFO and the transmit shift register (when the FIFO and register become empty). After the FIFO transmission end interrupt request signal has occurred, clear the interrupt request signal (INTUBTIT) held pending in the pending mode (UBFIC0.UBITM bit = 0) by clearing the FIFO (UBFIC0.UBTFC bit = 1).

Caution If the FIFO transmission end interrupt request signal is generated (all transmit data are not transmitted) because writing the next transmit data to transmit FIFO is delayed, do not clear the FIFO.

(5) Reception timeout interrupt request signal (INTUBTITO)

(a) Single mode

Cannot be used.

(b) FIFO mode

The reception timeout interrupt request signal is generated if data is stored in receive FIFO when the next data does not come (start bit is not detected) even after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed, when the timeout counter function is used (UBFIC1.UBTCE bit = 1).

The reception timeout interrupt request signal is not generated while reception is disabled.

If receive data of the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits is not received, the timing of reading the number of receive data less than the specified number can be set by the reception timeout interrupt request signal.

Since the timeout counter starts counting at start bit detection, a receive timeout interrupt request signal does not occur if data of 1 character has not been received.

15.6 Control Modes

(1) Single mode/FIFO mode

The single mode or FIFO mode can be selected by using the UBFIC0.UBMOD bit.

(a) Single mode

- Each of the UBRX and UBTX registers consists of 8 bits × 1 stage.
- When 1 byte of data is received, the INTUBTIR signal is generated.
- If the next reception operation of UARTB is ended before the receive data of the UBRX register is read after the INTUBTIR signal has been generated, the INTUBTIRE signal is generated and an overrun error occurs.

(b) FIFO mode

- Receive FIFO (UBRXAP register) consists of 16 bits × 16 stages and transmit FIFO consists of 8 bits × 16 stages.
- Receive FIFO can recognize error data by reading the 16-bit UBRXAP register only when a reception error (parity error or framing error) occurs.
- Transmission is started when transmission is enabled (UBCTL0.UBTXE bit = 1) after transmit data of at least the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less are written to transmit FIFO.
- The pending mode or pointer mode can be selected for the generation timing of the INTUBTIT and INTUBTIR signals.

(2) Pending mode/pointer mode

The pending mode or pointer mode can be selected by using the UBFIC0.UBITM and UBFIC0.UBIRM bits in the FIFO mode (UBFIC0.UBMOD bit = 1).

If transmission is started by writing data of more than double the amount set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to transmit FIFO, the transmission end interrupt request signal (INTUBTIT) may occur more than once. The reception end interrupt request signal (INTUBTIR) may also occur more than once if the number of receive data set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits is 8 bytes or less in receive FIFO. In the pending or pointer mode, it can be specified how an interrupt is handled after it has been held pending.

(a) Pending mode

(i) During transmission (writing to transmit FIFO)

- If the data of the first transmission end interrupt request signal (INTUBTIT) is not written to transmit FIFO after the interrupt has occurred, the second INTUBTIT signal does not occur (is held pending) even if the generation condition of the second INTUBTIT signal is satisfied (when transmit data of the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits is transferred from transmit FIFO to the transmit shift register).

  When data for the first INTUBTIT signal is later written to transmit FIFO, the pending INTUBTIT signal is generated[Note].

  Note The number of pending interrupts is as follows.

  When trigger is set to 1 byte (UBFIC2.UBTT3 to UBFIC2.UBTT0 bits = 0000): 15 times max.
  When trigger is set to 2 bytes (UBFIC2.UBTT3 to UBFIC2.UBTT0 bits = 0001): 7 times max.
  :
  When trigger is set to 6 bytes (UBFIC2.UBTT3 to UBFIC2.UBTT0 bits = 0101): 1 time max.
  When trigger is set to 7 bytes (UBFIC2.UBTT3 to UBFIC2.UBTT0 bits = 0110): 1 time max.
  When trigger is set to 8 bytes (UBFIC2.UBTT3 to UBFIC2.UBTT0 bits = 0111): 1 time max.

- In the pending mode, transmit data of the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits is always written to transmit FIFO when the transmission end interrupt request signal (INTUBTIT) occurs. Writing data to transmit FIFO is prohibited if the data is more or less than the specified number. If data more or less than the specified number is written, the operation is not guaranteed.
- Fix the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to 0000 (set number of transmit data: 1 byte) to write transmit data to transmit FIFO by DMA. If any other setting is made, the operation is not guaranteed.

(ii) During reception (reading from receive FIFO)

- If data for the first reception end interrupt request signal (INTUBTIR) is not read from receive FIFO, the second INTUBTIR signal does not occur (is held pending) even if the generation condition of the second INTUBTIR is satisfied (if receive data of the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits can be read from receive FIFO). When data for the first INTUBTIR signal is later read from the receive FIFO, the pending INTUBTIR signal is generated[Note].

  Note The number of pending interrupts is as follows.

  When trigger is set to 1 byte (UBFIC2.UBRT3 to UBFIC2.UBRT0 bits = 0000): 15 times max.
  When trigger is set to 2 bytes (UBFIC2.UBRT3 to UBFIC2.UBRT0 bits = 0001): 7 times max.
  :
  When trigger is set to 6 bytes (UBFIC2.UBRT3 to UBFIC2.UBRT0 bits = 0101): 1 time max.
  When trigger is set to 7 bytes (UBFIC2.UBRT3 to UBFIC2.UBRT0 bits = 0110): 1 time max.
  When trigger is set to 8 bytes (UBFIC2.UBRT3 to UBFIC2.UBRT0 bits = 0111): 1 time max.

- In the pending mode, receive data of the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits is always read from receive FIFO when the reception end interrupt request signal (INTUBTIR) occurs. Reading data from receive FIFO is prohibited if the data is more or less than the specified number. If data more or less than the specified number is read, the operation is not guaranteed.
- Fix the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits to 0000 (set number of receive data: 1 byte) to read receive data from receive FIFO by DMA. If any other setting is made, the operation is not guaranteed.

(b) Pointer mode

(i) During transmission (writing to transmit FIFO)

- Each time the data of 1 byte is transferred to the transmit shift register from transmit FIFO, a transmission end interrupt request signal (INTUBTIT) occurs.
- In the pointer mode, be sure to fix the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to 0000 (set number of transmit data: 1 byte) as the number of transmit data set as the trigger for transmit FIFO when the transmission end interrupt request signal (INTUBTIT) occurs. If any other setting is made, the operation is not guaranteed.
- Writing transmit data to transmit FIFO by DMA is prohibited. The operation is not guaranteed if DMA control is used.
- After the transmission end interrupt request signal (INTUBTIT) has been acknowledged, data of the number of empty bytes of transmit FIFO can be written to transmit FIFO by referencing the UBFIS1 register.

(ii) During reception (reading from receive FIFO)

- Each time the data of 1 byte is transferred to receive FIFO from the receive shift register, a reception end interrupt request signal (INTUBTIR) occurs.
- In the pointer mode, be sure to fix the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits to 0000 (set number of receive data: 1 byte) as the number of receive data set as the trigger for receive FIFO when the reception end interrupt request signal (INTUBTIR) occurs. If any other setting is made, the operation is not guaranteed.
- Reading receive data from receive FIFO by DMA is prohibited. The operation is not guaranteed if DMA control is used.
- After the reception end interrupt request signal (INTUBTIR) has been acknowledged, data of the number of bytes stored in receive FIFO can be read from receive FIFO by referencing the UBFIS0 register. In some cases, however, data is not stored in receive FIFO even though the INTUBTIR signal is generated (UBFIS0.UBRB4 to UBFIS0.UBRB0 bits = 00000). In these cases, do not read data from receive FIFO. Always read data from receive FIFO when the number of bytes stored in receive FIFO is 1 byte or more (UBRB4 to UBRB0 bits = other than 00000).

15.7 Operation

15.7.1 Data format

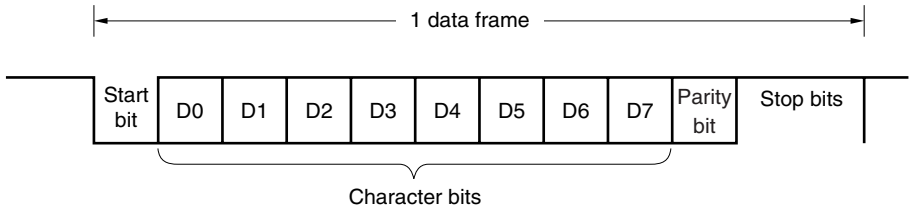
Full-duplex serial data transmission and reception can be performed.

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 15-3.

The character bit length within one data frame, the type of parity, and the stop bit length are specified by UARTB control register 0 (UBCTL0).

Also, data is transferred with LSB first/MSB first.

Figure 15-3. Asynchronous Serial Interface Transmit/Receive Data Format (LSB-First Transfer)



- 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

15.7.2 Transmit operation

In the single mode (UBFIC0.UBMOD bit = 0), transmission is enabled when the UBCTL0.UBTXE bit is set to 1, and transmission is started when transmit data is written to the UBTX register.

In the FIFO mode (UBFIC0.UBMOD bit = 1), transmission is started when transmit data of at least the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less is written to transmit FIFO and then the UBTXE bit is set to 1.

Caution Setting the UBCTL0.UBTXE bit to 1 before writing transmit data to transmit FIFO in the FIFO mode is prohibited. The operation is not guaranteed if this setting is made.

(1) Transmission enabled state

This state is set by the UBCTL0.UBTXE bit.

- UBTXE = 1: Transmission enabled state
- UBTXE = 0: Transmission disabled state

However, because this bit is also used by CSIB2, enable transmission after setting the CB2CTL0.CB2PWR bit to 0.

Since UARTB does not have a CTS (transmission enabled signal) input pin, a port should be used to confirm whether the destination is in the reception enabled state.

(2) Starting a transmit operation

- **In single mode (UBFIC0.UBMOD bit = 0)**
  In the single mode, transmission is started when transmit data is written to the UBTX register while transmission is enabled.
- **In FIFO mode (UBFIC0.UBMOD bit = 1)**
  In the FIFO mode, transmission is started when transmit data of at least the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less is written to transmit FIFO and then transmission is enabled (UBTXE bit = 1).

Data in the transmit data register (UBTX register in single mode or transmit FIFO in the FIFO mode) is transferred to the transmit shift register when transmission is started. Then, the transmit shift register outputs data to the TXDB pin sequentially beginning with the LSB (the transmit data is transferred sequentially starting with the start bit). The start bit, parity bit, and stop bits are added automatically.

(3) Transmission interrupt request signal

(a) Transmission end interrupt request signal (INTUBTIT)

- **In single mode (UBFIC0.UBMOD bit = 0)**
  In the single mode, the transmission end interrupt request signal (INTUBTIT) occurs when transmit data can be written to the UBTX register (when 1 byte of data is transferred from the UBTX register to the transmit shift register).

- **In FIFO mode (UBFIC0.UBMOD bit = 1)**
  In the FIFO mode, the INTUBTIT signal occurs when transmit data of the number set as the trigger specified by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits is transferred from transmit FIFO to the transmit shift register (if transmit data of the number set as the trigger can be written).

- **If pending mode is specified (UBFIC0.UBITM bit = 0) in FIFO mode**
  If the pending mode is specified in the FIFO mode, the second INTUBTIT signal is held pending after the first INTUBTIT signal has occurred, until as many transmit data as the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits are written to transmit FIFO, even if the generation condition of the second INTUBTIT signal is satisfied. When as many transmit data as the number set as the trigger are written to transmit FIFO in response to the first INTUBTIT signal, the second pending INTUBTIT signal is generated.

- **If pointer mode is specified (UBFIC0.UBITM bit = 1) in FIFO mode**
  If the pointer mode is specified in the FIFO mode, the second INTUBTIT signal occurs when the generation condition of the second INTUBTIT signal is satisfied even if as many transmit data as the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits are not written to transmit FIFO when the first INTUBTIT signal occurs.

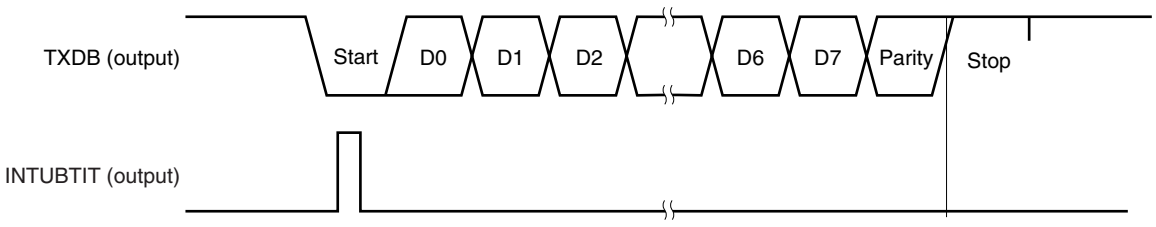
(b) FIFO transmission end interrupt request signal (INTUBTIF)

The FIFO transmission end interrupt request signal (INTUBTIF) occurs when no more data is in transmit FIFO and the transmit shift register in the FIFO mode (UBFIC0.UBMOD bit = 1). After the INTUBTIF signal has occurred, clear the pending INTUBTIT signal in the pending mode (UBFIC0.UBITM bit = 0) by clearing the FIFO (UBFIC0.UBTFC bit = 1). If the INTUBTIF signal occurs because writing the next transmit data to transmit FIFO is delayed (if all transmit data have not been transmitted), do not clear the FIFO.

If the data to be transmitted next has not been written to the transmit data register, the transmit operation is suspended.

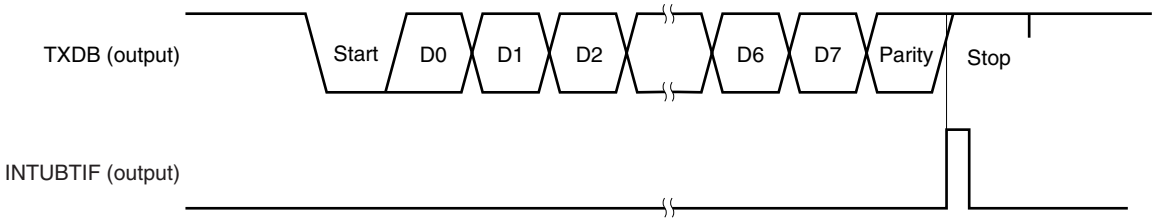
Caution In the single mode, the transmission end interrupt request signal (INTUBTIT) occurs when the UBTX register becomes empty (when 1 byte of data is transferred from the UBTX register to the transmit shift register). In the FIFO mode, the FIFO transmission end interrupt request signal (INTUBTIF) occurs when data is no longer in transmit FIFO and the transmit shift register (when the FIFO and register are empty). However, the INTUBTIT signal or INTUBTIF signal is not generated if the transmit data register becomes empty due to $\overline{\text{RESET}}$ input.

Figure 15-4. Timing of Asynchronous Serial Interface Transmission End Interrupt Request Signal (INTUBTIT)



Remark In the FIFO mode, the INTUBTIT signal occurs at the above timing when as many transmit data as the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits are serially transferred.

Figure 15-5. Timing of Asynchronous Serial Interface FIFO Transmission End Interrupt Request Signal (INTUBTIF)



Remark The INTUBTIF signal occurs at the above timing when data is no longer in transmit FIFO and the transmit shift register (when the FIFO and register are empty).

15.7.3 Continuous transmission operation

- **In single mode (UBFIC0.UBMOD bit = 0)**
  In the single mode, the next data can be written to the UBTX register as soon as the transmit shift register has started a shift operation. The timing of transfer can be identified by the transmission end interrupt request signal (INTUBTIT). By writing the next transmit data to the UBTX register via the INTUBTIT signal within one data frame transmission period, data can be transmitted without an interval and an efficient communication rate can be realized.

  Caution Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing. If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed.

- **If pending mode is specified (UBFIC0.UBITM bit = 0) in FIFO mode**
  If transmit data of at least the number set as the transmit trigger by UBFIC2.UBTT3 to UBFIC2.UBTT0 bits and 16 bytes or less is written to transmit FIFO, transmission starts.
  If the pending mode is specified in the FIFO mode, as many of the next transmit data as the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits can be written to transmit FIFO as soon as the transmit shift register has started shifting the last data of the specified number of data. The timing of transfer can be identified by the INTUBTIT signal. By writing as many of the next transmit data as the number set as the trigger to transmit FIFO or writing the data to the FIFO within the transmission period of the data in transmit FIFO via the INTUBTIT signal, data can be transmitted without an interval and an efficient communication rate can be realized.

  Caution Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing (this can also be done by the FIFO transmission end interrupt request signal (INTUBTIF)). If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed. To write transmit data to transmit FIFO by DMA, set the number of transmit data specified as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to 1 byte; otherwise the operation will not be guaranteed.

- **If pointer mode is specified (UBFIC0.UBITM bit = 1) in FIFO mode**
  If the pointer mode is specified in the FIFO mode, a INTUBTIT signal occurs and the next data can be written to transmit FIFO as soon as the transmit shift register has started shifting the number of transmit data set as the trigger. At this time, as many data as the number of empty bytes of transmit FIFO can be written by referencing the UBFIS1 register. The timing of transfer can be identified by the INTUBTIT signal. By writing as many of the next transmit data as the number specified as the trigger to transmit FIFO or writing the data to the FIFO within the transmission period of the data in transmit FIFO via the INTUBTIT signal, data can be transmitted without an interval and an efficient communication rate can be realized.

  Caution Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing (this can also be done by the FIFO transmission end interrupt request signal (INTUBTIF)). If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed.

15.7.4 Receive operation

The awaiting reception state is set by setting the UBCTL0.UBPWR bit to 1 and then setting the UBCTL0.UBRXE bit to 1. RXDB pin sampling begins and a start bit is detected. When the start bit is detected, the receive operation begins, and data is stored sequentially in the receive shift register according to the baud rate that was set.

In the single mode (UBFIC0.UBMOD bit = 0), a reception end interrupt request signal (INTUBTIR) is generated each time the reception of one frame of data is completed. Normally, the receive data is transferred from the UBRX register to memory by this interrupt servicing.

In the FIFO mode (UBFIC0.UBMOD bit = 1), the INTUBTIR signal occurs when as many receive data as the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits are transferred to receive FIFO.

If the pending mode is specified (UBFIC0.UBIRM bit = 0) in the FIFO mode, as many receive data as the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits can be read from receive FIFO.

If the pointer mode is specified (UBFIC0.UBIRM bit = 1) in the FIFO mode, as many data as the number of bytes stored in receive FIFO (0 bytes or more) can be read from receive FIFO by referencing the number of receive data specified as the trigger by the UBRT3 to UBRT0 bits (1 byte) or the UBFIS0 register.

Caution If the pointer mode is specified in the FIFO mode and if as many data as the number of bytes stored in receive FIFO are read by referencing the UBFIS0 register, no data may be stored in receive FIFO (UBFIS0.UBRB4 to UBFIS0.UBRB0 bits = 00000) even though the reception end interrupt request signal (INTUBTIR) has occurred. In this case, do not read data from receive FIFO. Be sure to read data from receive FIFO after confirming that the number of bytes stored in receive FIFO = 1 byte or more (UBRB4 to UBRB0 bits = other than 00000).

(1) Reception enabled state

This state is set by the UBCTL0.UBRXE bit.

- UBRXE = 1: Reception enabled state
- UBRXE = 0: Reception disabled state

However, because this bit is also used by CSIB2, enable reception after setting the CB2CTL0.CB2PWR bit to 0 and disabling the CSIB2 operation.

In the reception disabled state, the reception hardware stands by in the initial state. At this time, the reception end interrupt request signal or reception error interrupt request signal does not occur, and the contents of the receive data register (UBRX register in the single mode or receive FIFO in the FIFO mode (UBRXAP register)) are retained.

(2) Starting a receive operation

A receive operation is started by the detection of a start bit.

The RXDB pin is sampled using the serial clock from UARTB control register 2 (UBCTL2).

(3) Reception interrupt request signal

(a) Reception end interrupt request signal (INTUBTIR)

- **In single mode (UBFIC0.UBMOD bit = 0)**

  When UBCTL0.UBRXE bit = 1 and the reception of one frame of data is ended (the stop bit is detected) in the single mode, a reception end interrupt request signal (INTUBTIR) is generated and the receive data in the receive shift register is transferred to the UBRX register at the same time.

  Also, if an overrun error occurs, the receive data at that time is not transferred to the UBRX register, and a reception error interrupt request signal (INTUBTIRE) is generated.

  If a parity error or framing error occurs during the reception operation, the reception operation continues up to the position at which the stop bit is received. After completion of reception, an INTUBTIRE signal occurs (the receive data in the receive shift register is transferred to the UBRX register).

  If the UBRXE bit is reset (0) during a receive operation, the receive operation is immediately stopped. At this time, the contents of the UBRX register remain unchanged, the contents of the UARTB status register (UBSTR) are cleared, and the INTUBTIR and INTUBTIRE signals do not occur.

  No INTUBTIR signal is generated when the UBRXE bit = 0 (reception is disabled).

- **In FIFO mode (UBFIC0.UBMOD bit = 1)**

  In the FIFO mode, the reception end interrupt request signal (INTUBTIR) occurs when data of one frame has been received (stop bit is detected) and when as many receive data as the number specified as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits are transferred from the receive shift register to receive FIFO. If an overflow error occurs, the receive data is not transferred to receive FIFO and the reception error interrupt request signal (INTUBTIRE) occurs.

  If a parity error or framing error occurs during reception, reception continues up to the reception position of the stop bit. After reception has been completed, the INTUBTIRE signal occurs and the receive data in the receive shift register is transferred to receive FIFO. At this time, error information is appended as the UBRXAP.UBPEF or UBRXAP.UBFEF bit = 1. If the INTUBTIRE signal occurs, the error data can be recognized by reading receive FIFO as a 16-bit register, UBRXAP.

(b) Reception timeout interrupt request signal (INTUBTITO) (only in FIFO mode)

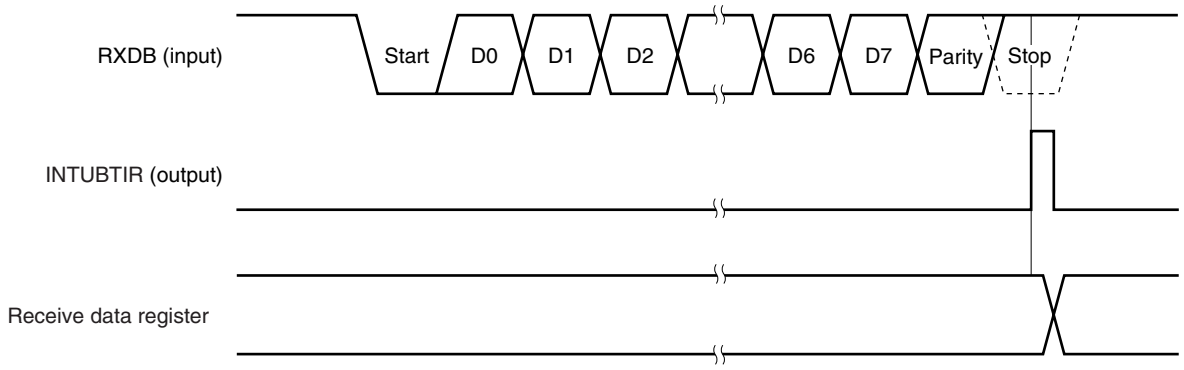
When the timeout counter function (UBFIC1.UBTCE bit = 1) is used in the FIFO mode, the reception timeout interrupt request signal (INTUBTITO) occurs if the next data does not come even after the next data reception wait time specified by the UBFIC1.UBTC4 to UBFIC1.UBTC0 bits has elapsed and if data is stored in receive FIFO.

The INTUBTITO signal does not occur while reception is disabled.

If as many receive data as the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits are not received, the timing of reading less receive data than the specified number can be set by the INTUBTITO signal.

Since the timeout counter starts counting at start bit detection, a receive timeout interrupt request signal does not occur if data of 1 character has not been received.

Figure 15-6. Timing of Asynchronous Serial Interface Reception End Interrupt Request Signal (INTUBTIR)



Cautions 1. Be sure to read all the data (the number of data indicated by the UBFIS0.UBRB4 to UBFIS0.UBRB0 bits) stored in the receive data register (UBRX register in the single mode or receive FIFO in the FIFO mode (UBRXAP register)) even when a reception error occurs.

Unless the receive data register is read, an overrun error occurs when the next data is received, causing the reception error status to persist.

If the pending mode is specified in the FIFO mode, however, be sure to clear the FIFO (UBFIC0.UBRFC bit = 1) after reading the data stored in receive FIFO.

In the FIFO mode, the FIFO can be cleared even without reading the data stored in receive FIFO.

If a parity error or framing error occurs in the FIFO mode, the UBRXAP register can be read in 16-bit (halfword) units.

2. Data is always received with one stop bit (1).

A second stop bit is ignored.

15.7.5 Reception error

In the single mode (UBFIC0.UBMOD bit = 0), the three types of errors that can occur during a receive operation are a parity error, framing error, and overrun error. In the FIFO mode (UBFIC0.UBMOD bit = 1), the three types of errors that can occur during a receive operation are a parity error, framing error, and overflow error.

As a result of data reception, the UBSTR.UBPE, UBSTR.UBFE, or UBSTR.UBOVE bit is set to 1 if a parity error, framing error, or overrun error occurs in the single mode. The UBSTR.UBOVF bit is set to 1 if an overflow error occurs in the FIFO mode. The UBRXAP.UBPEF or UBRXAP.UBFEF bit is set to 1 if a parity error or framing error occurs in the FIFO mode. At the same time, a reception error interrupt request signal (INTUBTIRE) occurs. The contents of the error can be detected by reading the contents of the UBSTR or UBRXAP register.

The contents of the UBSTR register are reset when 0 is written to the UBOVF, UBPE, UBFE, or UBOVE bit, or the UBCTL0.UBPWR or UBCTL0.UBRXE bit. The contents of the UBRXAP register are reset when 0 is written to the UBCTL0.UBPWR bit.

Table 15-4. Reception Error Causes

Error Flag	Valid Operation Mode	Error Flag	Reception Error	Cause
UBPE	Single mode	UBPE	Parity error	The parity specification during transmission does not match the parity of the receive data
UBFE		UBFE	Framing error	No stop bit detected
UBOVE		UBOVE	Overrun error	The reception of the next data is ended before data is read from the UBRX register
UBOVF	FIFO mode	UBOVF	Overflow error	The reception of the next data is ended while receive FIFO is full and before data is read.
UBPEF		UBPEF	Parity error	The parity specification during transmission does not match the parity of the data to be received.
UBFEF		UBFEF	Framing error	The stop bit is not detected when the target data is loaded.

15.7.6 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.

(1) Even parity

(a) 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

(b) 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.

(2) Odd parity

(a) 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

(b) 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.

(3) 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”.

(4) No parity

No parity bit is added to the transmit data.

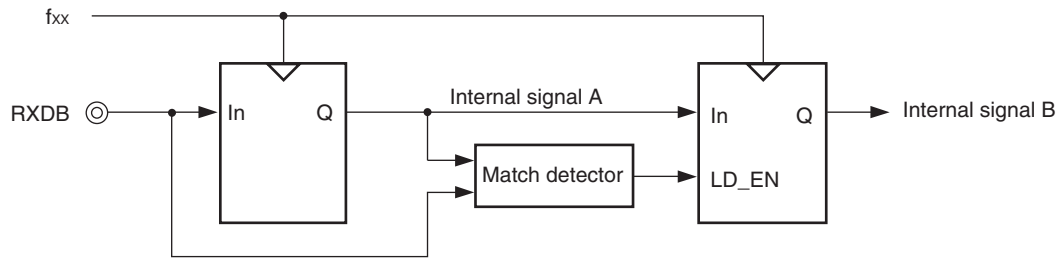
During reception, the receive operation is performed as if there were no parity bit. Since there is no parity bit, no parity error is generated.

15.7.7 Receive data noise filter

The RXDB signal is sampled at the rising edge of the peripheral clock (fxx). 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 15-8**).

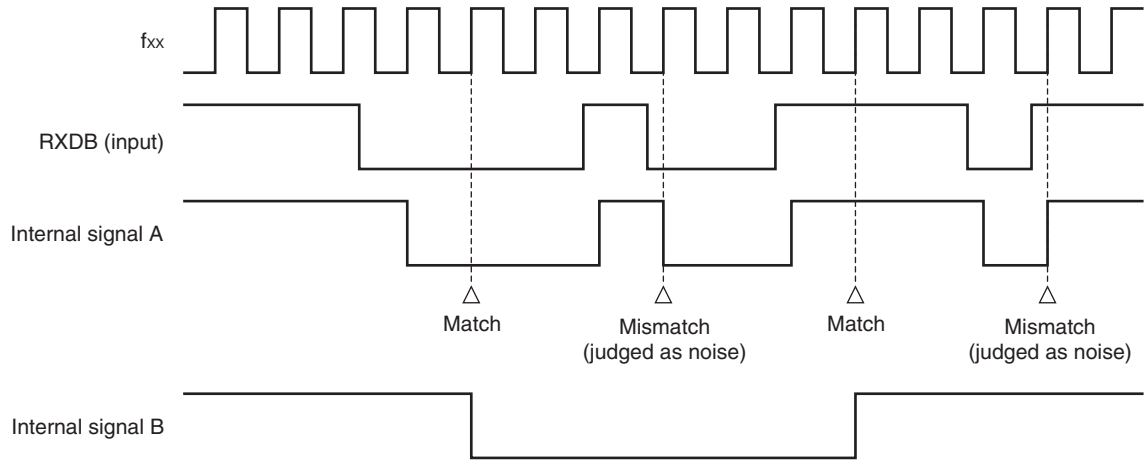
Also, since the circuit is configured as shown in Figure 15-7, internal processing during a receive operation is delayed by up to 2 clocks according to the external signal status.

Figure 15-7. Noise Filter Circuit



Remark fxx: Peripheral clock

Figure 15-8. Timing of RXDB Signal Judged as Noise



Remark fxx: Peripheral clock

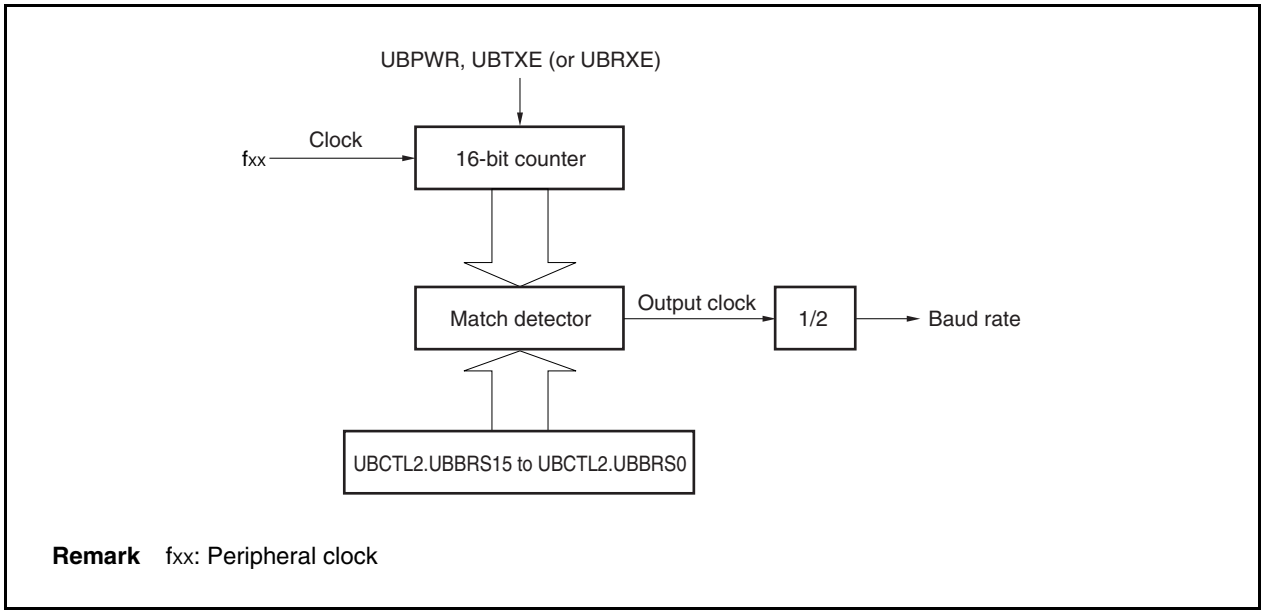
15.8 Dedicated Baud Rate Generator (BRG)

A dedicated baud rate generator, which consists of a 16-bit programmable counter, generates serial clocks during transmission/reception in UARTB. The dedicated baud rate generator output can be selected as the serial clock for each channel.

Separate 16-bit counters exist for transmission and for reception. The baud rate for transmission/reception is the same at the same channel.

(1) Baud rate generator configuration

Figure 15-9. Baud Rate Generator Configuration



Remark fxx: Peripheral clock

(a) Base clock (Clock)

When UBCTL0.UBPWR bit = 1, the peripheral clock (fxx) is supplied to the transmission/reception unit. This clock is called the base clock. When the UBPWR bit = 0, the clock signal is fixed at low level.

(2) Serial clock generation

A serial clock can be generated according to the settings of the UBCTL2 register.

The 16-bit counter divisor value can be selected according to the UBCTL2.UBBRS15 to UBCTL2.UBBRS0 bits.

(a) Baud rate

The baud rate is the value obtained according to the following formula.

$$\text{Baud rate} = \frac{\text{Base clock frequency}}{2 \times k} \text{ [bps]}$$

Base clock frequency = fxx

k = Value set according to UBCTL2.UBBRS15 to UBCTL2.UBBRS0 bits (k = 4, 5, 6, ..., 65535)

(b) Baud rate error

The baud rate error is obtained according to the following formula.

$$\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (normal baud rate)}} - 1 \right) \times 100 \text{ [\%]}$$

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 paragraph (4).

Example: Base clock (fxx) = 64 MHz = 64,000,000 Hz

Settings of UBCTL2.UBBRS15 to UBCTL2.UBBRS0 bits = 0000000001100110B (k = 102)

Target baud rate = 312500 bps

Baud rate = 64 M/(2 × 102)

= 64000000/(2 × 102) = 313725 [bps]

Error = (313725/312500 − 1) × 100

= 0.392 [%]

When base clock (fxx) = 60 MHz and k = 96, the error is 0%.

(3) Baud rate setting example

Table 15-5. Baud Rate Generator Setting Data (1/2)

Baud Rate (bps)	fxx = 64 MHz			fxx = 60 MHz			fxx = 50 MHz		
	k (Decimal)	k (Hexadecimal)	ERR	k (Decimal)	k (Hexadecimal)	ERR	k (Decimal)	k (Hexadecimal)	ERR
300	–	–	–	–	–	–	–	–	–
600	53333	D055H	0.001	50000	C350H	0.000	41667	A2C3H	–0.001
1200	26667	682BH	–0.001	25000	61A8H	0.000	20833	5161H	0.002
2400	13333	3415H	0.003	12500	30D4H	0.000	10417	28B1H	–0.003
4800	6667	1A0BH	–0.005	6250	186AH	0.000	5208	1458H	0.006
9600	3333	0D05H	0.010	3125	0C35H	0.000	2604	0A2CH	0.006
19200	1667	0683H	–0.020	1563	061BH	0.000	1302	0516H	0.006
31250	1024	0400H	0.000	960	03C0H	0.000	800	0320H	0.000
38400	833	0341H	0.040	781	030DH	0.000	651	028BH	0.006
76800	417	01A1H	–0.080	391	0187H	0.000	326	0146H	–0.147
153600	208	00D0H	0.160	195	00C3H	0.000	163	00A3H	–0.147
312500	102	0066H	0.392	96	0060H	0.000	80	0050H	0.000
500000	64	0040H	0.000	60	003CH	0.000	50	0032H	0.000
1000000	32	0020H	0.000	30	001EH	0.000	25	0019H	0.000
2000000	16	0010H	0.000	15	000FH	0.000	13	000DH	–3.846
3000000	11	000BH	–3.030	10	000AH	0.000	8	0008H	4.167
4000000	8	0008H	0.000	8	0008H	–6.250	–	–	–
5000000	6	0006H	6.667	6	0006H	0.000	–	–	–
5333333	6	0006H	0.000	–	–	–	–	–	–

Caution The maximum allowable frequency of the peripheral clock (fxx) is 64 MHz.
The maximum transfer speed of the baud rate is 5.33 Mbps.

Remark fxx: Peripheral clock
k: Settings of UBCTL2.UBBRS15 to UBCTL2.UBBRS0 bits
ERR: Baud rate error [%]

Table 15-5. Baud Rate Generator Setting Data (2/2)

Baud Rate (bps)	f_{XX} = 40 MHz			f_{XX} = 32 MHz		
	k (Decimal)	k (Hexadecimal)	ERR	k (Decimal)	k (Hexadecimal)	ERR
300	–	–	–	53333	D055H	0.001
600	33333	8235H	0.001	26667	682BH	–0.001
1200	16667	411BH	–0.002	13333	3415H	0.003
2400	8333	208DH	0.004	6667	1A0BH	–0.005
4800	4167	1047H	–0.008	3333	0D05H	0.010
9600	2083	0823H	0.016	1667	0683H	–0.020
19200	1042	0412H	–0.032	833	0341H	0.040
31250	640	0280H	0.000	512	0200H	0.000
38400	521	0209H	–0.032	417	01A1H	–0.080
76800	260	0104H	0.160	208	00D0H	0.160
153600	130	0082H	0.160	104	0068H	0.160
312500	64	0040H	0.000	51	0033H	0.392
500000	40	0028H	0.000	32	0020H	0.000
1000000	20	0014H	0.000	16	0010H	0.000
2000000	10	000AH	0.000	8	0008H	0.000
3000000	7	0007H	–4.762	–	–	–
4000000	–	–	–	–	–	–
5000000	–	–	–	–	–	–
5333333	–	–	–	–	–	–

Caution The maximum allowable frequency of the peripheral clock (f_{XX}) is 64 MHz. The maximum transfer speed of the baud rate is 5.33 Mbps.

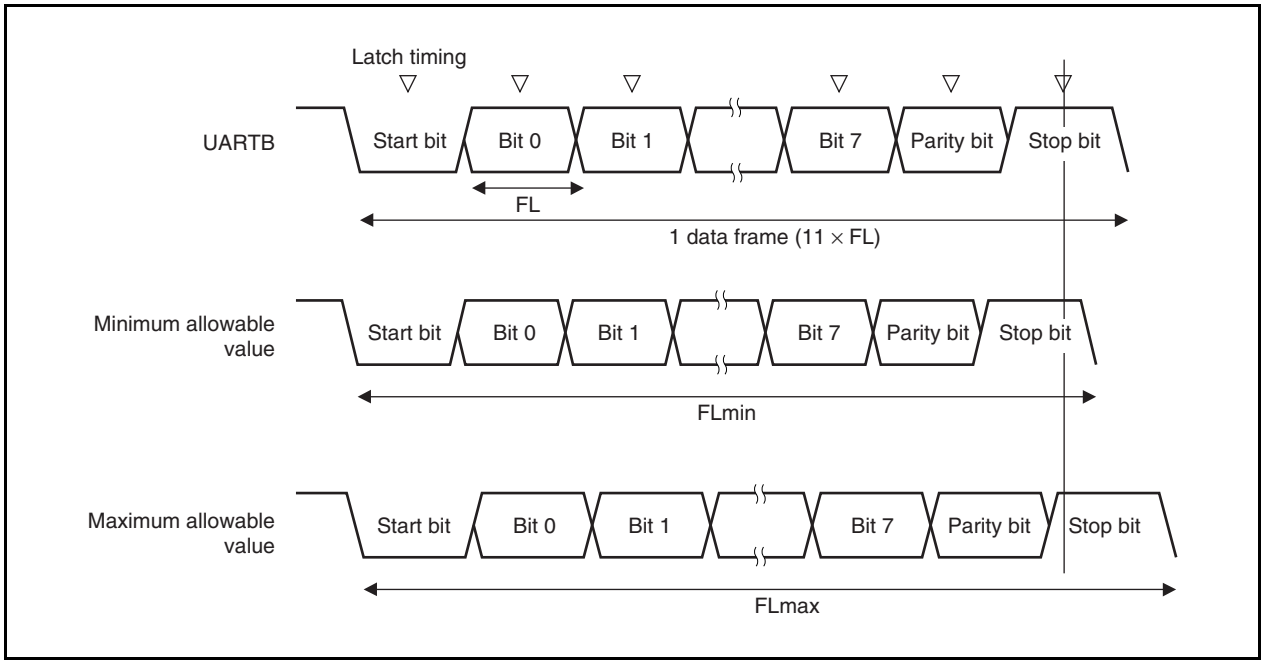
Remark f_{XX}: Peripheral clock
k: Settings of UBCTL2.UBBRS15 to UBCTL2.UBBRS0 bits
ERR: Baud rate error [%]

(4) Allowable baud rate range during reception

The degree to which a discrepancy from the transmission destination's baud rate is allowed during reception is shown below.

Caution The equations described below should be used to set the baud rate error during reception so that it always is within the allowable error range.

Figure 15-10. Allowable Baud Rate Range During Reception



As shown in Figure 15-10, after the start bit is detected, the receive data latch timing is determined according to the counter that was set by the UBCTL2 register. If all data up to the final data (stop bit) is in time for this latch timing, the data can be received normally.

Applying this to 11-bit reception is, theoretically, as follows.

$FL = (Brate)^{-1}$

Brate: UARTB baud rate
k: UBCTL2 set value
FL: 1-bit data length
Latch timing margin: 2 clocks

$$\text{Minimum allowable value: } FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$$

Therefore, the maximum baud rate that can be received at the transfer destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2} Brate$$

Similarly, the maximum allowable value 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 minimum baud rate that can be received at the transfer destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k - 2} Brate$$

The allowable baud rate error of UARTB and the transfer destination can be obtained as follows from the expressions described above for computing the minimum and maximum baud rate values.

Table 15-6. Maximum and Minimum Allowable Baud Rate Error

Division Ratio (k)	Maximum Allowable Baud Rate Error	Minimum Allowable Baud Rate Error
4	+2.33 %	−2.44
8	+3.53 %	−3.61
16	+4.14 %	−4.19
32	+4.45 %	−4.48
64	+4.61 %	−4.62
128	+4.68 %	−4.69
256	+4.72 %	−4.73
512	+4.74 %	−4.74
1024	+4.75 %	−4.75
2048	+4.76 %	−4.76
4096	+4.76 %	−4.76
8192	+4.76 %	−4.76
16384	+4.76 %	−4.76
32768	+4.76 %	−4.76
65535	+4.76 %	−4.76

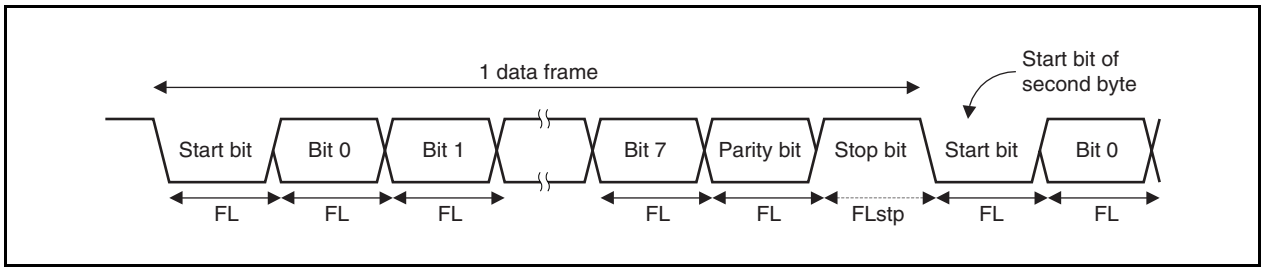
Remarks 1. The reception precision depends on the number of bits in one frame, the base clock frequency, and the division ratio (k). The higher the base clock frequency and the larger the division ratio (k), the higher the precision.

2. k: UBCTL2 set 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 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 15-11. 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_{XX} yields the following equation.

$$FLstp = FL + 2/(f_{XX})$$

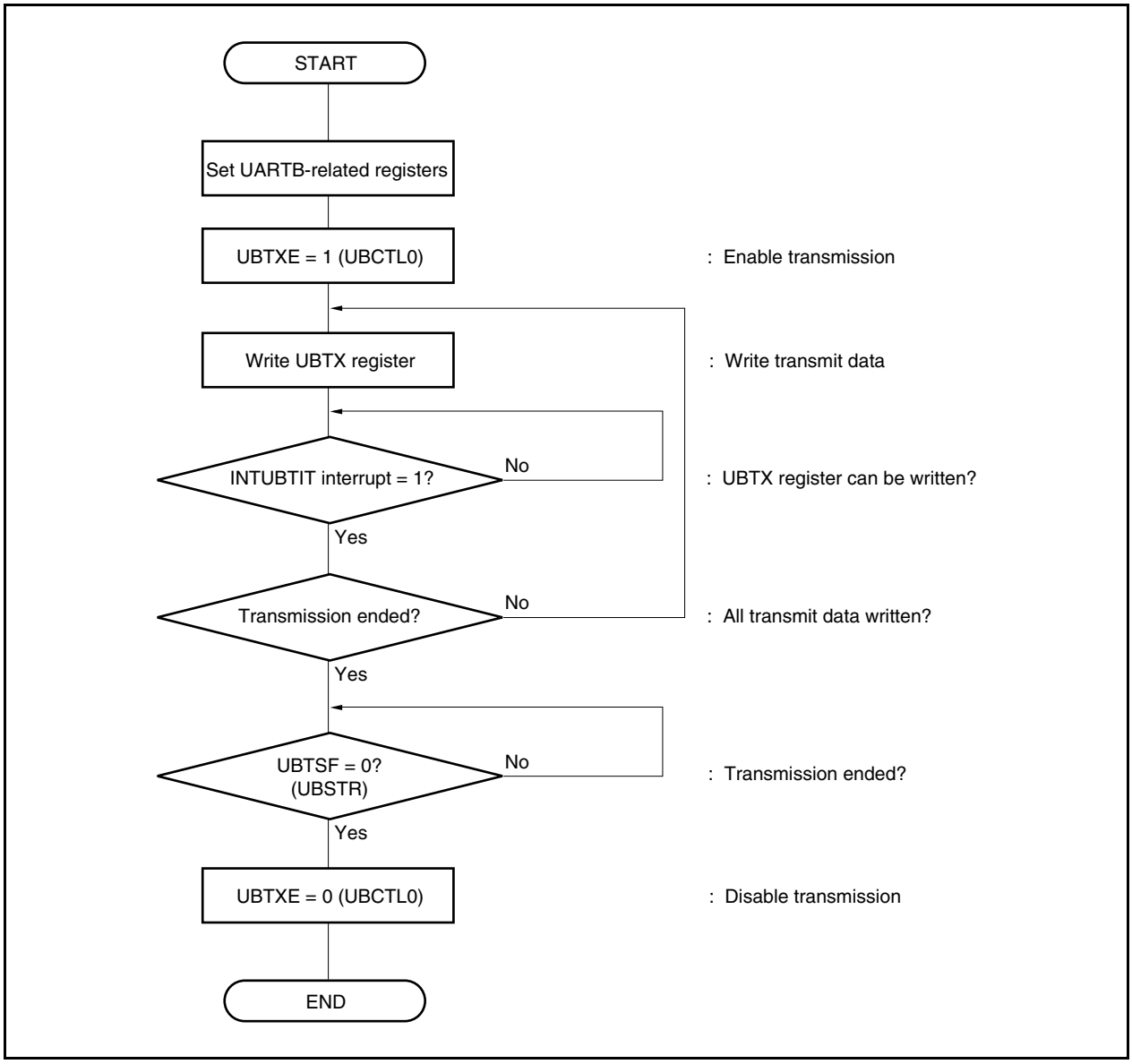
Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times FL + 2/(f_{XX})$$

15.9 Control Flow

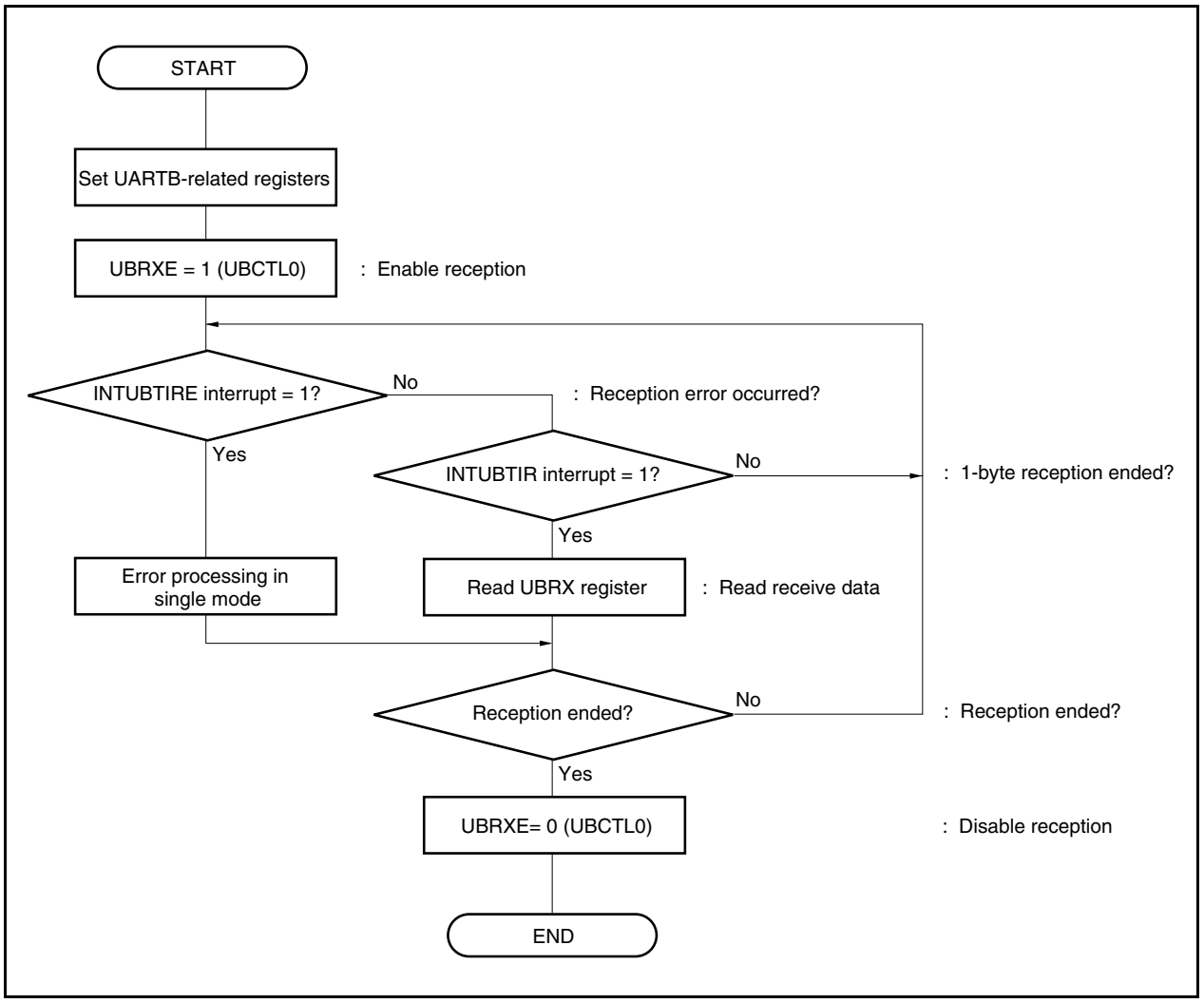
(1) Example of continuous transmission processing flow in single mode (CPU control)

Figure 15-12. Example of Continuous Transmission Processing Flow in Single Mode (CPU Control)



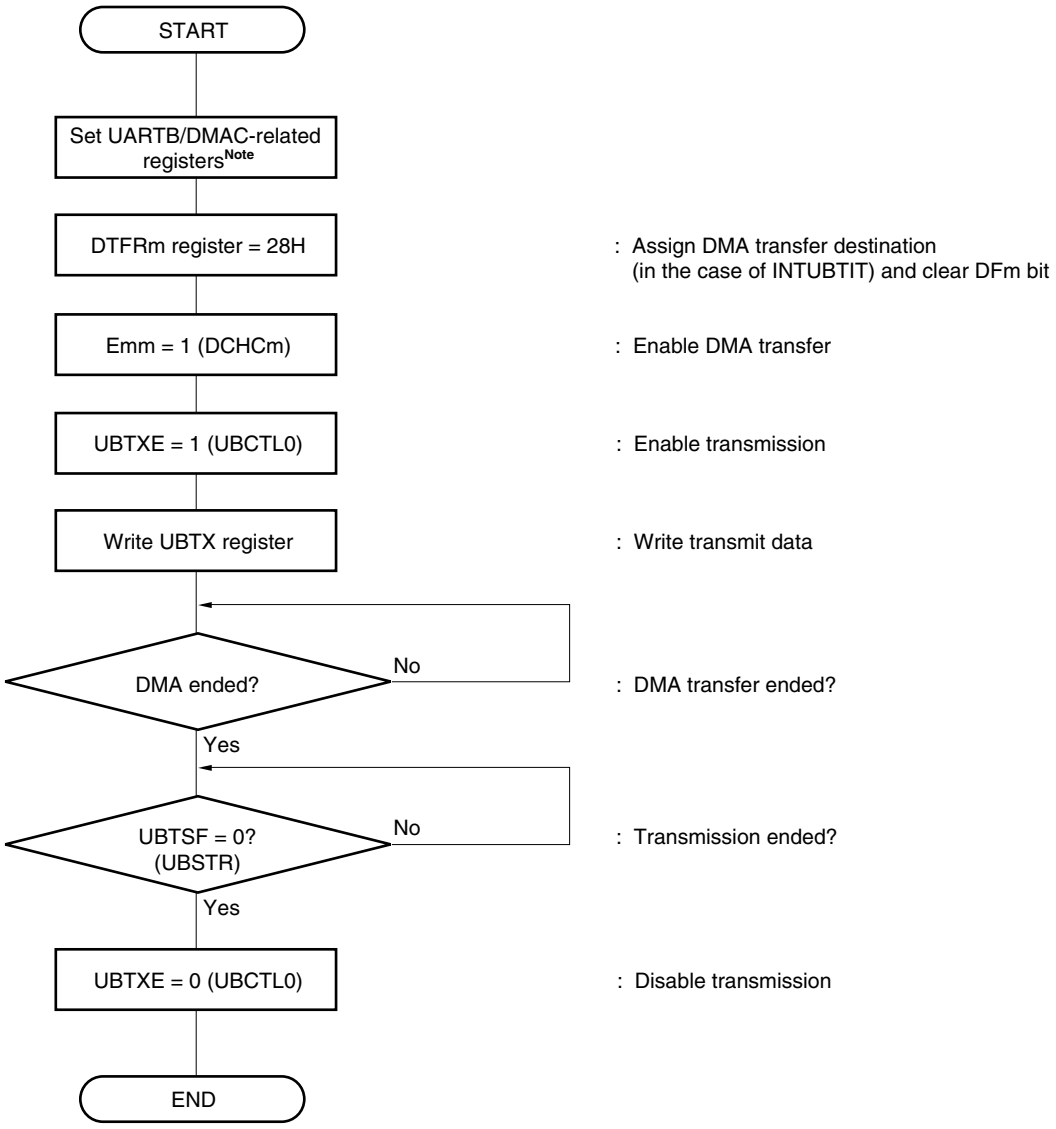
(2) Example of continuous reception processing flow in single mode (CPU control)

Figure 15-13. Example of Continuous Reception Processing Flow in Single Mode (CPU Control)



(3) Example of continuous transmission processing flow in single mode (DMA control)

Figure 15-14. Example of Continuous Transmission Processing Flow in Single Mode (DMA Control)

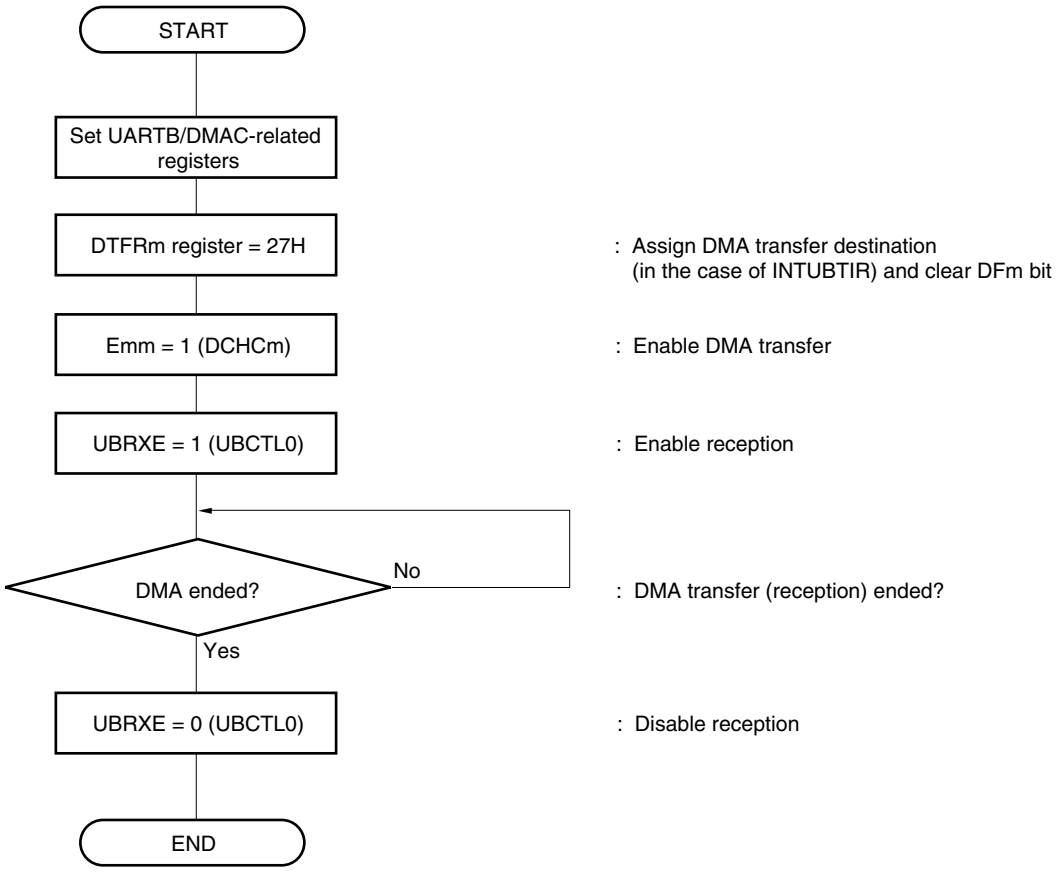


Note In this control flow example, transmission of the first byte of the data is executed by a CPU write operation. Exercise care in setting the number of data for DMA transfer (DBCm register) and the source address (DSAmH, DSAmL registers).

Remark m = 0 to 3

(4) Example of continuous reception processing flow in single mode (DMA control)

Figure 15-15. Example of Continuous Reception Processing Flow in Single Mode (DMA Control)



Remark m = 0 to 3

(5) Example of continuous transmission processing flow in FIFO mode (CPU control)

Figure 15-16. Example of Continuous Transmission Processing Flow in FIFO Mode (CPU Control)

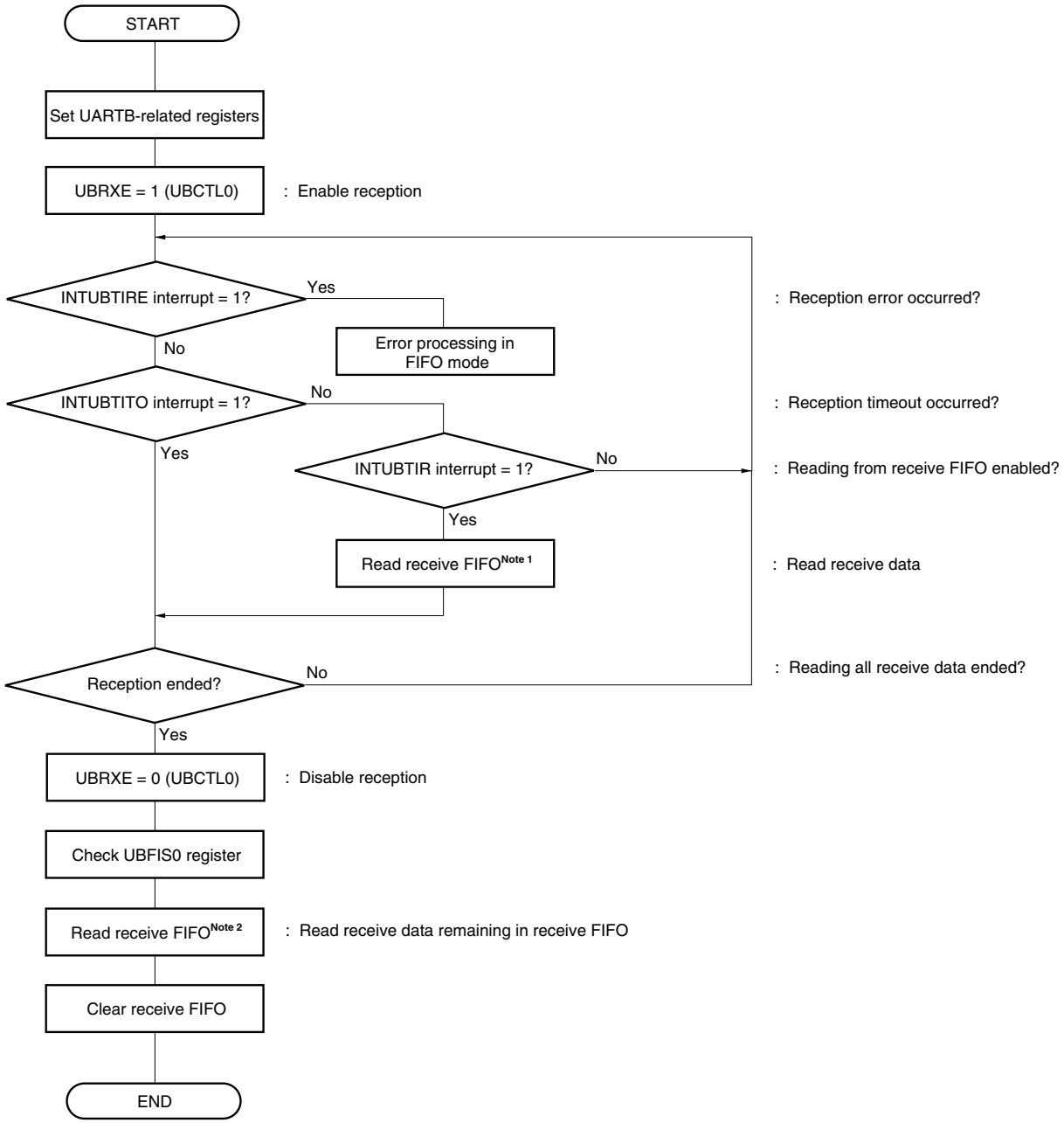
Notes 1. Write more transmit data than the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to transmit FIFO.

2. This is the case where transmission is ended (transmit FIFO and the transmit shift register become empty) before the next transmit data is written. To continue data transmission, clear the INTUBTIF and INTUBTIT signals and write the next data to transmit FIFO.

3. In the pending mode (UBFIC0.UBITM bit = 0), write as many transmit data as the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits of to transmit FIFO. In the pointer mode (UBITM bit = 1), reference the UBFIS1.UBTB4 to UBFIS1.UBTB0 bits and write as many data as the number of empty bytes in transmit FIFO to transmit FIFO.
Write 16-byte data to fully use the 8-bit × 16-stage FIFO function.

(6) Example of continuous reception processing in FIFO mode (CPU control)

Figure 15-17. Example of Continuous Reception Processing in FIFO Mode (CPU Control)

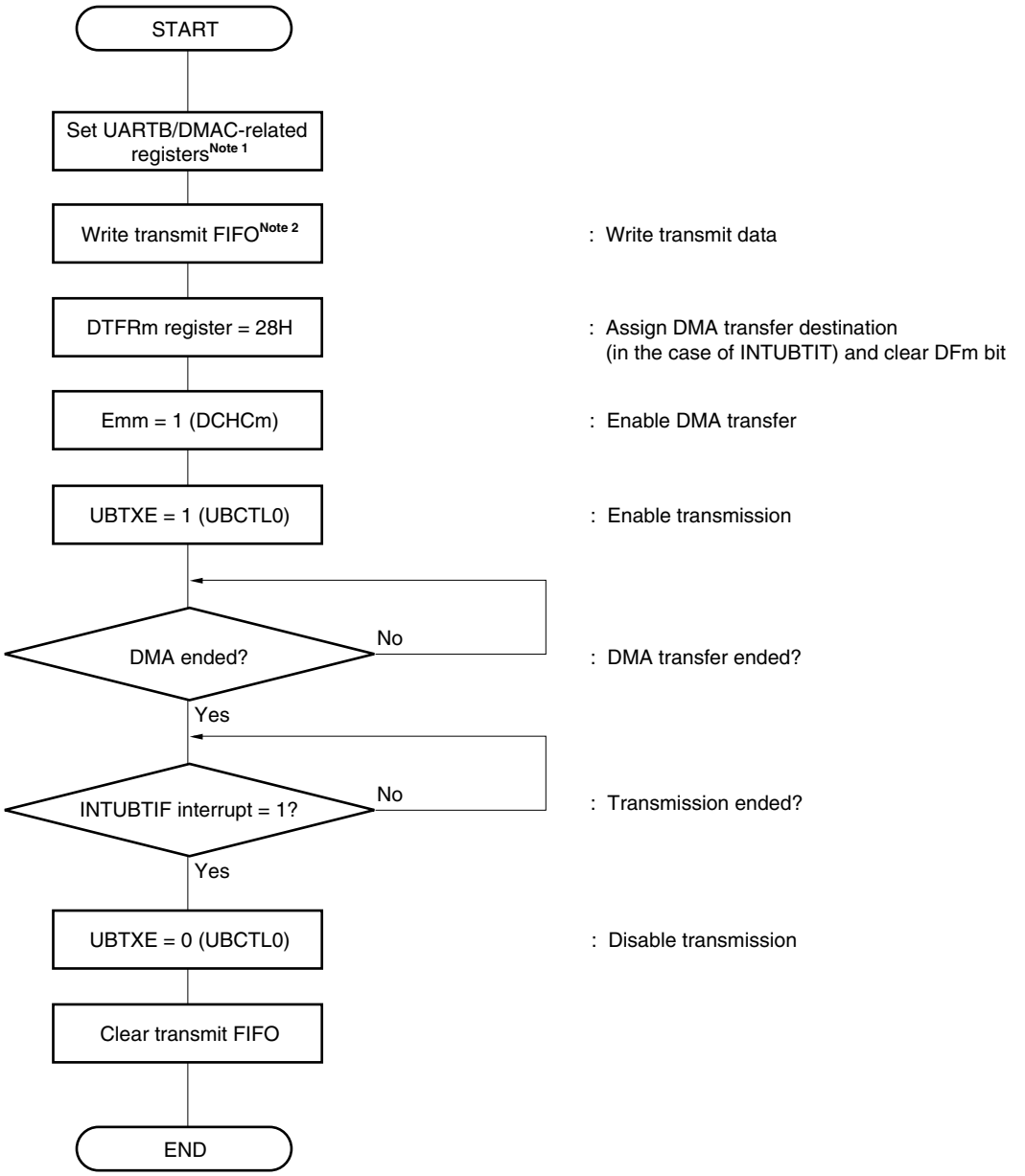


Notes 1. Read as many receive data as the number set as the trigger by the UBFIC2.UBRT3 to UBFIC2.UBRT0 bits from receive FIFO in the pending mode (UBFIC0.UBIRM bit = 0). In the pointer mode (UBIRM bit = 1), reference the UBFIS0.UBRB4 to UBFIS0.UBRB0 bits and read as many data as the number of bytes stored in receive FIFO from receive FIFO.

2. Read as many data (remaining receive data less than the number set as the trigger) as the number of bytes stored in receive FIFO from receive FIFO by referencing the UBFIS0.UBRB4 to UBFIS0.UBRB0 bits.

(7) Example of continuous transmission (pending mode) processing in FIFO mode (DMA control)

Figure 15-18. Example of Continuous Transmission (Pending Mode) Processing in FIFO Mode (DMA Control)



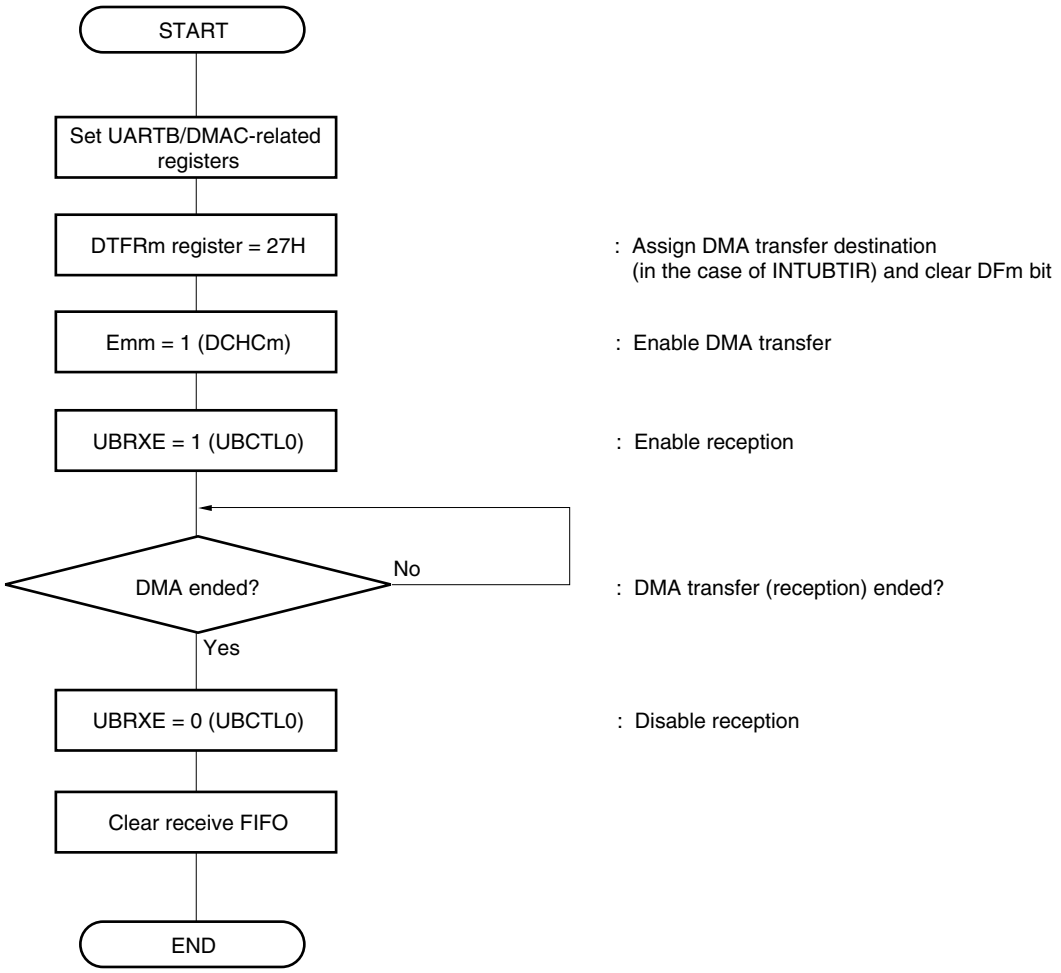
Notes 1. In this control flow example, transmission of the data described in **Note 2** is executed by a CPU write operation. Exercise care in setting the number of data for DMA transfer (DBCm register) and the source address (DSAmH, DSAmL registers).

2. Write as many transmit data as the number set as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits (= 1 byte) to transmit FIFO.

Remark m = 0 to 3

(8) Example of continuous reception (pending mode) processing flow in FIFO mode (DMA control)

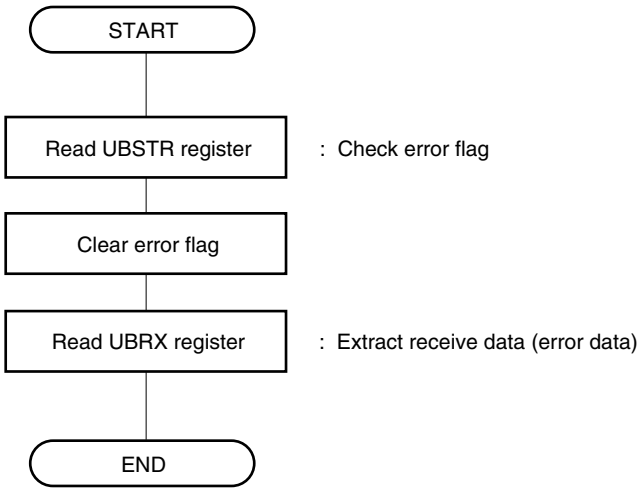
Figure 15-19. Example of Continuous Reception (Pending Mode) Processing Flow in FIFO Mode (DMA Control)



Remark m = 0 to 3

(9) Example of reception error processing in single mode

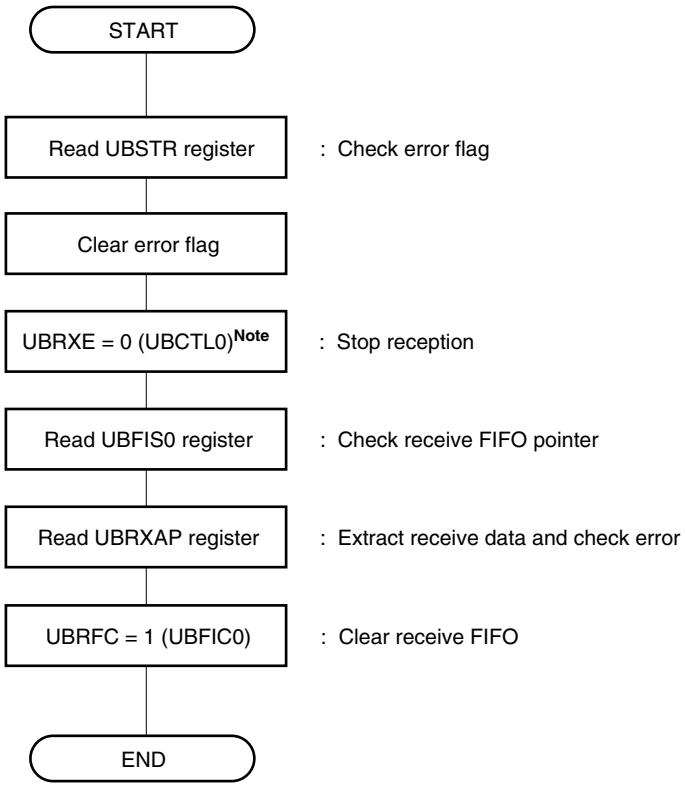
Figure 15-20. Example of Reception Error Processing Flow in Single Mode



Caution Reception can be continued by completing this control flow before reception of the next data is ended. If the next data is received before this control flow is ended, a reception error interrupt request signal (INTUBTIRE) may occur even if the data has been received correctly.

(10) Example of reception error processing flow in FIFO mode (1)

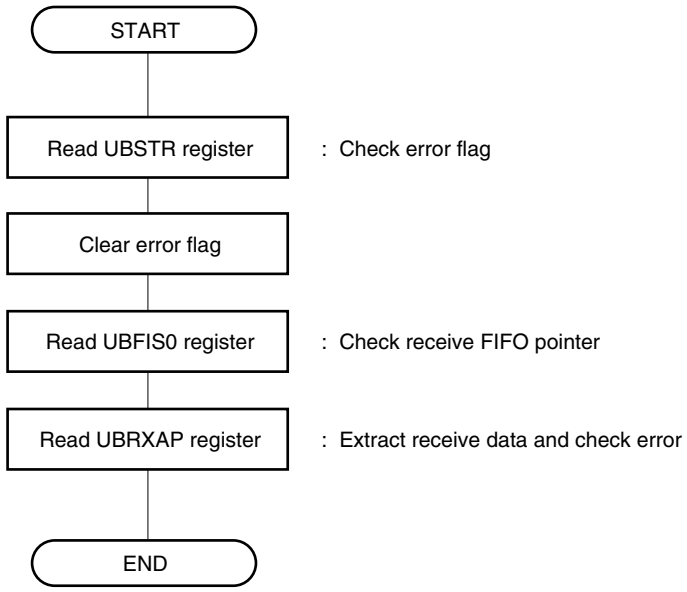
Figure 15-21. Example of Reception Error Processing Flow in FIFO Mode (1)



Note If the error flag is cleared when UBRXE bit = 0, the UBCTL0 register does not have to be set.

(11) Example of reception error processing flow in FIFO mode (2)

Figure 15-22. Example of Reception Error Processing Flow in FIFO Mode (2)



Note Reception can be continued by completing this control flow before reception of the next data is ended. Extract the receive data and check if a reception error has occurred before receive FIFO becomes empty. Note that this control flow is valid only when a parity error or a framing error occurs. If an overflow error occurs, receive FIFO must be cleared (UBFIC0.UBRFC bit = 1).

If the next data is received before this control flow is ended, a reception error interrupt request signal (INTUBTIRE) may occur even if the data has been received correctly.

15.10 Cautions

Cautions concerning UARTB are shown below.

(1) When supply clock to UARTB is stopped

When the supply of clocks to UARTB is stopped (for example, IDLE and STOP modes), operation stops with each register retaining the value it had immediately before the supply of clocks was stopped. The TXDB pin output also holds and outputs the value it had immediately before the supply of clocks was stopped. However, operation is not guaranteed after the supply of clocks is restarted. Therefore, after the supply of clocks is restarted, the circuits should be initialized by setting the UBPWR bit = 0, UBRXE bit = 0, and UBTXE bit = 0.

(2) Caution on setting UBCTL0 register

- When using UARTB, set the external pins related to the UARTB function to the alternate function and set the UBCTL2 register. Then set the UBCTL0.UBPWR bit to 1 before setting the other bits.
- Be sure to input a high level to the RXDB pin when setting the external pins related to the UARTB function to the alternate function. If a low level is input, it is judged that a falling edge is input after the UBCTL0.UBRXE bit has been set to 1, and reception may be started.

(3) Caution on setting UBFIC2 register

Be sure to clear the UBCTL0.UBTXE bit (to disable transmission) and UBCTL0.UBRXE bit (to disable reception) to 0 before writing data to the UBFIC2 register. If data is written to the UBFIC2 register with the UBTXE or UBRXE bit set to 1, the operation is not guaranteed.

(4) Transmission interrupt request signal

In the single mode, the transmission end interrupt request signal (INTUBTIT) occurs when the UBTX register becomes empty (when 1 byte of data is transferred from the UBTX register to the transmit shift register). In the FIFO mode, the FIFO transmission end interrupt request signal (INTUBTIF) occurs when data is no longer in transmit FIFO and the transmit shift register (when the FIFO and register are empty). However, the INTUBTIT signal or INTUBTIF signal does not occur if the transmit data register becomes empty due to $\overline{\text{RESET}}$ input.

(5) Initialization during continuous transmission in single mode

Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing. If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed.

(6) Initialization during continuous transmission (pending mode) in FIFO mode

Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing (this can also be done by checking the FIFO transmission end interrupt request signal (INTUBTIF)). If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed.

To write transmit data to transmit FIFO by DMA control, set the number of transmit data specified as the trigger by the UBFIC2.UBTT3 to UBFIC2.UBTT0 bits to 1 byte; otherwise the operation will not be guaranteed.

(7) Initialization during continuous transmission (pointer mode) in FIFO mode

Confirm that the UBSTR.UBTSF bit is 0 before executing initialization during transmission processing (this can also be done by checking the FIFO transmission end interrupt request signal (INTUBTIF)). If initialization is executed while the UBTSF bit is 1, the transmit data is not guaranteed.

(8) Receive operation in FIFO mode (pointer mode specified)

If the pointer mode is specified in the FIFO mode and if as many data as the number of bytes stored in receive FIFO are read by referencing the UBFIS0 register, no data may be stored in receive FIFO (UBFIS0.UBRB4 to UBFIS0.UBRB0 bits = 00000) even though the reception end interrupt request signal (INTUBTIR) has occurred. In this case, do not read data from receive FIFO. Be sure to read data from receive FIFO after confirming that the number of bytes stored in receive FIFO = 1 byte or more (UBRB4 to UBRB0 bits = other than 00000).

CHAPTER 16 CLOCKED SERIAL INTERFACE B (CSIB)

16.1 Mode Switching Between CSIB and Other Serial Interface

16.1.1 Mode switching between CSIB0 and UARTA0

In the V850E/IF3 and V850E/IG3, CSIB0 and UARTA0 function alternately, and these functions cannot be used at the same time. To use CSIB0 and UARTA0, the PMC4, PFC4, and PFCE4 registers must be set in advance.

Caution The operations related to transmission and reception of CSIB0 or UARTA0 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 16-1. Mode Switch Settings of CSIB0 and UARTA0

After reset: 00H R/W Address: FFFFF448H

	7	6	5	4	3	2	1	0
PMC4	PMC47	PMC46	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40

After reset: 00H R/W Address: FFFFF468H

	7	6	5	4	3	2	1	0
PFC4	PFC47	PFC46	PFC45	PFC44	PFC43	PFC42	PFC41	PFC40

After reset: 00H R/W Address: FFFFF708H

	7	6	5	4	3	2	1	0
PFCE4	PFCE47	PFCE46	PFCE45	PFCE44	PFCE43	PFCE42	PFCE41	PFCE40

PMC42	PFCE42	PFC42	Specification of alternate function of P42 pin
0	×	×	I/O port
1	0	0	$\overline{\text{SCKB0}}$ I/O
1	0	1	INTP13 input
1	1	0	Setting prohibited
1	1	1	Setting prohibited

PMC41	PFCE41	PFC41	Specification of alternate function of P41 pin
0	×	×	I/O port
1	0	0	SOB0 output
1	0	1	TXDA0 output
1	1	0	Setting prohibited
1	1	1	Setting prohibited

PMC40	PFC40	Specification of alternate function of P40 pin
0	×	I/O port
1	0	SIB0 input
1	0	RXDA0 input

Remark × = don't care

16.1.2 Mode switching between CSIB1 and UARTA2

In the V850E/IF3 and V850E/IG3, CSIB1 and UARTA2 function alternately, and these functions cannot be used at the same time. To use CSIB1 and UARTA2, the PMC3, PFC3, and PFCE3 registers must be set in advance.

Caution The operations related to transmission and reception of CSIB1 or UARTA2 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 16-2. Mode Switch Settings of CSIB1 and UARTA2

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	PFCE37	PFCE36	PFCE35	PFCE34	0	PFCE32	PFCE31	PFCE30

PMC34	PFCE34	PFC34	Specification of alternate function of P34 pin
0	×	×	I/O port
1	0	0	$\overline{\text{SCKB1}}$ I/O
1	0	1	INTP11 input
1	1	0	$\overline{\text{CS0}}$ Note output
1	1	1	Setting prohibited

PMC33	PFC33	Specification of alternate function of P33 pin
0	×	I/O port
1	0	SOB1 output
1	0	TXDA2 output

PMC32	PFCE32	PFC32	Specification of alternate function of P32 pin
0	×	×	I/O port
1	0	0	SIB1 input
1	0	1	RXDA2 input
1	1	0	$\overline{\text{CS1}}$ Note output
1	1	1	Setting prohibited

Note μPD70F3454GC-8EA-A only

Remark × = don't care

16.1.3 Mode switching between CSIB2 and UARTB

In the V850E/IF3 and V850E/IG3, CSIB2 and UARTB function alternately, and these functions cannot be used at the same time. To use CSIB2 and UARTB, the PMC3, PFC3, and PFCE3 registers must be set in advance.

Caution The operations related to transmission and reception of CSIB2 or UARTB are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 16-3. Mode Switch Settings of CSIB2 and UARTB

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	PFCE37	PFCE36	PFCE35	PFCE34	0	PFCE32	PFCE31	PFCE30

PMC37	PFCE37	PFC37	Specification of alternate function of P37 pin
0	×	×	I/O port
1	0	0	$\overline{\text{SCKB2}}$ I/O
1	0	1	INTP12 input
1	1	0	ASTB[Note] output
1	1	1	Setting prohibited

PMC36	PFCE36	PFC36	Specification of alternate function of P36 pin
0	×	×	I/O port
1	0	0	SOB2 output
1	0	1	TXDB output
1	1	0	Setting prohibited
1	1	1	Setting prohibited

PMC35	PFCE35	PFC35	Specification of alternate function of P35 pin
0	×	×	I/O port
1	0	0	SIB2 input
1	0	1	RXDB input
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Note μPD70F3454GC-8EA-A only

Remark × = don't care

16.2 Features

- Transfer rate: 8 Mbps (using internal clock)
- Master mode and slave mode selectable
- 8-bit to 16-bit transfer, 3-wire serial interface
- Interrupt request signals (INTCBnRE, INTCBnT, INTCBnR)
- Serial clock and data phase switchable
- Transfer data length selectable in 1-bit units between 8 and 16 bits
- Transfer data MSB-first/LSB-first switchable
- 3-wire transfer SOBn: Serial data output
  SIBn: Serial data input
  $\overline{\text{SCKBn}}$: Serial clock I/O

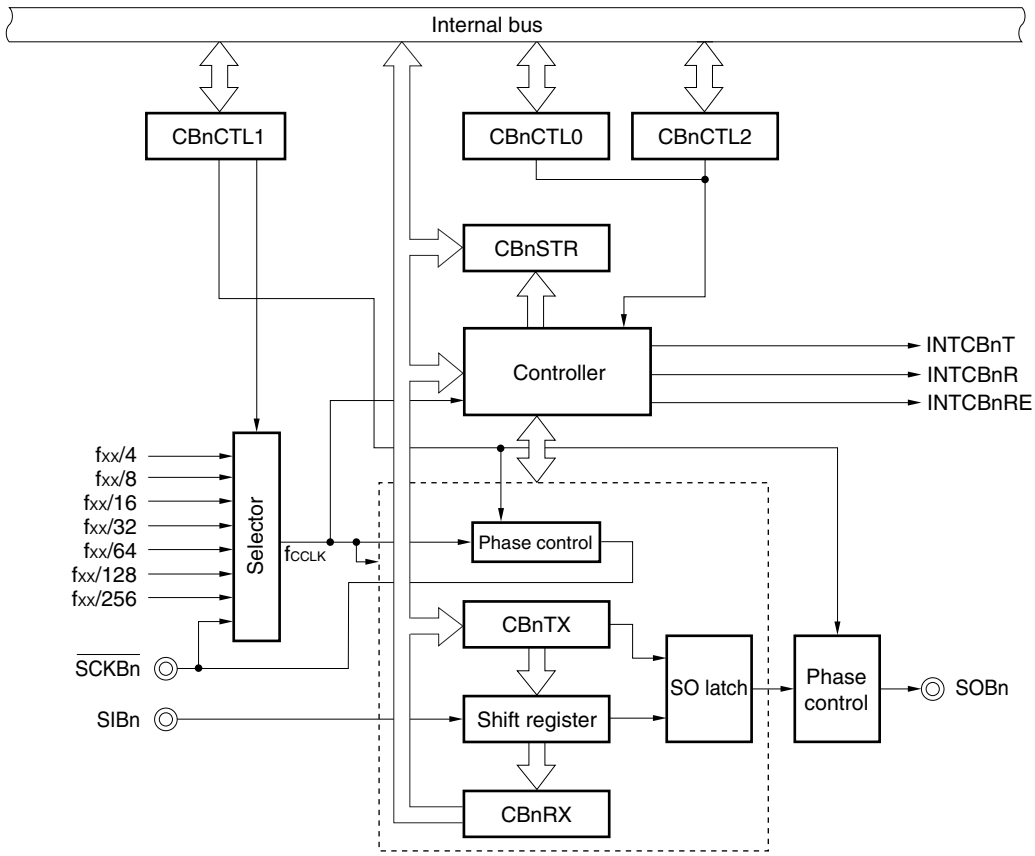
  Transmission mode, reception mode, and transmission/reception mode specifiable

Remark n = 0 to 2

16.3 Configuration

The following shows the block diagram of CSIBn.

Figure 16-4. Block Diagram of CSIBn



Remarks 1. n = 0 to 2

2. f_{CCLK}: Communication clock (8 MHz (max.))

CSIBn includes the following hardware.

Table 16-1. Configuration of CSIBn

Item	Configuration
Registers	CSIBn receive data register (CBnRX) CSIBn transmit data register (CBnTX)
Control registers	CSIBn control register 0 (CBnCTL0) CSIBn control register 1 (CBnCTL1) CSIBn control register 2 (CBnCTL2) CSIBn status register (CBnSTR)

(1) CSIBn receive data register (CBnRX)

The CBnRX register is a 16-bit buffer register that holds receive data.

This register is read-only, in 16-bit units.

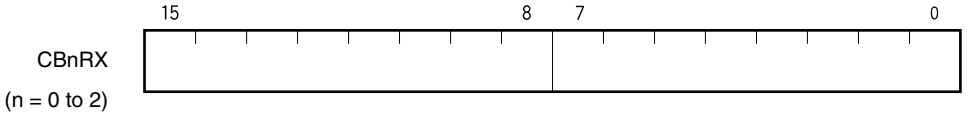
The receive operation is started by reading the CBnRX register in the reception enabled status.

If the transfer data length is 8 bits, the lower 8 bits of this register are read-only in 8-bit units as the CBnRXL register.

Reset sets this register to 0000H.

In addition to reset, the CBnRX register can be initialized by clearing (to 0) the CBnCTL0.CBnPWR bit.

After reset: 0000H R Address: CB0RX FFFFFD04H, CB0RXL FFFFFD04H,
CB1RX FFFFFD14H, CB1RXL FFFFFD14H,
CB2RX FFFFFD24H, CB2RXL FFFFFD24H



CBnRX
(n = 0 to 2)

(2) CSIBn transmit data register (CBnTX)

The CBnTX register is a 16-bit buffer register used to write the CSIBn transfer data.

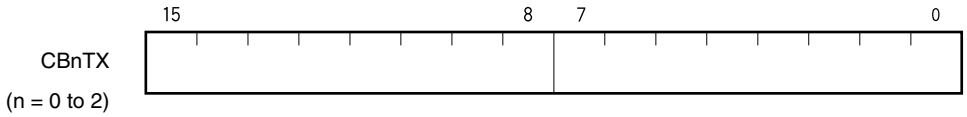
This register can be read or written in 16-bit units.

The transmit operation is started by writing data to the CBnTX register in the transmission enabled status.

If the transfer data length is 8 bits, the lower 8 bits of this register can be read or written in 8-bit units as the CBnTXL register.

Reset sets this register to 0000H.

After reset: 0000H R/W Address: CB0TX FFFFFD06H, CB0TXL FFFFFD06H,
CB1TX FFFFFD16H, CB1TXL FFFFFD16H,
CB2TX FFFFFD26H, CB2TXL FFFFFD26H



CBnTX
(n = 0 to 2)

Remark The communication start conditions are shown below.

Transmission mode (CBnTXE bit = 1, CBnRXE bit = 0):	Write to CBnTX register
Transmission/reception mode (CBnTXE bit = 1, CBnRXE bit = 1):	Write to CBnTX register
Reception mode (CBnTXE bit = 0, CBnRXE bit = 1):	Read from CBnRX register

16.4 Control Registers

The following registers are used to control CSIBn.

- CSIBn control register 0 (CBnCTL0)
- CSIBn control register 1 (CBnCTL1)
- CSIBn control register 2 (CBnCTL2)
- CSIBn status register (CBnSTR)

(1) CSIBn control register 0 (CBnCTL0)

CBnCTL0 is a register that controls the CSIBn serial transfer operation.

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

Reset sets this register to 01H.

After reset: 01H R/W Address: CB0CTL0 FFFFFD00H, CB1CTL0 FFFFFD10H, CB2CTL0 FFFFFD20H

CBnCTL0 (n = 0 to 2)

<7>	<6>	<5>	<4>	3	2	1	<0>
CBnPWR	CBnTXE[Note]	CBnRXE[Note]	CBnDIR[Note]	0	0	CBnTMS[Note]	CBnSCE

CBnPWR	Specification of CSIBn operation disable/enable
0	Disable CSIBn operation and reset the CBnSTR register
1	Enable CSIBn operation
• The CBnPWR bit controls the CSIBn operation and resets the internal circuit.	

CBnTXE[Note]	Specification of transmit operation disable/enable
0	Disable transmit operation
1	Enable transmit operation
• The SOBn output is low level when the CBnTXE bit is 0.	

CBnRXE[Note]	Specification of receive operation disable/enable
0	Disable receive operation
1	Enable receive operation
• When the CBnRXE bit is cleared to 0, no reception end interrupt is output even when the prescribed data is transferred in order to disable the receive operation, and the receive data (CBnRX register) is not updated.	

Note These bits can only be rewritten when the CBnPWR bit = 0. However, CBnPWR bit = 1 can also be set at the same time as rewriting these bits.

Caution Be sure to set bits 3 and 2 to "0".

CBnDIR[Note 1]	Specification of transfer direction mode (MSB/LSB)
0	MSB first
1	LSB first

CBnTMS[Note 1]	Transfer mode specification
0	Single transfer mode
1	Continuous transfer mode

- When using single transmission or transmission/reception mode with communication type 2 or 4 (CBnCTL1.CBnDAP bit = 1), write the transfer data to the CBnTX register after checking that the CBnSTR.CBnTSF bit is 0.
- When using DMA, use the continuous transfer mode.

CBnSCE	Specification of start transfer disable/enable
0	Communication start trigger invalid
1	Communication start trigger valid

- In master mode
  This bit enables or disables the communication start trigger.
  (a) In single reception mode
  Set the CBnSCE bit to 0 before reading the receive data (CBnRX register)[Note 2].
  (b) In continuous reception mode
  Set the CBnSCE bit to 0 one communication clock before reception of the last data is ended[Note 3].
- In slave mode
  This bit enables or disables the communication start trigger.
  (a) In single reception mode or continuous reception mode
  Set the CBnSCE bit to 1[Note 4].
- In single transmission or transmission/reception mode, or continuous transmission or transmission/reception mode
  The function of the CBnSCE bit is invalid. It is recommended to set this bit to 1.

Notes 1. These bits can only be rewritten when the CBnPWR bit = 0. However, the CBnPWR bit can be set to 1 at the same time as these bits are rewritten.

2. If the CBnSCE bit is read while it is 1, the next communication operation is started.

3. The CBnSCE bit is not set to 0 one communication clock before the end of the last data reception, the next communication operation is automatically started.
To start communication operation again after reading the last data, set the CBnSCE bit to 1 and perform a dummy read of the CBnRX register.

4. To start the reception, a dummy read is necessary.

(a) How to use CBnSCE bit

(i) In single reception mode

<1> When the reception of the last data is ended with INTCBnR interrupt servicing, clear the CBnSCE bit to 0, and then read the CBnRX register.

<2> When the reception is disabled after the reception of the last data has been ended, check that the CBnSTR.CBnTSF bit is 0, and then clear the CBnPWR and CBnRXE bits to 0. To continue reception, set the CBnSCE bit to 1 and start the next receive operation by performing a dummy read of the CBnRX register.

(ii) In continuous reception mode

<1> Clear the CBnSCE bit to 0 during reception of the last data with INTCBnR interrupt servicing by the reception before the last reception, and then read the CBnRX register.

<2> After receiving the INTCBnR signal of the last reception, read the last data from the CBnRX register.

<3> When the reception is disabled after the reception of the last data has been ended, check that the CBnSTR.CBnTSF bit is 0, and then clear the CBnPWR and CBnRXE bits to 0. To continue reception, set the CBnSCE bit to 1 and start the next receive operation by performing a dummy read of the CBnRX register.

Caution In continuous reception mode, the serial clock is not stopped until the reception executed when the CBnSCE bit is cleared to 0 is ended after the reception is started by a dummy read.

(2) CSIBn control register 1 (CBnCTL1)

CBnCTL1 is an 8-bit register that controls the CSIBn serial transfer operation.

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

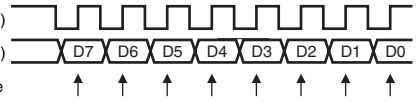
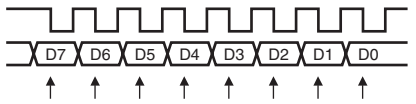
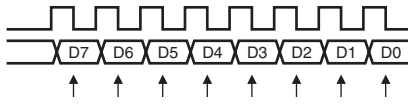
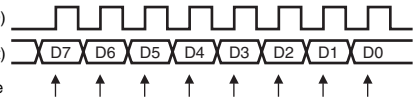
Reset sets this register to 00H.

Caution The CBnCTL1 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0.

After reset: 00H R/W Address: CB0CTL1 FFFFFD01H, CB1CTL1 FFFFFD11H, CB2CTL1 FFFFFD21H

CBnCTL1 (n = 0 to 2)

7	6	5	4	3	2	1	0
0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0

	CBnCKP	CBnDAP	Specification of data transmission/ reception timing in relation to $\overline{SCKBn}$
Communication type 1	0	0	
Communication type 2	0	1	
Communication type 3	1	0	
Communication type 4	1	1	

CBnCKS2	CBnCKS1	CBnCKS0	Communication clock (f_{CCLK})	Mode
0	0	0	$f_{XX}/4$	Master mode
0	0	1	$f_{XX}/8$	Master mode
0	1	0	$f_{XX}/16$	Master mode
0	1	1	$f_{XX}/32$	Master mode
1	0	0	$f_{XX}/64$	Master mode
1	0	1	$f_{XX}/128$	Master mode
1	1	0	$f_{XX}/256$	Master mode
1	1	1	External clock ($\overline{SCKBn}$)	Slave mode

Caution Set f_{CCLK} to 8 MHz or lower.

(3) CSIBn control register 2 (CBnCTL2)

CBnCTL2 is an 8-bit register that controls the number of CSIBn serial transfer bits.

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

Reset sets register to 00H.

Caution The CBnCTL2 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0 or when both the CBnTXE and CBnRXE bits = 0.

After reset: 00H R/W Address: CB0CTL2 FFFFFD02H, CB1CTL2 FFFFFD12H, CB2CTL2 FFFFFD22H

CBnCTL2 (n = 0 to 2)

7	6	5	4	3	2	1	0
0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0

CBnCL3	CBnCL2	CBnCL1	CBnCL0	Serial register bit length
0	0	0	0	8 bits
0	0	0	1	9 bits
0	0	1	0	10 bits
0	0	1	1	11 bits
0	1	0	0	12 bits
0	1	0	1	13 bits
0	1	1	0	14 bits
0	1	1	1	15 bits
1	×	×	×	16 bits

Remark If the number of transfer bits is other than 8 or 16, prepare and use data stuffed from the LSB of the CBnTX and CBnRX registers.

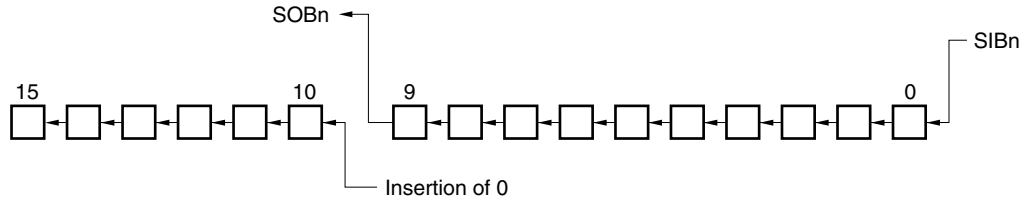
(a) Transfer data length change function

The CSIBn transfer data length can be set in 1-bit units between 8 and 16 bits using the CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits.

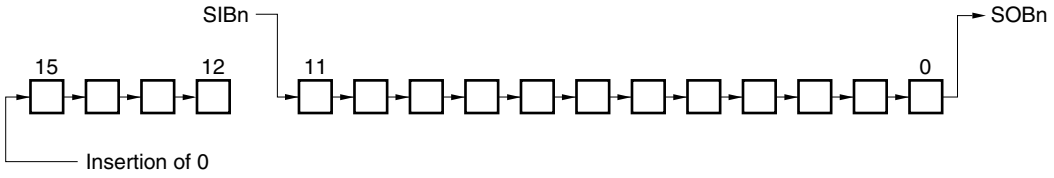
When the transfer bit length is set to a value other than 16 bits, set the data to the CBnTX or CBnRX register starting from the LSB, regardless of whether the transfer start bit is the MSB or LSB. Any data can be set for the higher bits that are not used, but the receive data becomes 0 following serial transfer.

Remark n = 0 to 2

(i) Transfer bit length = 10 bits, MSB first



(ii) Transfer bit length = 12 bits, LSB first



(4) CSIBn status register (CBnSTR)

CBnSTR is an 8-bit register that displays the CSIBn status.

This register can be read or written in 8-bit or 1-bit units, but the CBnTSF flag is read-only.

Reset sets this register to 00H.

In addition to reset, the CBnSTR register can be initialized by clearing (0) the CBnCTL0.CBnPWR bit.

After reset: 00H R/W Address: CB0STR FFFFFD03H, CB1STR FFFFFD13H, CB2STR FFFFFD23H

	<7>	6	5	4	3	2	1	<0>
CBnSTR	CBnTSF	0	0	0	0	0	0	CBnOVE

(n = 0 to 2)

CBnTSF	Communication status flag
0	Communication stopped
1	Communicating

- During transmission, this register is set when data is prepared in the CBnTX register, and during reception, it is set when a dummy read of the CBnRX register is performed.
  When transfer ends, this flag is cleared to 0 at the last edge of the clock.

CBnOVE	Overrun error flag
0	No overrun
1	Overrun

- An overrun error occurs when the next reception starts without performing a CPU read of the value of the CBnRX register, upon end of the receive operation.
  The CBnOVE flag displays the overrun error occurrence status in this case.
- The CBnOVE flag is cleared by writing 0 to it. It cannot be set even by writing 1 to it.

Caution In single transfer mode, writing to the CBnTX register with the CBnTSF bit set to 1 is ignored. This has no influence on the operation during transfer.

For example, if the next data is written to the CBnTX register when DMA is started by generating the INTCBnR signal, the written data is not transferred because the CBnTSF bit is set to 1.

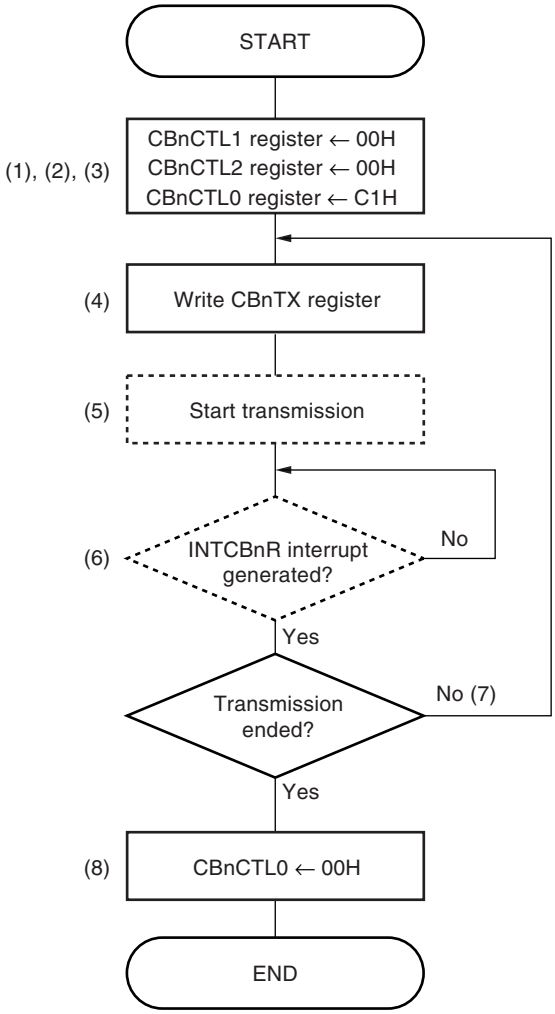
Use the continuous transfer mode, not the single transfer mode, for such applications.

16.5 Operation

16.5.1 Single transfer mode (master mode, transmission mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/4 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

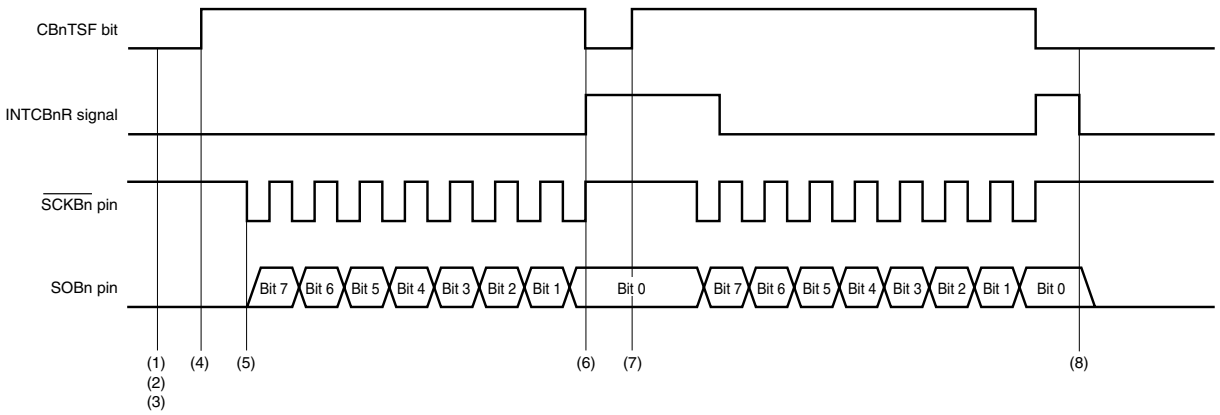


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/4, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write C1H to the CBnCTL0 register, and select the transmission mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and transmission is started.

(5) When transmission is started, output the serial clock to the $\overline{\text{SCKBn}}$ pin, and output the transmit data from the SOBn pin in synchronization with the serial clock.

(6) When transmission of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output and transmit data output, generate the reception end interrupt request signal (INTCBnR) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

(7) To continue transmission, start the next transmission by writing the transmit data to the CBnTX register again after the INTCBnR signal is generated.

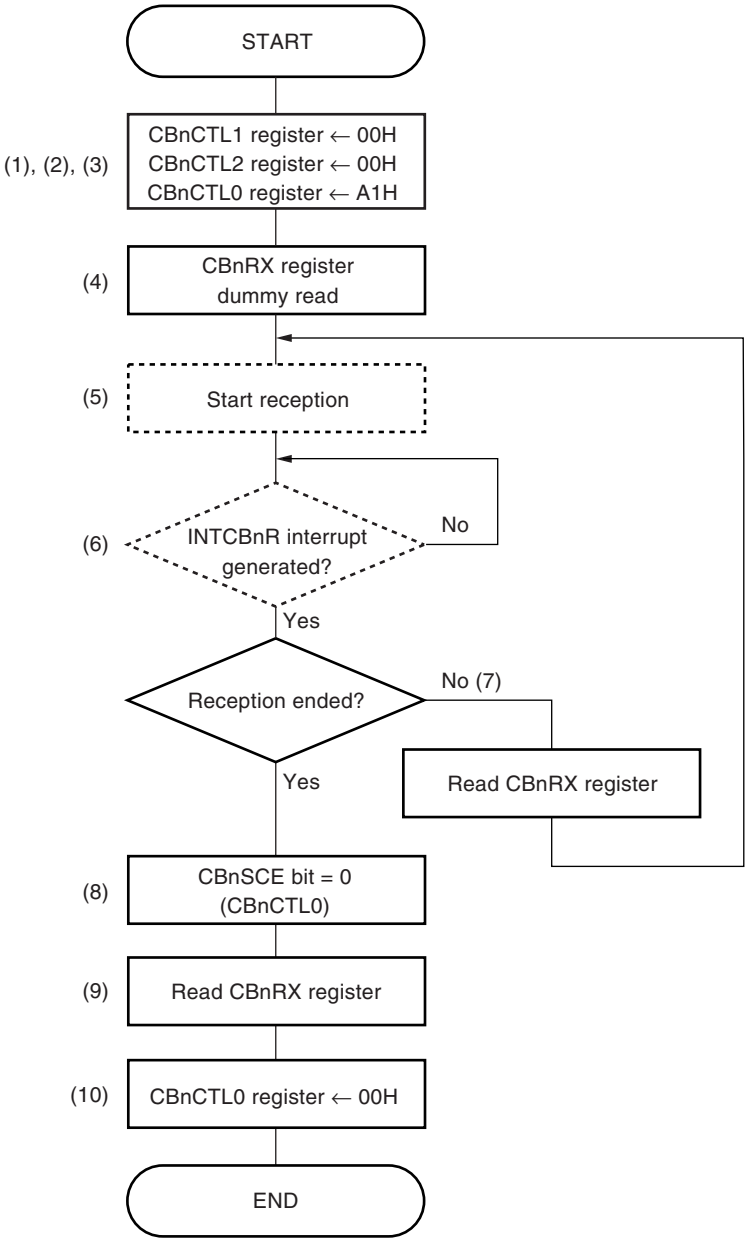
(8) To end transmission, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnTXE bit = 0.

Remark n = 0 to 2

16.5.2 Single transfer mode (master mode, reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/4 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

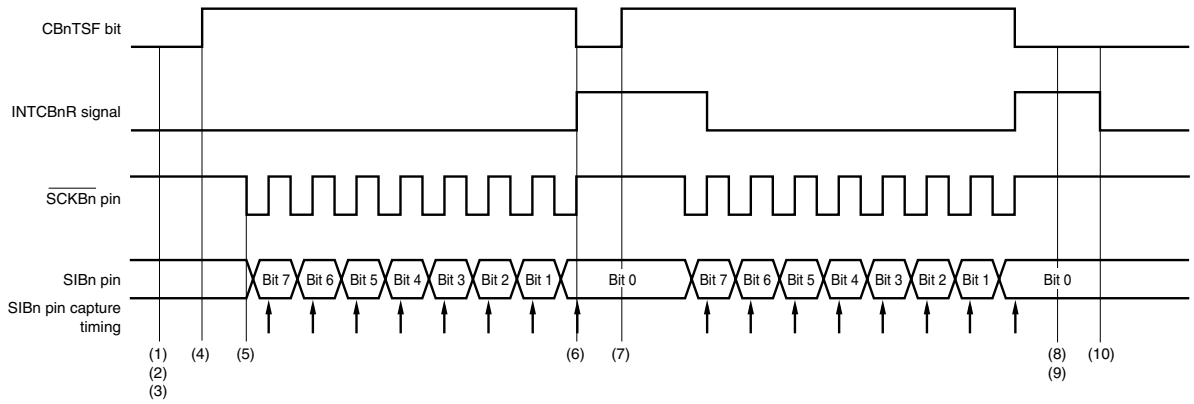


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/4, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write A1H to the CBnCTL0 register, and select the reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by performing a dummy read of the CBnRX register, and reception is started.

(5) When reception is started, output the serial clock to the $\overline{\text{SCKBn}}$ pin, and capture the receive data of the SIBn pin in synchronization with the serial clock.

(6) When reception of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output and data capturing, generate the reception end interrupt request signal (INTCBnR) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

(7) To continue reception, read the CBnRX register with the CBnCTL0.CBnSCE bit = 1 remained after the INTCBnR signal is generated.

(8) To read the CBnRX register without starting the next reception, write the CBnSCE bit = 0.

(9) Read the CBnRX register.

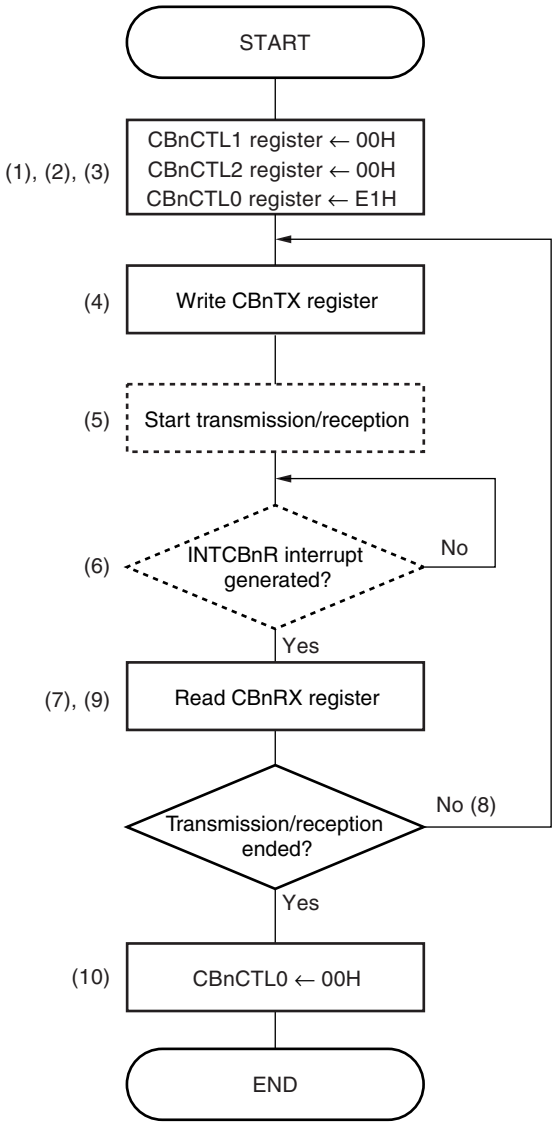
(10) To end reception, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnRXE bit = 0.

Remark n = 0 to 2

16.5.3 Single transfer mode (master mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/4 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

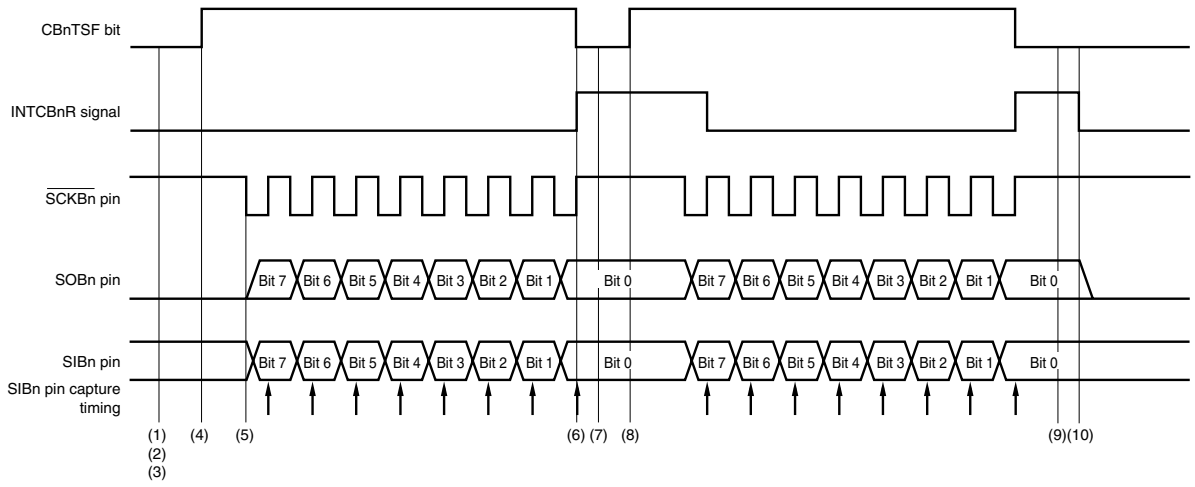


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/4, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write E1H to the CBnCTL0 register, and select the transmission/reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and transmission/reception is started.

(5) When transmission/reception is started, output the serial clock to the $\overline{SCKBn}$ pin, output the transmit data to the SOBn pin in synchronization with the serial clock, and capture the receive data of the SIBn pin.

(6) When transmission/reception of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output, transmit data output, and data capturing, generate the reception end interrupt request signal (INTCBnR) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

(7) Read the CBnRX register.

(8) To continue transmission/reception, write the transmit data to the CBnTX register again.

(9) Read the CBnRX register.

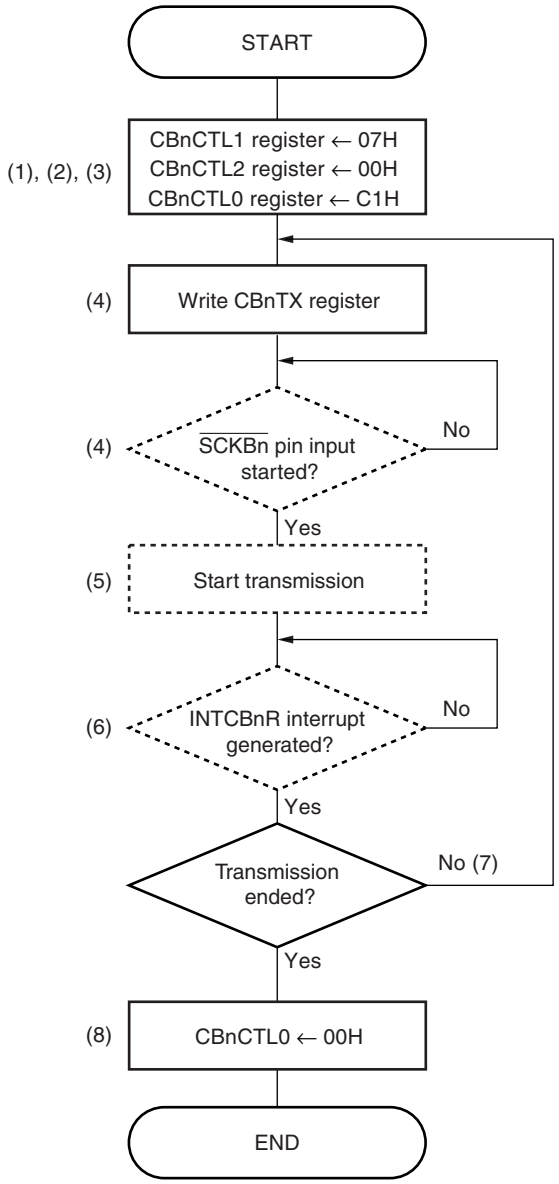
(10) To end transmission/reception, write the CBnCTL0.CBnPWR bit = 0, the CBnCTL0.CBnTXE bit = 0, and the CBnCTL0.CBnRXE bit = 0.

Remark n = 0 to 2

16.5.4 Single transfer mode (slave mode, transmission mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

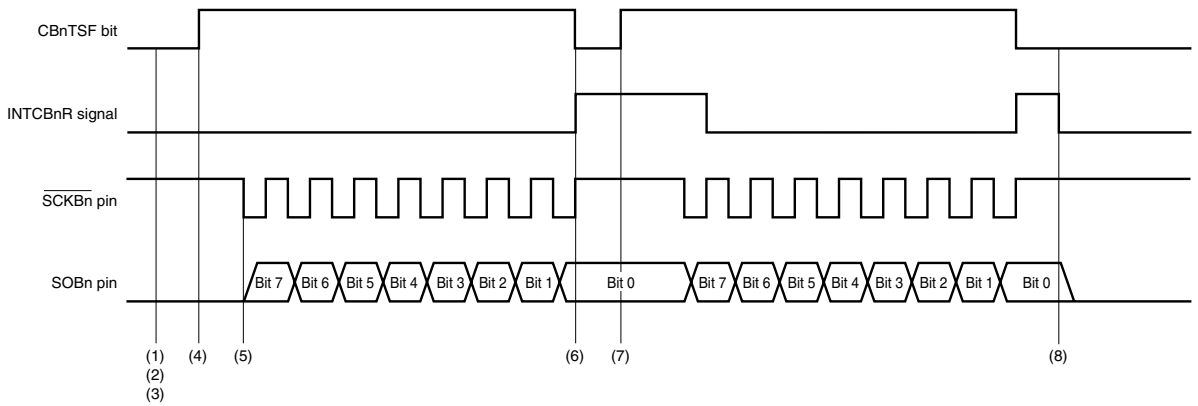


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$), and slave mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write C1H to the CBnCTL0 register, and select the transmission mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data from the SOBn pin in synchronization with the serial clock.

(6) When transmission of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output and transmit data output, generate the reception end interrupt request signal (INTCBnR) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

(7) To continue transmission, write the transmit data to the CBnTX register again after the INTCBnR signal is generated, and wait for a serial clock input.

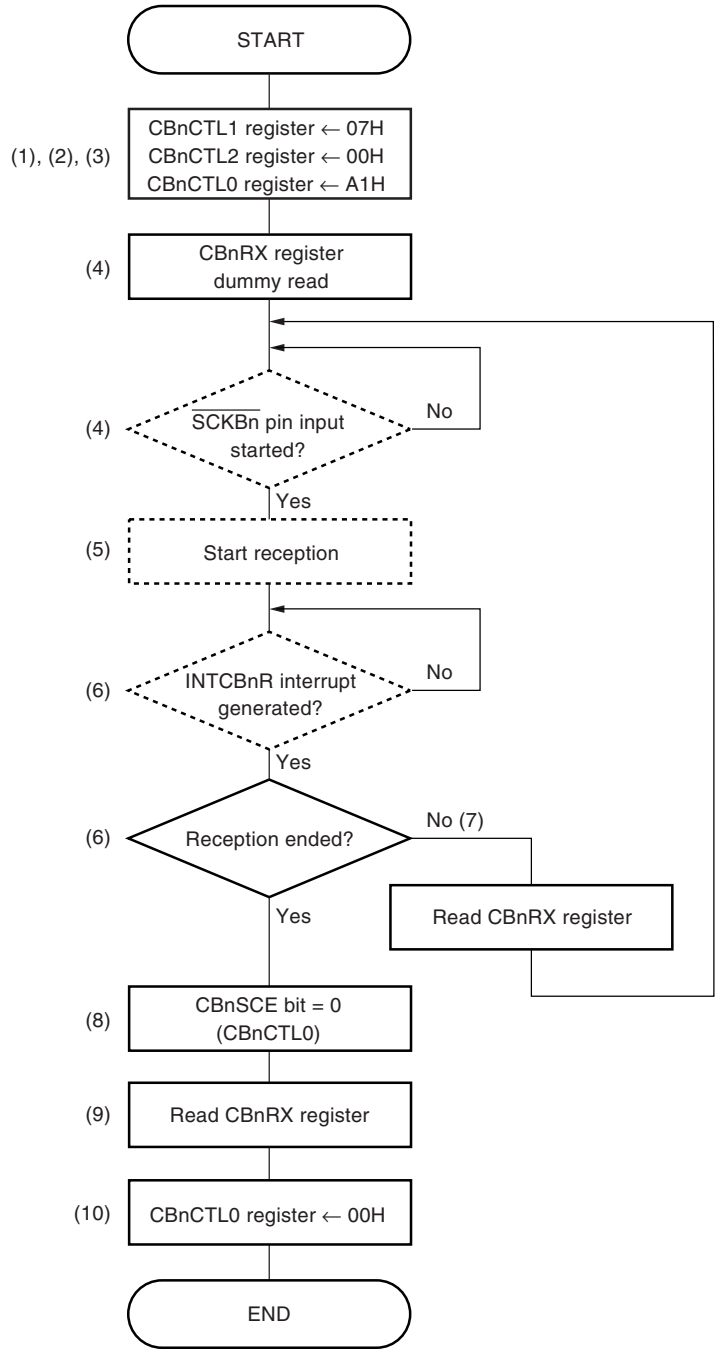
(8) To end transmission, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnTXE bit = 0.

Remark n = 0 to 2

16.5.5 Single transfer mode (slave mode, reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

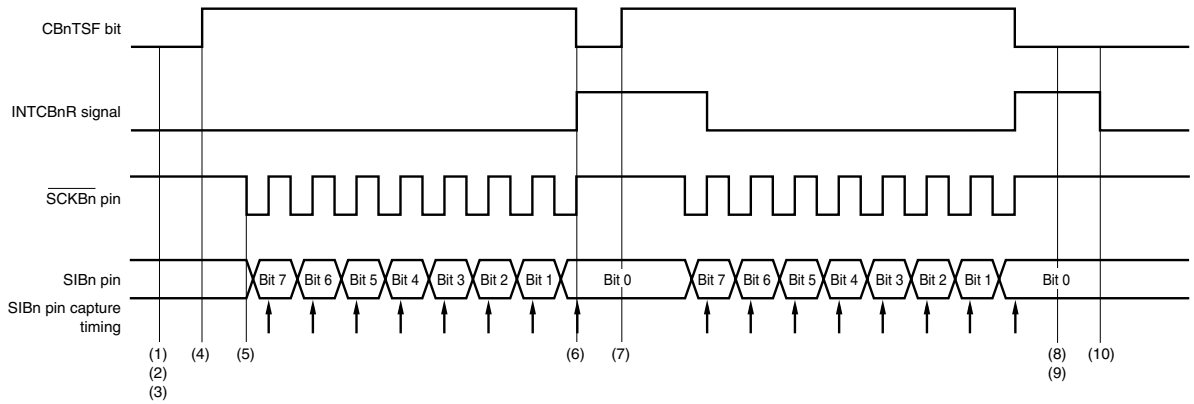


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$), and slave mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write A1H to the CBnCTL0 register, and select the reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by performing a dummy read of the CBnRX register, and the device waits for a serial clock input.

(5) When a serial clock is input, capture the receive data of the SIBn pin in synchronization with the serial clock.

(6) When reception of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output and data capturing, generate the reception end interrupt request signal (INTCBnR) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

(7) To continue reception, read the CBnRX register with the CBnCTL0.CBnSCE bit = 1 remained after the INTCBnR signal is generated, and wait for a serial clock input.

(8) To end reception, write the CBnSCE bit = 0.

(9) Read the CBnRX register.

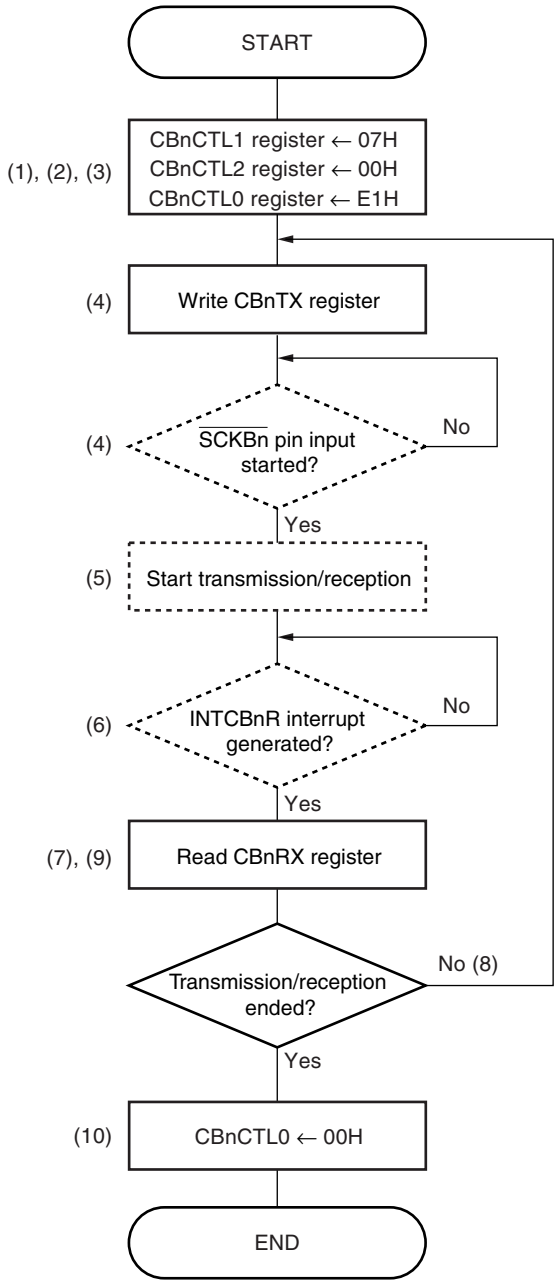
(10) To end reception, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnRXE bit = 0.

Remark n = 0 to 2

16.5.6 Single transfer mode (slave mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

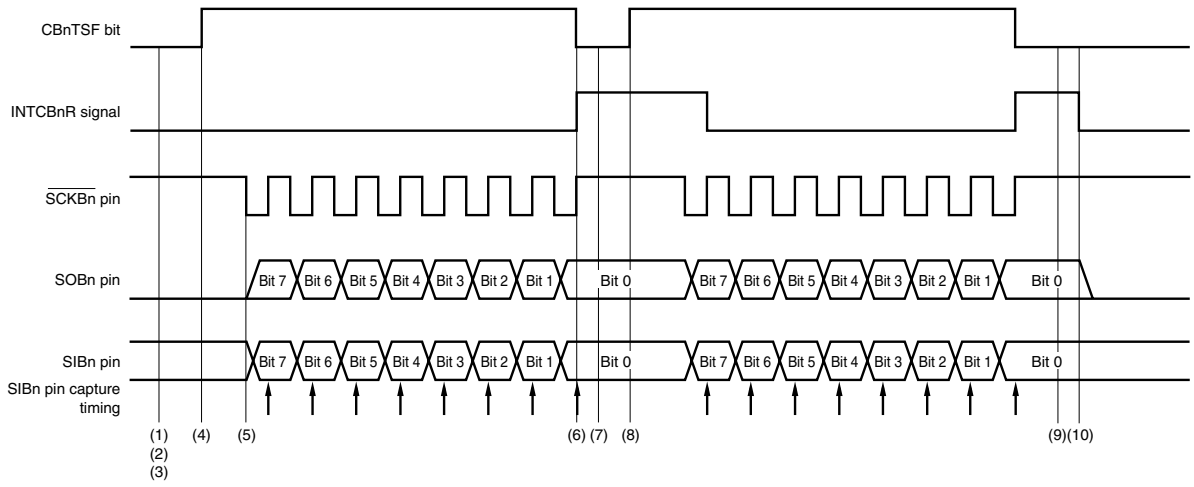


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$), and slave mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write E1H to the CBnCTL0 register, and select the transmission/reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data to the SOBn pin in synchronization with the serial clock, and capture the receive data of the SIBn pin.

(6) When transmission/reception of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output, transmit data output, and data capturing, generate the reception end interrupt request signal (INTCBnR) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

(7) Read the CBnRX register.

(8) To continue transmission/reception, write the transmit data to the CBnTX register again, and wait for a serial clock input.

(9) Read the CBnRX register.

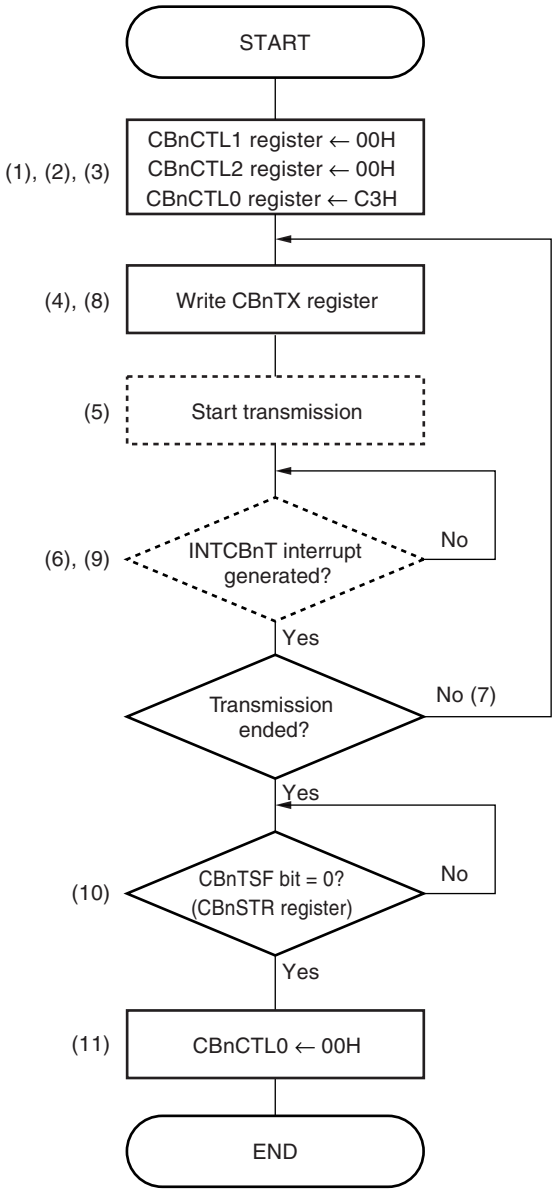
(10) To end transmission/reception, write the CBnCTL0.CBnPWR bit = 0, the CBnCTL0.CBnTXE bit = 0, and the CBnCTL0.CBnRXE bit = 0.

Remark n = 0 to 2

16.5.7 Continuous transfer mode (master mode, transmission mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/4 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

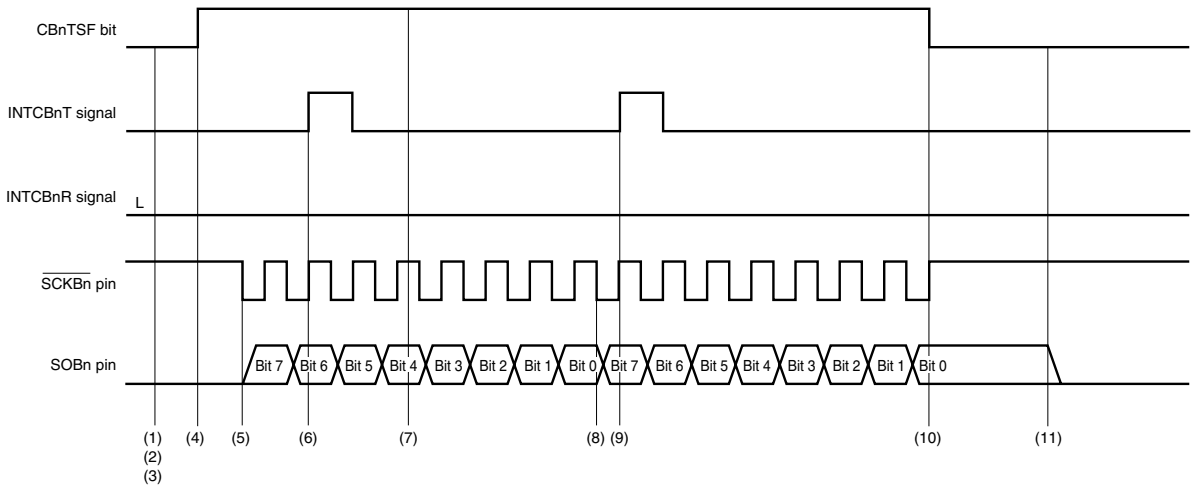


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/4, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write C3H to the CBnCTL0 register, and select the transmission mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and transmission is started.

(5) When transmission is started, output the serial clock to the $\overline{\text{SCKBn}}$ pin, and output the transmit data from the SOBn pin in synchronization with the serial clock.

(6) When transfer of the transmit data from the CBnTX register to the shift register is ended and writing to the CBnTX register is enabled, the transmission enable interrupt request signal (INTCBnT) is generated.

(7) To continue transmission, write the transmit data to the CBnTX register again after the INTCBnT signal is generated.

(8) When a new transmit data is written to the CBnTX register before communication end, the next communication is started following communication end.

(9) The transfer of the transmit data from the CBnTX register to the shift register is ended and the INTCBnT signal is generated. To end continuous transmission at the current transmission, do not write to the CBnTX register.

(10) When the next transmit data is not written to the CBnTX register before transfer end, stop the serial clock output to the $\overline{\text{SCKBn}}$ pin after transfer end, and clear the CBnTSF bit to 0.

(11) To release the transmission enable status, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnTXE bit = 0 after checking that the CBnTSF bit = 0.

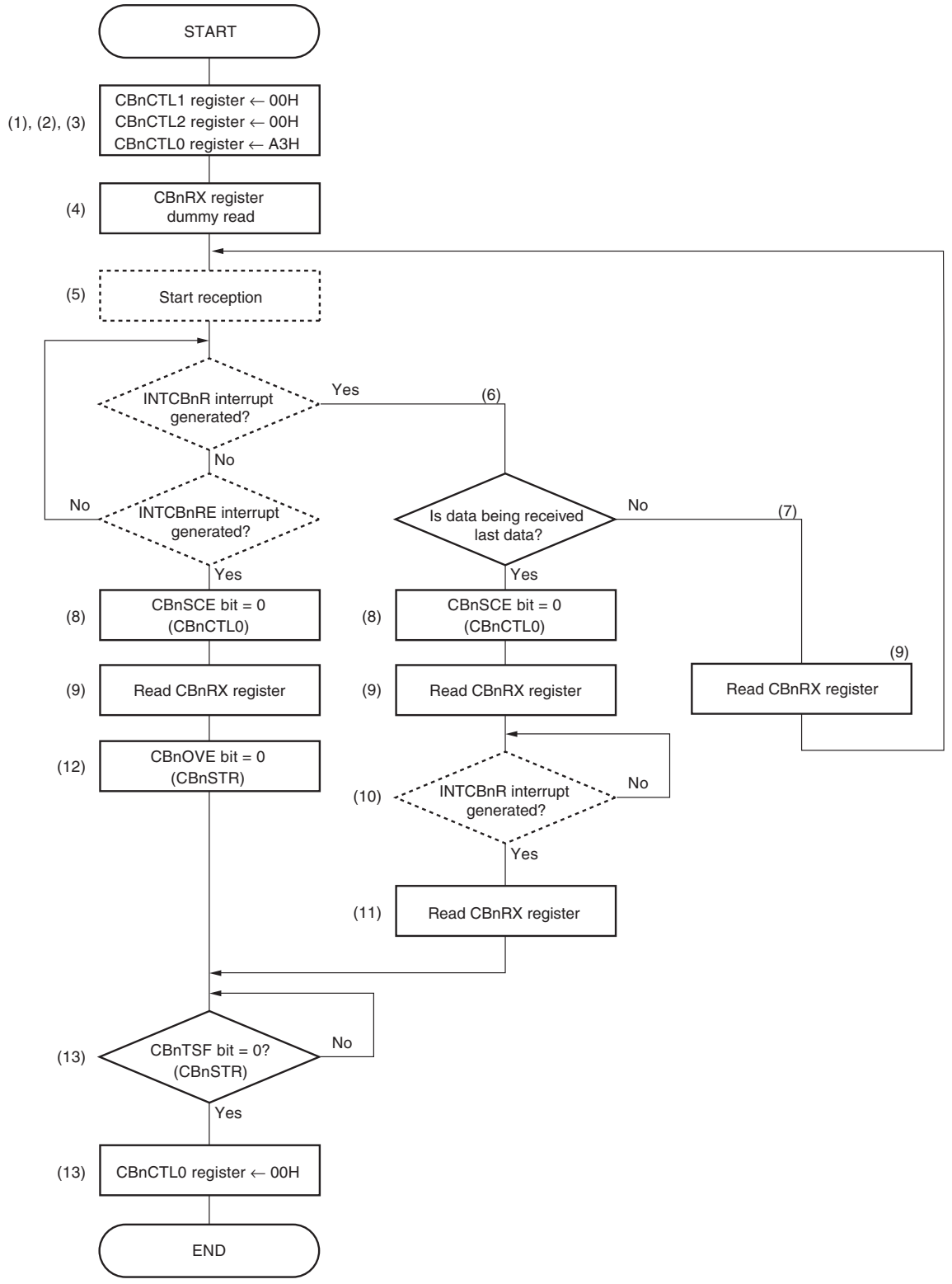
Caution In continuous transmission mode, the reception end interrupt request signal (INTCBnR) is not generated.

Remark n = 0 to 2

16.5.8 Continuous transfer mode (master mode, reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/4 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

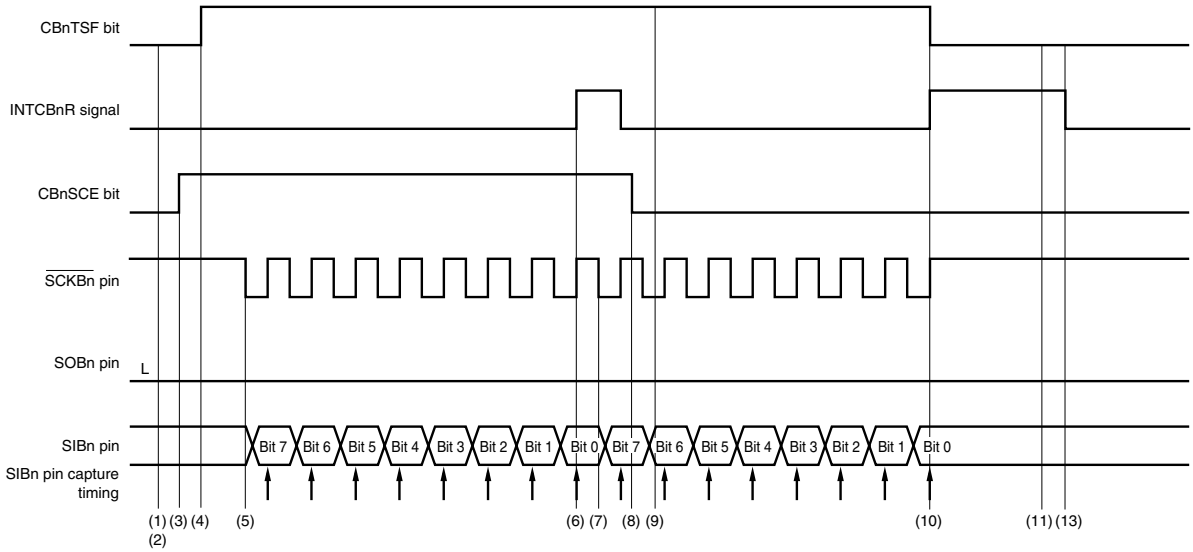


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/4, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write A3H to the CBnCTL0 register, and select the reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by performing a dummy read of the CBnRX register, and reception is started.

(5) When reception is started, output the serial clock to the $\overline{\text{SCKBn}}$ pin, and capture the receive data of the SIBn pin in synchronization with the serial clock.

(6) When reception is ended, the reception end interrupt request signal (INTCBnR) is generated, and reading of the CBnRX register is enabled.

(7) When the CBnCTL0.CBnSCE bit = 1 upon communication end, the next communication is started following communication end.

(8) To end continuous reception at the current reception, write the CBnSCE bit = 0.

(9) Read the CBnRX register.

(10) When reception is ended, the INTCBnR signal is generated, and reading of the CBnRX register is enabled. When the CBnSCE bit = 0 is set before communication end, stop the serial clock output to the $\overline{\text{SCKBn}}$ pin, and clear the CBnTSF bit to 0, to end the receive operation.

(11) Read the CBnRX register.

(12) If an overrun error occurs, write the CBnSTR.CBnOVE bit = 0, and clear the error flag.

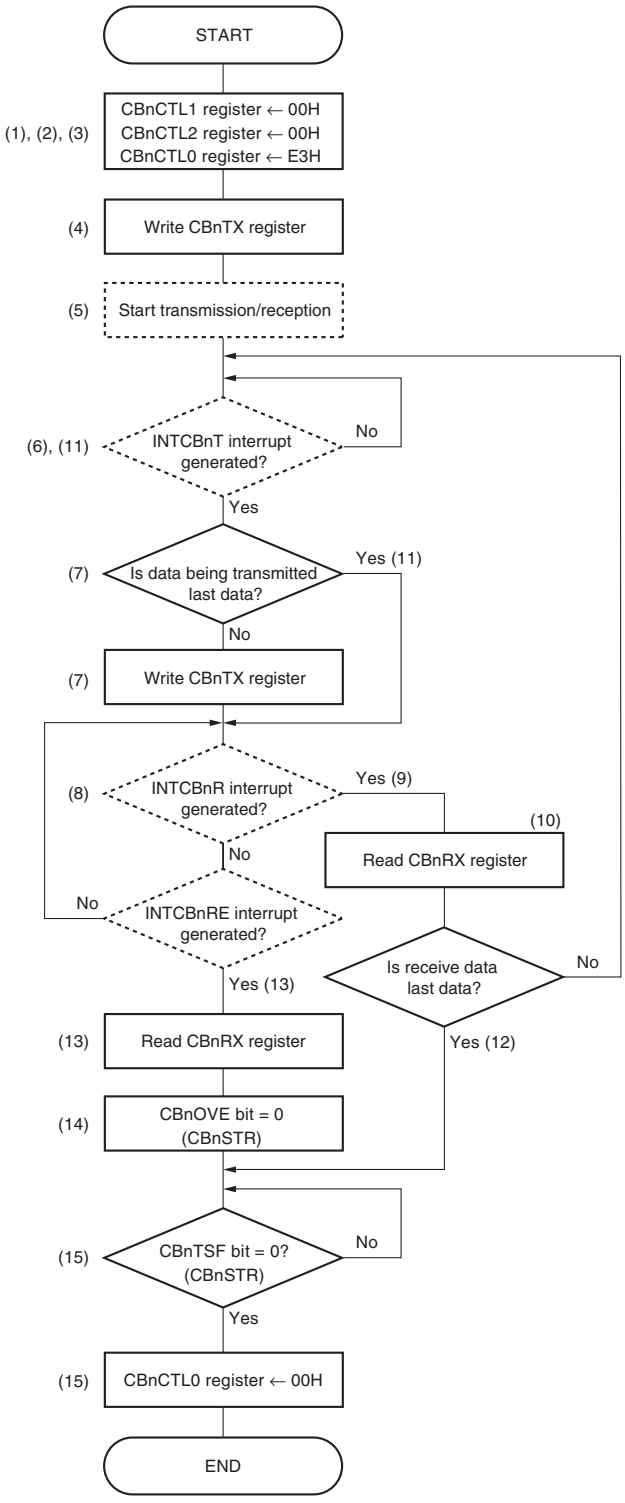
(13) To release the reception enable status, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnRXE bit = 0 after checking that the CBnTSF bit = 0.

Remark n = 0 to 2

16.5.9 Continuous transfer mode (master mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/4 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow



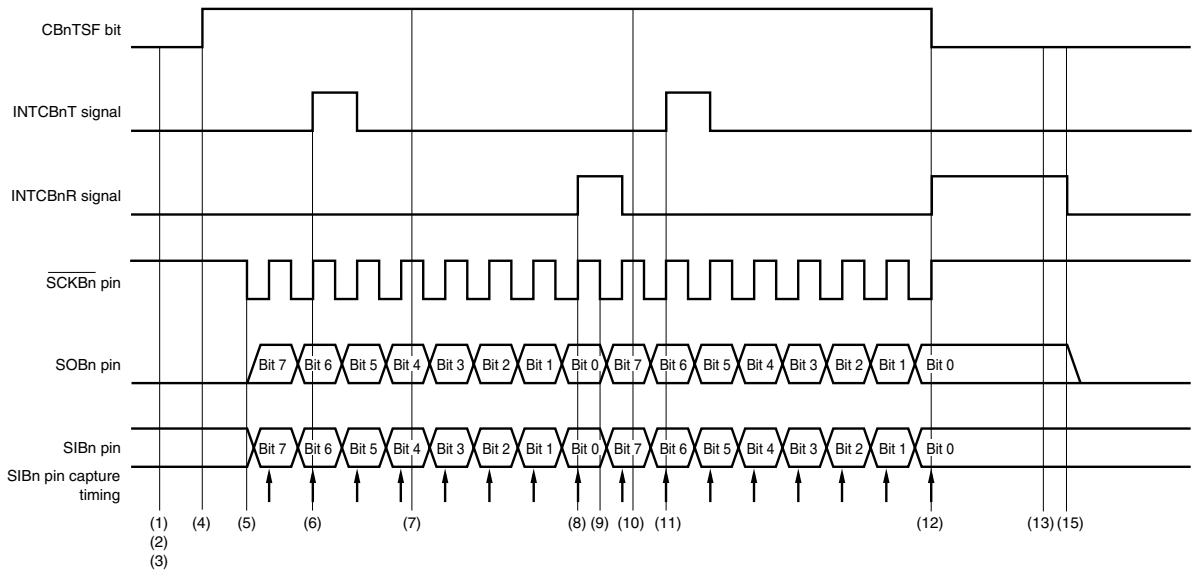
Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing

(1/2)



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/4, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write E3H to the CBnCTL0 register, and select the transmission/reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and transmission/reception is started.

(5) When transmission/reception is started, output the serial clock to the $\overline{\text{SCKBn}}$ pin, output the transmit data to the SOBn pin in synchronization with the serial clock, and capture the receive data of the SIBn pin.

(6) When transfer of the transmit data from the CBnTX register to the shift register is ended and writing to the CBnTX register is enabled, the transmission enable interrupt request signal (INTCBnT) is generated.

(7) To continue transmission/reception, write the transmit data to the CBnTX register again after the INTCBnT signal is generated.

(8) When one transmission/reception is ended, the reception end interrupt request signal (INTCBnR) is generated, and reading of the CBnRX register is enabled.

(9) When a new transmit data is written to the CBnTX register before communication end, the next communication is started following communication end.

(10) Read the CBnRX register.

Remark n = 0 to 2

(2/2)

(11) The transfer of the transmit data from the CBnTX register to the shift register is ended and the INTCBnT signal is generated. To end continuous transmission/reception at the current transmission/reception, do not write to the CBnTX register.

(12) When the next transmit data is not written to the CBnTX register before transfer end, stop the serial clock output to the $\overline{\text{SCKBn}}$ pin after transfer end, and clear the CBnTSF bit to 0.

(13) When the reception error interrupt request signal (INTCBnRE) is generated, read the CBnRX register.

(14) If an overrun error occurs, write the CBnSTR.CBnOVE bit = 0, and clear the error flag.

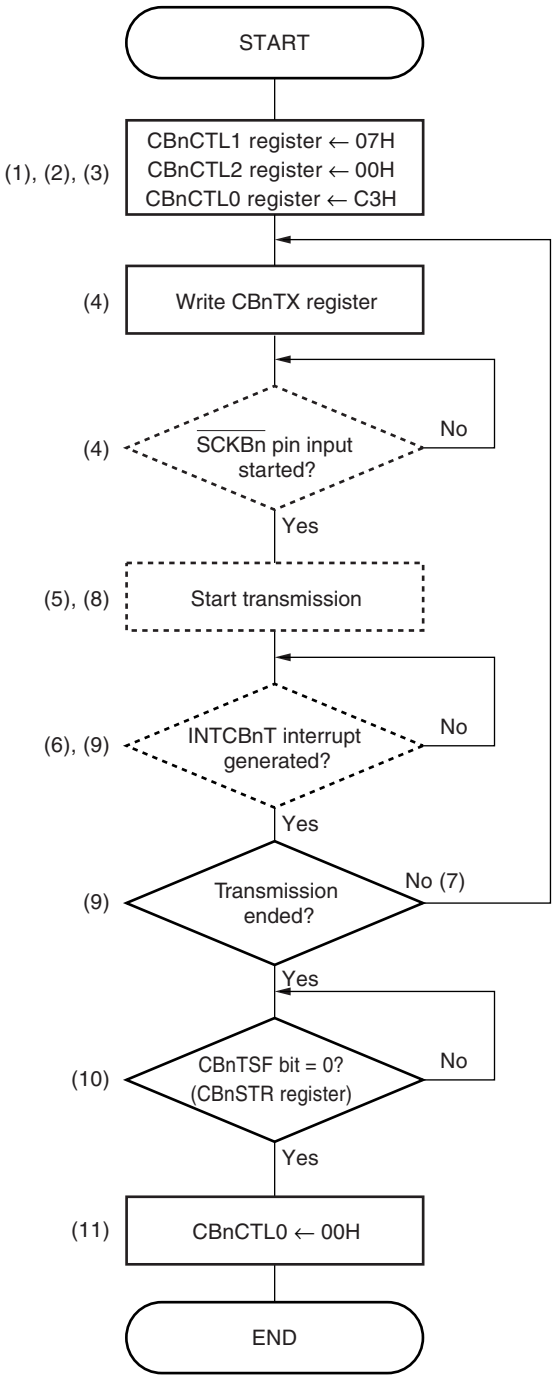
(15) To release the transmission/reception enable status, write the CBnCTL0.CBnPWR bit = 0, the CBnCTL0.CBnTXE bit = 0, and the CBnCTL0.CBnRXE bit = 0 after checking that the CBnTSF bit = 0.

Remark n = 0 to 2

16.5.10 Continuous transfer mode (slave mode, transmission mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

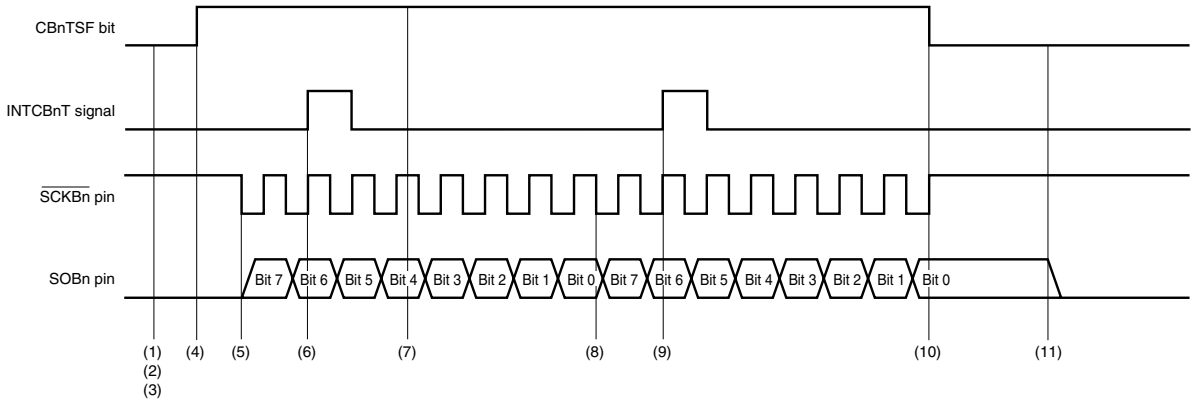


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$), and slave mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write C3H to the CBnCTL0 register, and select the transmission mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data from the SOBn pin in synchronization with the serial clock.

(6) When transfer of the transmit data from the CBnTX register to the shift register is ended and writing to the CBnTX register is enabled, the transmission enable interrupt request signal (INTCBnT) is generated.

(7) To continue transmission, write the transmit data to the CBnTX register again after the INTCBnT signal is generated.

(8) When a serial clock is input following end of the transmission of the transfer data length set with the CBnCTL2 register, continuous transmission is started.

(9) When transfer of the transmit data from the CBnTX register to the shift register is ended and writing to the CBnTX register is enabled, the INTCBnT signal is generated. To end continuous transmission at the current transmission, do not write to the CBnTX register.

(10) When the clock of the transfer data length set with the CBnCTL2 register is input without writing to the CBnTX register, clear the CBnTSF bit to 0 to end transmission.

(11) To release the transmission enable status, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnTXE bit = 0 after checking that the CBnTSF bit = 0.

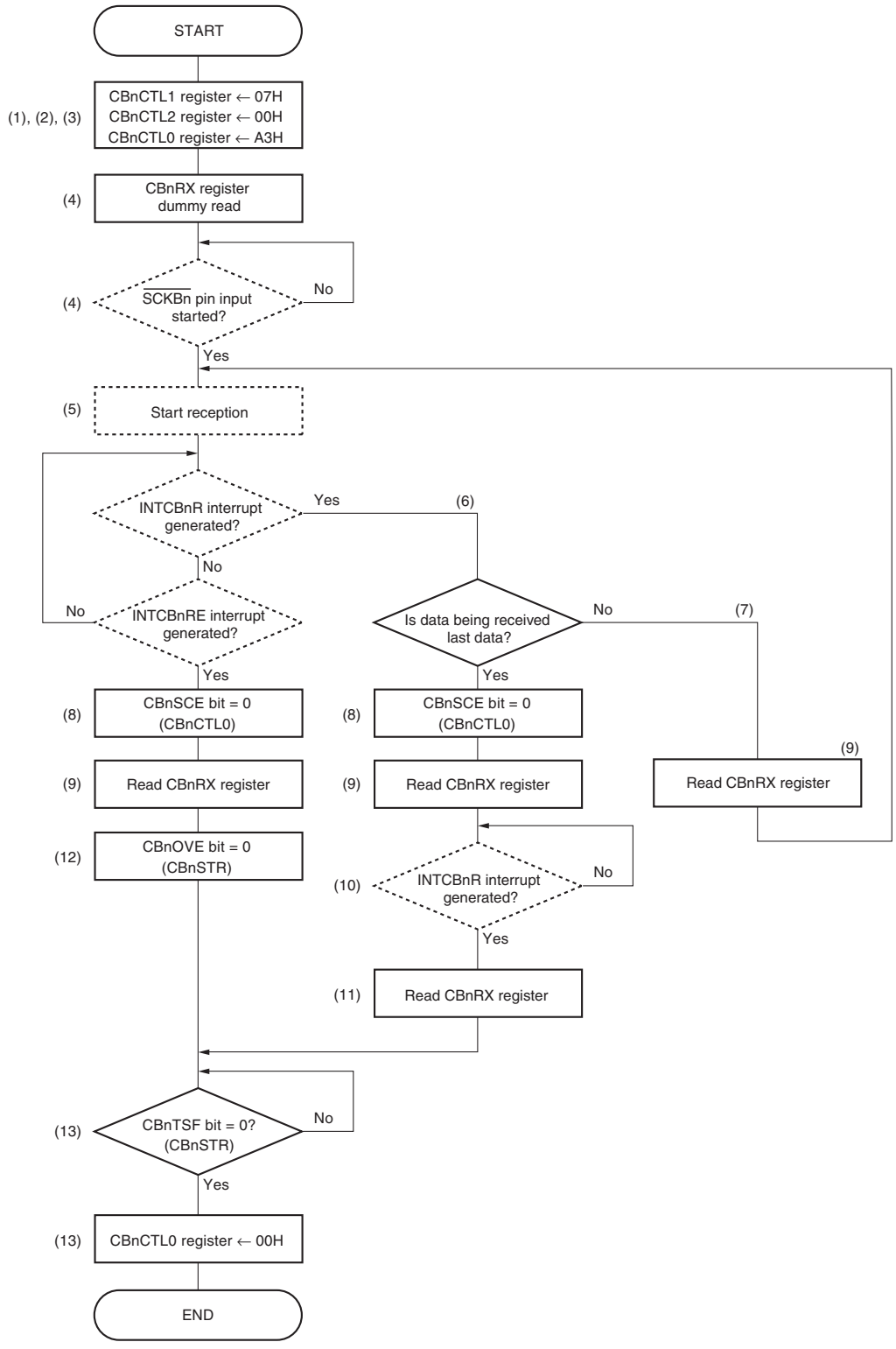
Caution In continuous transmission mode, the reception end interrupt request signal (INTCBnR) is not generated.

Remark n = 0 to 2

16.5.11 Continuous transfer mode (slave mode, reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

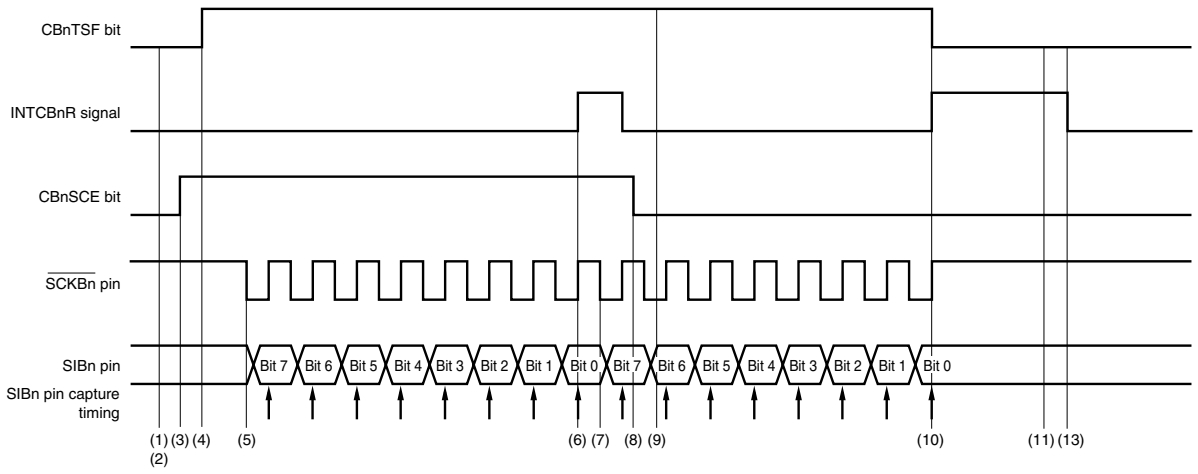


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing



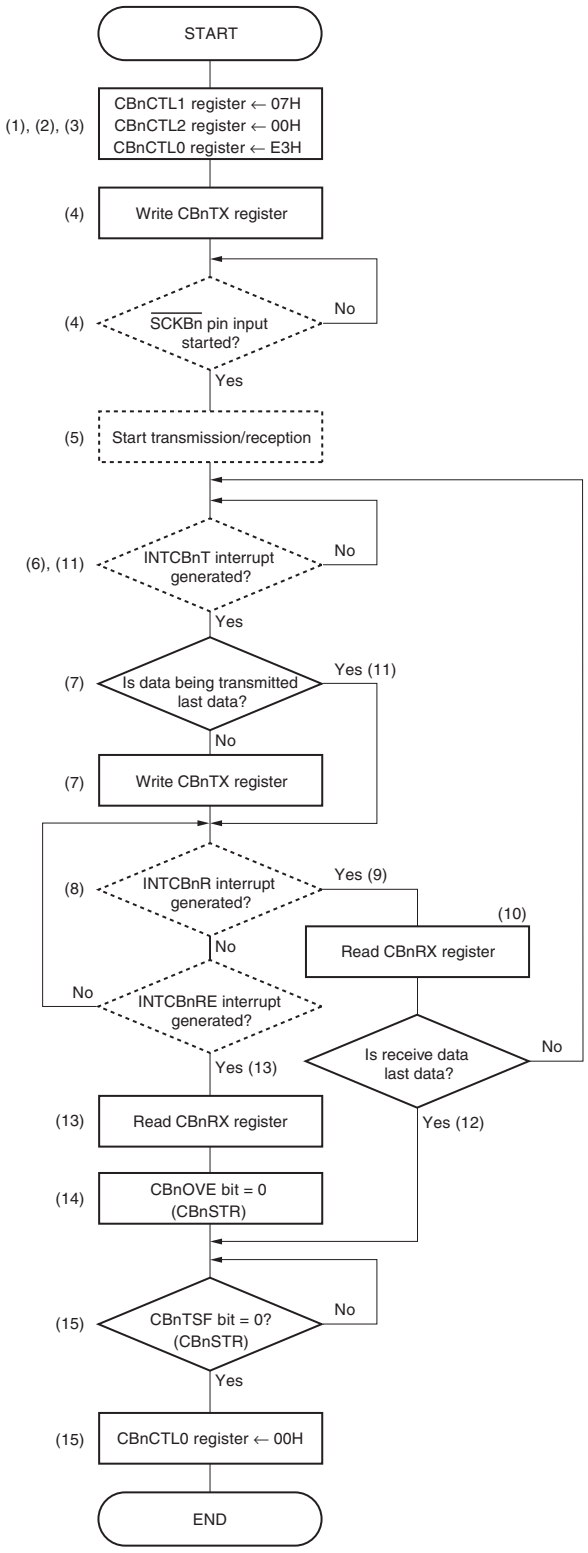
(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$), and slave mode.
(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.
(3) Write A3H to the CBnCTL0 register, and select the reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).
(4) The CBnSTR.CBnTSF bit is set to 1 by performing a dummy read of the CBnRX register, and the device waits for a serial clock input.
(5) When a serial clock is input, capture the receive data of the SIBn pin in synchronization with the serial clock.
(6) When reception is ended, the reception end interrupt request signal (INTCBnR) is generated, and reading of the CBnRX register is enabled.
(7) When a serial clock is input in the CBnCTL0.CBnSCE bit = 1 status, continuous reception is started.
(8) To end continuous reception at the current reception, write the CBnSCE bit = 0.
(9) Read the CBnRX register.
(10) When reception is ended, the INTCBnR signal is generated, and reading of the CBnRX register is enabled. When the CBnSCE bit = 0 is set before communication end, clear the CBnTSF bit to 0 to end the receive operation.
(11) Read the CBnRX register.
(12) If an overrun error occurs, write the CBnSTR.CBnOVE bit = 0, and clear the error flag.
(13) To release the reception enable status, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnRXE bit = 0 after checking that the CBnTSF bit = 0.

Remark n = 0 to 2

16.5.12 Continuous transfer mode (slave mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow



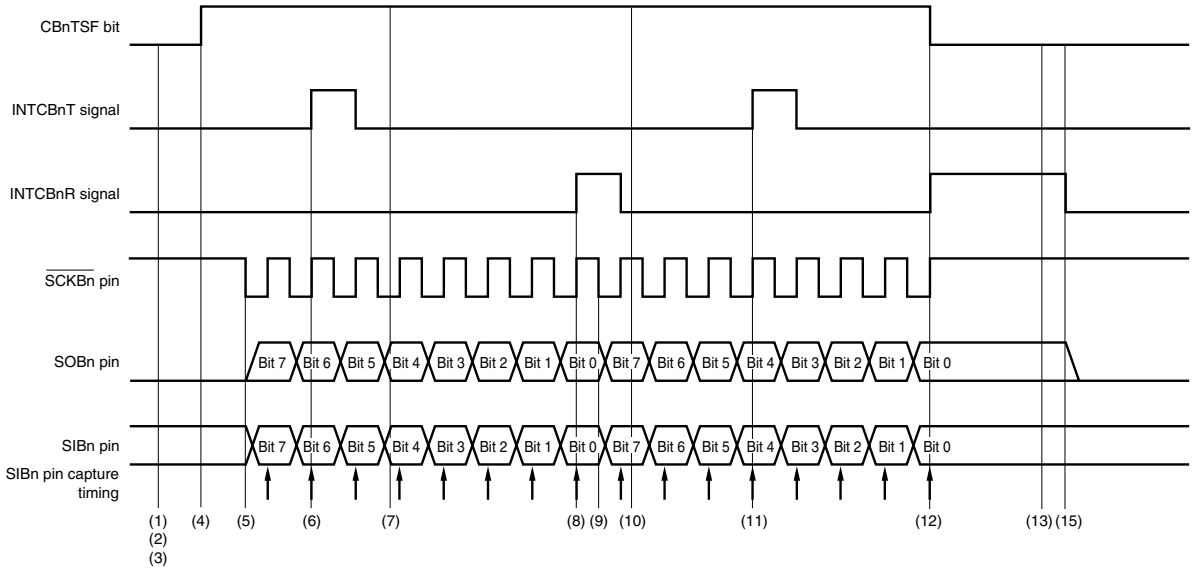
Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0 to 2

(2) Operation timing

(1/2)



(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$), and slave mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write E3H to the CBnCTL0 register, and select the transmission/reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data to the SOBn pin in synchronization with the serial clock, and capture the receive data of the SIBn pin.

(6) When transfer of the transmit data from the CBnTX register to the shift register is ended and writing to the CBnTX register is enabled, the transmission enable interrupt request signal (INTCBnT) is generated.

(7) To continue transmission, write the transmit data to the CBnTX register again after the INTCBnT signal is generated.

(8) When reception of the transfer data length set with the CBnCTL2 register is completed, the reception end interrupt request signal (INTCBnR) is generated, and reading of the CBnRX register is enabled.

(9) When a serial clock is input continuously, continuous transmission/reception is started.

(10) Read the CBnRX register.

(11) When transfer of the transmit data from the CBnTX register to the shift register is ended and writing to the CBnTX register is enabled, the INTCBnT signal is generated. To end continuous transmission/reception at the current transmission/reception, do not write to the CBnTX register.

Remark n = 0 to 2

(2/2)

(12) When the clock of the transfer data length set with the CBnCTL2 register is input without writing to the CBnTX register, the INTCBnR signal is generated. Clear the CBnTSF bit to 0 to end transmission/reception.
(13) When the reception error interrupt request signal (INTCBnRE) is generated, read the CBnRX register.
(14) If an overrun error occurs, write the CBnSTR.CBnOVE bit = 0, and clear the error flag.
(15) To release the transmission/reception enable status, write the CBnCTL0.CBnPWR bit = 0, the CBnCTL0.CBnTXE bit = 0, and the CBnCTL0.CBnRXE bit = 0 after checking that the CBnTSF bit = 0.

Remark n = 0 to 2

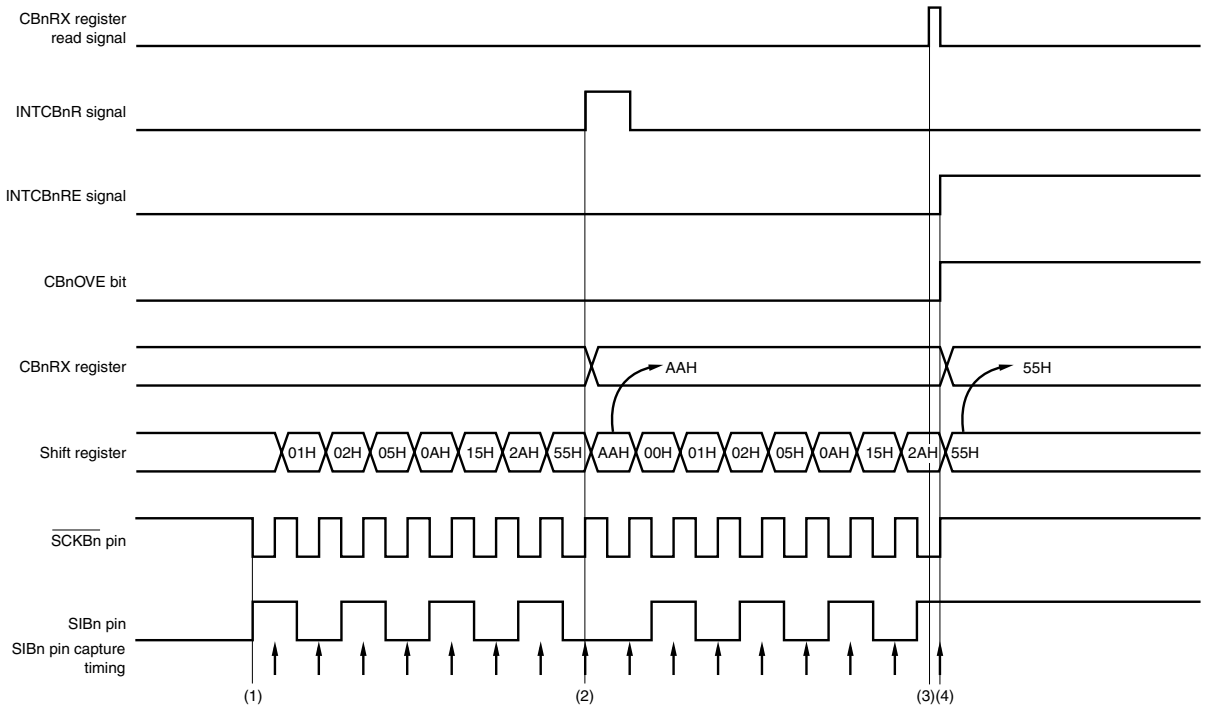
16.5.13 Reception error

When transfer is performed with reception enabled (CBnCTL0.CBnRXE bit = 1) in the continuous transfer mode, the reception error interrupt request signal (INTCBnRE) is generated when the next receive operation is ended before the CBnRX register is read after the reception end interrupt request signal (INTCBnR) is generated, and the overrun error flag (CBnSTR.CBnOVE) is set to 1.

Even if an overrun error has occurred, the previous receive data is lost since the CBnRX register is updated. Even if a reception error has occurred, the INTCBnRE signal is generated again upon the next reception end if the CBnRX register is not read.

To avoid an overrun error, end reading the CBnRX register until one half clock before sampling the last bit of the next receive data from the INTCBnR signal generation.

(1) Operation timing

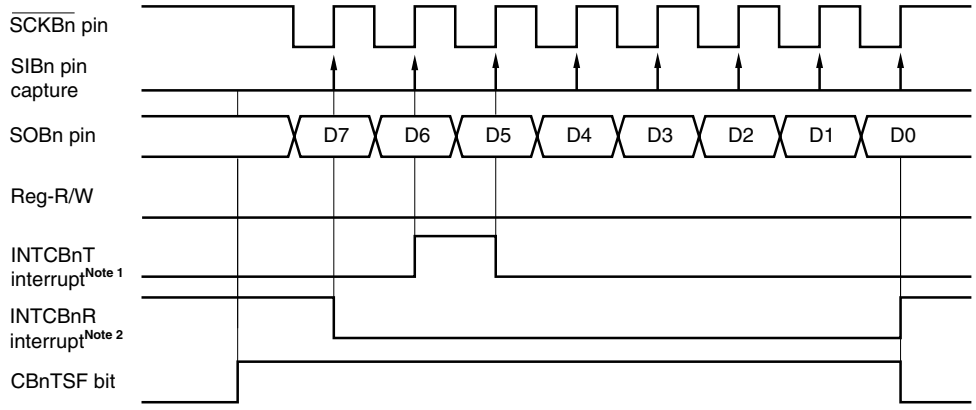


(1) Start continuous transfer.
(2) End of the first transfer
(3) The CBnRX register cannot be read until one half-clock before the end of the second transfer.
(4) An overrun error occurs, and the reception error interrupt request signal (INTCBnRE) is generated. The receive data is overwritten.

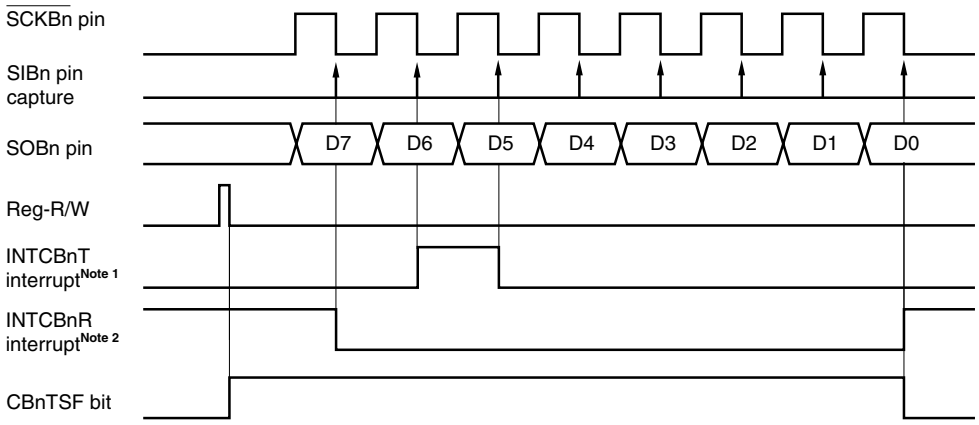
Remark n = 0 to 2

16.5.14 Clock timing

(i) Communication type 1 (CBnCKP and CBnDAP bits = 00)



(ii) Communication type 3 (CBnCKP and CBnDAP bits = 10)



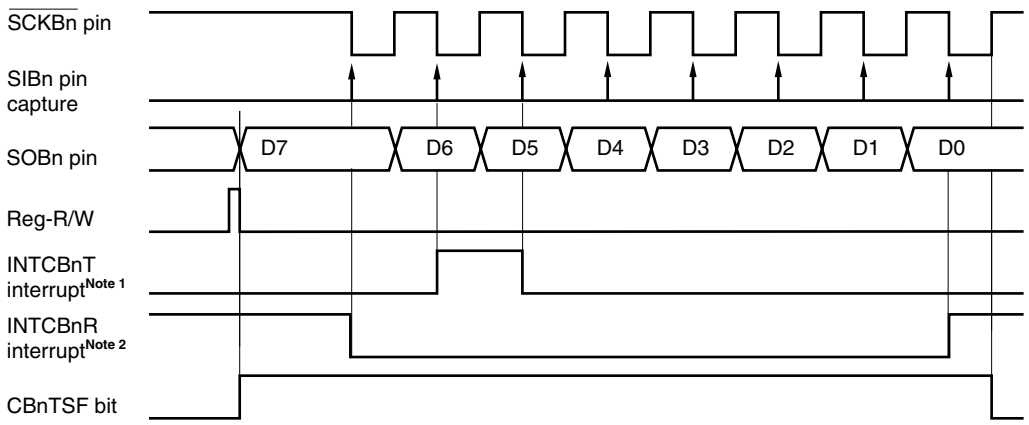
Notes 1. The INTCBnT interrupt is set when the data written to the CBnTX register is transferred to the data shift register in the continuous transmission or continuous transmission/reception mode. In the single transmission or single transmission/reception mode, the INTCBnT interrupt request signal is not generated, but the INTCBnR interrupt request signal is generated upon end of communication.

2. The INTCBnR interrupt occurs if reception is correctly ended and receive data is ready in the CBnRX register while reception is enabled. In the single mode, the INTCBnR interrupt request signal is generated even in the transmission mode, upon end of communication.

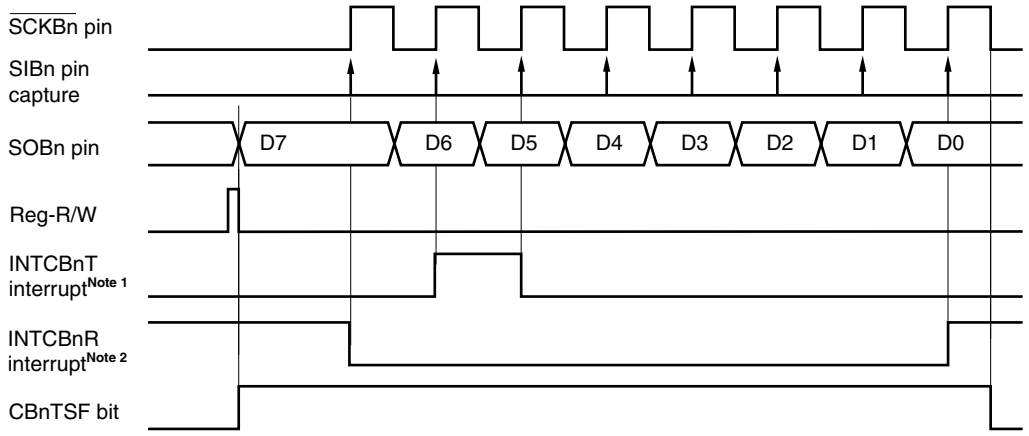
Caution In single transfer mode, writing to the CBnTX register with the CBnTSF bit set to 1 is ignored. This has no influence on the operation during transfer.
For example, if the next data is written to the CBnTX register when DMA is started by generating the INTCBnR signal, the written data is not transferred because the CBnTSF bit is set to 1.
Use the continuous transfer mode, not the single transfer mode, for such applications.

Remark n = 0 to 2

(iii) Communication type 2 (CBnCKP and CBnDAP bits = 01)



(iv) Communication type 4 (CBnCKP and CBnDAP bits = 11)



Notes 1. The INTCBnT interrupt is set when the data written to the CBnTX register is transferred to the data shift register in the continuous transmission or continuous transmission/reception mode. In the single transmission or single transmission/reception mode, the INTCBnT interrupt request signal is not generated, but the INTCBnR interrupt request signal is generated upon end of communication.

2. The INTCBnR interrupt occurs if reception is correctly ended and receive data is ready in the CBnRX register while reception is enabled. In the single mode, the INTCBnR interrupt request signal is generated even in the transmission mode, upon end of communication.

Caution In single transfer mode, writing to the CBnTX register with the CBnTSF bit set to 1 is ignored. This has no influence on the operation during transfer.

For example, if the next data is written to the CBnTX register when DMA is started by generating the INTCBnR signal, the written data is not transferred because the CBnTSF bit is set to 1.

Use the continuous transfer mode, not the single transfer mode, for such applications.

Remark n = 0 to 2

16.6 Output Pins

(1) $\overline{\text{SCKBn}}$ pin

When CSIBn operation is disabled (CBnCTL0.CBnPWR bit = 0), the $\overline{\text{SCKBn}}$ pin output status is as follows.

Remark n = 0 to 2

CBnCKP	CBnCKS2	CBnCKS1	CBnCKS0	$\overline{\text{SCKBn}}$ Pin Output
0	1	1	1	High impedance
	Other than above			Fixed to high level
1	1	1	1	High impedance
	Other than above			Fixed to low level

Remark The output level of the $\overline{\text{SCKBn}}$ pin changes if any of the CBnCTL1.CBnCKP and CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits is rewritten.

(2) SOBn pin

When CSIBn operation is disabled (CBnPWR bit = 0), the SOBn pin output status is as follows.

Remark n = 0 to 2

CBnTXE	CBnDAP	CBnDIR	SOBn Pin Output
0	×	×	Fixed to low level
1	0	×	SOBn latch value (low level)
	1	0	CBnTX value (MSB)
		1	CBnTX value (LSB)

Remarks 1. The SOBn pin output changes when any one of the CBnCTL0.CBnTXE, CBnCTL0.CBnDIR, or CBnCTL1.CBnDAP bits is rewritten.

2. ×: Don't care

CHAPTER 17 I²C BUS

To use the I²C bus function, use the P30/SCL and P31/SDA pins as the serial transmit/receive data and set them to N-ch open-drain output.

In the V850E/IF3 and V850E/IG3, one channel of I²C bus is provided.

17.1 Mode Switching Between I²C and UARTA1

In the V850E/IF3 and V850E/IG3, I²C and UARTA1 function alternately, and these pins cannot be used at the same time. To switch between I²C and UARTA1, the PMC3, PFC3, and PFCE3 registers must be set in advance.

Caution The operations related to transmission and reception of I²C or UARTA1 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 17-1. Mode Switch Settings of I²C and UARTA1

After reset: 00H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	PFC37	PFC36	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	PFCE37	PFCE36	PFCE35	PFCE34	0	PFCE32	PFCE31	PFCE30

PMC31	PFCE31	PFC31	Specification of alternate function of P31 pin
0	×	×	I/O port
1	0	0	TXDA1 output
1	0	1	SDA I/O
1	1	0	Setting prohibited
1	1	1	Setting prohibited

PMC30	PFCE30	PFC30	Specification of alternate function of P30 pin
0	×	×	I/O port
1	0	0	RXDA1 input
1	0	1	SCL I/O
1	1	0	Setting prohibited
1	1	1	Setting prohibited

Remark × = don't care

17.2 Features

The I²C has the following two modes.

- Operation stop mode
- I²C (Inter IC) bus mode (multimaster supported)

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

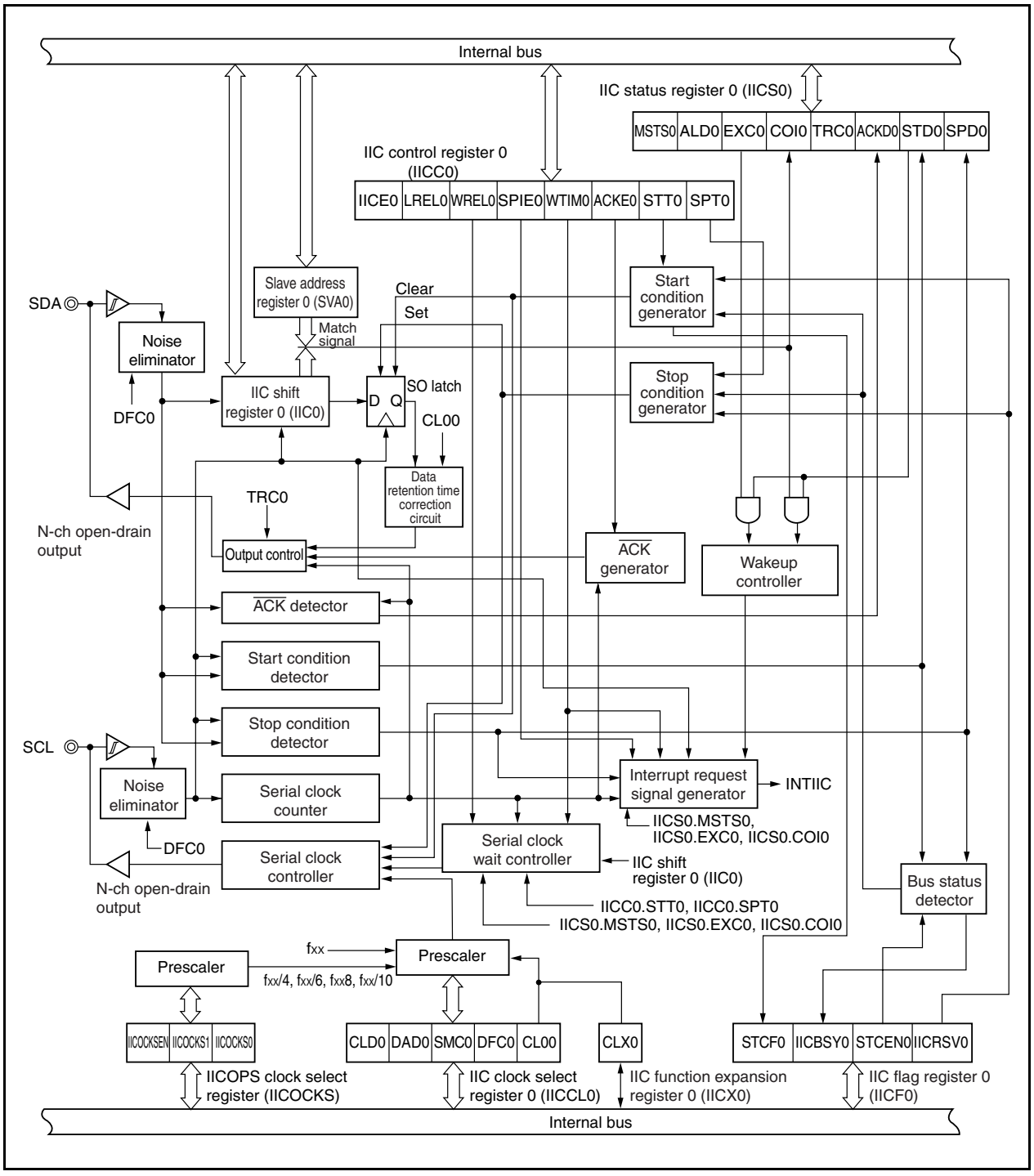
(2) I²C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCL) line and a serial data bus (SDA) line.

This mode complies with the I²C bus format and the master device can generate “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received state and data by hardware. This function can simplify the part of application program that controls the I²C bus.

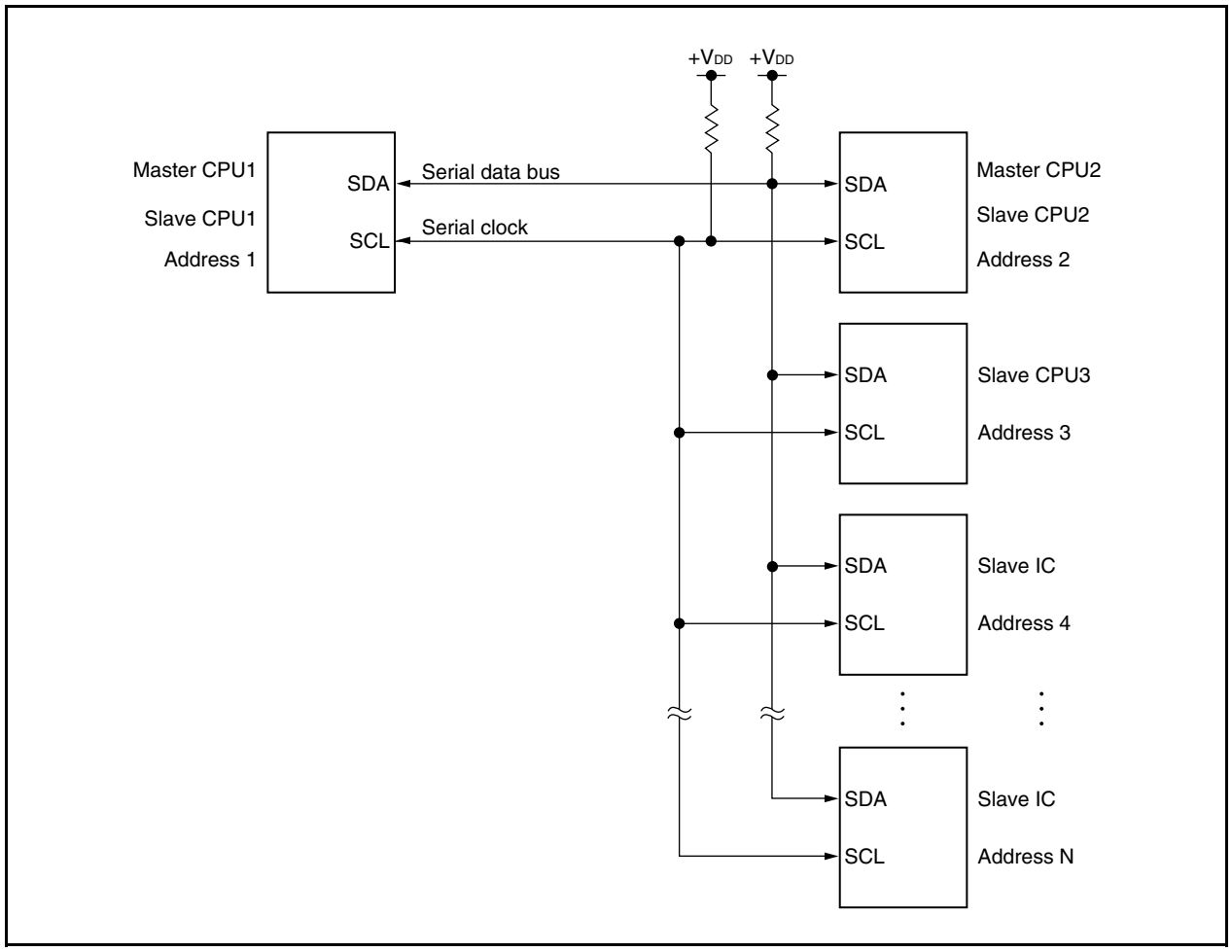
Since the SCL and SDA pins are used for N-ch open drain outputs, I²C requires pull-up resistors for the serial clock line and the serial data bus line.

Figure 17-2. Block Diagram of I²C



A serial bus configuration example is shown below.

Figure 17-3. Serial Bus Configuration Example Using I²C Bus



17.3 Configuration

I²C includes the following hardware.

Table 17-1. Configuration of I²C

Item	Configuration
Registers	IIC shift register 0 (IIC0) Slave address register 0 (SVA0)
Control registers	IIC control register 0 (IICC0) IIC status register 0 (IICS0) IIC flag register 0 (IICF0) IIC clock select register 0 (IICCL0) IIC function expansion register 0 (IICX0) IICOPS clock select register (IICOCKS)

(1) IIC shift register 0 (IIC0)

The IIC0 register is used to convert 8-bit serial data to 8-bit parallel data and to convert 8-bit parallel data to 8-bit serial data. The IIC0 register can be used for both transmission and reception.

Write and read operations to the IIC0 register are used to control the actual transmit and receive operations.

The IIC0 register can be read or written in 8-bit units.

Reset sets IIC0 to 00H.

(2) Slave address register 0 (SVA0)

The SVA0 register sets local addresses when in slave mode.

The SVA0 register can be read or written in 8-bit units.

Reset sets SVA0 to 00H.

(3) SO latch

The SO latch is used to retain the SDA pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request signal (INTIIC) when the address received by this register matches the address value set to the SVA0 register or when an extension code is received.

(5) Prescaler

This selects the sampling clock to be used.

(6) Serial clock counter

This counter counts the serial clocks that are output and the serial clocks that are input during transmit/receive operations and is used to verify that 8-bit data was sent or received.

(7) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIIC).

An I²C interrupt is generated following either of two triggers.

- Falling of the eighth or ninth clock of the serial clock (set by IICC0.WTIM0 bit)
- Interrupt request generated when a stop condition is detected (set by IICC0.SPIE0 bit)

(8) Serial clock controller

In master mode, this circuit generates the clock output via the SCL pin from a sampling clock.

(9) Serial clock wait controller

This circuit controls the wait timing.

(10) $\overline{\text{ACK}}$ generator, stop condition detector, start condition detector, and $\overline{\text{ACK}}$ detector

These circuits are used to generate and detect various statuses.

(11) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

(12) Start condition generator

This circuit generates a start condition when the IICC0.STT0 bit is set.

However, in the communication reservation disabled status (IICF0.IICRSV0 bit = 1), when the bus is not released (IICF0.IICBSY0 bit = 1), start condition requests are ignored and the IICF0.STCF0 bit is set to 1.

(13) Stop condition generator

A stop condition is generated when the IIC0.SPT0 bit is set (1).

(14) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the IICF0.STCEN0 bit.

17.4 Registers

I²C is controlled by the following registers.

- IIC control register 0 (IICC0)
- IIC status register 0 (IICS0)
- IIC flag register 0 (IICF0)
- IIC clock select register 0 (IICCL0)
- IIC function expansion register 0 (IICX0)
- IICOPS clock select register (IICOCKS)

The following registers are also used.

- IIC shift register 0 (IIC0)
- Slave address register 0 (SVA0)

Remark For the alternate-function pin settings, see **Table 4-14 Settings When Port Pins Are Used for Alternate Functions**.

(1) IIC control register 0 (IICC0)

The IICC0 register is used to enable/stop I²C operations, set wait timing, and set other I²C operations.

The IICC0 register can be read or written in 8-bit or 1-bit units. However, set the SPIE0, WTIM0, and ACKE0 bits when the IICE0 bit is 0 or during the wait period. When setting the IICE0 bit from "0" to "1", these bits can also be set at the same time.

Reset sets this register to 00H.

(1/4)

After reset: 00H R/W Address: FFFFFD82H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICC0	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I²C operation enable/disable specification
0	Stop operation. Reset the IICS0 register[Note 1]. Stop internal operation.
1	Enable operation.

Be sure to set this bit to 1 when the SCL and SDA lines are high level.

Condition for clearing (IICE0 bit = 0)	Condition for setting (IICE0 bit = 1)
• Cleared by instruction • Reset	• Set by instruction

LREL0[Note 2]	Exit from communications
0	Normal operation
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCL and SDA lines are set to high impedance. The STT0, SPT0, IICS0.MSTS0, IICS0.EXC0, IICS0.COI0, IICS0.TRC0, IICS0.ACKD0, and IICS0.STD0 bits are cleared to 0.

The standby mode following exit from communications remains in effect until the following communications entry conditions are met.

- After a stop condition is detected, restart is in master mode.
- An address match or extension code reception occurs after the start condition.

Condition for clearing (LREL0 bit = 0)	Condition for setting (LREL0 bit = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Notes 1. The IICS0 register, and the IICF0.STCF0, IICF0.IICBSY0, IICCL0.CLD0, and IICCL0.DAD0 bits are reset.

2. This flag's signal is invalid when the IICE0 bit = 0.

Caution If the I²C operation is enabled (IICE0 bit = 1) when the SCL line is high level and the SDA line is low level, the start condition is detected immediately. To avoid this, after enabling the I²C operation, immediately set the LREL0 bit to 1 with a bit manipulation instruction.

(2/4)

WREL0[Note]	Wait cancellation control
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared to 0 after wait is canceled.

Condition for clearing (WREL0 bit = 0)	Condition for setting (WREL0 bit = 1)
• Automatically cleared after execution • Reset	• Set by instruction

SPIE0[Note]	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable

Condition for clearing (SPIE0 bit = 0)	Condition for setting (SPIE0 bit = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIM0[Note]	Control of wait and interrupt request generation
0	Interrupt request is generated at the eighth clock's falling edge. Master mode: After output of eight clocks, clock output is set to low level and wait is set. Slave mode: After input of eight clocks, the clock is set to low level and wait is set for master device.
1	Interrupt request is generated at the ninth clock's falling edge. Master mode: After output of nine clocks, clock output is set to low level and wait is set. Slave mode: After input of nine clocks, the clock is set to low level and wait is set for master device.

An interrupt is generated at the falling of the 9th clock during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. When in master mode, a wait is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a wait is inserted at the falling edge of the ninth clock after $\overline{ACK}$ is issued. However, when the slave device has received an extension code, a wait is inserted at the falling edge of the eighth clock.

Condition for clearing (WTIM0 bit = 0)	Condition for setting (WTIM0 bit = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKE0[Note]	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDA line is set to low level.

The ACKE0 bit setting is invalid for address reception. In this case, $\overline{ACK}$ is generated when the addresses match. However, the ACKE0 bit setting is valid for address reception of the extension code.

Condition for clearing (ACKE0 bit = 0)	Condition for setting (ACKE0 bit = 1)
• Cleared by instruction • Reset	• Set by instruction

Note This flag's signal is invalid when the IICE0 bit = 0.

(3/4)

STT0	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in STOP mode): Generate a start condition (for starting as master). The SDA line is changed from high level to low level while the SCL line is high level and then the start condition is generated. Next, after the rated amount of time has elapsed, the SCL line is changed to low level (wait status). When a third party is communicating • When communication reservation function is enabled (IICF0.IICRSV0 bit = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. • When communication reservation function is disabled (IICRSV0 bit = 1) The IICF0.STCF0 bit is set to 1 and the information set (1) to the STT0 bit is cleared. No start condition is generated. In the wait state (when master device): Generates a restart condition after releasing the wait.

Cautions concerning set timing

For master reception: Cannot be set to 1 during transfer. Can be set to 1 only when the ACKE0 bit has been cleared to 0 and slave has been notified of final reception.

For master transmission: A start condition may not be generated normally during the $\overline{\text{ACK}}$ period. Set to 1 during the wait period that follows output of the ninth clock.

- Cannot be set to 1 at the same time as the SPT0 bit.
- When the STT0 bit is set to 1, setting the STT0 bit to 1 again is disabled until the setting is cleared to 0.

Condition for clearing (STT0 bit = 0)	Condition for setting (STT0 bit = 1)
• When the STT0 bit is set to 1 in the communication reservation disabled status • Cleared by loss in arbitration • Cleared when start condition is generated by master device • When the LREL0 bit = 1 (exit from communications) • When the IICE0 bit = 0 (operation stop) • Reset	• Set by instruction

Remark The STT0 bit is 0 if it is read after data setting.

(4/4)

SPT0	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master device's transfer). After the SDA line goes to low level, either set the SCL line to high level or wait until the SCL pin goes to high level. Next, after the rated amount of time has elapsed, the SDA line is changed from low level to high level and a stop condition is generated.

Cautions concerning setting timing

For master reception: Cannot be set to 1 during transfer. Can be set to 1 only when the ACKE0 bit has been cleared to 0 and during the wait period after slave has been notified of final reception.

For master transmission: A stop condition may not be generated normally during the $\overline{\text{ACK}}$ period. Set to 1 during the wait period that follows output of the ninth clock.

- Cannot be set to 1 at the same time as the STT0 bit.
- The SPT0 bit can be set to 1 only when in master mode[Note].
- When the WTIM0 bit has been cleared to 0, if the SPT0 bit is set to 1 during the wait period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock.
  The WTIM0 bit should be changed from 0 to 1 during the wait period following output of eight clocks, and the SPT0 bit should be set to 1 during the wait period that follows output of the ninth clock.
- When the SPT0 bit is set to 1, setting the SPT0 bit to 1 again is disabled until the setting is cleared to 0.

Condition for clearing (SPT0 bit = 0)	Condition for setting (SPT0 bit = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • When the LREL0 bit = 1 (exit from communications) • When the IICE0 bit = 0 (operation stop) • Reset	• Set by instruction

Note Set the SPT0 bit to 1 only in master mode. However, the SPT0 bit must be set to 1 and a stop condition generated before the first stop condition is detected following the switch to operation enable status. For details, see **17.15 Cautions**.

Caution When the IICS0.TRC0 bit is set to 1, the WREL0 bit is set to 1 during the ninth clock and wait is canceled, after which the TRC0 bit is cleared to 0 and the SDA line is set to high impedance.

Remark The SPT0 bit is 0 if it is read after data setting.

(2) IIC status register 0 (IICS0)

The IICS0 register indicates the status of the I²C bus.

The IICS0 register is read-only, in 8-bit or 1-bit units.

However, the IICS0 register can only be read when the IICC0.STT0 bit is 1 or during the wait period.

Reset sets this register to 00H.

(1/3)

After reset: 00H R Address: FFFFFD86H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	Master device status
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS0 bit = 0)	Condition for setting (MSTS0 bit = 1)
• When a stop condition is detected • When the ALD0 bit = 1 (arbitration loss) • Cleared by the IICC0.LREL0 bit = 1 (exit from communications) • When the IICC0.IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD0	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. The MSTS0 bit is cleared to 0.

Condition for clearing (ALD0 bit = 0)	Condition for setting (ALD0 bit = 1)
• Automatically cleared after the IICS0 register is read[Note] • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

Note This bit is also cleared when a bit manipulation instruction is executed for another bit in the IICS0 register.

(2/3)

EXC0	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC0 bit = 0)	Condition for setting (EXC0 bit = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by the LREL0 bit = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

COI0	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI0 bit = 0)	Condition for setting (COI0 bit = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by the LREL0 bit = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 • Reset	• When the received address matches the local address (SVA0 register) (set at the rising edge of the eighth clock).

TRC0	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDA line is set for high impedance.
1	Transmit status. The value in the SO latch is enabled for output to the SDA line (valid starting at the rising edge of the first byte’s ninth clock).

Condition for clearing (TRC0 bit = 0)	Condition for setting (TRC0 bit = 1)
• When a stop condition is detected • Cleared by the LREL0 bit = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Cleared by the IICC0.WREL0 bit = 1[Note] (wait release) • When the ALD0 bit changes from 0 to 1 (arbitration loss) • Reset Master • When “1” is output to the first byte’s LSB (transfer direction specification bit) Slave • When a start condition is detected When not used for communication	Master • When a start condition is generated • When “0” is output to the first byte’s LSB (transfer direction specification bit) Slave • When “1” is input in the first byte’s LSB (transfer direction specification bit)

Note The IICS0.TRC0 bit is cleared to 0 and the SDA line become high impedance when the IICC0.WREL0 bit is set to 1 and wait state is released at the ninth clock with the TRC0 bit = 1.

(3/3)

ACKD0	Detection of $\overline{ACK}$
0	$\overline{ACK}$ was not detected.
1	$\overline{ACK}$ was detected.

Condition for clearing (ACKD0 bit = 0)	Condition for setting (ACKD0 bit = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by the LREL0 bit = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• After the SDA pin is set to low level at the rising edge of the SCL pin's ninth clock

STD0	Detection of start condition
0	Start condition was not detected.
1	Start condition was detected. This indicates that the address transfer period is in effect

Condition for clearing (STD0 bit = 0)	Condition for setting (STD0 bit = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by the LREL0 bit = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPD0	Detection of stop condition
0	Stop condition was not detected.
1	Stop condition was detected. The master device's communication is terminated and the bus is released.

Condition for clearing (SPD0 bit = 0)	Condition for setting (SPD0 bit = 1)
• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

(3) IIC flag register 0 (IICF0)

IICF0 is a register that set the operation mode of I²C and indicate the status of the I²C bus.

These registers can be read or written in 8-bit or 1-bit units. However, the STCF0 and IICBSY0 bits are read-only.

The IICRSV0 bit can be used to enable/disable the communication reservation function (see **17.14 Communication Reservation**).

The STCEN0 bit can be used to set the initial value of the IICBSY0 bit (see **17.15 Cautions**).

The IICRSV0 and STCEN0 bits can be written only when the operation of I²C is disabled (IICC0.IICE0 bit = 0).

When operation is enabled, the IICF0 register can be read.

Reset sets this register to 00H.

After reset: 00H R/W[Note] Address: FFFFFD8AH

	<7>	<6>	5	4	3	2	<1>	<0>
IICF0	STCF0	IICBSY0	0	0	0	0	STCEN0	IICRSV0

STCF0	IICC0.STT0 clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear STT0 flag

Condition for clearing (STCF0 bit = 0)	Condition for setting (STCF0 bit = 1)
• Clearing by setting the STT0 bit = 1 • When the IICE0 bit = 1 → 0 (operation stop) • Reset	• Generating start condition unsuccessful and the STT0 bit cleared to 0 when communication reservation is disabled (IICRSV0 bit = 1).

IICBSY0	I²C bus status flag
0	Bus release status (initial communication status when STCEN0 bit = 1)
1	Bus communication status (initial communication status when STCEN0 bit = 0)

Condition for clearing (IICBSY0 bit = 0)	Condition for setting (IICBSY0 bit = 1)
• Detection of stop condition • When the IICE0 bit = 1 → 0 (operation stop) • Reset	• Detection of start condition • Setting of the IICE0 bit when the STCEN0 bit = 0

STCEN0	Initial start enable trigger
1	After operation is enabled (IICE0 bit = 1), enable generation of a start condition upon detection of a stop condition.
	After operation is enabled (IICE0 bit = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCEN0 bit = 0)	Condition for setting (STCEN0 bit = 1)
• Detection of start condition • Reset	• Setting by instruction

IICRSV0	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSV0 bit = 0)	Condition for setting (IICRSV0 bit = 1)
• Clearing by instruction • Reset	• Setting by instruction

Note Bits 6 and 7 are read-only bits.

Cautions 1. Write to the STCEN0 bit only when the operation is stopped (IICE0 bit = 0).

2. As the bus release status (IICBSY0 bit = 0) is recognized regardless of the actual bus status when the STCEN0 bit = 1, when generating the first start condition (STT0 bit = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

3. Write to the IICRSV0 bit only when the operation is stopped (IICE0 bit = 0).

(4) IIC clock select register 0 (IICCL0)

The IICCL0 register is used to set the transfer clock for the I²C bus.

The IICCL0 register can be read or written in 8-bit or 1-bit units. However, the CLD0 and DAD0 bits are read-only. The SMC0 and CL00 bits are set in combination with the IICX0.CLX0, IICOCKS.IICOCKS1, and IICOCKS.IICOCKS0 bits (see **17.4 (7) I²C transfer clock setting method**).

Set the IICCL0 register when the IICC0.IICE0 bit = 0.

Reset sets this register to 00H.

After reset: 00H R/W[Note] Address: FFFFFD84H

	7	6	<5>	<4>	3	2	1	0
IICCL0	0	0	CLD0	DAD0	SMC0	DFC0	0	CL00

CLD0	Detection of SCL pin level (valid only when IICC0.IICE0 bit = 1)
0	The SCL pin was detected at low level.
1	The SCL pin was detected at high level.

Condition for clearing (CLD0 bit = 0)	Condition for setting (CLD0 bit = 1)
• When the SCL pin is at low level • When the IICE0 bit = 1 → 0 (operation stop) • Reset	• When the SCL pin is at high level

DAD0	Detection of SDA pin level (valid only when IICE0 bit = 1)
0	The SDA pin was detected at low level.
1	The SDA pin was detected at high level.

Condition for clearing (DAD0 bit = 0)	Condition for setting (DAD0 bit = 1)
• When the SDA pin is at low level • When the IICE0 bit = 1 → 0 (operation stop) • Reset	• When the SDA pin is at high level

SMC0	Operation mode switching
0	Operates in standard mode.
1	Operates in high-speed mode.

DFC0	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Digital filter can be used only in high-speed mode. In high-speed mode, the transfer clock does not vary regardless of DFC0 bit set/clear. The digital filter is used for noise elimination in high-speed mode.	

CL00	Communication clock selection	
	Normal mode	High-speed mode
0	Fxx/44	Fxx/24
1	Fxx/86	Fxx/24

Note Bits 4 and 5 are read-only bits.

Remark Fxx: Selection clock

(5) IIC function expansion register 0 (IICX0)

This register sets the function expansion of I²C (valid only in high-speed mode).

This register can be read or written in 8-bit or 1-bit units. The CLX0 bit is set in combination with the IICCL0.SMC0, IICCL0.CL00, IICOCKS.IICOCKS1, and IICOCKS.IICOCKS0 bits (see **17.4 (7) I²C transfer clock setting method**).

Set the IICX0 register when the IICC0.IICE0 bit = 0.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFD85H

	7	6	5	4	3	2	1	<0>
IICX0	0	0	0	0	0	0	0	CLX0

CLX0	Clock select expansion bit
0	Communicate at transfer rate set by the IICCL0.CL00 bit.
1	Communicate at double transfer rate set by the IICCL0.CL00 bit in high-speed mode .

(6) IICOPS clock select register (IICOCKS)

This register controls the division clock of I²C.

This register can be read or written in 8-bit or 1-bit units. The IICOCKS1 and IICOCKS0 bits are set in combination with the IICCL0.SMC0, IICCL0.CL00, and IICX0.CLX0 bits (see **17.4 (7) I²C transfer clock setting method**).

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFD90H

	7	6	5	4	3	2	1	0
IICOCKS	0	0	0	IICOCKSEN	0	0	IICOCKS1	IICOCKS0

IICOCKSEN	Specification of I²C division clock operation
0	I²C division clock operation stop
1	I²C division clock operation enable

IICOCKS1	IICOCKS0	I²C division clock selection
0	0	fxx/4
0	1	fxx/6
1	0	fxx/8
1	1	fxx/10

(7) I²C transfer clock setting method

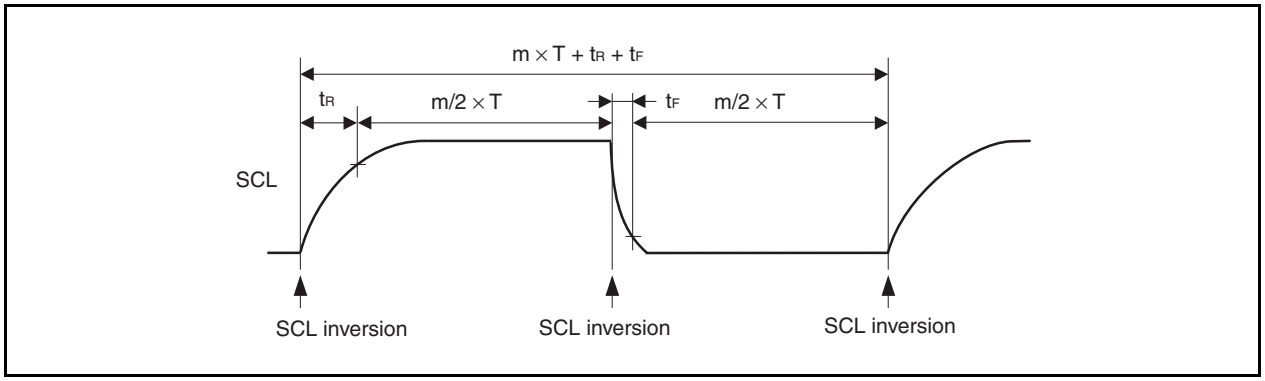
The I²C transfer clock frequency (f_{SCL}) is calculated using the following expression.

$$f_{SCL} = 1/(m \times T + t_R + t_F)$$

m = 96, 120, 144, 192, 240, 344, 352, 440, 516, 688, 860 (see **Table 17-2 Selection Clock Setting**.)
T: 1/f_{XX}
t_R: SCL rise time
t_F: SCL fall time

For example, the I²C transfer clock frequency (f_{SCL}) when f_{XX} = 64 MHz, m = 192, t_R = 200 ns, and t_F = 50 ns is calculated using following expression.

$$f_{SCL} = 1/(192 \times 15.6 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 308 \text{ kHz}$$



The selection clock is set using a combination of the IICCL0.SMC0, IICCL0.CL00, IICX0.CLX0, IICOCKS.IICOCKS1, and IICOCKS.IICOCKS0 bits.

Table 17-2. Selection Clock Setting

IICX0	IICCL0		Selection Clock	Transfer Clock (f_{XX}/m)	Settable Internal System Clock Frequency (f_{XX}) Range	Operation Mode
Bit 0	Bit 3	Bit 0				
CLX0	SMC0	CL00				
0	0	0	f_{XX}/8 (when IICOCKS = 12H)	f_{XX}/352	32.00 MHz to 33.52 MHz	Normal mode (SMC0 bit = 0)
			f_{XX}/10 (when IICOCKS = 13H)	f_{XX}/440	32.00 MHz to 41.90 MHz	
0	0	1	f_{XX}/4 (when IICOCKS = 10H)	f_{XX}/344	32.00 MHz to 33.52 MHz	
			f_{XX}/6 (when IICOCKS = 11H)	f_{XX}/516	32.00 MHz to 50.28 MHz	
			f_{XX}/8 when (IICOCKS = 12H)	f_{XX}/688	33.52 MHz to 64.00 MHz	
			f_{XX}/10 (when IICOCKS = 13H)	f_{XX}/860	41.90 MHz to 64.00 MHz	
0	1	x	f_{XX}/4 (when IICOCKS = 10H)	f_{XX}/96	32.00 MHz to 33.52 MHz	High-speed mode (SMC0 bit = 1)
			f_{XX}/6 (when IICOCKS = 11H)	f_{XX}/144	32.00 MHz to 50.28 MHz	
			f_{XX}/8 (when IICOCKS = 12H)	f_{XX}/192	32.00 MHz to 64.00 MHz	
			f_{XX}/10 (when IICOCKS = 13H)	f_{XX}/240	40.00 MHz to 64.00 MHz	
1	0	x	Setting prohibited			
1	1	x	f_{XX}/8 (when IICOCKS = 12H)	f_{XX}/96	32.00 MHz to 33.52 MHz	High-speed mode (SMC0 bit = 1)
			f_{XX}/10 (when IICOCKS = 13H)	f_{XX}/120	40.00 MHz to 41.90 MHz	

Remark x: don't care

(8) IIC shift register 0 (IIC0)

The IIC0 shift register is used for serial transmission/reception (shift operations) that is synchronized with the serial clock.

The IIC0 shift register can be read or written in 8-bit units, but data should not be written to the IIC0 shift register during a data transfer.

Access (read/write) the IIC0 shift register only during the wait period. Accessing this register in communication states other than the wait period is prohibited. However, for the master device, the IIC0 shift register can be written once only after the transmission trigger bit (IICC0.STT0 bit) has been set to 1.

When the IIC0 shift register is written during wait, the wait is cancelled and data transfer is started.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFD80H

	7	6	5	4	3	2	1	0
IIC0								

(9) Slave address register 0 (SVA0)

The SVA0 register holds the I²C bus's slave addresses.

However, rewriting this register is prohibited when the IICS0.STD0 bit = 1 (start condition detection).

The SVA0 register can be read or written in 8-bit units, but bit 0 is fixed to 0.

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFD83H

	7	6	5	4	3	2	1	0
SVA0								0

17.5 Functions

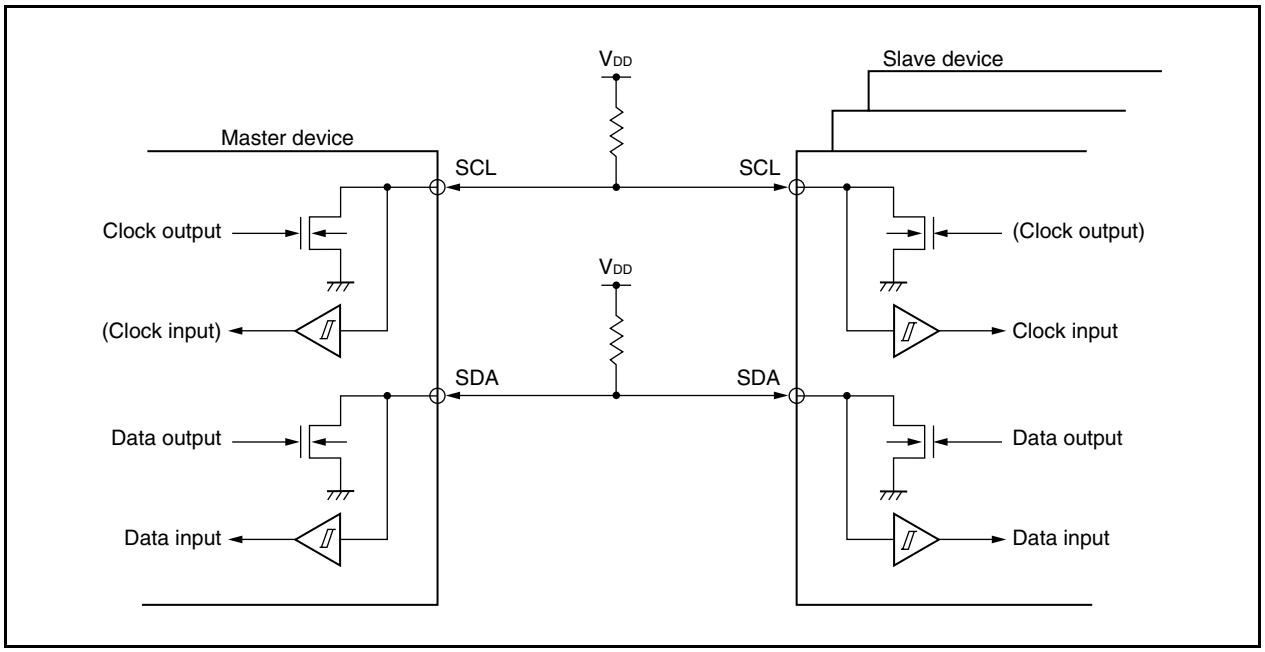
17.5.1 Pin configuration

The serial clock pin (SCL) and serial data bus pin (SDA) are configured as follows.

SCL ..... This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.
SDA ..... This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

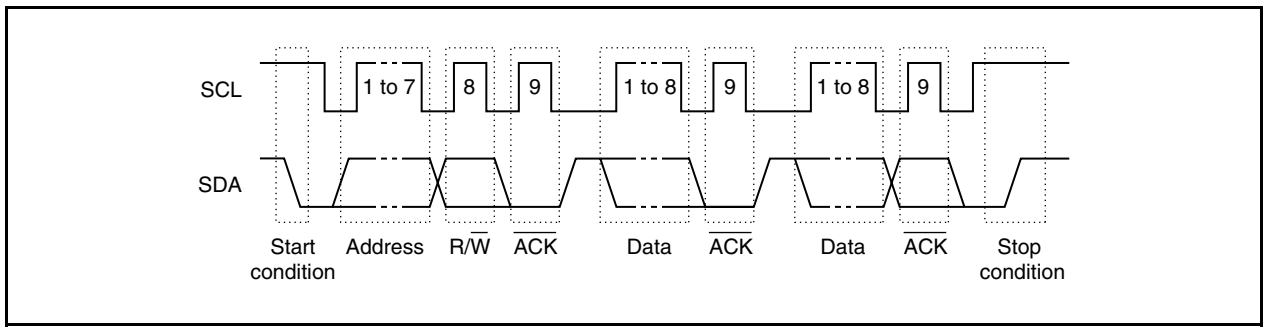
Figure 17-4. Pin Configuration Diagram



17.6 I²C Bus Definitions and Control Methods

The following section describes the I²C bus's serial data communication format and the status generated by the I²C bus. The transfer timing for the "start condition", "address", "transfer direction specification", "data", and "stop condition" generated via the I²C bus's serial data bus is shown below.

Figure 17-5. I²C Bus's Serial Data Transfer Timing



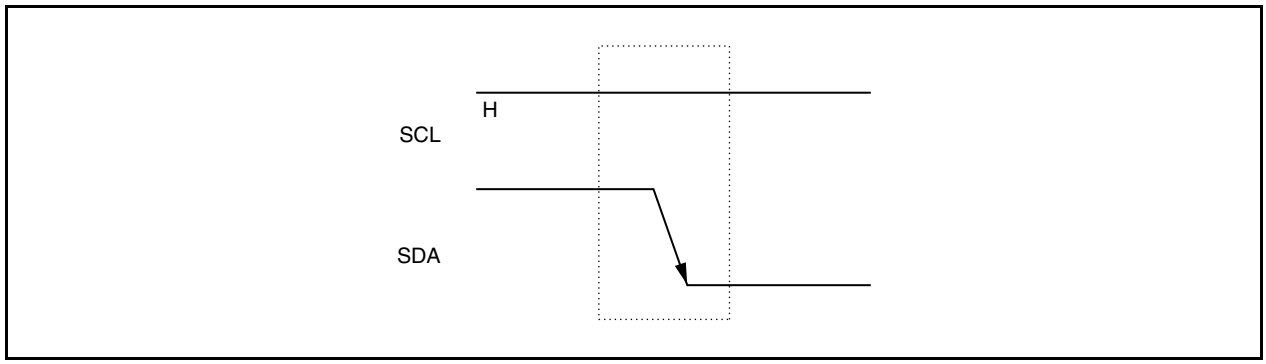
The master device generates the start condition, slave address, and stop condition.
ACK can be generated by either the master or slave device (normally, it is generated by the device that receives 8-bit data).

The serial clock (SCL) is continuously output by the master device. However, in the slave device, the SCL's low-level period can be extended and a wait can be inserted.

17.6.1 Start condition

A start condition is met when the SCL pin is at high level and the SDA pin changes from high level to low level. The start conditions for the SCL pin and SDA pin are generated when the master device starts a serial transfer to the slave device. Start conditions can be detected when the device is used as a slave.

Figure 17-6. Start Conditions



A start condition is generated when the IICC0.STT0 bit is set to 1 after a stop condition has been detected (IICS0.SPD0 bit = 1). When a start condition is detected, IICS0.STD0 bit is set to 1.

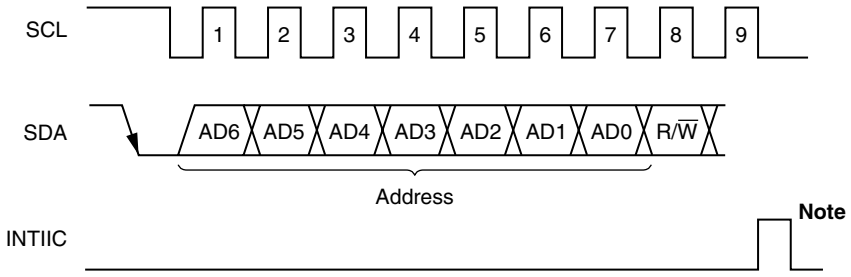
17.6.2 Addresses

The 7 bits of data that follow the start condition are defined as an address.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the SVA0 register. If the address data matches the SVA0 values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 17-7. Address



Note The interrupt request signal (INTIIC) is generated if a local address or extension code is received during slave device operation.

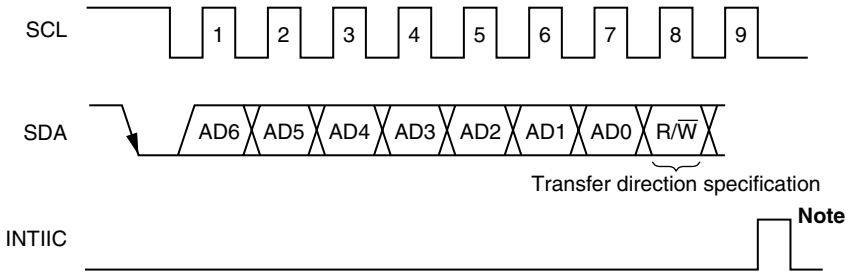
The slave address and the eighth bit, which specifies the transfer direction as described in **17.6.3 Transfer direction specification** below, are together written to the IIC0 register and are then output. Received addresses are written to the IIC0 register.

The slave address is assigned to the higher 7 bits of the IIC0 register.

17.6.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction. When this transfer direction specification bit has a value of 0, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of 1, it indicates that the master device is receiving data from a slave device.

Figure 17-8. Transfer Direction Specification



Note The interrupt request signal (INTIIC) is generated if a local address or extension code is received during slave device operation.

17.6.4 $\overline{\text{ACK}}$

$\overline{\text{ACK}}$ is used to confirm the serial data status of the transmitting and receiving devices.

The receiving device returns $\overline{\text{ACK}}$ for every 8 bits of data it receives.

The transmitting device normally receives $\overline{\text{ACK}}$ after transmitting 8 bits of data. When $\overline{\text{ACK}}$ is returned from the receiving device, the reception is judged as normal and processing continues. The detection of $\overline{\text{ACK}}$ is confirmed with the IICS0.ACKD0 bit.

When the master device is the receiving device, after receiving the final data, it does not return $\overline{\text{ACK}}$ and generates the stop condition. When the slave device is the receiving device and does not return $\overline{\text{ACK}}$, the master device generates either a stop condition or a restart condition, and then stops the current transmission. Failure to return $\overline{\text{ACK}}$ may be caused by the following factors.

(a) Reception was not performed normally.
(b) The final data was received.
(c) The receiving device (slave) does not exist for the specified address.

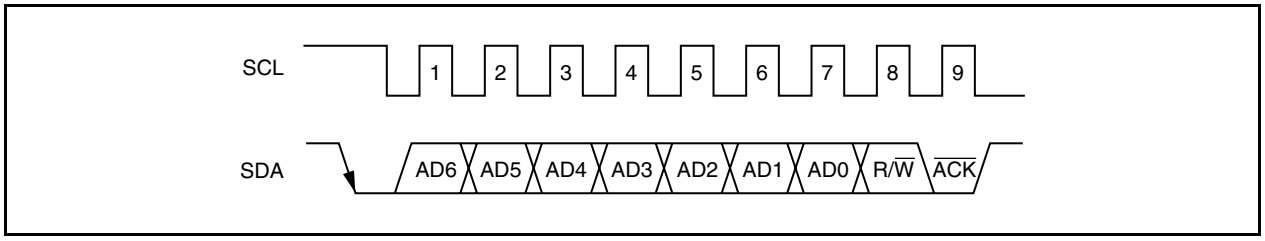
When the receiving device sets the SDA line to low level during the ninth clock, $\overline{\text{ACK}}$ is generated (normal reception).

When the IICC0.ACKE0 bit is set to 1, automatic $\overline{\text{ACK}}$ generation is enabled. Transmission of the eighth bit following the 7 address data bits causes the IICS0.TRC0 bit to be set. Normally, set the ACKE0 bit to 1 for reception (TRC0 bit = 0).

When the slave device is receiving (when TRC0 bit = 0), if the slave device cannot receive data or does not need to receive any more data, clear the ACKE0 bit to 0 to indicate to the master that no more data can be received.

Similarly, when the master device is receiving (when TRC0 bit = 0) and the subsequent data is not needed, clear the ACKE0 bit to 0 to prevent $\overline{\text{ACK}}$ from being generated. This notifies the slave device (transmitting device) of the end of the data transmission (transmission stopped).

Figure 17-9. $\overline{\text{ACK}}$



When the local address is received, $\overline{\text{ACK}}$ is automatically generated regardless of the value of the ACKE0 bit. No $\overline{\text{ACK}}$ is generated if the received address is not a local address (NACK).

When receiving the extension code, set the ACKE0 bit to 1 in advance to generate $\overline{\text{ACK}}$.

The $\overline{\text{ACK}}$ generation method during data reception is based on the wait timing setting, as described by the following.

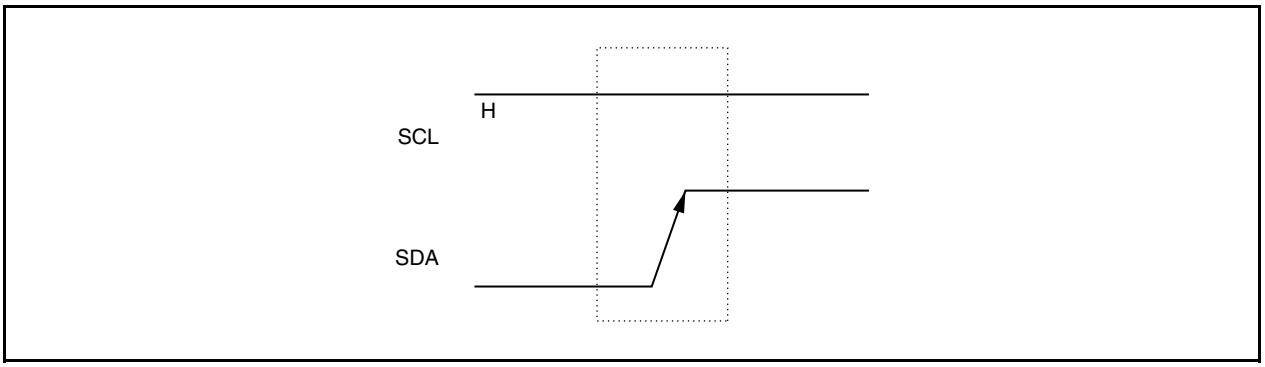
- When 8-clock wait is selected (IICC0.WTIM0 bit = 0):
  $\overline{\text{ACK}}$ is generated at the falling edge of the SCL pin's eighth clock if the ACKE0 bit is set to 1 before the wait state cancellation.
- When 9-clock wait is selected (IICC0.WTIM0 bit = 1):
  $\overline{\text{ACK}}$ is generated if the ACKE0 bit is set to 1 in advance.

17.6.5 Stop condition

When the SCL pin is at high level, changing the SDA pin from low level to high level generates a stop condition.

A stop condition is generated when serial transfer from the master device to the slave device has been completed. Stop conditions can be detected when the device is used as a slave.

Figure 17-10. Stop Condition



A stop condition is generated when the IICC0.SPT0 bit is set to 1. When the stop condition is detected, the IICS0.SPD0 bit is set to 1 and the interrupt request signal (INTIIC) is generated when the IICC0.SPIE0 bit is set to 1.

17.6.6 Wait state

The wait state is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCL pin to low level notifies the communication partner of the wait status. When wait status has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 17-11. Wait State (1/2)

(a) When master device has a nine-clock wait and slave device has an eight-clock wait (master: transmission, slave: reception, and IICC0.ACKE0 bit = 1)

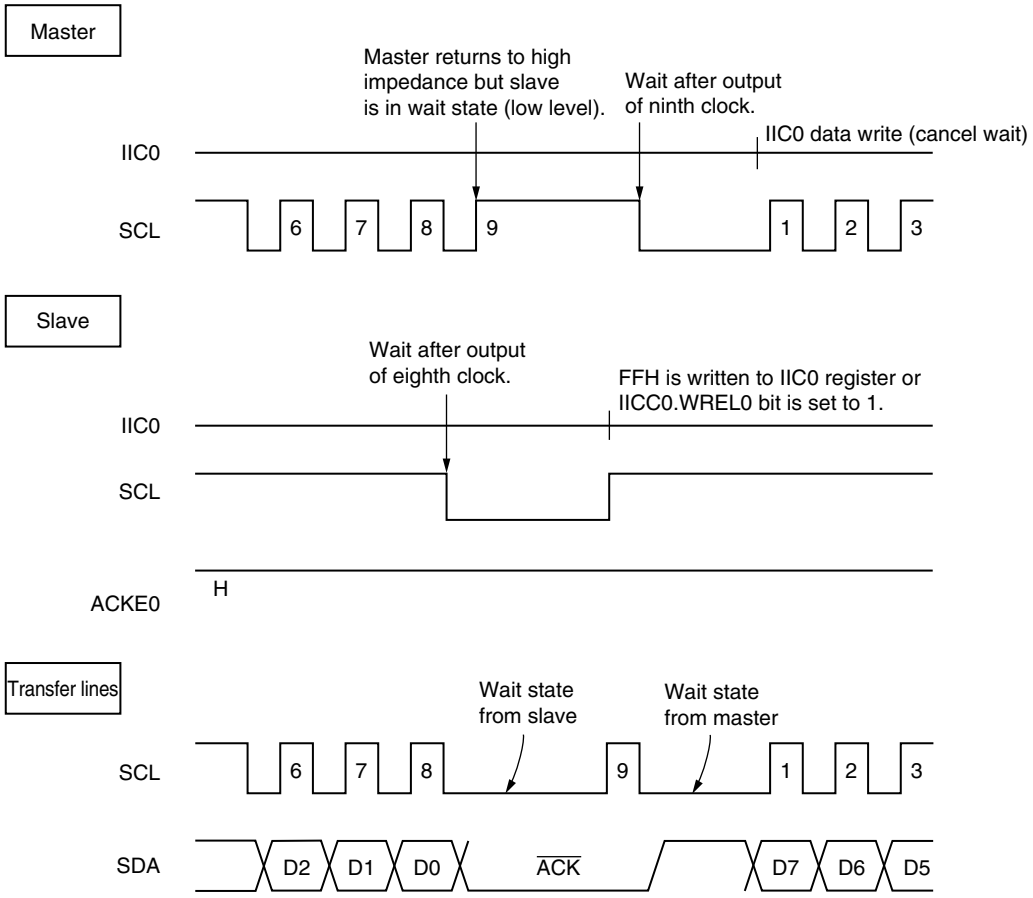
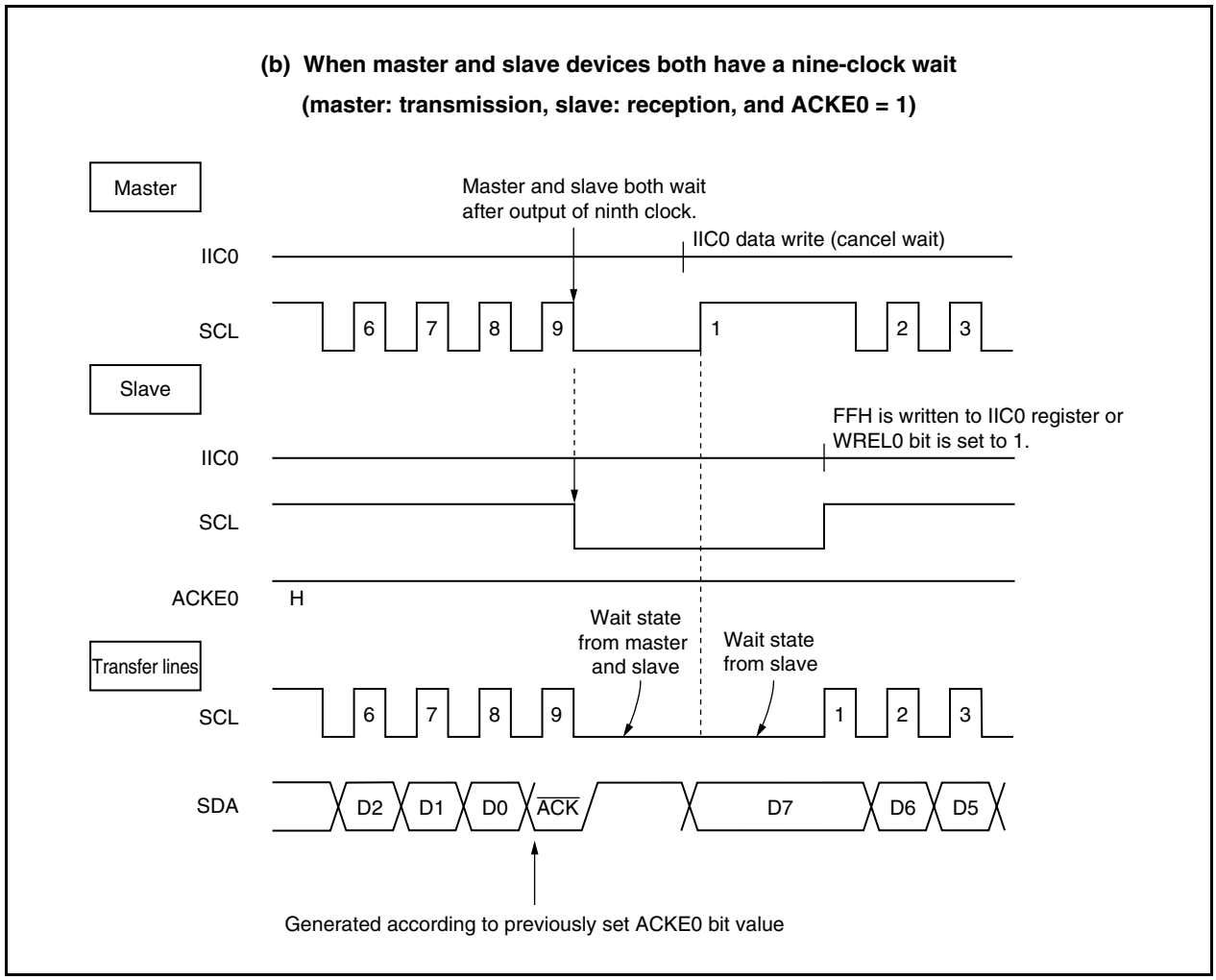


Figure 17-11. Wait State (2/2)



A wait state is automatically generated after a start condition is generated. Moreover, a wait state is automatically generated depending on the setting of the IICC0.WTIM0 bit.

Normally, when the IICC0.WREL0 bit is set to 1 or when FFH is written to the IIC0 register, the wait status is canceled and the transmitting side writes data to the IIC0 register to cancel the wait status.

The master device can also cancel the wait status via either of the following methods.

- By setting the IICC0.STT0 bit to 1
- By setting the IICC0.SPT0 bit to 1

17.6.7 Wait state cancellation method

In the case of I²C, wait state can be canceled normally in the following ways.

- By writing data to the IIC0 register
- By setting the IICC0.WREL0 bit to 1 (wait state cancellation)
- By setting the IICC0.STT0 bit to 1 (start condition generation)[Note]
- By setting the IICC0.SPT0 bit to 1 (stop condition generation)[Note]

Note Master only

If any of these wait state cancellation actions is performed, I²C will cancel wait state and restart communication.

When canceling wait state and sending data (including address), write data to the IIC0 register.

To receive data after canceling wait state, or to end data transmission, set the WREL0 bit to 1.

To generate a restart condition after canceling wait state, set the STT0 bit to 1.

To generate a stop condition after canceling wait state, set the SPT0 bit to 1.

Execute cancellation only once for each wait state.

For example, if data is written to the IIC0 register following wait state cancellation by setting the WREL0 bit to 1, conflict between the SDA line change timing and IIC0 register write timing may result in the data output to the SDA line may be incorrect.

Even in other operations, if communication is stopped halfway, clearing the IICC0.IICE0 bit to 0 will stop communication, enabling wait state to be cancelled.

If the I²C bus dead-locks due to noise, etc., setting the IICC0.LREL0 bit to 1 causes the communication operation to be exited, enabling wait state to be cancelled.

17.7 I²C Interrupt Request Signals (INTIIC)

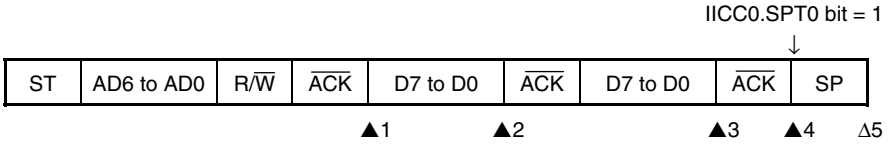
The following shows the value of the IICS0 register at the INTIIC interrupt request signal generation timing and at the INTIIC signal timing.

Remark ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
$\overline{ACK}$: Acknowledge
D7 to D0: Data
SP: Stop condition

17.7.1 Master device operation

(1) Start ~ Address ~ Data ~ Data ~ Stop (normal transmission/reception)

<1> When IICC0.WTIM0 bit = 0



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X000B

▲3: IICS0 register = 1000X000B (WTIM0 bit = 1[Note])

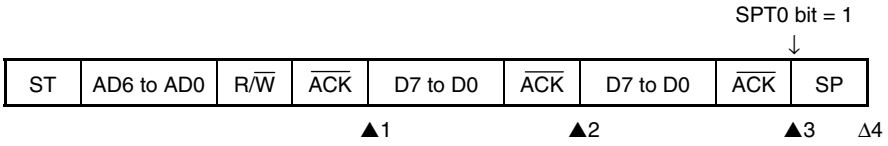
▲4: IICS0 register = 1000XX00B

Δ 5: IICS0 register = 00000001B

Note To generate a stop condition, set the WTIM0 bit to 1 and change the timing of the generation of the interrupt request signal (INTIIC).

Remark ▲: Always generated
Δ: Generated only when IICC0.SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X100B

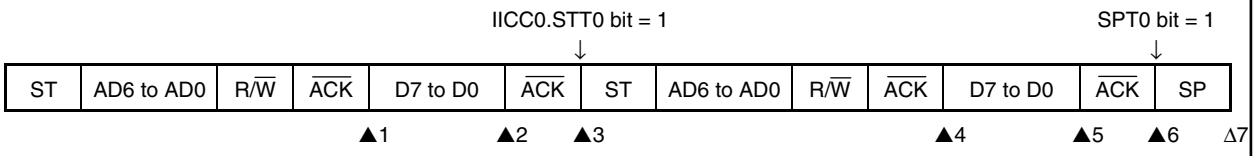
▲3: IICS0 register = 1000XX00B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

<1> When WTIM0 bit = 0



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X000B (WTIM0 bit = 1[Note 1])

▲3: IICS0 register = 1000XX00B (WTIM0 bit = 0[Note 2])

▲4: IICS0 register = 1000X110B

▲5: IICS0 register = 1000X000B (WTIM0 bit = 1[Note 3])

▲6: IICS0 register = 1000XX00B

Δ 7: IICS0 register = 00000001B

Notes 1. To generate a start condition, set the WTIM0 bit to 1 and change the timing of the generation of the interrupt request signal (INTIIC).

2. Clear the WTIM0 bit to 0 to make the settings original.

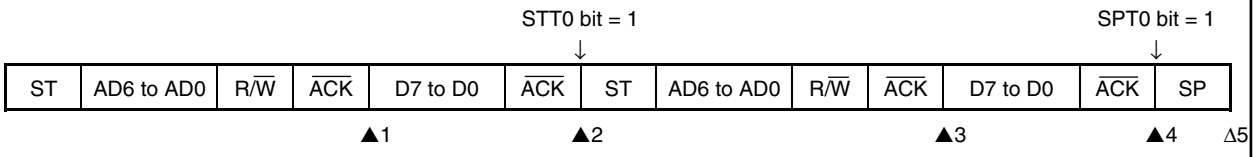
3. To generate a stop condition, set the WTIM0 bit to 1 and change the timing of the generation of the interrupt request signal (INTIIC).

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000XX00B

▲3: IICS0 register = 1000X110B

▲4: IICS0 register = 1000XX00B

Δ 5: IICS0 register = 00000001B

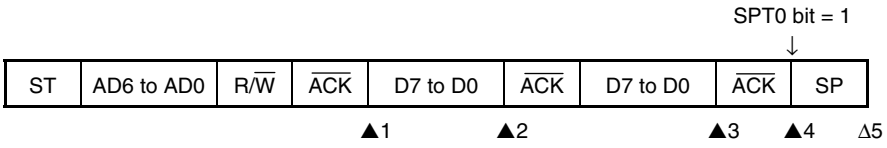
Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

(3) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

<1> When WTIM0 bit = 0



▲1: IICS0 register = 1010X110B

▲2: IICS0 register = 1010X000B

▲3: IICS0 register = 1010X000B (WTIM0 bit = 1[Note])

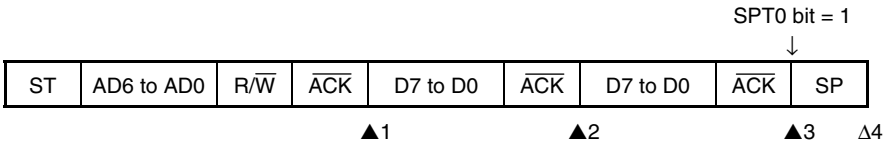
▲4: IICS0 register = 1010XX00B

Δ 5: IICS0 register = 00000001B

Note To generate a stop condition, set the WTIM0 bit to 1 and change the timing of the generation of the interrupt request signal (INTIIC).

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 1010X110B

▲2: IICS0 register = 1010X100B

▲3: IICS0 register = 1010XX00B

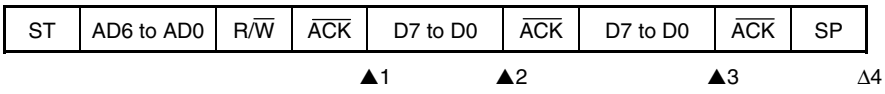
Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

17.7.2 Slave device operation (when receiving slave address data (address match))

(1) Start ~ Address ~ Data ~ Data ~ Stop

<1> When IICC0.WTIM0 bit = 0



▲1: IICS0 register = 0001X110B

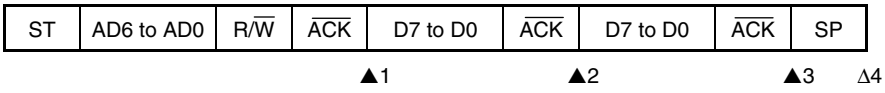
▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0001X000B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when IICC0.SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X100B

▲3: IICS0 register = 0001XX00B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, address match)

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1		▲2					▲3	▲4	Δ5

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0001X110B

▲4: IICS0 register = 0001X000B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

<2> When WTIM0 bit = 1 (after restart, address match)

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1		▲2					▲3	▲4	Δ5

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001XX00B

▲3: IICS0 register = 0001X110B

▲4: IICS0 register = 0001XX00B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

(3) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, address mismatch (extension code))

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1		▲2				▲3		▲4	Δ5

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0010X010B

▲4: IICS0 register = 0010X000B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

<2> When WTIM0 bit = 1 (after restart, address mismatch (extension code))

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1		▲2				▲3	▲4		▲5 Δ6

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001XX00B

▲3: IICS0 register = 0010X010B

▲4: IICS0 register = 0010X110B

▲5: IICS0 register = 0010XX00B

Δ 6: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

(4) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, address mismatch (= not extension code))

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 00000110B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1 (after restart, address mismatch (= not extension code))

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001XX00B

▲3: IICS0 register = 00000110B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

17.7.3 Slave device operation (when receiving extension code)

Always under communication when receiving the extension code.

(1) Start ~ Code ~ Data ~ Data ~ Stop

<1> When IICC0.WTIM0 bit = 0

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0010X000B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when IICC0.SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010X100B

▲4: IICS0 register = 0010XX00B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(2) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, address match)

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0001X110B

▲4: IICS0 register = 0001X000B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

<2> When WTIM0 bit = 1 (after restart, address match)

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 ▲4 ▲5 Δ6

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010XX00B

▲4: IICS0 register = 0001X110B

▲5: IICS0 register = 0001XX00B

Δ 6: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

(3) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, extension code reception)

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0010X010B

▲4: IICS0 register = 0010X000B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1 (after restart, extension code reception)

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 ▲4 ▲5 ▲6 Δ7

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010XX00B

▲4: IICS0 register = 0010X010B

▲5: IICS0 register = 0010X110B

▲6: IICS0 register = 0010XX00B

Δ 7: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(4) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, address mismatch (= not extension code))

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 00000110B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1 (after restart, address mismatch (= not extension code))

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010XX00B

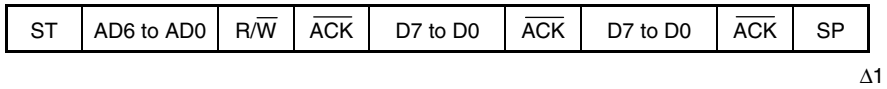
▲4: IICS0 register = 00000110B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

17.7.4 Operation without communication

(1) Start ~ Code ~ Data ~ Data ~ Stop



Δ1

Δ 1: IICS0 register = 00000001B

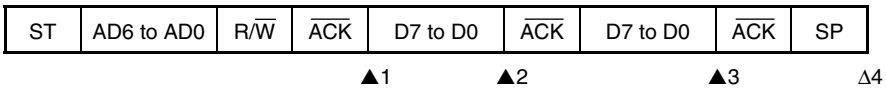
Remark Δ: Generated only when IICC0.SPIE0 bit = 1

17.7.5 Arbitration loss operation (operation as slave after arbitration loss)

When used as master in the multi-master system, check the arbitration result by reading the IICS0.MSTS0 bit for checking arbitration result by each INTIIC interrupt occurrence.

(1) When arbitration loss occurs during transmission of slave address data

<1> When IICC0.WTIM0 bit = 0



▲1: IICS0 register = 0101X110B

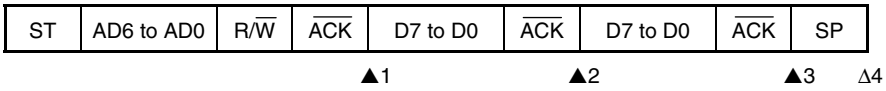
▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0001X000B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when IICC0.SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 0101X110B

▲2: IICS0 register = 0001X100B

▲3: IICS0 register = 0001XX00B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(2) When arbitration loss occurs during transmission of extension code

<1> When WTIM0 bit = 0

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1: IICS0 register = 0110X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0010X000B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICS0 register = 0110X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010X100B

▲4: IICS0 register = 0010XX00B

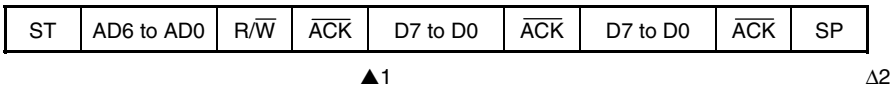
Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

17.7.6 Operation when arbitration loss occurs (no communication after arbitration loss)

When used as master in the multi-master system, check the arbitration result by reading the IICS0.MSTS0 bit for checking arbitration result by each INTIIC interrupt occurrence.

(1) When arbitration loss occurs during transmission of slave address data



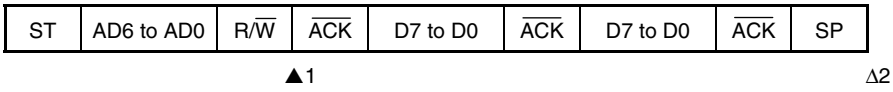
▲1: IICS0 register = 01000110B

Δ 2: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when IICC0.SPIE0 bit = 1

(2) When arbitration loss occurs during transmission of extension code



▲1: IICS0 register = 0110X010B

IICC0.LREL0 bit is set to 1 by software

Δ 2: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

(3) When arbitration loss occurs during data transfer

<1> When IICC0.WTIM0 bit = 0

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 Δ3

▲1: IICS0 register = 10001110B

▲2: IICS0 register = 01000000B

Δ 3: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

<2> When WTIM0 bit = 1

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 Δ3

▲1: IICS0 register = 10001110B

▲2: IICS0 register = 01000100B

Δ 3: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

(4) When arbitration loss occurs due to restart condition during data transfer

<1> Not extension code (Example: Address mismatch)

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to Dn	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1					▲2			Δ3

▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 01000110B

Δ 3: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care
Dn = D6 to D0

<2> Extension code

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to Dn	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1					▲2			Δ3

▲1: IICS0 register = 1000X110B

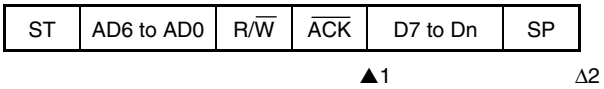
▲2: IICS0 register = 0110X010B

IICC0.LREL0 bit is set to 1 by software

Δ 3: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care
Dn = D6 to D0

(5) When arbitration loss occurs due to stop condition during data transfer



▲1 Δ2

▲1: IICS0 register = 1000X110B

Δ 2: IICS0 register = 01000001B

Remark ▲: Always generated

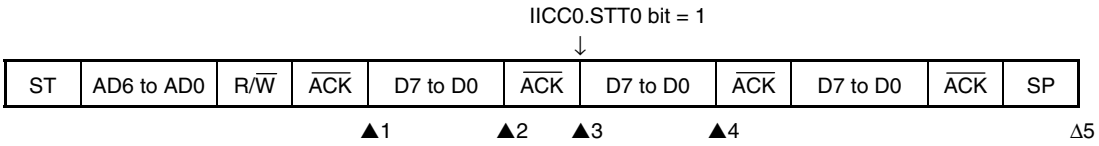
Δ: Generated only when SPIE0 bit = 1

X: don't care

Dn = D6 to D0

(6) When arbitration loss occurs due to low level of SDAn pin when attempting to generate a restart condition

<1> When WTIM0 bit = 0



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X000B (WTIM0 bit = 1)

▲3: IICS0 register = 1000X100B (WTIM0 bit = 0)

▲4: IICS0 register = 01000000B

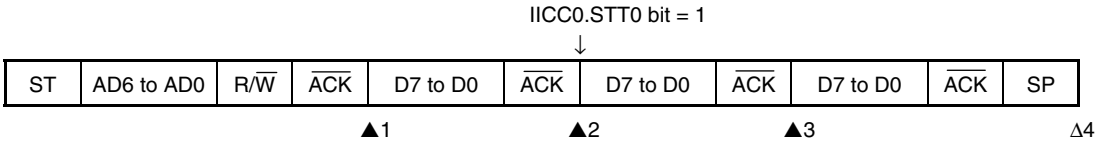
Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X100B

▲3: IICS0 register = 01000100B

Δ 4: IICS0 register = 00000001B

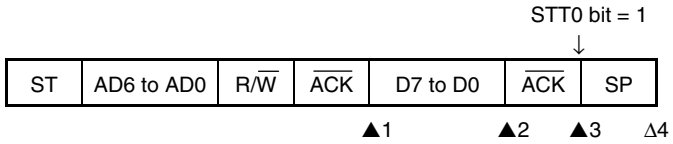
Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

(7) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

<1> When WTIM0 bit = 0



▲1: IICS0 register = 1000X110B

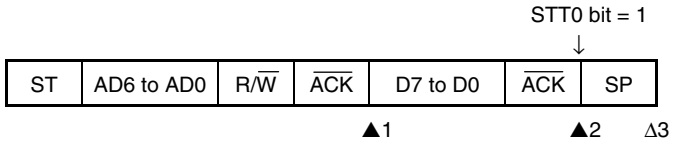
▲2: IICS0 register = 1000X000B (WTIM0 bit = 1)

▲3: IICS0 register = 1000XX00B

Δ 4: IICS0 register = 01000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 1000X110B

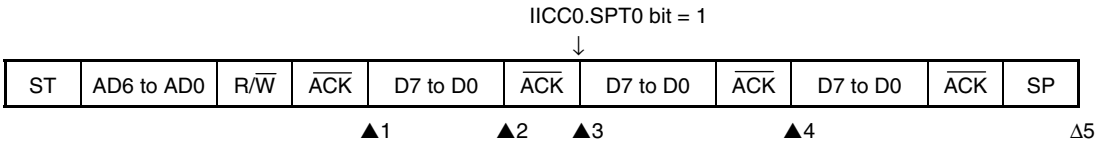
▲2: IICS0 register = 1000XX00B

Δ 3: IICS0 register = 01000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(8) When arbitration loss occurs due to low level of SDAn pin when attempting to generate a stop condition

<1> When WTIM0 bit = 0



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X000B (WTIM0 bit = 1)

▲3: IICS0 register = 1000X100B (WTIM0 bit = 0)

▲4: IICS0 register = 01000100B

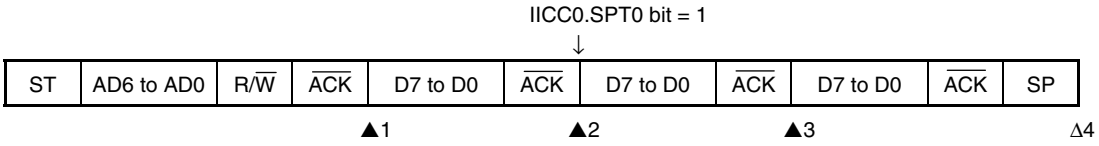
Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X100B

▲3: IICS0 register = 01000100B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

17.8 Interrupt Request Signal (INTIIC) Generation Timing and Wait Control

The setting of the IICC0.WTIM0 bit determines the timing by which the INTIIC signal is generated and the corresponding wait control, as shown below.

Table 17-3. INTIIC Signal Generation Timing and Wait Control

WTIM0 Bit	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9 Notes 1, 2	8 Note 2	8 Note 2	9	8	8
1	9 Notes 1, 2	9 Note 2	9 Note 2	9	9	9

Notes 1. The slave device's INTIIC signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to the SVA0 register.
At this point, $\overline{ACK}$ is generated regardless of the value set to the IICC0.ACKE0 bit. For a slave device that has received an extension code, the INTIIC signal occurs at the falling edge of the eighth clock.
When the address does not match after restart, the INTIIC signal is generated at the falling edge of the ninth clock, but no wait occurs.

2. If the received address does not match the contents of the SVA0 register and extension codes have not been received, neither the INTIIC signal nor a wait occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- Slave device operation: Interrupt and wait timing are determined depending on the conditions in Notes 1 and 2 above regardless of the WTIM0 bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM0 bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- By writing data to the IIC0 register
- By setting the IICC0.WREL0 bit (canceling wait state)
- By setting the IICC0.STT0 bit (generating start condition) Note
- By setting the IICC0.SPT0 bit (generating stop condition) Note

Note Master only

When an 8-clock wait has been selected (WTIM0 bit = 0), whether or not $\overline{ACK}$ has been generated must be determined prior to wait cancellation.

(5) Stop condition detection

The INTIIC signal is generated when a stop condition is detected.

17.9 Address Match Detection Method

When in I²C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match detection is performed automatically by hardware. An INTIIC interrupt request signal occurs when a local address has been set to the SVA0 register and when the address set to the SVA0 register matches the slave address sent by the master device, or when an extension code has been received.

17.10 Error Detection

In I²C bus mode, the status of the serial data bus (SDA) during data transmission is captured by the IIC0 register of the transmitting device, so the IIC0 register data prior to transmission can be compared with the transmitted IIC0 register data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

17.11 Extension Code

(1) When the higher 4 bits of the receive address are either 0000 or 1111, the extension code flag (EXC0) is set for extension code reception and an interrupt request signal (INTIIC) is issued at the falling edge of the eighth clock.
The local address stored in the SVA0 register is not affected.

(2) If 11110xx0 is set to the SVA0 register by a 10-bit address transfer and 11110xx0 is transferred from the master device, the results are as follows. Note that the INTIIC signal occurs at the falling edge of the eighth clock.

- Higher 4 bits of data match: IICS0.EXC0 bit = 1
- 7 bits of data match: IICS0.COI0 bit = 1

(3) Since the processing after the INTIIC signal occurs differs according to the data that follows the extension code, such processing is performed by software. The slave that has received an extension code is always under communication, even if the addresses mismatch.
For example, when operation as a slave is not desired after the extension code is received, set the IICC0.LREL0 bit to 1 and the CPU will enter the next communication wait state.

Table 17-4. Extension Code Bit Definitions

Slave Address	R/$\overline{W}$ Bit	Description
0000 000	0	General call address
0000 000	1	Start byte
0000 001	X	CBUS address
0000 010	X	Address that is reserved for different bus format
1111 0xx	X	10-bit slave address specification

17.12 Arbitration

When several master devices simultaneously generate a start condition (when the IICC0.STT0 bit is set to 1 before the IICS0.STD0 bit is set to 1), communication among the master devices is performed as the number of clocks is adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (IICS0.ALD0 bit) is set (1) via the timing by which the arbitration loss occurred, and the SCL and SDA lines are both set for high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request signal (INTIIC) (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0 bit = 1 setting that has been made by software.

For details of interrupt request timing, see **17.7 I²C Interrupt Request Signals (INTIIC)**.

Figure 17-12. Arbitration Timing Example

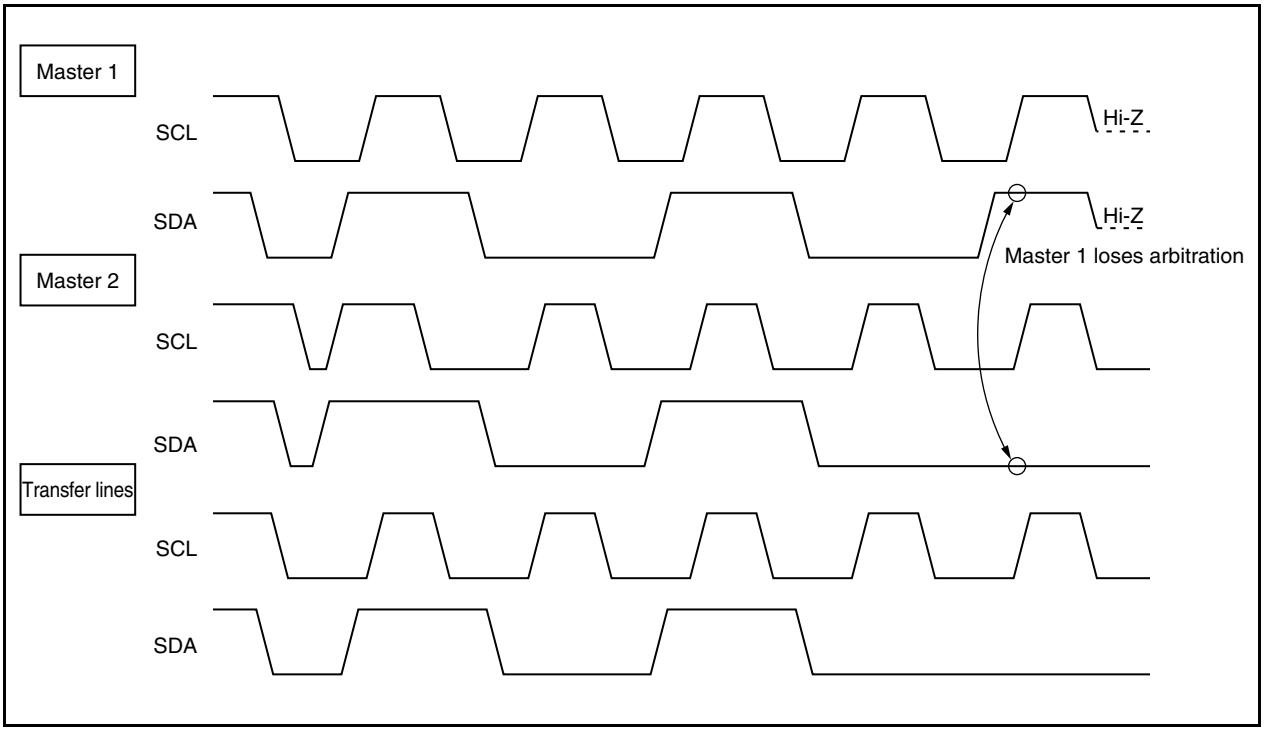


Table 17-5. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ transfer period after data reception	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when IICC0.SPIE0 bit = 1)[Note 2]
When the SDA pin is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIE0 bit = 1)[Note 2]
When the SDA pin is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When the SCL pin is at low level while attempting to generate a restart condition	

Notes 1. When the IICC0.WTIM0 bit = 1, an interrupt request occurs at the falling edge of the ninth clock. When the WTIM0 bit = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

2. When there is a possibility that arbitration will occur, set the SPIE0 bit = 1 for master device operation.

17.13 Wakeup Function

The I²C bus slave function is a function that generates an interrupt request signal (INTIIC) when a local address or extension code has been received.

This function makes processing more efficient by preventing unnecessary interrupt requests from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

However, when a stop condition is detected, the IICC0.SPIE0 bit is set regardless of the wake up function, and this determines whether interrupt requests are enabled or disabled.

17.14 Communication Reservation

17.14.1 When communication reservation function is enabled (IICF0.IICRSV0 bit = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released when the IICC0.LREL0 bit was set to “1”).

If the IICC0.STT0 bit is set (1) while the bus is not used, a start condition is automatically generated and wait status is set after the bus is released (after a stop condition is detected).

A communication is automatically started as the master by setting the IICC0.SPIE0 bit to 1, detecting the bus release due to an interrupt request (INTIIC) occurrence (detecting a stop condition), and then writing the address to the IIC0 register. Before detecting a stop condition, data written to the IIC0 register is set to invalid.

When the STT0 bit has been set (1), the operation mode (as start condition or as communication reservation) is determined according to the bus status.

If the bus has been released .....a start condition is generated
If the bus has not been released (standby mode) .....communication reservation

To detect which operation mode has been determined for the STT0 bit, set the STT0 bit (1), wait for the wait period, then check the IICS0.MSTS0 bit.

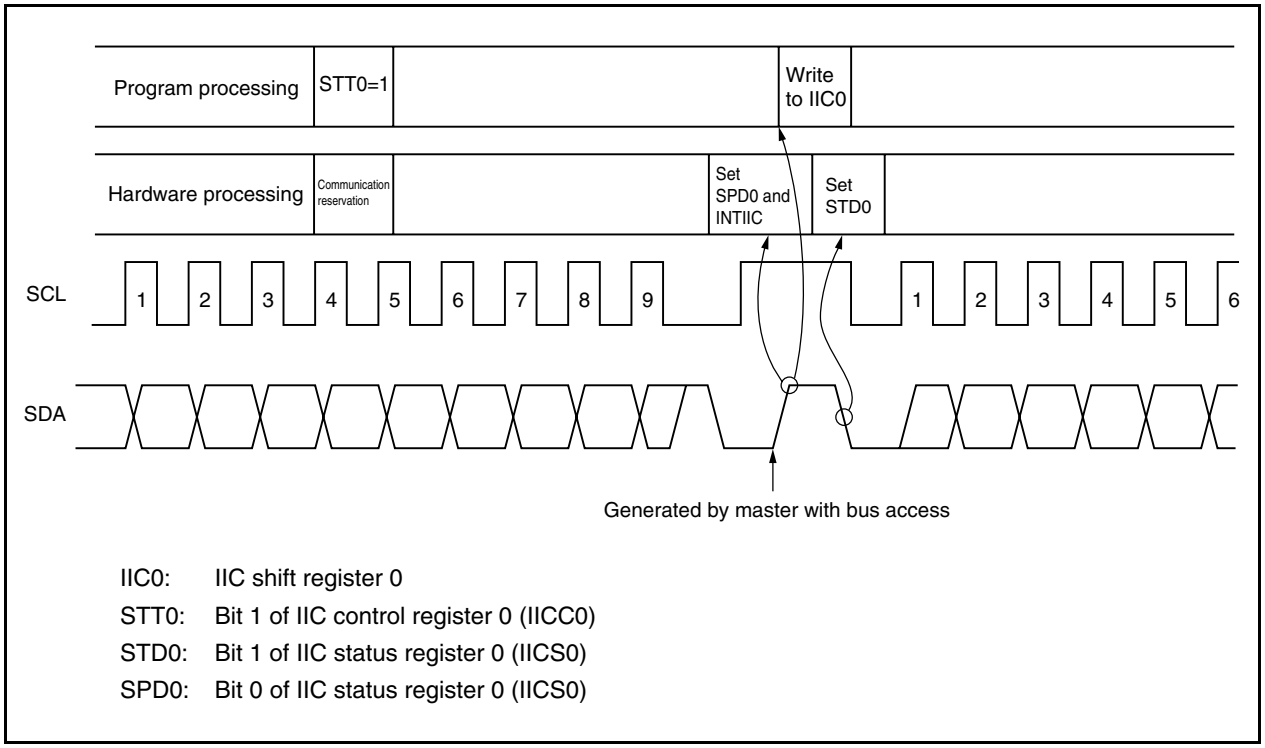
Wait periods, which should be set via software, are listed in Table 17-6. These wait periods can be set via the settings for the IICX0.CLX0, IICCL0.SMC0, and IICCL0.CL00 bits.

Table 17-6. Wait Periods

Selection Clock	CLX0	SMC0	CL00	Wait Clock	Wait Time When f_{XX} = 64 MHz
f_{XX}/8 (IICOCKS = 12H)	0	0	0	23 clocks	2.88 μs
f_{XX}/10 (IICOCKS = 13H)	0	0	0	23 clocks	3.59 μs
f_{XX}/4 (IICOCKS = 10H)	0	0	1	43 clocks	2.69 μs
f_{XX}/6 (IICOCKS = 11H)	0	0	1	43 clocks	4.03 μs
f_{XX}/8 (IICOCKS = 12H)	0	0	1	43 clocks	5.38 μs
f_{XX}/10 (IICOCKS = 13H)	0	0	1	43 clocks	6.72 μs
f_{XX}/4 (IICOCKS = 10H)	0	1	x	15 clocks	0.94 μs
f_{XX}/6 (IICOCKS = 11H)	0	1	x	15 clocks	1.41 μs
f_{XX}/8 (IICOCKS = 12H)	0	1	x	15 clocks	1.88 μs
f_{XX}/10 (IICOCKS = 13H)	0	1	x	15 clocks	2.34 μs
f_{XX}/8 (IICOCKS = 12H)	1	1	x	9 clocks	1.13 μs
f_{XX}/10 (IICOCKS = 13H)	1	1	x	9 clocks	1.41 μs

The communication reservation timing is shown below.

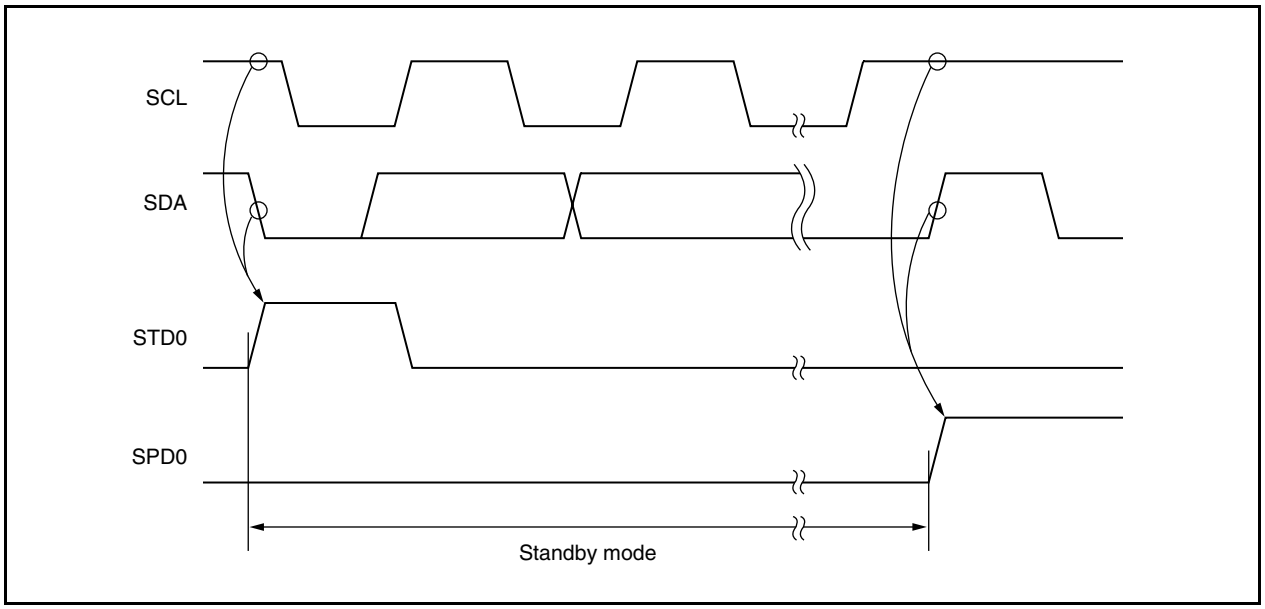
Figure 17-13. Communication Reservation Timing



IIC0: IIC shift register 0
STT0: Bit 1 of IIC control register 0 (IICC0)
STD0: Bit 1 of IIC status register 0 (IICS0)
SPD0: Bit 0 of IIC status register 0 (IICS0)

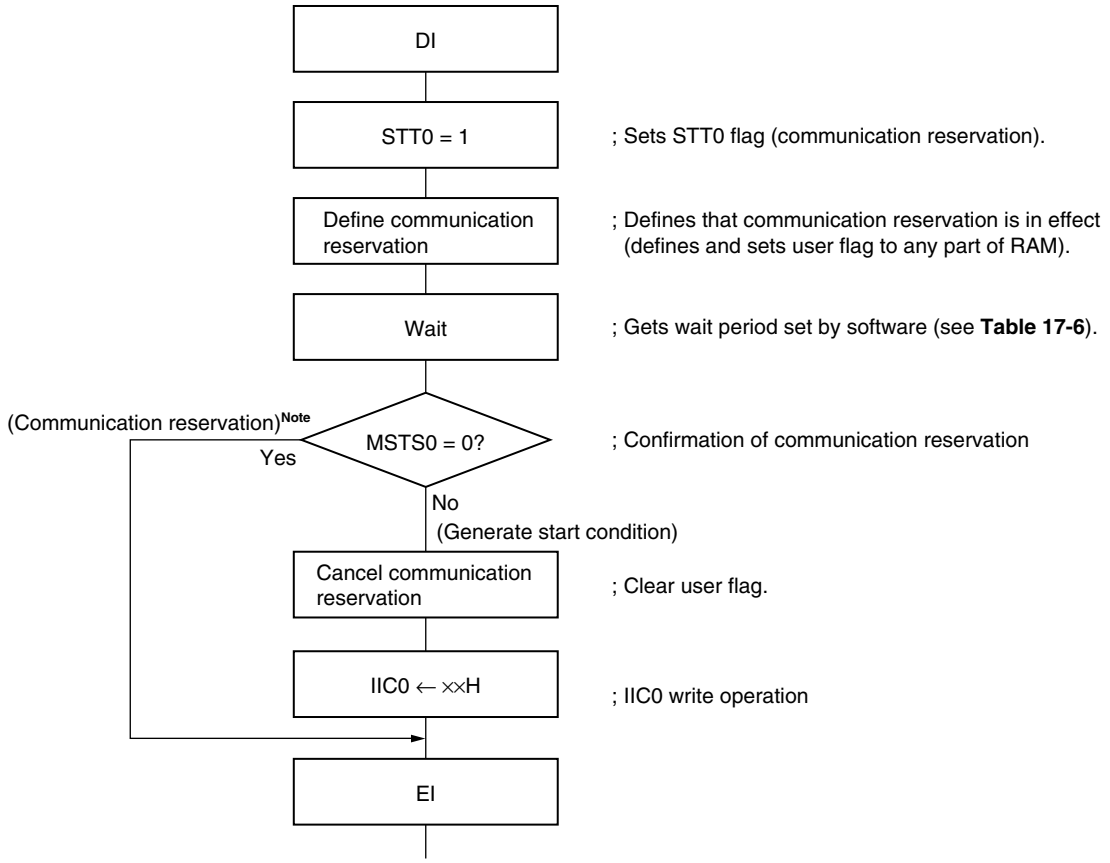
Communication reservations are accepted via the following timing. After the IICS0.STD0 bit is set to 1, a communication reservation can be made by setting the IICC0.STT0 bit to 1 before a stop condition is detected.

Figure 17-14. Timing for Accepting Communication Reservations



The communication reservation flowchart is illustrated below.

Figure 17-15. Communication Reservation Flowchart



Note The communication reservation operation executes a write to the IIC0 register when a stop condition interrupt request occurs.

17.14.2 When communication reservation function is disabled (IICF0.IICRSV0 bit = 1)

When the IICC0.STT0 bit is set when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released when the IICC0.LREL0 bit was set to 1)

To confirm whether the start condition was generated or request was rejected, check the IICF0.STCF0 flag. The time shown in Table 17-7 is required until the STCF0 flag is set after setting the STT0 bit = 1. Therefore, secure the time by software.

Table 17-7. Wait Periods

Selection Clock	CLX0	SMC0	CL00	Wait Clock	Wait Time When f_{XX} = 64 MHz
f_{XX}/8 (IICOCKS = 12H)	0	0	0	5 clocks	0.63 μs
f_{XX}/10 (IICOCKS = 13H)	0	0	0	5 clocks	0.78 μs
f_{XX}/4 (IICOCKS = 10H)	0	0	1	5 clocks	0.31 μs
f_{XX}/6 (IICOCKS = 11H)	0	0	1	5 clocks	0.47 μs
f_{XX}/8 (IICOCKS = 12H)	0	0	1	5 clocks	0.63 μs
f_{XX}/10 (IICOCKS = 13H)	0	0	1	5 clocks	0.78 μs

17.15 Cautions

(1) When IICF0.STCEN0 bit = 0

Immediately after I²C operation is enabled, the bus communication status (IICF0.IICBSY0 bit = 1) is recognized regardless of the actual bus status. To execute master communication in the status where a stop condition has not been detected, generate a stop condition and then release the bus before starting the master communication.

Use the following sequence for generating a stop condition.

<1> Set the IICCL0 register.
<2> Set the IICC0.IICE0 bit.
<3> Set the IICC0.SPT0 bit.

(2) When IICF0.STCEN0 bit = 1

Immediately after I²C operation is enabled, the bus released status (IICBSY0 bit = 0) is recognized regardless of the actual bus status. To generate the first start condition (IICC0.STT0 bit = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) When the IICC0.IICE0 bit of the V850E/IF3 and V850E/IG3 is set to 1 while communications with other devices are in progress, the start condition may be detected depending on the status of the communication line. Be sure to set the IICC0.IICE0 bit to 1 when the SCL and SDA lines are high level.

(4) Determine the operation clock frequency by the IICCL0, IICX0, and IICOCKS registers before enabling the operation (IICC0.IICE0 bit = 1). To change the operation clock frequency, clear the IICC0.IICE0 bit to 0 once.

(5) After the IICC0.STT0 and IICC0.SPT0 bits have been set to 1, they must not be re-set without being cleared to 0 first.

(6) If transmission has been reserved, set the IICC0.SPIE0 bit to 1 so that an interrupt request is generated by the detection of a stop condition. After an interrupt request has been generated, the wait state will be released by writing communication data to I²C, then transferring will begin. If an interrupt is not generated by the detection of a stop condition, transmission will halt in the wait state because an interrupt request was not generated. However, it is not necessary to set the SPIE0 bit to 1 for the software to detect the IICS0.MSTS0 bit.

17.16 Communication Operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the V850E/IF3 and V850E/IG3 as the master in a single master system is shown below.

This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

(2) Master operation in multimaster system

In the I²C bus multimaster system, whether the bus is released or used cannot be judged by the I²C bus specifications when the bus takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the V850E/IF3 and V850E/IG3 take part in a communication with bus released state.

This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the V850E/IF3 and V850E/IG3 lose in arbitration and are specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

(3) Slave operation

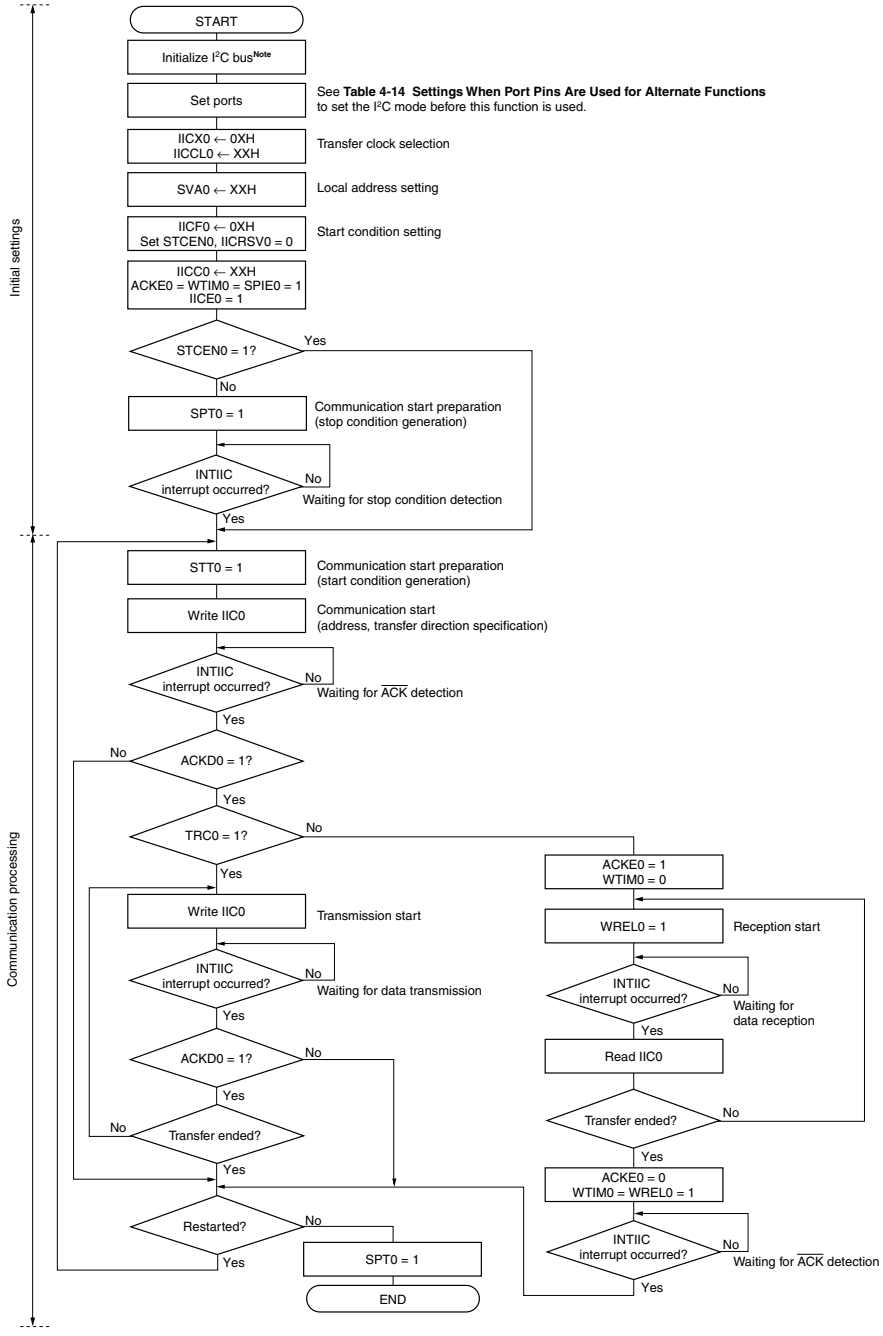
An example of when the V850E/IF3 and V850E/IG3 are used as the slave is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIIC interrupt occurrence (communication waiting). When the INTIIC interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

17.16.1 Master operation in single master system

Figure 17-16. Master Operation in Single Master System

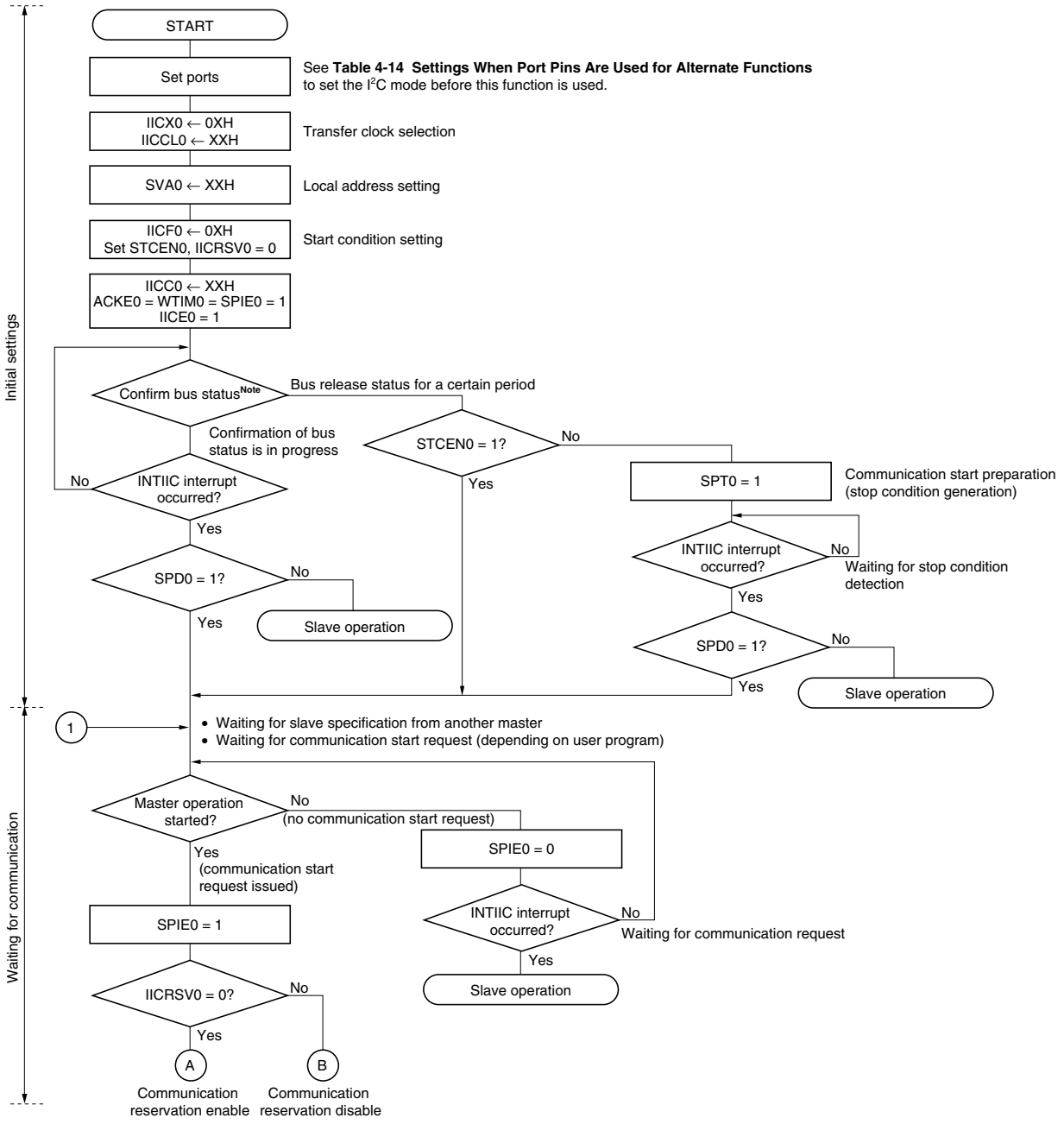


Note Release the I²C bus (SCL, SDA pins = high level) in conformity with the specifications of the product in communication. For example, when the EEPROM™ outputs a low level to the SDA pin, set the SCL pin to the output port and output clock pulses from that output port until when the SDA pin is constantly high level.

Remark For the transmission and reception formats, conform to the specifications of the product in communication.

17.16.2 Master operation in multimaster system

Figure 17-17. Master Operation in Multimaster System (1/3)



Note Confirm that the bus release status (IICCL0.CLD0 bit = 1, IICCL0.DAD0 bit = 1) has been maintained for a certain period (1 frame, for example). When the SDA pin is constantly low level, determine whether to release the I²C bus (SCL, SDA pins = high level) by referring to the specifications of the product in communication.

Figure 17-17. Master Operation in Multimaster System (2/3)

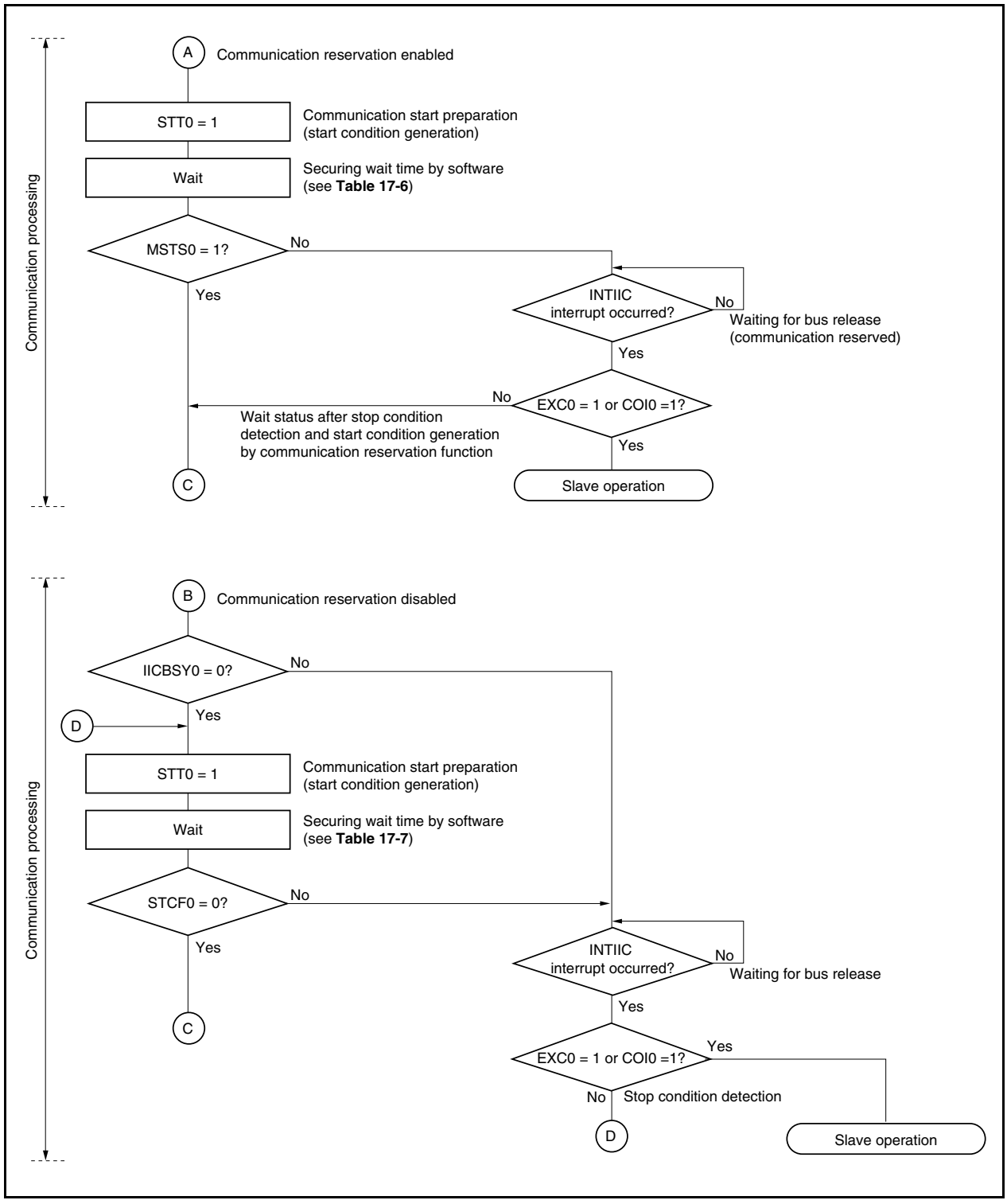
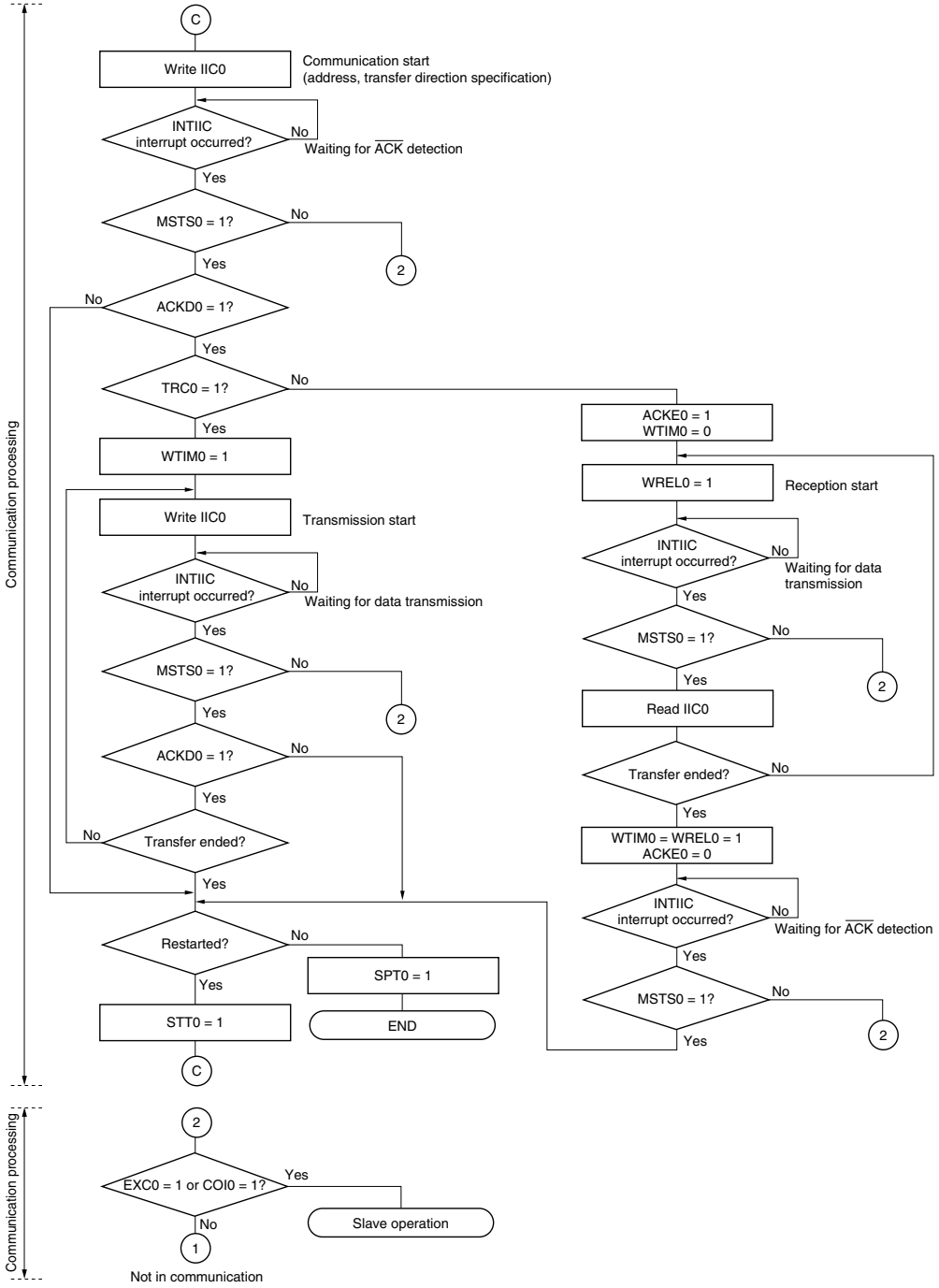


Figure 17-17. Master Operation in Multimaster System (3/3)



Remarks 1. Conform the transmission and reception formats to the specifications of the product in communication.

2. When using the V850E/IF3 and V850E/IG3 as the master in the multimaster system, read the IICS0.MSTS0 bit for each INTIIC interrupt occurrence to confirm the arbitration result.

3. When using the V850E/IF3 and V850E/IG3 as the slave in the multimaster system, confirm the status using the IICS0 and IICF0 registers for each INTIIC interrupt occurrence to determine the next processing.

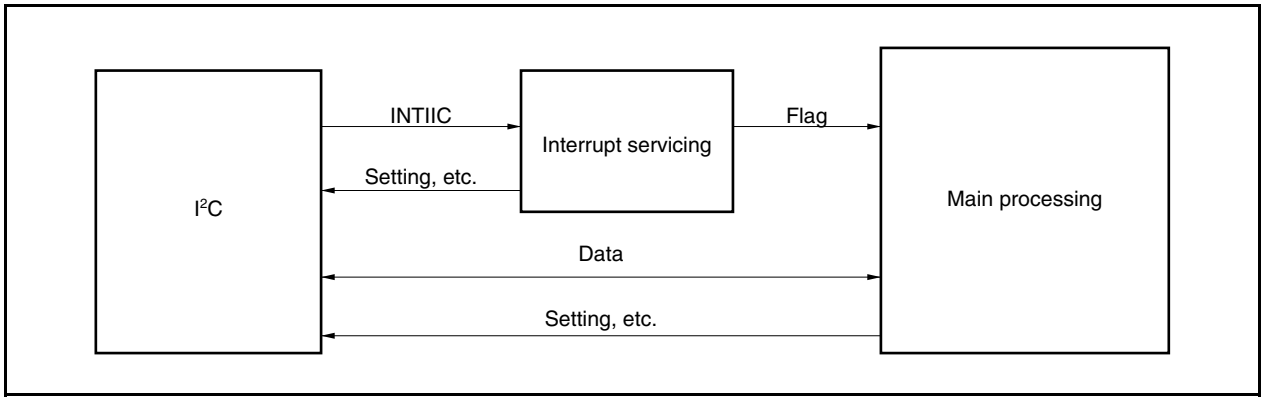
17.16.3 Slave operation

The following shows the processing procedure of the slave operation.

Basically, the operation of the slave device is event-driven. Therefore, processing by an INTIIC interrupt (processing requiring a significant change of the operation status, such as stop condition detection during communication) is necessary.

The following description assumes that data communication does not support extension codes. Also, it is assumed that the INTIIC interrupt servicing performs only status change processing and that the actual data communication is performed during the main processing.

Figure 17-18. Software Outline During Slave Operation



Therefore, the following three flags are prepared so that the data transfer processing can be performed by transmitting these flags to the main processing instead of the INTIIC signal.

(1) Communication mode flag

This flag indicates the following communication statuses.

Clear mode: Data communication not in progress

Communication mode: Data communication in progress (valid address detection stop condition detection, $\overline{\text{ACK}}$ from master not detected, address mismatch)

(2) Ready flag

This flag indicates that data communication is enabled. This is the same status as an INTIIC interrupt during normal data transfer. This flag is set in the interrupt processing block and cleared in the main processing block. The ready flag for the first data for transmission is not set in the interrupt processing block, so the first data is transmitted without clearance processing (the address match is regarded as a request for the next data).

(3) Communication direction flag

This flag indicates the direction of communication and is the same as the value of the IICS0.TRC0 bit.

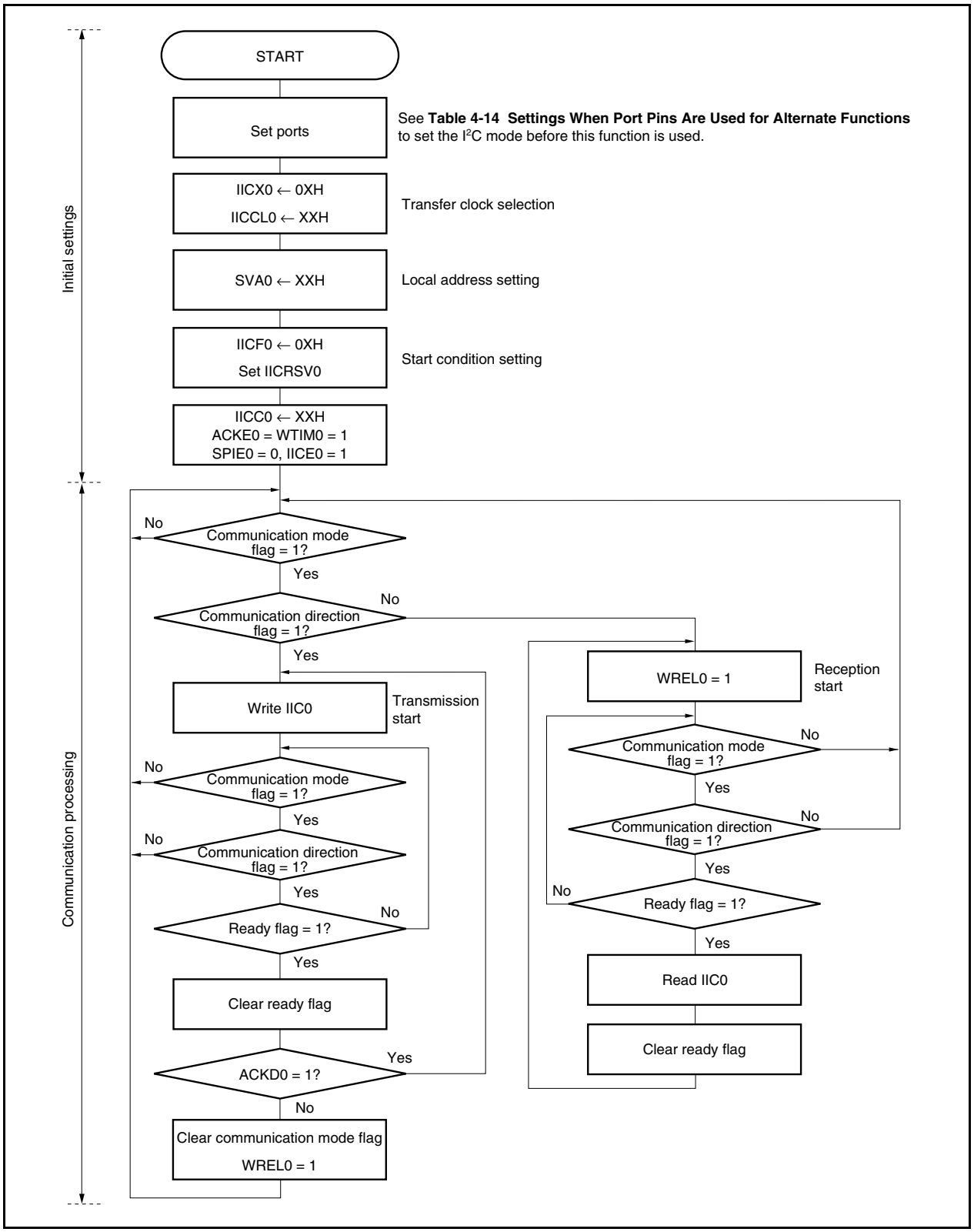
The following shows the operation of the main processing block during slave operation.

Start I²C and wait for the communication enabled status. When communication is enabled, perform transfer using the communication mode flag and ready flag (the processing of the stop condition and start condition is performed by interrupts, conditions are confirmed by flags).

For transmission, repeat the transmission operation until the master device stops returning $\overline{\text{ACK}}$. When the master device stops returning $\overline{\text{ACK}}$, transfer is end.

For reception, receive the required number of data and do not return $\overline{ACK}$ for the next data immediately after transfer is end. After that, the master device generates the stop condition or restart condition. This causes exit from communications.

Figure 17-19. Slave Operation Flowchart (1)

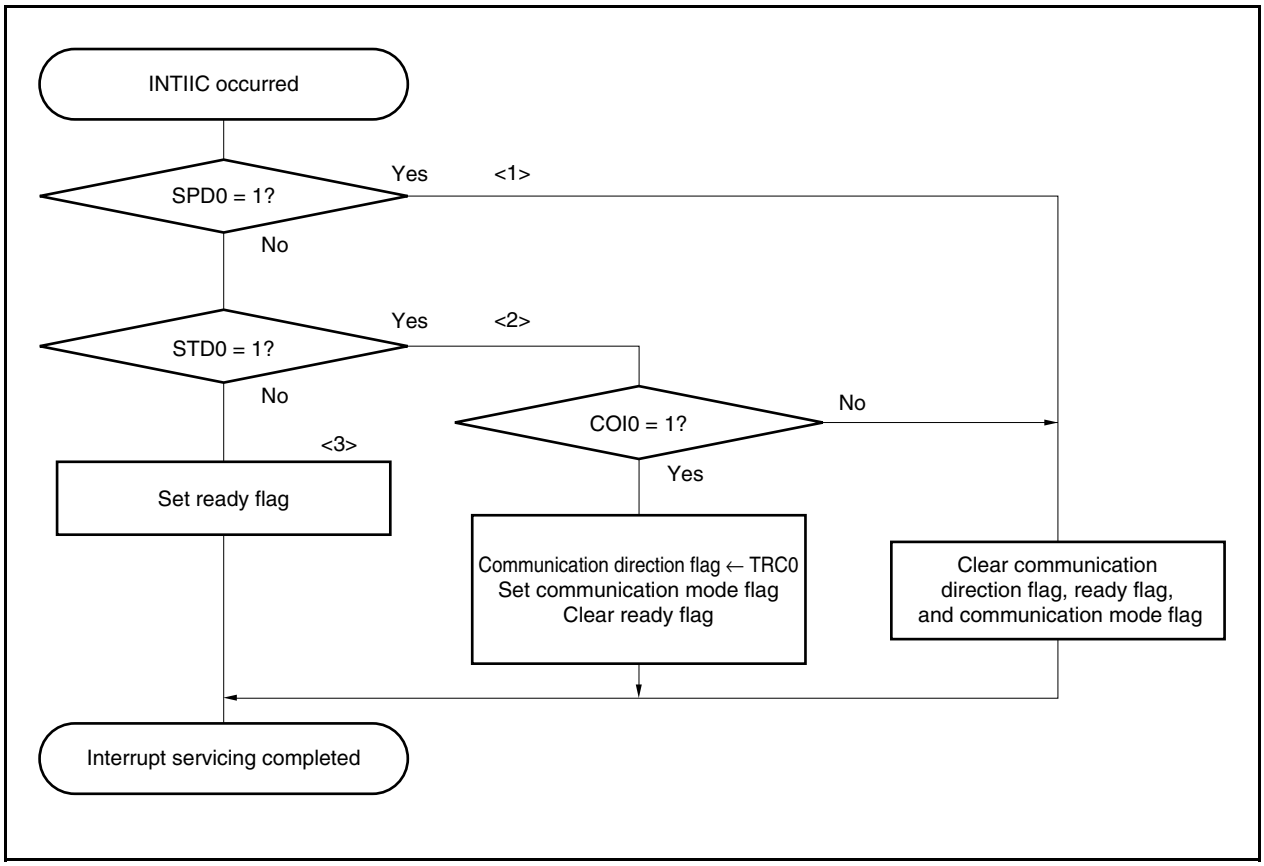


The following shows an example of the processing of the slave device by an INTIIC interrupt (it is assumed that no extension codes are used here). During an INTIIC interrupt, the status is confirmed and the following steps are executed.

<1> When a stop condition is detected, communication is terminated.
<2> When a start condition is detected, the address is confirmed. If the address does not match, communication is terminated. If the address matches, the communication mode is set and wait is released, and operation returns from the interrupt (the ready flag is cleared).
<3> For data transmission/reception, when the ready flag is set, operation returns from the interrupt while the I²C bus remains in the wait status.

Remark <1> to <3> in the above correspond to <1> to <3> in **Figure 17-20 Slave Operation Flowchart (2)**.

Figure 17-20. Slave Operation Flowchart (2)



17.17 Timing of Data Communication

When using I²C bus mode, the master device generates an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the IICS0.TRC0 bit that specifies the data transfer direction and then starts serial communication with the slave device.

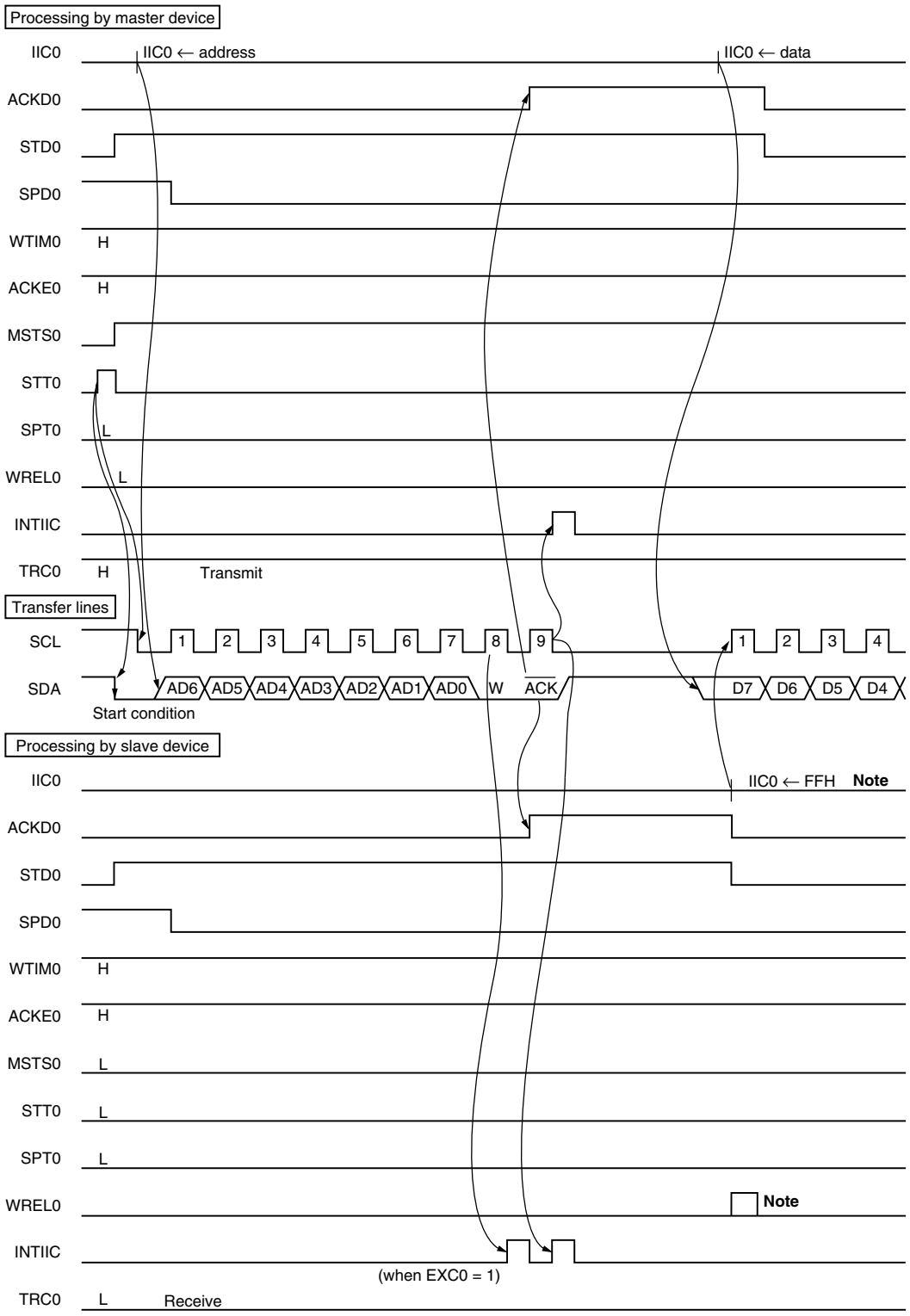
The IIC0 register's shift operation is synchronized with the falling edge of the serial clock (SCL pin). The transmit data is transferred to the SO latch and is output (MSB first) via the SDA pin.

Data input via the SDA pin is captured by the IIC0 register at the rising edge of the SCL pin.

The data communication timing is shown below.

Figure 17-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)

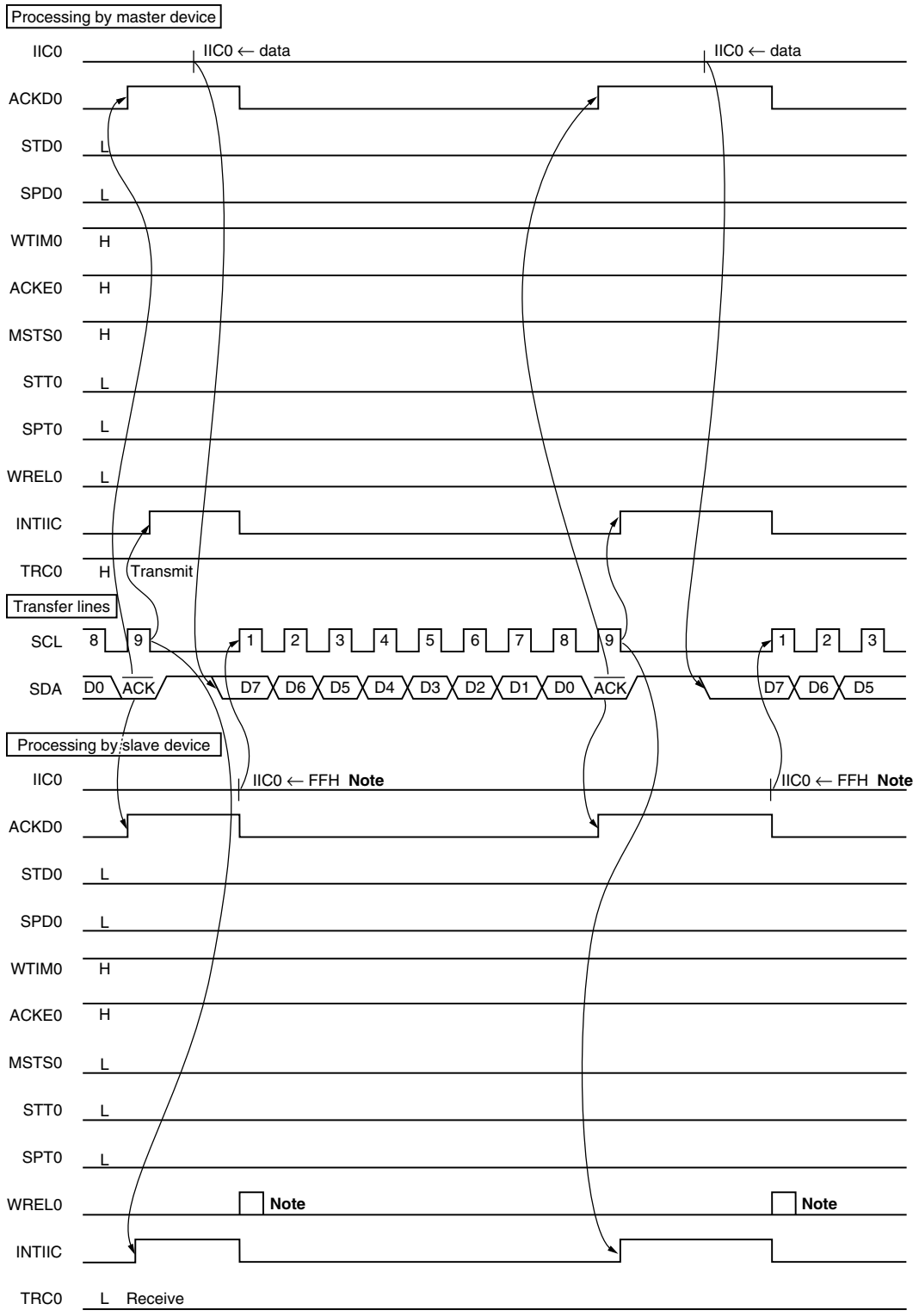
(a) Start condition ~ address



Note To cancel slave wait, write FFH to IIC0 or set WREL0.

Figure 17-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

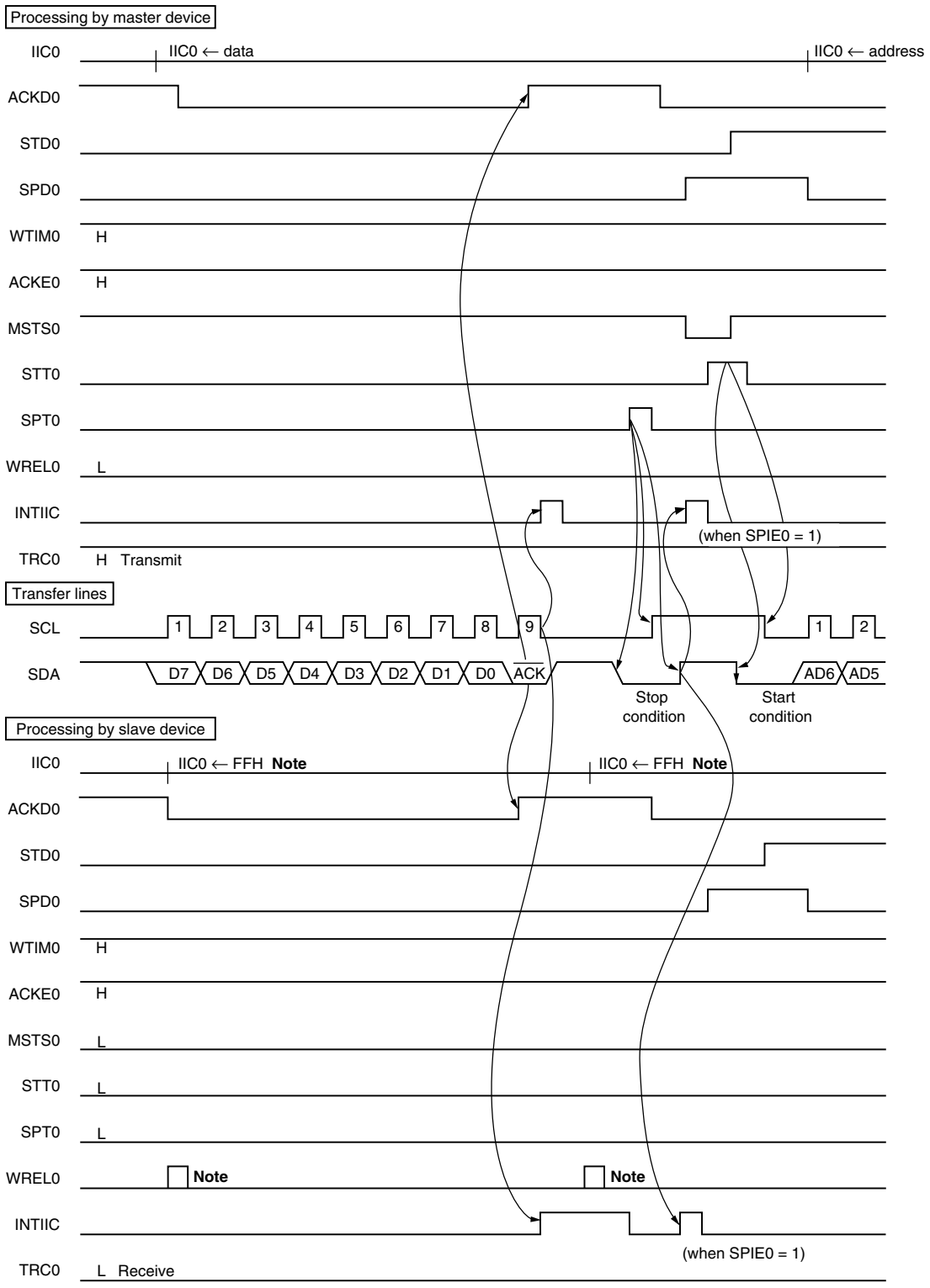
(b) Data



Note To cancel slave wait, write FFH to IIC0 or set WREL0.

Figure 17-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

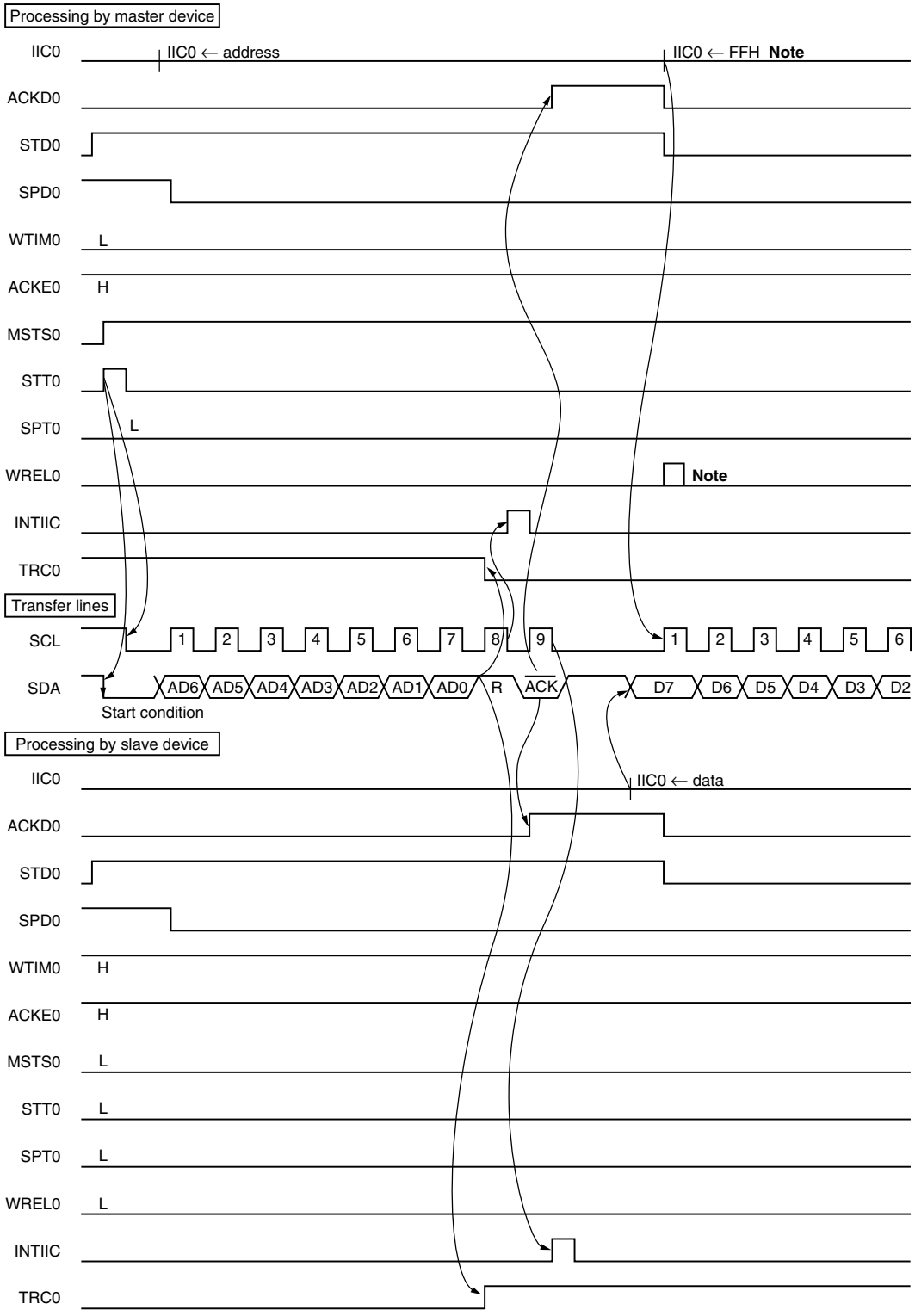
(c) Stop condition



Note To cancel slave wait, write FFH to IIC0 or set WREL0.

Figure 17-22. Example of Slave to Master Communication (When 8-Clock Wait for Master and 9-Clock Wait for Slave Are Selected) (1/3)

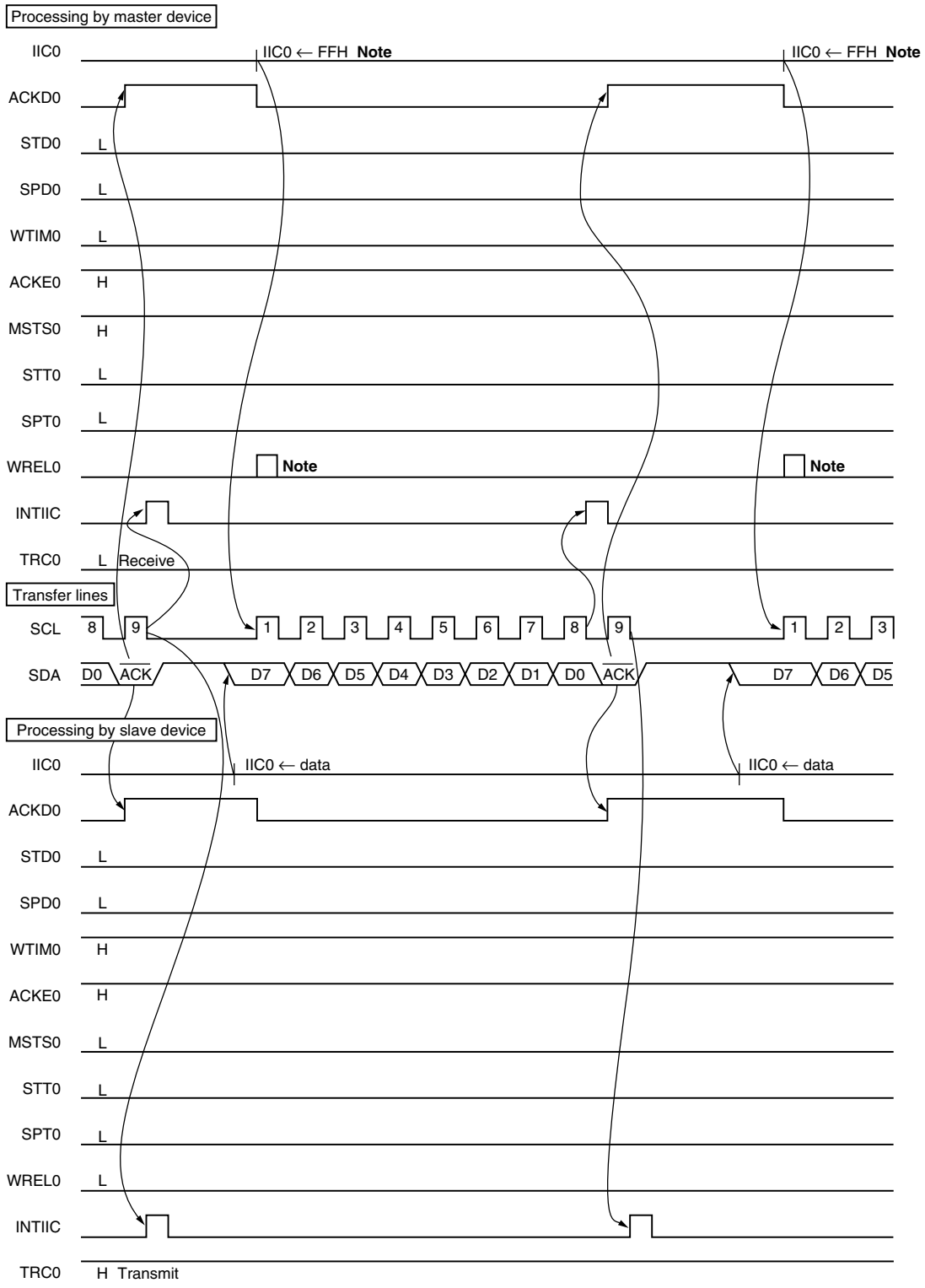
(a) Start condition ~ address



Note To cancel master wait, write FFH to IIC0 or set WREL0.

Figure 17-22. Example of Slave to Master Communication
(When 8-Clock Wait for Master and 9-Clock Wait for Slave Are Selected) (2/3)

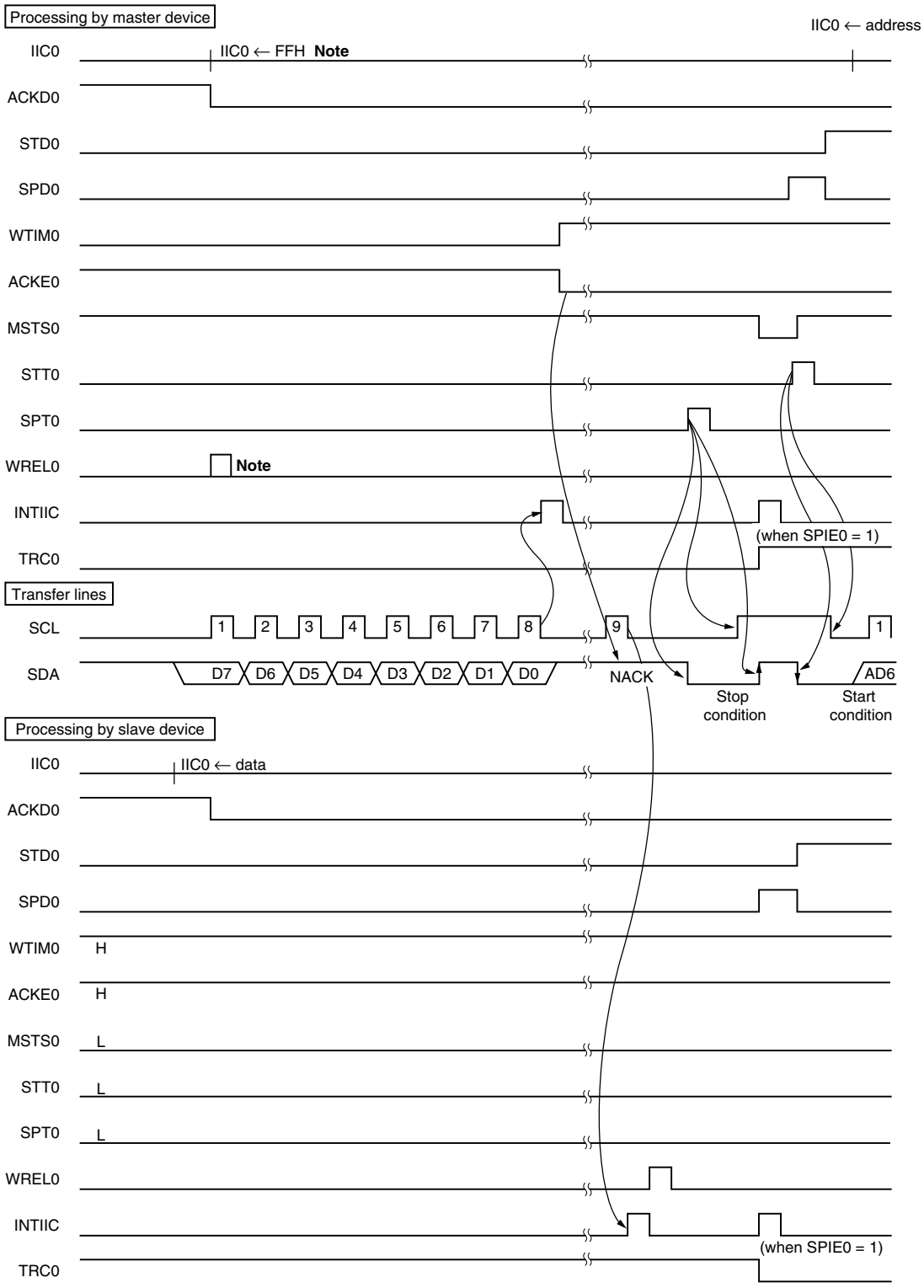
(b) Data



Note To cancel master wait, write FFH to IIC0 or set WREL0.

Figure 17-22. Example of Slave to Master Communication
(When 8-Clock Wait for Master and 9-Clock Wait for Slave Are Selected) (3/3)

(c) Stop condition



Note To cancel master wait, write FFH to IIC0 or set WREL0.

CHAPTER 18 BUS CONTROL FUNCTION

The μPD70F3454GC-8EA-A is provided with an external bus interface function via which external memories such as ROM and RAM, and I/O can be connected. The μPD70F3453GC-8EA-A, GF version of the V850E/IG3, and the V850E/IF3 are not provided with an external bus interface function.

This chapter describes the μPD70F3454GC-8EA-A as a target microcontroller.

18.1 Features

- 16-bit/8-bit data bus sizing function
- 2-space chip select function
- Wait function
  - Programmable wait function, through which up to 7 wait states can be inserted for each memory block
  - Address setup wait and address hold wait insertion functions, through which 1 wait state can be inserted for each memory block
  - External wait function via $\overline{\text{WAIT}}$ pin
- Idle state insertion function
- External device connection enabled via bus control/port alternate function pins
- Separate bus mode (8-bit address bus, 8-bit/16-bit data bus)
- Multiplexed bus mode (16-bit address bus, 8-bit/16-bit data bus)
- Support for little endian
- External bus clock frequency: 32 MHz/16 MHz selectable function

18.2 Bus Control Pins

The following pins are used for connection to external devices.

(1) In separate bus mode

Bus Control Pin (Function in Alternate Function Mode)	Function in Port Mode	Register Switching Port Mode/Alternate Function
Data bus (AD0 to AD15)	PDL0 to PDL9, PDL10 to PDL15	PMCDL
Address bus (A0 to A7)	P10 to P17	PMC1
Chip select ($\overline{CS0}$, $\overline{CS1}$)	P34, P32	PMC3
Read/write control (ASTB, $\overline{WR0}$, $\overline{WR1}$, $\overline{RD}$)	P37, P46, P45, P47	PMC3, PMC4
External wait control ($\overline{WAIT}$)	P44	PMC4
External bus clock output (CLKOUT)	P07	PMC0

(2) In multiplexed bus mode

Bus Control Pin (Function in Alternate Function Mode)	Function in Port Mode	Register Switching Port Mode/Alternate Function
Address data bus (AD0 to AD15)	PDL0 to PDL9, PDL10 to PDL15	PMCDL
Chip select ($\overline{CS0}$, $\overline{CS1}$)	P34, P32	PMC3
Read/write control (ASTB, $\overline{WR0}$, $\overline{WR1}$, $\overline{RD}$)	P37, P46, P45, P47	PMC3, PMC4
External wait control ($\overline{WAIT}$)	P44	PMC4
External bus clock output (CLKOUT)	P07	PMC0

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

The status of each pin is as follows when the internal ROM, internal RAM, and on-chip peripheral I/O is accessed.

Table 18-1. Pin Status List In Internal ROM, Internal RAM, and On-Chip Peripheral I/O Access

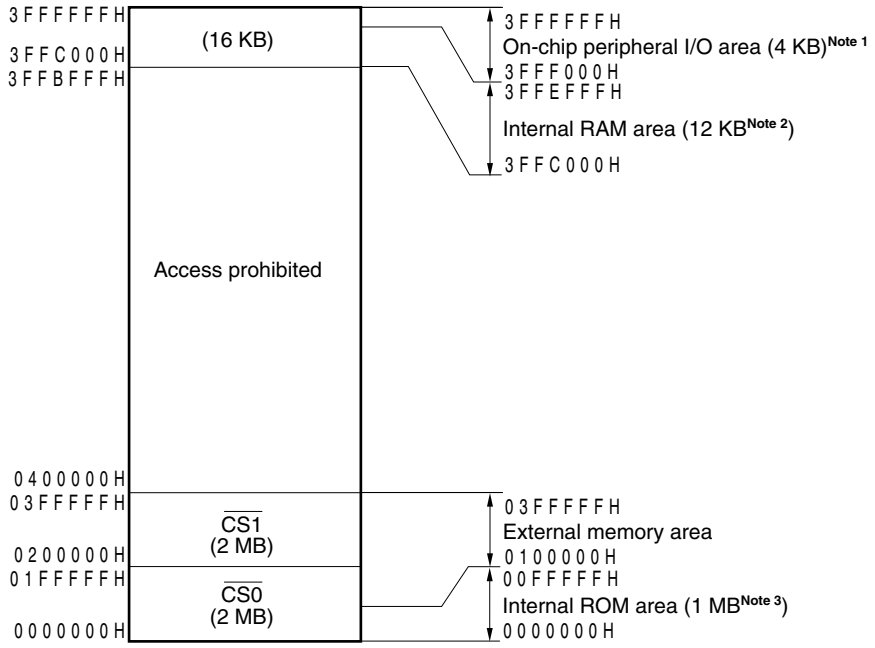
Access Destination	Internal ROM	Internal RAM	On-Chip Peripheral I/O
Address bus	Undefined	Undefined	**Note 1**
Data bus	Hi-Z	Hi-Z	Hi-Z
External bus control signal	Inactive[Note 2]	Inactive[Note 2]	Inactive[Note 2]

Notes 1. While the on-chip peripheral I/O is accessed, the address the on-chip peripheral I/O accesses is also output to the external address bus.

2. The $\overline{WAIT}$ pin does not input any signal during this operation.

18.3 Memory Block Function

The 64 MB memory space is divided into memory blocks of lower 2 MB units. The programmable wait function and bus cycle operation mode can be independently controlled for each block.



Notes 1. Addresses 3FFF000H to 3FFFFFFH are access-prohibited. To access the on-chip peripheral I/O, specify addresses FFFF000H to FFFFFFFH.

2. μPD70F3453: 8 KB
μPD70F3454: 12 KB

3. μPD70F3453: 128 KB
μPD70F3454: 256 KB

18.3.1 Chip select control function

Of the 64 MB address space (linear), the lower 4 MB (0000000H to 03FFFFFH) has two chip select functions, $\overline{CS0}$ and $\overline{CS1}$. The areas selected by $\overline{CS0}$ and $\overline{CS1}$ are fixed.

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

Chip Select Pin	Area
$\overline{CS0}$	0000000H to 01FFFFFH (2 MB)
$\overline{CS1}$	0200000H to 03FFFFFH (2 MB)

18.4 Bus Cycle Type Control Function

In the μPD70F3454GC-8EA-A, SRAM, external ROM, and external I/O can be connected directly.

(1) Bus cycle type configuration register 0 (BCT0)

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

Reset sets this register to CCCCH.

Cautions 1. Do not access an external memory area until the initial setting of the BCT0 register is complete. However, it is possible to access external memory areas whose initialization settings are complete.

2. The set contents of each register are invalid for the CSn space where operations are prohibited.

After reset: CCCCH R/W Address: FFFFF480H

	15	14	13	12	11	10	9	8
BCT0	1	1	0	0	1	1	0	0

$\overline{CSn}$ signal

	7	6	5	4	3	2	1	0
	ME1	1	0	0	ME0	1	0	0
$\overline{CSn}$ signal	$\overline{CS1}$				$\overline{CS0}$			

MEn	Memory controller operation enable for each CSn space (n = 0, 1)
0	Operation disabled
1	Operation enabled

Caution Be sure to set bits 0, 1, 4, 5, 8, 9, 12, and 13 to "0", and set bits 2, 6, 10, 11, 14, and 15 to "1". If they are set other than above, the operation is not guaranteed.

18.5 Bus Access

18.5.1 Number of access clocks

The number of base clocks (MIN. value) necessary for accessing each resource is as follows.

Resource (Bus Width) \ Bus Cycle Configuration		Instruction Fetch (Normal Access)	Instruction Fetch (Branch)	Operand Data Access
Internal ROM (32 bits)		1	2	5
Internal RAM (32 bits)		1 Note 1	1 Note 1	1
On-chip peripheral I/O (16 bits)		–	–	3 Note 2
External memory (16 bits)	Separate bus mode	3 + n	3 + n	3 + n
	Multiplexed bus mode	3 + n	3 + n	3 + n

Notes 1. This value is 2 if there is conflict with data access.

2. Depending on the set value of the VSWC register.

Remarks 1. Unit: Clock/access

2. n: Number of wait state inserted

18.5.2 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 BSC register.

(1) Bus size configuration register (BSC)

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

Reset sets this register to 5555H.

Caution Write to the BSC register after reset, and then do not change the set value. Also, do not access an external memory area until the initial setting of the BSC register is complete. However, it is possible to access external memory areas whose initialization settings are complete.

After reset: 5555H R/W Address: FFFFF066H

	15	14	13	12	11	10	9	8
BSC	0	1	0	1	0	1	0	1
$\overline{CSn}$ signal								

	7	6	5	4	3	2	1	0
	0	1	0	1	0	BS10	0	BS00
$\overline{CSn}$ signal						$\overline{CS1}$		$\overline{CS0}$

BSn0	Specification of data bus width of each CSn space (n = 0, 1)
0	8 bits
1	16 bits

Caution Be sure to set bits 1, 3, 5, 7, 9, 11, 13, and 15 to "0", and set bits 4, 6, 8, 10, 12, and 14 to "1". If they are set other than above, the operation is not guaranteed.

18.5.3 Endian function

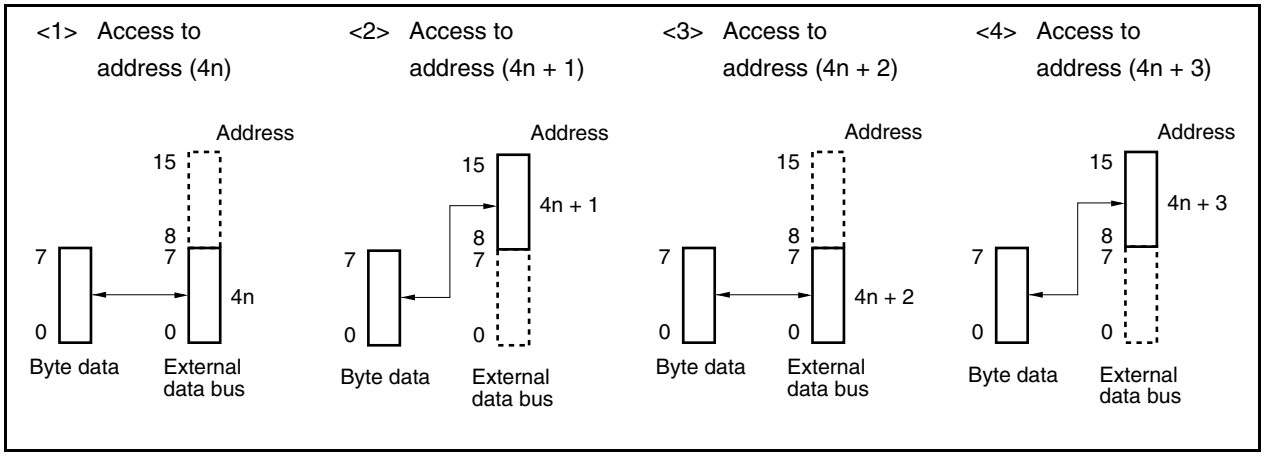
The μPD70F3454GC-8EA-A corresponds to little endian.

18.5.4 Bus width

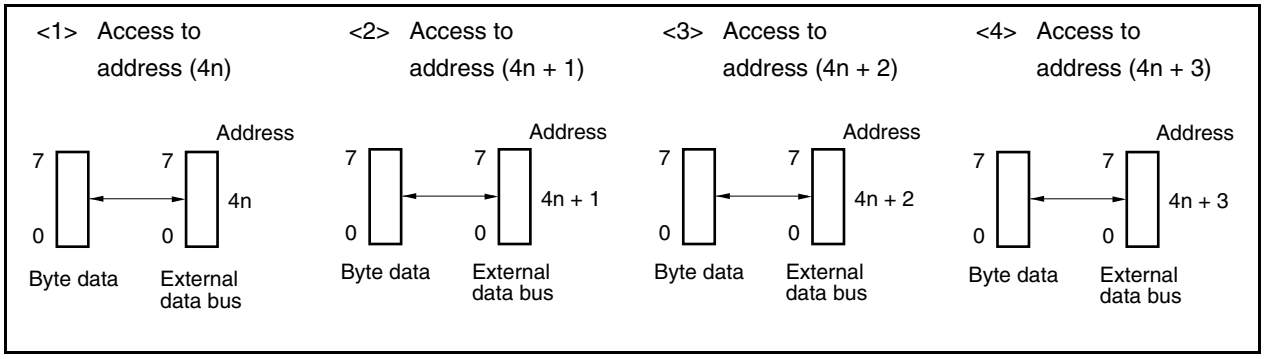
The μPD70F3454GC-8EA-A is accesses on-chip peripheral I/O and external memory in 8-bit, 16-bit, or 32-bit units. The following shows the operation for each type of access. All data is accessed in order starting from the lower order side.

(1) Byte access (8 bits)

(a) When the data bus width is 16 bits

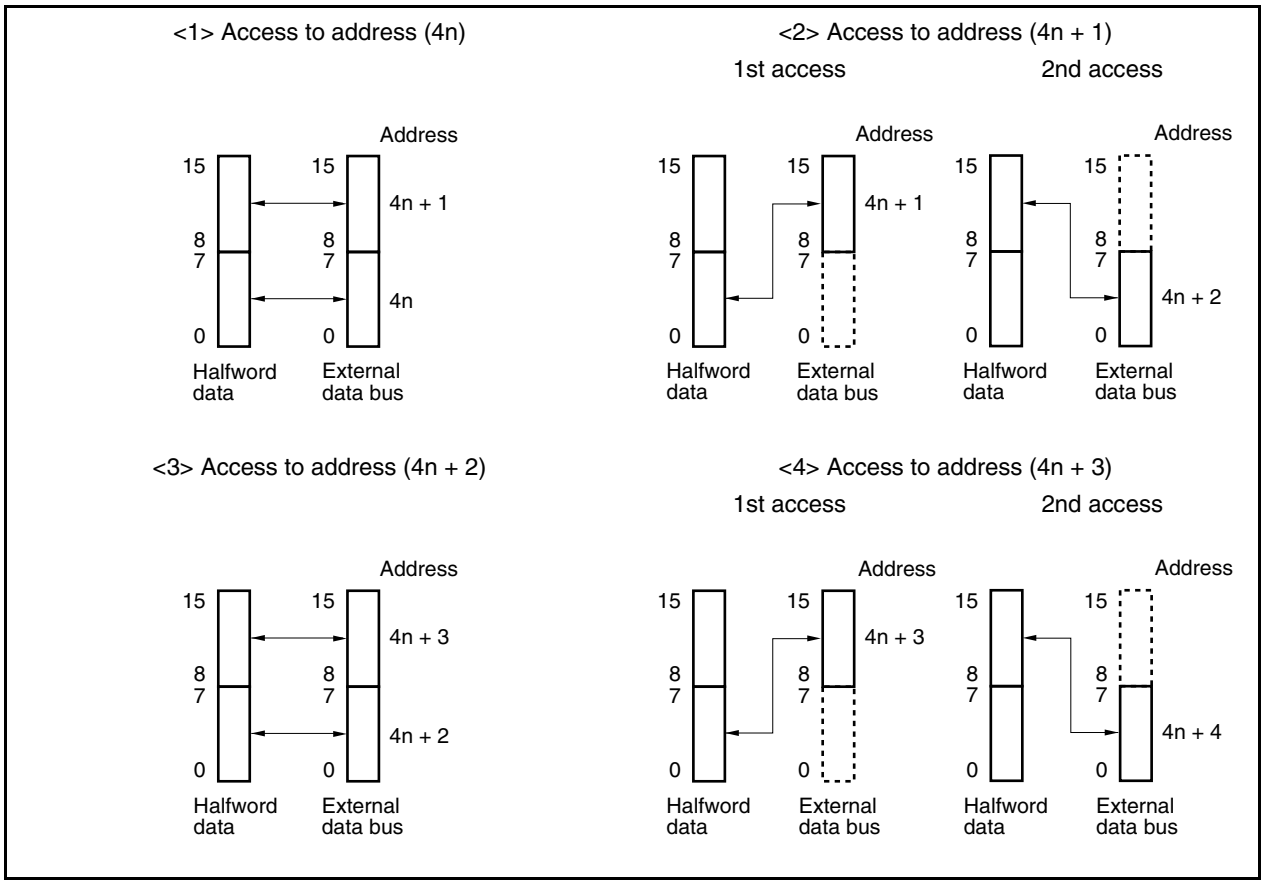


(b) When the data bus width is 8 bits

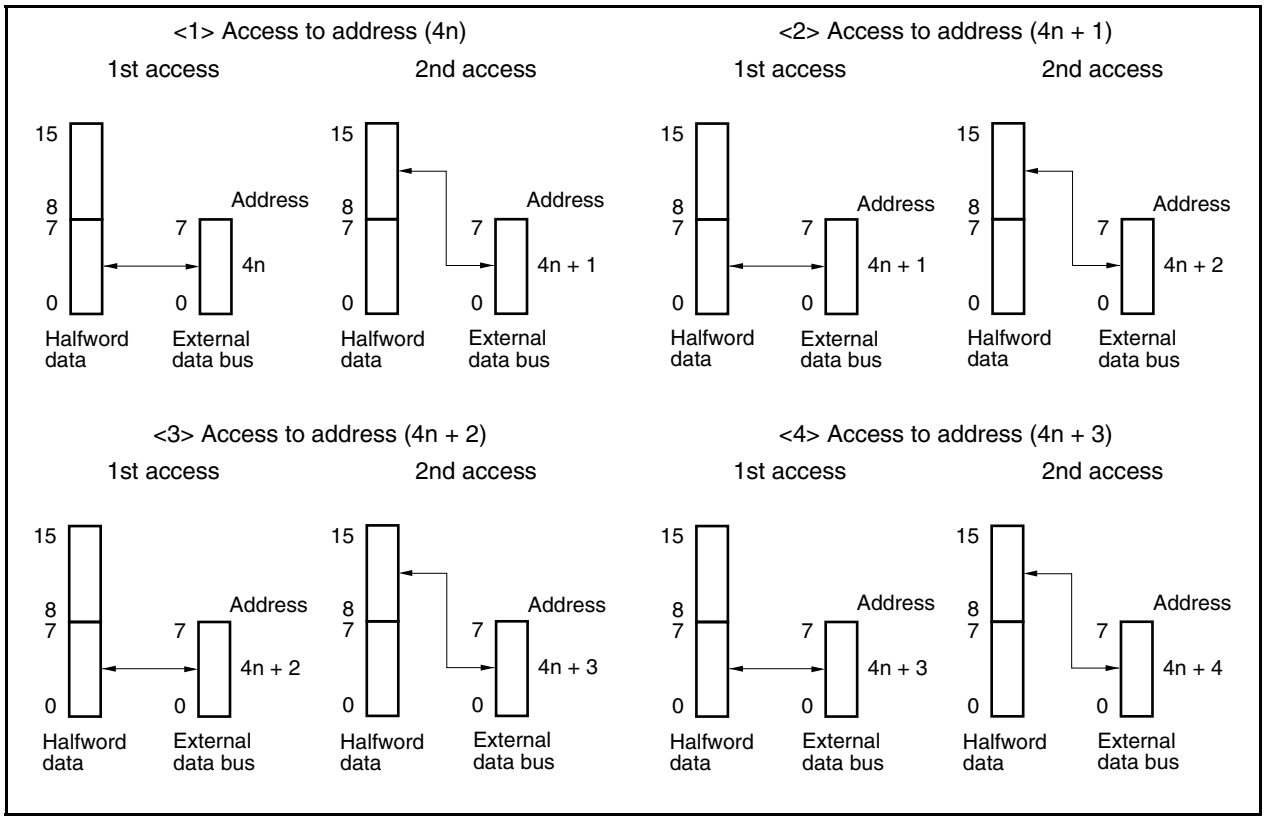


(2) Halfword access (16 bits)

(a) When the data bus width is 16 bits



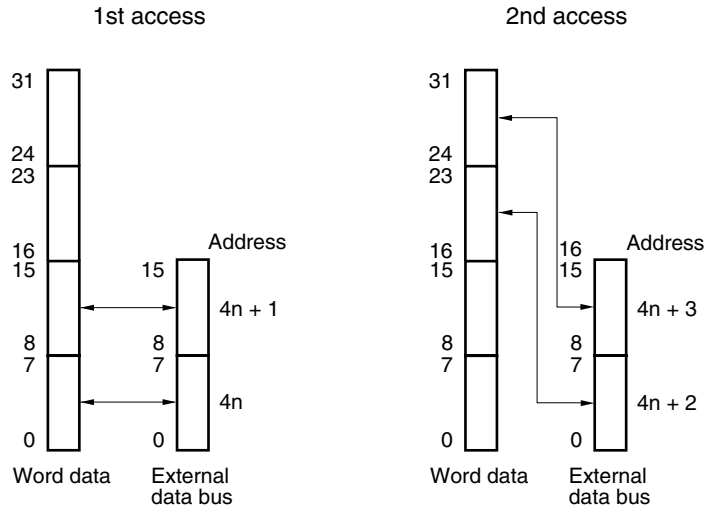
(b) When the data bus width is 8 bits



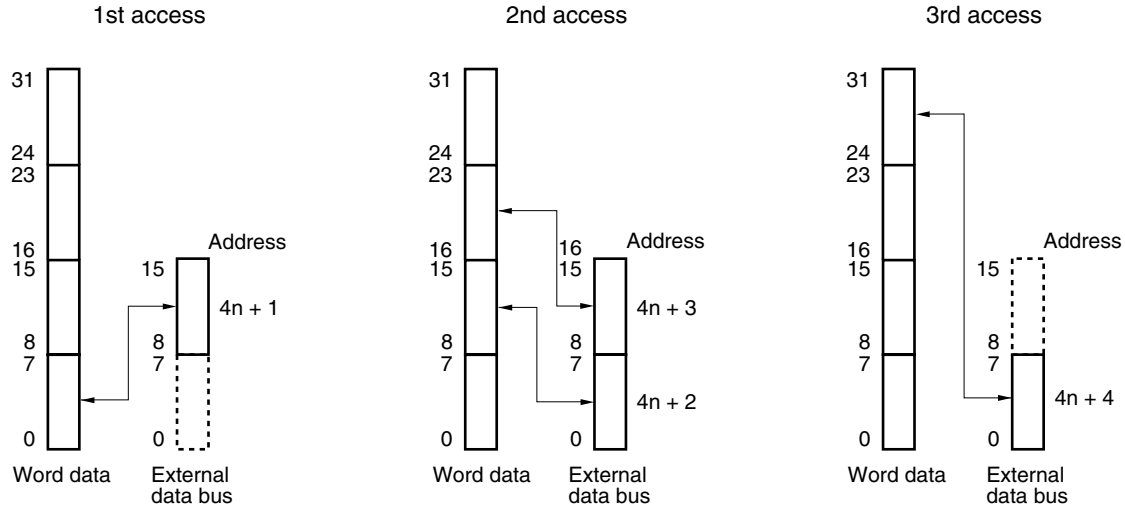
(3) Word access (32 bits)

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

<1> Access to address (4n)

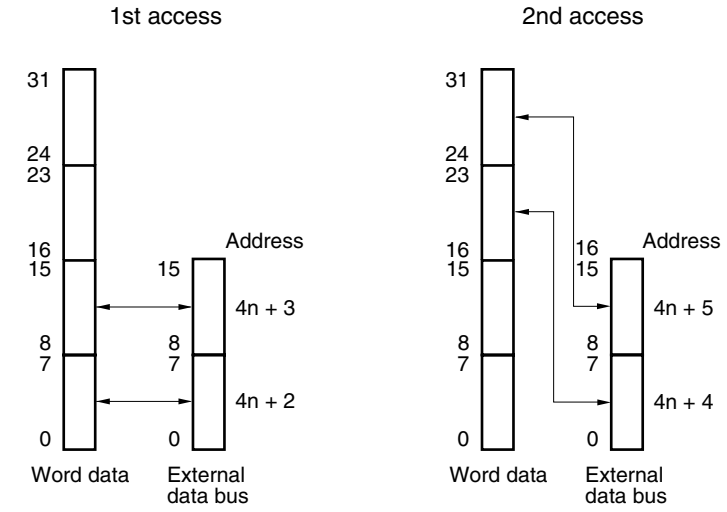


<2> Access to address (4n + 1)

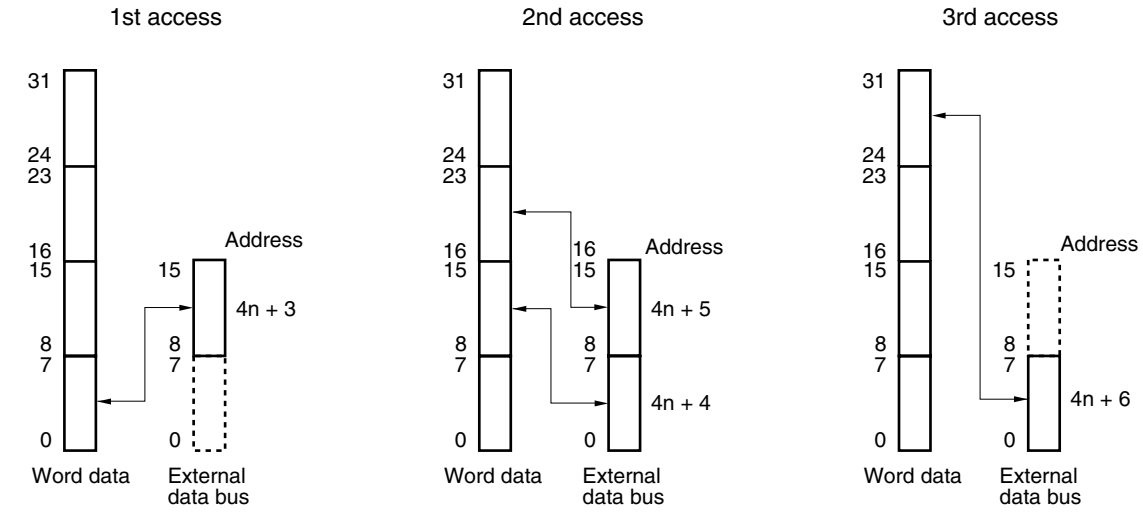


(a) When the data bus width is 16 bits (2/2)

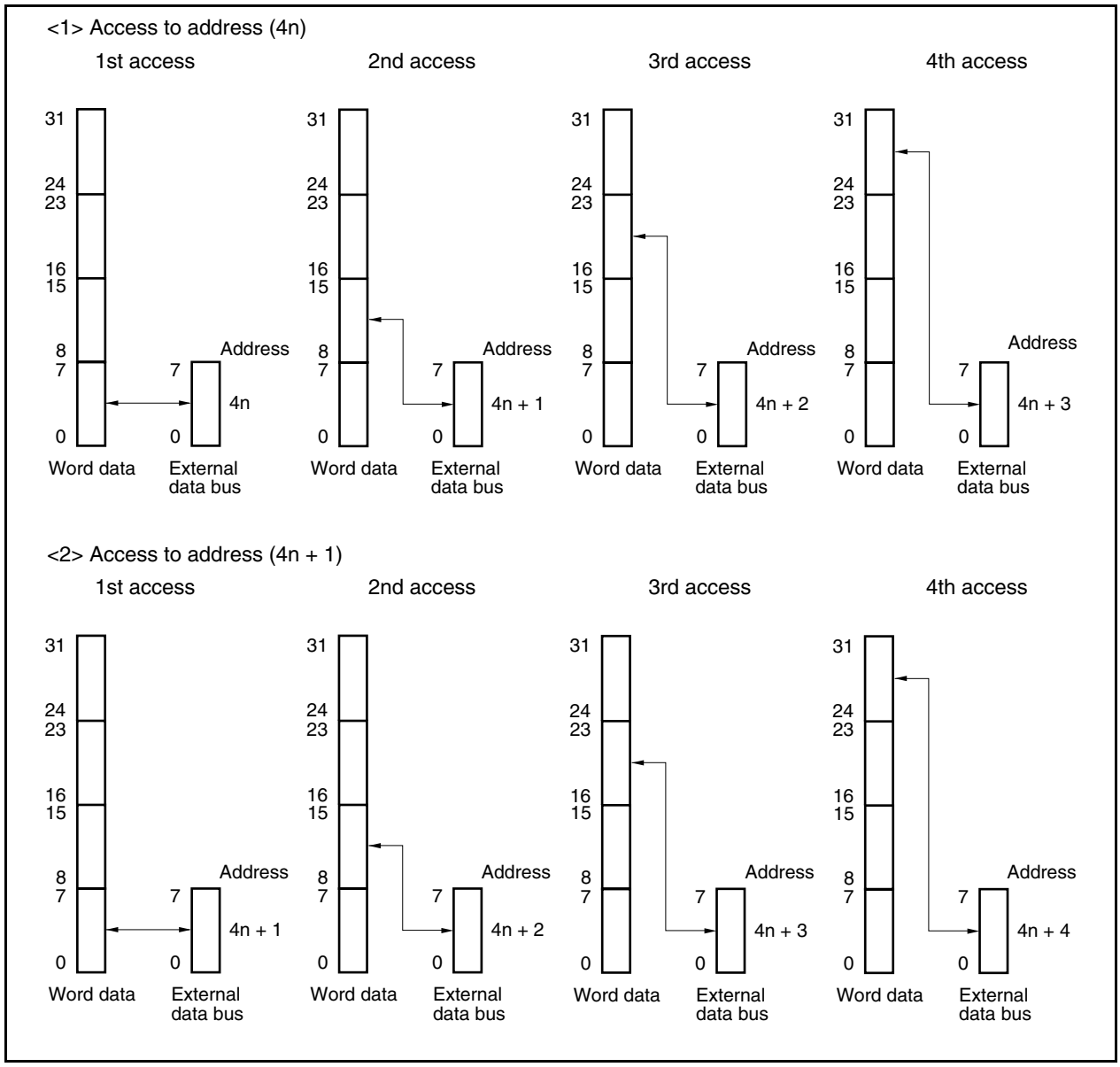
<3> Access to address (4n + 2)



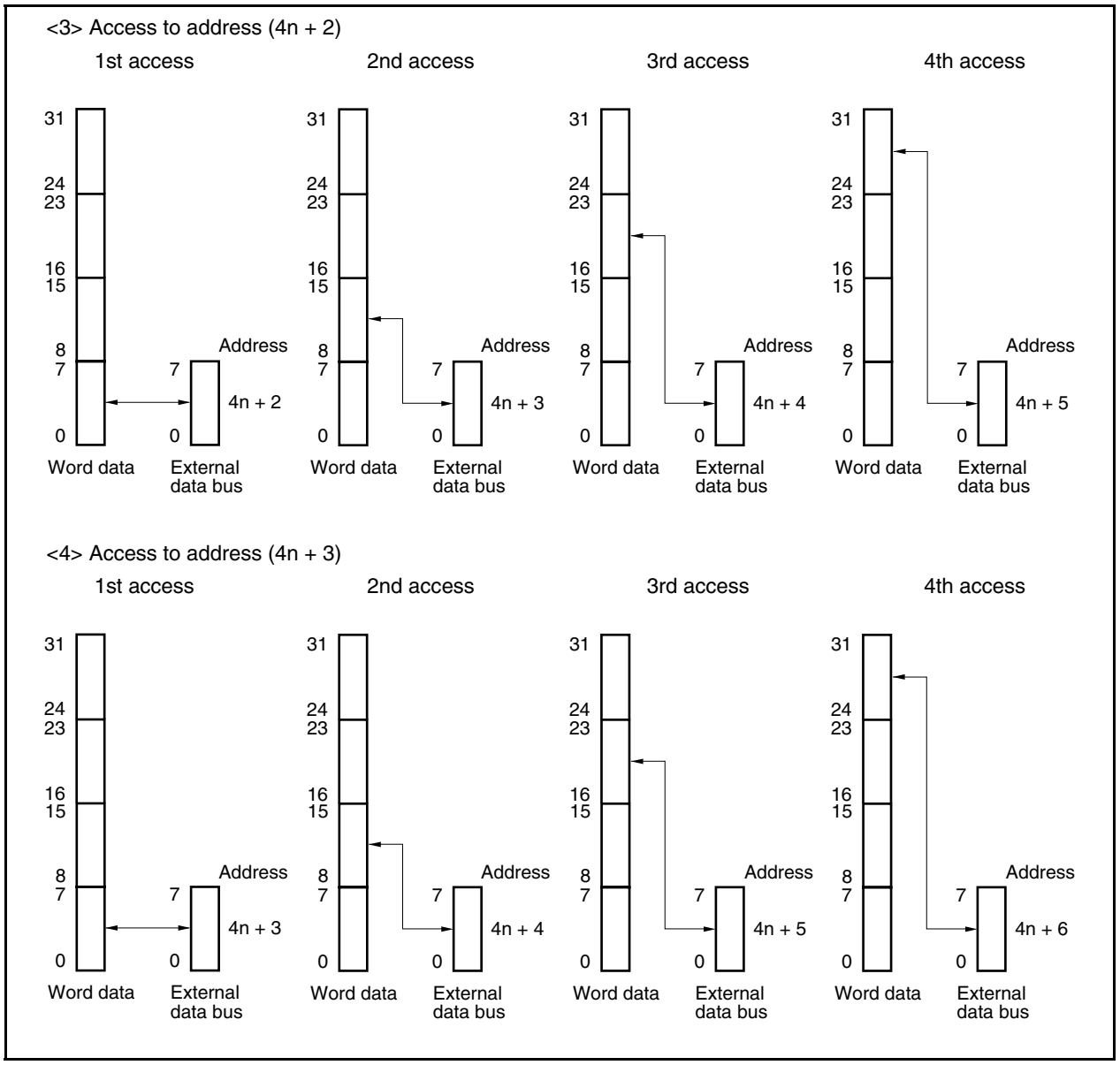
<4> Access to address (4n + 3)



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



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



18.6 Wait Function

18.6.1 Programmable wait function

(1) Data wait control register 0 (DWC0)

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

The number of wait states can be specified by program using DWC0 register. Just after system reset, all blocks have 7 data wait states inserted.

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

Reset sets this register to 7777H.

Note SRAM read/write cycle

Cautions 1. The internal ROM and internal RAM areas are not subject to programmable waits and ordinarily no wait access is carried out.

The on-chip peripheral I/O area is not subject to programmable waits, with wait control performed by each on-chip peripheral function only.

2. Write to the DWC0 register after reset, and then do not change the set value. Also, do not access an external memory area until the initial setting of the DWC0 register is complete. However, it is possible to access external memory areas whose initialization settings are complete.

After reset: 7777H R/W Address: FFFFF484H

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

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

DWn2	DWn1	DWn0	Specification of number of wait states inserted in each CSn space (n = 0, 1)
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

Caution Be sure to set bits 3, 7, 11, and 15 to “0”, and set bits 8 to 10 and 12 to 14 to “1”. If they are set other than above, the operation is not guaranteed.

(2) Address wait control register (AWC)

The AWC register can set an address setup wait state or address hold wait state that is to be inserted in each bus cycle. The address setup wait state is inserted before T1 state and the address hold wait state is inserted after T1 state.

Address setup wait state and address hold wait state insertion can be set with the AWC register for each CS space.

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

Reset sets this register to FFFFH.

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

2. During address setup wait state and address hold wait state, the $\overline{\text{WAIT}}$ pin-based external wait function is disabled.

3. Write to the AWC register after reset, and then do not change the set value.

After reset: FFFFH R/W Address: FFFFF488H

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

$\overline{\text{CSn}}$ signal

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

AHWn	Specification of address hold wait state inserted in each CSn space (n = 0, 1)
0	Not inserted
1	Inserted

ASWn	Specification of address setup wait state inserted in each CSn space (n = 0, 1)
0	Not inserted
1	Inserted

Caution Be sure to set bits 4 to 15 to "1". If they are set to "0", the operation is not guaranteed.

18.6.2 External wait function

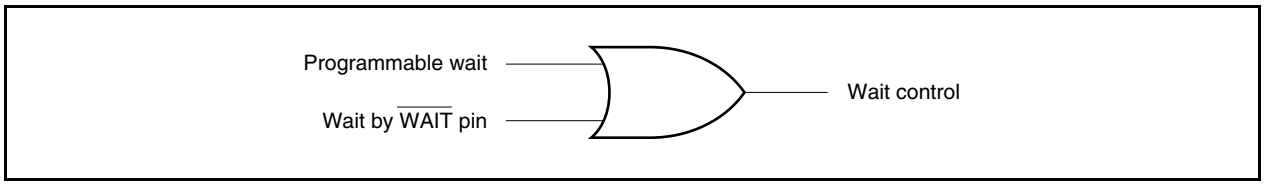
When an extremely slow memory, I/O, or asynchronous system is connected, an arbitrary number of wait states can be inserted in the bus cycle by the external wait pin ($\overline{\text{WAIT}}$) for synchronization with the external device.

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 the external bus clock frequency.

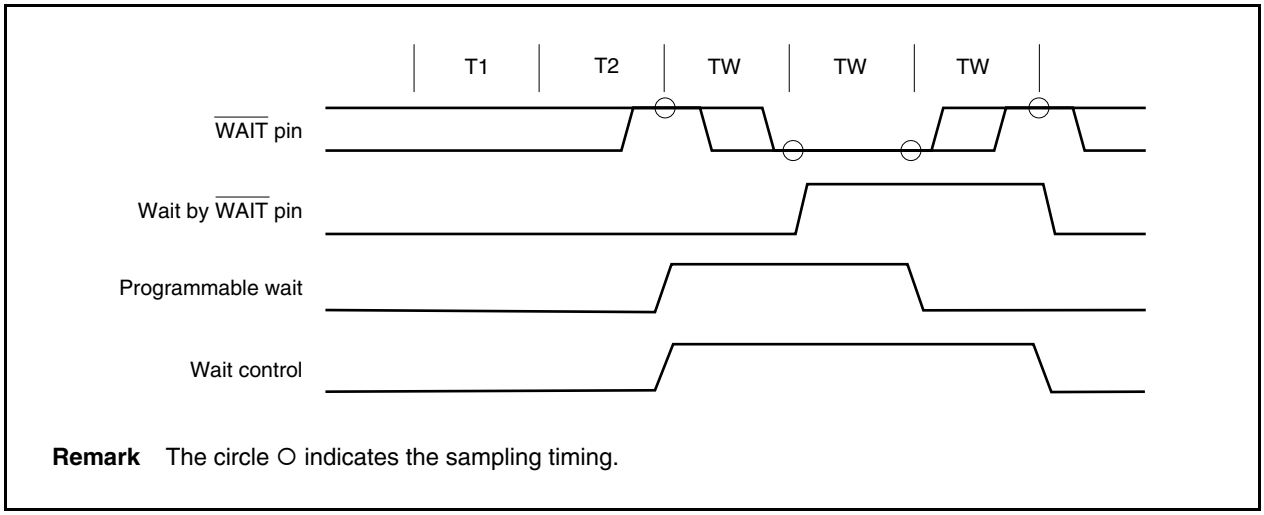
18.6.3 Relationship between programmable wait and external wait

A wait cycle is inserted as the result of an OR operation between the wait cycle specified by the set value of the programmable wait and the wait cycle 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 18-1. Example of Wait Insertion



Remark The circle ○ indicates the sampling timing.

18.6.4 Bus cycles in which wait function is valid

In the μPD70F3454GC-8EA-A, the number of waits can be specified for each memory block. The following shows the bus cycles in which the wait function is valid and the registers used for wait setting.

Table 18-2. Bus Cycles in Which Wait Function Is Valid

Bus Cycle	Wait Type	Programmable Wait Setting			Wait by $\overline{\text{WAIT}}$ Pin
		Register	Bit	Number of Waits	
SRAM, external ROM, external I/O cycles	Address setup wait	AWC	ASWn	0, 1	× (invalid)
	Address hold wait	AWC	AHWn	0, 1	× (invalid)
	Data wait	DWC0	DWn2 to DWn0	0 to 7	√ (valid)

Remark n = 0, 1

18.7 Idle State Insertion Function

The idle state is inserted after a read cycle or a write cycle to the SRAM, external ROM, or external I/O.

(1) Bus cycle control register (BCC)

To facilitate interfacing with low-speed memory devices, an idle state (TI) can be inserted into the current bus cycle after the T2 state (after TW state if a data wait state is inserted) to secure the data output float delay time on memory read access for each CS space. The bus cycle following the T2 state (or TW state) starts after the idle state is inserted.

An idle state can be inserted after a write access by using the bus clock division control register (DVC).

The idle state insertion setting can be specified by program using the BCC register. Immediately after the system reset, idle state insertion is automatically programmed for all memory blocks. For the timing when an idle state is inserted, see **18.8 Bus Timing**.

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

Reset sets this register to AAAAH.

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

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

3. The chip select signal ($\overline{CSn}$) does not become active in the idle state (n = 0, 1).

After reset: AAAAH R/W Address: FFFFF48AH

	15	14	13	12	11	10	9	8
BCC	1	0	1	0	1	0	1	0
$\overline{CSn}$ signal								

	7	6	5	4	3	2	1	0
	1	0	1	0	BC11	0	BC01	0
$\overline{CSn}$ signal					$\overline{CS1}$		$\overline{CS0}$	

BCn1	Specification of idle state inserted in each CSn space (n = 0, 1)
0	Not inserted
1	Inserted

Insertion of an idle state can be specified for each CSn space after completion of a read cycle or a write cycle.
If the DVC.BCWI bit = 0, however, the idle state is inserted only after completion of a read cycle and not after completion of a write cycle.

Caution Be sure to set bits 0, 2, 4, 6, 8, 10, 12, and 14 to "0", and set bits 5, 7, 9, 11, 13, and 15 to "1". If they are set other than above, the operation is not guaranteed.

(2) Bus clock division control register (DVC)

The DVC register is used to specify insertion of an idle state (TI) after completion of a write cycle, and an external bus clock frequency.

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

Reset sets this register to 81H.

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

2. Write to the DVC register after reset once (initial setting), and then do not change the set value. Also, do not access an external memory area until the initial setting of the DVC register is complete.

However, it is possible to access external memory areas whose initialization settings are complete.

After reset: 81H R/W Address: FFFFF48EH

	7	6	5	4	3	2	1	0
DVC	BCWI	0	0	0	0	0	DVC1	DVC0

BCWI	Specification of idle state inserted after write cycle ends
0	Not inserted
1	Inserted (only when BCC.BCn1 bit = 1)

DVC1	DVC0	Specification of external bus clock frequency (f_{BUS})
0	0	$f_{CLK}/1$ Note
0	1	$f_{CLK}/2$
1	0	Setting prohibited
1	1	$f_{CLK}/4$

Note Can be set only when $f_{CLK} \leq 32$ MHz.
Setting prohibited when 32 MHz < $f_{CLK} \leq 64$ MHz

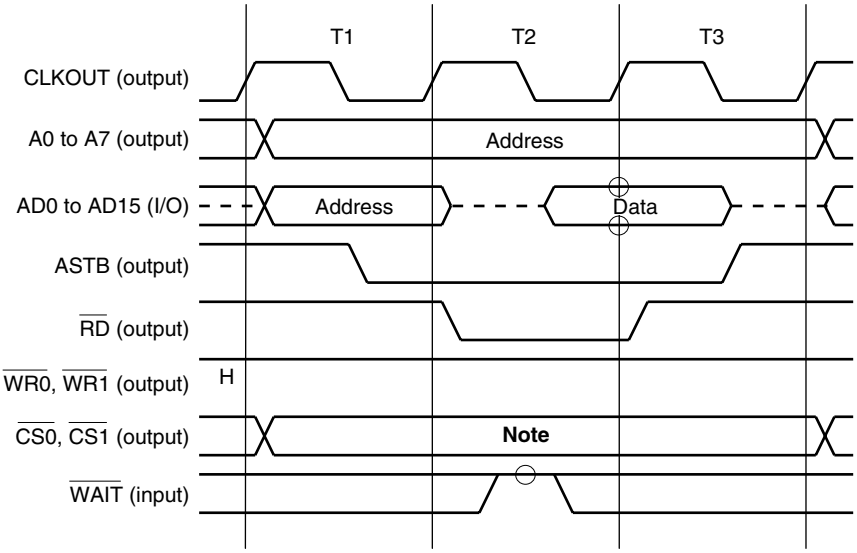
Cautions 1. Be sure to set the CLKOUT pin in the port mode before changing the setting of the DVC1 and DVC0 bits. Changing the setting of the DVC1 and DVC0 bits while the alternate function (CLKOUT) is used is prohibited.

2. Set the external bus clock frequency (f_{BUS}) in a range of 16 MHz ≤ f_{BUS} ≤ 32 MHz.

3. Be sure to set bits 2 to 6 to "0". If they are set to "1", the operation is not guaranteed.

18.8 Bus Timing

(1) Read cycle (basic cycle)

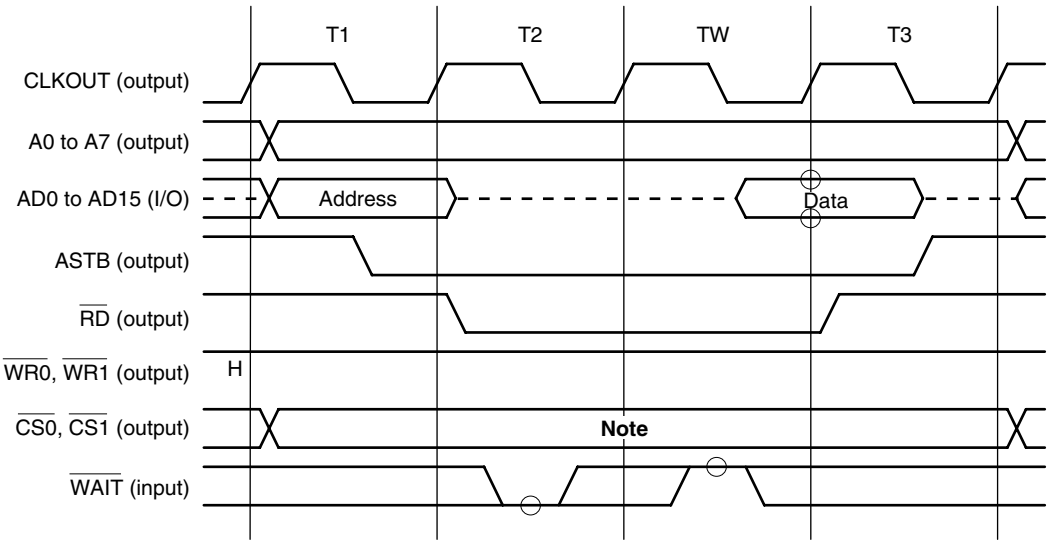


Note Only the CS space that can be accessed becomes active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(2) Read cycle (when data wait state (1 wait) insertion)

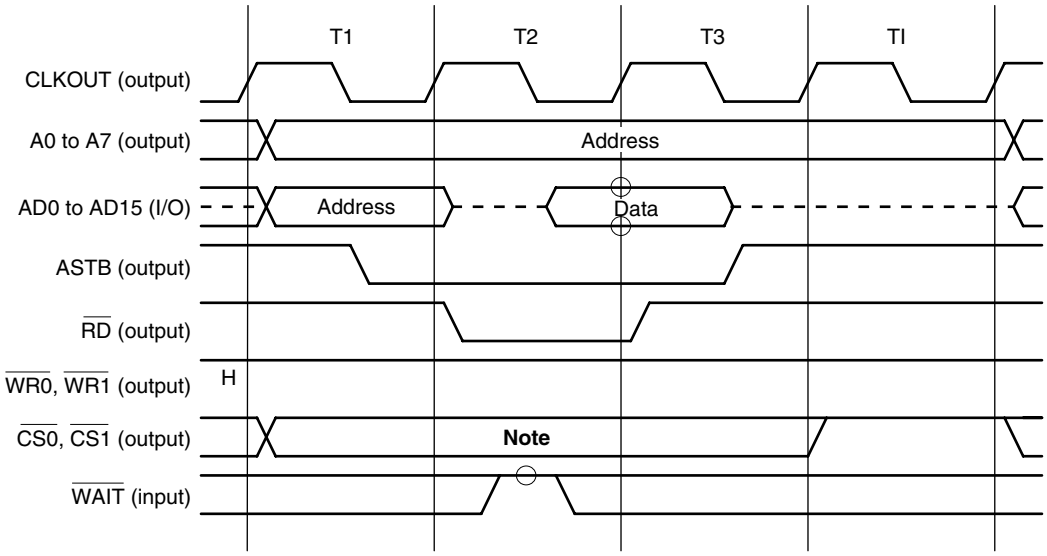


Note Only the CS space that can be accessed becomes active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(3) Read cycle (when idle state insertion)

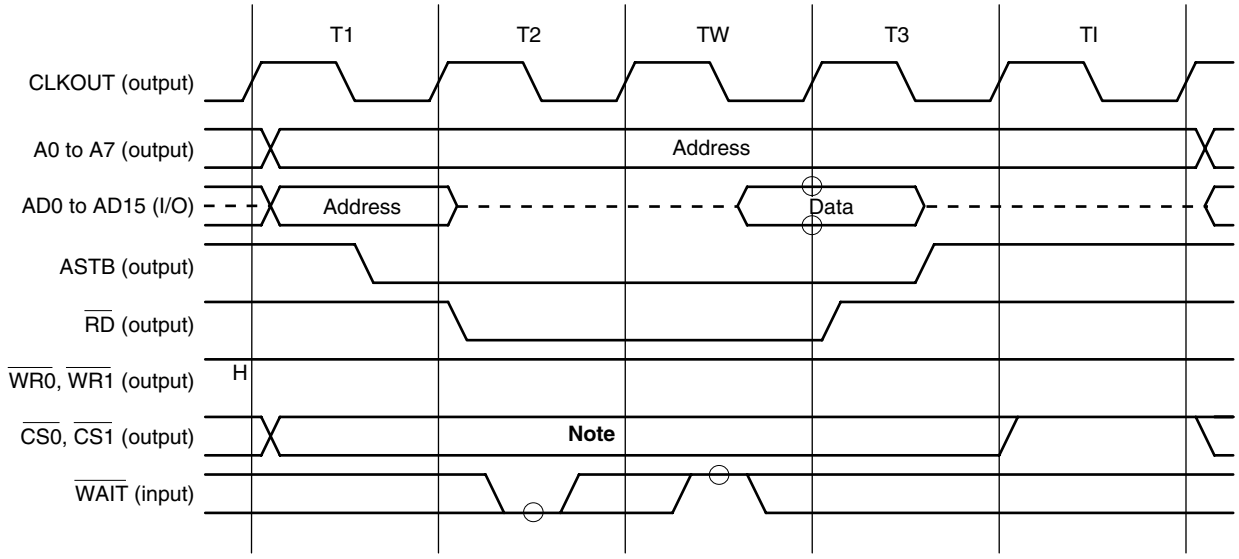


Note Only the CS space that can be accessed becomes active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(4) Read cycle (when data wait state (1 wait), idle state insertion)

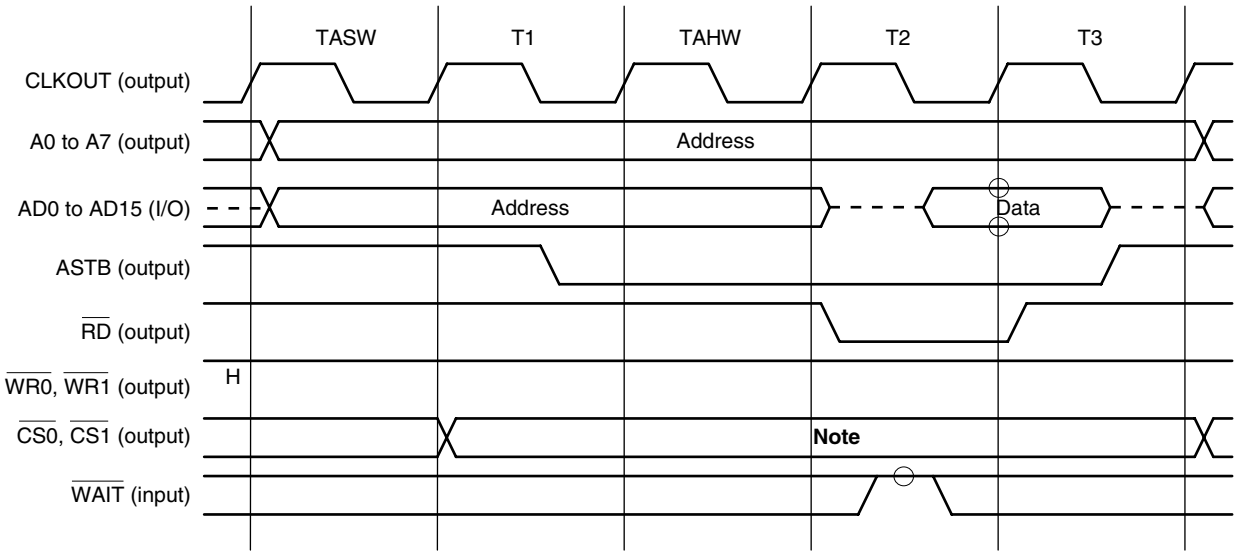


Note Only the CS space that can be accessed becomes active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(5) Read cycle (when address setup wait state, address hold wait state insertion)

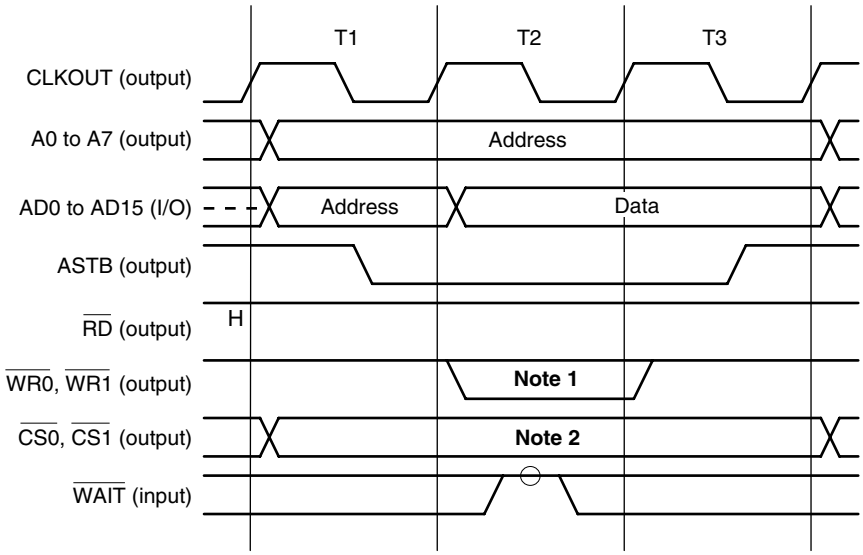


Note Only the CS space that can be accessed becomes active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(6) Write cycle (basic cycle)



Notes 1. The levels of these signals are as follows, depending on the access data bus width.

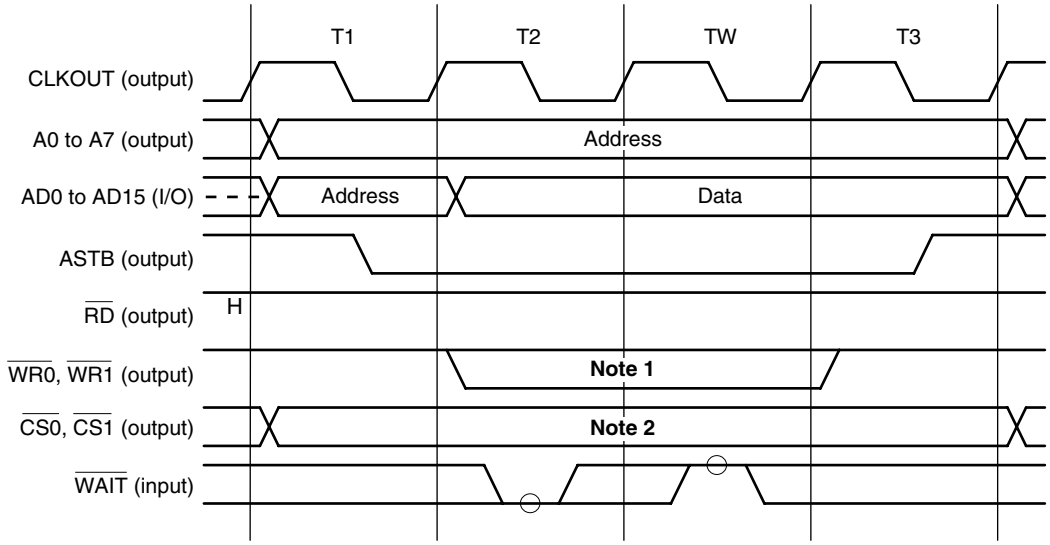
Access Data Bus Width	$\overline{WR1}$	$\overline{WR0}$
16 bits	Low level	Low level
8 bits	High level	Low level

2. Only the CS space that can be accessed becomes active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(7) Write cycle (when data wait state (1 wait) insertion)



Notes 1. The levels of these signals are as follows, depending on the access data bus width.

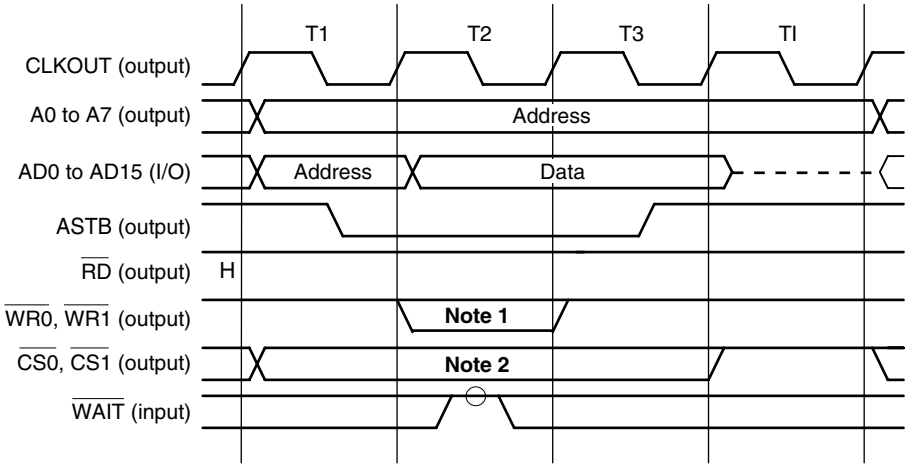
Access Data Bus Width	$\overline{WR1}$	$\overline{WR0}$
16 bits	Low level	Low level
8 bits	High level	Low level

2. Only the CS space that can be accessed becomes active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(8) Write cycle (when idle state insertion)



Notes 1. The levels of these signals are as follows, depending on the access data bus width.

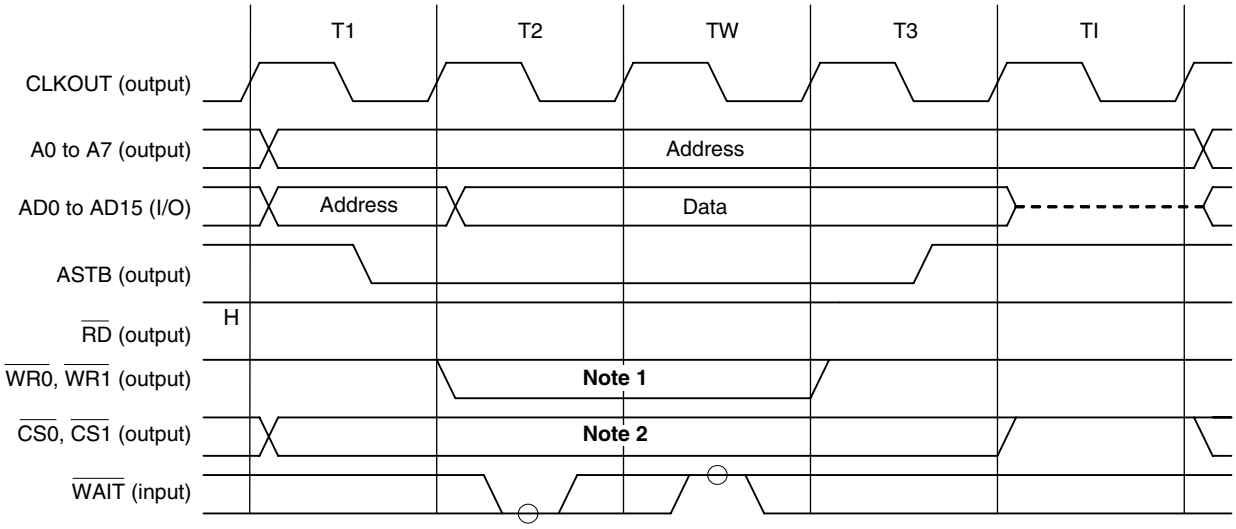
Access Data Bus Width	$\overline{WR1}$	$\overline{WR0}$
16 bits	Low level	Low level
8 bits	High level	Low level

2. Only the CS space that can be accessed becomes active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(9) Write cycle (when data wait state (1 wait), idle state insertion)



Notes 1. The levels of these signals are as follows, depending on the access data bus width.

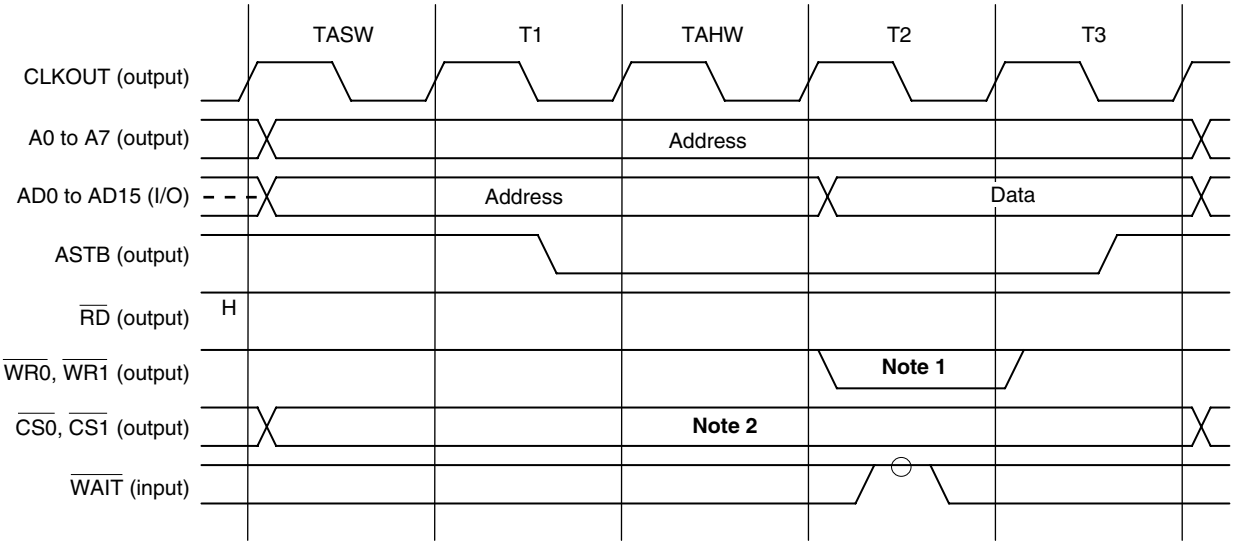
Access Data Bus Width	$\overline{WR1}$	$\overline{WR0}$
16 bits	Low level	Low level
8 bits	High level	Low level

2. Only the CS space that can be accessed becomes active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

(10) Write cycle (when address setup wait state, address hold wait state insertion)



Notes 1. The levels of these signals are as follows, depending on the access data bus width.

Access Data Bus Width	$\overline{WR1}$	$\overline{WR0}$
16 bits	Low level	Low level
8 bits	High level	Low level

2. Only the CS space that can be accessed becomes active.

Remarks 1. The circle ○ indicates the sampling timing.

2. The broken lines indicate the high-impedance state.

18.9 Bus Priority Order

There are two external bus cycles: instruction fetch and operand data access.

In order of priority, operand data access is the higher and instruction fetch is the lower.

An instruction fetch may be inserted between a read access and write access during a read modify write access.

Table 18-3. Bus Priority Order

Priority Order	External Bus Cycle	Bus Master
High	Operand data access	CPU
Low	Instruction fetch	CPU

18.10 Boundary Operation Conditions

18.10.1 Program space

Branching to the on-chip peripheral I/O area is prohibited. If the above is performed, undefined data is fetched, and fetching from the external memory is not performed.

18.10.2 Data space

The μPD70F3454GC-8EA-A is provided with an address misalign function.

Through this function, regardless of the data format (word or halfword), 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, a 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 lower 2 bits are 10, a halfword-length bus cycle will be generated 2 times.

CHAPTER 19 DMA FUNCTIONS (DMA CONTROLLER)

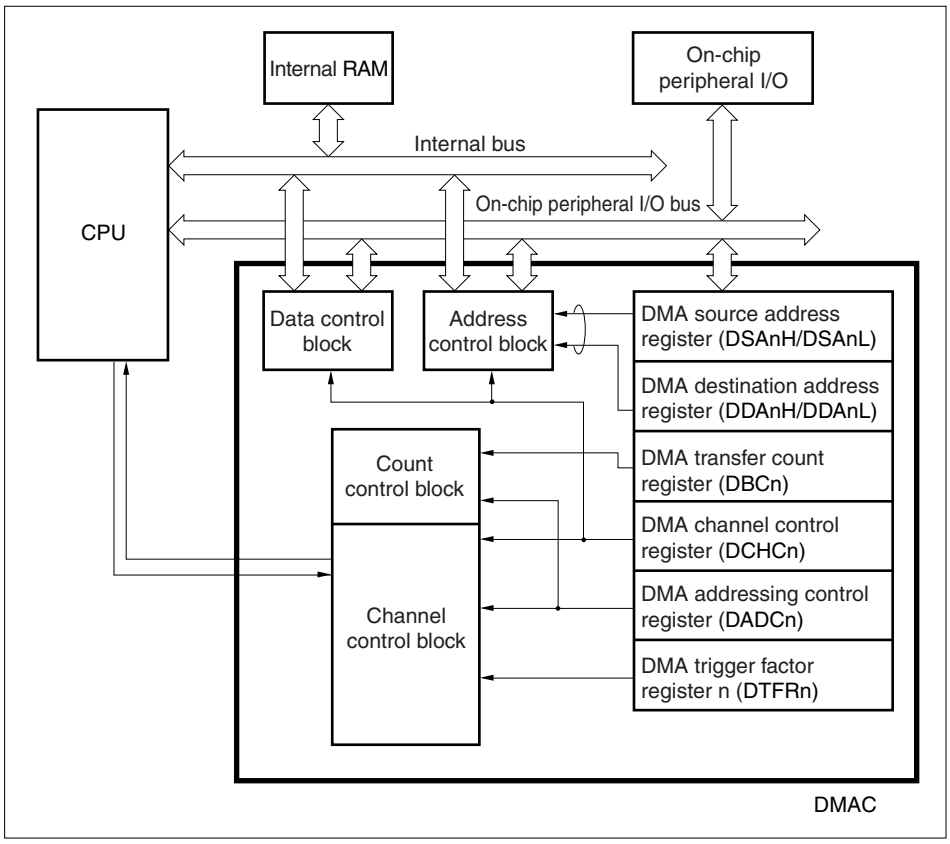
The V850E/IF3 and V850E/IG3 include a direct memory access (DMA) controller (DMAC) that executes and controls DMA transfer.

The DMAC controls data transfers between the internal memory and peripheral I/O, or between peripheral I/Os, based on requests by interrupts from the on-chip peripheral I/O (serial interface, timer, and A/D converter) or DMA requests issued by software triggers.

19.1 Features

- 4 independent DMA channels
- Transfer unit: 8/16 bits
- Maximum transfer count: 65536 (2^{16})
- Transfer type: 2-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, A/D converter)
  - Requests by software trigger
- Transfer targets
  - Internal memory ↔ peripheral I/O
  - Peripheral I/O ↔ peripheral I/O
- Next address setting function

19.2 Configuration



V850E/IF3, V850E/IG3

Remark n = 0 to 3

19.3 Control Registers

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

The DSA0 to DSA3 registers set the DMA transfer source address (28 bits) for DMA channel n (n = 0 to 3). These registers are divided into two 16-bit registers, DSAnH and DSAnL.

Since these registers are configured as 2-stage FIFO buffer registers consisting of the master register and slave register, a new transfer source address for DMA transfer can be specified during DMA transfer (see **19.8 Next Address Setting Function**). In this case, the newly set value of the DSAn register is transferred to the slave register and becomes valid only when DMA transfer has been completed normally and the DCHCn.TCn bit is set to 1, or when the DCHCn.INITn bit is set to 1 (n = 0 to 3). However, the set value of the DSAn register is invalid even when the DCHCn.Enn bit is cleared to 0 to disable DMA transfer and then the DSAn register is set.

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

The DSA0H to DSA3H registers can be read or written in 16-bit units.

Reset makes these registers undefined.

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.

After reset: Undefined R/W Address: DSA0H FFFFF082H, DSA1H FFFFF08AH, DSA2H FFFFF092H, DSA3H FFFFF09AH

DSAnH (n = 0 to 3)

15	14	13	12	11	10	9	8
IRSn	0	0	0	SAn27	SAn26	SAn25	SAn24

7	6	5	4	3	2	1	0
SAn23	SAn22	SAn21	SAn20	SAn19	SAn18	SAn17	SAn16

IRSn	DMA transfer source specification
0	On-chip peripheral I/O
1	Internal RAM

SAn27 to SAn16	Set the DMA transfer source address (A27 to A16). During DMA transfer, these bits store the next DMA transfer source address.

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

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

The DSA0L to DSA3L registers can be read or written in 16-bit units.

Reset makes these registers undefined.

After reset: Undefined R/W Address: DSA0L FFFFF080H, DSA1L FFFFF088H, DSA2L FFFFF090H, DSA3L FFFFF098H

DSAnL (n = 0 to 3)

15	14	13	12	11	10	9	8
SAn15	SAn14	SAn13	SAn12	SAn11	SAn10	SAn9	SAn8

7	6	5	4	3	2	1	0
SAn7	SAn6	SAn5	SAn4	SAn3	SAn2	SAn1	SAn0

SAn15 to SAn0	Set the DMA transfer source address (A15 to A0). During DMA transfer, these bits store the next DMA transfer source address.

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

The DDA0 to DDA3 registers set the DMA transfer destination address (28 bits) 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 consisting of the master register and slave register, a new transfer destination address for DMA transfer can be specified during DMA transfer (see **19.8 Next Address Setting Function**). In this case, the newly set value of the DDAn register is transferred to the slave register and becomes valid only when DMA transfer has been completed normally and the DCHCn.TCn bit is set to 1, or when the DCHCn.INITn bit is set to 1 (n = 0 to 3). However, the set value of the DDAn register is invalid even when the DCHCn.Enn bit is cleared to 0 to disable DMA transfer and then the DDAn register is set.

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

The DDA0H to DDA3H registers can be read or written in 16-bit units.

Reset makes these registers undefined.

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.

After reset: Undefined R/W Address: DDA0H FFFFF086H, DDA1H FFFFF08EH, DDA2H FFFFF096H, DDA3H FFFFF09EH

DDAnH (n = 0 to 3)

15	14	13	12	11	10	9	8
IRAn	0	0	0	DAn27	DAn26	DAn25	DAn24

7	6	5	4	3	2	1	0
DAn23	DAn22	DAn21	DAn20	DAn19	DAn18	DAn17	DAn16

IRAn	DMA transfer destination specification
0	On-chip peripheral I/O
1	Internal RAM

DAn27 to DAn16	Set the DMA transfer destination address (A27 to A16). During DMA transfer, these bits store the next DMA transfer destination address.

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

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

The DDA0L to DDA3L registers can be read or written in 16-bit units.

Reset makes these registers undefined.

After reset: Undefined R/W Address: DDA0L FFFFF084H, DDA1L FFFFF08CH, DDA2L FFFFF094H, DDA3L FFFFF09CH

DDAnL (n = 0 to 3)

15	14	13	12	11	10	9	8
DAn15	DAn14	DAn13	DAn12	DAn11	DAn10	DAn9	DAn8

7	6	5	4	3	2	1	0
DAn7	DAn6	DAn5	DAn4	DAn3	DAn2	DAn1	DAn0

DAn15 to DAn0	Set the DMA transfer destination address (A15 to A0). During DMA transfer, these bits store the next DMA transfer destination address.

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

The DBC0 to DBC3 registers are 16-bit registers that set the byte transfer count for DMA channel n (n = 0 to 3). These registers store the remaining transfer count during DMA transfer.

Since these registers are configured as 2-stage FIFO buffer registers consisting of the master register and slave register, a new DMA byte transfer count for DMA transfer can be specified during DMA transfer (see **19.8 Next Address Setting Function**). In this case, the newly set value of the DBCn register is transferred to the slave register and becomes valid only when DMA transfer has been completed normally and the DCHCn.TCn bit is set to 1, or when the DCHCn.INITn bit is set to 1 (n = 0 to 3). However, the set value of the DBCn register is invalid even when the DCHCn.Enn bit is cleared to 0 to disable DMA transfer and then the DBCn register is set.

These registers are decremented by 1 for each transfer, and transfer ends when a borrow occurs.

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

Reset makes these registers undefined.

Caution Do not set the DBCn register while DMA is suspended.

Remark If the DBCn register is read during DMA transfer after a terminal count has occurred without the register being overwritten, the value set immediately before the DMA transfer will be read out (0000H will not be read, even if DMA transfer has ended).

After reset: Undefined R/W Address: DBC0 FFFFF0C0H, DBC1 FFFFF0C2H, DBC2 FFFFF0C4H, DBC3 FFFFF0C6H

DBCn (n = 0 to 3)

15	14	13	12	11	10	9	8
BCn15	BCn14	BCn13	BCn12	BCn11	BCn10	BCn9	BCn8

7	6	5	4	3	2	1	0
BCn7	BCn6	BCn5	BCn4	BCn3	BCn2	BCn1	BCn0

BCn15 to BCn0	Transfer count setting (store remaining transfer count during DMA transfer)
0000H	Transfer count 1 or remaining transfer count
0001H	Transfer count 2 or remaining transfer count
:	:
FFFFH	Transfer count 65536 (2^{16}) or remaining transfer count

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

The DADC0 to DADC3 registers are 16-bit registers that control the DMA transfer mode for DMA channel n (n = 0 to 3). These registers cannot be accessed during a DMA operation.

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

Reset sets these registers to 0000H.

Cautions 1. The DSn0 bit sets how many bits of data are to be transferred.
If the transfer data size is set to 16 bits, transfer is always started from an address with the lowest bit of the address aligned to "0". In this case, transfer cannot be started from an odd address.

2. Set the DADCn register when the target channel is in one of the following periods (the operation is not guaranteed if the register is set at any other time).
- **Period from system reset to the generation of the first DMA transfer request**
- **Period from end of DMA transfer (after terminal count) to the generation of the next DMA transfer request**
- **Period from forced termination of DMA transfer (after the DCHCn.INITn bit was set to 1) to the generation of the next DMA transfer request**

After reset: 0000H R/W Address: DADC0 FFFFF0D0H, DADC1 FFFFF0D2H, DADC2 FFFFF0D4H, DADC3 FFFFF0D6H

DADCn (n = 0 to 3)

15	14	13	12	11	10	9	8
0	DSn0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SADn1	SADn0	DADn1	DADn0	TMn1	TMn0	0	0

DSn0	Setting of transfer data size for DMA transfer
0	8 bits
1	16 bits

SADn1	SADn0	Setting of count direction of transfer source address for DMA channel n
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

DADn1	DADn0	Setting of count direction of transfer destination address for DMA channel n
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

TMn1	TMn0	Setting of transfer mode during DMA transfer
0	0	Single transfer mode
0	1	Single-step transfer mode
1	0	Setting prohibited
1	1	Block transfer mode

Caution Be sure to set bits 15, 13 to 8, 1, and 0 to "0". If they are set to "1", the operation is not guaranteed.

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

The DCHC0 to DCHC3 registers are 8-bit registers that control the DMA transfer operating mode for DMA channel n (n = 0 to 3).

These registers can be read or written in 8-bit or 1-bit units. (However, bit 7 is read-only.)

Reset sets these registers to 00H.

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

2. Set the MLEn bit when the target channel is in one of the following periods (the operation is not guaranteed if the bit is set at any other time).
- **Period from system reset to the generation of the first DMA transfer request**
- **Period from end of DMA transfer (after terminal count) to the generation of the next DMA transfer request**
- **Period from forced termination of DMA transfer (after the INITn bit was 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 operation is performed in the same manner as when transfer is ended (the TCn bit is set to 1). (The Enn bit is cleared to 0 upon forced termination, regardless of the value of the MLEn bit.)
In this case, the Enn bit must be set to 1 and the TCn bit must be read (cleared to 0) when the next DMA transfer request is made.

4. Upon end of DMA transfer (during terminal count), each bit is updated with the Enn bit cleared to 0 and then the TCn bit set to 1. If the statuses of the TCn bit and Enn bit are polled and if the DCHCn register is read while each bit is updated, therefore, a value indicating the status "transfer not ended and prohibited" (TCn bit = 0 and Enn bit = 0) may be read (this is not abnormal).

5. Be sure to read (clear to 0) the TCn bit after end of DMA transfer (after terminal count). The TCn bit does not have to be read (cleared to 0) only if the following two conditions are satisfied.
- **The MLEn bit is set to 1 upon end of DMA transfer (during terminal count).**
- **The next DMA transfer (hardware DMA) start factor is an interrupt from the on-chip peripheral I/O (hardware DMA)**

If even one of these conditions is not satisfied, be sure to read (clear to 0) the TCn bit before the next DMA transfer request is generated.

The operation cannot be guaranteed if the next DMA transfer request is generated while the TCn bit is set to 1.

6. Do not set the Enn and STGn bits while DMA is suspended. Otherwise, the operation is not guaranteed.

7. Do not end DMA transfer by clearing the Enn bit to 0.

8. The relationship between the status of DMA transfer and the register value is as follows.
- **DMA transfer is in progress: TCn bit = 0, Enn bit = 1**
- **DMA transfer is aborted: TCn bit = 0, Enn bit = 0**
- **DMA transfer is stopped (ends): TCn bit = 1**

After reset: 00H R/W Address: DCHC0 FFFFF0E0H, DCHC1 FFFFF0E2H, DCHC2 FFFFF0E4H, DCHC3 FFFFF0E6H

	<7>	6	5	4	<3>	<2>	<1>	<0>
DCHCn	TCn	0	0	0	MLEn	INITn	STGn	Enn

(n = 0 to 3)

TCn[Note 1]	Status bit that indicates whether DMA transfer via DMA channel n has ended or not
0	DMA transfer has not ended.
1	DMA transfer has ended.
This bit is set (1) at the last DMA transfer and cleared (0) when it is read. If DMA transfer is executed to transfer data from the internal RAM, this bit is set (1) 4 clocks after end of the last transfer.	

MLEn	When this bits is set (1) at DMA transfer end (at the terminal count output), the Enn bit is not cleared (0) and the DMA transfer enabled state is retained. If the next DMA transfer start factor is input from an on-chip peripheral I/O (hardware DMA), the DMA transfer request is acknowledged even if the TCn bit is not read. If the next DMA transfer start factor is input by setting the STGn bit to 1 (software DMA), the DMA transfer request is acknowledged if the TCn bit is read and cleared (0). When this bit is cleared (0) at DMA transfer end (at the terminal count output), the Enn bit is cleared (0) and the DMA transfer disabled state is entered. At the next DMA transfer request, the TCn bit must be read and the Enn bit must be set (1).

INITn[Note 2]	If this bit is set (1) during DMA transfer or while DMA is suspended, DMA transfer is forcibly terminated.

STGn[Note 2]	If this bit is set (1) in the DMA transfer enabled state (TCn bit = 0, Enn bit = 1), DMA transfer is started.

Enn	Setting whether DMA transfer via DMA channel n is to be enabled or disabled
0	DMA transfer disabled
1	DMA transfer enabled
• This bit is cleared (0) when DMA transfer ends. It is also cleared (0) when DMA transfer is forcibly terminated by setting (1) the INITn bit. • If the Enn bit is set (1), do not set it until DMA transfer has been ended the number of times set by the DBCn register or DMA transfer is forcibly terminated by the INITn bit.	

Notes 1. TCn bit is read-only.

2. INITn and STGn bits are write-only. If these bits are read, 0 is read.

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

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

The DTFR0 to DTFR3 registers are 8-bit registers that control the DMA transfer start trigger via interrupt requests from on-chip peripheral I/O.

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

These registers can be read or written in 8-bit or 1-bit units. However, only bit 7 (DFn) can be read or written in 1-bit units; bits 5 to 0 (IFCn5 to IFCn0) can only be read or written in 8-bit units.

Reset sets these registers to 00H.

Cautions 1. Be sure to follow the steps below when changing the DTFRn register settings.

- **When the values to be set to the IFCn5 to IFCn0 bits are not set to the IFCm5 to IFCm0 bits of another channel (n = 0 to 3, m = 0 to 3, n ≠ m)**
  - **<1> Stop the DMAn operation of the channel to be rewritten (DCHCn.Enn bit = 0).**
  - **<2> Change the DTFRn register settings. (Be sure to set DFn bit = 0 and change the settings in the 8-bit manipulation.)**
  - **<3> To clear a DMA transfer request, clear the DMA transfer request flag (DTFRn.DFn) to 0.**
  - **<4> Enable the DMAn operation (Enn bit = 1).**
- **When the values to be set to the IFCn5 to IFCn0 bits are set to the IFCm5 to IFCm0 bits of another channel (n = 0 to 3, m = 0 to 3, n ≠ m)**
  - **<1> Stop the DMAn operation of the channel to be rewritten (DCHCn.Enn bit = 0).**
  - **<2> Stop the DMAm operation of the channel where the same values are set to the IFCm5 to IFCm0 bits as the values to be used to rewrite the IFCn5 to IFCn0 bits (DCHCm.Emm bit = 0).**
  - **<3> Change the DTFRn register settings. (Be sure to set the DFn bit = 0 and change the settings in the 8-bit manipulation.)**
  - **<4> To clear a DMA transfer request, clear the DMA transfer request flag (DTFRn.DFn) to 0.**
  - **<5> Enable the DMAn operation (Enn and Emm bits = 1).**

2. An interrupt request from an on-chip peripheral I/O input in the standby mode (IDLE or STOP mode) is held pending as a DMA transfer start factor. The held DMA start factor is executed after restoring to the normal operation mode.

3. If the start factor of DMA transfer is changed using the IFCn5 to IFCn0 bits, be sure to set (0) the DFn bit by instruction immediately after.

After reset: 00H R/W Address: DTFR0 FFFFF810H, DTFR1 FFFFF812H, DTFR2 FFFFF814H, DTFR3 FFFFF816H

	<7>	6	5	4	3	2	1	0
DTFRn (n = 0 to 3)	DFn	0	IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0

DFn[Note]	DMA transfer request flag
0	DMA transfer not requested
1	DMA transfer requested

Note Do not set the DFn bit to “1” by software.

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 forcibly terminated by software), stop the operation of the source causing the interrupt, and then write 0 to the DFn bit (for example, disable reception in the case of serial reception). If it is clear that the interrupt will not occur until DMA transfer is resumed next, it is not necessary to stop the operation of the source causing the interrupt.

Cautions 1. For the IFCn5 to IFCn0 bits, see Table 19-1 DMA Start Factors.

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

Table 19-1. DMA Start Factors (1/2)

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	INTLVIL
0	0	0	0	1	0	INTLVIH
0	0	0	0	1	1	INTP11
0	0	0	1	0	0	INTP12
0	0	0	1	0	1	INTP13
0	0	0	1	1	0	INTP15
0	0	0	1	1	1	INTTB0OV_BASE[Note]
0	0	1	0	0	0	INTTB1OV_BASE[Note]
0	0	1	0	0	1	INTCMP0L
0	0	1	0	1	0	INTCMP0F
0	0	1	0	1	1	INTCMP1L
0	0	1	1	0	0	INTCMP1F
0	0	1	1	0	1	INTTB0CC0
0	0	1	1	1	0	INTTB0CC1
0	0	1	1	1	1	INTTB0CC2
0	1	0	0	0	0	INTTB0CC3
0	1	0	0	0	1	INTTB1CC0
0	1	0	0	1	0	INTTB1CC1
0	1	0	0	1	1	INTTB1CC2
0	1	0	1	0	0	INTTB1CC3
0	1	0	1	0	1	INTTTEQC00
0	1	0	1	1	0	INTTTEQC01
0	1	0	1	1	1	INTTTEQC10
0	1	1	0	0	0	INTTTEQC11
0	1	1	0	0	1	INTTA0CC0
0	1	1	0	1	0	INTTA0CC1
0	1	1	0	1	1	INTTA1CC0
0	1	1	1	0	0	INTTA1CC1
0	1	1	1	0	1	INTTA2CC0
0	1	1	1	1	0	INTTA2CC1
0	1	1	1	1	1	INTTA3CC0
1	0	0	0	0	0	INTTA3CC1
1	0	0	0	0	1	INTTA4CC0
1	0	0	0	1	0	INTTA4CC1
1	0	0	0	1	1	INTDMA0
1	0	0	1	0	0	INTDMA1
1	0	0	1	0	1	INTDMA2
1	0	0	1	1	0	INTDMA3

Remark n = 0 to 3

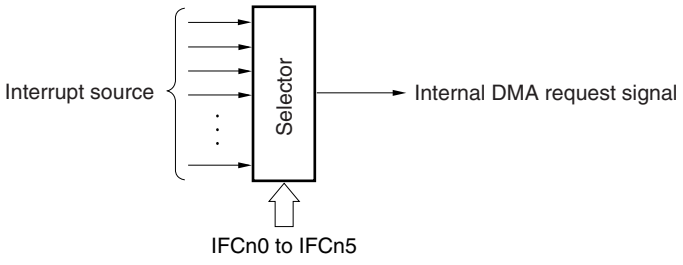
Note INTTBaOV_BASE is an interrupt signal before INTTBaOV is culled by the TMQa option in the 6-phase PWM output mode (a = 0, 1). For details, see **Figure 10-2 TMQn Option**.

Table 19-1. DMA Start Factors (2/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt Source
1	0	0	1	1	1	INTUBTIR
1	0	1	0	0	0	INTUBTIT
1	0	1	0	0	1	INTUBTIF
1	0	1	0	1	0	INTUA0R
1	0	1	0	1	1	INTUA0T
1	0	1	1	0	0	INTCB0R
1	0	1	1	0	1	INTCB0T
1	0	1	1	1	0	INTUA1R
1	0	1	1	1	1	INTUA1T
1	1	0	0	0	0	INTCB1R
1	1	0	0	0	1	INTCB1T
1	1	0	0	1	0	INTUA2R
1	1	0	0	1	1	INTUA2T
1	1	0	1	0	0	INTCB2R
1	1	0	1	0	1	INTCB2T
1	1	0	1	1	0	INTIIC
1	1	0	1	1	1	INTAD0
1	1	1	0	0	0	INTAD1
1	1	1	0	0	1	INTAD2
1	1	1	0	1	0	INTTM0EQ0
1	1	1	0	1	1	INTTM1EQ0
1	1	1	1	0	0	INTTM2EQ0
1	1	1	1	0	1	INTTM3EQ0
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).



Cautions 1. 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.

2. If the frequency of the CPU clock falls below the clock of each on-chip peripheral I/O because of the setting of prescaler 2 of the clock generator, the DMA transfer start factor may not be acknowledged.

19.4 Transfer Modes

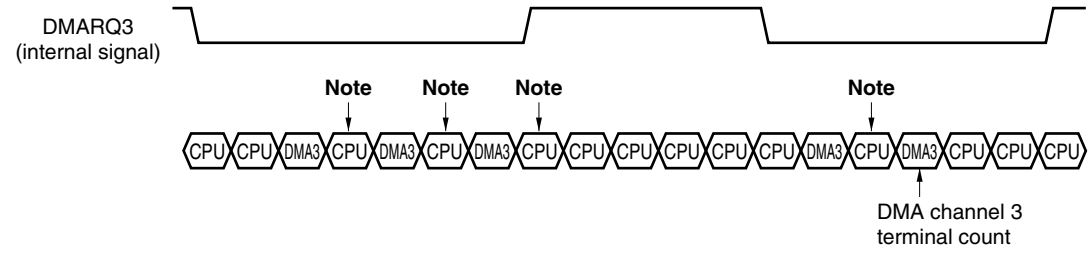
19.4.1 Single transfer mode

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. If another DMA transfer request with a lower priority occurs one clock after single transfer has been completed, however, this request does not take precedence even if the previous DMA transfer request signal with a higher priority remains active. DMA transfer with the newly requested lower priority request is executed after the CPU bus has been released.

Figures 19-1 to 19-4 show examples of single transfer.

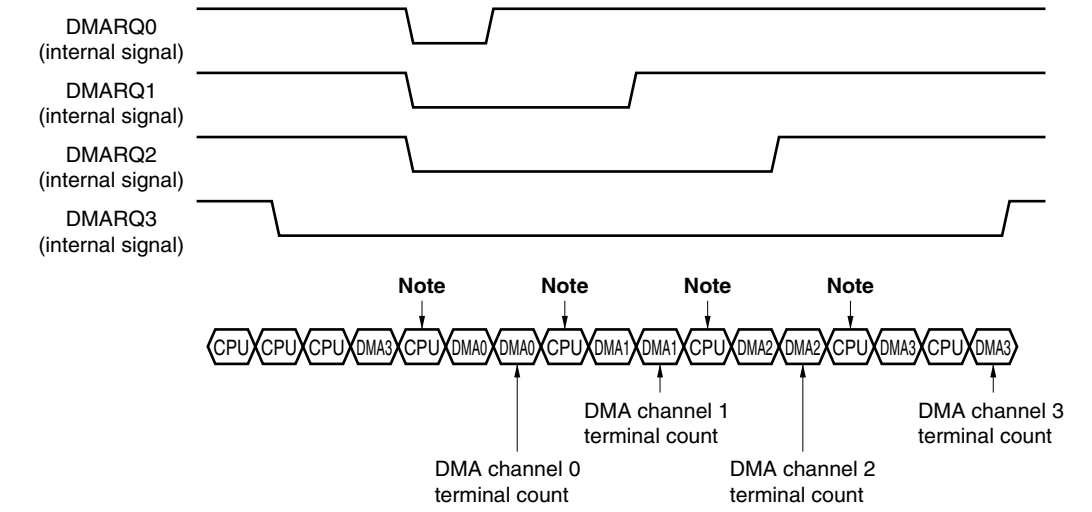
Figure 19-1. Single Transfer Example 1



Note The bus is always released.

Figure 19-2 shows an example of a single transfer in which a higher priority DMA request is issued. DMA channels 0 to 2 are in the block transfer mode and channel 3 is in the single transfer mode.

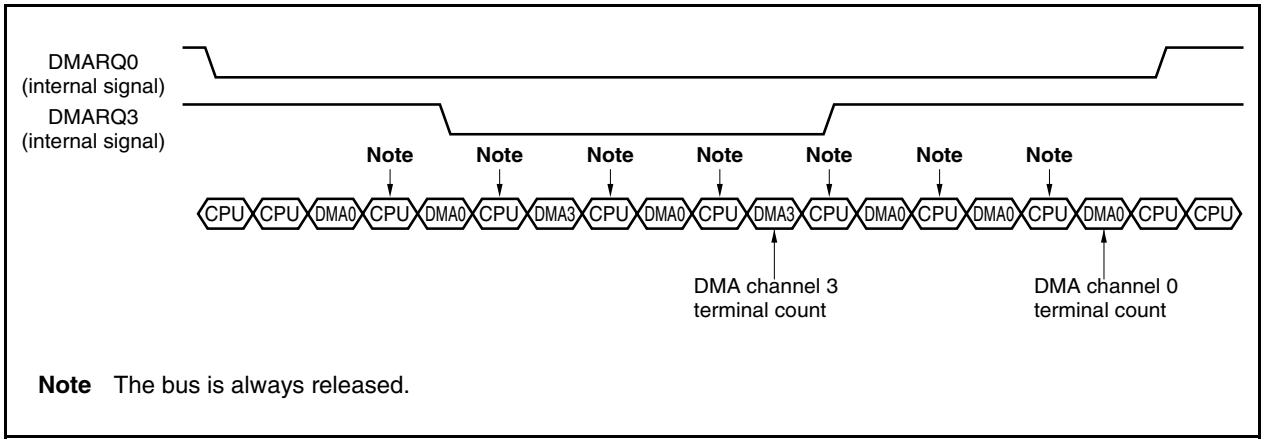
Figure 19-2. Single Transfer Example 2



Note The bus is always released.

Figure 19-3 is an example of single transfer where a DMA transfer request with a lower priority is issued one clock after single transfer has been completed. DMA channels 0 and 3 are used for single transfer. If two DMA transfer request signals become active at the same time, two DMA transfer operations are alternately executed.

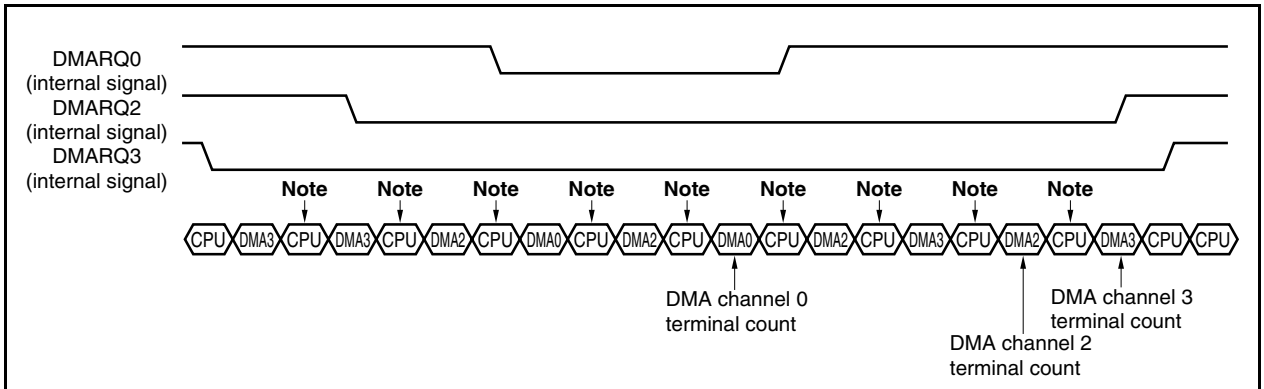
Figure 19-3. Single Transfer Example 3



Note The bus is always released.

Figure 19-4 is an example of single transfer where two or more DMA transfer requests with a lower priority are issued one clock after single transfer has been completed. DMA channels 0, 2, and 3 are used for single transfer. If three or more DMA transfer request signals become active at the same time, two DMA transfer operations are alternately executed, starting from the one with the highest priority.

Figure 19-4. Single Transfer Example 4



Note The bus is always released.

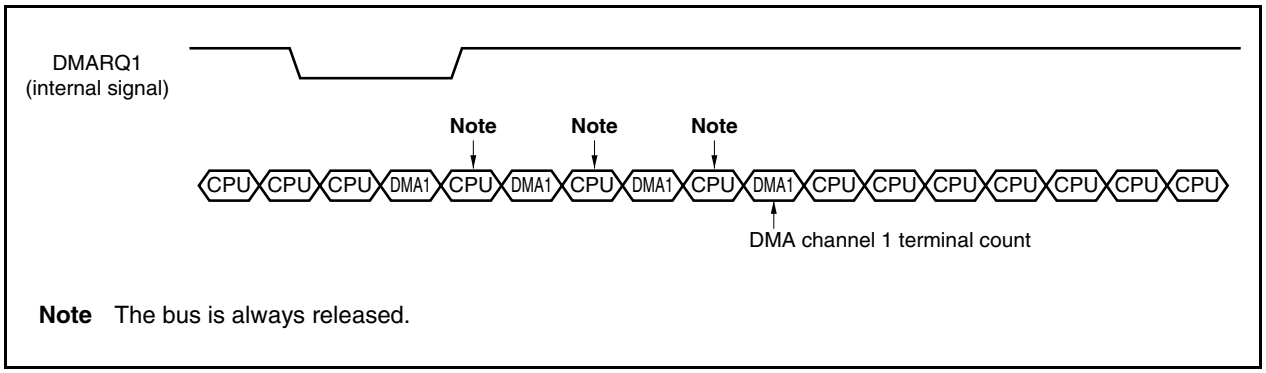
19.4.2 Single-step transfer mode

In single-step transfer mode, the DMAC releases the bus at each byte/halfword transfer. If there is a subsequent DMA transfer request signal, transfer is performed again. 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.

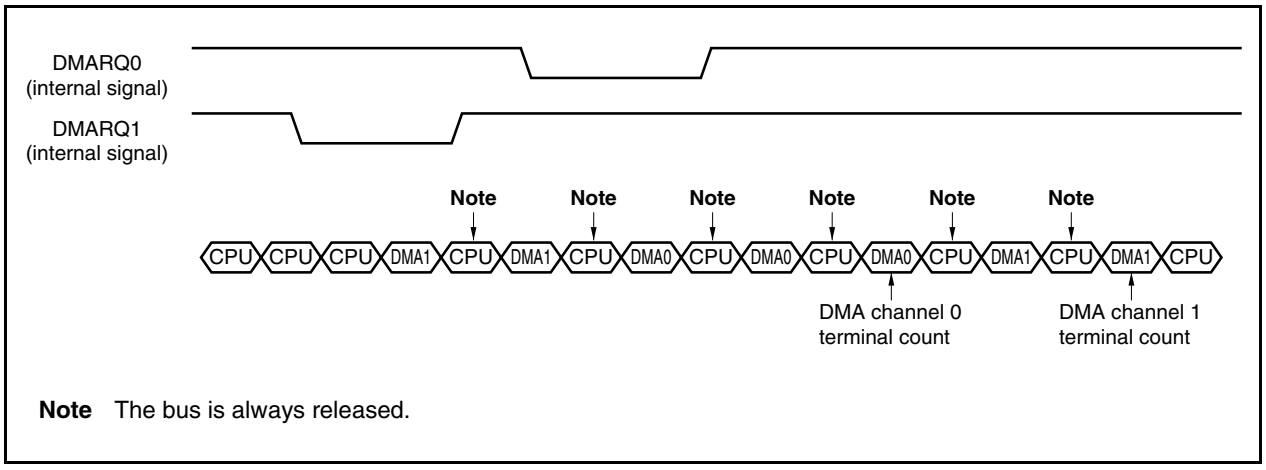
The following shows an example of a single-step transfer. Figure 19-6 shows an example of single-step transfer made in which a higher priority DMA request is issued. DMA channels 0 and 1 are in the single-step transfer mode.

Figure 19-5. Single-Step Transfer Example 1



Note The bus is always released.

Figure 19-6. Single-Step Transfer Example 2



Note The bus is always released.

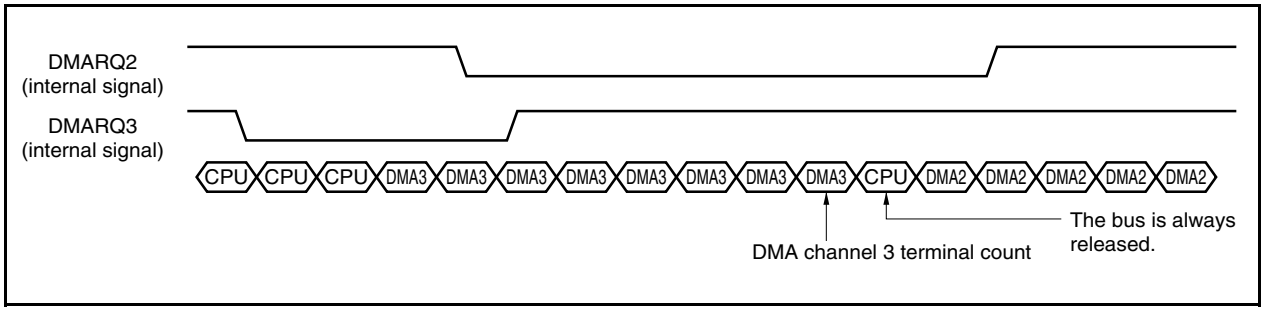
19.4.3 Block transfer mode

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 19-7. Block Transfer Example



19.5 Transfer Types

19.5.1 2-cycle transfer

In 2-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 a read cycle and a write cycle.

19.6 Transfer Target

19.6.1 Transfer type and transfer target

Table 19-2 lists the relationship between the transfer type and transfer target. The mark "√" means "transfer possible", and the mark "×" means "transfer impossible".

Table 19-2. Relationship Between Transfer Type and Transfer Target

		Destination		
		Internal ROM	On-Chip Peripheral I/O[Note]	Internal RAM
Source	On-chip peripheral I/O[Note]	×	√	√
	Internal RAM	×	√	×
	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 19-2.

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 DMA transfer is executed to transfer data of an on-chip peripheral I/O register (as a transfer source or destination), be sure to specify the same transfer size as the register size. For example, to execute DMA transfer of an 8-bit register, be sure to specify byte (8-bit) transfer.

19.7 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 Do not start two or more DMA channels with the same start factor, in which case a DMA channel with a lower priority may be acknowledged before a DMA channel with a higher priority.

19.8 Next Address Setting Function

The DSAnH, DSAnL, DDAnH, DDAnL, and DBCn registers are two-stage FIFO buffer registers consisting of a master register and a slave register (n = 0 to 3).

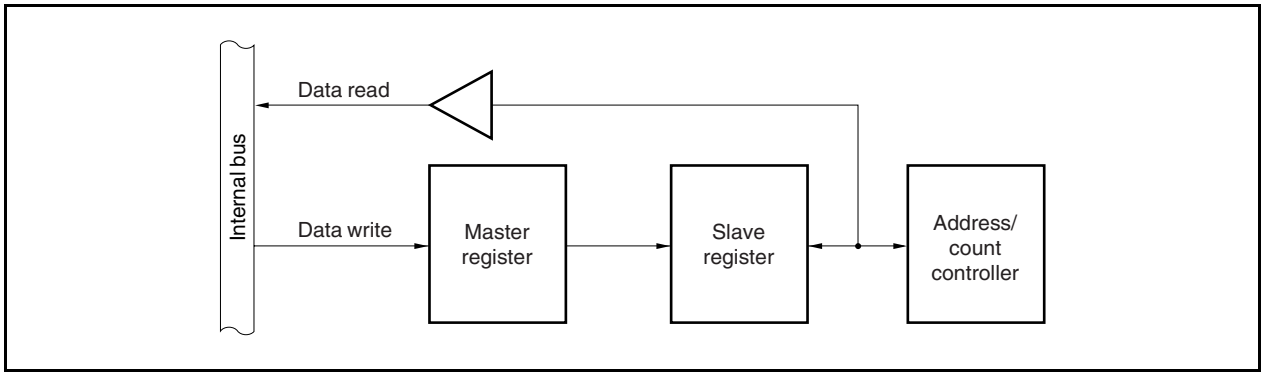
When the terminal count is issued, these registers are automatically rewritten with the value that was set immediately before.

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 To make new DMA transfer setting, confirm that DMA transfer has been started. If a new setting is made before the start of DMA transfer, the set value is overwritten to both the master and slave registers.

Figure 19-8 shows the configuration of the buffer register.

Figure 19-8. Buffer Register Configuration



The actual DMA transfer is executed in accordance with the contents of the slave register.

The set value to be reflected upon the master register and slave register differs as follows, depending on the timing (period) of setting.

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

The set values are reflected on both the master and slave registers.

(2) During DMA transfer (period from the generation of DMA transfer request to completion of DMA transfer)

The set value is reflected only on the master register and not on the slave register (the slave register holds the set value for the next DMA transfer).

After completion of DMA transfer, however, the contents of the master register are automatically overwritten to the slave register.

If the value of a 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) Period from completion of DMA transfer to start of next DMA transfer

The set value is reflected on both the master and slave registers.

Remark "Completion of DMA transfer" means either of the following cases.

- End of DMA transfer (terminal count)
- Forced termination of DMA transfer (setting DCHCn.INITn bit to 1).

19.9 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 both 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 both 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 ended through manipulation (setting to 1) of the DCHCn.STGn bit, 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 ended 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 the next time (the second time) after checking whether DMA transfer started by the first manipulation of the STGn bit has been ended.

End of DMA transfer can be checked by checking the contents of the DBCn register.

(1) Request from software

If the DCHCn.STGn, DCHCn.Enn, and DCHCn.TCn bits 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 DCHCn.Enn and DCHCn.TCn bits 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

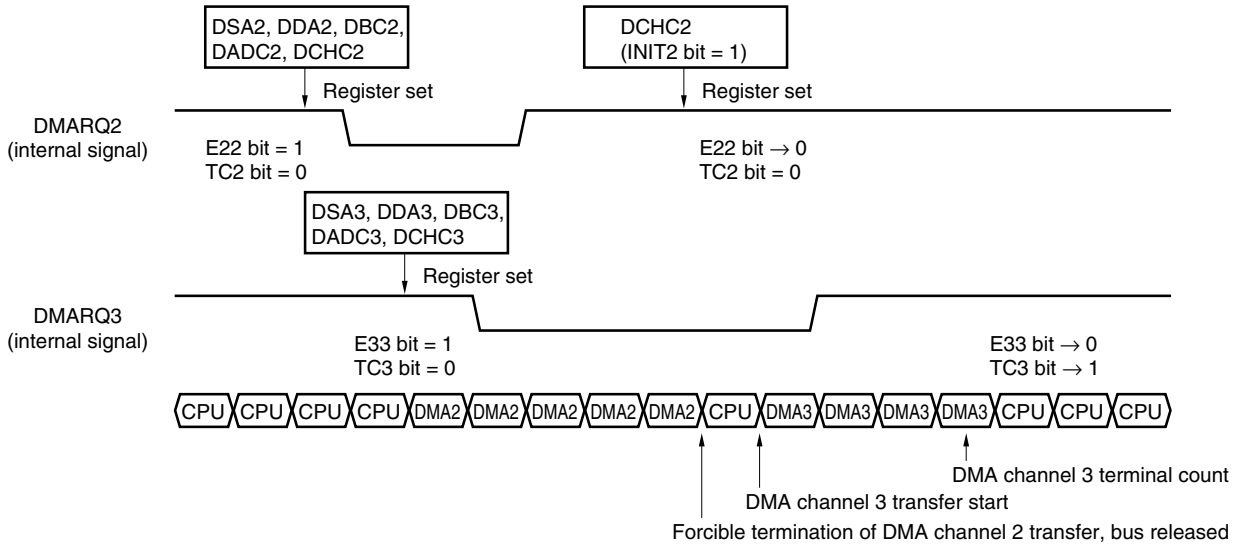
19.10 Forcible Termination

DMA transfer can be forcibly terminated by the DCHCn.INITn bit (n = 0 to 3).

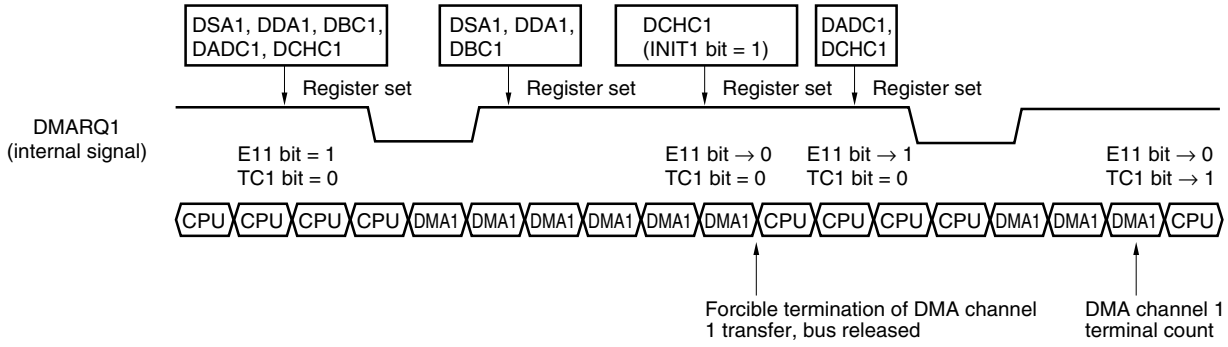
An example of forcible termination by the DCHCn.INITn bit is illustrated below (n = 0 to 3).

Figure 19-9. 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 **19.8 Next Address Setting Function**, **19.3.4 DMA addressing control registers 0 to 3 (DADC0 to DADC3)**, and **19.3.5 DMA channel control registers 0 to 3 (DCHC0 to DCHC3)**).

19.11 Times Related to DMA Transfer

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

Table 19-3. Number of Minimum 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 wait cycles specified by VSWC register

Notes 1. If an external interrupt (INTPn) is specified as the DMA transfer start factor, noise elimination time is added (n = 11 to 13, 15, 17, 18).

2. Two clocks for the DMA cycle

The minimum number of execution clocks during the DMA cycle in each mode is as follows.

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 always inserted between the read cycle and write cycle of DMA transfer.

19.12 Cautions

(1) Memory boundary

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

(4) 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.

(5) Timing of setting DCHCn.TCn bit

The DCHCn.TCn bit is usually set to 1 at the end of DMA transfer. In the case of DMA transfer that is initiated from the internal RAM, however, it is set 4 clocks after end of the last transfer (n = 0 to 3).

(6) Read values of DSAn and DDAn registers

If the values of the DSAn and DDAn registers are read during DMA transfer, 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

(7) CLR1, NOT1, and SET1 instructions

Write the CLR1, NOT1, and SET1 instructions after reading a register and then manipulating the target bit. To set the DCHCn.Enn bit to 1 by using the SET1 instruction, therefore, the TCn bit is cleared to 0 when the DCHCn.TCn bit = 1 (n = 0 to 3).

19.13 DMA Transfer End

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

CHAPTER 20 INTERRUPT/EXCEPTION PROCESSING FUNCTION

The V850E/IF3 and V850E/IG3 are provided with a dedicated interrupt controller (INTC) for interrupt servicing and can process a total of 89 to 96 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/IF3 and V850E/IG3 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).

20.1 Features

○ Interrupts
- Non-maskable interrupts: 1 source (external: none, internal: 1 source)
- Maskable interrupts (the number of maskable interrupt sources differs depending on the product)
  V850E/IF3: 88 sources (external: 15 sources, internal: 73 sources)
  V850E/IG3: 95 sources (external: 21 sources, internal: 74 sources)
- 8 levels of programmable priorities (maskable interrupts)
- Multiple interrupt control according to priority
- Masks can be specified for each maskable interrupt request.
- Noise elimination, edge detection, and valid edge specification for external interrupt request signals.

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

Interrupt sources are listed in Table 20-1.

Table 20-1. Interrupt Source List (1/4)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Control Register	Generating Source	Generating Unit				
Reset	Interrupt	RESET	–	$\overline{\text{RESET}}$ pin input	Pin	–	0000H	00000000H	Undefined
			–	WDT overflow (WDTRES)	WDT				
Non-maskable	Interrupt	INTWDT	–	WDT overflow	WDT		0010H	00000010H	nextPC
Software exception	Exception	TRAP0n[Note 1]	–	TRAP instruction	–	–	004nH	00000040H	nextPC
	Exception	TRAP1n[Note 1]	–	TRAP instruction	–	–	005nH	00000050H	nextPC
Exception trap	Exception	ILGOP/ DBG0	–	Illegal instruction code/ DBTRAP instruction	–	–	0060H	00000060H	nextPC
Maskable	Interrupt	INTLVIL	LVILIC	LVI low level voltage detection	LVI	0	0080H	00000080H	nextPC
	Interrupt	INTLVIH	LVIHIC	LVI high level voltage detection	LVI	1	0090H	00000090H	nextPC
	Interrupt	INTP00	PIC00	INTP00 pin valid edge input	Pin	2	00A0H	000000A0H	nextPC
	Interrupt	INTP01	PIC01	INTP01 pin valid edge input	Pin	3	00B0H	000000B0H	nextPC
	Interrupt	INTP02[Note 2]	PIC02[Note 2]	INTP02 pin valid edge input	Pin	4	00C0H	000000C0H	nextPC
	Interrupt	INTP03[Note 2]	PIC03[Note 2]	INTP03 pin valid edge input	Pin	5	00D0H	000000D0H	nextPC
	Interrupt	INTP04[Note 2]	PIC04[Note 2]	INTP04 pin valid edge input	Pin	6	00E0H	000000E0H	nextPC
	Interrupt	INTP05[Note 2]	PIC05[Note 2]	INTP05 pin valid edge input	Pin	7	00F0H	000000F0H	nextPC
	Interrupt	INTP06[Note 2]	PIC06[Note 2]	INTP06 pin valid edge input	Pin	8	0100H	00000100H	nextPC
	Interrupt	INTP07[Note 2]	PIC07[Note 2]	INTP07 pin valid edge input	Pin	9	0110H	00000110H	nextPC
	Interrupt	INTP08	PIC08	INTP08 pin valid edge input	Pin	10	0120H	00000120H	nextPC
	Interrupt	INTP09	PIC09	INTP09 pin valid edge input	Pin	11	0130H	00000130H	nextPC
	Interrupt	INTP10	PIC10	INTP10 pin valid edge input	Pin	12	0140H	00000140H	nextPC
	Interrupt	INTP11	PIC11	INTP11 pin valid edge input	Pin	13	0150H	00000150H	nextPC
	Interrupt	INTP12	PIC12	INTP12 pin valid edge input	Pin	14	0160H	00000160H	nextPC
	Interrupt	INTP13	PIC13	INTP13 pin valid edge input	Pin	15	0170H	00000170H	nextPC
	Interrupt	INTP14	PIC14	INTP14 pin valid edge input	Pin	16	0180H	00000180H	nextPC
	Interrupt	INTP15	PIC15	INTP15 pin valid edge input	Pin	17	0190H	00000190H	nextPC
	Interrupt	INTP16	PIC16	INTP16 pin valid edge input	Pin	18	01A0H	000001A0H	nextPC
	Interrupt	INTP17	PIC17	INTP17 pin valid edge input	Pin	19	01B0H	000001B0H	nextPC
	Interrupt	INTP18	PIC18	INTP18 pin valid edge input	Pin	20	01C0H	000001C0H	nextPC
	Interrupt	INTCMP0L	CMPIC0L	ADC0 overvoltage detection L (comparator output)	ADC0 (comparator)	21	01D0H	000001D0H	nextPC
	Interrupt	INTCMP0F	CMPIC0F	ADC0 overvoltage detection F (comparator output)	ADC0 (comparator)	22	01E0H	000001E0H	nextPC
	Interrupt	INTCMP1L	CMPIC1L	ADC1 overvoltage detection L (comparator output)	ADC1 (comparator)	23	01F0H	000001F0H	nextPC
	Interrupt	INTCMP1F	CMPIC1F	ADC1 overvoltage detection F (comparator output)	ADC1 (comparator)	24	0200H	00000200H	nextPC

Notes 1. n = 0 to FH

2. V850E/IG3 only

Table 20-1. Interrupt Source List (2/4)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Control Register	Generating Source	Generating Unit				
Maskable	Interrupt	INTTB0OV	TB0OVIC	TAB0 overflow[Note 2]	TAB0	25	0210H	00000210H	nextPC
	Interrupt	INTTB0CC0	TB0CCIC0	TAB0CCR0 capture input/ compare match[Note 3]	TAB0	26	0220H	00000220H	nextPC
	Interrupt	INTTB0CC1	TB0CCIC1	TAB0CCR1 capture input/ compare match	TAB0	27	0230H	00000230H	nextPC
	Interrupt	INTTB0CC2	TB0CCIC2	TAB0CCR2 capture input/ compare match	TAB0	28	0240H	00000240H	nextPC
	Interrupt	INTTB0CC3	TB0CCIC3	TAB0CCR3 capture input/ compare match	TAB0	29	0250H	00000250H	nextPC
	Interrupt	INTTB1OV	TB1OVIC	TAB1 overflow[Note 2]	TAB1	30	0260H	00000260H	nextPC
	Interrupt	INTTB1CC0	TB1CCIC0	TAB1CCR0 capture input/ compare match[Note 3]	TAB1	31	0270H	00000270H	nextPC
	Interrupt	INTTB1CC1	TB1CCIC1	TAB1CCR1 capture input/ compare match	TAB1	32	0280H	00000280H	nextPC
	Interrupt	INTTB1CC2	TB1CCIC2	TAB1CCR2 capture input/ compare match	TAB1	33	0290H	00000290H	nextPC
	Interrupt	INTTB1CC3	TB1CCIC3	TAB1CCR3 capture input/ compare match	TAB1	34	02A0H	000002A0H	nextPC
	Interrupt	INTTTIOV0	TT0OVIC	TMT0 overflow	TMT0	35	02B0H	000002B0H	nextPC
	Interrupt	INTTTEQC00	TT0CCIC0	TT0CCR0 capture input[Note 4]/ compare match	TMT0	36	02C0H	000002C0H	nextPC
	Interrupt	INTTTEQC01	TT0CCIC1	TT0CCR1 capture input[Note 4]/ compare match	TMT0	37	02D0H	000002D0H	nextPC
	Interrupt	INTTIEC0[Note 1]	TT0IECIC[Note 1]	Encoder input interrupt 0	TMT0	38	02E0H	000002E0H	nextPC
	Interrupt	INTTTIOV1	TT1OVIC	TMT1 overflow	TMT1	39	02F0H	000002F0H	nextPC
	Interrupt	INTTTEQC10	TT1CCIC0	TT1CCR0 capture input/ compare match	TMT1	40	0300H	00000300H	nextPC
	Interrupt	INTTTEQC11	TT1CCIC1	TT1CCR1 capture input/ compare match	TMT1	41	0310H	00000310H	nextPC
	Interrupt	INTTIEC1	TT1IECIC	Encoder input interrupt 1	TMT1	42	0320H	00000320H	nextPC
	Interrupt	INTTA0OV	TA0OVIC	TAA0 overflow	TAA0	43	0330H	00000330H	nextPC

Notes 1. V850E/IG3 only

2. When TABm is used in the 6-phase PWM output mode, it functions as INTTBmOV (trough interrupt) from the TMQm option (TMQOPm) (m = 0, 1).

3. When TABm is used in the 6-phase PWM output mode, it functions as INTTBmCC0 (peak interrupt) from the TMQm option (TMQOPm) (m = 0, 1).

4. V850E/IG3 only
In the V850E/IF3, compare match only

Table 20-1. Interrupt Source List (3/4)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Control Register	Generating Source	Generating Unit				
Maskable	Interrupt	INTTA0CC0	TA0CCIC0	TA0CCR0 compare match	TAA0	44	0340H	00000340H	nextPC
	Interrupt	INTTA0CC1	TA0CCIC1	TA0CCR1 compare match	TAA0	45	0350H	00000350H	nextPC
	Interrupt	INTTA1OV	TA1OVIC	TAA1 overflow	TAA1	46	0360H	00000360H	nextPC
	Interrupt	INTTA1CC0	TA1CCIC0	TA1CCR0 compare match	TAA1	47	0370H	00000370H	nextPC
	Interrupt	INTTA1CC1	TA1CCIC1	TA1CCR1 compare match	TAA1	48	0380H	00000380H	nextPC
	Interrupt	INTTA2OV	TA2OVIC	TAA2 overflow	TAA2	49	0390H	00000390H	nextPC
	Interrupt	INTTA2CC0	TA2CCIC0	TA2CCR0 capture input/ compare match	TAA2	50	03A0H	000003A0H	nextPC
	Interrupt	INTTA2CC1	TA2CCIC1	TA2CCR1 capture input/ compare match	TAA2	51	03B0H	000003B0H	nextPC
	Interrupt	INTTA3OV	TA3OVIC	TAA3 overflow	TAA3	52	03C0H	000003C0H	nextPC
	Interrupt	INTTA3CC0	TA3CCIC0	TA3CCR0 capture input[Note]/ compare match	TAA3	53	03D0H	000003D0H	nextPC
	Interrupt	INTTA3CC1	TA3CCIC1	TA3CCR1 capture input[Note]/ compare match	TAA3	54	03E0H	000003E0H	nextPC
	Interrupt	INTTA4OV	TA4OVIC	TAA4 overflow	TAA4	55	03F0H	000003F0H	nextPC
	Interrupt	INTTA4CC0	TA4CCIC0	TA4CCR0 capture input/ compare match	TAA4	56	0400H	00000400H	nextPC
	Interrupt	INTTA4CC1	TA4CCIC1	TA4CCR1 capture input/ compare match	TAA4	57	0410H	00000410H	nextPC
	Interrupt	INTDMA0	DMAIC0	DMA channel 0 transfer end	DMA0	58	0420H	00000420H	nextPC
	Interrupt	INTDMA1	DMAIC1	DMA channel 1 transfer end	DMA1	59	0430H	00000430H	nextPC
	Interrupt	INTDMA2	DMAIC2	DMA channel 2 transfer end	DMA2	60	0440H	00000440H	nextPC
	Interrupt	INTDMA3	DMAIC3	DMA channel 3 transfer end	DMA3	61	0450H	00000450H	nextPC
	Interrupt	INTUBTIRE	UREIC	UARTB reception error	UARTB	62	0460H	00000460H	nextPC
	Interrupt	INTUBTIR	URIC	UARTB reception end	UARTB	63	0470H	00000470H	nextPC
	Interrupt	INTUBTIT	UTIC	UARTB transmission end	UARTB	64	0480H	00000480H	nextPC
	Interrupt	INTUBTIF	UIFIC	UARTB FIFO transmission end	UARTB	65	0490H	00000490H	nextPC
	Interrupt	INTUBTITO	UTOIC	UARTB reception timeout	UARTB	66	04A0H	000004A0H	nextPC
	Interrupt	INTUA0RE	UA0REIC	UARTA0 reception error	UARTA0	67	04B0H	000004B0H	nextPC
	Interrupt	INTUA0R	UA0RIC	UARTA0 reception end	UARTA0	68	04C0H	000004C0H	nextPC
	Interrupt	INTUA0T	UA0TIC	UARTA0 transmission enable	UARTA0	69	04D0H	000004D0H	nextPC
	Interrupt	INTCB0RE	CB0REIC	CSIB0 reception error	CSIB0	70	04E0H	000004E0H	nextPC
	Interrupt	INTCB0R	CB0RIC	CSIB0 reception end	CSIB0	71	04F0H	000004F0H	nextPC
	Interrupt	INTCB0T	CB0TIC	CSIB0 transmission enable	CSIB0	72	0500H	00000500H	nextPC
	Interrupt	INTUA1RE	UA1REIC	UARTA1 reception error	UARTA1	73	0510H	00000510H	nextPC
	Interrupt	INTUA1R	UA1RIC	UARTA1 reception end	UARTA1	74	0520H	00000520H	nextPC
	Interrupt	INTUA1T	UA1TIC	UARTA1 transmission enable	UARTA1	75	0530H	00000530H	nextPC

Note V850E/IG3 only
In the V850E/IF3, compare match only

Table 20-1. Interrupt Source List (4/4)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Control Register	Generating Source	Generating Unit				
Maskable	Interrupt	INTCB1RE	CB1REIC	CSIB1 reception error	CSIB1	76	0540H	00000540H	nextPC
	Interrupt	INTCB1R	CB1RIC	CSIB1 reception end	CSIB1	77	0550H	00000550H	nextPC
	Interrupt	INTCB1T	CB1TIC	CSIB1 transmission enable	CSIB1	78	0560H	00000560H	nextPC
	Interrupt	INTUA2RE	UA2REIC	UARTA2 reception error	UARTA2	79	0570H	00000570H	nextPC
	Interrupt	INTUA2R	UA2RIC	UARTA2 reception end	UARTA2	80	0580H	00000580H	nextPC
	Interrupt	INTUA2T	UA2TIC	UARTA2 transmission enable	UARTA2	81	0590H	00000590H	nextPC
	Interrupt	INTCB2RE	CB2REIC	CSIB2 reception error	CSIB2	82	05A0H	000005A0H	nextPC
	Interrupt	INTCB2R	CB2RIC	CSIB2 reception end	CSIB2	83	05B0H	000005B0H	nextPC
	Interrupt	INTCB2T	CB2TIC	CSIB2 transmission enable	CSIB2	84	05C0H	000005C0H	nextPC
	Interrupt	INTIIC	IICIC	IIC serial transfer end	IIC	85	05D0H	000005D0H	nextPC
	Interrupt	INTAD0	AD0IC	ADC0 conversion end	ADC0	86	05E0H	000005E0H	nextPC
	Interrupt	INTAD1	AD1IC	ADC1 conversion end	ADC1	87	05F0H	000005F0H	nextPC
	Interrupt	INTAD2	AD2IC	ADC2 conversion end	ADC2	88	0600H	00000600H	nextPC
	Interrupt	INTTM0EQ0	TM0EQIC0	TM0CMP0 compare match	TMM0	89	0610H	00000610H	nextPC
	Interrupt	INTTM1EQ0	TM1EQIC0	TM0CMP1 compare match	TMM1	90	0620H	00000620H	nextPC
	Interrupt	INTTM2EQ0	TM2EQIC0	TM0CMP2 compare match	TMM2	91	0630H	00000630H	nextPC
	Interrupt	INTTM3EQ0	TM3EQIC0	TM0CMP3 compare match	TMM3	92	0640H	00000640H	nextPC
	Interrupt	INTADT0	ADT0IC	ADTRG0 pin valid edge input	Pin	93	0650H	00000650H	nextPC
	Interrupt	INTADT1	ADT1IC	ADTRG1 pin valid edge input	Pin	94	0660H	00000660H	nextPC

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

Restored PC: The value of the program counter (PC) saved to EIPC, FEPC, or DBPC 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 interrupt 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).

20.2 Non-Maskable Interrupts

A non-maskable interrupt request signal 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 interrupt request signals.

The non-maskable interrupt signals of the V850E/IF3 and V850E/IG3 are the non-maskable interrupt request signals generated by the overflow of the watchdog timer (INTWDT).

INTWDT functions when the WDTM.WDM1 and WDTM.WDM0 bits are set to “01”.

20.2.1 Operation

If a non-maskable interrupt request signal (INTWDT) 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 the exception code (0010H) to the higher halfword (FECC) of ECR.
(4) Sets the PSW.NP and PSW.ID bits (1) and clears the PSW.EP bit (0).
(5) Loads the handler address (00000010H) of the non-maskable interrupt routine to the PC, and transfers control.

The following shows the non-maskable interrupt servicing.

Figure 20-1. Non-Maskable Interrupt Servicing

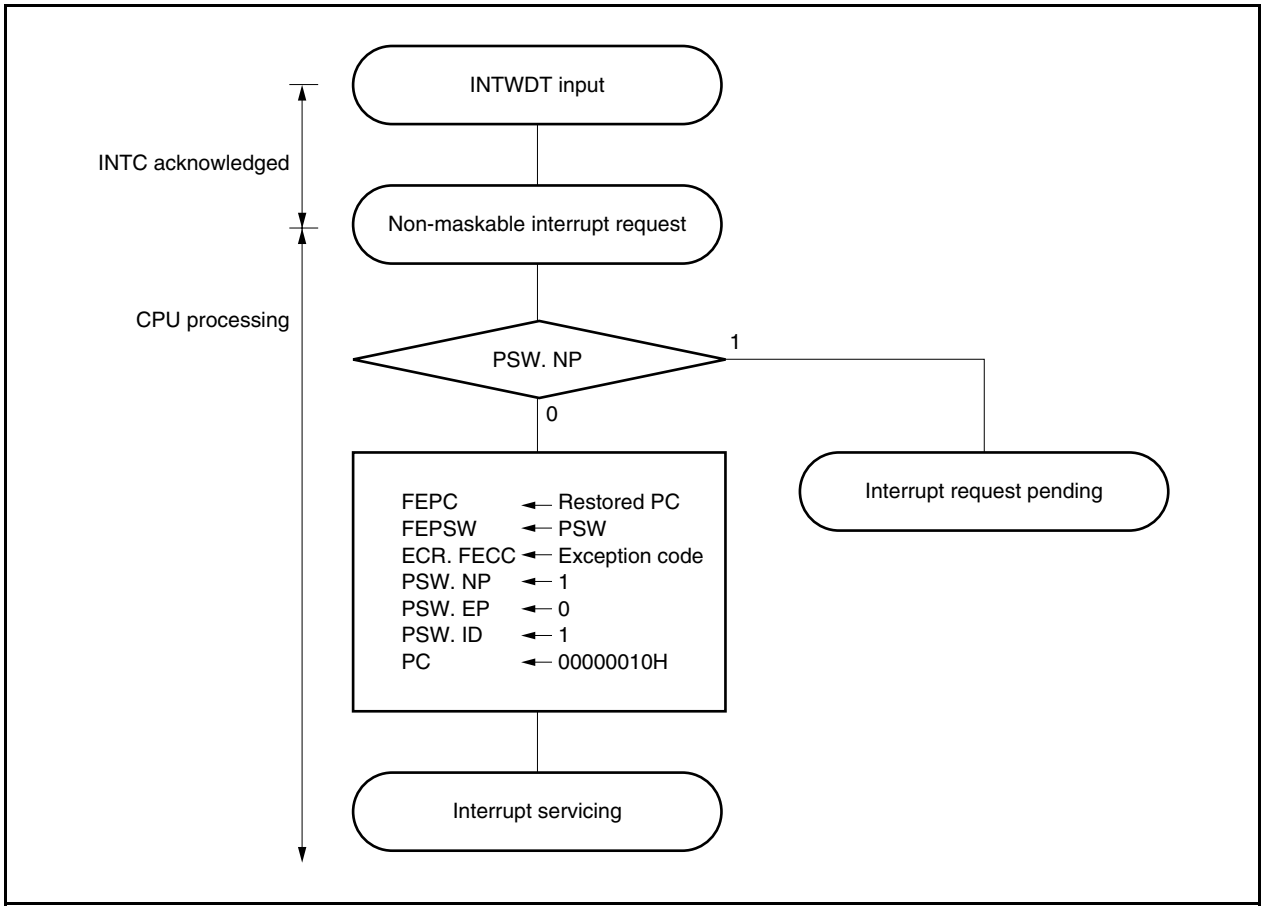
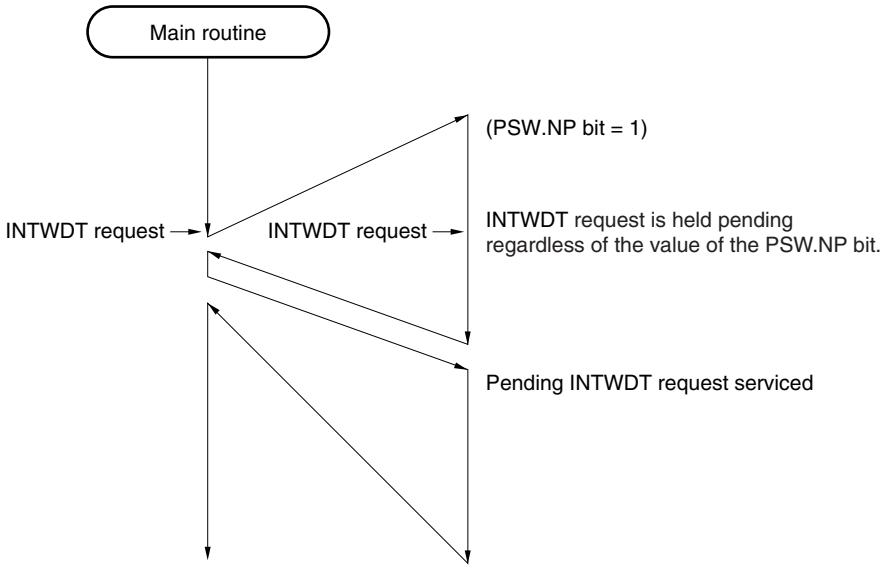
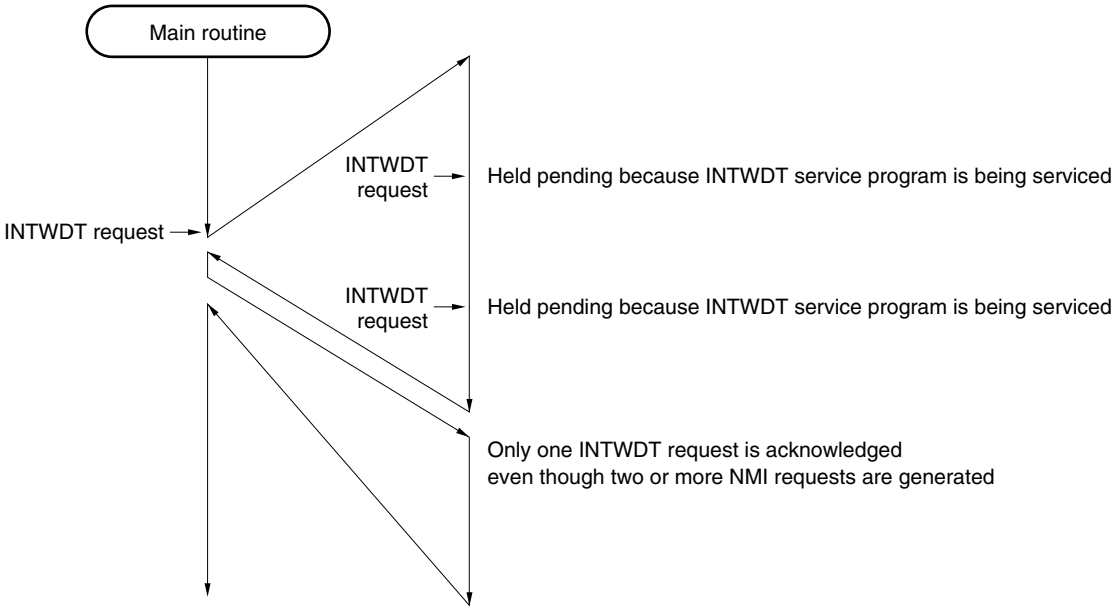


Figure 20-2. Acknowledging Non-Maskable Interrupt Request

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



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



20.2.2 Restore

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

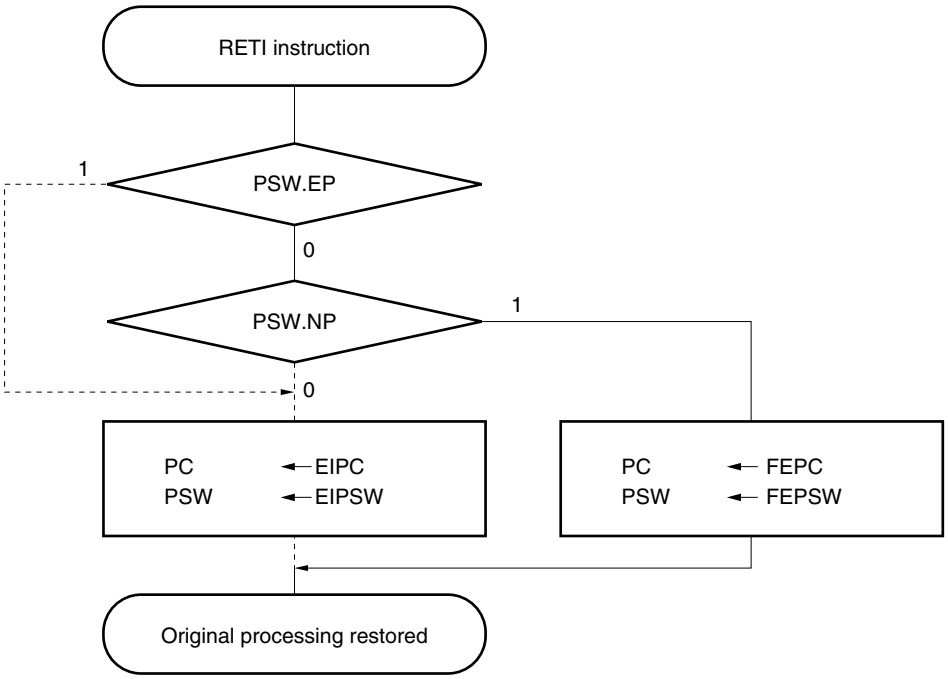
When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

<1> Loads the restored PC and PSW from FEPC and FEPSW because the PSW.EP bit is 0 and the PSW.NP bit is 1.

<2> Transfers control back to the address of the restored PC and PSW.

The following illustrates how the RETI instruction is processed.

Figure 20-3. RETI Instruction Processing



Caution When the EP and NP bits are changed by the LDSR instruction during non-maskable interrupt servicing, in order to restore the PC and PSW correctly during restoring by the RETI instruction, it is necessary to set EP back to 0 and NP back to 1 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

20.2.3 Non-maskable interrupt status flag (NP)

The NP flag is a status flag that indicates that non-maskable interrupt (INTWDT) servicing is in progress. The NP flag is allocated to the PSW.

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

The flag is cleared to 00000020H after reset.

After reset: 00000020H

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

NP	Non-maskable interrupt (INTWDT) servicing status
0	No non-maskable interrupt servicing
1	Non-maskable interrupt servicing in progress

20.3 Maskable Interrupts

Maskable interrupt request signals can be masked by interrupt control registers. The V850E/IF3 and V850E/IG3 have 95 maskable interrupt sources.

If two or more maskable interrupt request signals 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 signal has been acknowledged, the acknowledgment of other maskable interrupt request signals is disabled and the interrupt disabled (DI) status is set.

When the EI instruction is executed in an interrupt service routine, the interrupt enabled (EI) status is set, which enables servicing of interrupts having a higher priority than the interrupt request signal 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 serviced as multiple interrupts.

To enable multiple interrupt servicing, however, save EIPC and EIPSW to memory or registers before executing the EI instruction, and execute the DI instruction before the RETI instruction to restore the original values of EIPC and EIPSW.

20.3.1 Operation

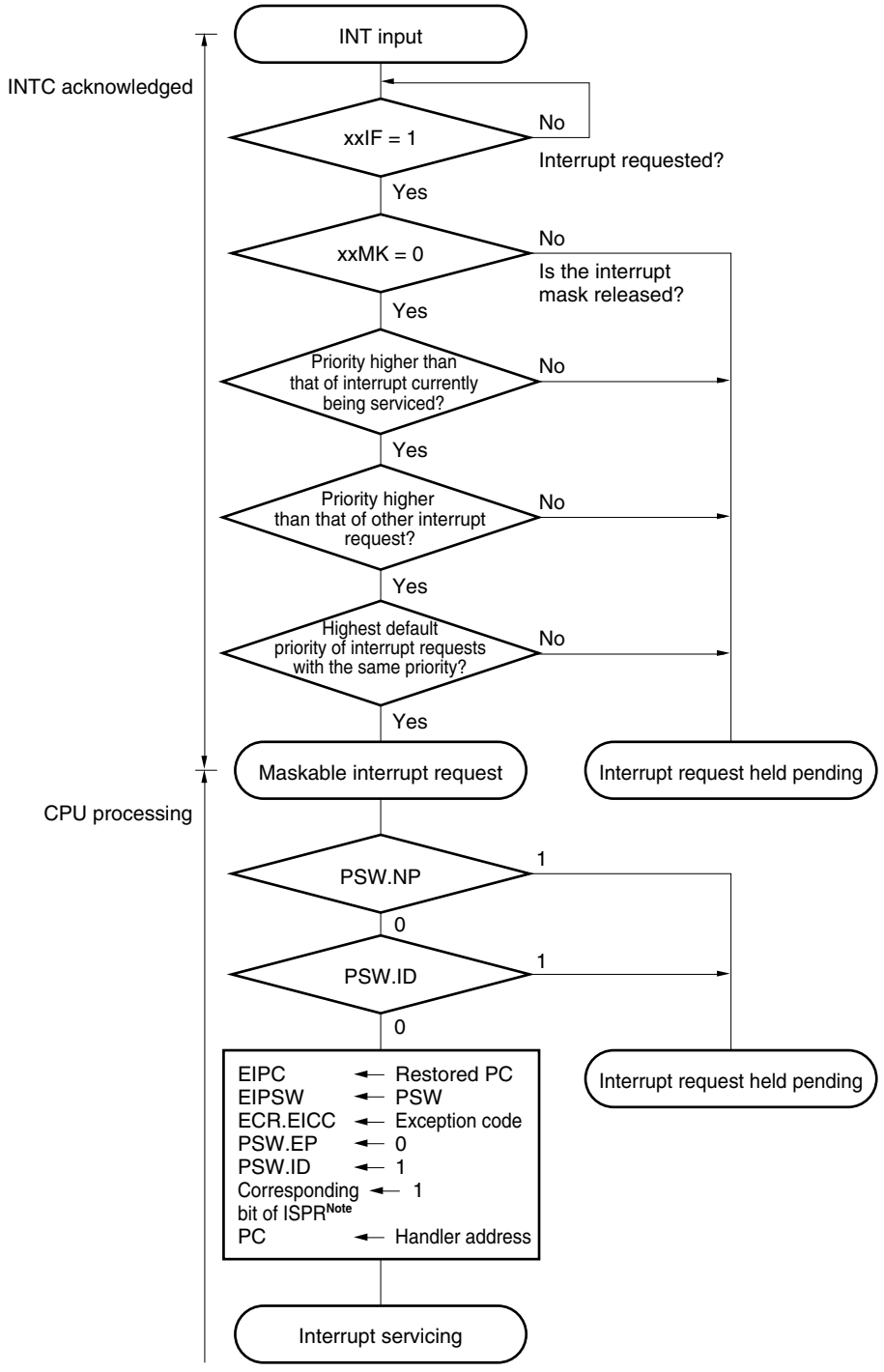
If a maskable interrupt occurs, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower halfword of ECR (EICC).
<4> Sets the PSW.ID bit to 1 and clears the PSW.EP bit to 0.
<5> Sets the handler address corresponding to each interrupt to the PC, and transfers control.

The maskable interrupt request signal masked by interrupt controller (INTC) and the maskable interrupt request signal generated while another interrupt is being serviced (while PSW.NP bit = 1 or ID bit = 1) are held pending inside the INTC. In this case, servicing a new maskable interrupt is started in accordance with the priority of the pending maskable interrupt request signal if either the maskable interrupt is unmasked or NP and ID bits are cleared to 0 by using the RETI or LDSR instruction.

How maskable interrupts are serviced is illustrated below.

Figure 20-4. Maskable Interrupt Servicing



Note For details of the ISPR register, see **20.3.6 In-service priority register (ISPR)**.

20.3.2 Restore

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

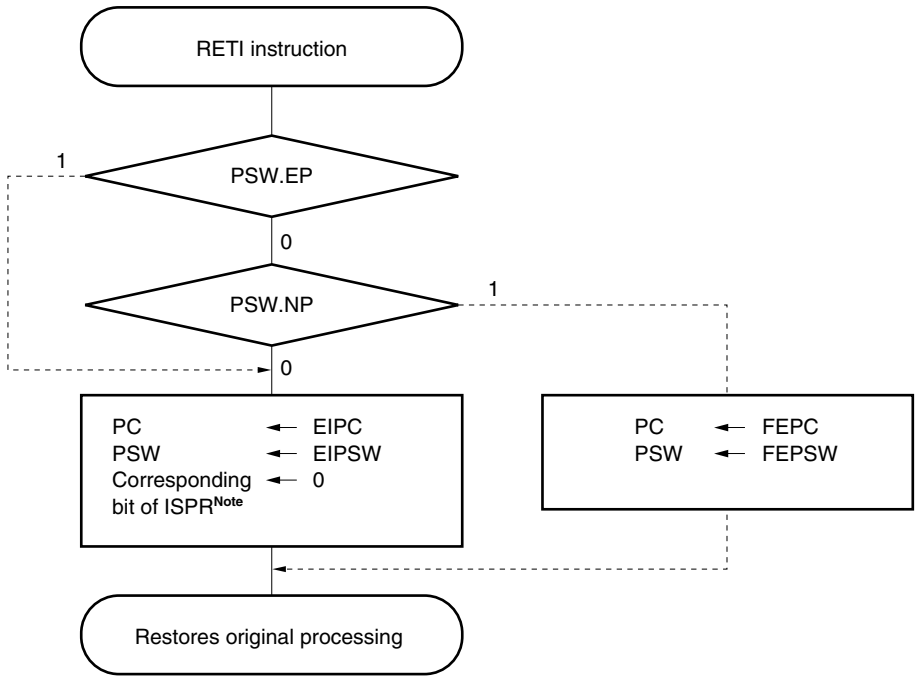
When the RETI instruction is executed, the CPU performs the following steps, and transfers control to the address of the restored PC.

<1> Loads the values of the PC and the PSW from EIPC and EIPSW because the PSW.EP bit is 0 and the PSW.NP bit is 0.

<2> Transfers control to the address of the restored PC and PSW.

The processing of the RETI instruction is shown below.

Figure 20-5. RETI Instruction Processing



Note For the ISPR register, see **20.3.6 In-service priority register (ISPR)**.

Caution When the EP and NP bits 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 EP back to 0 and NP back to 1 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

20.3.3 Priorities of maskable interrupts

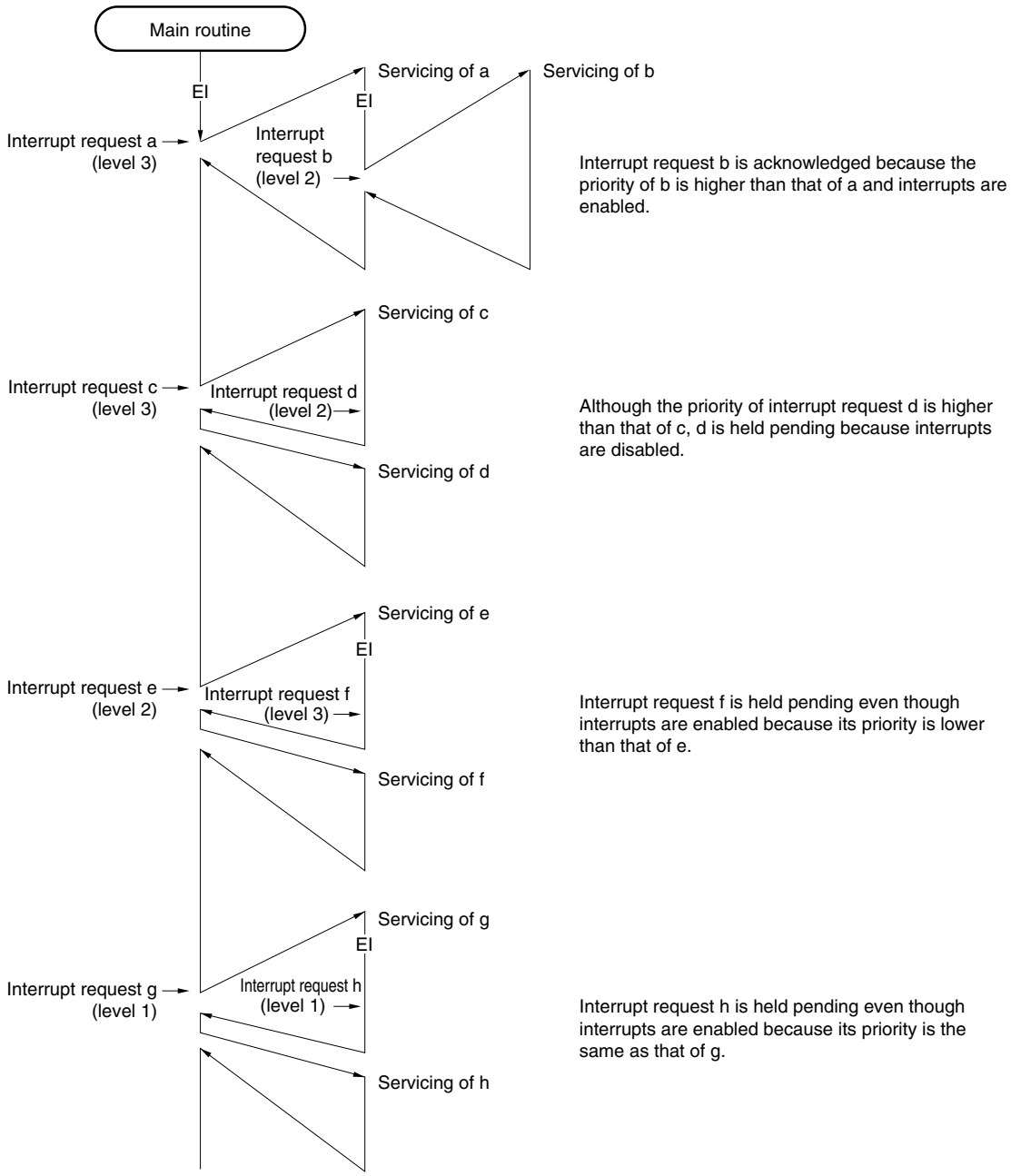
The INTC provides multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced. Multiple interrupts can be controlled by priority levels.

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 signal type (default priority level) beforehand. For more information, see **Table 20-1 Interrupt Source List**. Programmable priority control customizes interrupt request signals into eight levels by the setting of the priority level specification flag.

Note that when an interrupt request signal is acknowledged, the PSW.ID flag 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 servicing program) to set the interrupt enabled mode.

Remark xx: Identification name of each peripheral unit (see **Table 20-2**)
n: Peripheral unit number (see **Table 20-2**)

Figure 20-6. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (1/2)

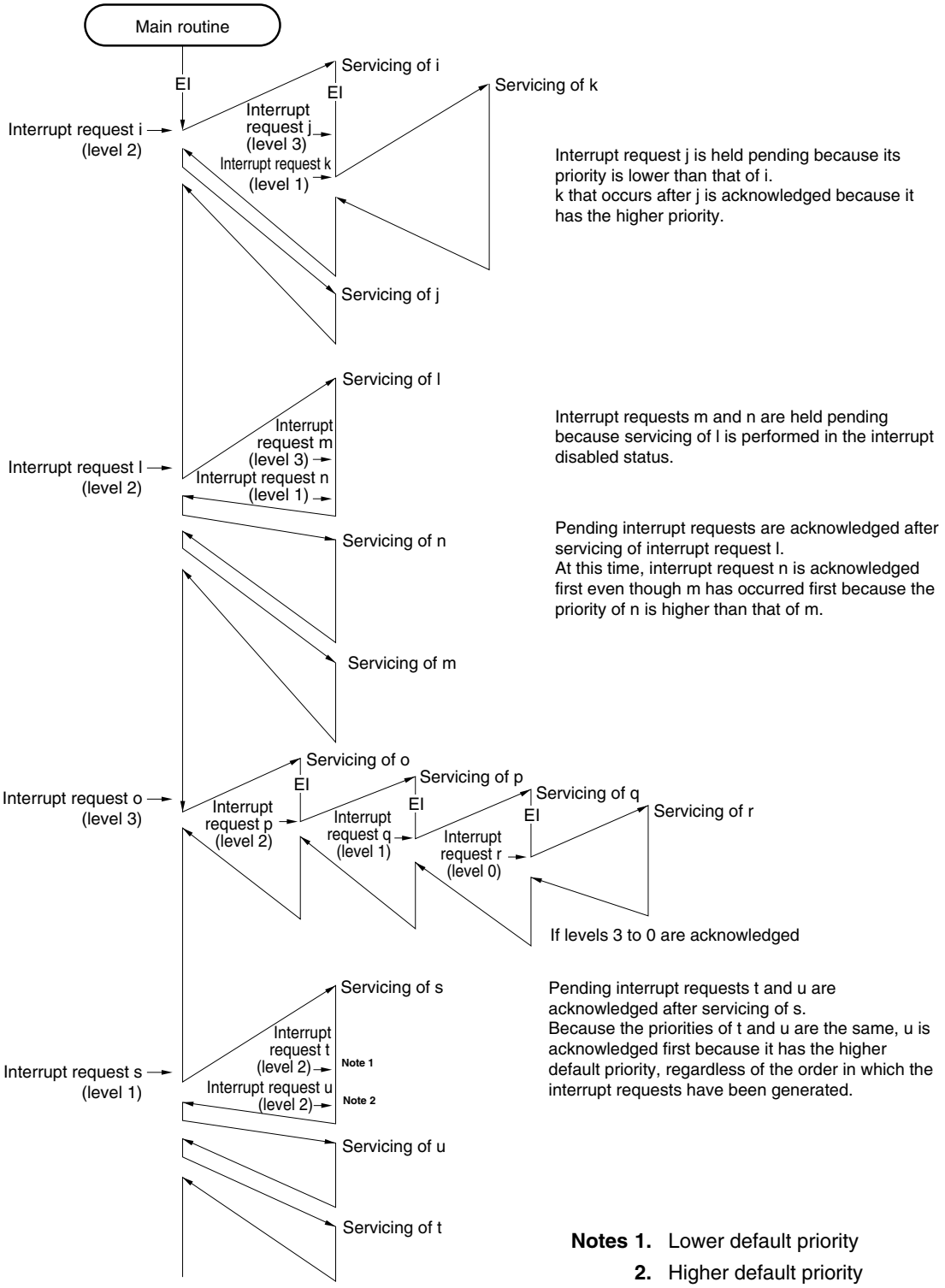


Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Remarks 1. a to u in the figure are the temporary names of interrupt request signals shown for the sake of explanation.

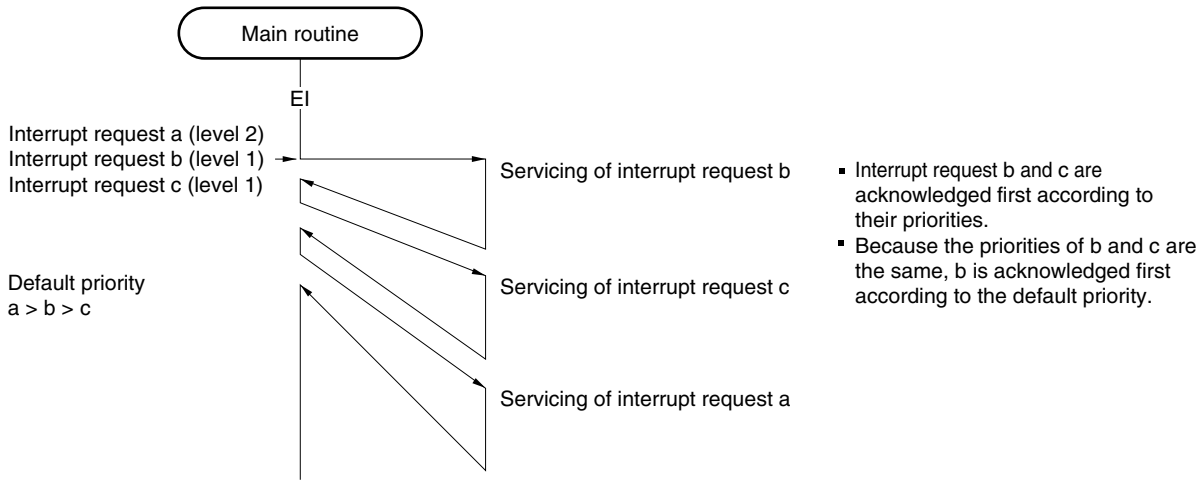
2. The default priority in the figure indicates the relative priority between two interrupt request signals.

Figure 20-6. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (2/2)



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

Figure 20-7. Example of Servicing Interrupt Request Signals Generated Simultaneously



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

Remarks 1. a to c in the figure are assumed names given to interrupt request signals for the sake of explanation.

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

20.3.4 Interrupt control registers (xxICn)

An xxICn register is assigned to each interrupt request signal (maskable interrupt) and sets the control conditions for each maskable interrupt request.

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

Reset sets these registers to 47H.

Cautions 1. Disable interrupts (DI) to read the xxICn.xxIFn bit. If the xxIFn bit is read while interrupts are enabled (EI), the correct value may not be read when acknowledging an interrupt and reading the bit conflict.

2. If generation of an interrupt source and a bit manipulation instruction (SET1, NOT1, or CLR1 (except TST1)) that manipulates the xxMKn or xxPRn2 to xxPRn0 bits of the interrupt source that has been generated conflict, the interrupt request signal may not be generated.
This can be avoided in the following two ways.

- **When a bit manipulation instruction is not used to the xxICn register**
  - **<1> Change from writing the xxMKn bit to a bit manipulation instruction that manipulates the IMRm register.**
  - **<2> Change from writing the xxPRn2 to xxPRn0 bits to a byte access to the xxICn register.**

- **When a bit manipulation instruction is used to the xxICn register**
  Execute a bit manipulation instruction that manipulates the xxICn register after executing a dummy write (byte access) with the unused xxICn.xxIFn bit cleared to 0 in the interrupt disabled (DI) status.

After reset: 47H R/W Address: FFFFF110H to FFFFF1CCH

	<7>	<6>	5	4	3	2	1	0
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0

xxIFn	Interrupt request flag[Note]
0	Interrupt request not issued
1	Interrupt request issued

xxMKn	Interrupt mask flag
0	Interrupt servicing enabled
1	Interrupt servicing disabled (pending)

xxPRn2	xxPRn1	xxPRn0	Interrupt priority specification bit
0	0	0	Specifies level 0 (highest).
0	0	1	Specifies level 1.
0	1	0	Specifies level 2.
0	1	1	Specifies level 3.
1	0	0	Specifies level 4.
1	0	1	Specifies level 5.
1	1	0	Specifies level 6.
1	1	1	Specifies level 7 (lowest).

Note The flag xxIFn is reset automatically by the hardware if an interrupt request signal is acknowledged.

Remark xx: Identification name of each peripheral unit (see **Table 20-2**)
n: Peripheral unit number (see **Table 20-2**)

The addresses and bits of the interrupt control registers are as follows.

Table 20-2. Addresses and Bits of Interrupt Control Registers (1/3)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF110H	LVILIC	LVILIF	LVILMK	0	0	0	LVILPR2	LVILPR1	LVILPR0
FFFFF112H	LVIHIC	LVIHIF	LVIHMK	0	0	0	LVIHPR2	LVIHPR1	LVIHPR0
FFFFF114H	PIC00	PIF00	PMK00	0	0	0	PPR002	PPR001	PPR000
FFFFF116H	PIC01	PIF01	PMK01	0	0	0	PPR012	PPR011	PPR010
FFFFF118H	PIC02Note	PIF02	PMK02	0	0	0	PPR022	PPR021	PPR020
FFFFF11AH	PIC03Note	PIF03	PMK03	0	0	0	PPR032	PPR031	PPR030
FFFFF11CH	PIC04Note	PIF04	PMK04	0	0	0	PPR042	PPR041	PPR040
FFFFF11EH	PIC05Note	PIF05	PMK05	0	0	0	PPR052	PPR051	PPR050
FFFFF120H	PIC06Note	PIF06	PMK06	0	0	0	PPR062	PPR061	PPR060
FFFFF122H	PIC07Note	PIF07	PMK07	0	0	0	PPR072	PPR071	PPR070
FFFFF124H	PIC08	PIF08	PMK08	0	0	0	PPR082	PPR081	PPR080
FFFFF126H	PIC09	PIF09	PMK09	0	0	0	PPR092	PPR091	PPR090
FFFFF128H	PIC10	PIF10	PMK10	0	0	0	PPR102	PPR101	PPR100
FFFFF12AH	PIC11	PIF11	PMK11	0	0	0	PPR112	PPR111	PPR110
FFFFF12CH	PIC12	PIF12	PMK12	0	0	0	PPR122	PPR121	PPR120
FFFFF12EH	PIC13	PIF13	PMK13	0	0	0	PPR132	PPR131	PPR130
FFFFF130H	PIC14	PIF14	PMK14	0	0	0	PPR142	PPR141	PPR140
FFFFF132H	PIC15	PIF15	PMK15	0	0	0	PPR152	PPR151	PPR150
FFFFF134H	PIC16	PIF16	PMK16	0	0	0	PPR162	PPR161	PPR160
FFFFF136H	PIC17	PIF17	PMK17	0	0	0	PPR172	PPR171	PPR170
FFFFF138H	PIC18	PIF18	PMK18	0	0	0	PPR182	PPR181	PPR180
FFFFF13AH	CMPIC0L	CMPIF0L	CMPMK0L	0	0	0	CMPPR0L2	CMPPR0L1	CMPPR0L0
FFFFF13CH	CMPIC0F	CMPIF0F	CMPMK0F	0	0	0	CMPPR0F2	CMPPR0F1	CMPPR0F0
FFFFF13EH	CMPIC1L	CMPIF1L	CMPMK1L	0	0	0	CMPPR1L2	CMPPR1L1	CMPPR1L0
FFFFF140H	CMPIC1F	CMPIF1F	CMPMK1F	0	0	0	CMPPR1F2	CMPPR1F1	CMPPR1F0
FFFFF142H	TB0OVIC	TB0OVIF	TB0OVMK	0	0	0	TB0OVPR2	TB0OVPR1	TB0OVPR0
FFFFF144H	TB0CCIC0	TB0CCIF0	TB0CCMK0	0	0	0	TB0CCPR02	TB0CCPR01	TB0CCPR00
FFFFF146H	TB0CCIC1	TB0CCIF1	TB0CCMK1	0	0	0	TB0CCPR12	TB0CCPR11	TB0CCPR10
FFFFF148H	TB0CCIC2	TB0CCIF2	TB0CCMK2	0	0	0	TB0CCPR22	TB0CCPR21	TB0CCPR20
FFFFF14AH	TB0CCIC3	TB0CCIF3	TB0CCMK3	0	0	0	TB0CCPR32	TB0CCPR31	TB0CCPR30
FFFFF14CH	TB1OVIC	TB1OVIF	TB1OVMK	0	0	0	TB1OVPR2	TB1OVPR1	TB1OVPR0
FFFFF14EH	TB1CCIC0	TB1CCIF0	TB1CCMK0	0	0	0	TB1CCPR02	TB1CCPR01	TB1CCPR00
FFFFF150H	TB1CCIC1	TB1CCIF1	TB1CCMK1	0	0	0	TB1CCPR12	TB1CCPR11	TB1CCPR10
FFFFF152H	TB1CCIC2	TB1CCIF2	TB1CCMK2	0	0	0	TB1CCPR22	TB1CCPR21	TB1CCPR20
FFFFF154H	TB1CCIC3	TB1CCIF3	TB1CCMK3	0	0	0	TB1CCPR32	TB1CCPR31	TB1CCPR30
FFFFF156H	TT0OVIC	TT0OVIF	TT0OVMK	0	0	0	TT0OVPR2	TT0OVPR1	TT0OVPR0
FFFFF158H	TT0CCIC0	TT0CCIF0	TT0CCMK0	0	0	0	TT0CCPR02	TT0CCPR01	TT0CCPR00
FFFFF15AH	TT0CCIC1	TT0CCIF1	TT0CCMK1	0	0	0	TT0CCPR12	TT0CCPR11	TT0CCPR10
FFFFF15CH	TT0IECICNote	TT0IECIF	TT0IECMK	0	0	0	TT0IECPR2	TT0IECPR1	TT0IECPR0
FFFFF15EH	TT1OVIC	TT1OVIF	TT1OVMK	0	0	0	TT1OVPR2	TT1OVPR1	TT1OVPR0
FFFFF160H	TT1CCIC0	TT1CCIF0	TT1CCMK0	0	0	0	TT1CCPR02	TT1CCPR01	TT1CCPR00
FFFFF162H	TT1CCIC1	TT1CCIF1	TT1CCMK1	0	0	0	TT1CCPR12	TT1CCPR11	TT1CCPR10

Note V850E/IG3 only

Table 20-2. Addresses and Bits of Interrupt Control Registers (2/3)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF164H	TT1IECIC	TT1IECIF	TT1IECMK	0	0	0	TT1IECPR2	TT1IECPR1	TT1IECPR0
FFFFF166H	TA0OVIC	TA0OVIF	TA0OVMK	0	0	0	TA0OVPR2	TA0OVPR1	TA0OVPR0
FFFFF168H	TA0CCIC0	TA0CCIF0	TA0CCMK0	0	0	0	TA0CCPR02	TA0CCPR01	TA0CCPR00
FFFFF16AH	TA0CCIC1	TA0CCIF1	TA0CCMK1	0	0	0	TA0CCPR12	TA0CCPR11	TA0CCPR10
FFFFF16CH	TA1OVIC	TA1OVIF	TA1OVMK	0	0	0	TA1OVPR2	TA1OVPR1	TA1OVPR0
FFFFF16EH	TA1CCIC0	TA1CCIF0	TA1CCMK0	0	0	0	TA1CCPR02	TA1CCPR01	TA1CCPR00
FFFFF170H	TA1CCIC1	TA1CCIF1	TA1CCMK1	0	0	0	TA1CCPR12	TA1CCPR11	TA1CCPR10
FFFFF172H	TA2OVIC	TA2OVIF	TA2OVMK	0	0	0	TA2OVPR2	TA2OVPR1	TA2OVPR0
FFFFF174H	TA2CCIC0	TA2CCIF0	TA2CCMK0	0	0	0	TA2CCPR02	TA2CCPR01	TA2CCPR00
FFFFF176H	TA2CCIC1	TA2CCIF1	TA2CCMK1	0	0	0	TA2CCPR12	TA2CCPR11	TA2CCPR10
FFFFF178H	TA3OVIC	TA3OVIF	TA3OVMK	0	0	0	TA3OVPR2	TA3OVPR1	TA3OVPR0
FFFFF17AH	TA3CCIC0	TA3CCIF0	TA3CCMK0	0	0	0	TA3CCPR02	TA3CCPR01	TA3CCPR00
FFFFF17CH	TA3CCIC1	TA3CCIF1	TA3CCMK1	0	0	0	TA3CCPR12	TA3CCPR11	TA3CCPR10
FFFFF17EH	TA4OVIC	TA4OVIF	TA4OVMK	0	0	0	TA4OVPR2	TA4OVPR1	TA4OVPR0
FFFFF180H	TA4CCIC0	TA4CCIF0	TA4CCMK0	0	0	0	TA4CCPR02	TA4CCPR01	TA4CCPR00
FFFFF182H	TA4CCIC1	TA4CCIF1	TA4CCMK1	0	0	0	TA4CCPR12	TA4CCPR11	TA4CCPR10
FFFFF184H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFF186H	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF188H	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFF18AH	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30
FFFFF18CH	UREIC	UREIF	UREMK	0	0	0	UREPR2	UREPR1	UREPR0
FFFFF18EH	URIC	URIF	URMK	0	0	0	URPR2	URPR1	URPR0
FFFFF190H	UTIC	UTIF	UTMK	0	0	0	UTPR2	UTPR1	UTPR0
FFFFF192H	UIFIC	UIFIF	UIFMK	0	0	0	UIFPR2	UIFPR1	UIFPR0
FFFFF194H	UTOIC	UTOIF	UTOMK	0	0	0	UTOPR2	UTOPR1	UTOPR0
FFFFF196H	UA0REIC	UA0REIF	UA0REMK	0	0	0	UA0REPR2	UA0REPR1	UA0REPR0
FFFFF198H	UA0RIC	UA0RIF	UA0RMK	0	0	0	UA0RPR2	UA0RPR1	UA0RPR0
FFFFF19AH	UA0TIC	UA0TIF	UA0TMK	0	0	0	UA0TPR2	UA0TPR1	UA0TPR0
FFFFF19CH	CB0REIC	CB0REIF	CB0REMK	0	0	0	CB0REPR2	CB0REPR1	CB0REPR0
FFFFF19EH	CB0RIC	CB0RIF	CB0RMK	0	0	0	CB0RPR2	CB0RPR1	CB0RPR0
FFFFF1A0H	CB0TIC	CB0TIF	CB0TMK	0	0	0	CB0TPR2	CB0TPR1	CB0TPR0
FFFFF1A2H	UA1REIC	UA1REIF	UA1REMK	0	0	0	UA1REPR2	UA1REPR1	UA1REPR0
FFFFF1A4H	UA1RIC	UA1RIF	UA1RMK	0	0	0	UA1RPR2	UA1RPR1	UA1RPR0
FFFFF1A6H	UA1TIC	UA1TIF	UA1TMK	0	0	0	UA1TPR2	UA1TPR1	UA1TPR0
FFFFF1A8H	CB1REIC	CB1REIF	CB1REMK	0	0	0	CB1REPR2	CB1REPR1	CB1REPR0
FFFFF1AAH	CB1RIC	CB1RIF	CB1RMK	0	0	0	CB1RPR2	CB1RPR1	CB1RPR0
FFFFF1ACH	CB1TIC	CB1TIF	CB1TMK	0	0	0	CB1TPR2	CB1TPR1	CB1TPR0
FFFFF1AEH	UA2REIC	UA2REIF	UA2REMK	0	0	0	UA2REPR2	UA2REPR1	UA2REPR0
FFFFF1B0H	UA2RIC	UA2RIF	UA2RMK	0	0	0	UA2RPR2	UA2RPR1	UA2RPR0
FFFFF1B2H	UA2TIC	UA2TIF	UA2TMK	0	0	0	UA2TPR2	UA2TPR1	UA2TPR0
FFFFF1B4H	CB2REIC	CB2REIF	CB2REMK	0	0	0	CB2REPR2	CB2REPR1	CB2REPR0
FFFFF1B6H	CB2RIC	CB2RIF	CB2RMK	0	0	0	CB2RPR2	CB2RPR1	CB2RPR0
FFFFF1B8H	CB2TIC	CB2TIF	CB2TMK	0	0	0	CB2TPR2	CB2TPR1	CB2TPR0

Table 20-2. Addresses and Bits of Interrupt Control Registers (3/3)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF1BAH	IICIC	IICIF	IICMK	0	0	0	IICPR2	IICPR1	IICPR0
FFFFF1BCH	AD0IC	AD0IF	AD0MK	0	0	0	AD0PR2	AD0PR1	AD0PR0
FFFFF1BEH	AD1IC	AD1IF	AD1MK	0	0	0	AD1PR2	AD1PR1	AD1PR0
FFFFF1C0H	AD2IC	AD2IF	AD2MK	0	0	0	AD2PR2	AD2PR1	AD2PR0
FFFFF1C2H	TM0EQIC0	TM0EQIF0	TM0EQMK0	0	0	0	TM0EQPR02	TM0EQPR01	TM0EQPR00
FFFFF1C4H	TM1EQIC0	TM1EQIF0	TM1EQMK0	0	0	0	TM1EQPR02	TM1EQPR01	TM1EQPR00
FFFFF1C6H	TM2EQIC0	TM2EQIF0	TM2EQMK0	0	0	0	TM2EQPR02	TM2EQPR01	TM2EQPR00
FFFFF1C8H	TM3EQIC0	TM3EQIF0	TM3EQMK0	0	0	0	TM3EQPR02	TM3EQPR01	TM3EQPR00
FFFFF1CAH	ADT0IC	ADT0IF	ADT0MK	0	0	0	ADT0PR2	ADT0PR1	ADT0PR0
FFFFF1CCH	ADT1IC	ADT1IF	ADT1MK	0	0	0	ADT1PR2	ADT1PR1	ADT1PR0

20.3.5 Interrupt mask registers 0 to 5 (IMR0 to IMR5)

The IMR0 to IMR5 registers set the interrupt mask state for the maskable interrupts. The IMR0.xxMKn to IMR3.xxMKn bits are equivalent to the xxICn.xxMKn bit.

The IMRm register can be read or written in 16-bit units (m = 0 to 5).

If the higher 8 bits of the IMRm register are used as the IMRmH register and the lower 8 bits as the IMRmL register, these registers can be read or written in 8-bit or 1-bit units.

Reset sets these registers to FFFFH.

Caution The device file defines the xxICn.xxMKn bit as a reserved word. If a bit is manipulated using the name of xxMKn, the contents of the xxICn register, instead of the IMRm register, are rewritten (as a result, the contents of the IMRm register are also rewritten).

(1/2)

After reset: FFFFH R/W Address: IMR5 FFFFF10AH
IMR5L FFFFF10AH, IMR5H FFFFF10BH

	15	14	13	12	11	10	9	8
IMR5 (IMR5H[Note 1])	1	ADT1MK	ADT0MK	TM3EQMK0	TM2EQMK0	TM1EQMK0	TM0EQMK0	AD2MK
	7	6	5	4	3	2	1	0
(IMR5L)	AD1MK	AD0MK	IICMK	CB2TMK	CB2RMK	CB2REMK	UA2TMK	UA2RMK

After reset: FFFFH R/W Address: IMR4 FFFFF108H
IMR4L FFFFF108H, IMR4H FFFFF109H

	15	14	13	12	11	10	9	8
IMR4 (IMR4H[Note 1])	UA2REMK	CB1TMK	CB1RMK	CB1REMK	UA1TMK	UA1RMK	UA1REMK	CB0TMK
	7	6	5	4	3	2	1	0
(IMR4L)	CB0RMK	CB0REMK	UA0TMK	UA0RMK	UA0REMK	UTOMK	UIFMK	UTMK

After reset: FFFFH R/W Address: IMR3 FFFFF106H
IMR3L FFFFF106H, IMR3H FFFFF107H

	15	14	13	12	11	10	9	8
IMR3 (IMR3H[Note 1])	URMK	UREMK	DMAMK3	DMAMK2	DMAMK1	DMAMK0	TA4CCMK1	TA4CCMK0
	7	6	5	4	3	2	1	0
(IMR3L)	TA4OVMK	TA3CCMK1	TA3CCMK0	TA3OVMK	TA2CCMK1	TA2CCMK0	TA2OVMK	TA1CCMK1

After reset: FFFFH R/W Address: IMR2 FFFFF104H
IMR2L FFFFF104H, IMR2H FFFFF105H

	15	14	13	12	11	10	9	8
IMR2 (IMR2H[Note 1])	TA1CCMK0	TA1OVMK	TA0CCMK1	TA0CCMK0	TA0OVMK	TT1IECMK	TT1CCMK1	TT1CCMK0
	7	6	5	4	3	2	1	0
(IMR2L)	TT1OVMK	TT0IECMK[Note 2]	TT0CCMK1	TT0CCMK0	TT0OVMK	TB1CCMK3	TB1CCMK2	TB1CCMK1

After reset: FFFFH R/W Address: IMR1 FFFFF102H
IMR1L FFFFF102H, IMR1H FFFFF103H

	15	14	13	12	11	10	9	8
IMR1 (IMR1H[Note 1])	TB1CCMK0	TB1OVMK	TB0CCMK3	TB0CCMK2	TB0CCMK1	TB0CCMK0	TB0OVMK	CMPMK1F
	7	6	5	4	3	2	1	0
(IMR1L)	CMPMK1L	CMPMK0F	CMPMK0L	PMK18	PMK17	PMK16	PMK15	PMK14

After reset: FFFFH R/W Address: IMR0 FFFFF100H
IMR0L FFFFF100H, IMR0H FFFFF101H

	15	14	13	12	11	10	9	8
IMR0 (IMR0H[Note 1])	PMK13	PMK12	PMK11	PMK10	PMK09	PMK08	PMK07[Note 2]	PMK06[Note 2]
	7	6	5	4	3	2	1	0
(IMR0L)	PMK05[Note 2]	PMK04[Note 2]	PMK03[Note 2]	PMK02[Note 2]	PMK01	PMK00	LVIHMK	LVILMK

Notes 1. When reading/writing bits 15 to 8 of the IMR0 to IMR5 registers in 8-bit or 1-bit units, specify these bits as bits 7 to 0 of the IMR0H to IMR5H registers.

2. These bits are valid only in the V850E/IG3.
Be sure to set these bits to 1 in the V850E/IF3.

Caution Set bit 15 of the IMR5 register (bit 7 of IMR5H register) to 1. The operation when these settings are changed is not guaranteed.

(2/2)

xxMKn	Interrupt mask flag setting
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Remark xx: Identification name of each peripheral unit (see **Table 20-2**)
n: Peripheral unit number (see **Table 20-2**)

20.3.6 In-service priority register (ISPR)

The ISPR register holds the priority level of the maskable interrupt currently acknowledged. When an interrupt request signal is acknowledged, the bit of this register corresponding to the priority level of that interrupt signal 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 signal 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.

Reset sets this register to 00H.

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.

After reset: 00H R Address: FFFFF1FAH

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0

ISPRn	Priority of interrupt currently being acknowledged
0	Interrupt request signal with priority n is not acknowledged
1	Interrupt request signal with priority n is being acknowledged

Remark n: 0 to 7 (priority level)

20.3.7 Maskable interrupt status flag (ID)

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

Reset sets this flag to 00000020H.

After rest: 00000020H

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

ID	Maskable interrupt servicing specification[Note]
0	Maskable interrupt request signal acknowledgment enabled
1	Maskable interrupt request signal acknowledgment disabled (pending)

Note Interrupt disable flag (ID) function

ID is set (1) by the DI instruction and cleared (0) by the EI instruction. Its value is also modified by the RETI instruction or LDSR instruction when referencing the PSW.

Non-maskable interrupt request signals and exceptions are acknowledged regardless of this flag. When a maskable interrupt request signal is acknowledged, the ID flag is automatically set (1) by hardware.

An interrupt request signal generated during the acknowledgment disabled period (ID flag = 1) can be acknowledged when the xxICn.xxIFn bit is set (1), and the ID flag is cleared (0).

20.4 External Interrupt Request Input Pins (INTP00 to INTP18, INTADT0, INTADT1)

20.4.1 Noise elimination

(1) Noise elimination of INTP00, INTP01, INTPa (V850E/IG3 only), INTP08 to INTP13, INTP17, INTP18, INTADT0, and INTADT1 pins

The INTP00, INTP01, INTPa (V850E/IG3 only), INTP08 to INTP13, INTP17, INTP18, INTADT0, and INTADT1 pins incorporate a noise eliminator that uses analog filter (a = 02 to 07). Unless, therefore, the input level of each pin is held for a certain time, an edge cannot be detected. An edge is detected after a certain time has elapsed.

(2) Noise elimination of INTP14 to INTP16 pins

The INTP14 to INTP16 pins incorporate a digital noise eliminator.

The sampling clock that performs digital sampling can be selected by the INTNFCm.INTNFCm2 to INTNFCm.INTNFCm0 bits (m = 14 to 16).

The system clock stops in the IDLE and STOP modes, so the INTP14 to INTP16 pins cannot be used to cancel the IDLE and STOP modes.

20.4.2 Edge detection

The valid edges of the INTPn pin can be selected by program (V850E/IF3: n = 00, 01, 08 to 18, V850E/IG3: n = 00 to 18). The edge that can be selected as the valid edge is one of the following.

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

The edge-detected INTPn signal becomes an interrupt source.

The valid edge is specified by the INTR0 to INTR2 and INTF0 to INTF2 registers.

(1) External interrupt rising edge specification register 0 (INTR0), external interrupt falling edge specification register 0 (INTF0)

The INTR0 and INTF0 registers are used to specify the trigger mode of the INTP00, INTP01, and INTPa (V850E/IG3 only) pins and can specify the valid edge independently for each pin (rising edge, falling edge, or both rising and falling edges) (a = 02 to 07).

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

Reset sets these registers to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port mode, an edge may be detected. Therefore, be sure to clear the INTF0n and INTR0n bits to 00, and then set the port mode (V850E/IF3: n = 0, 1, V850E/IG3: n = 0 to 7).

After reset: 00H R/W Address: FFFFFC20H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
INTR0	INTR07Note	INTR06Note	INTR05Note	INTR04Note	INTR03Note	INTR02Note	INTR01	INTR00

After reset: 00H R/W Address: FFFFFC00H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
INTF0	INTF07Note	INTF06Note	INTF05Note	INTF04Note	INTF03Note	INTF02Note	INTF01	INTF00

Note Valid only in the V850E/IG3.
In the V850E/IF3, be sure to set these bits to 0.

Remark For the valid edge specification, see **Table 20-3**.

Table 20-3. Valid Edge Specification of INTP00 to INTP07 Pins

INTF0n	INTR0n	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution When not using these pins as the INTP0n pins, be sure to set the INTF0n and INTR0n bits to 00.

Remark V850E/IF3: n = 0, 1
V850E/IG3: n = 0 to 7

(2) External interrupt rising edge specification register 1 (INTR1), external interrupt falling edge specification register 1 (INTF1)

The INTR1 and INTF1 registers are used to specify the trigger mode of the INTP08 to INTP13, INTP17, and INTP18 pins and can specify the valid edge independently for each pin (rising edge, falling edge, or both rising and falling edges).

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

Reset sets these registers to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port mode, an edge may be detected. Therefore, be sure to clear the INTFn and INTRn bits to 00, and then set the port mode (n = 08 to 13, 17, 18).

After reset: 00H R/W Address: FFFFFC22H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
INTR1	INTR18	INTR17	INTR13	INTR12	INTR11	INTR10	INTR09	INTR08

After reset: 00H R/W Address: FFFFFC02H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
INTF1	INTF18	INTF17	INTF13	INTF12	INTF11	INTF10	INTF09	INTF08

Remark For the valid edge specification, see **Table 20-4**.

Table 20-4. Valid Edge Specification of INTP08 to INTP13, INTP17, and INTP18 Pins

INTFn	INTRn	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution When not using these pins as the INTPn pins, be sure to set the INTFn and INTRn bits to 00.

Remark n = 08 to 13, 17, 18

(3) External interrupt rising edge specification register 2 (INTR2), external interrupt falling edge specification register 2 (INTF2)

The INTR2 and INTF2 registers are used to specify the trigger mode of the INTP14 to INTP16 pins and can specify the valid edge independently for each pin (rising edge, falling edge, or both rising and falling edges).

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

Reset sets these registers to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port mode, an edge may be detected. Therefore, be sure to clear the INTF1n and INTR1n bits to 00, and then set the port mode (n = 4 to 6).

After reset: 00H R/W Address: FFFFFC24H

	7	6	5	4	3	<2>	<1>	<0>
INTR2	0	0	0	0	0	INTR16	INTR15	INTR14

After reset: 00H R/W Address: FFFFFC04H

	7	6	5	4	3	<2>	<1>	<0>
INTF2	0	0	0	0	0	INTF16	INTF15	INTF14

Remark For the valid edge specification, see **Table 20-5**.

Table 20-5. Valid Edge Specification of INTP14 to INTP16 Pins

INTF1n	INTR1n	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution When not using these pins as the INTP1n pins, be sure to set the INTF1n and INTR1n bits to 00.

Remark n = 4 to 6

20.5 Software Exception

A software exception is generated when the CPU executes the TRAP instruction, and can always be acknowledged.

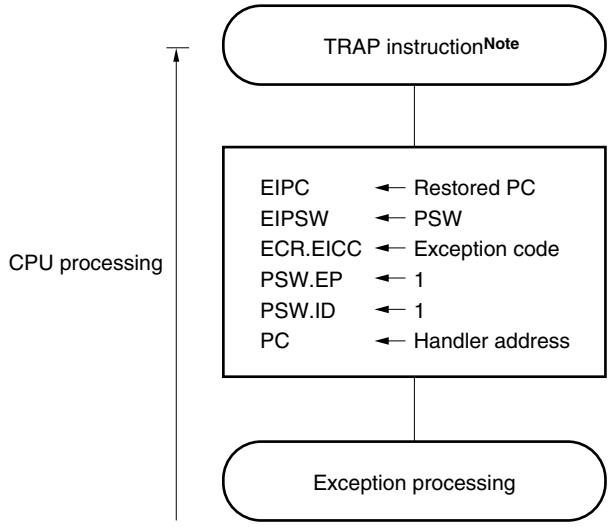
20.5.1 Operation

If a software exception occurs, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower 16 bits (EICC) of ECR (interrupt source).
<4> Sets the PSW.EP and PSW.ID bits (1).
<5> Sets the handler address (00000040H or 00000050H) corresponding to the software exception to the PC, and transfers control.

The processing of a software exception is shown below.

Figure 20-8. 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.

20.5.2 Restore

Execution is restored from software exception processing by the RETI instruction.

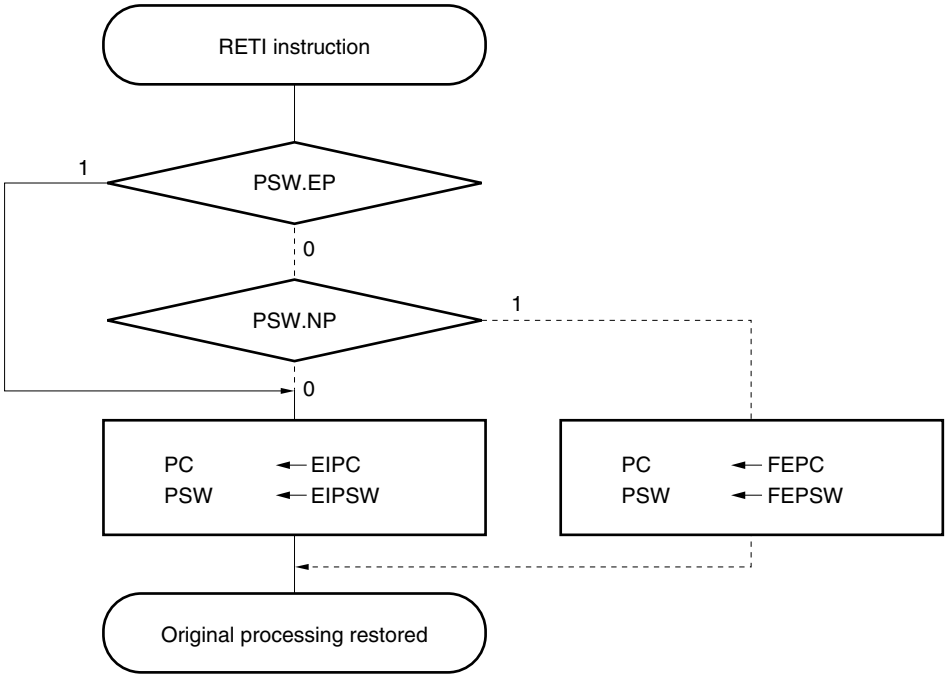
When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

<1> Loads the restored PC and PSW from EIPC and EIPSW because the PSW.EP bit is 1.

<2> Transfers control to the address of the restored PC and PSW.

The processing of the RETI instruction is shown below.

Figure 20-9. RETI Instruction Processing



Caution When the PSW.EP and PSW.NP bits are changed by the LDSR instruction during software exception processing, in order to restore the PC and PSW correctly during restoring by the RETI instruction, it is necessary to set the EP bit back to 1 and clear the NP bit to 0 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

20.5.3 Exception status flag (EP)

The EP flag is a status flag used to indicate that exception processing is in progress. This flag is set when an exception occurs. The EP flag is allocated to the PSW.

This flag is set to 00000020H after reset.

After reset: 00000020H

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

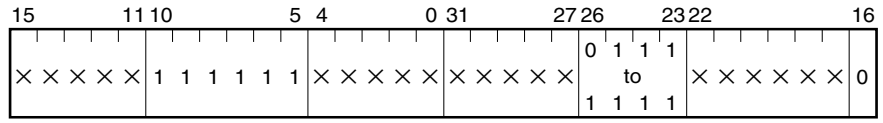
EP	Exception processing status
0	Exception processing not in progress
1	Exception processing in progress

20.6 Exception Trap

An exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. In the V850E/IF3 and V850E/IG3, an illegal opcode trap (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

20.6.1 Illegal opcode definition

The illegal instruction has an opcode (bits 10 to 5) of 111111B, a sub-opcode (bits 26 to 23) of 0111B to 1111B, and a sub-opcode (bit 16) of 0B. An exception trap is generated when an instruction applicable to this illegal instruction is executed.



×: Arbitrary

Caution Since it is possible that this instruction may be assigned to an illegal opcode in the future, it is recommended that it not be used.

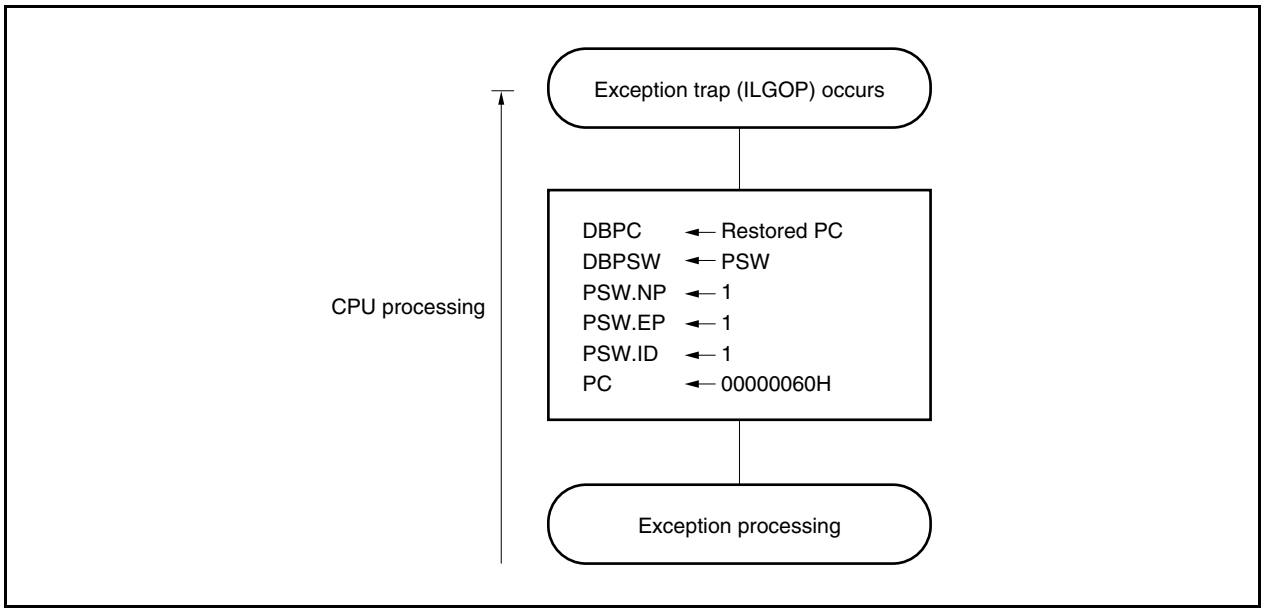
(1) Operation

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 PSW.NP, PSW.EP, and PSW.ID bits (1).
<4> Sets the handler address (00000060H) corresponding to the exception trap to the PC, and transfers control.

The processing of the exception trap is shown below.

Figure 20-10. Exception Trap Processing



(2) Restore

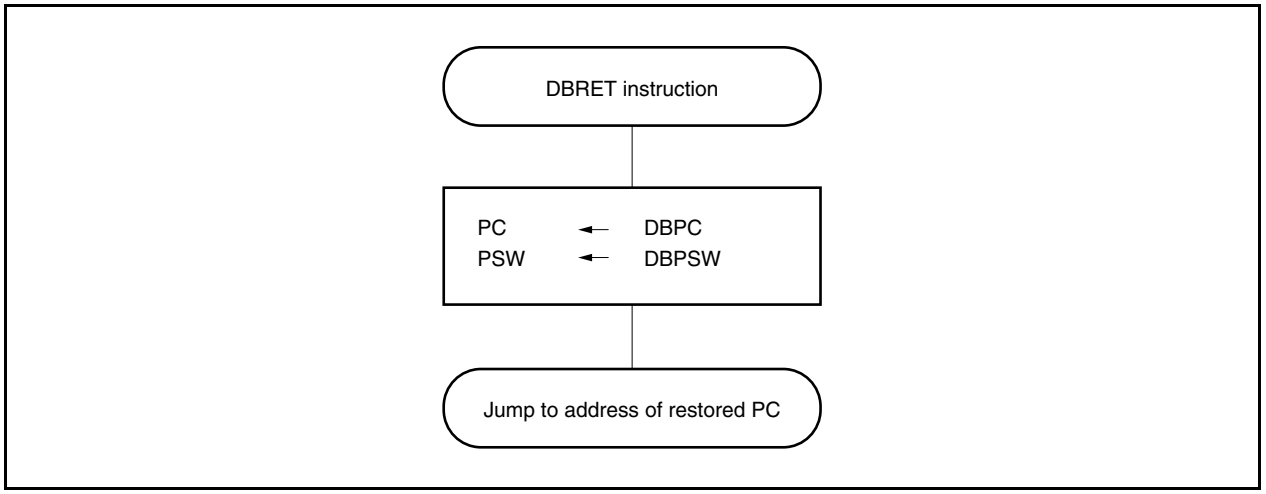
Execution is restored from an exception trap by the DBRET instruction. When the DBRET instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

<1> Loads the restored PC and PSW from DBPC and DBPSW.
<2> Transfers control to the address of the restored PC and PSW.

Caution DBPC and DBPSW can be accessed only during the period between when the illegal opcode is executed and when the DBRET instruction is executed.

The restore processing from an exception trap is shown below.

Figure 20-11. Restore Processing from Exception Trap



20.6.2 Debug trap

The debug trap is an exception that can be acknowledged anytime and is generated by execution of the DBTRAP instruction.

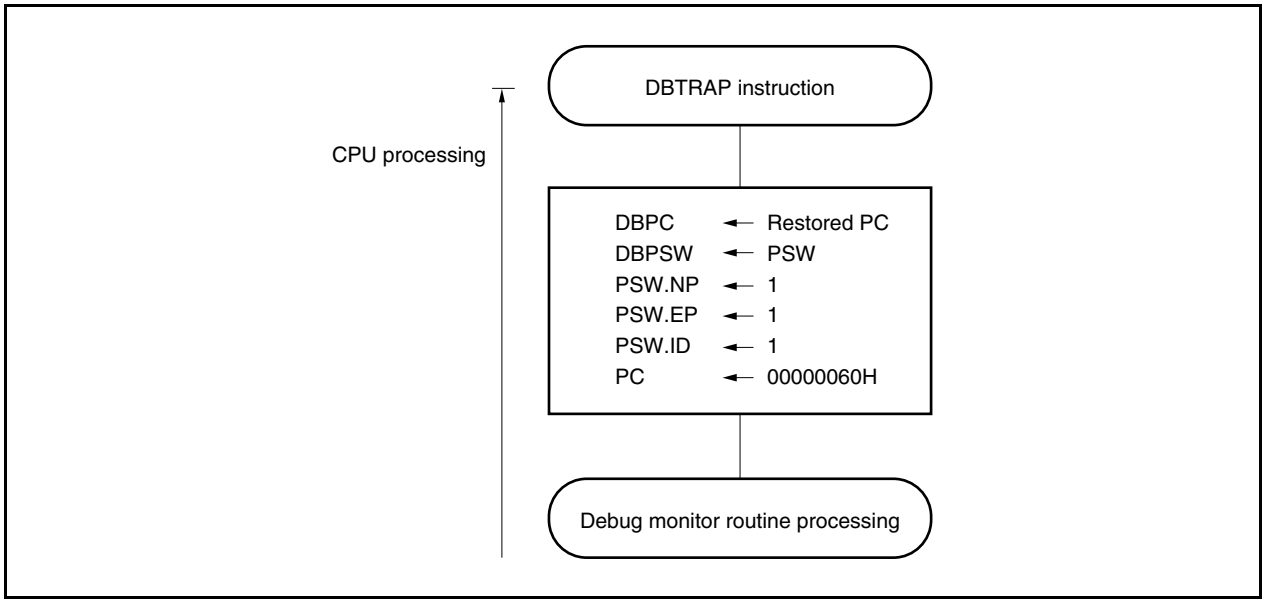
When the debug trap is generated, the CPU performs the following processing.

(1) Operation

<1> Saves the restored PC to DBPC.
<2> Saves the current PSW to DBPSW.
<3> Sets the PSW.NP, PSW.EP and PSW.ID bits (1).
<4> Sets the handler address (00000060H) corresponding to the debug trap to the PC and transfers control.

The processing of the debug trap is shown below.

Figure 20-12. Debug Trap Processing



(2) Restore

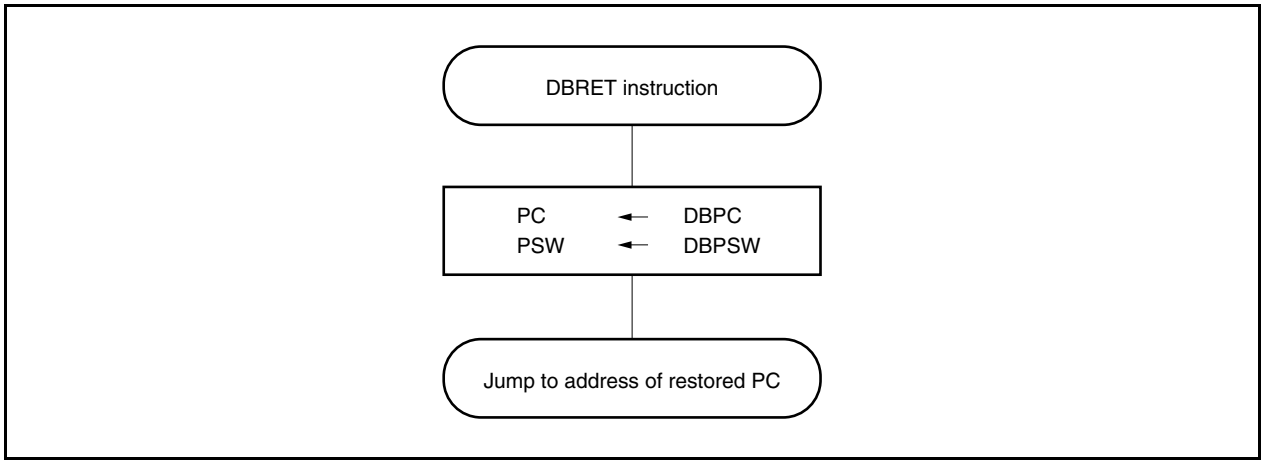
Execution is restored from a debug trap by the DBRET instruction. When the DBRET instruction is executed, the CPU performs the following processing and transfers control to the address of the restored PC.

<1> Loads the restored PC and PSW from DBPC and DBPSW.
<2> Transfers control to the address of the restored PC and PSW.

Caution DBPC and DBPSW can be accessed only during the period between when the DBTRAP is executed and when the DBRET instruction is executed.

The restore processing from a debug trap is shown below.

Figure 20-13. Restore Processing from Debug Trap



20.7 Multiple Interrupt Servicing Control

Multiple interrupt servicing control is a process by which an interrupt request that is currently being serviced can be interrupted during servicing if there is an interrupt request signal with a higher priority level, and the higher priority interrupt request signal is acknowledged and serviced first.

If there is an interrupt request signal with a lower priority level than the interrupt request currently being serviced, that interrupt request signal is held pending.

Multiple interrupt servicing control of maskable interrupts is executed when interrupts are enabled (PSW.ID bit = 0). Thus, to execute multiple interrupts, it is necessary to set the interrupt enabled state (PSW.ID bit = 0) even in an interrupt servicing routine.

If maskable interrupts are enabled or a software exception is generated in a maskable interrupt or software exception servicing program, it is necessary to save EIPC and EIPSW.

This is accomplished by the following procedure.

(1) Acknowledgment of maskable interrupt signals in servicing 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)

...
...
...
...
- DI instruction (interrupt acknowledgment disabled)
- Saved value restored to EIPSW
- Saved value restored to EIPC
- RETI instruction

← Maskable interrupt acknowledgment

(2) Generation of exception in servicing program

Servicing program of maskable interrupt or exception

... ... • EIPC saved to memory or register • EIPSW saved to memory or register ... • TRAP instruction ... • Saved value restored to EIPSW • Saved value restored to EIPC • RETI instruction	← Exception such as TRAP instruction acknowledged.

The priority order for multiple interrupt servicing control has 8 levels, from 0 to 7 for each maskable interrupt request signal (0 is the highest priority), but it can be set as desired via software. The priority order is set using the xxPRn0 to xxPRn2 bits of the interrupt control request register (xxICn), provided for each maskable interrupt request signal. After system reset, an interrupt request signal 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 signal 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.

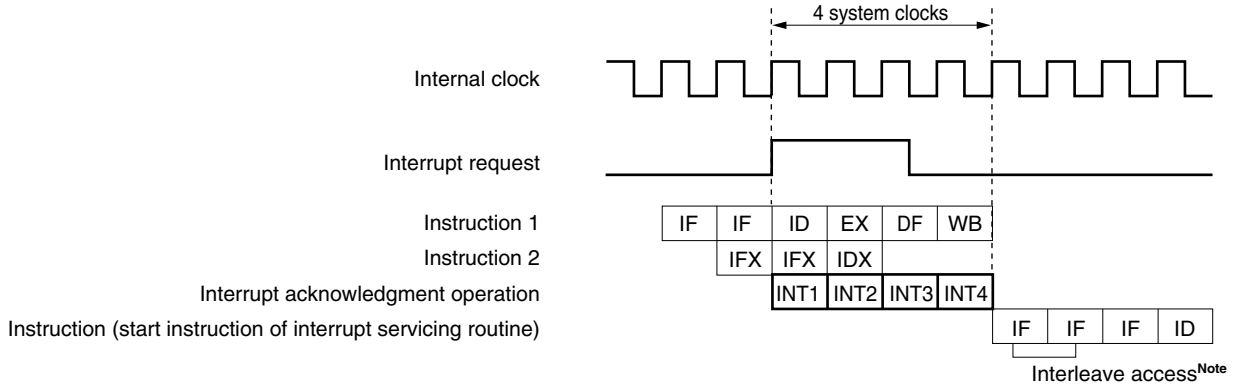
Remark xx: Identification name of each peripheral unit (see **Table 20-2**)
n: Peripheral unit number (see **Table 20-2**)

20.8 Interrupt Response Time of CPU

Except the following cases, the interrupt response time of the CPU is 4 clocks minimum. To input interrupt request signals successively, input the next interrupt request signal at least 4 clocks after the preceding interrupt.

- In IDLE/STOP mode
- When interrupt request non-sampling instructions are successively executed (see **20.9 Periods in Which CPU Does Not Acknowledge Interrupts**.)
- When an on-chip peripheral I/O register is accessed

Figure 20-14. Pipeline Operation at Interrupt Request Acknowledgment (Outline)



Note For interleave accesses, 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 latency time (internal system clock)			Conditions
	Internal interrupt	External interrupt		
		INTP00, INTP01, INTPa[Note 1], INTP08 to INTP13, INTP17, INTP18, INTADT0, INTADT1	INTP14 to INTP16	
Minimum	4	4 + Analog filter time	4 + **Note 3** + Digital noise filter	The following cases are exceptions. • In IDLE/STOP mode • Access to external bus • Two or more interrupt request non-sample instructions are executed in succession • Access to on-chip peripheral I/O register
Maximum	7[Note 2]	7 + Analog filter time	7 + **Note 3** + Digital noise filter	

Notes 1. V850E/IG3 only
2. When LD instruction is executed to internal ROM (during align access)
3. For the number of internal system clocks, see **4.6 (1) Digital noise elimination 0 control register n (INTNFCn)**.

Remark a = 02 to 07

20.9 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 request non-sample instruction and the next instruction (interrupt is held pending).

The interrupt request non-sample instructions are as follows.

- EI instruction
- DI instruction
- LDSR reg2, 0x5 instruction (for PSW)
- Store instruction for the command register (PRCMD).
- Store instructions or bit manipulation instructions excluding tst1 instruction for the following registers.
  - Interrupt-related registers:
    Interrupt control register (xxICn) and interrupt mask registers 0 to 5 (IMR0 to IMR5)
  - Power save control register (PSC)

Remark xx: Identification name of each peripheral unit (see **Table 20-2**)
n: Peripheral unit number (see **Table 20-2**)

20.10 Caution

Note that if a port is set to external interrupt input (INTPn), the timer/counter-related interrupt, serial interface-related interrupt, and A/D converter-related interrupt, which are alternate functions, do not occur (V850E/IF3: n = 00, 01, 08 to 18, ADT0, ADT1, V850E/IG3: n = 00 to 18, ADT0, ADT1).

CHAPTER 21 STANDBY FUNCTION

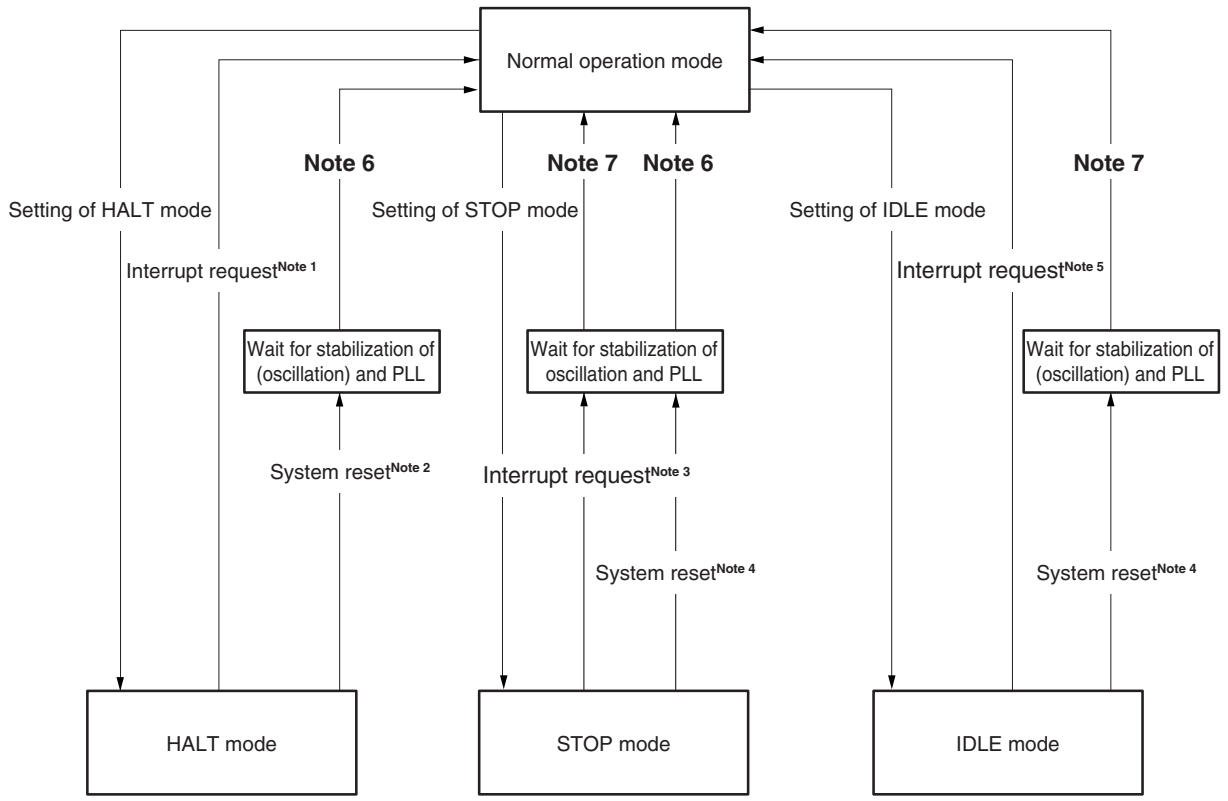
21.1 Overview

The power consumption of the system can be effectively reduced by using the standby modes in combination and selecting the appropriate mode for the application. The available standby modes are listed in Table 21-1.

Table 21-1. Standby Modes

Mode	Functional Outline
HALT mode	Mode to stop only the operating clock of the CPU
IDLE mode	Mode to stop all the operations of the internal circuit except the oscillator, PLL, CSIB in the slave mode, clock monitor, low-voltage detector (LVI), power-on-clear circuit (POC)
STOP mode	Mode to stop all the operations of the internal circuit except the CSIB in the slave mode, low-voltage detector (LVI), power-on-clear circuit (POC)

Figure 21-1. Status Transition



Notes 1. Non-maskable interrupt request signal (INTWDT) or unmasked maskable interrupt request signal

2. $\overline{\text{RESET}}$ pin input, reset signal (WDTRES) generation by watchdog timer overflow, reset signal (LVIRES) generation by low-voltage detector (LVI), or reset signal (POCRES) generation by power-on-clear circuit (POC)

3. Unmasked external interrupt request signal (INTP00, INTP01, INTP02 to INTP07 (V850E/IG3 only), INTP08 to INTP13, INTP17, INTP18, INTADT0, or INTADT1) or unmasked internal interrupt request signal from (CSIB-related interrupt request signal in the slave mode) peripheral functions operable in STOP mode

4. $\overline{\text{RESET}}$ pin input, reset signal (LVIRES) generation by low-voltage detector (LVI), or reset signal (POCRES) generation by power-on-clear circuit (POC)

5. Unmasked external interrupt request signal (INTP00, INTP01, INTP02 to INTP07 (V850E/IG3 only), INTP08 to INTP13, INTP17, INTP18, INTADT0, or INTADT1) or unmasked internal interrupt request signal (CSIB-related interrupt request signal in the slave mode) from peripheral functions operable in IDLE mode

6. Oscillation stabilization time count by oscillation stabilization time wait control (OST)
The oscillation stabilization time is necessary after release of reset because the PLL is initialized by a reset. The stabilization time is the time determined by default.

7. Oscillation stabilization time count by oscillation stabilization time wait control (OST)
The stabilization time is determined by the setting of the OSTS register.

21.2 Control Registers

(1) Power save control register (PSC)

The PSC register is an 8-bit register that controls the standby function. The STB bit of this register is used to specify the standby mode. This register is a special register (see **3.4.8 Special registers**). This register can be written only by a combination of specific sequences.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF1FEH

	7	6	5	<4>	3	2	<1>	0
PSC	0	0	0	INTM	0	0	STB	0

INTM	Standby mode control[Note 2] by maskable interrupt request (INTxx[Note 1])
0	Standby mode release by INTxx request enabled
1	Standby mode release by INTxx request disabled

STB	Sets operation mode
0	Normal mode
1	Standby mode

Notes 1. For details, see **Table 20-1 Interrupt Source List**.

2. The setting is valid only in the IDLE mode and STOP mode.

Cautions 1. Be sure to set bits 0, 2, 3, and 5 to 7 to "0".

2. Before setting a standby mode by setting the STB bit to 1, be sure to set the PCC register to 03H and then set the STB bit to 1. Otherwise, the standby mode may not be set or released. After releasing the standby mode, change the value of the PCC register to the desired value.

3. To set the IDLE mode or STOP mode, set the PCC register to 03H, and the PSMR.PSM0 bit in that order and then set the STB bit to 1.

(2) Power save mode register (PSMR)

The PSMR register is an 8-bit register that controls the operation in the software standby mode.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF820H

	7	6	5	4	3	2	1	<0>
PSMR	0	0	0	0	0	0	0	PSM0

PSM0	Operation in software standby mode specification
0	IDLE mode
1	STOP mode

Cautions 1. Be sure to set bits 1 to 7 to "0".

2. The PSM0 bit is valid only when the PSC.STB bit is 1.

21.3 HALT Mode

21.3.1 Setting and operation status

The HALT mode is set when a dedicated instruction (HALT) is executed in the normal operation mode.

When HALT mode is set, clock supply is stopped to the CPU only. The clock generator and PLL continue operating. Clock supply to the other on-chip peripheral functions continues.

As a result, program execution is stopped, and the internal RAM retains the contents before the HALT mode was set. The on-chip peripheral functions that are independent of instruction processing by the CPU continue operating.

Table 21-3 shows the operation status in the HALT mode.

The average power consumption of the system can be reduced by using the HALT mode in combination with the normal operation mode for intermittent operation.

Cautions 1. Insert five or more NOP instructions after the HALT instruction.

2. If the HALT instruction is executed while an interrupt request is being held pending, the HALT mode is set but is released immediately by the pending interrupt request.

21.3.2 Releasing HALT mode

The HALT mode is released by a non-maskable interrupt request signal (INTWDT), an unmasked maskable interrupt request signal, and a reset signal ($\overline{\text{RESET}}$ pin input, reset signal (WDTRES) generation by watchdog timer overflow, reset signal (LVIRES) generation by low-voltage detector (LVI), or reset signal (POCRES) generation by power-on-clear circuit (POC)).

After the HALT mode has been released, the normal operation mode is restored.

(1) Releasing HALT mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The HALT mode is released by a non-maskable interrupt request signal (INTWDT) or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request. If the HALT mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

(a) If an interrupt request signal with a priority lower than or same as the interrupt currently being serviced is generated, the HALT mode is released, but the newly generated interrupt request signal is not acknowledged. The interrupt request signal itself is retained. Therefore, execution starts at the next instruction after the HALT instruction.

(b) If an interrupt request signal with a priority higher than that of the interrupt currently being serviced is issued (including a non-maskable interrupt request signal), the HALT mode is released and that interrupt request signal is acknowledged. Therefore, execution branches to the handler address.

Table 21-2. Operation After Releasing HALT Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address	
Unmasked maskable interrupt request signal	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(2) Releasing HALT mode by $\overline{\text{RESET}}$ pin input or WDTRES signal generation

The same operation as the normal reset operation is performed.

Table 21-3. Operation Status in HALT Mode

Setting of HALT Mode / Item		Operation Status
Clock generator, PLL		Operates
System clock (f_{XX})		Supply
CPU		Stops operation
External bus interface[Note]		See **Table 2-2 Pin Operation Status in Operation Modes**.
DMA		Operable
Interrupt controller		Operable
Timer	TAA0 to TAA4	Operable
	TAB0, TAB1	Operable
	TMT0, TMT1	Operable
	TMM0 to TMM3	Operable
Watchdog timer		Operable
Serial interface	CSIB0 to CSIB2	Operable
	UARTA0 to UARTA2	Operable
	UARTB	Operable
	I^2C	Operable
A/D converters 0 to 2		Operable
Clock monitor		Operable
Low-voltage detector		Operable
Power-on-clear circuit		Operable
Port function		Retains status before HALT mode was set.
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the HALT mode was set.

Note μPD70F3454GC-8EA-A only

21.4 IDLE Mode

21.4.1 Setting and operation status

The IDLE mode is set by clearing (0) the PSMR.PSM0 bit and setting (1) the PSC.STB bit in the normal operation mode.

In the IDLE mode, the clock generator and PLL continue operation but clock supply to the CPU and other on-chip peripheral functions stops.

As a result, program execution stops and the contents of the internal RAM before the IDLE mode was set are retained. The CPU and other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate with an external clock continue operating.

Table 21-5 shows the operation status in the IDLE mode.

The IDLE mode can reduce the power consumption more than the HALT mode because it stops the operation of the on-chip peripheral functions. The clock generator and PLL do not stop, so the normal operation mode can be restored without waiting for the oscillation stabilization time after the IDLE mode has been released, in the same manner as when the HALT mode is released.

Caution Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the IDLE mode.

21.4.2 Releasing IDLE mode

The IDLE mode is released by an unmasked external interrupt request signal (INTP00, INTP01, INTP02 to INTP07 (V850E/IG3 only), INTP08 to INTP13, INTP17, INTP18, INTADT0, or INTADT1 pin input), an unmasked internal interrupt request signal (CSIB-related interrupt request signal in the slave mode) from the peripheral functions operable in the IDLE mode, or a reset signal ($\overline{\text{RESET}}$ pin input, reset signal (LVIRES) generation by low-voltage detector (LVI), or reset signal (POCRES) generation by power-on-clear circuit (POC)).

After the IDLE mode has been released, the normal operation mode is restored.

(1) Releasing IDLE mode by unmasked maskable interrupt request signal

The IDLE mode is released by an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request. If the IDLE mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is processed as follows.

Caution When PSC.INTM bit = 1, the IDLE mode cannot be released by the unmasked maskable interrupt request signal.

(a) If an interrupt request with a priority lower than or same as the interrupt request signal currently being serviced is generated, the IDLE mode is released, but the newly generated interrupt is not acknowledged. The interrupt request signal itself is retained. Therefore, execution starts at the next instruction after the IDLE instruction.

(b) If an interrupt request signal with a priority higher than that of the interrupt request signal currently being serviced is issued (including a non-maskable interrupt request signal), the IDLE mode is released and that interrupt request signal is acknowledged. Therefore, execution branches to the handler address.

Table 21-4. Operation After Releasing IDLE Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Unmasked maskable interrupt request	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(2) Releasing IDLE mode by $\overline{\text{RESET}}$ pin input

The same operation as the normal reset operation is performed.

Table 21-5. Operation Status in IDLE Mode

Item \ Setting of IDLE Mode		Operation Status
Clock generator, PLL		Operates
System clock (f_{XX})		Stops supply
CPU		Stops operation
External bus interface[Note]		See **Table 2-2 Pin Operation Status in Operation Modes**.
DMA		Stops operation
Interrupt controller		Stops operation
Timer	TAA0 to TAA4	Stops operation
	TAB0, TAB1	Stops operation
	TMT0, TMT1	Stops operation
	TMM0 to TMM3	Stops operation
Watchdog timer		Stops operation
Serial interface	CSIB0 to CSIB2	Operable when $\overline{\text{SCKBn}}$ input clock is selected as count clock (in slave mode) (n = 0 to 2)
	UARTA0 to UARTA2	Stops operation
	UARTB	Stops operation
	I^2C	Stops operation
A/D converters 0 to 2		Stops operation
Clock monitor		Operable
Low-voltage detector		Operable
Power-on-clear circuit		Operable
Port function		Retains status before IDLE mode was set.
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the IDLE mode was set.

Note μPD70F3454GC-8EA-A only

21.5 STOP Mode

21.5.1 Setting and operation status

The STOP mode is set by setting (1) the PSMR.PSM0 bit and setting (1) the PSC.STB bit in the normal operation mode.

In the STOP mode, the clock generator stops operation. Clock supply to the CPU and the on-chip peripheral functions is stopped.

As a result, program execution is stopped, and the contents of the internal RAM before the STOP mode was set are retained. The CPU and other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate with an external clock continue operating.

Table 21-7 shows the operation status in the STOP mode.

Because the STOP stops operation of the clock generator, it reduces the power consumption to a level lower than the IDLE mode. When the external clock is not used, the power consumption can be minimized with only leakage current flowing.

Caution Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the STOP mode.

21.5.2 Releasing STOP mode

The STOP mode is released by an unmasked external interrupt request signal (INTP00, INTP01, INTP02 to INTP07 (V850E/IG3 only), INTP08 to INTP13, INTP17, INTP18, INTADT0, or INTADT1 pin input), an unmasked internal interrupt request signal (CSIB-related interrupt signal in the slave mode) from the peripheral functions operable in the STOP mode, or a reset signal ($\overline{\text{RESET}}$ pin input, reset signal (LVIRES) generation by low-voltage detector (LVI), or reset signal (POCRES) generation by power-on-clear circuit (POC)).

After the STOP mode has been released, the normal operation mode is restored after the oscillation stabilization time has been secured.

(1) Releasing STOP mode by unmasked maskable interrupt request signal

The STOP mode is released by an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request. If the STOP mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

Caution When PSC.INTM bit = 1, the STOP mode cannot be released by the unmasked maskable interrupt request signal.

(a) If an interrupt request with a priority lower than or same as the interrupt request currently being serviced is generated, the STOP mode is released, but the newly generated interrupt is not acknowledged. The interrupt request itself is retained. Therefore, execution starts at the next instruction after the STOP instruction.

(b) If an interrupt request with a priority higher than that of the interrupt request currently being serviced is issued, the STOP mode is released and that interrupt request is acknowledged. Therefore, execution branches to the handler address.

Table 21-6. Operation After Releasing STOP Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Unmasked maskable interrupt request	Execution branches to the handler address or the next instruction is executed after securing oscillation stabilization time	The next instruction is executed after securing oscillation stabilization time

(2) Releasing STOP mode by $\overline{\text{RESET}}$ pin input

The same operation as the normal reset operation is performed.

Table 21-7. Operation Status in STOP Mode

Setting of STOP Mode / Item		Operation Status
Clock generator, PLL		Stops operation
System clock (f_{XX})		Stops supply
CPU		Stops operation
External bus interface[Note]		See **Table 2-2 Pin Operation Status in Operation Modes**.
DMA		Stops operation
Interrupt controller		Stops operation
Timer	TAA0 to TAA4	Stops operation
	TAB0, TAB1	Stops operation
	TMT0, TMT1	Stops operation
	TMM0 to TMM3	Stops operation
Watchdog timer		Stops operation
Serial interface	CSIB0 to CSIB2	Operable when $\overline{\text{SCKBn}}$ input clock is selected as count clock (in slave mode) (n = 0 to 2)
	UARTA0 to UARTA2	Stops operation
	UARTB	Stops operation
	I^2C	Stops operation
A/D converters 0 to 2		Stops operation
Clock monitor		Stops operation
Low-voltage detector		Operable
Power-on-clear circuit		Operable
Port function		Retains status before STOP mode was set.
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the STOP mode was set.

Note μPD70F3454GC-8EA-A only

21.6 Securing Oscillation Stabilization Time

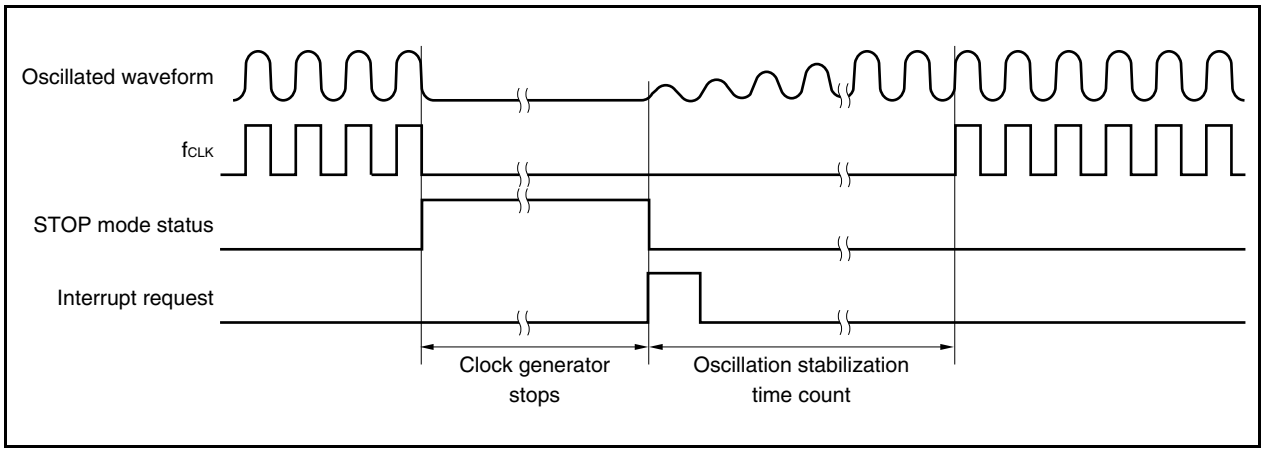
When the STOP mode is released, the oscillation stabilization time set by the OSTS register elapses. The oscillation stabilization time is the reset value of the OSTS register, $2^{14}/f_X$ (2.048 ms at f_X = 8 MHz), if the STOP mode is released by $\overline{\text{RESET}}$ pin input.

However, the actual oscillation stabilization time is half this value (after reset: $2^{13}/f_X$ (1.024 ms at f_X = 8 MHz), and the other half is the stabilization time of the PLL. Set an oscillation stabilization time double that of the oscillation stabilization time of the oscillator used when the STOP mode is released. If the oscillation stabilization time of the oscillator used is longer than $2^{13}/f_X$ when the STOP mode is released by $\overline{\text{RESET}}$ pin input, secure the oscillation stabilization time with the low-level width of the $\overline{\text{RESET}}$ signal.

The timer for counting the oscillation stabilization time secures oscillation stabilization time equal to the overflow time of the watchdog timer.

The operation performed when the STOP mode is released by an interrupt request signal is shown below.

Figure 21-2. Oscillation Stabilization Time



Caution For details of the OSTS register, see 5.3 (5) Oscillation stabilization time select register (OSTS).

CHAPTER 22 RESET FUNCTIONS

22.1 Overview

- System reset by $\overline{\text{RESET}}$ pin input
- System reset signal (WDTRES) generation by watchdog timer (WDT) overflow
- System reset signal (LVIRES) generation by low-voltage detector (LVI)
- System reset signal (POCRES) generation by power-on-clear circuit (POC)
- Forced reset by on-chip debug function (DCU) and reset mask function (see **CHAPTER 26 ON-CHIP DEBUG FUNCTION**.)

22.2 Control Register

(1) Reset source flag register (RESF)

The RESF register is an 8-bit register that indicates occurrence of a reset request from the watchdog timer (WDT) or low-voltage detector (LVI).

The WDTRF or LVIRF bit of this register is set to 1 when the internal reset source signal from WDT or LVI is asserted. The WDTRF or LVIRF bit is cleared by reset via the $\overline{\text{RESET}}$ pin or by a bit manipulation instruction or store instruction (writing 0 to the WDTRF or LVIRF bit).

The RESF register is a special register and can be written only in a combination of specific sequences (see **3.4.8 Special registers**).

This register can be read or written in 8-bit or 1-bit units. However, bits 0 and 4 can only be cleared (0) by writing.

This register is set to 00H by $\overline{\text{RESET}}$ pin input and reset by the power-on-clear circuit (POC). The default value differs if the source of reset is other than the $\overline{\text{RESET}}$ pin input and reset by the power-on-clear circuit (POC). For details on reset conflict, see **Caution** below.

After reset: 00H[Note] R/W Address: FFFFF888H

	7	6	5	4	3	2	1	0
RESF	0	0	0	WDTRF	0	0	0	LIVRF

WDTRF	Occurrence of reset signal from watchdog timer (WDT)
0	Read: No reset request, Write: Clear
1	Reset request

LIVRF	Occurrence of reset signal from low-voltage detector (LVI)
0	Read: No reset request, Write: Clear
1	Reset request

Note After reset by $\overline{\text{RESET}}$ pin input or power-on-clear circuit (POC): 00H
After reset by watchdog timer overflow: 10H
After reset by low-voltage detector (LVI): 01H

Caution If setting (occurrence of reset of set source) and clearing (occurrence of system reset or writing 0 to the WDTRF or LVIRF bit) of the RESF register conflict, the priorities are as follows.

1. **Occurrence of reset via $\overline{\text{RESET}}$ pin input (clearing RESF register)**
2. **Occurrence of reset by WDT or LVI (setting RESF register)**
3. **Writing 0 to the WDTRF or LVIRF bit by a bit manipulation or store instruction (clearing RESF register)**

If the occurrence of reset via the $\overline{\text{RESET}}$ pin input and the occurrence of reset by the WDT or LVI conflict, the RESF register is not set but cleared (00H).

22.3 Operation

(1) Reset operation by $\overline{\text{RESET}}$ pin input

When a low level is input to the $\overline{\text{RESET}}$ pin, the V850E/IF3 and V850E/IG3 are reset, and each hardware unit is initialized to a specific status.

The oscillator continues oscillation even while a low level is input to the $\overline{\text{RESET}}$ pin but the oscillation mode is initialized to the clock-through mode (PLLCTL register = 01H) and the CPU clock (f_{CPU}) division to f_{XX}/8 (PCC register = 03H).

The reset status is released when the $\overline{\text{RESET}}$ pin input goes from low to high. After the reset status is released, the oscillation stabilization time of the oscillator and lockup time of PLL (default value of OSTS register for the total time: $2^{14}/f_X$ (2.05 ms (f_X = 8 MHz)) elapse, and then the CPU starts program execution. After release of reset, therefore, the operation is started in the clock-through mode and at f_{XX}/8.

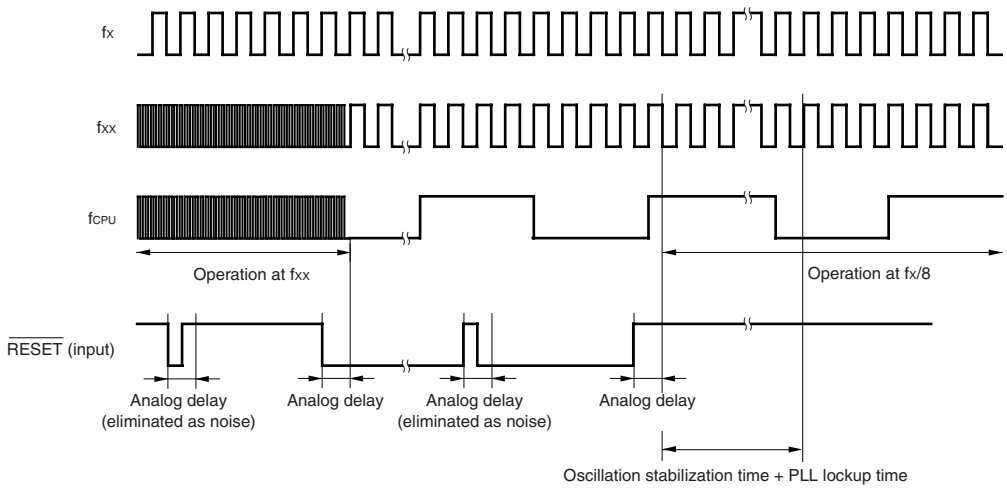
The status of each hardware unit during the reset period and after the reset status is released is shown below.

Hardware	During Reset Period	After Reset Is Released
Clock generator: Oscillator (f_X) Internal system clock (f_{CLK}) CPU clock (f_{CPU}) External bus clock (f_{BUS})[Note]	Oscillation/supply continues However, the CPU clock (f_{CPU}) is initialized to f_{XX}/8.	
Clock generator: Peripheral clock (f_{XX} to f_{XX}/4096)	Oscillation/supply stops	Oscillation/supply starts after securing of oscillation stabilization time
Clock generator: Watchdog timer clock (f_{XX}/1024)	Oscillation/supply stops	Oscillation/supply starts
CPU	Initialized	Program execution starts after securing of oscillation stabilization time
Internal RAM	Retains value immediately before reset input only in the STOP mode during reset input. Otherwise, undefined.	
Ports (including alternate-function pins)	High impedance	
On-chip peripheral I/O registers (other than ports)	Initialized to specific status	
On-chip peripheral functions other than above	Stops operation	Can start operation

Note μPD70F3454GC-8EA-A only

The reset operation by $\overline{\text{RESET}}$ pin input is illustrated below.

Figure 22-1. Reset Operation by $\overline{\text{RESET}}$ Pin Input



Caution After release of reset, make sure that the oscillation stabilization time (1.024 ms (at fx = 8 MHz)) and PLL lockup time (1.024 ms (at fx = 8 MHz)) elapse. If an oscillation stabilization time of 1.024 ms is not sufficient to secure stable oscillation, keep the $\overline{\text{RESET}}$ pin low for the deficient time.

Remark The relationship between fxx and fx in the above timing chart is fxx = 8 × fx.

The operation after release of reset is the same in both the PLL mode and clock-through mode and is started in the clock-through mode. Set the PLL mode by software control (setting PLLCTL.SELPLL bit to 1). To improve noise immunity, it is recommended to set the PLL mode and then speed up the CPU clock (example: PCC register = 00H (fxx operation)).

(2) Reset operation (WDTRES) by overflow of watchdog timer (WDT)

If the reset mode is set to reset upon overflow of the watchdog timer (WDT) (WDTM.WDM1 and WDTM.WDM0 bits = 10 or 11), the system is reset and each hardware is initialized to a specific state when WDT overflows (INTWDT).

If the INTWDT interrupt request signal is generated, the RESF.WDTRF bit is set to 1, indicating that internal reset has occurred.

The operations during the reset period and after release of reset, other than the operation of the RESF register, are the same as the reset operation by $\overline{\text{RESET}}$ pin input (see **(1) Reset operation by $\overline{\text{RESET}}$ pin input**).

(3) Reset operation (LVIRES) by low-voltage detector (LVI)

When LVI operation is enabled, the supply voltage (V_{DD0}, V_{DD1}) and detection voltage (V_{LVI}) are compared and if the supply voltage drops below the detection voltage, the system is reset (when the LVIM.LVIMD bit is set to "1") and each hardware is initialized to a specific state.

The system is reset when V_{DD0}, $V_{DD1} < V_{LVI}$ and reset is released when V_{DD0}, $V_{DD1} \geq V_{LVI}$. After a reset is released, when the oscillation stabilization time (default value of the OSTS register: $2^{14}/f_X$) of the oscillator has elapsed, the CPU starts executing the program.

The oscillator stops during a reset, so secure the oscillation stabilization time.

The status of each hardware during the reset period and after reset release is the same as the reset operation by the $\overline{\text{RESET}}$ pin (see **(1) Reset operation by $\overline{\text{RESET}}$ pin input**).

For details of the reset operation by low-voltage detector (LVI), see **CHAPTER 23 LOW-VOLTAGE DETECTOR**.

(4) Reset operation (POCRES) by power-on-clear circuit (POC)

When the supply voltage (V_{DD0}, V_{DD1}) and detection voltage (V_{POC0}) are compared and if the supply voltage drops below the detection voltage (including at power application), the system is reset and each hardware is initialized to a specific state.

The system is reset when V_{DD0}, $V_{DD1} < V_{POC0}$ and reset is released when V_{DD0}, $V_{DD1} \geq V_{POC0}$. After a reset is released, when the oscillation stabilization time (default value of the OSTS register: $2^{14}/f_X$) of the oscillator has elapsed, the CPU starts executing the program.

The oscillator stops during a reset, so secure the oscillation stabilization time.

The status of each hardware during the reset period and after reset release is the same as the reset operation by the $\overline{\text{RESET}}$ pin (see **(1) Reset operation by $\overline{\text{RESET}}$ pin input**).

For details of the reset operation by power-on-clear circuit (POC), see **CHAPTER 24 POWER-ON-CLEAR CIRCUIT**.

CHAPTER 23 LOW-VOLTAGE DETECTOR

23.1 Functions

The low-voltage detector (LVI) has the following functions.

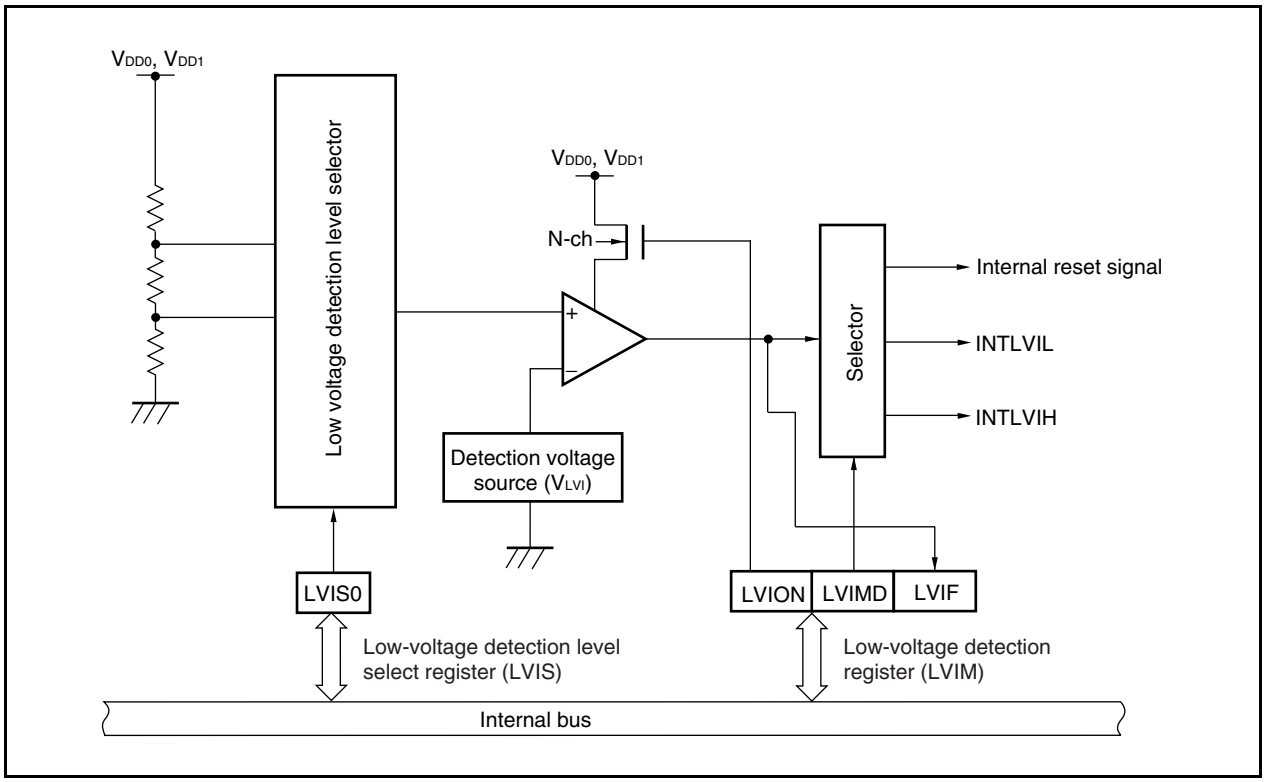
- Compares the supply voltage (V_{DD0}, V_{DD1}) and detection voltage (V_{LVI}) and generates an interrupt request signal (INTLVIL, INTLVIH) or internal reset signal (LVIRES) when V_{DD0}, $V_{DD1} < V_{LVI}$.
- The level of the supply voltage to be detected can be changed by software (in two steps).
- An interrupt request signal (INTLVIL, INTLVIH) or internal reset signal (LVIRES) can be selected.
- Can operate in STOP mode.
- Operation can be stopped by software.

If the low-voltage detector is used to generate a reset signal, the RESF.LVIRF bit is set to 1 when the reset signal is generated. For details of RESF register, see **CHAPTER 22 RESET FUNCTIONS**.

23.2 Configuration

The block diagram is shown below.

Figure 23-1. Block Diagram of Low-Voltage Detector



23.3 Control Registers

(1) Low-voltage detection register (LVIM)

The LVIM register is used to enable or disable low voltage detection, and to set the operation mode of the low-voltage detector. The LVIM register is a special register. It can be written only by a combination of specific sequences (see **3.4.8 Special registers**).

This register can be read or written in 8-bit or 1-bit units. However, bit 0 is read-only.

Reset other than reset by the low-voltage detector (LVI) sets this register to 00H.

After reset: 00H R/W Address: FFFFF890H

	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF

LVION	Low voltage detection operation enable or disable
0	Disable operation.
1	Enable operation.

LVIMD	Selection of operation mode of low voltage detection
0	Generate interrupt request signal INTLVIL when supply voltage < detection voltage. Generate interrupt request signal INTLVIH when supply voltage > detection voltage.
1	Generate internal reset signal LVIRES when supply voltage < detection voltage.

LVIF	Low voltage detection flag
0	When supply voltage > detection voltage, or when operation is disabled
1	Supply voltage < detection voltage

Cautions 1. After setting the LVION bit to 1, wait for TBD before checking the voltage using the LVIF bit.

2. The value of the LVIF flag is output as the output signals INTLVIL or INTLVIH when the LVION bit = 1 and LVIMD bit = 0.

3. If the LVION bit = 1 and LVIMD bit = 1, the low-voltage detector (LVI) cannot be stopped until a reset request other than that of by the LVI is generated.

4. Be sure to set bits 2 to 6 to "0".

(2) Low-voltage detection level select register (LVIS)

The LVIS register is used to select the level of low voltage to be detected.

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

Reset other than reset by the low-voltage detector (LVI) sets this register to 00H.

After reset: 00H R/W Address: FFFFF891H

	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	0	0	0	LVIS0

LVIS0	Detection level
0	4.4 V ±0.2 V
1	4.2 V ±0.2 V

Cautions 1. The LVIS register cannot be written until a reset request due to something other than the low-voltage detector (LVI) is generated after the LVIM.LVION and LVIM.LVIMD bits are set to 1.

2. Be sure to clear bits 1 to 7 to "0".

23.4 Operation

Depending on the setting of the LVIM.LVIMD bit, an interrupt request signal (INTLVIL, INTLVIH) or an internal reset signal (LVIRES) is generated.

23.4.1 To use for internal reset signal

<To start operation>

<1> Mask the interrupt of the low-voltage detector (LVI).

<2> Select the voltage to be detected by using the LVIS.LVIS0 bit.

<3> Set the LVIM. LVION bit to 1 (to enable operation).

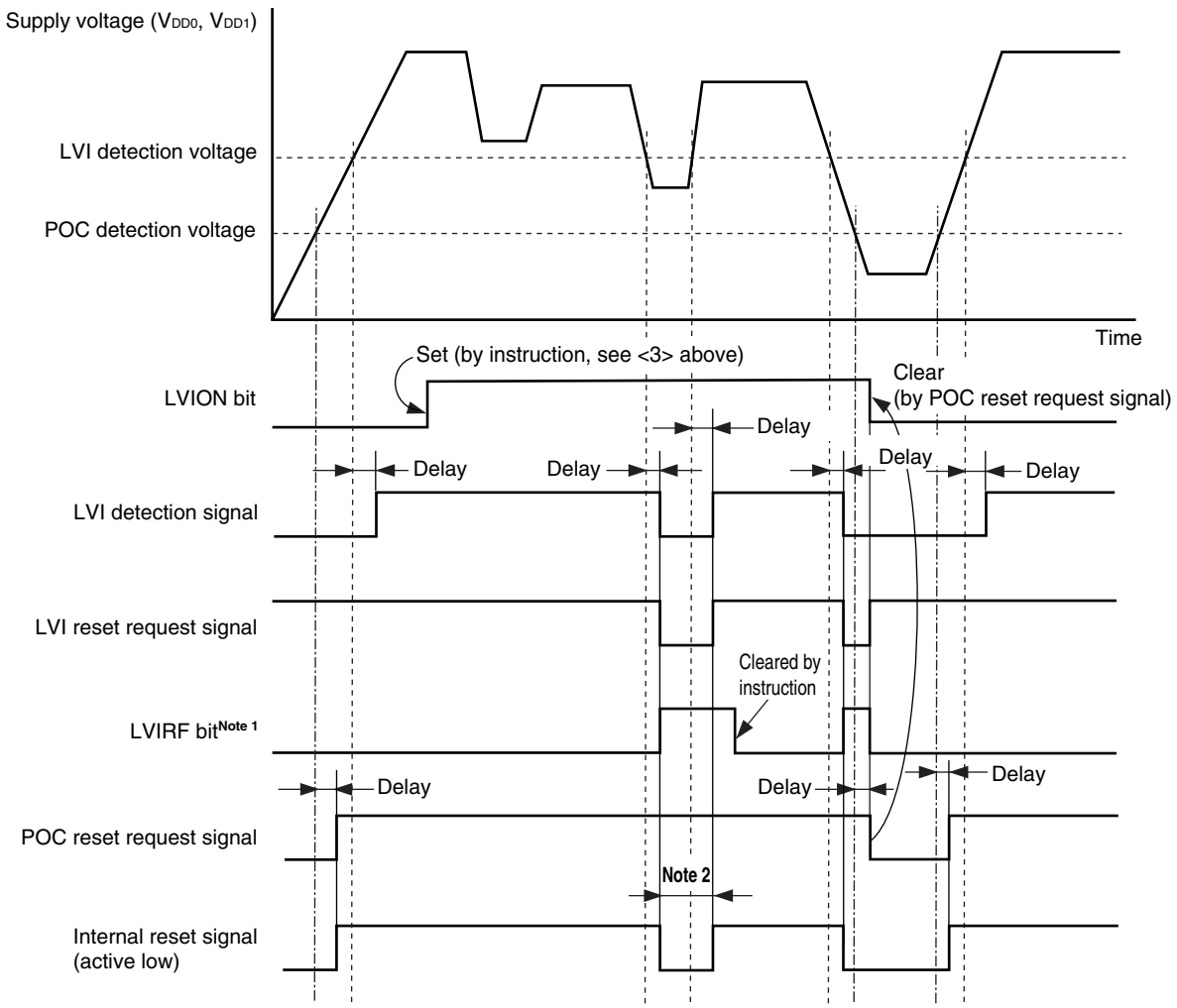
<4> Insert a wait cycle of TBD or more by software.

<5> By using the LVIM.LVIF bit, check if the supply voltage > detection voltage.

<6> Set the LVIM.LVIMD bit to 1 (to generate an internal reset signal).

Caution If the LVIMD bit is set to 1, the contents of the LVIM and LVIS registers cannot be changed until a reset request other than the low-voltage detector (LVI) is generated.

Figure 23-2. Operation Timing of Low-Voltage Detector (LVIMD Bit = 1)



Notes 1. The LVIRF bit is bit 0 of the reset source flag register (RESF). For details of RESF, see **CHAPTER 22 RESET FUNCTIONS**.

2. During the period in which the supply voltage is the set voltage or lower, the internal reset signal is retained (internal reset state).

23.4.2 To use for interrupt

<To start operation>

<1> Mask the interrupt of the low-voltage detector (LVI).

<2> Select the voltage to be detected by using the LVIS.LVIS0 bit.

<3> Set the LVIM.LVION bit to 1 (to enable operation).

<4> Insert a wait cycle of TBD or more by software.

<5> By using the LVIM.LVIF bit, check if the supply voltage > detection voltage.

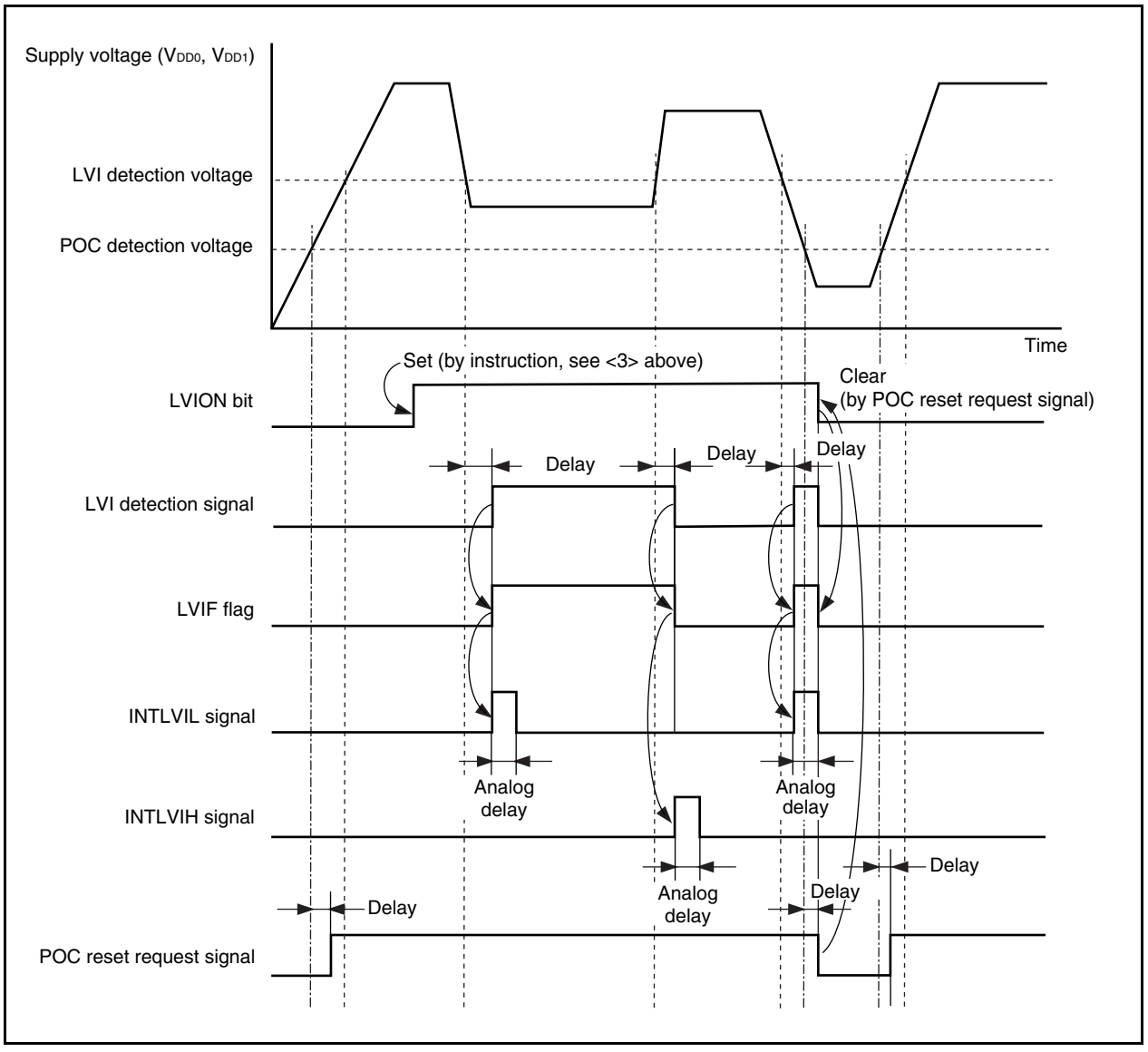
<6> Clear the interrupt request flag of LVI.

<7> Unmask the interrupt of LVI.

<To stop operation>

Set the LVION bit to 0.

Figure 23-3. Operation Timing of Low-Voltage Detector (LVIMD Bit = 0)



CHAPTER 24 POWER-ON CLEAR CIRCUIT

24.1 Function

Functions of the power-on-clear circuit (POC) are shown below.

- Generates a reset signal (POCRES) upon power application.
- Compares the supply voltage (V_{DD0}, V_{DD1}) and detection voltage (V_{POC0}), and generates a reset signal when V_{DD0}, $V_{DD1} < V_{POC0}$ (detection voltage (V_{POC0}): 3.7 V ±0.2 V).

Remark The V850E/IF3 and V850E/IG3 have the reset source flag register (RESF) that indicates generation of a reset signal (WDTRES) by watchdog timer overflow and a reset signal (LVIRES) by low-voltage detector (LVI).

The RESF register is not cleared to 00H when a reset signal (WDTRES or LVIRES) is generated, and its flag corresponding to the reset source is set to 1.

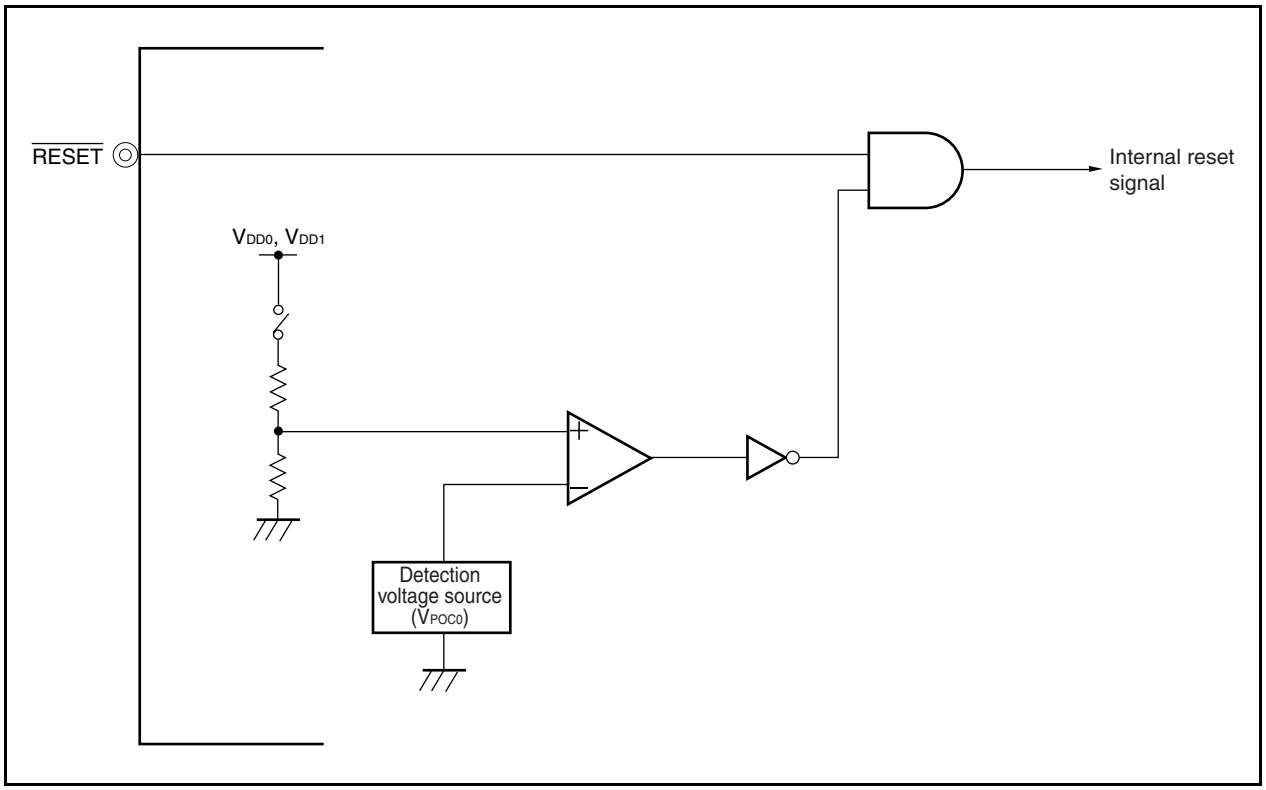
The RESF register is cleared (00H) when a reset signal (POCRES) by power-on-clear circuit (POC) is generated.

For details of the RESF register, see **CHAPTER 22 RESET FUNCTIONS**.

24.2 Configuration

The block diagram is shown below.

Figure 24-1. Block Diagram of Power-on-Clear Circuit



24.3 Operation

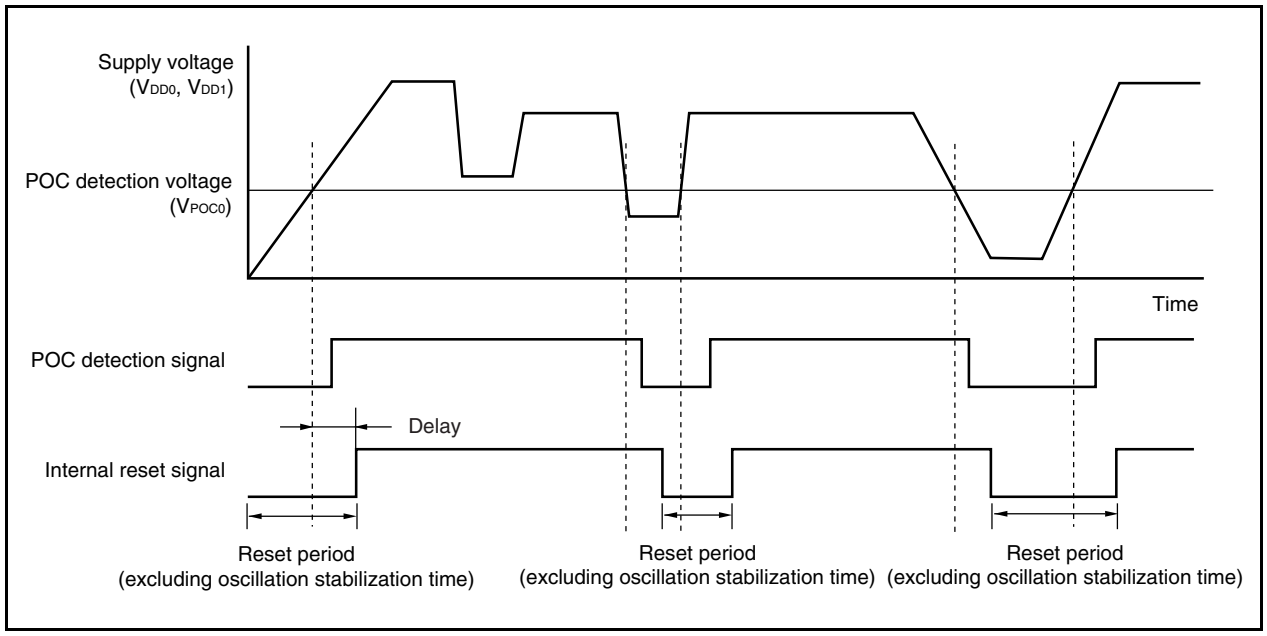
When the supply voltage and detection voltage are compared and if the supply voltage drops below the detection voltage (including at power application), the system is reset and each hardware is initialized to the specific status.

The system is reset from when low voltage is detected until the supply voltage becomes higher than the detection voltage. After a reset is released, when the oscillation stabilization time (default value of the OSTS register: 2^{14}/fx) of the oscillator has elapsed, the CPU starts executing the program.

The status of each hardware during the reset period and after reset release is the same as the reset operation by the $\overline{\text{RESET}}$ pin (see **22.3 (1) Reset operation by $\overline{\text{RESET}}$ pin input**).

The following shows the timing chart.

Figure 24-2. Timing of Reset Signal Generation by Power-on-Clear Circuit



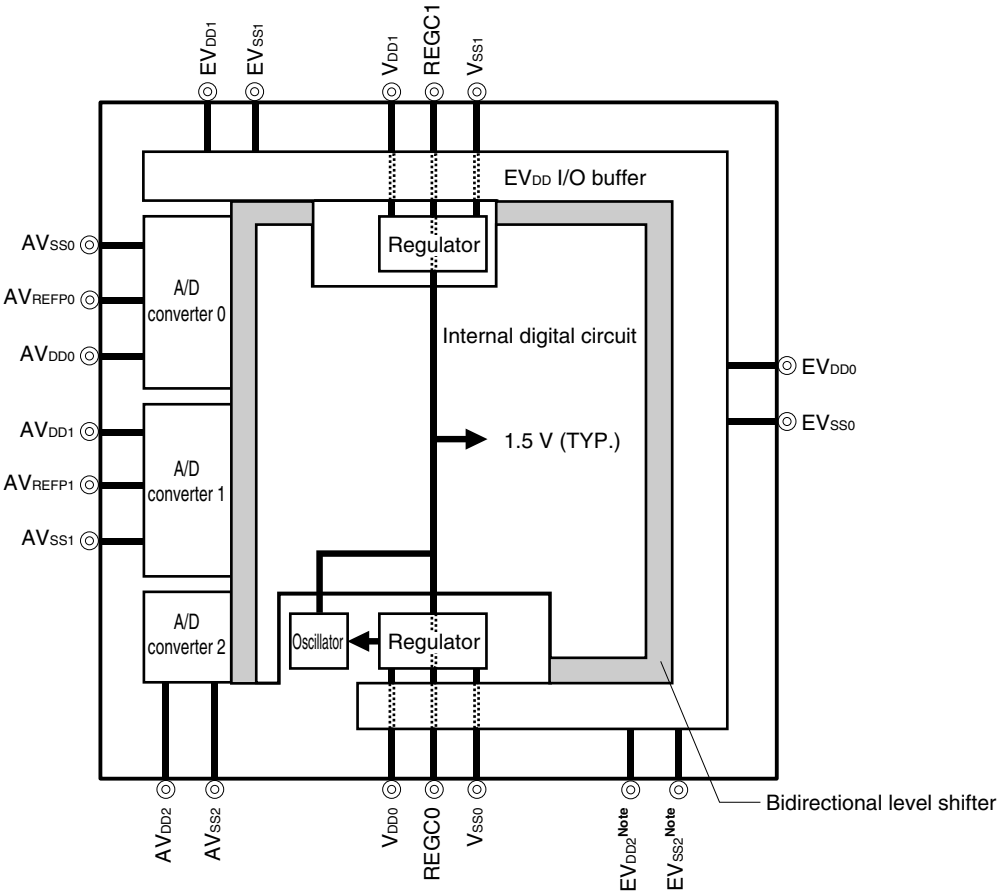
CHAPTER 25 REGULATOR

25.1 Overview

The V850E/IF3 and V850E/IG3 have an internal regulator to realize a 5 V single power supply operation.

This regulator supplies a stepped-down V_{DD0} and V_{DD1} power supply voltage to the oscillation block and internal logic circuits (except the A/D converters 0 to 2 and I/O buffers). The regulator output voltage (REGC0, REGC1 pins) is set to 1.5 V (TYP.).

Figure 25-1. Regulator



Note V850E/IG3 only

Caution Use the regulator with a setting of V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} (V850E/IG3 only) = AV_{DD0} = AV_{DD1} = AV_{DD2} = AV_{REF0} = AV_{REF1}.

25.2 Operation

The regulator of this product always operates in any mode (normal operation mode, HALT mode, IDLE mode, STOP mode, or during reset).

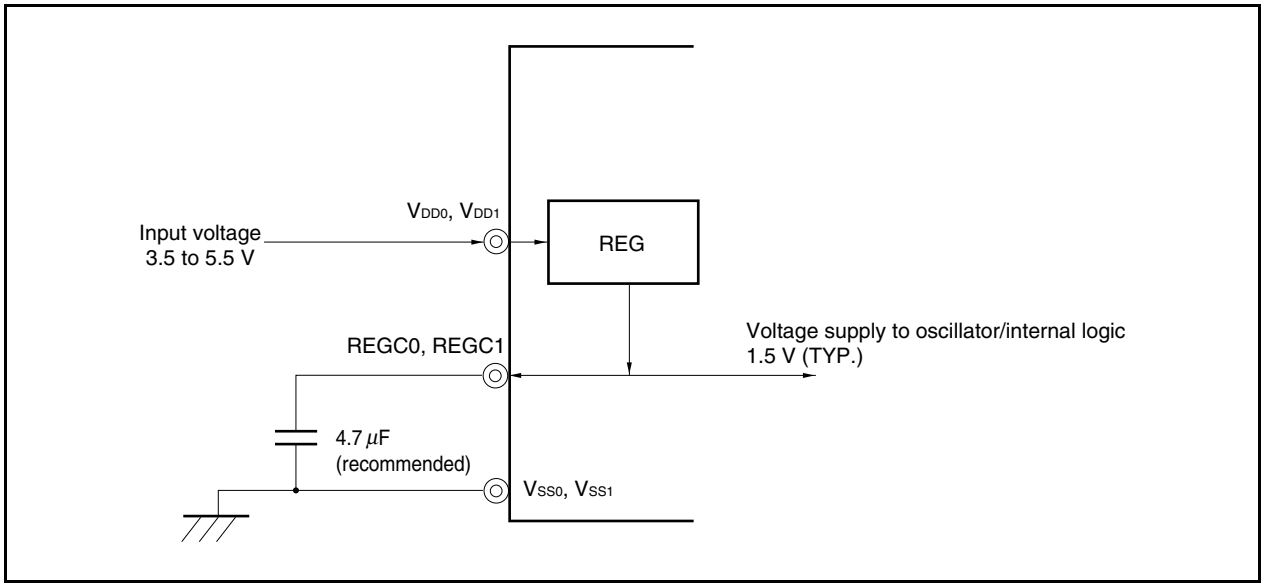
Be sure to connect a capacitor (4.7 μF (recommended value))[Note] to the REGC0 and REGC1 pins to stabilize the regulator output.

A diagram of the regulator pin connection method is shown below.

Note Use the low-ESR (0.5 Ω or lower) of the series resistance component ESR.

Caution The V850E/IF3 and V850E/IG3 have two regulators each. Therefore, connect a capacitor to each of the REGC0 and REGC1 pins.

Figure 25-2. Connection of REGC0 and REGC1 Pins



CHAPTER 26 ON-CHIP DEBUG FUNCTION

The on-chip debug function of the V850E/IF3 and V850E/IG3 can be realized in the following two ways.

- Debugging using DCU (debug control unit) (using MINICUBE)
  By using the $\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO pins as debug interface pins, on-chip debugging is realized by the internal DCU of the V850E/IG3[Note].
- Debugging without using DCU (using MINICUBE2)
  On-chip debugging is realized by MINICUBE2 without using the DCU but by using the user resources.

Note The V850E/IF3 does not have an internal DCU.

The following table shows the features of the two on-chip debug functions.

Table 26-1. On-Chip Debug Function Features

		Debugging Using DCU	Debugging Without Using DCU
Target product		V850E/IG3	V850E/IF3, V850E/IG3
Debug interface pins		$\overline{\text{DRST}}$, DCK, DMS, DDI, DDO	• When UARTA0 is used RXDA0, TXDA0 • When CSIB0 is used SIB0, SOB0, $\overline{\text{SCKB0}}$, HS (P43)
Securing of user resources		Not required	Required
Hardware break function		2 points	2 points (V850E/IG3 only)
Software break function	Internal ROM area	4 points	4 points
	RAM area	2000 points	2000 points
Real-time RAM monitor function[Note 1]		Available	Available
Dynamic memory modification (DMM) function[Note 2]		Available	Available
Mask function		Reset, INTWDT, $\overline{\text{WAIT}}$[Note 3]	$\overline{\text{RESET}}$, $\overline{\text{WAIT}}$[Note 3]
ROM security function		10-byte ID code authentication	10-byte ID code authentication
Hardware used		MINICUBE	MINICUBE2
Trace function		Not supported	Not supported
Debug interrupt interface function (DBINT)		Not supported	Not supported

Notes 1. This is a function which reads out memory contents during program execution.
2. This is a function which rewrites RAM contents during program execution.
3. μPD70F3454GC-8EA-A only

26.1 Debugging Using DCU

The program can be debugged by using the debug interface pins ($\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO) and connecting an on-chip debug simulator (MINICUBE).

Caution Only the V850E/IG3 has a DCU.

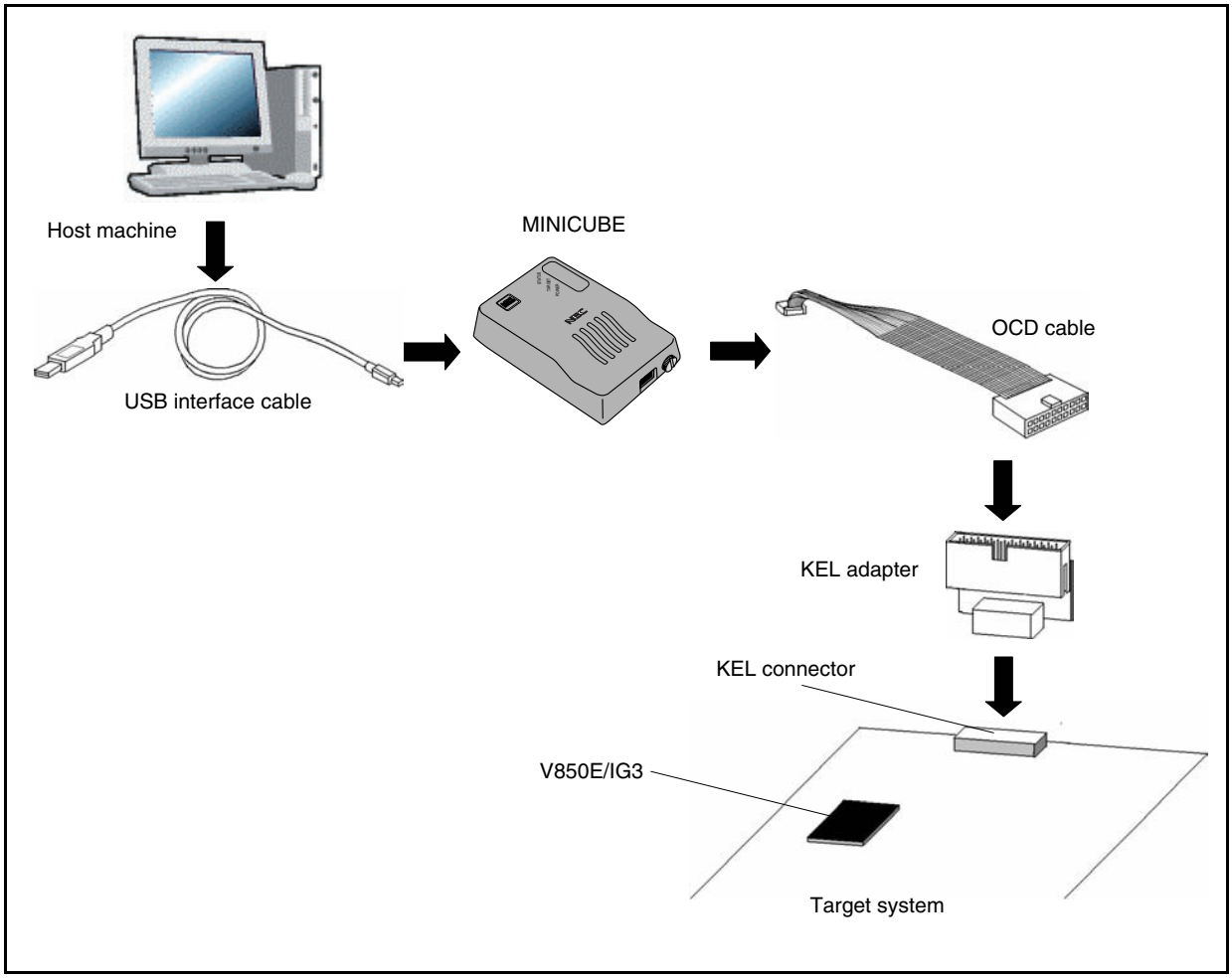
26.1.1 Circuit connection examples

When the MINICUBE is used, use of the following KEL connector is recommended.

○ Part number
- 8830E-026-170S: Straight type
- 8830E-026-170L: Right-angle type

It is necessary to mount an emulator and circuit for connection on the target system.

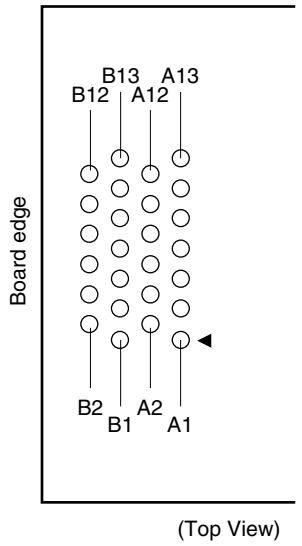
Figure 26-1. Connection Example of On-Chip Debug Emulator (MINICUBE)



(1) Pin configuration

The following figure shows the pin configuration of the emulator connector (on the target system side).

Figure 26-2. Pin Configuration of Emulator Connector (on Target System Side)



Caution Design the board based on the dimensions of the connector when actually mounting the connector on the board.

(2) Pin functions

The following table shows the pin functions of the emulator connector (on the target system side).

Table 26-2. Pin Functions of Emulator Connector (on Target System Side)

Pin No.	Pin Name	I/O	Pin Function
A1	(Reserved 1)	–	(Connect to GND)
A2	(Reserved 2)	–	(Connect to GND)
A3	(Reserved 3)	–	(Connect to GND)
A4	(Reserved 4)	–	(Connect to GND)
A5	(Reserved 5)	–	(Connect to GND)
A6	(Reserved 6)	–	(Connect to GND)
A7	DDI	Output	Data output for debug serial interface
A8	DCK	Output	Clock output for debug serial interface
A9	DMS	Output	Transfer mode select output for debug serial interface
A10	DDO	Input	Data input for debug serial interface
A11	$\overline{\text{DRST}}$	Output	DCU reset output
A12	(Reserved 7)	–	(Leave open)
A13	FLMD0	Output	Control signal for flash memory downloading
B1	GND	–	–
B2	GND	–	–
B3	GND	–	–
B4	GND	–	–
B5	GND	–	–
B6	GND	–	–
B7	GND	–	–
B8	GND	–	–
B9	GND	–	–
B10	GND	–	–
B11	PORT0_IN	–	(Connect to GND)
B12	PORT1_IN	–	(Connect to GND)
B13	V_{DD}	–	5 V input (for monitoring power application to target)

Cautions 1. The connection of the pins not supported in the V850E/IG3 depends on the emulator used.

2. The pattern on the target board must satisfy the following conditions.

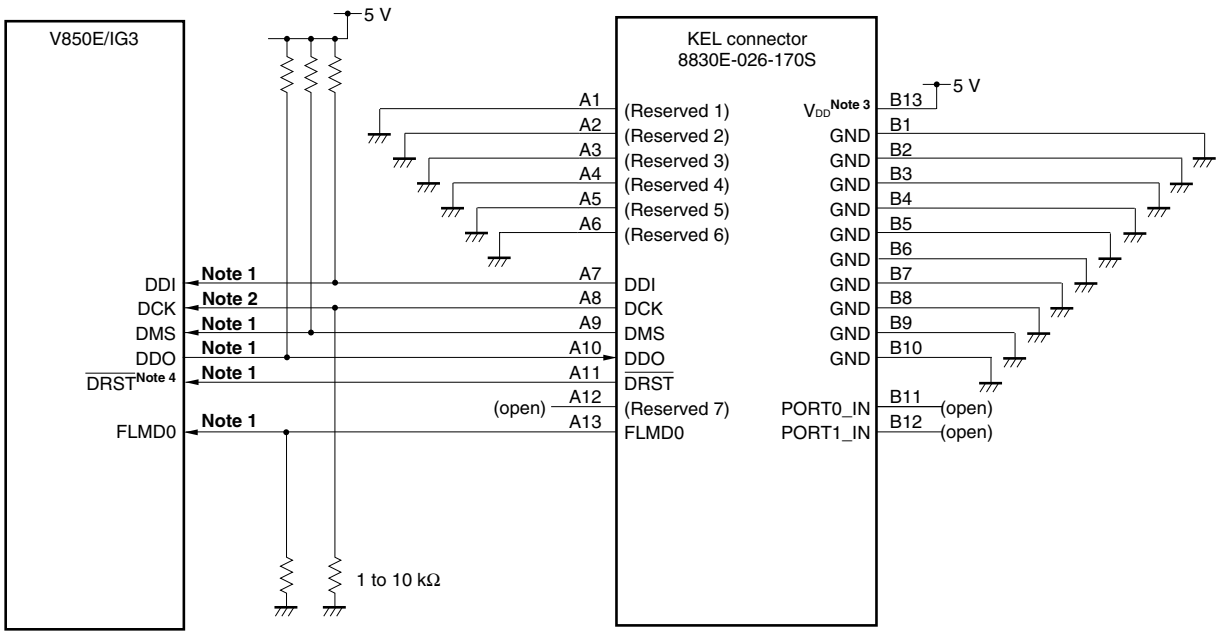
- **Keep the pattern length to within 100 mm.**
- **Shield the clock signal with GND.**

Remark Input/output is as viewed from the emulator side.

(3) Recommended circuit example

The following figure shows an example of the recommended circuit of the emulator connector (on the target system side).

Figure 26-3. Example of Recommended Connection of Emulator



Notes 1. Keep the pattern length to within 100 mm.

2. Shield the DCK signal with GND.

3. For detecting power supply to the target board.

4. When $\overline{\text{DRST}}$ pin is high level: On-chip debug mode
When $\overline{\text{DRST}}$ pin is low level: Normal operation mode
The $\overline{\text{DRST}}$ pin is internally pulled down in the V850E/IG3.

Caution The DDO signal is 5 V output, and the input level of the DDI, DCK, DMS, and $\overline{\text{DRST}}$ signals is TTL level.

26.1.2 Interface signals

The interface signals on the V850E/IG3 side are described below.

(1) $\overline{\text{DRST}}$

This is a reset input signal for the on-chip debug unit. It is a negative-logic signal that asynchronously initializes the debug control unit (DCU).

MINICUBE changes the level of the $\overline{\text{DRST}}$ signal from low to high for output and starts the on-chip debug unit of the V850E/IG3 when it detects VDD of the target system after the integrated debugger is started. If VDD is not detected from the target system, the output signals ($\overline{\text{DRST}}$, DCK, DMS, DDI, and FLMD0 pins) from the MINICUBE go into a high-impedance state.

When the $\overline{\text{DRST}}$ signal goes high, a reset signal is also generated in the V850E/IG3.

When starting debugging by starting the integrated debugger, a reset signal is always generated.

(2) DCK

This is a clock input signal. It supplies a 20 MHz clock from MINICUBE. In the on-chip debug unit, the DMS and DDI signals are sampled at the rising edge of the DCK signal, and the data DDO is output at its falling edge.

(3) DMS

This is a transfer mode select signal. The transfer status in the debug unit changes depending on the level of the DMS signal.

(4) DDI

This is a data input signal. It is sampled in the on-chip debug unit at the rising edge of DCK.

(5) DDO

This is a data output signal. It is output from the on-chip debug unit at the falling edge of the DCK signal.

(6) FLMD0

The flash self programming function is used for the function to download data to the flash memory via the integrated debugger. During flash self programming, the FLMD0 pin must be kept high. In addition, connect a pull-down resistor to the FLMD0 pin.

The FLMD0 pin can be controlled in either of the following two ways.

<1> To control from MINICUBE

Connect the FLMD0 signal of MINICUBE to the FLMD0 pin of the V850E/IG3.

In the normal mode, nothing is driven by MINICUBE (high impedance).

During a break, MINICUBE raises the FLMD0 pin to the high level when the download function of the integrated debugger is executed.

<2> To control from port

Connect any port of the device to the FLMD0 pin of the V850E/IG3.

The same port as the one used by the user program to realize the flash self programming function may be used.

On the console of the integrated debugger, make a setting to raise the port pin to high level before executing the download function, or lower the port pin after executing the download function.

For details, refer to the **ID850QB Ver. 3.20 Integrated Debugger Operation User's Manual (U17964E)**.

26.1.3 Maskable functions

Reset, INTWDT, and $\overline{\text{WAIT}}$[Note] signals can be masked.

The maskable functions with the debugger (ID850QB) and the corresponding functions are shown below.

Note μPD70F3454GC-8EA-A only

Table 26-3. Maskable Functions

Maskable Functions with Debugger (ID850QB)	Corresponding Function of V850E/IG3
NMI0	Non-maskable interrupt request signal (INTWDT) generation
NMI1	×
NMI2	×
STOP	×
HOLD	×
RESET	$\overline{\text{RESET}}$ pin input, reset signal (WDTRES) generation by watchdog timer overflow, reset signal (LVIRES) generation by low-voltage detector (LVI), reset signal (POCRES) generation by power-on-clear circuit (POC)
WAIT	$\overline{\text{WAIT}}$ pin[Note] input

Note μPD70F3454GC-8EA-A only

26.1.4 Cautions

(1) If a reset signal is input (from the target system or a reset signal from an internal reset source) during RUN (program execution), the break function may malfunction.

(2) Even if the reset signal is masked by the mask function, the I/O buffer (port pin) may be reset if a reset signal is input from a pin.

(3) Because a software breakpoint set in the on-chip flash memory is made temporarily invalid by reset signal ($\overline{\text{RESET}}$ pin input, reset signal (WDTRES) generation by watchdog timer overflow, reset signal (LVIRES) generation by low-voltage detector (LVI), or reset signal (POCRES) generation by power-on-clear circuit (POC)). The breakpoint becomes valid again when a hardware break or forced break occurs, but a software break does not occur until then.

(4) Pin reset during a break is masked and the CPU and peripheral I/O are not reset. If pin reset or internal reset is generated as soon as the flash memory is rewritten by DMA or read by the RAM monitor function while the user program is being executed, the CPU and peripheral I/O may not be correctly reset.

(5) In the on-chip debug mode, the DDO pin is forcibly set to the high-level output.

(6) The flash memory of the device used in debugging is rewritten during debugging, so the number of flash memory rewrites cannot be guaranteed. Therefore, do not use the device used in debugging for a mass production product.

(7) Because the DDI and DCK pins function alternately as the CSIB0 I/O pins (SOB0, $\overline{\text{SCKB0}}$), UARTA0 output pin (TXDA0), and external interrupt pin (INTP13), CSIB0, UARTA0, and INTP13 cannot be used while the on-chip debug function is being used.

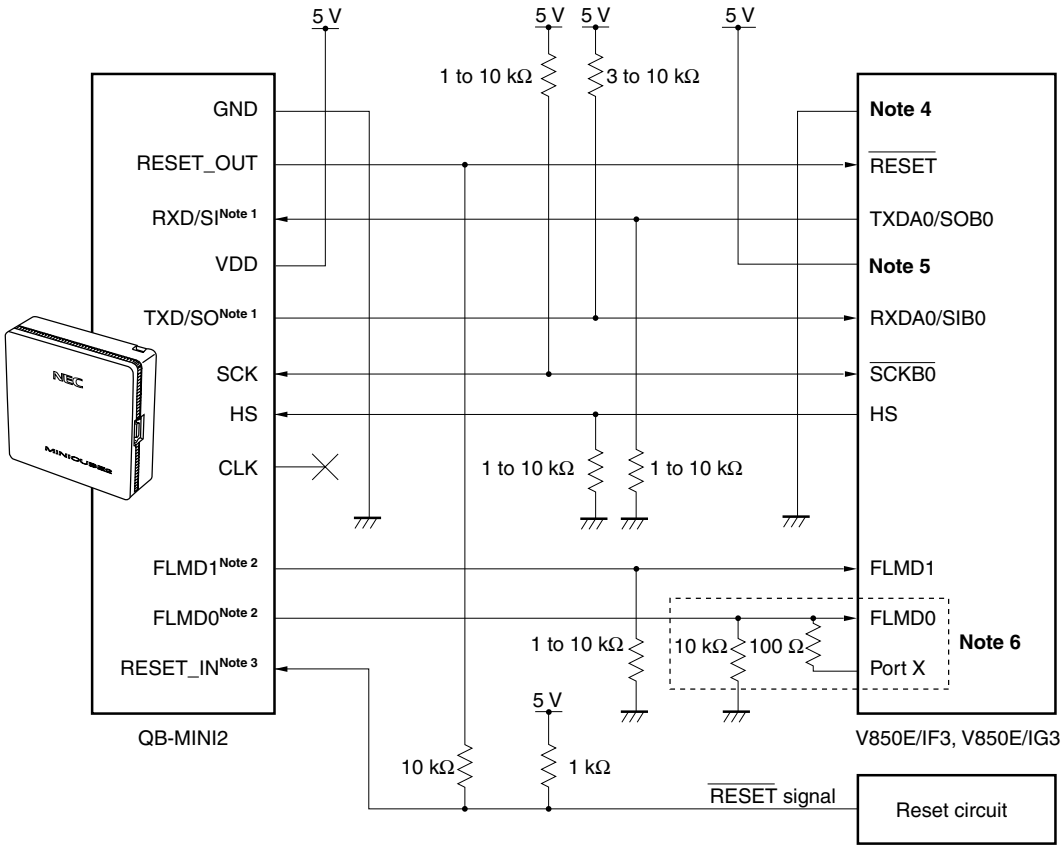
(8) When the on-chip debug function is used, the clock generator and PLL continue operating even if the STOP mode is set.

26.2 Debugging Without Using DCU

The following describes how to implement an on-chip debug function using MINICUBE2 with the UARTA0 pins (RXDA0, TXDA0) or CSIB0 pins (SIB0, SOB0, $\overline{\text{SCKB0}}$, HS (P43)) as debug interfaces, without using the DCU.

26.2.1 Circuit connection examples

Figure 26-4. Circuit Connection Example When UARTA0/CSIB0 Is Used for Communication Interface



Notes 1. Connect TXDA0/SOB0 (transmit side) of the V850E/IF3 and V850E/IG3 to RXD/SI (receive side) of the target connector, and TXD/SO (transmit side) of the target connector to RXDA0/SIB0 (receive side) of the V850E/IF3 and V850E/IG3.

2. The V850E/IF3, V850E/IG3-side pin connected to this pin (FLMD0, FLMD1) can be used as an alternate-function pin other than while the memory is rewritten during a break in debugging, because this pin is in Hi-Z state.

3. This connection is designed assuming that the $\overline{\text{RESET}}$ signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less).

4. EV_{SS0}, EV_{SS1}, EV_{SS2} (V850E/IG3 only), V_{SS0}, V_{SS1}, AV_{SS0}, AV_{SS1}, AV_{SS2}

5. EV_{DD0}, EV_{DD1}, EV_{DD2} (V850E/IG3 only), V_{DD0}, V_{DD1}, AV_{DD0}, AV_{DD1}, AV_{DD2}

6. The circuit enclosed by broken lines is designed for flash self programming, which controls the FLMD0 pin via ports. Use the port for inputting or outputting the high level. When flash self programming is not performed, a pull-down resistance for the FLMD0 pin can be within 1 to 10 kΩ.

Remark See **Table 26-4** for pins used when UARTA0 or CSIB0 is used for communication interface.

Table 26-4. Wiring Between V850E/IF3, V850E/IG3, and MINICUBE2 (1/2)

Pin Configuration of MINICUBE2 (QB-MINI2)			When UARTA0 Used			
Signal Name	I/O	Pin Function	Pin Name	Pin No.		
				IF3	IG3	
				GC	GC	GF
SI/RxD	Input	Pin to receive commands and data from V850E/IF3 and V850E/IG3	TXDA0	39	48	76
SO/TxD	Output	Pin to transmit commands and data to V850E/IF3 and V850E/IG3	RXDA0	38	47	75
SCK	Output	Clock output pin for 3-wire serial communication	Not needed	Not needed	Not needed	Not needed
CLK	Output	Clock output pin to V850E/IF3 and V850E/IG3	Not needed	Not needed	Not needed	Not needed
			Not needed	Not needed	Not needed	Not needed
RESET_OUT	Output	Reset output pin to V850E/IF3 and V850E/IG3	$\overline{\text{RESET}}$	35	40	68
FLMD0	Output	Output pin to set V850E/IF3 and V850E/IG3 to debug mode or programming mode	FLMD0	37	46	74
FLMD1	Output	Output pin to set programming mode	FLMD1	60	76	4
HS	Input	Handshake signal for CSI0 + HS communication	Not needed	Not needed	Not needed	Not needed
GND	–	Ground	V_{SS0}	32	37	65
			V_{SS1}	66	85	13
			AV_{SS0}	6	7	35
			AV_{SS1}	11	12	40
			AV_{SS2}	22	27	55
			EV_{SS0}	54	64	92
			EV_{SS1}	79	1	29
			EV_{SS2}	Not needed	42	70
RESET_IN	Input	Reset input pin on the target system				

Table 26-4. Wiring Between V850E/IF3, V850E/IG3, and MINICUBE2 (2/2)

Pin Configuration of MINICUBE2 (QB-MINI2)			When CSIB0-HS Used			
Signal Name	I/O	Pin Function	Pin Name	Pin No.		
				IF3	IG3	
				GC	GC	GF
SI/RxD	Input	Pin to receive commands and data from V850E/IF3 and V850E/IG3	SOB0	39	48	76
SO/TxD	Output	Pin to transmit commands and data to V850E/IF3 and V850E/IG3	SIB0	38	47	75
SCK	Output	Clock output pin for 3-wire serial communication	$\overline{\text{SCKB0}}$	40	49	77
CLK	Output	Clock output pin to V850E/IF3 and V850E/IG3	Not needed	Not needed	Not needed	Not needed
			Not needed	Not needed	Not needed	Not needed
RESET_OUT	Output	Reset output pin to V850E/IF3 and V850E/IG3	$\overline{\text{RESET}}$	35	40	68
FLMD0	Output	Output pin to set V850E/IF3 and V850E/IG3 to debug mode or programming mode	FLMD0	37	46	74
FLMD1	Output	Output pin to set programming mode	FLMD1	60	76	4
HS	Input	Handshake signal for CSI0 + HS communication	P43	41	50	78
GND	–	Ground	V_{SS0}	32	37	65
			V_{SS1}	66	85	13
			AV_{SS0}	6	7	35
			AV_{SS1}	11	12	40
			AV_{SS2}	22	27	55
			EV_{SS0}	54	64	92
			EV_{SS1}	79	1	29
			EV_{SS2}	Not needed	42	70
RESET_IN	Input	Reset input pin on the target system				

26.2.2 Maskable functions

Reset signal can only be masked.

The maskable functions with the debugger (ID850QB) and the corresponding functions are shown below.

Table 26-5. Maskable Functions

Maskable Functions with Debugger (ID850QB)	Corresponding Function of V850E/IF3, V850E/IG3
NMI0	×
NMI1	×
NMI2	×
STOP	×
HOLD	×
RESET	Reset signal generation by $\overline{\text{RESET}}$ pin input
WAIT	$\overline{\text{WAIT}}$ pin[Note] input

Note μPD70F3454GC-8EA-A only

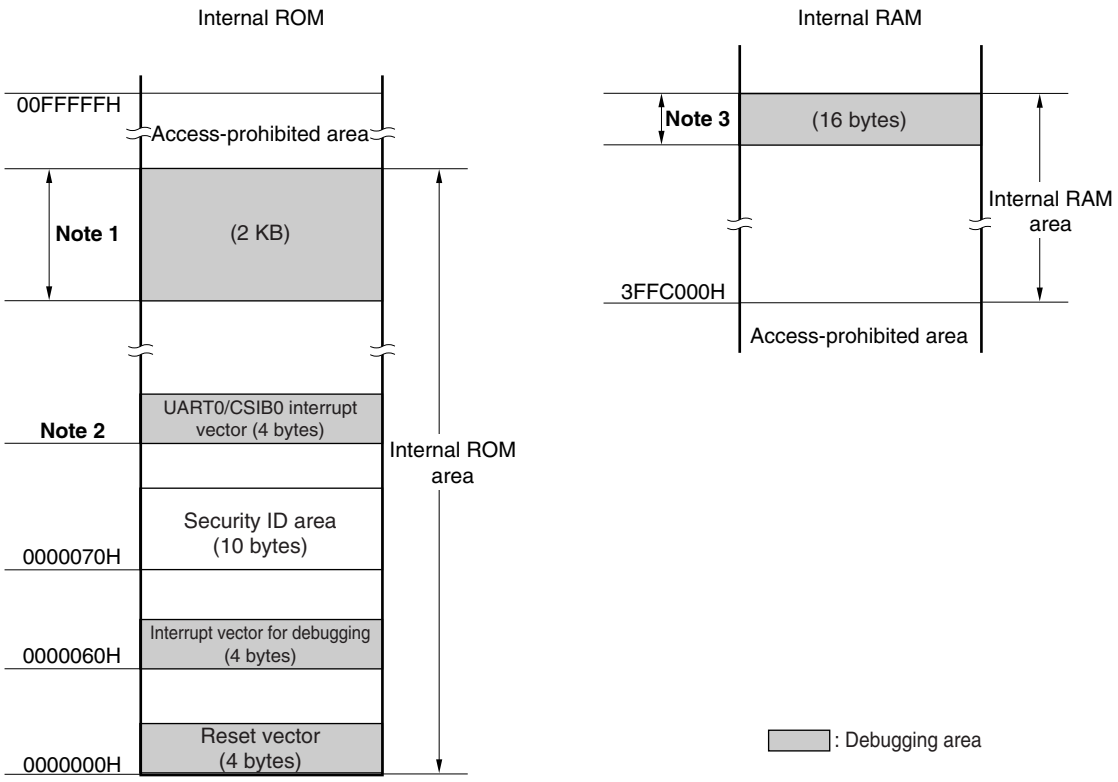
26.2.3 Securing of user resources

The user must prepare the following to perform communication between MINICUBE2 and the V850E/IF3, V850E/IG3 and implement each debug function. These items need to be set in the user program or using the compiler options.

(1) Securement of memory space

The shaded portions in Figure 26-5 are the areas reserved for placing the debug monitor program, so user programs and data cannot be allocated in these spaces. These spaces must be secured so as not to be used by the user program.

Figure 26-5. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address values vary depending on the product.

	Internal ROM size	Debugging area
μPD70F3451 (V850E/IF3)	128 KB	001F800H to 001FFFFH
μPD70F3453 (V850E/IG3)		
μPD70F3452 (V850E/IF3)	256 KB	003F800H to 003FFFFH
μPD70F3454 (V850E/IG3)		

2. Start address values when UARTA0 and CSIB0 are used are as follows.

Target serial interface	Interrupt name	Start address
UARTA0	INTUA0RE	000004B0H
	INTUA0R	000004C0H
	INTUA0T	000004D0H
CSIB0	INTCB0RE	000004E0H
	INTCB0R	000004F0H
	INTCB0T	00000500H

3. Address values vary depending on the product.

	Internal RAM size	Debugging area
μPD70F3451 (V850E/IF3)	8 KB	3FFDFF0H to 3FFDFFFH
μPD70F3453 (V850E/IG3)		
μPD70F3452 (V850E/IF3)	12 KB	3FFEFF0H to 3FFEFFFH
μPD70F3454 (V850E/IG3)		

- Security ID setting

  The ID code must be embedded in the area between 0000070H and 0000079H in Figure 26-5, to prevent the memory from being read by an unauthorized person. For details, see **26.3 ROM Security Function**.

(2) Reset vector

A reset vector includes the jump instruction for the debug monitor program.

[How to secure areas]

It is not necessary to secure this area intentionally. When downloading a program, however, the debugger rewrites the reset vector in accordance with the following cases. If the rewritten pattern does not match the following cases, the debugger generates an error (F0c34 when using the ID850QB).

(a) When two nop instructions are placed in succession from address 0

Before rewriting		After rewriting
0x0 nop	→	Jumps to debug monitor program at 0x0
0x2 nop		0x4 xxxx
0x4 xxxx		

(b) When two 0xFFFF are successively placed from address 0 (already erased device)

Before rewriting		After rewriting
0x0 0xFFFF	→	Jumps to debug monitor program at 0x0
0x2 0xFFFF		0x4 xxxx
0x4 xxxx		

(c) The *jr* instruction is placed at address 0 (when using CA850)

Before rewriting		After rewriting
0x0 jr disp22	→	Jumps to debug monitor program at 0x0
		0x4 jr disp22 - 4

(d) *mov32* and *jmp* are placed in succession from address 0 (when using IAR compiler ICCV850)

Before rewriting		After rewriting
0x0 mov imm32,reg1	→	Jumps to debug monitor program at 0x0
0x6 jmp [reg1]		0x4 mov imm32,reg1
		0xa jmp [reg1]

(e) The jump instruction for the debug monitor program is placed at address 0

Before rewriting		After rewriting
Jumps to debug monitor program at 0x0	→	No change

(3) Securement of area for debug monitor program

The shaded portions in Figure 26-5 are the areas where the debug monitor program is allocated. The monitor program performs initialization processing for debug communication interface and RUN or break processing for the CPU. The internal ROM area must be filled with 0xFF. This area must not be rewritten by the user program.

[How to secure areas]

It is not necessarily required to secure this area if the user program does not use this area.

To avoid problems that may occur during the debugger startup, however, it is recommended to secure this area in advance, using the compiler.

The following shows examples for securing the area, using the NEC Electronics compiler CA850. Add the assemble source file and link directive code, as shown below.

- Assemble source (Add the following code as an assemble source file.)

```
-- Secures 2 KB space for monitor ROM section
.section "MonitorROM", const
.space   0x800, 0xff

-- Secures interrupt vector for debugging
.section "DBG0"
.space   4, 0xff

-- Secures interrupt vector for serial communication
-- Change the section name according to the serial communication mode used
.section "INTCB0RE"
.space   4, 0xff
.section "INTCB0R"
.space   4, 0xff
.section "INTCB0T"
.space   4, 0xff

-- Secures 16-byte space for monitor RAM section
.section "MonitorRAM", bss
.lcomm   monitorramsym, 16, 4;    -- defines symbol monitorramsym
```

- Link directive (Add the following code to the link directive file.)

  The following shows an example when the internal ROM has 256 KB (end address is 003FFFFH) and internal RAM has 12 KB (end address is 3FFEFFFH).

```
MROMSEG       : !LOAD ?R V0x03f800{
              MonitorROM    = $PROGBITS   ?A MonitorROM;
};

MRAMSEG       : !LOAD ?RW V0x03ffeff0{
              MonitorRAM    = $NOBITS     ?AW MonitorRAM;
};
```

(4) Securement of communication serial interface

UARTA0 or CSIB0 is used for communication between MINICUBE2 and the V850E/IF3, V850E/IG3. The settings related to the serial interface modes are performed by the debug monitor program, but if the setting is changed by the user program, a communication error may occur.

To prevent such a problem from occurring, communication serial interface must be secured in the user program.

[How to secure communication serial interface]

- Serial interface registers
  Do not set the registers related to UARTA0 and CSIB0 in the user program.

- Interrupt mask register
  When UARTA0 is used, do not mask the reception end interrupt (INTUA0R). When CSIB0 is used, do not mask the reception end interrupt (INTCB0R).

(a) When UARTA0 is used

	7	6	5	4	3	2	1	0
UA0RIC	×	0	×	×	×	×	×	×

(b) When CSIB0 is used

	7	6	5	4	3	2	1	0
CB0RIC	×	0	×	×	×	×	×	×

Remark ×: don't care

- Port registers when UARTA0 is used
  When UARTA0 is used for communication, port registers are set to make the TXDA0 and RXDA0 pins valid by the debug monitor program. Do not change the following register settings with the user program during debugging. (The same value can be overwritten.)

	7	6	5	4	3	2	1	0
PFCE4	×	×	×	×	×	×	0	0
PFC4	×	×	×	×	×	×	1	1
PMC4	×	×	×	×	×	×	1	1

Remark ×: don't care

- Port registers when CSIB0 is used

  When CSIB0 is used, port registers are set to make the SIB0, SOB0, $\overline{\text{SCKB0}}$, and HS (P43) pins valid by the debug monitor program. Do not change the following register settings with the user program during debugging. (The same value can be overwritten.)

(a) SIB0, SOB0, and $\overline{\text{SCKB0}}$ settings

	7	6	5	4	3	2	1	0
PFCE4	×	×	×	×	×	0	0	0

	7	6	5	4	3	2	1	0
PFC4	×	×	×	×	×	0	0	0

	7	6	5	4	3	2	1	0
PMC4	×	×	×	×	×	1	1	1

(b) HS (P43 pin) settings

	7	6	5	4	3	2	1	0
PM4	×	×	×	×	×	×	×	0

	7	6	5	4	3	2	1	0
P4	×	×	×	×	×	×	×	**Note**

Note Writing to this bit is prohibited.

The values corresponding to the HS pin are changed by the monitor program according to the debugger status. To perform port register settings in 8-bit units, the user program can usually use read-modify-write. If an interrupt for debugging occurs before writing, however, an unexpected operation may be performed.

Remark ×: don't care

26.2.4 Cautions

(1) Handling of device that was used for debugging

Do not mount a device that was used for debugging on a mass-produced product, because the flash memory was rewritten during debugging and the number of rewrites of the flash memory cannot be guaranteed. Moreover, do not embed the debug monitor program into mass-produced products.

(2) When breaks cannot be executed

Forced breaks cannot be executed if one of the following conditions is satisfied.

- Interrupts are disabled (DI)
- Interrupts issued for the serial interface, which is used for communication between MINICUBE2 and the V850E/IF3, V850E/IG3, are masked
- Standby mode is entered while standby release by a maskable interrupt is prohibited
- Mode for communication between MINICUBE2 and the V850E/IF3, V850E/IG3 is UARTA0, and the peripheral clock has been stopped

(3) When pseudo real-time RAM monitor (RRM) function and DMM function do not operate

The pseudo RRM function and DMM function do not operate if one of the following conditions is satisfied.

- Interrupts are disabled (DI)
- Interrupts issued for the serial interface, which is used for communication between MINICUBE2 and the V850E/IF3, V850E/IG3, are masked
- Standby mode is entered while standby release by a maskable interrupt is prohibited
- Mode for communication between MINICUBE2 and the V850E/IF3, V850E/IG3 is UARTA0, and the peripheral clock has been stopped
- Mode for communication between MINICUBE2 and the V850E/IF3, V850E/IG3 is UARTA0, and a clock different from the one specified in the debugger is used for communication

(4) Standby release with pseudo RRM and DMM functions enabled

The standby mode is released by the pseudo RRM function and DMM function if one of the following conditions is satisfied.

- Mode for communication between MINICUBE2 and the V850E/IF3, V850E/IG3 is CSIB0
- Mode for communication between MINICUBE2 and the V850E/IF3, V850E/IG3 is UARTA0, and the peripheral clock has not stopped.

(5) Writing to peripheral I/O registers that requires a specific sequence, using DMM function

Peripheral I/O registers that requires a specific sequence cannot be written with the DMM function.

(6) Flash self programming

If a space where the debug monitor program is allocated is rewritten by flash self programming, the debugger can no longer operate normally.

26.3 ROM Security Function

26.3.1 Security ID

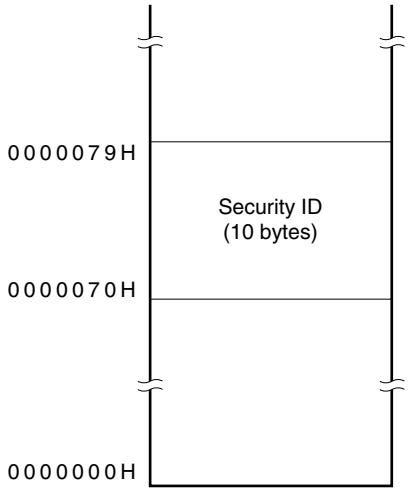
The flash memory versions of the V850E/IF3 and V850E/IG3 perform authentication using a 10-byte ID code to prevent the contents of the flash memory from being read by an unauthorized person during on-chip debugging by the on-chip debug emulator.

Set the ID code in the 10-byte on-chip flash memory area from 0000070H to 0000079H to allow the debugger perform ID authentication.

If the IDs match, the security is released and reading flash memory and using the on-chip debug emulator are enabled.

- Set the 10-byte ID code to 0000070H to 0000079H.
- Bit 7 of 0000079H is the on-chip debug emulator enable flag.
  (0: Disable, 1: Enable)
- When the on-chip debug emulator is started, the debugger requests ID input. When the ID code input on the debugger and the ID code set in 0000070H to 0000079H match, the debugger starts.
- Debugging cannot be performed if the on-chip debug emulator enable flag is 0, even if the ID codes match.

Figure 26-6. Security ID Area



Caution After the flash memory is erased, 1 is written to the entire area.

26.3.2 Setting

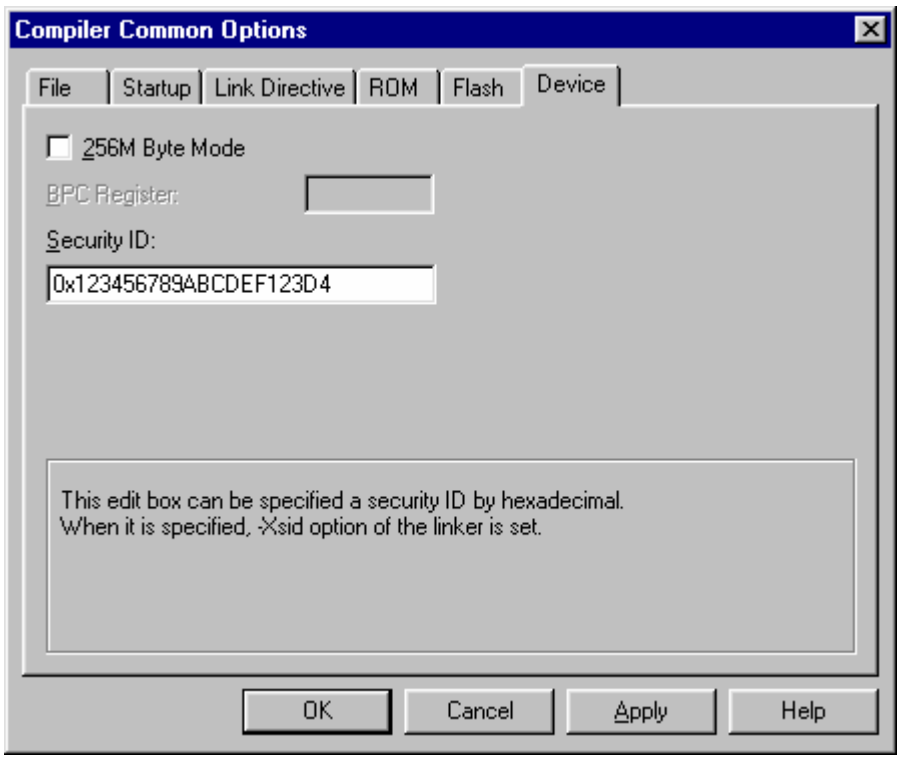
The following shows how to set the ID code as shown in Table 26-6.

When the ID code is set as shown in Table 26-6, the ID code input in the configuration dialog box of the ID850QB is “123456789ABCDEF123D4” (the ID code is case-insensitive).

Table 26-6. ID Code

Address	Value
0x70	0x12
0x71	0x34
0x72	0x56
0x73	0x78
0x74	0x9A
0x75	0xBC
0x76	0xDE
0x77	0XF1
0x78	0x23
0x79	0xD4

The ID code can be specified for the device file that supports CA850 Ver. 3.10 or later and the security ID using the PM+ compiler common option setting.



[Program example (when using CA850 Ver. 3.10 or later)]

```
#---------------------------------------
#       SECURITYID
#---------------------------------------
        .section    "SECURITY_ID"   --Interrupt handler address 0x70
        .word       0x78563412      --0-3 byte code
        .word       0xF1DEBC9A      --4-7 byte code
        .hword      0xD423          --8-9 byte code
```

Remark Add the above program example to the startup files.

CHAPTER 27 FLASH MEMORY

27.1 Features

- ○ All area batch erase or erase in block units (2 KB)
- ○ Communication through serial interface from the flash programmer
- ○ Erase/write voltage: Erase/write is possible with a single power supply
- ○ On-board programming
- ○ Flash memory self programming possible
- ○ Secure rewriting of entire flash memory area by self programming using boot swap function
- ○ Rewriting method
  - Rewriting by communication with flash programmer via serial interface (on-board/off-board programming)
  - Rewriting flash memory by user program (self programming)
- ○ Rewriting flash memory and read disable function supported (security enforced)
- ○ Interrupts can be acknowledged during self programming.

Table 27-1. Rewrite Method

Rewrite Method	Functional Outline	Operation Mode
On-board programming	Flash memory can be rewritten after the device is mounted on the target system, by using a dedicated flash programmer.	Flash memory programming mode
Off-board programming	Flash memory can be rewritten before the device is mounted on the target system, by using a dedicated flash programmer and a dedicated program adapter board (FA series).	
Self programming	Flash memory can be rewritten by executing a user program that has been written to the flash memory in advance by means of on-board/off-board programming. (During self programming, instructions cannot be fetched from or data access cannot be made to the on-chip flash memory area. Therefore, the rewrite program must be transferred to the internal RAM or external memory (μPD70F3454GC-8EA-A only) in advance).	Normal operation mode

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 27-2. Basic Functions

Function	Functional Outline	Support (√: Supported, ×: Not supported)	
		On-Board/Off-Board Programming	Self Programming
Block erasure	The contents of specified memory blocks are erased.	√	√
Chip erasure	The contents of the entire memory area are erased all at once.	√	× (supported by specifying area for block erasure)
Write	Writing to specified addresses, and a verify check to see if write level is secured are performed.	√	√
Verify/checksum	Data read from the flash memory is compared with data transferred from the flash programmer.	√	× (Can be read by user program)
Blank check	The erasure status of the entire memory is checked.	√	√
Security setting	Use of the block erase command, chip erase command, program command, and read command can be prohibited.	√	× (Only values set by on-board/off-board programming can be retained)

Table 27-3. Security Functions

Function	Function Outline	Support	
		On-Board/Off-Board Programming	Self Programming
Block erase command prohibit	Execution of a block erase command on all blocks is prohibited. Setting of prohibition can be initialized by execution of a chip erase command.	For details, see **27.1.2 Security function**.	
Chip erase command prohibit	Execution of block erase and chip erase commands on all blocks is prohibited. Once prohibition is set, setting of prohibition cannot be initialized because the chip erase command cannot be executed.		
Program command prohibit	Write and block erase commands on all blocks are prohibited. Setting of prohibition can be initialized by execution of the chip erase command.		
Read command prohibit	Read command on all blocks is prohibited. Setting of prohibition can be initialized by execution of a chip erase command.		

128 KB or 256 KB of on-chip flash memory is provided in the V850E/IF3 and V850E/IG3.

- μPD70F3451 (V850E/IF3), 70F3453 (V850E/IG3): 128 KB on-chip flash memory version
- μPD70F3452 (V850E/IF3), 70F3454 (V850E/IG3): 256 KB on-chip flash memory version

Flash memory can be rewritten with the flash programmer or using the self programming mode.

Writing to the flash memory can be performed with the flash programmer that is connected to the target system.

Writing in the self programming mode can be performed with an application program, without using the flash programmer.

Flash memory versions are commonly used in the following development environments and mass production applications.

- For altering software after the V850E/IF3 and V850E/IG3 is soldered onto the target system.
- For differentiating software according to the specification in small scale production of various models.
- For data adjustment when starting mass production.

27.1.1 Erase units

(1) All area batch erase

Flash memory area 128 KB or 256 KB can be erased at the same time.

(2) Erase in block units

Can be erased in block units.

- μPD70F3451 (V850E/IF3), 70F3453 (V850E/IG3): Block 0 to block 63: Each 2 KB
- μPD70F3452 (V850E/IF3), 70F3454 (V850E/IG3): Block 0 to block 127: Each 2 KB

27.1.2 Security function

The commands and functions can be secured when the flash memory is rewritten.

As a factory-set condition in the V850E/IF3 and V850E/IG3, “All enabled” is selected and the flash memory to which nothing has been written is secured.

(1) In flash memory programming mode

Security Setting (Flag) / Command	All Enabled	Reading Prohibited	Writing Prohibited	Chip Erase Prohibited	Block Erase Prohibited	Boot Block Cluster Rewriting Prohibited
Read	√	×	√	√	√	√
Write	√	√	×	√	√	△
Chip erase	√	√	√	×	√	×
Block erase	√	√	×	×	×	△
Changing of security setting	√	**Note 1**	**Note 1**	**Note 1**	**Note 1**	**Notes 1, 2**
Other commands (such as blank check and verify)	√	√	√	√	√	√

√: Command can be accepted.

△: Command can be accepted in areas other than boot block cluster.

×: Protect error

Notes 1. Enabled setting can be disabled. However, a protect error occurs if disabled setting is enabled. To enable security that has been disabled, the chip must be erased. If chip erase prohibition or boot block cluster rewriting prohibition is selected, the security setting cannot be enabled.

2. If boot block cluster rewriting prohibition is set, the boot block cluster last block number (128 KB version: 63, 256 KB version: 127) cannot be changed.

(2) In self programming mode

Security Setting (Flag) / Command	All Enabled	Reading Prohibited	Writing Prohibited	Chip Erase Prohibited	Block Erase Prohibited	Boot Block Cluster Rewriting Prohibited
FlashWordRead	√	√	√	√	√	√
FlashWordWrite	√	√	√	√	√	△
FlashBlockErase	√	√	√	√	√	△
FlashBootSwap	√	√	√	√	√	×
FlashSetInfo	√	**Note 1**	**Note 1**	**Note 1**	**Note 1**	**Notes 1, 2**
Other functions (such as FlashBlockBlankCheck and FlashBlockVerify)	√	√	√	√	√	√

√: Command can be accepted.

△: Function can be executed in an area other than boot block cluster.

×: Protect error

Notes 1. Enabled setting can be disabled. However, a protect error occurs if disabled setting is enabled. To enable security that has been disabled, the chip must be erased in the flash self programming mode. If chip erase prohibition is selected, the security setting cannot be enabled.

2. If boot block cluster rewriting prohibition is set, the boot block cluster last block number (128 KB version: 63, 256 KB version: 127) and boot swap cluster setting flag cannot be changed.

27.2 Writing with Flash Programmer

Writing can be performed either on-board or off-board using a flash programmer (PG-FP4, FL-PR4) and MINICUBE2.

(1) On-board programming

The contents of the flash memory are rewritten after the V850E/IF3 or V850E/IG3 is mounted on the target system. Mount connectors, etc., on the target system to connect the flash programmer.

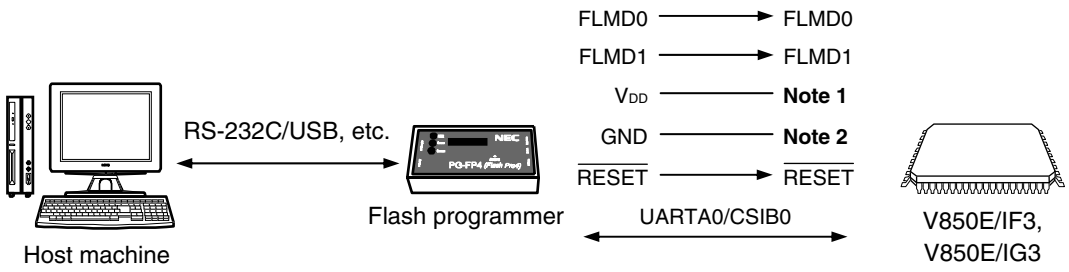
(2) Off-board programming

Writing to a flash memory is performed before mounting the V850E/IF3 or V850E/IG3 on the target system.

Remark FL-PR4 is a product of Naito Densei Machida Mfg. Co., Ltd.

27.3 Flash Memory Programming Environment

The following shows the environment required for writing programs to the flash memory of the V850E/IF3 and V850E/IG3.



Notes 1. V_{DD0}, V_{DD1}, EV_{DD0}, EV_{DD1}, EV_{DD2} (V850E/IG3 only), AV_{DD0}, AV_{DD1}, AV_{DD2}, AV_{REFP0}, AV_{REFP1}

2. V_{SS0}, V_{SS1}, EV_{SS0}, EV_{SS1}, EV_{SS2} (V850E/IG3 only), AV_{SS0}, AV_{SS1}, AV_{SS2}

A host machine is required for controlling the flash programmer.

UARTA0 or CSIB0 is used for the interface between the flash programmer and the V850E/IF3, V850E/IG3 to perform writing, erasing, etc. Supply the operating clock of the V850E/IF3, V850E/IG3 via the oscillator configured on the V850E/IF3, V850E/IG3 board using a resonator and a capacitor.

Table 27-4. Environment and Communication Mode

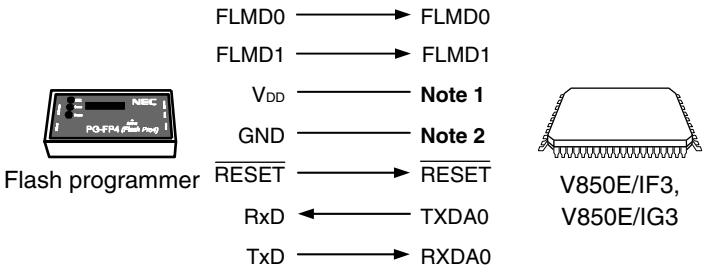
Environment	Communication Mode		
	UARTA0	CSIB0	CSIB0 for Handshake
Flash programmer (PG-FP4, FL-PR4)	√	√	√
MINICUBE2	√	×	√

Remark √: Supported, ×: Not supported

27.4 Communication Method of Flash Memory Programming

(1) UART0 communication method

Transfer rate: 9600 to 153,600 bps (LSB first)



Notes 1. V_{DD0}, V_{DD1}, EV_{DD0}, EV_{DD1}, EV_{DD2} (V850E/IG3 only), AV_{DD0}, AV_{DD1}, AV_{DD2}, AV_{REFP0}, AV_{REFP1}

2. V_{SS0}, V_{SS1}, EV_{SS0}, EV_{SS1}, EV_{SS2} (V850E/IG3 only), AV_{SS0}, AV_{SS1}, AV_{SS2}

Cautions 1. Supply the operating clock of the V850E/IF3, V850E/IG3 via the oscillator configured on the V850E/IF3, V850E/IG3 board using a resonator and a capacitor.

2. For details, refer to the user's manual of each programmer.

Table 27-5. Wiring Correspondence Between Dedicated Flash Programmer and V850E/IF3, V850E/IG3

Pin No.	Dedicated Flash Programmer (PG-FP4)	I/O (PG-FP4 Side)	V850E/IF3, V850E/IG3			
			Pin Name	Pin No.		
				V850E/IF3	V850E/IG3	
				GC	GC	GF
1	GND	–	V_{SS0}	32	37	65
			V_{SS1}	66	85	13
			EV_{SS0}	54	64	92
			EV_{SS1}	79	1	29
			EV_{SS2}	–	42	70
			AV_{SS0}	6	7	35
			AV_{SS1}	11	12	40
			AV_{SS2}	22	27	55
2	$\overline{\text{RESET}}$	Output	$\overline{\text{RESET}}$	35	40	68
3	SI/RxD	Input	TXDA0	39	48	76
4	V_{DD}	–	V_{DD0}	30	35	63
			V_{DD1}	68	87	15
			EV_{DD0}	55	65	93
			EV_{DD1}	80	100	28
			EV_{DD2}	–	41	69
			AV_{DD0}	8	9	37
			AV_{DD1}	9	10	38
			AV_{DD2}	21	26	54
			AV_{REFP0}	7	8	36
			AV_{REFP1}	10	11	39
5	SO/TxD	Output	RXDA0	38	47	75
6	V_{PP}	×	NC	–	–	–
7	SCK	×	NC	–	–	–
8	H/S	×	NC	–	–	–
9	CLK[Note 1]	Output	X1[Note 1]	33	38	66
10	VDE	×	NC	–	–	–
11	V_{DD2}	–	NC	–	–	–
12	FLMD1	Output	**Note 2**	60	76	4
13	RFU-1	×	NC	–	–	–
14	FLMD0	Output	FLMD0	37	46	74
15	Not used	×	NC	–	–	–
16	Not used	×	NC	–	–	–

Notes 1. In the V850E/IF3 and V850E/IG3, external clock input is prohibited. Mount the resonator on board.

2. Connect to FLMD1 or GND via a resistor.

Remark NC: No Connection

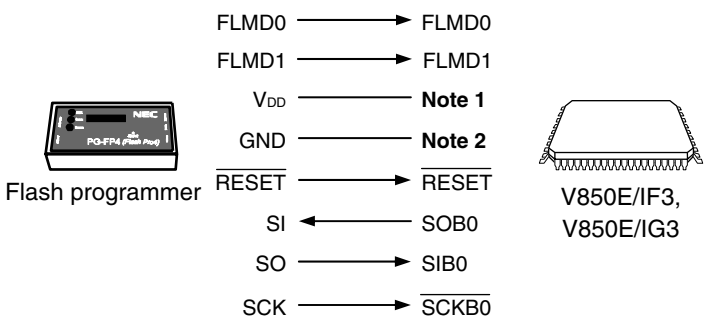
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(2) CSIB0 communication method

Transfer rate: Up to 2.5 Mbps (MSB first)



Notes 1. V_{DD0}, V_{DD1}, EV_{DD0}, EV_{DD1}, EV_{DD2} (V850E/IG3 only), AV_{DD0}, AV_{DD1}, AV_{DD2}, AV_{REFP0}, AV_{REFP1}

2. V_{SS0}, V_{SS1}, EV_{SS0}, EV_{SS1}, EV_{SS2} (V850E/IG3 only), AV_{SS0}, AV_{SS1}, AV_{SS2}

Cautions 1. Supply the operating clock of the V850E/IF3, V850E/IG3 via the oscillator configured on the V850E/IF3, V850E/IG3 board using a resonator and a capacitor.

2. For details, refer to the user's manual of each programmer.

The flash programmer outputs (master) transfer clocks and the V850E/IF3 or V850E/IG3 operates as a slave.

Table 27-6. Wiring Correspondence Between Dedicated Flash Programmer and V850E/IF3, V850E/IG3

Pin No.	Flash Programmer (PG-FP4)	I/O (PG-FP4 Side)	V850E/IF3, V850E/IG3			
			Pin Name	Pin No.		
				V850E/IF3	V850E/IG3	
				GC	GC	GF
1	GND	–	V_{SS0}	32	37	65
			V_{SS1}	66	85	13
			EV_{SS0}	54	64	92
			EV_{SS1}	79	1	29
			EV_{SS2}	–	42	70
			AV_{SS0}	6	7	35
			AV_{SS1}	11	12	40
			AV_{SS2}	22	27	55
2	$\overline{\text{RESET}}$	Output	$\overline{\text{RESET}}$	35	40	68
3	SI/RxD	Input	SOB0	39	48	76
4	V_{DD}	–	V_{DD0}	30	35	63
			V_{DD1}	68	87	15
			EV_{DD0}	55	65	93
			EV_{DD1}	80	100	28
			EV_{DD2}	–	41	69
			AV_{DD0}	8	9	37
			AV_{DD1}	9	10	38
			AV_{DD2}	21	26	54
			AV_{REFP0}	7	8	36
			AV_{REFP1}	10	11	39
5	SO/TxD	Output	SIB0	38	47	75
6	V_{PP}	×	NC	–	–	–
7	SCK	Output	$\overline{\text{SCKB0}}$	40	49	77
8	H/S	×	NC	–	–	–
9	CLKZ[Note 1]	Output	X1[Note 1]	33	38	66
10	VDE	×	NC	–	–	–
11	V_{DD2}	–	NC	–	–	–
12	FLMD1	Output	**Note 2**	60	76	4
13	RFU-1	×	NC	–	–	–
14	FLMD0	Output	FLMD0	37	46	74
15	Not used	×	NC	–	–	–
16	Not used	×	NC	–	–	–

Notes 1. In the V850E/IF3 and V850E/IG3, external clock input is prohibited. Mount the resonator on board.

2. Connect to FLMD1 or GND via a resistor.

Remark NC: No Connection

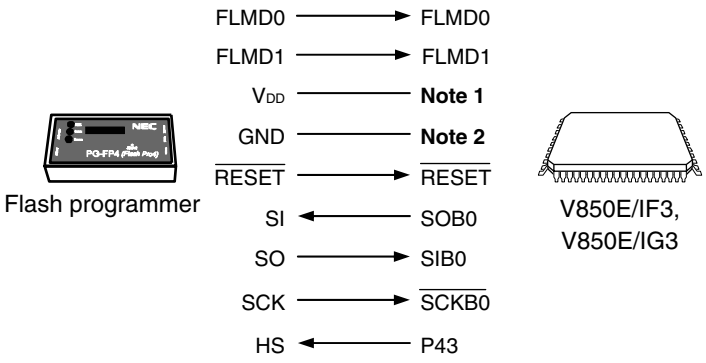
GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

(3) CSIB0 communication method supporting handshake

Transfer rate: Up to 2.5 Mbps (MSB first)



Notes 1. V_{DD0}, V_{DD1}, EV_{DD0}, EV_{DD1}, EV_{DD2} (V850E/IG3 only), AV_{DD0}, AV_{DD1}, AV_{DD2}, AV_{REFP0}, AV_{REFP1}

2. V_{SS0}, V_{SS1}, EV_{SS0}, EV_{SS1}, EV_{SS2} (V850E/IG3 only), AV_{SS0}, AV_{SS1}, AV_{SS2}

Cautions 1. Supply the operating clock of the V850E/IF3, V850E/IG3 via the oscillator configured on the V850E/IF3, V850E/IG3 board using a resonator and a capacitor.

2. For details, refer to the user's manual of each programmer.

The flash programmer outputs the transfer clock, and the V850E/IF3 and V850E/IG3 operate as slaves.

When the PG-FP4 is used as the flash programmer, it sends the following signals to the V850E/IF3 and V850E/IG3. For details, refer to the **PG-FP4 User's Manual (U15260E)**.

Table 27-7. Wiring Correspondence Between Dedicated Flash Programmer and V850E/IF3, V850E/IG3

Pin No.	Flash Programmer (PG-FP4)	I/O (PG-FP4 Side)	V850E/IF3, V850E/IG3			
			Pin Name	Pin No.		
				V850E/IF3	V850E/IG3	
				GC	GC	GC
1	GND	–	V_{SS0}	32	37	65
			V_{SS1}	66	85	13
			EV_{SS0}	54	64	92
			EV_{SS1}	79	1	29
			EV_{SS2}	–	42	70
			AV_{SS0}	6	7	35
			AV_{SS1}	11	12	40
			AV_{SS2}	22	27	55
2	$\overline{\text{RESET}}$	Output	$\overline{\text{RESET}}$	35	40	68
3	SI/RxD	Input	SOB0	39	48	76
4	V_{DD}	–	V_{DD0}	30	35	63
			V_{DD1}	68	87	15
			EV_{DD0}	55	65	93
			EV_{DD1}	80	100	28
			EV_{DD2}	–	41	69
			AV_{DD0}	8	9	37
			AV_{DD1}	9	10	38
			AV_{DD2}	21	26	54
			AV_{REFP0}	7	8	36
			AV_{REFP1}	10	11	39
5	SO/TxD	Output	SIB0	38	47	75
6	V_{PP}	×	NC	–	–	–
7	SCK	Output	$\overline{\text{SCKB0}}$	40	49	77
8	H/S	Input	P43	41	50	78
9	CLK[Note 1]	Output	X1[Note 1]	33	38	66
10	VDE	×	NC	–	–	–
11	V_{DD2}	–	NC	–	–	–
12	FLMD1	Output	**Note 2**	60	76	4
13	RFU-1	×	NC	–	–	–
14	FLMD0	Output	FLMD0	37	46	74
15	Not used	×	NC	–	–	–
16	Not used	×	NC	–	–	–

Notes 1. In the V850E/IF3 and V850E/IG3, external clock input is prohibited. Mount the resonator on board.

2. Connect to FLMD1 or GND via a resistor.

Remark NC: No Connection

GC (V850E/IF3): 80-pin plastic LQFP (14 × 14)

GC (V850E/IG3): 100-pin plastic LQFP (fine pitch) (14 × 14)

GF (V850E/IG3): 100-pin plastic LQFP (14 × 20)

27.5 Pin Processing During Flash Memory Programming

When performing on-board programming, mount a connector on the target system to connect to the flash programmer.

In the flash memory programming mode, all the pins not used for flash memory programming become the same status as that immediately after reset in the normal operation mode. Therefore, because all the ports become high-impedance status, pin processing is required when the external device does not acknowledge the high-impedance status.

27.5.1 Power supply

Supply the same power supplies (V_{DD0}, V_{DD1}, V_{SS0}, V_{SS1}, EV_{DD0}, EV_{DD1}, EV_{DD2} (V850E/IG3 only), EV_{SS0}, EV_{SS1}, EV_{SS2} (V850E/IG3 only), AV_{DD0}, AV_{DD1}, AV_{DD2}, AV_{SS0}, AV_{SS1}, AV_{SS2}, AV_{REFP0}, and AV_{REFP1}) as in the normal operation mode. Connect V_{DD} and GND of the flash programmer to V_{DD0}, V_{DD1}, V_{SS0}, V_{SS1}, EV_{DD0}, EV_{DD1}, EV_{DD2} (V850E/IG3 only), EV_{SS0}, EV_{SS1}, EV_{SS2} (V850E/IG3 only), AV_{DD0}, AV_{DD1}, AV_{DD2}, AV_{SS0}, AV_{SS1}, AV_{SS2}, AV_{REFP0}, and AV_{REFP1}. (V_{DD} of the flash programmer is provided with a power supply monitoring function.)

In the flash memory programming mode (including flash memory self programming), insert capacitors between V_{DD0}, V_{DD1} pins and V_{SS0}, V_{SS1} pins, and between EV_{DD0}, EV_{DD1}, EV_{DD2} (V850E/IG3 only) pins and EV_{SS0}, EV_{SS1}, EV_{SS2} (V850E/IG3 only) pins to stabilize the power supply voltage.

27.5.2 Pins used

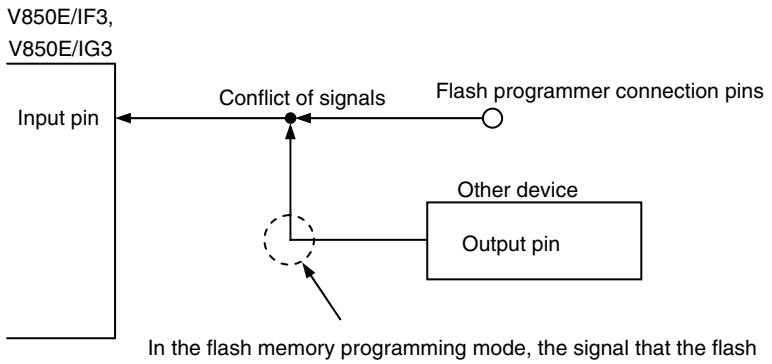
The following shows the pins used by each interface.

Communication Mode	Pins Used
UARTA0	TXD0, RXD0
CSIB0	SOB0, SIB0, $\overline{\text{SCKB0}}$
CSIB0 for handshake	SOB0, SIB0, $\overline{\text{SCKB0}}$, P43

When connecting a flash programmer to an interface pin that is connected to other devices on-board, care should be taken to avoid a conflict of signals or the malfunction of other devices.

(1) Conflict of signals

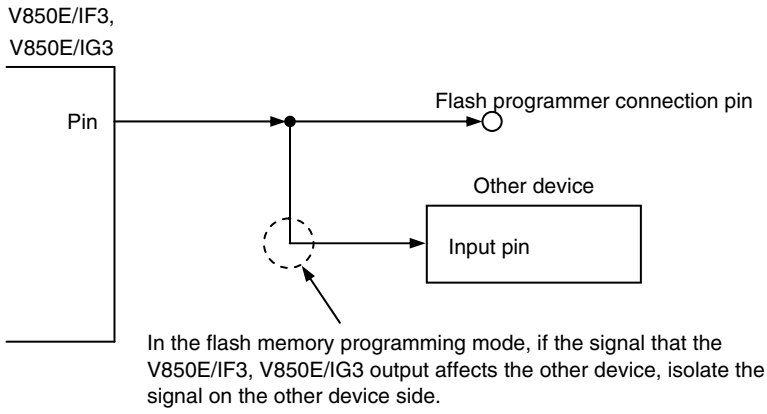
When the flash programmer (output) is connected to an interface pin (input) that is connected to another device (output), a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the other device or set the other device to the output high-impedance status.

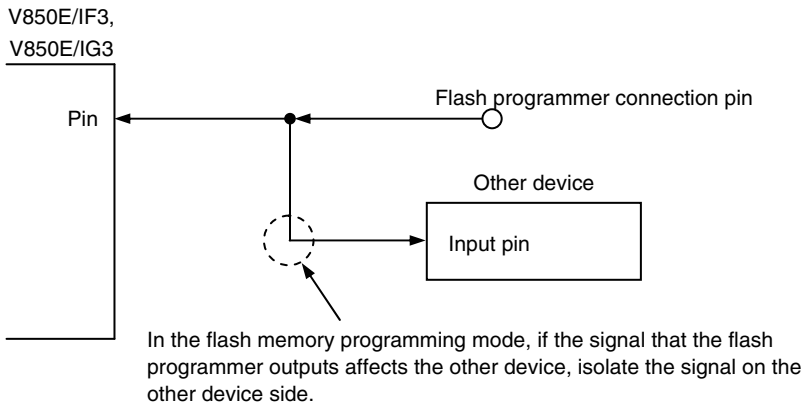


In the flash memory programming mode, the signal that the flash programmer sends out conflicts with signals another device outputs. Therefore, isolate the signals on the other device side.

(2) Malfunction of other device

When the flash programmer (output or input) is connected to an interface pin (input or output) that is connected to another device (input), the signal is output to the other device, causing 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.

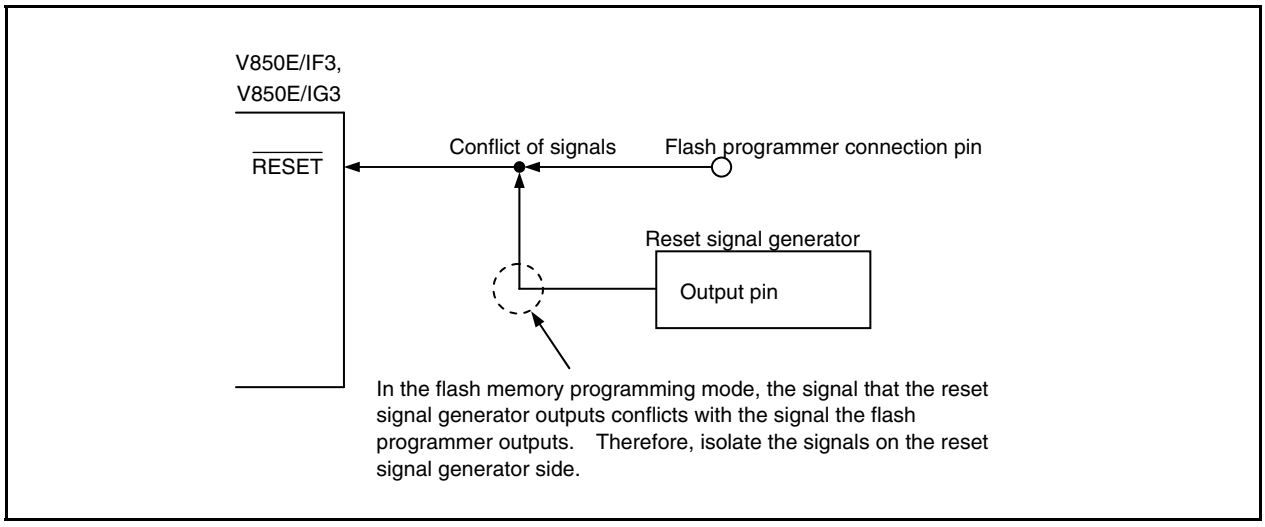




27.5.3 $\overline{\text{RESET}}$ pin

When the reset signal of the flash programmer is connected to the $\overline{\text{RESET}}$ pin that 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 a reset signal is input from the user system in the flash memory programming mode, the programming operation will not be performed correctly. Therefore, do not input signals other than the reset signals from the flash programmer.

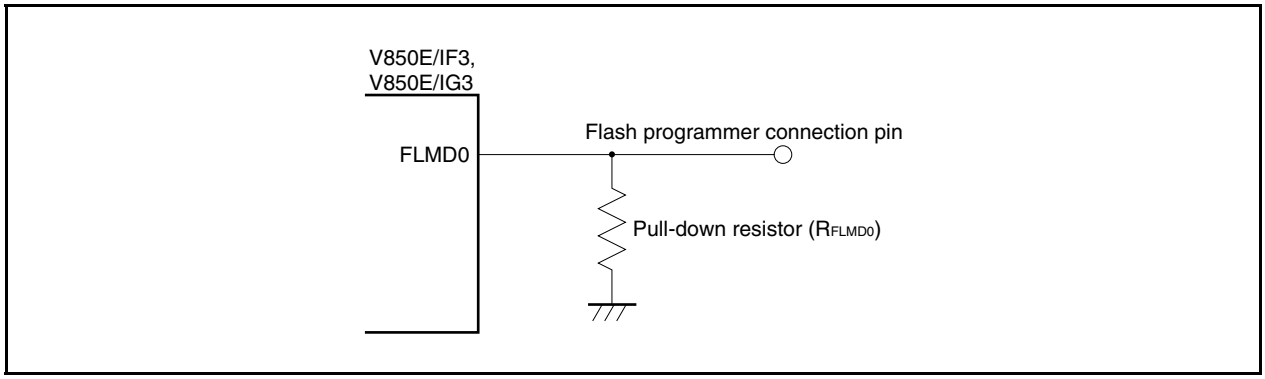


27.5.4 FLMD0, FLMD1 pins

(1) FLMD0 pin

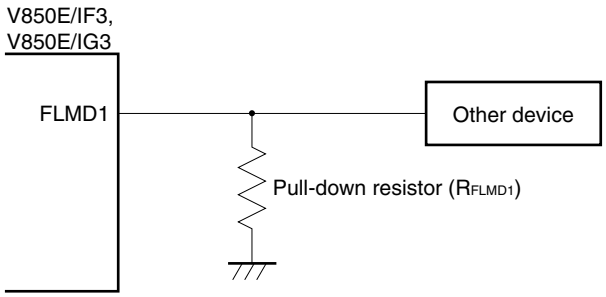
In the normal operation mode, input a voltage of EV_{SS0}, EV_{SS1}, or EV_{SS2} (V850E/IG3) level to the FLMD0 pin. In the flash memory programming mode, supply a write voltage of EV_{DD0}, EV_{DD1}, EV_{DD2} (V850E/IG3 only) level to the FLMD0 pin.

Because the FLMD0 pin serves as a write protection pin in the self programming mode, a voltage of EV_{DD0}, EV_{DD1}, or EV_{DD2} (V850E/IG3 only) level must be supplied to the FLMD0 pin via port control, etc., before writing to the flash memory. For details, see **27.7.7 (1) FLMD0 pin**.



(2) FLMD1 pin

When 0 V is input to the FLMD0 pin, the FLMD1 pin does not function. When EV_{DD0}, EV_{DD1}, or EV_{DD2} (V850E/IG3 only) is supplied to the FLMD0 pin, the flash memory programming mode is entered, so 0 V must be input to the FLMD1 pin. The following shows an example of the connection of the FLMD1 pin.



Caution If the EV_{DD0}, EV_{DD1}, or EV_{DD2} (V850E/IG3 only) signal is input to the FLMD1 pin from another device during on-board programming and immediately after reset, isolate this signal.

Table 27-8. Relationship Between FLMD0 and FLMD1 Pins and Operation Mode When Reset Is Released

FLMD0	FLMD1	Operation Mode
0	Don't care	Normal operation mode
EV_{DD}	0	Flash memory programming mode
EV_{DD}	EV_{DD}	Setting prohibited

Remark EV_{DD}: EV_{DD0}, EV_{DD1}, EV_{DD2} (V850E/IG3 only)

27.5.5 Port pins

When the flash memory programming mode is set, all the port pins except the pin that communicates with the flash programmer change to the high-impedance status. These port pins need not be processed. If problems such as disabling of the high-impedance status should occur to the external devices connected to the ports, connect them to EV_{DD0}, EV_{DD1}, EV_{DD2} (V850E/IG3 only), or EV_{SS0}, EV_{SS1}, EV_{SS2} (V850E/IG3 only) via resistors.

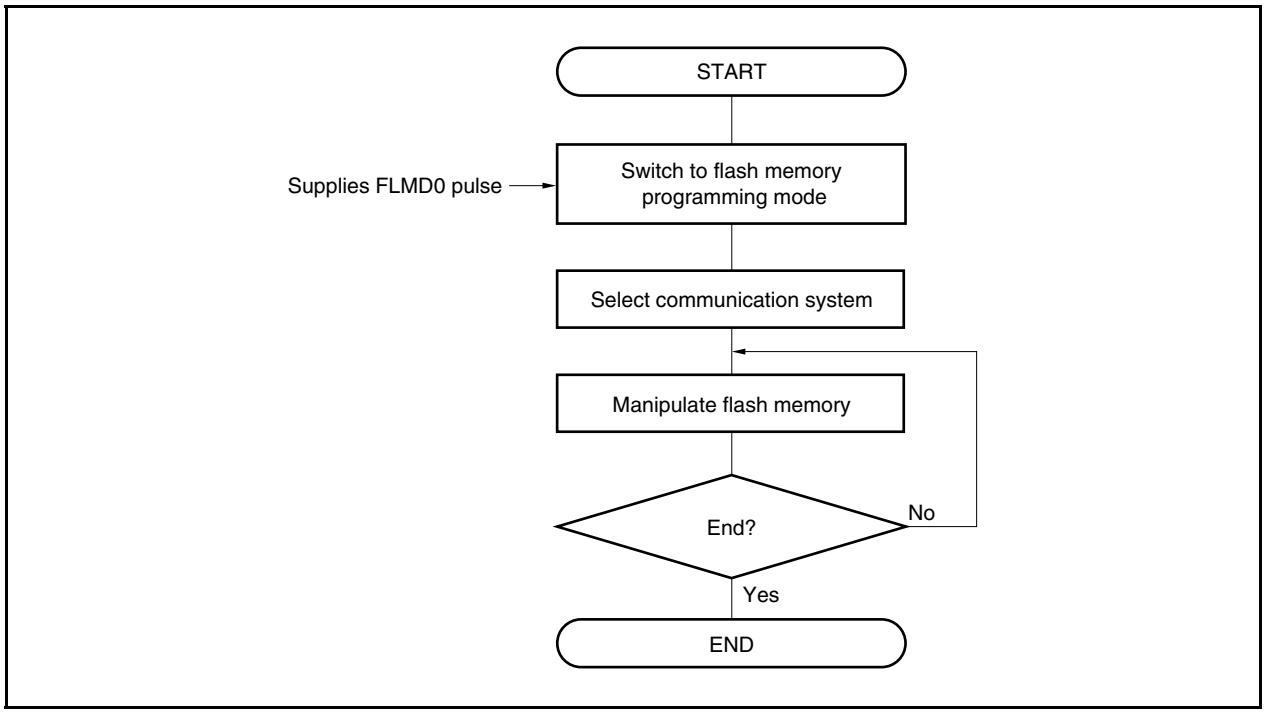
27.5.6 Other signal pins

Connect X1 and X2 in the same status as in the normal operation mode.

27.6 Flash Memory Programming Mode

27.6.1 Flash memory control

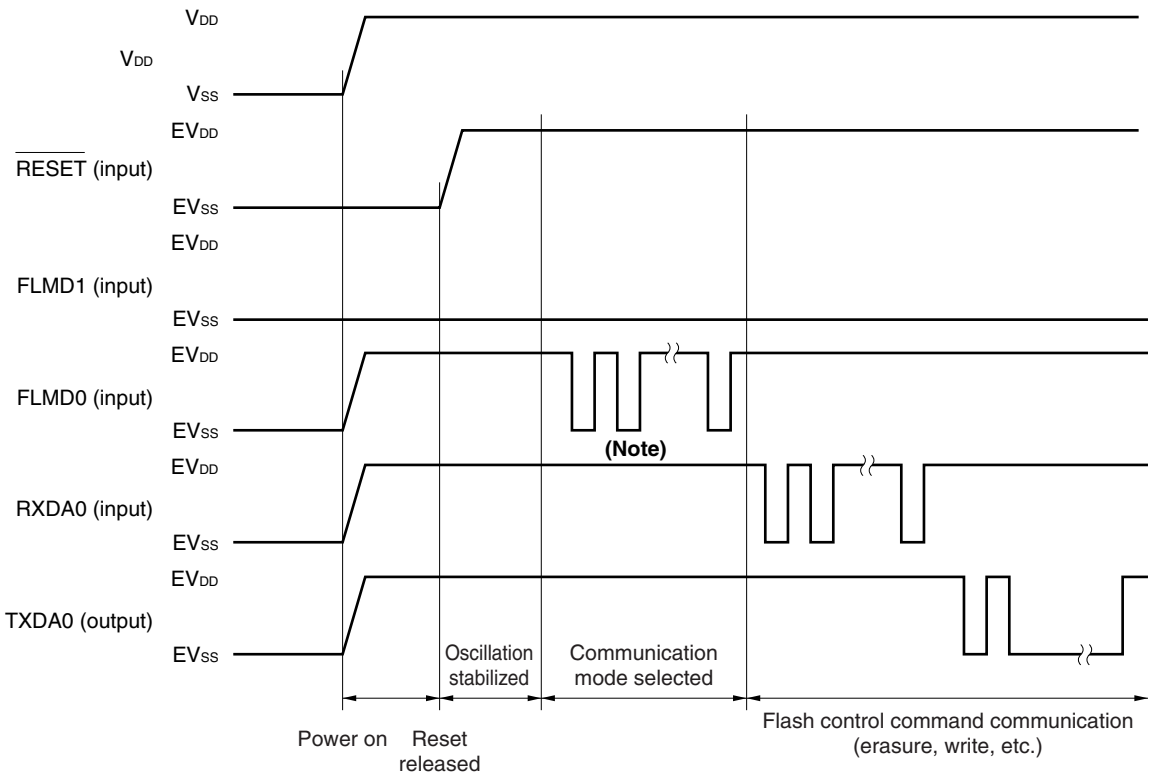
The following shows the procedure for manipulating the flash memory.



27.6.2 Selection of communication mode

In the V850E/IF3 and V850E/IG3, the communication mode is selected by inputting pulses (11 pulses max.) to the FLMD0 pin after switching to the flash memory programming mode. The FLMD0 pulse is generated by the flash programmer.

The following shows the relationship between the number of pulses and the communication mode.



Note The number of clocks is as follows depending on the communication mode.

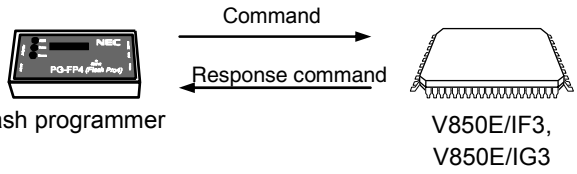
FLMD0 pulse	Communication mode	Remarks
0	UARTA0	Communication rate: 9,600 bps (after reset), LSB first
8	CSIB0	V850E/IF3 and V850E/IG3 perform slave operation, MSB first
11	CSIB0 for handshake	V850E/IF3 and V850E/IG3 perform slave operation, MSB first
Other	RFU	Setting prohibited

Caution When UARTA0 is selected, the receive clock is calculated based on the reset command sent from the flash programmer after receiving the FLMD0 pulse.

Remark V_{DD}: V_{DD0}, V_{DD1}
EV_{DD}: EV_{DD0}, EV_{DD1}, EV_{DD2} (V850E/IG3 only)
EV_{SS}: EV_{SS0}, EV_{SS1}, EV_{SS2} (V850E/IG3 only)

27.6.3 Communication commands

The V850E/IF3 and V850E/IG3 communicate with the flash programmer by means of commands. The commands sent from the flash programmer to the V850E/IF3 and V850E/IG3 are called “commands”. The response signals sent from the V850E/IF3 and V850E/IG3 to the flash programmer are called “response commands”.



The following shows the commands for flash memory control in the V850E/IF3 and V850E/IG3. All of these commands are issued from the dedicated flash programmer, and the V850E/IF3 and V850E/IG3 perform the processing corresponding to the commands.

Classification	Command Name	Support			Function
		UARTA0	CSIB0	**Note**	
Verify	Block verify command	√	√	√	Compares the contents of the specified block and the input data
Erase	Chip erase command	√	√	√	Erases the contents of the entire flash memory
	Block erase command	√	√	√	Erases the specified block contents.
Blank check	Block blank check command	√	√	√	Checks the erase state of the specified block.
Data write	Write command	√	√	√	Writes data to the specified block.
Data read	Read command	√	√	√	Reads out data of the specified block.
System setting and control	Status command	√	×	√	Obtains the status of operations.
	Oscillation frequency setting command	√	√	√	Sets the oscillation frequency.
	Baud rate setting command	√	×	×	Changes the baud rate when UART0 is selected.
	Silicon signature command	√	√	√	Reads out the silicon signature information.
	Version acquisition command	√	√	√	Reads out the device version and firmware version.
	Security setting command	√	√	√	Sets the security information and the boot block size.
	Checksum command	√	√	√	Sends the checksum value of the specified block data.
	Reset command	√	√	√	Used for communication synchronization detecting.

Note CSIB0 for handshake

The V850E/IF3 and V850E/IG3 send back response commands for the commands issued from the flash programmer. The response commands sent from the V850E/IF3 and V850E/IG3 are listed below.

Response Command Name	Function
ACK (Acknowledge)	Acknowledges command/data, etc.
NAK (Not acknowledge)	Acknowledges illegal frame, etc.
Command number error	Acknowledges illegal command/data, etc.
Parameter error	Acknowledges illegal parameter, etc.
Checksum error	Acknowledges checksum of frame
Protect error	Acknowledges when protection is in effect
During processing (BUSY)	Acknowledges during processing
Other than above	Error

27.7 Flash Memory Self Programming

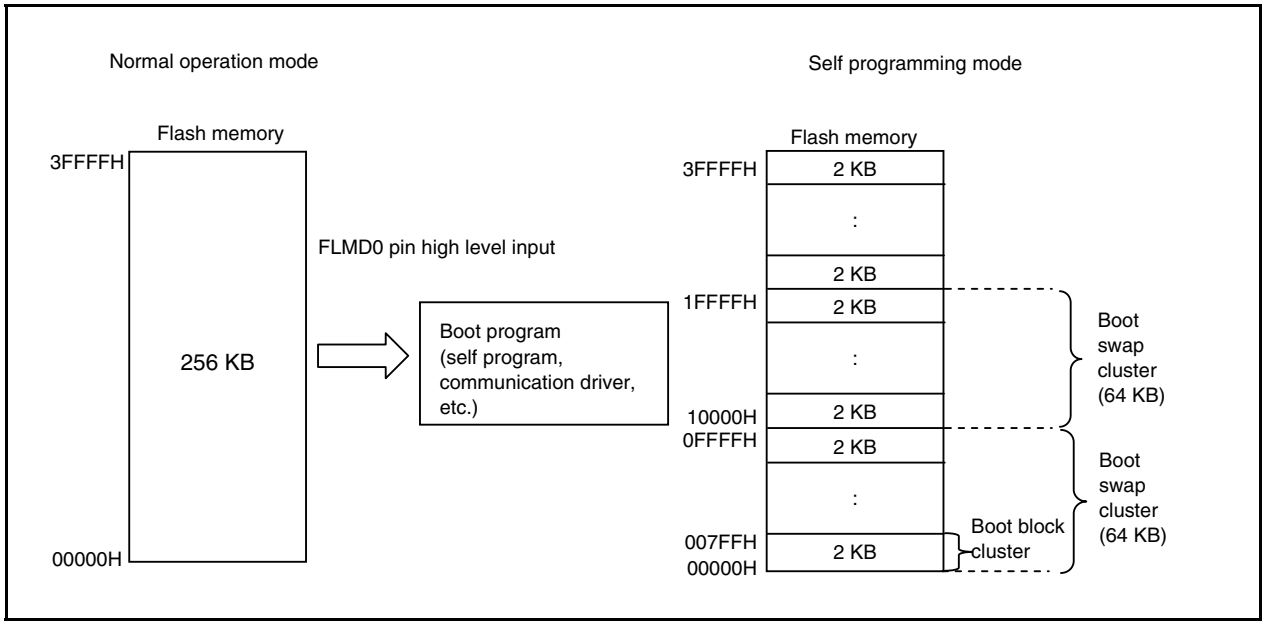
The V850E/IF3 and V850E/IG3 support a self programming function to rewrite the flash memory with a user program. By using flash memory self programming, the flash memory can be rewritten by user applications. This flash memory self programming can be used for program upgrades in the field.

27.7.1 Outline of flash memory self programming

Flash memory self programming is used to erase or write the flash memory by calling the flash function from a program stored in an area other than the flash memory area to be erased or written. To store the program that implements self programming in the area to be erased or written, copy the program to the internal RAM area, execute the program at the copy destination, and call the flash function.

To call the flash function, change the mode from the normal operation mode to the self programming mode by using the flash programming mode control register.

Figure 27-1. Outline of Self Programming



(1) Boot swap cluster

The contents of the boot swap cluster of the lower address side (00000H to 0FFFFH) and the boot swap cluster of the higher address side (10000H to 1FFFFH) can be interchanged while flash memory programming is performed.

(2) Boot block cluster

By specifying the boot block cluster from 00000H in 2 KB units, the contents of the boot block cluster can be protected from rewriting.

27.7.2 Self programming library

The flash memory version provides the self programming functions listed in Table 27-9. Write/erase of the flash memory is performed by calling the library function from the application program.

The characteristics of the self programming library are as below.

- The contents of the flash memory can be changed by the application program without using a dedicated writing tool.
- The external interface can be defined by users because it has an independent structure.
- Can be used in C language using the basic library functions provided by NEC Electronics.

Figure 27-2. Overview

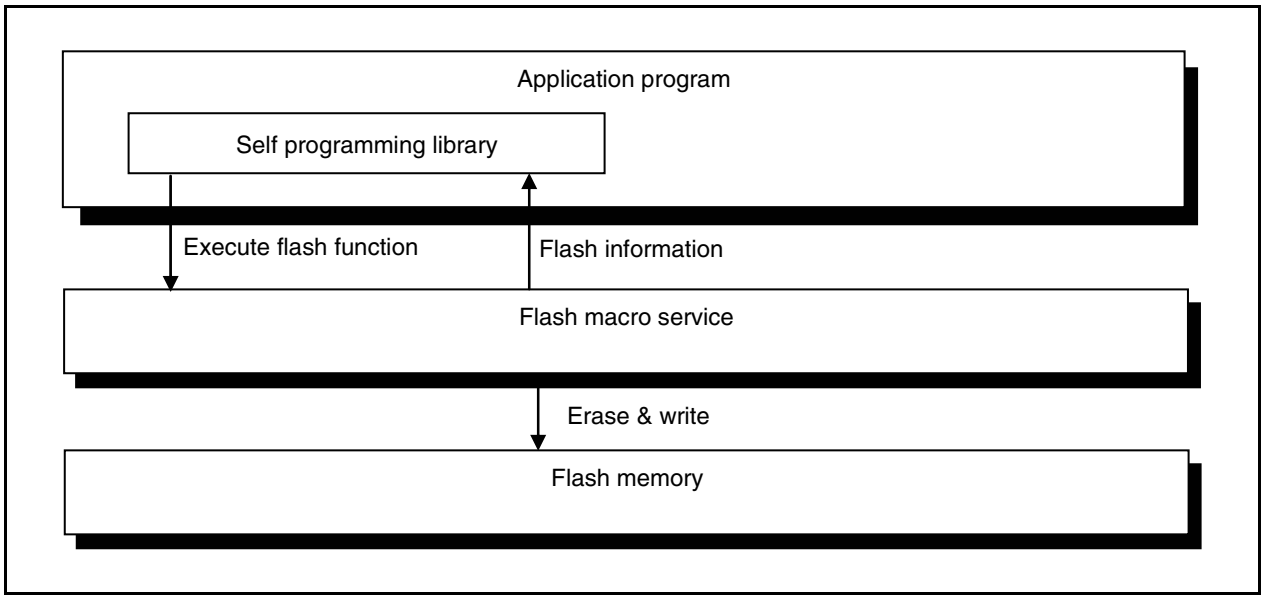


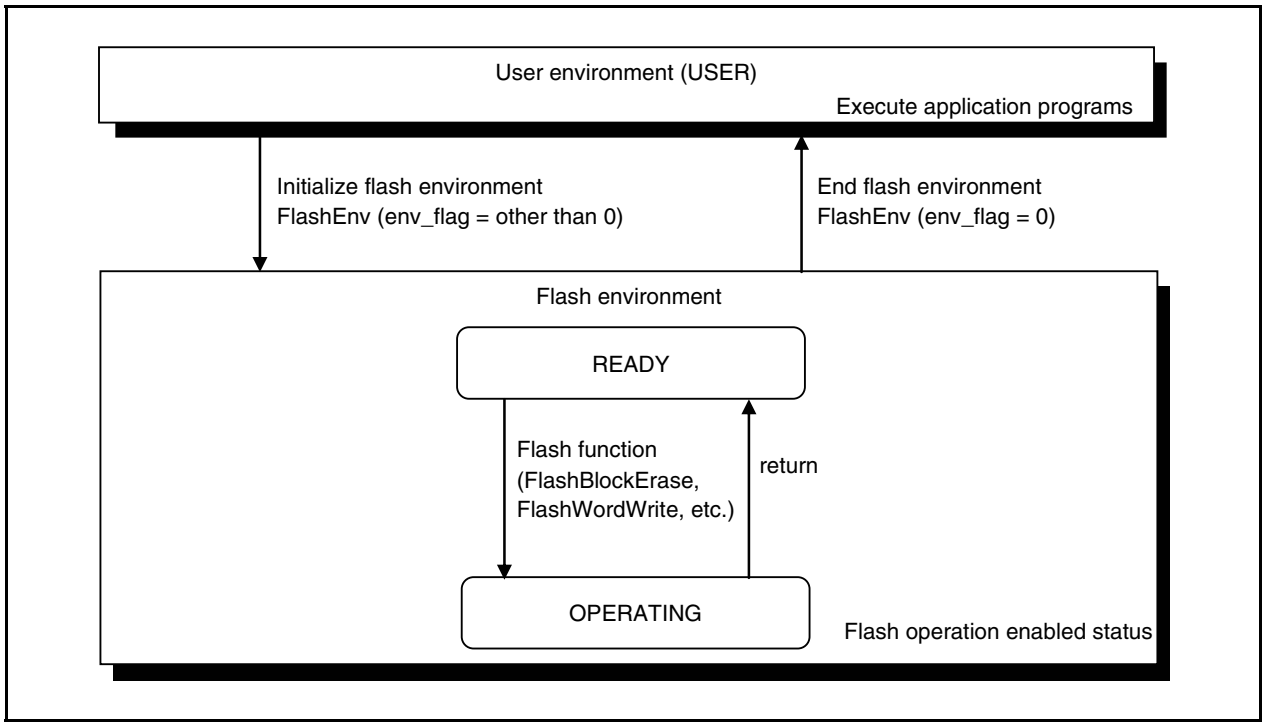
Table 27-9. Flash Function List

Type	Function Name	Abbr.	Function
Initialize	FlashEnv	FLE	Initializes flash control.
Erase	FlashBlockErase	FLBE	Erases the specified block.
Write	FlashWordWrite	FLWW	Successively writes the specified memory contents from the specified flash memory address, for the number of words specified in 4-byte units.
Check	FlashBlockBlankCheck	FLBBC	Checks the erase status of the specified block.
	FlashBlockIVerify	FLBIV	Perform internal verify for the specified block.
	FlashFLMDCheck	FLFC	Inputs FLMD0 pin and checks FLMD0 setting register value.
Obtain information	FlashGetInfo	FLGI	Reads out information about the flash memory.
Setting	FlashSetInfo	FLSI	Sets the flash information.
	FlashBootSwap	FLBS	Interchanges the contents of the boot swap cluster.
	FlashWordRead	FLWR	Reads out the data from the specified address.

27.7.3 Flash environment

To execute flash memory self programming, the environment must be changed from the user environment to a flash environment. The flash environment is an environment in which the flash memory is rewritten or erased by using a flash function (self programming library).

Figure 27-3. Flash Environment



(1) User environment

This is the status in which user's application program operation without flash memory manipulation is performed. To set the flash environment, make the FLMD0 pin high, and execute flash environment starting processing FlashEnv (env_flag = other than 0) on the internal RAM.

(2) Flash environment

In this environment, the flash memory can be erased or written by a flash function (self programming library). To change from this environment to the user environment, execute flash environment end processing FlashEnv (env_flag = 0) and then input a low level to the FLMD0 pin in the user environment.

27.7.4 Restrictions

Restrictions for the flash memory self programming mode are explained below.

(1) Restrictions concerning status

Table 27-10. Restrictions Concerning Status

Types	Status Name	Restrictions
User environment	USER	Operate in the normal operation mode.
Flash environment	READY	• Do not write in the RAM area that is being used by the flash self programming library. • Stabilize the high level voltage to FLMD0 pin.
	OPERATION	• Do not input the reset signal. • Do not write in the RAM area that is being used by the flash self programming library. • Stabilize the high-level voltage to FLMD0 pin.

(2) Restrictions concerning flash environment

Table 27-11. Restrictions Concerning Flash Environment

Item	Restrictions
Maskable interrupt	It is necessary to set the JMP instruction for jumping to the interrupt servicing at the 4th byte from the beginning of internal RAM.
Clock input to X1 pin	Do not stop the clock input to the X1 pin in the flash environment.
Power save mode	Do not use the power save mode.
Flash memory area	Do not execute a program in flash memory in the flash environment. Execute the program on the internal RAM. Do not read the flash memory area. To read the flash memory area, use flash function Flash Word Read.

(3) Other restrictions

(a) Program execution in and DMA transfer with internal RAM

The CPU may not operate correctly if all the following conditions are satisfied. In this status, only the reset signal can be accepted.

[Condition]

- Bit manipulation instruction (SET1, CLR1, or NOT1) in internal RAM
- Data access instruction to misaligned address of internal RAM

DMA transfer to the internal RAM is executed while any of the above instructions is executed.
Therefore, take either of the following evasive actions.

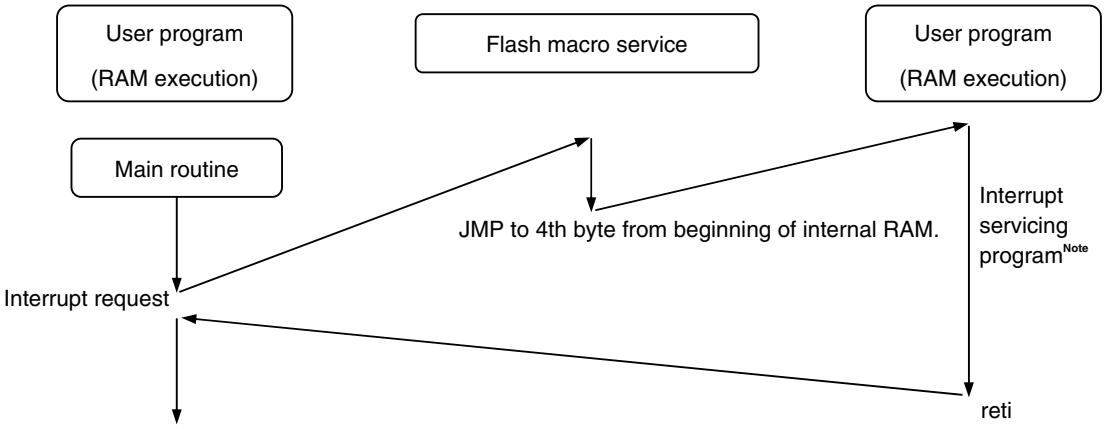
[Action]

- To execute DMA transfer with the internal RAM, do not execute a bit manipulation instruction (SET1, CLR1, or NOT1) in the internal RAM or a data access instruction to a misaligned address.
- When executing a bit manipulation instruction (SET1, CLR1, or NOT1) in the internal RAM or a data access instruction to a misaligned address, do not execute DMA transfer with the internal RAM.

27.7.5 Interrupt servicing in flash environment

In the flash environment, maskable interrupt servicing can be performed. However, such servicing differs from that of normal interrupt. An overview of interrupt servicing in the flash environment is provided below.

Figure 27-4. Overview of Interrupt Servicing



Note Read the ECR register of the CPU by an interrupt servicing program and check the interrupt source.

When an interrupt is generated in the flash environment, the flash macro service unconditionally jumps to the 4th byte from the beginning of the internal RAM. Judge whether the interrupt corresponds to the interrupt request bit of the interrupt control register, and if it does, execute the interrupt servicing, ending it with the reti instruction.

(1) Cautions to use interrupt servicing in flash environment

- When the interrupt-enable status is not set in the user program, it is necessary to set interrupt servicing at the 4th byte from the head address of the internal RAM. If this is not done, program runaway may occur due to interrupts.
- Do not access to the flash memory area for interrupt servicing.
- Do not execute the flash functions for interrupt servicing.
- To use general-purpose register for interrupt servicing, perform save and restore as part of the interrupt servicing.
- Do not use the handler address (00000000H) of reset and the handler address (00000050H) of the software exception (Trap10) because these addresses are used by the flash macro service.
- When an interrupt is generated during FlashEnv execution, the interrupt is suspended for 50 μs maximum.
- Interrupt handler switching is performed as part of the FlashEnv function (beginning of flash memory $\Leftrightarrow$ 4th byte from beginning of internal RAM). Note that on switching/returning to the self programming mode, there is the possibility of execution jumping to both of the interrupt handlers.
- An interrupt hold time of approx. 100 clocks (f_{CPU}) occurs every time a flash function is executed.
- Do not provide library code for interrupt handlers.

(2) Interrupt response time in flash environment

Unlike in the case of a normal interrupt, interrupts in the flash environment are serviced via flash macro service, resulting in a longer interrupt response time.

Flash macro service processing time (the maximum value) = 3 [clocks (f_{CPU})]

Remark f_{CPU}: CPU clock

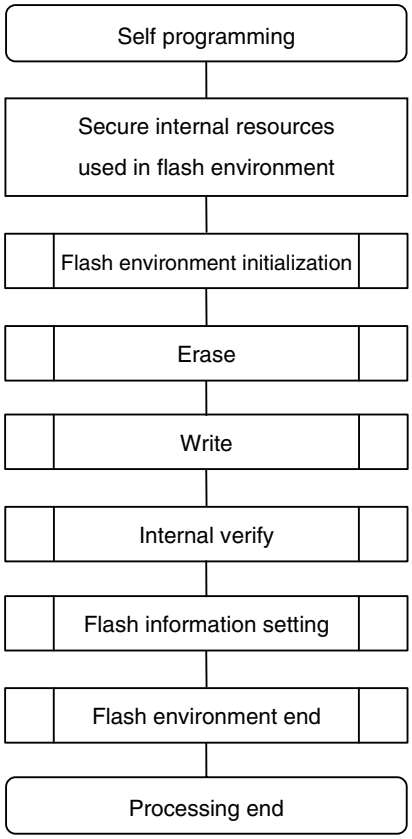
27.7.6 Flash memory self programming flow

(1) The overall flash memory self programming flow (recommended) is outlined below.

Caution Use the flash memory self programming mode according to 27.7.4 Restrictions.

(a) When performing rewrite in one time

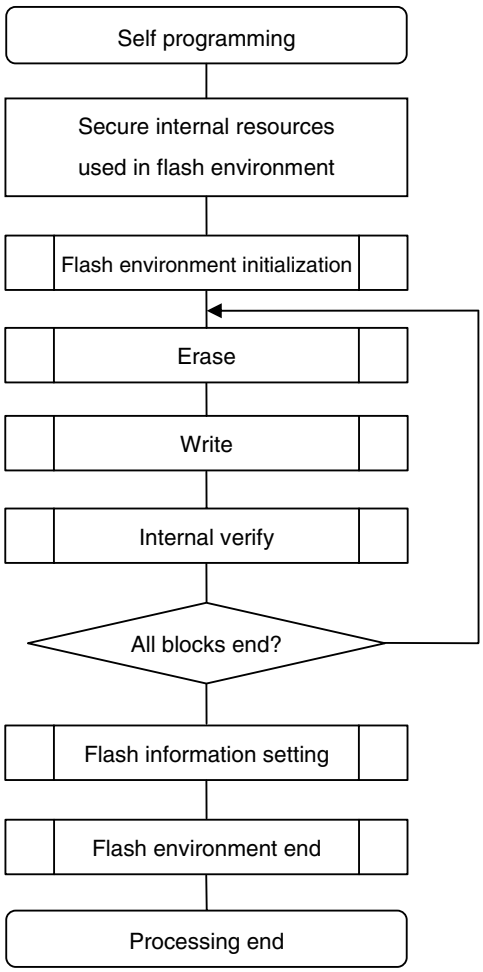
Figure 27-5. Overall Flash Memory Self Programming Flow (1/2)



Remark When not setting security, it is not necessary to execute the flash information setting.

(b) When performing rewrite in block units

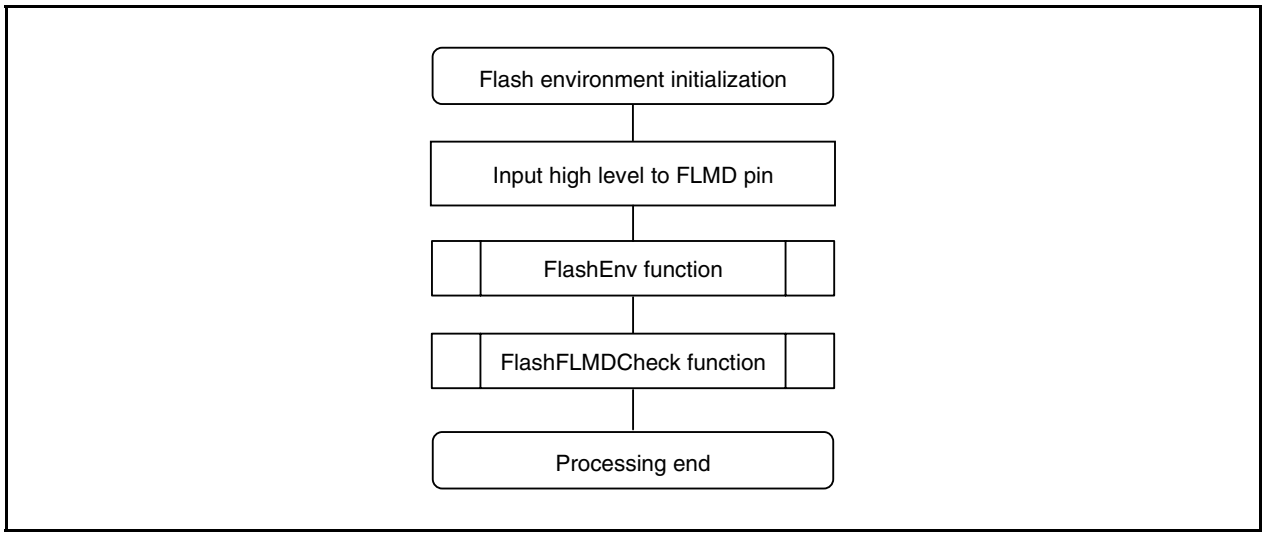
Figure 27-5. Overall Flash Memory Self Programming Flow (2/2)



Remark When not setting security, it is not necessary to execute the flash information setting.

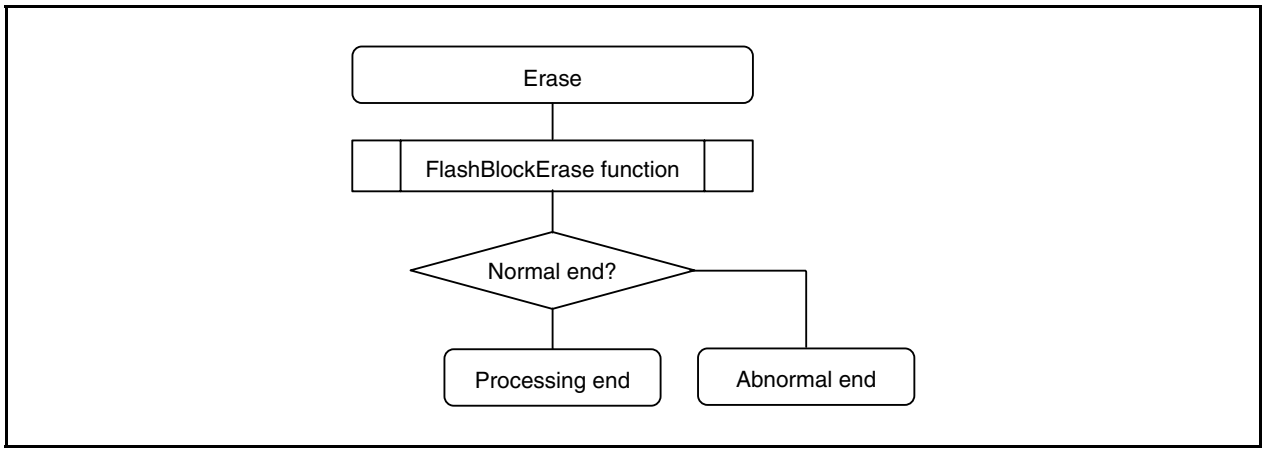
(2) Flash environment initialization flow

Figure 27-6. Flash Environment Initialization Processing Flow



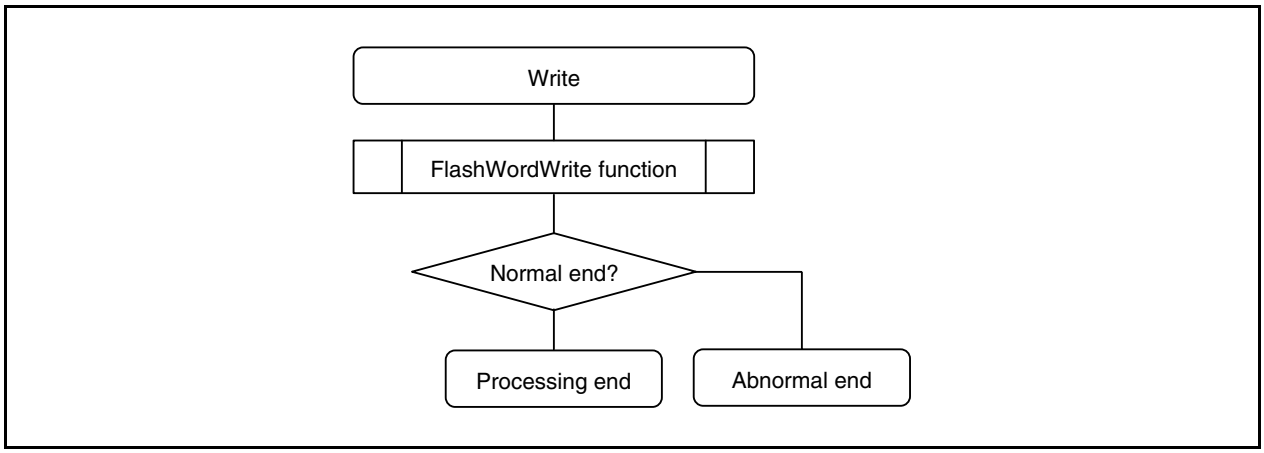
(3) Erasing flow

Figure 27-7. Erase Processing Flow



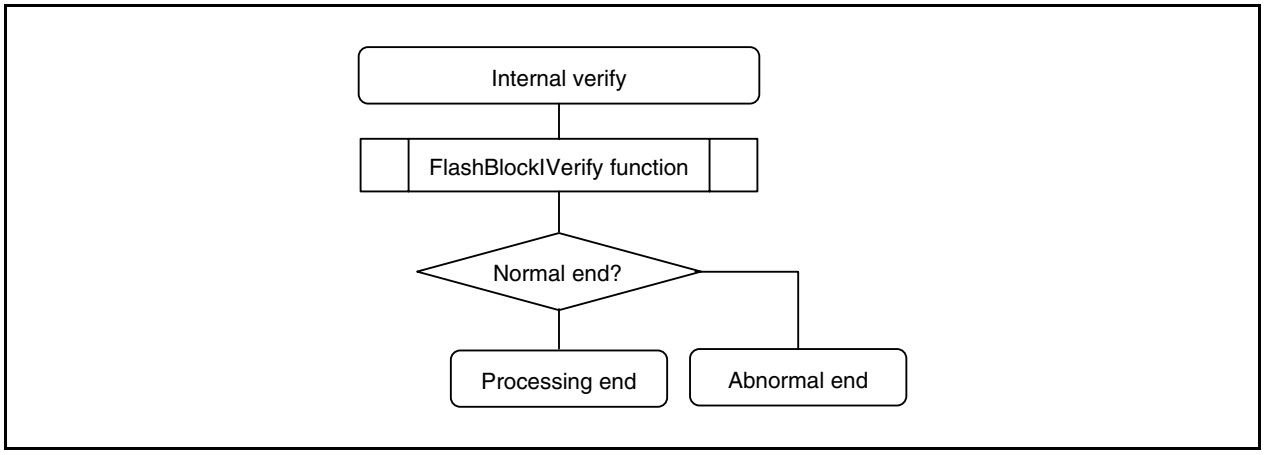
(4) Writing flow

Figure 27-8. Write Processing Flow



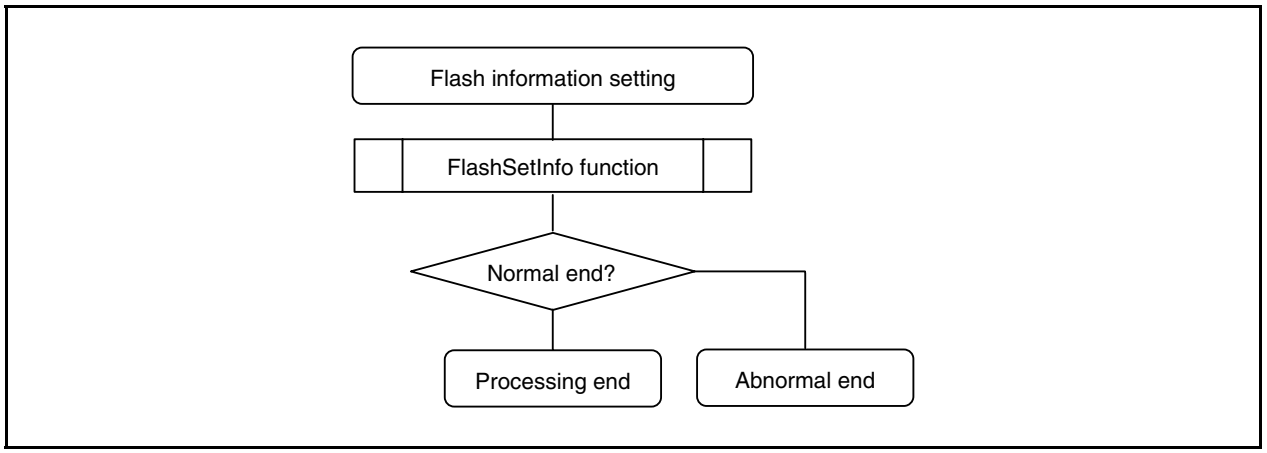
(5) Internal verify processing flow

Figure 27-9. Internal Verify Processing Flow



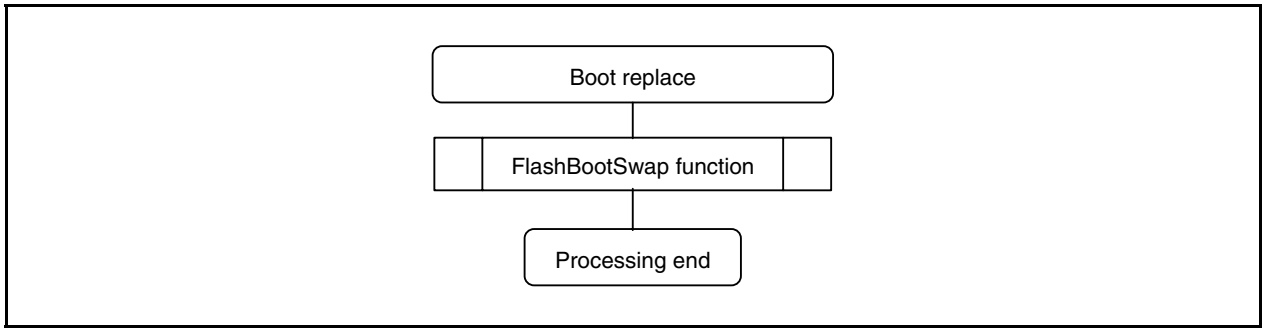
(6) Flash information setting flow

Figure 27-10. Flash Information Setting Flow



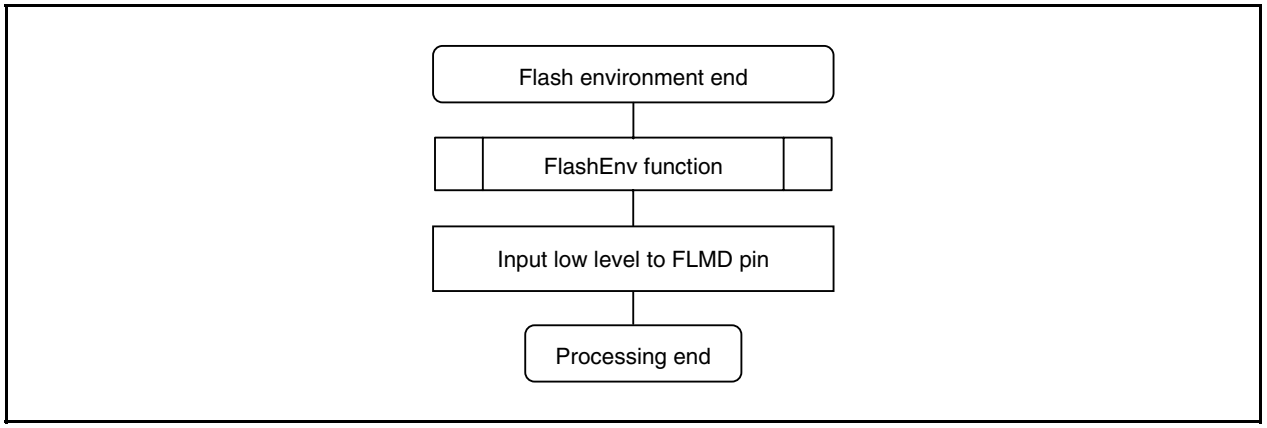
(7) Boot swap processing flow

Figure 27-11. Boot Swap Processing Flow



(8) Flash environment end processing flow

Figure 27-12. Flash Environment End Processing Flow



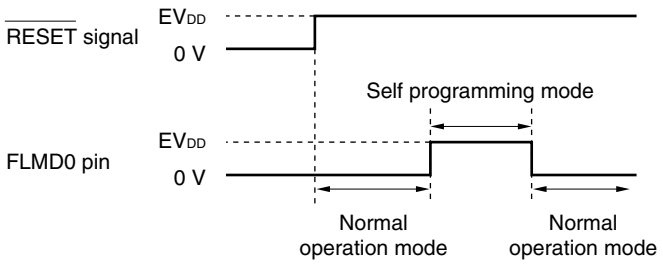
27.7.7 Pin processing

(1) FLMD0 pin

The FLMD0 pin is used to set the operation mode when reset is released and to protect the flash memory from being written during self rewriting. It is therefore necessary to keep the voltage applied to the FLMD0 pin at 0 V when reset is released and a normal operation is executed. It is also necessary to apply a voltage of EV_{DD0}, EV_{DD1}, EV_{DD2} (V850E/IG3 only) level to the FLMD0 pin during the self programming mode period via port control before the memory is rewritten.

When self programming has been completed, the voltage on the FLMD0 pin must be returned to 0 V.

Figure 27-13. Mode Change Timing



Remark EV_{DD}: EV_{DD0}, EV_{DD1}, EV_{DD2} (V850E/IG3 only)

Caution Make sure that the FLMD0 pin is at 0 V when reset is released.

CHAPTER 28 ELECTRICAL SPECIFICATIONS (TARGET)

28.1 V850E/IF3

28.1.1 Absolute maximum ratings

(T_A = 25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}	$V_{DDa} = EV_{DDb} = AV_{DDk}$		−0.5 to +6.5	V
	V_{SS}	$V_{SSa} = EV_{SSb} = AV_{SSk}$		−0.5 to +0.5	V
	EV_{DD}	$V_{DDa} = EV_{DDb} = AV_{DDk}$		−0.5 to +6.5	V
	EV_{SS}	$V_{SSa} = EV_{SSb} = AV_{SSk}$		−0.5 to +0.5	V
	AV_{DD}	$V_{DDa} = EV_{DDb} = AV_{DDk}$		−0.5 to +6.5	V
	AV_{SS}	$V_{SSa} = EV_{SSb} = AV_{SSk}$		−0.5 to +0.5	V
Input voltage	V_{I1}	**Note 1**		−0.5 to EV_{DD} + 0.5 Note 2	V
	V_{I2}	X1, X2		−0.5 to V_{RO} + 0.35	V
Output current, low	I_{OL}	All pins	Per pin	4	mA
			Total of all pins	42	mA
Output current, high	I_{OH}	All pins	Per pin	−4	mA
			Total of all pins	−42	mA
Analog input voltage	V_{IAN}	P70/ANI20 to P73/ANI23, ANI00 to ANI05, ANI10 to ANI17		−0.5 to AV_{DD} + 0.5 Note 2	V
Analog reference input voltage	V_{IREF}	AV_{REFP0}, AV_{REFP1}		−0.5 to AV_{DD} + 0.5 Note 2	V
Comparator reference input voltage	V_{CREF}	CREF0L, CREF1L, CREF0F, CREF1F		−0.5 to AV_{DD} + 0.5 Note 2	V
Operating ambient temperature	T_A	In normal operating mode		−40 to +85	°C
		In flash memory programming mode		−40 to +85	°C
Operating ambient temperature	T_{stg}			−40 to +125	°C

Notes 1. P00, P01, P10 to P17, P20 to P27, P30 to P37, P40 to P47, PDL0 to PDL9, $\overline{\text{RESET}}$, FLMD0

2. Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

Cautions 1. Do not directly connect the output pins (or I/O pins in the output state) of IC products to other output pins (including I/O pins in the output state), power supply pins such as V_{DD} and EV_{DD}, or GND pin. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

The ratings and conditions indicated for DC characteristics and AC characteristics represent the quality assurance range during normal operation.

Remark a = 0, 1

b = 0, 1

k = 0 to 2

28.1.2 Capacitance

(T_A = 25°C, V_{DD0} = V_{SS0} = V_{DD1} = V_{SS1} = EV_{DD0} = EV_{SS0} = EV_{DD1} = EV_{SS1} = AV_{DD0} = AV_{SS0} = AV_{DD1} = AV_{SS1} = AV_{DD2} = AV_{SS2} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	fc = 1 MHz Unmeasured pins returned to 0 V	**Note 1**			15	pF
I/O capacitance	C_{IO}		**Note 2**			15	pF

Notes 1. ANI00 to ANI05, ANI10 to ANI17, $\overline{\text{RESET}}$

2. P00, P01, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P70 to P73, PDL0 to PDL9

Cautions 1. Excludes the FLMD0, X1, and X2 pins.

2. In addition to input capacitance, sampling capacitance is added to the ANI00 to ANI05, ANI10 to ANI17, and ANI20 to ANI27 pins when sampling.

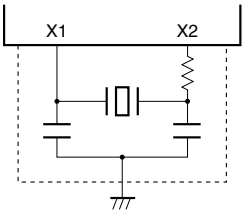
28.1.3 Operating conditions

(T_A = –40 to +85°C, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
System clock frequency	f_{XX}	PLL mode	32		64	MHz
		Clock through mode	4		8	MHz
CPU clock frequency	f_{CPU}	PLL mode	4		64	MHz
		Clock through mode	0.5		8	MHz
V_{DD}, EV_{DD} voltage	V_{DD}, EV_{DD}	V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2}	3.5		5.5	V
AV_{DD} voltage	AV_{DD}	When A/D converters 0 to 2 are operating	4.0		5.5	V
		When A/D converters 0 to 2 are not operating	3.5		5.5	V

28.1.4 Clock oscillator characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic /crystal resonator		Oscillation frequency (f_X)		4		8	MHz
		Oscillation stabilization time	After reset release		$2^{14}/f_X$		ms
			After STOP mode release		**Note**		ms

Note The value varies depending on the setting of the oscillation stabilization time select register (OSTS).

Cautions 1. Connect the oscillator as close to the X1 and X2 pins as possible.

2. Do not cross the wiring with the other signal lines in the area enclosed by the broken lines in the above figure.

3. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

4. Inputting an external clock to the V850E/IF3 is prohibited.

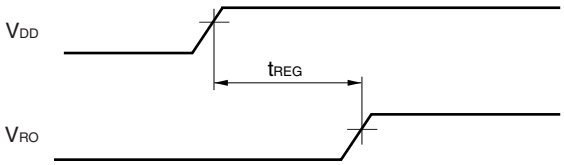
28.1.5 Regulator characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = 3.5 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage	V_{DD}		3.5		5.5	V
Output voltage	V_{RO}			1.5		V
Output voltage stabilization time	t_{REG}	Stabilizing capacitor: C = 4.7 μF[Note 1]			1[Note 2]	ms

Notes 1. Connect a stabilizing capacitor between the REGC0 pin and V_{SS0} pin, and between the REGC1 pin and V_{SS1} pin.

2. Internal reset signal is output until the power-on-clear circuit (POC) output voltage stabilizes during t_{REG} period.



28.1.6 DC characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V) (1/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	**Note 1**		0.7EV_{DD}		EV_{DD}	V
	V_{IH2}	**Note 2**		0.8EV_{DD}		EV_{DD}	V
	V_{IH3}	**Note 3**		0.7AV_{DD}		AV_{DD}	V
Input voltage, low	V_{IL1}	**Note 1**		EV_{SS}		0.3EV_{DD}	V
	V_{IL2}	**Note 2**		EV_{SS}		0.2EV_{DD}	V
	V_{IL3}	**Note 3**		AV_{SS}		0.3AV_{DD}	V
Input leakage current, high	I_{LIH1}	V_I = **Note 4**	Other than X1			5	μA
	I_{LIH2}		X1			20	μA
Input leakage current, low	I_{LIL1}	V_I = 0 V	Other than X1			–5	μA
	I_{LIL2}		X1			–20	μA
Output leakage current, high	I_{LOH}	V_O = **Note 4**				5	μA
Output leakage current, low	I_{LOL}	V_O = 0 V				–5	μA
Output voltage, high	V_{OH1}	**Note 5**	I_{OH} = –1.0 mA	EV_{DD} – 1.0			V
Output voltage, low	V_{OL1}	**Note 5**	I_{OL} = 1.0 mA			0.4	V
Pull to up resistor	R_{L1}			10	30	100	kΩ

Notes 1. P33, P36, P41, and PDL0 to PDL9 pins

2. P00, P01, P10 to P17, P20 to P27, P30 to P32, P34, P35, P37, P40, P42 to P47, $\overline{\text{RESET}}$, and FLMD0 pins

3. P70 to P73 pins

4. AV_{DD0} = AV_{DD1} = AV_{DD2} = EV_{DD0} = EV_{DD1}

5. P00, P01, P10 to P17, P20 to P27, P30 to P37, P40 to P47, and PDL0 to PDL9 pins

Remark The characteristics of alternate-function pins are the same as those of port pins.

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V) (2/2)

Parameter	Symbol	Conditions		MIN.	TYP.[Note 1]	MAX.	Unit
V_{DD} supply current[Note 2]	I_{DD1}	f_{XX} = 64 MHz	Normal operation		64	93	mA
	I_{DD2}		HALT mode		42	60	mA
	I_{DD3}		IDLE mode		5	10	mA
	I_{DD4}	STOP mode			40	800	μA

Notes 1. The TYP. value is a reference value when V_{DD0} = V_{DD1} = 5.0 V and T_A = 25°C.

2. The current consumed by the EV_{DD} system (output buffer and pull-up resistor) and the operating currents of A/D converters 0 to 2, the operational amplifier, and the comparator are not included.

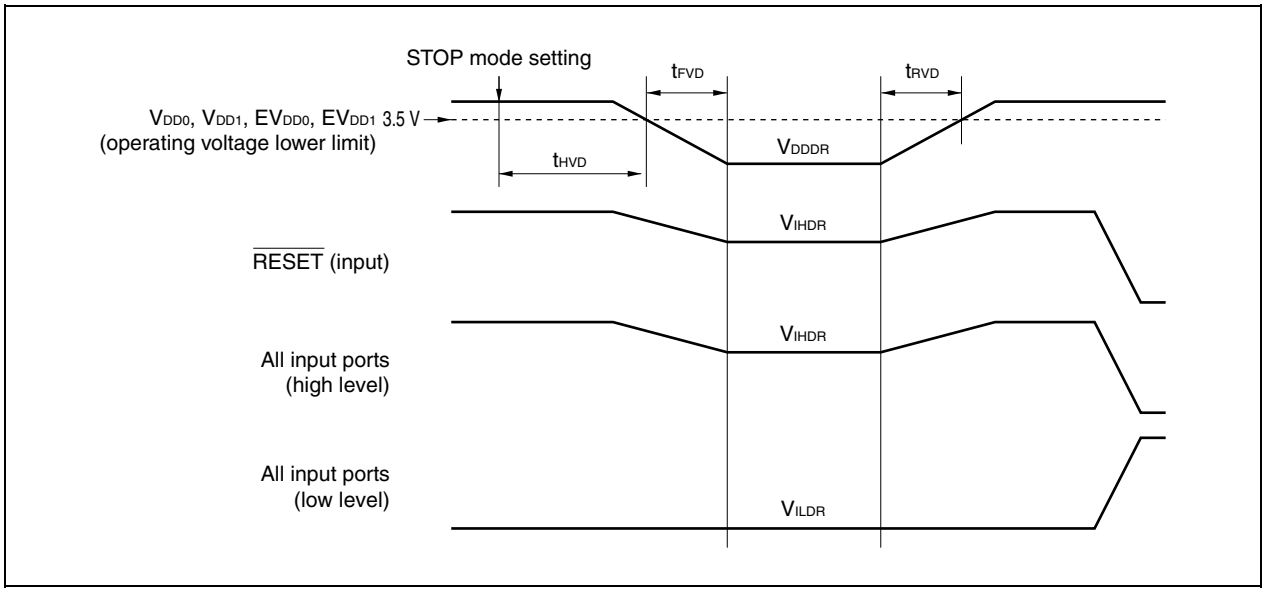
28.1.7 Data retention characteristics

STOP mode (T_A = –40 to +85°C, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention voltage	V_{DDDR}	STOP mode	**Note**		5.5	V
Data retention current	I_{DDDR}	V_{DD0} = V_{DD1} = V_{DDDR}		40	800	μA
Supply voltage rise time	t_{RVD}		1			μs
Supply voltage fall time	t_{FVD}		1			μs
Supply voltage hold time (from STOP mode setting)	t_{HVD}		0			ms
Data retention input voltage, high	V_{IHDR}	All input ports	0.9V_{DDDR}		V_{DDDR}	V
Data retention input voltage, low	V_{ILDR}	All input ports	EV_{SS}		0.1V_{DDDR}	V

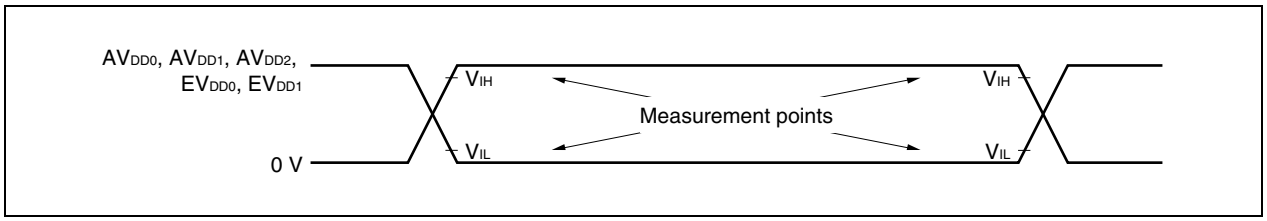
Note When the low-voltage detector (LVI) reset mode is not used (LVIM.LVIMD bit = 0): POC detection voltage (V_{POC0})

When the low-voltage detector (LVI) reset mode is used (LVIM.LVIMD bit = 1): LVI detection voltage (V_{LVI0}/V_{LVI1})

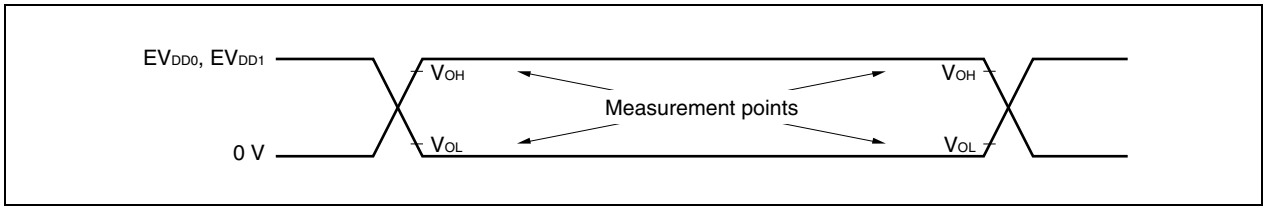


28.1.8 AC characteristics

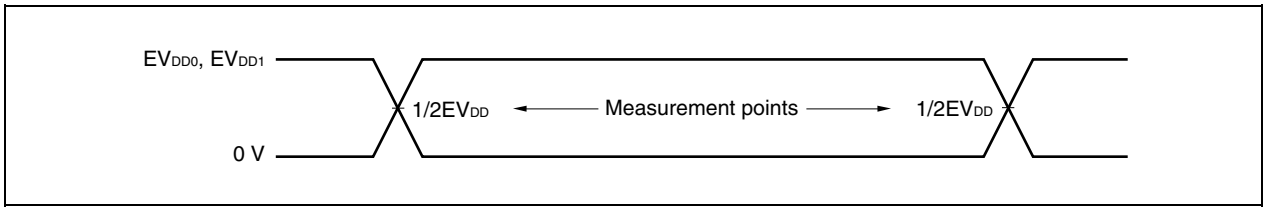
AC Test Input Measurement Points (Pins Other than CSIB0 to CSIB2)



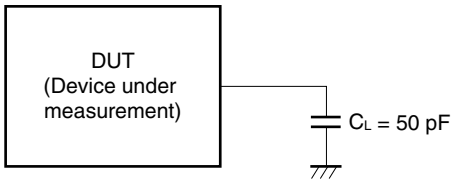
AC Test Output Measurement Points (Pins Other than CSIB0 to CSIB2)



AC Test I/O Measurement Points (CSIB0 to CSIB2 Pins)



Load Conditions

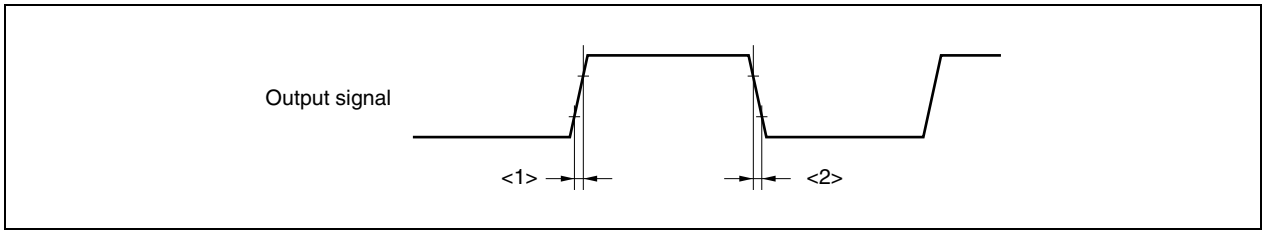


Caution If the load capacitance exceeds 50 pF due to the circuit configuration, bring the load capacitance of the device to 50 pF or less by inserting a buffer or by some other means.

(1) Output signal timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output rise time	t_{OR}	<1>	PDL0 to PDL9		8	ns
			Other than above		15	ns
Output fall time	t_{OF}	<2>	PDL0 to PDL9		8	ns
			Other than above		15	ns



(2) Reset, external interrupt timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

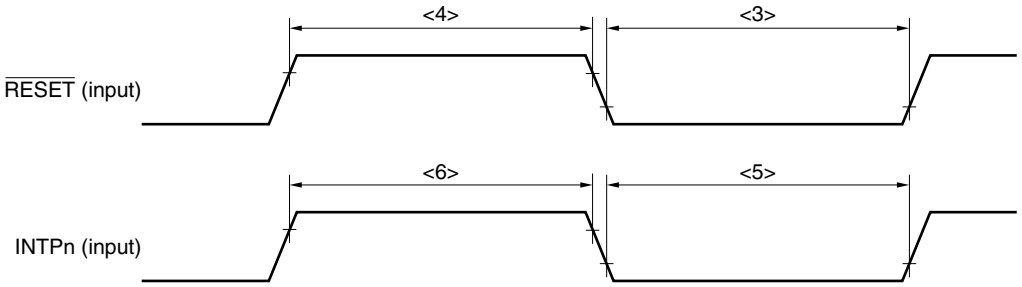
Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ low-level width	t_{WRSL}	<3>	Power is on, STOP mode is released	500 + T_{os}		ns
			Other than above	500		ns
$\overline{\text{RESET}}$ high-level width	t_{WRSH}	<4>		500		ns
INTPn low-level width	t_{WITL}	<5>	n = 00, 01, 08 to 13, 17, 18 (analog noise elimination)	500		ns
			n = 14 to 16 (digital noise elimination)	$4T_{smp}$		ns
INTPn high-level width	t_{WITH}	<6>	n = 00, 01, 08 to 13, 17, 18 (analog noise elimination)	500		ns
			n = 14 to 16 (digital noise elimination)	$4T_{smp}$		ns

Remarks 1. T_{os}: Oscillation stabilization time

T_{smp}: Noise elimination sampling clock cycle (set by INTNFCn register)

2. After reset release, a 1 ms oscillation stabilization time is internally secured when the oscillation frequency (f_X) = 8 MHz. The oscillation stabilization time is therefore (T_{os} + 1) ms. After STOP mode release, an oscillation stabilization time half the value set to the OSTS register is internally secured. Therefore, T_{os} = 0 ns is acceptable if sufficient stabilization time can be secured by the OSTS register setting.

Reset/Interrupt



Remark n = 00, 01, 08 to 13, 17, 18

(3) Timer timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
TIn high-level width[Notes 1, 2]	t_{WTIHn}	<7>	n = B00 to B03, B10 to B13	12T + 10		ns
			n = A20, A21, A40, A41	$3T_{smp1}$ + 10		ns
TIn low-level width[Notes 1, 2]	t_{WTILn}	<8>	n = B00 to B03, B10 to B13	12T + 10		ns
			n = A20, A21, A40, A41	$3T_{smp1}$ + 10		ns
EVTBm high-level width[Note 1]	t_{WEVBHm}	<9>	m = 0, 1	12T + 10		ns
EVTBm low-level width[Note 1]	t_{WEVBLm}	<10>	m = 0, 1	12T + 10		ns
TRGBm high-level width[Note 1]	t_{WTRHm}	<11>	m = 0, 1	12T + 10		ns
TRGBm low-level width[Note 1]	t_{WTRLm}	<12>	m = 0, 1	12T + 10		ns
TENC10/TENC11 high-level width[Note 3]	t_{WENCH1}	<13>		$3T_{smp2}$ + 10		ns
TENC10/TENC11 low-level width[Note 3]	t_{WENCL1}	<14>		$3T_{smp2}$ + 10		ns
TECR1 high-level width[Note 3]	t_{WCRH1}	<15>		$3T_{smp2}$ + 10		ns
TECR1 low-level width[Note 3]	t_{WCRL1}	<16>		$3T_{smp2}$ + 10		ns
TIT10/TIT11 high-level width[Note 3]	t_{WTITH1}	<17>		$3T_{smp2}$ + 10		ns
TIT10/TIT11 low-level width[Note 3]	t_{WTITL1}	<18>		$3T_{smp2}$ + 10		ns
EVTT1 high-level width[Note 3]	t_{WEVTH1}	<19>		$3T_{smp2}$ + 10		ns
EVTT1 low-level width[Note 3]	t_{WEVTL1}	<20>		$3T_{smp2}$ + 10		ns
TENC10/TENC11 input time differential[Note 3]	t_{PHUD1}	<21>		$3T_{smp2}$ + 10		ns

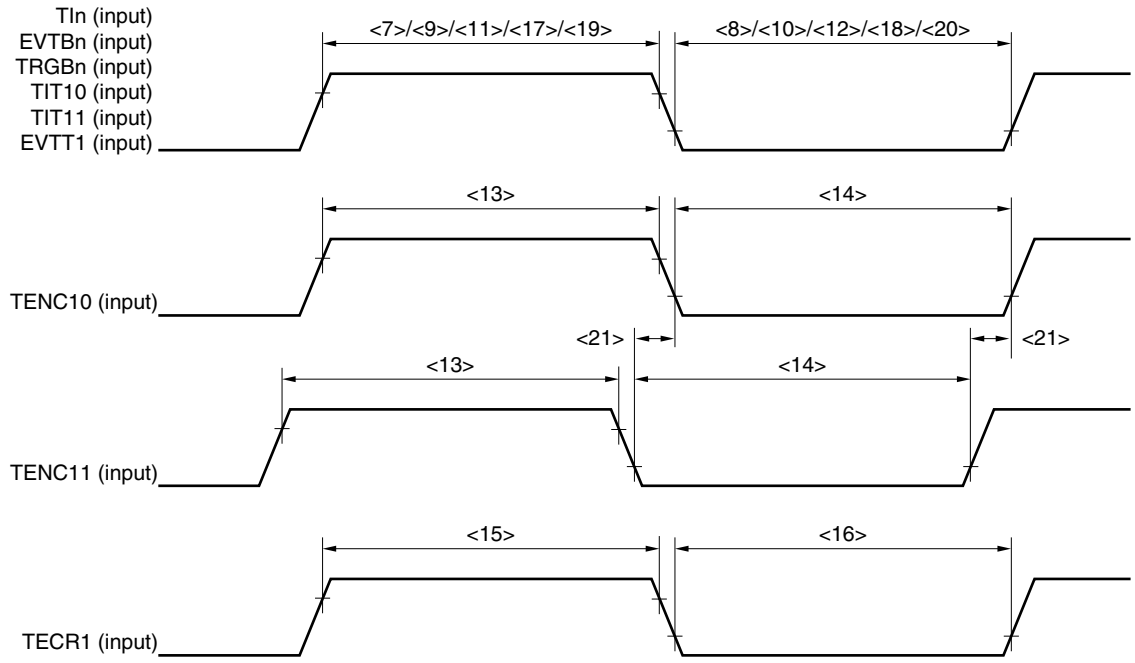
Notes 1. T = 1/fxx

2. T_{smp1}: Noise elimination sampling clock cycle (set by TANFC2 and TANFC4 registers)

3. T_{smp2}: Noise elimination sampling clock cycle (set by TTNFC1 register)

Remark The above specification shows a pulse width that is accurately detected as a valid edge. Even if a pulse narrower than the above specification is input, therefore, it may be detected as a valid edge.

Timer Input Timing



Remark n = A20, A21, A40, A41, B00 to B03, B10 to B13

(4) CSIB Timing

(a) Master mode

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKBn}$ cycle	t_{KCYM}	<22>		125		ns
$\overline{SCKBn}$ high-/low-level width	t_{KWHM}, t_{KWLM}	<23>		$t_{KCYM}/2 - 10$		ns
SIBn setup time (to $\overline{SCKBn}\uparrow$)	t_{SSIM}	<24>		30		ns
SIBn setup time (to $\overline{SCKBn}\downarrow$)				30		ns
SIBn hold time (from $\overline{SCKBn}\uparrow$)	t_{HSIM}	<25>		30		ns
SIBn hold time (from $\overline{SCKBn}\downarrow$)				30		ns
SOBn output delay time (from $\overline{SCKBn}\downarrow$)	t_{DSOM}	<26>			30	ns
SOBn output delay time (from $\overline{SCKBn}\uparrow$)					30	ns
SOBn output hold time (from $\overline{SCKBn}\uparrow$)	t_{HSOM}	<27>		$t_{KCYM}/2 - 10$		ns
SOBn output hold time (from $\overline{SCKBn}\downarrow$)				$t_{KCYM}/2 - 10$		ns

Remark n = 0 to 2

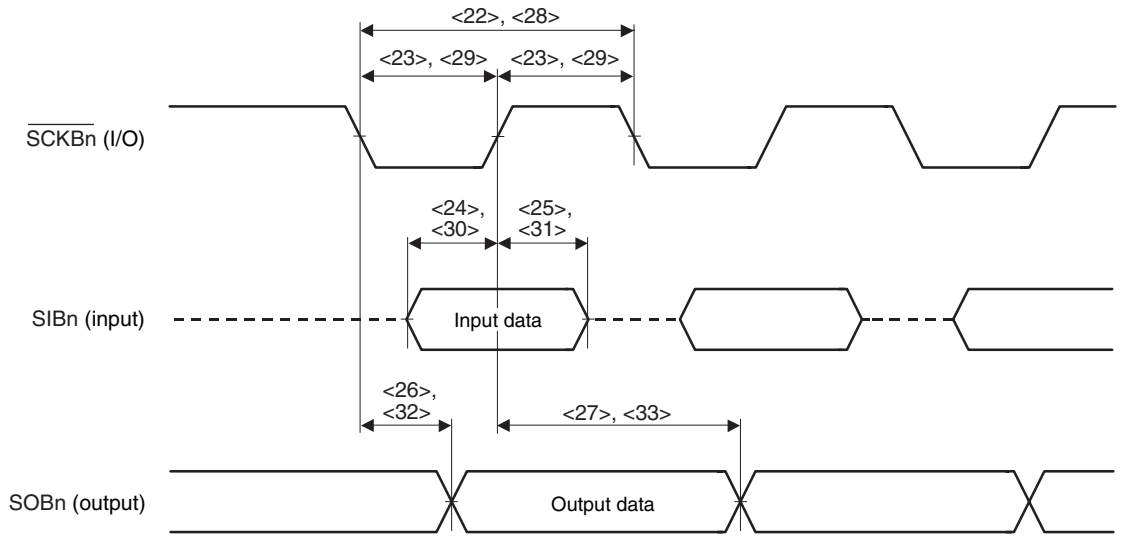
(b) Slave mode

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKBn}$ cycle	t_{KCYS}	<28>		125		ns
$\overline{SCKBn}$ high-/low-level width	t_{KWHS}, t_{KWLS}	<29>		$t_{KCYS}/2 - 10$		ns
SIBn setup time (to $\overline{SCKBn}\uparrow$)	t_{SSIS}	<30>		30		ns
SIBn setup time (to $\overline{SCKBn}\downarrow$)				30		ns
SIBn hold time (from $\overline{SCKBn}\uparrow$)	t_{HSIS}	<31>		30		ns
SIBn hold time (from $\overline{SCKBn}\downarrow$)				30		ns
SOBn output delay time (from $\overline{SCKBn}\downarrow$)	t_{DSOS}	<32>			30	ns
SOBn output delay time (from $\overline{SCKBn}\uparrow$)					30	ns
SOBn output hold time (from $\overline{SCKBn}\uparrow$)	t_{HSOS}	<33>		$t_{KCYS}/2 - 10$		ns
SOBn output hold time (from $\overline{SCKBn}\downarrow$)				$t_{KCYS}/2 - 10$		ns

Remark n = 0 to 2

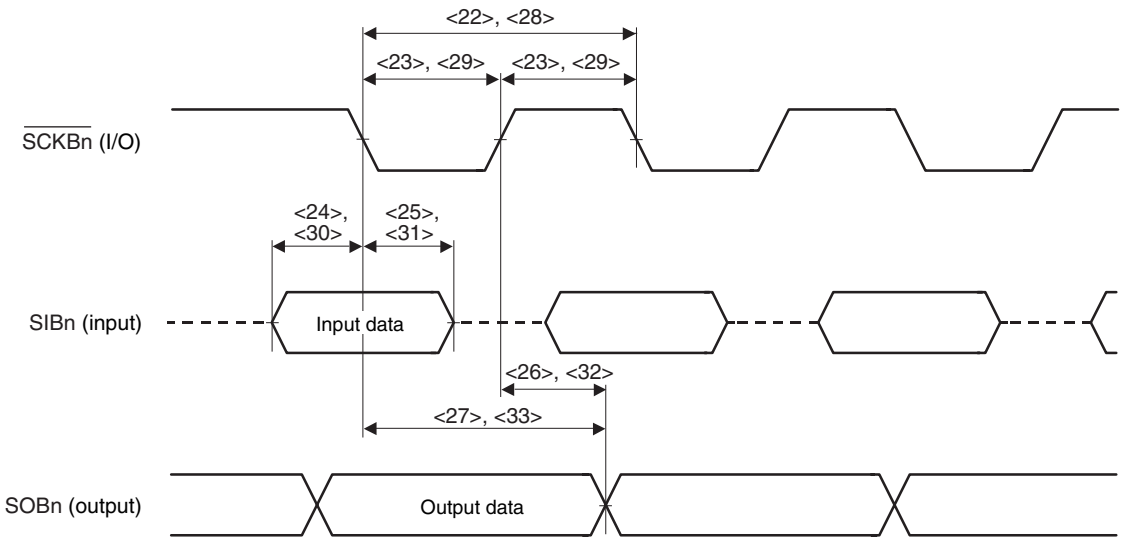
CSIB timing when CBnCKP and CBnDAP bits of CBnCTL1 register = 00



Remarks 1. Broken lines indicate high impedance.

2. n = 0 to 2

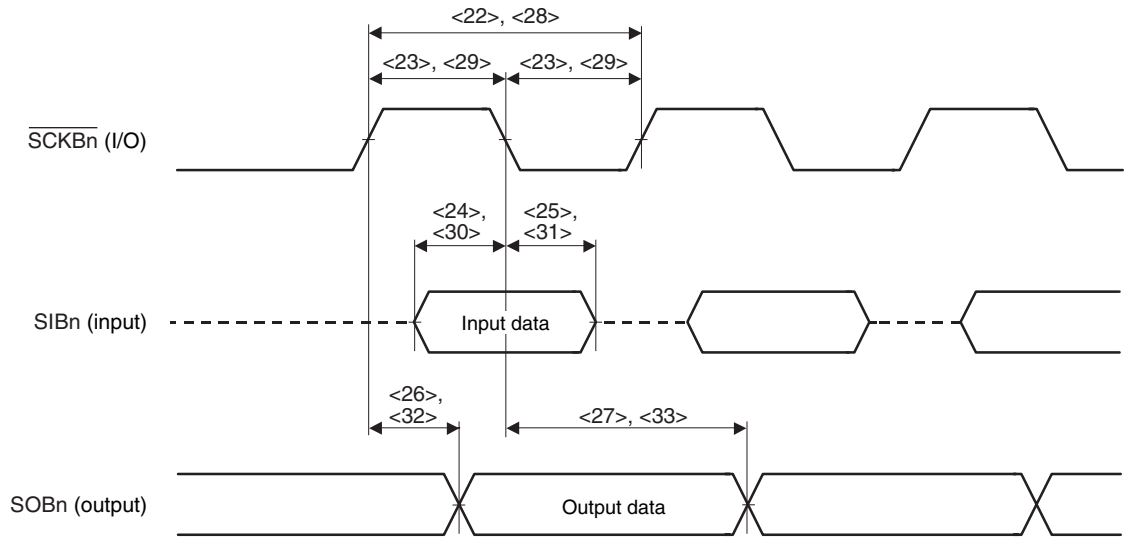
CSIB timing when CBnCKP and CBnDAP bits of CBnCTL1 register = 01



Remarks 1. Broken lines indicate high impedance.

2. n = 0 to 2

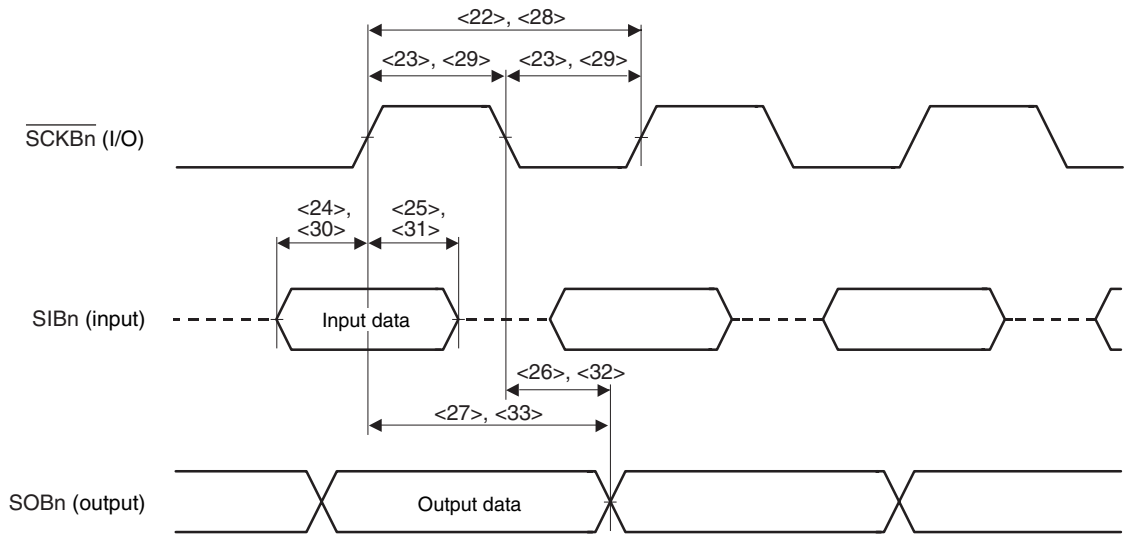
CSIB timing when CBnCKP and CBnDAP bits of CBnCTL1 register = 10



Remarks 1. Broken lines indicate high impedance.

2. n = 0 to 2

CSIB timing when CBnCKP and CBnDAP bits of CBnCTL1 register = 11



Remarks 1. Broken lines indicate high impedance.

2. n = 0 to 2

(5) I^2C bus timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter		Symbol		Standard Mode		High-Speed Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCL clock frequency		f_{CLK}	–	0	100	0	400	kHz
Bus free time (between stop condition and start condition)		t_{BUF}	<34>	4.7	–	1.3	–	μs
Hold time[Note 1]		$t_{HD:STA}$	<35>	4.0	–	0.6	–	μs
SCL clock low-level width		t_{LOW}	<36>	4.7	–	1.3	–	μs
SCL clock high-level width		t_{HIGH}	<37>	4.0	–	0.6	–	μs
Start/restart condition setup time		$t_{SU:STA}$	<38>	4.7	–	0.6	–	μs
Data hold time	CBUS-compatible master	$t_{HD:DAT}$	<39>	5.0	–	–	–	μs
	I^2C mode			0[Note 2]	–	0[Note 2]	0.9[Note 3]	μs
Data setup time		$t_{SU:DAT}$	<40>	250	–	100[Note 4]	–	ns
SDA, SCL signal rise time		t_R	<41>	–	1000	20 + 0.1Cb[Note 5]	300	ns
SDA, SCL signal fall time		t_F	<42>	–	300	20 + 0.1Cb[Note 5]	300	ns
Stop condition setup time		$t_{SU:STO}$	<43>	4.0	–	0.6	–	μs
Pulse width of spike suppressed by input filter		t_{SP}	<44>	–	–	0	50	ns
Each bus line capacitive load		Cb	–	–	400	–	400	pF

Notes 1. The first clock pulse is generated after a hold time during the start condition.

2. The system must internally supply a hold time of at least 300 ns for the SDA signal (at V_{IHmin}. of SCL signal) to fill the undefined area at the falling edge of SCL.

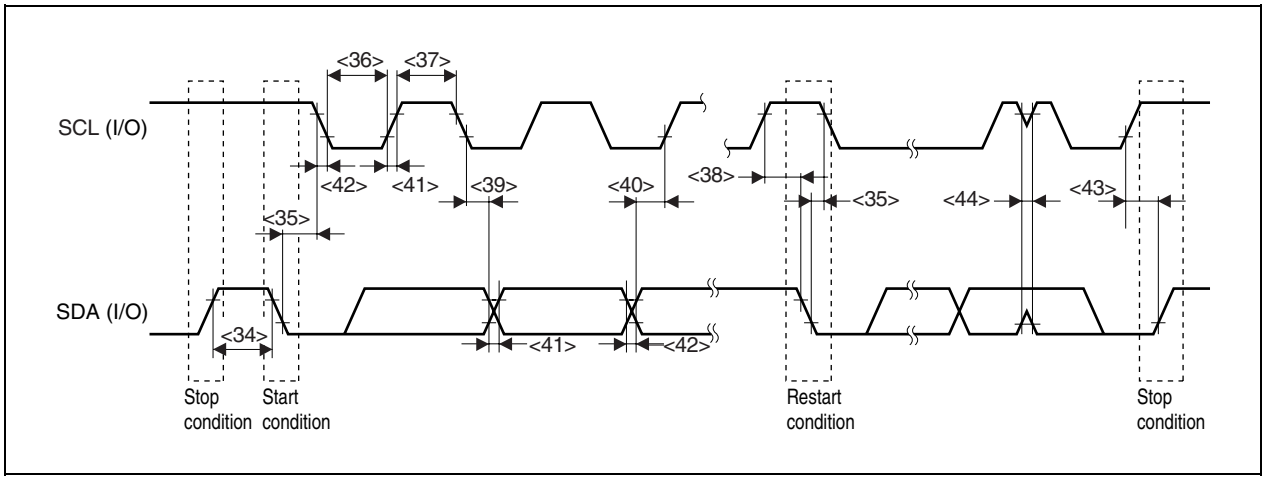
3. If the system does not extend the low hold time (t_{LOW}) of the SCL signal, the maximum data hold time ($t_{HD:DAT}$) must be satisfied.

4. The high-speed mode I^2C bus can be used in the standard mode I^2C bus system. In this case, make sure that the following conditions are satisfied.

- If system does not extend the low status hold time of the SCL signal
  $t_{SU:DAT} \geq 250$ ns
- If system extends the low status hold time of SCL signal
  Sends the next data bit to the SDA line before the SCL line is released ($t_{Rmax.} + t_{SU:DAT}$ = 1000 + 250 = 1250 ns: standard mode I^2C bus specification).

5. Cb: Total capacitance of one bus line (unit: pF)

I²C bus timing



(6) High-impedance control timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Oscillation stop → timer output high impedance	t_{CLM}	When clock monitor is operating		65	μs
Input to TOBnOFF → timer output high impedance	t_{HTQn}			300	ns
Input to TOA2OFF → timer output high impedance	t_{HTP2}			300	ns
Input to ANI00/ANI05 → timer output high impedance	t_{ANI0}			10	μs
Input to ANI10/ANI15 to ANI12/ANI17 → timer output high impedance	t_{ANI1}			10	μs

Remark n = 0, 1

28.1.9 Characteristics of A/D converters 0, 1

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = AV_{REFP0} = AV_{REFP1} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			12	12	12	bit
Overall error[Note 1]					±10	LSB
Conversion time	t_{CONV}	f_{AD01} = 16 MHz, ADAnCTC register = 0BH or 0CH	2.0			μs
		f_{AD01} = 12 MHz, ADAnCTC register = 00H			7.42	μs
Zero scale error[Note 1]					±10	LSB
Full-scale error[Note 1]					±10	LSB
Integral linearity error[Note 1]					±4	LSB
Differential linearity error[Note 1]					±2.5	LSB
Analog reference voltage	AV_{DD}		4.0		5.5	V
Analog input voltage	V_{IAN}		AV_{SS}		AV_{DD}	V
AV_{DD} supply current[Note 2]	AI_{DD}	Operating		4.5	7.5	mA
	AI_{DDS}	In STOP mode[Note 3]		3.5	17.5	μs

Notes 1. Excludes quantization error (±0.5 LSB).

2. This value is for only one A/D converter (A/D converter 0 or 1).

3. Stop the operation of A/D converters 0 and 1 (ADnSCM.ADnCE bit = 0) before setting STOP mode.

Remarks 1. LSB: Least Significant Bit

2. f_{AD01}: Base clock of A/D converters 0 and 1

3. n = 0, 1

28.1.10 Characteristics of A/D converter 2

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Note 1]					±4.0	LSB
Conversion time	t_{CONV}		3.88		10	μs
Zero scale error[Note 1]					±4.0	LSB
Full-scale error[Note 1]					±4.0	LSB
Integral linearity error[Note 1]					±4.0	LSB
Differential linearity error[Note 1]					±2.0	LSB
Analog reference voltage	AV_{DD}		4.0		5.5	V
Analog input voltage	V_{IAN}		AV_{SS}		AV_{DD}	V
AV_{DD} supply current	AI_{DD}	During operation		3.5	7	mA
	AI_{DDS}	In STOP mode[Note 2]		1	10	μA

Notes 1. Excludes quantization error (±0.5 LSB).

2. Stop the operation of A/D converter 2 (AD2M0.AD2CE bit = 0) before setting STOP mode.

Remark LSB: Least Significant Bit

28.1.11 Operational amplifier characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IO}				±9.0		mV
Input voltage range	V_I	Gain = 2.500		0.04AV_{DD}		0.36AV_{DD}	V
		Gain = 5.000		0.02AV_{DD}		0.18AV_{DD}	V
		Gain = 10.00		0.01AV_{DD}		0.085AV_{DD}	V
Slew rate[Note 1]	S_R			10	15		V/μs
Gain error		**Note 2**	Gain = 2.500 to 4.444		±1.0	±1.3	%
			Gain = 5.000 to 6.667		±1.0	±1.5	%
			Gain = 8.000, 10.00		±1.0	±1.7	%
		Note 3	Gain = 2.500 to 4.444		±1.0	±2.0	%
			Gain = 5.000 to 6.667		±1.0	±2.1	%
			Gain = 8.000, 10.00		±1.0	±2.2	%
Operating current[Note 4]	I_{OPDD}	During operation			1.8	2.6	mA
	A_{IDDS}	In STOP mode[Note 5]			1.0	10	μA

Notes 1. Inclination characteristic of output voltage from 10% to 90%

2. AV_{DD0} = AV_{DD1} = 4.5 to 5.5 V

3. AV_{DD0} = AV_{DD1} = 4.0 to 5.5 V

4. Four operational amplifiers are provided in total. The value shows the operating current per operational amplifier.

5. Stop operational amplifier operation (OP0CTL0.OP0EN bit = 0, OP1CTL0.OP12EN, OP11EN, and OP10EN bits = 0)) before setting STOP mode.

Remark Power supplies AV_{DD0} and AV_{DD1} are used for the operational amplifier.

28.1.12 Comparator characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IO}			±3.0		mV
Input voltage range	V_I		AV_{SS}		AV_{DD}	V
Comparator reference voltage (full range)	CREFnF		$0.02AV_{DD}$ + 0.1		$0.92AV_{DD}$ – 0.1	V
Comparator reference voltage (low range)	CREFnL		$0.02AV_{DD}$ + 0.1		$0.5AV_{DD}$ – 0.1	V
Response time	t_{CR}	Input amplitude = 100 mV, at rising edge[Note 1]		1.0		μs
	t_{CF}	Input amplitude = 100 mV, at falling edge[Note 2]		1.0		μs
Operating current[Note 3]	I_{CPDD}	During operation			250	μA
	A_{IDDS}	In STOP mode[Note 4]		2.0	20	nA

Notes 1. Characteristics of pulse response when ANIm input changes from the comparator reference voltage – 100 mV to the comparator reference voltage + 100 mV

2. Characteristics of pulse response when ANIm input changes from the comparator reference voltage + 100 mV to the comparator reference voltage – 100 mV

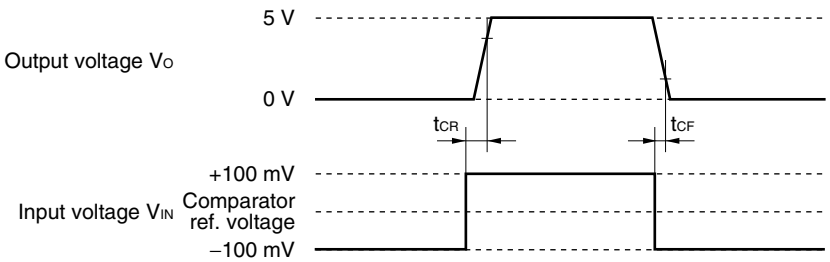
3. Four comparators are provided in total. The value shows the operating current per comparator.

4. Stop comparator operation (CMPnCTL0 register = 00H) before setting STOP mode.

Remarks 1. Power supplies for the comparators are AV_{DD0} and AV_{DD1}.

2. m = 05, 15 to 17
n = 0, 1

Comparator Characteristics



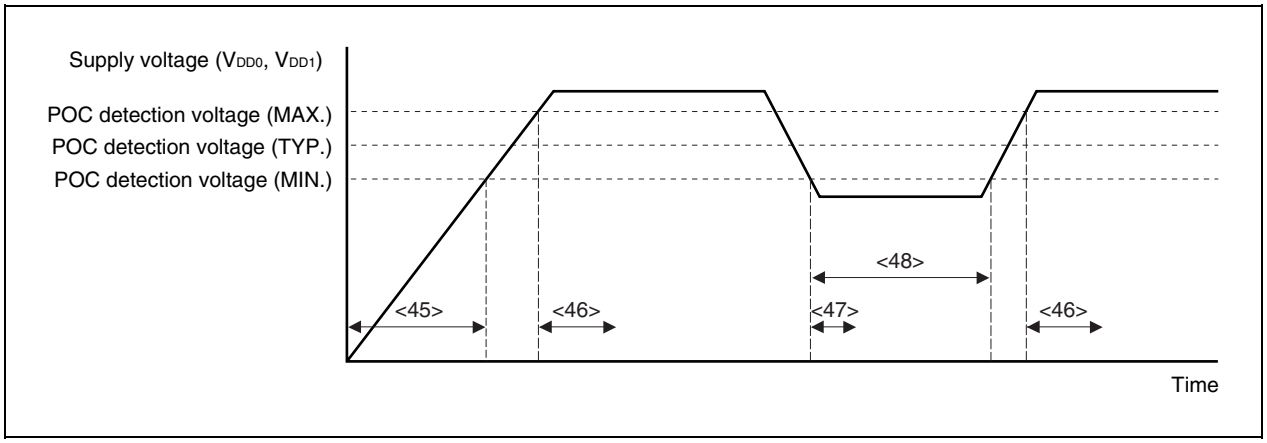
28.1.13 Power-on-clear circuit (POC)

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	TYP.	MAX.	Unit
POC detection voltage	V_{POC0}			3.5	3.7	3.9	V
Supply voltage rise time	t_{PTH}	<45>	V_{DD0}, V_{DD1} = 0 to 3.5 V	2.5 μs		1.8 s	
Response time 1[Note 1]	t_{PTHD}	<46>	After V_{DD0} and V_{DD1} reach 3.9 V on power application			3.0	ms
Response time 2[Note 2]	t_{PD}	<47>	After V_{DD0} and V_{DD1} drop to 3.5 V on power off			1.0	ms
Minimum width of V_{DD0}, V_{DD1}	t_{PW}	<48>		0.2			ms

Notes 1. The time required to release a reset signal (POCRES) after the POC detection voltage is detected.

2. The time required to output a reset signal (POCRES) after the POC detection voltage is detected.

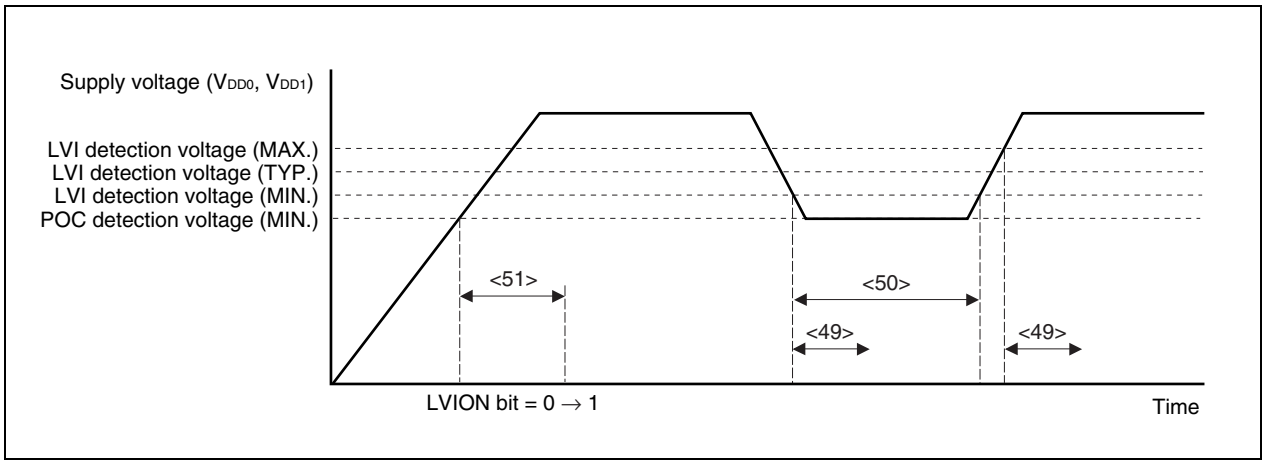


28.1.14 Low-voltage detector (LVI)

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	TYP.	MAX.	Unit
LVI detection voltage	V_{LVI0}		LVIS.LVIS0 bit = 0	4.2	4.4	4.6	V
	V_{LVI1}		LVIS.LVIS0 bit = 1	4.0	4.2	4.4	V
Response time 1[Note]	t_{LD}	<49>	After V_{DD0} and V_{DD1} reach V_{LVI0}/V_{LVI1} (MAX.) or drop to V_{LVI0}/V_{LVI1} (MIN.)		0.2	2.0	ms
Minimum width of V_{DD0}, V_{DD1}	t_{LW}	<50>		0.2			ms
Reference voltage stabilization wait time	t_{LWAIT}	<51>	After V_{DD0} and V_{DD1} reach POC detection voltage (MIN.) and the LVIM.LVION bit is changed from 0 to 1		0.1		ms

Note The time required to output an interrupt request signal (INTLVIL, INTLVIH) or internal reset signal (LVIRES) after the LVI detection voltage is detected.



28.1.15 Flash memory programming characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Rewrite count	C_{ERWR}	**Note**		100		Times

Note Rewrite as follows.

Example when three rewrites: Shipped product →E→P→E→P→E→P (P: Write, E: Erase)

28.2 V850E/IG3

28.2.1 Absolute maximum ratings

(T_A = 25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}	$V_{DDa} = EV_{DDb} = AV_{DDk}$		–0.5 to +6.5	V
	V_{SS}	$V_{SSa} = EV_{SSb} = AV_{SSk}$		–0.5 to +0.5	V
	EV_{DD}	$V_{DDa} = EV_{DDb} = AV_{DDk}$		–0.5 to +6.5	V
	EV_{SS}	$V_{SSa} = EV_{SSb} = AV_{SSk}$		–0.5 to +0.5	V
	AV_{DD}	$V_{DDa} = EV_{DDb} = AV_{DDk}$		–0.5 to +6.5	V
	AV_{SS}	$V_{SSa} = EV_{SSb} = AV_{SSk}$		–0.5 to +0.5	V
Input voltage	V_{I1}	**Note 1**		–0.5 to EV_{DD} + 0.5 Note 2	V
	V_{I2}	X1, X2		–0.5 to V_{RO} + 0.35	V
Output current, low	I_{OL}	All pins	Per pin	4	mA
			Total of all pins	63	mA
Output current, high	I_{OH}	All pins	Per pin	–4	mA
			Total of all pins	–63	mA
Analog input voltage	V_{IAN}	P70/ANI20 to P77/ANI27, ANI00 to ANI05, ANI10 to ANI17		–0.5 to AV_{DD} + 0.5 Note 2	V
Analog reference input voltage	V_{IREF}	AV_{REFP0}, AV_{REFP1}		–0.5 to AV_{DD} + 0.5 Note 2	V
Comparator reference input voltage	V_{CREF}	CREF0L, CREF1L, CREF0F, CREF1F		–0.5 to AV_{DD} + 0.5 Note 2	V
Operating ambient temperature	T_A	In normal operating mode		–40 to +85	°C
		In flash memory programming mode		–40 to +85	°C
Operating ambient temperature	T_{stg}			–40 to +125	°C

Notes 1. P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, PDL0 to PDL15, $\overline{\text{RESET}}$, FLMD0, $\overline{\text{DRST}}$

2. Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

Cautions 1. Do not directly connect the output pins (or I/O pins in the output state) of IC products to other output pins (including I/O pins in the output state), power supply pins such as V_{DD} and EV_{DD}, or GND pin. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

The ratings and conditions indicated for DC characteristics and AC characteristics represent the quality assurance range during normal operation.

Remark a = 0, 1

b = 0 to 2

k = 0 to 2

28.2.2 Capacitance

(T_A = 25°C, V_{DD0} = V_{SS0} = V_{DD1} = V_{SS1} = EV_{DD0} = EV_{SS0} = EV_{DD1} = EV_{SS1} = EV_{DD2} = EV_{SS2} = AV_{DD0} = AV_{SS0} = AV_{DD1} = AV_{SS1} = AV_{DD2} = AV_{SS2} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	fc = 1 MHz Unmeasured pins returned to 0 V	**Note 1**			15	pF
I/O capacitance	C_{IO}		**Note 2**			15	pF
Output capacitance	C_O		**Note 3**			15	pF

Notes 1. ANI00 to ANI05, ANI10 to ANI17, $\overline{\text{RESET}}$

2. P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P70 to P77, PDL0 to PDL15

3. DDO

Cautions 1. Excludes the FLMD0, $\overline{\text{DRST}}$, X1, and X2 pins.

2. In addition to input capacitance, sampling capacitance is added to the ANI00 to ANI05, ANI10 to ANI17, and ANI20 to ANI27 pins when sampling.

28.2.3 Operating conditions

(T_A = –40 to +85°C, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V)

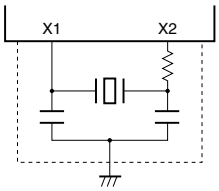
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
System clock frequency	f_{XX}	PLL mode		32		64	MHz
		Clock through mode		4		8	MHz
CPU clock frequency	f_{CPU}	PLL mode		4		64	MHz
		Clock through mode		0.5		8	MHz
V_{DD}, EV_{DD} voltage	V_{DD}, EV_{DD}	**Note 1**	When external bus is not used	3.5		5.5	V
			When external bus is used[Note 2]	4.0		5.5	V
AV_{DD} voltage	AV_{DD}	When A/D converters 0 to 2 are operating		4.0		5.5	V
		When A/D converters 0 to 2 are not operating		3.5		5.5	V

Notes 1. V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2}

2. μPD70F3454GC-8EA-A only

28.2.4 Clock oscillator characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic /crystal resonator		Oscillation frequency (f_X)		4		8	MHz
		Oscillation stabilization time	After reset release		$2^{14}/f_X$		ms
			After STOP mode release		**Note**		ms

Note The value varies depending on the setting of the oscillation stabilization time select register (OSTS).

Cautions 1. Connect the oscillator as close to the X1 and X2 pins as possible.

2. Do not cross the wiring with the other signal lines in the area enclosed by the broken lines in the above figure.

3. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

4. Inputting an external clock to the V850E/IG3 is prohibited.

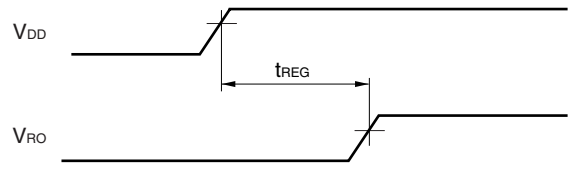
28.2.5 Regulator characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = 3.5 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage	V_{DD}		3.5		5.5	V
Output voltage	V_{RO}			1.5		V
Output voltage stabilization time	t_{REG}	Stabilizing capacitor C = 4.7 μF[Note 1]			1[Note 2]	ms

Notes 1. Connect a stabilizing capacitor between the REGC0 pin and V_{SS0} pin, and between the REGC1 pin and V_{SS1} pin.

2. Internal reset signal is output until the power-on-clear circuit (POC) output voltage stabilizes during t_{REG} period.



28.2.6 DC characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V) (1/2)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	**Note 1**			0.7EV_{DD}		EV_{DD}	V
	V_{IH2}	**Note 2**			0.8EV_{DD}		EV_{DD}	V
	V_{IH3}	**Note 3**			2.2		EV_{DD}	V
	V_{IH4}	**Note 4**			0.7AV_{DD}		AV_{DD}	V
Input voltage, low	V_{IL1}	**Note 1**			EV_{SS}		0.3EV_{DD}	V
	V_{IL2}	**Note 2**			EV_{SS}		0.2EV_{DD}	V
	V_{IL3}	**Note 3**			EV_{SS}		0.8	V
	V_{IL4}	**Note 4**			AV_{SS}		0.3AV_{DD}	V
Input leakage current, high	I_{LIH1}	V_I = **Note 5**, **Note 6**	Other than X1				5	μA
	I_{LIH2}		X1				20	μA
Input leakage current, low	I_{LIL1}	V_I = 0 V	Other than X1				–5	μA
	I_{LIL2}		X1				–20	μA
Output leakage current, high	I_{LOH}	V_O = **Note 5**					5	μA
Output leakage current, low	I_{LOL}	V_O = 0 V					–5	μA
Output voltage, high	V_{OH1}	**Note 7**	I_{OH} = –1.0 mA	Total of pins = –57 mA	EV_{DD} – 1.0			V
Output voltage, low	V_{OL1}	**Note 7**	I_{OL} = 1.0 mA	Total of pins = 57 mA			0.4	V
Pull-up resistor	R_{L1}				10	30	100	kΩ
Pull-down resistor[Note 8]	R_{L2}				10	30	100	kΩ

Notes 1. P33, P36, P41, and PDL0 to PDL15 pins

2. P00 to P07, P10 to P17, P20 to P27, P30 to P32, P34, P35, P37, P40, P42 to P47, $\overline{\text{RESET}}$, and FLMD0 pins

3. $\overline{\text{DRST}}$, DDI, DCK, and DMS pins

4. P70 to P77 pins

5. AV_{DD0} = AV_{DD1} = AV_{DD2} = EV_{DD0} = EV_{DD1} = EV_{DD2}

6. Except for $\overline{\text{DRST}}$ pin

7. P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, PDL0 to PDL15, and DDO pins

8. $\overline{\text{DRST}}$ pin only

Remark The characteristics of alternate-function pins are the same as those of port pins.

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V) (2/2)

Parameter	Symbol	Conditions		MIN.	TYP.[Note 1]	MAX.	Unit
V_{DD} supply current[Note 2]	I_{DD1}	f_{XX} = 64 MHz	Normal operation		64	93	mA
	I_{DD2}		HALT mode		42	60	mA
	I_{DD3}		IDLE mode		5	10	mA
	I_{DD4}	STOP mode			40	800	μA

Notes 1. The TYP. value is a reference value when V_{DD0} = V_{DD1} = 5.0 V and T_A = 25°C.

2. The current consumed by the EV_{DD} system (output buffer and pull-up resistor) and the operating currents of A/D converters 0 to 2, the operational amplifier, and the comparator are not included.

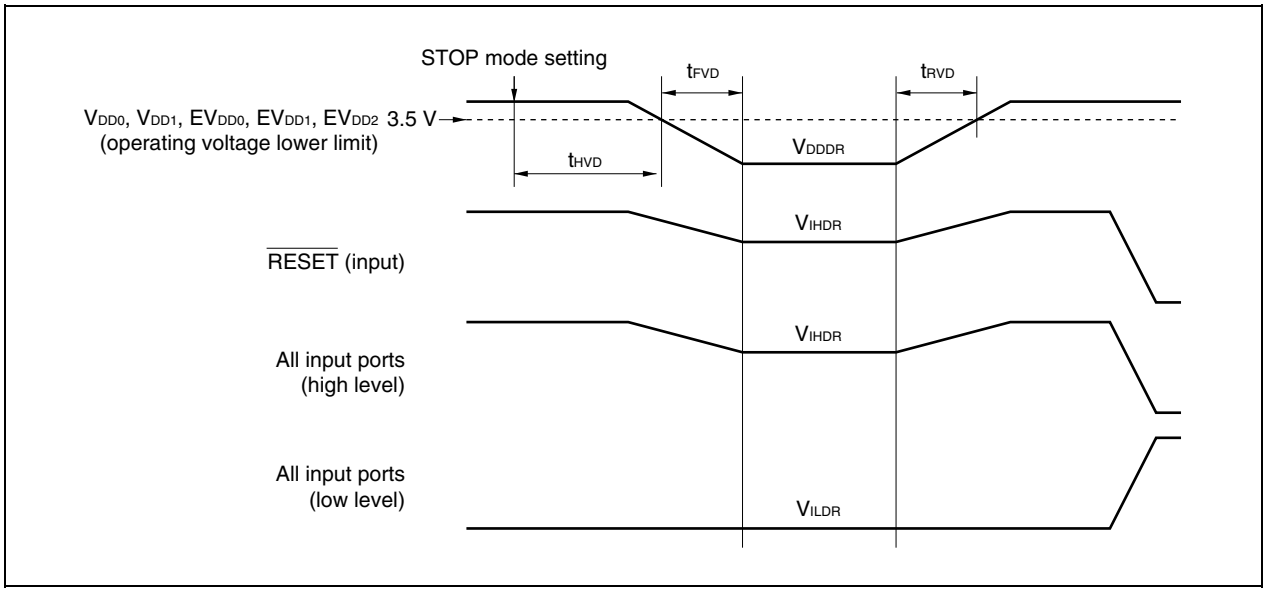
28.2.7 Data retention characteristics

STOP mode (T_A = –40 to +85°C, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention voltage	V_{DDDR}	STOP mode	**Note**		5.5	V
Data retention current	I_{DDDR}	V_{DD0} = V_{DD1} = V_{DDDR}		40	800	μA
Supply voltage rise time	t_{RVD}		1			μs
Supply voltage fall time	t_{FVD}		1			μs
Supply voltage retention time (from STOP mode setting)	t_{HVD}		0			ms
Data retention input voltage, high	V_{IHDR}	All input ports	0.9V_{DDDR}		V_{DDDR}	V
Data retention input voltage, low	V_{ILDR}	All input ports	EV_{SS}		0.1V_{DDDR}	V

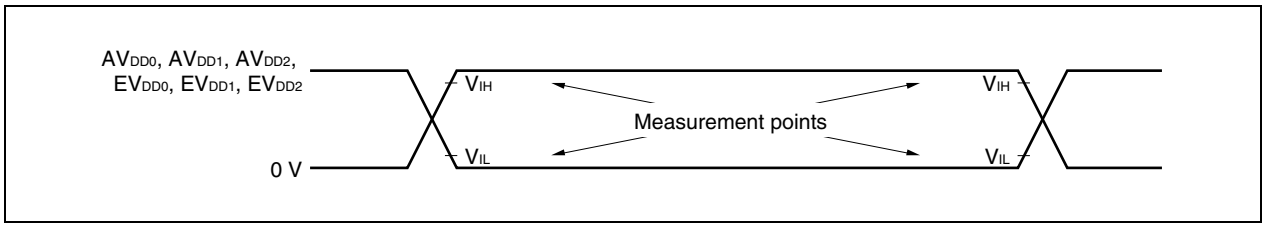
Note When the low-voltage detector (LVI) reset mode is not used (LVIM.LVIMD bit = 0): POC detection voltage (V_{POC0})

When the low-voltage detector (LVI) reset mode is used (LVIM.LVIMD bit = 1): LVI detection voltage (V_{LVI0}/V_{LVI1})

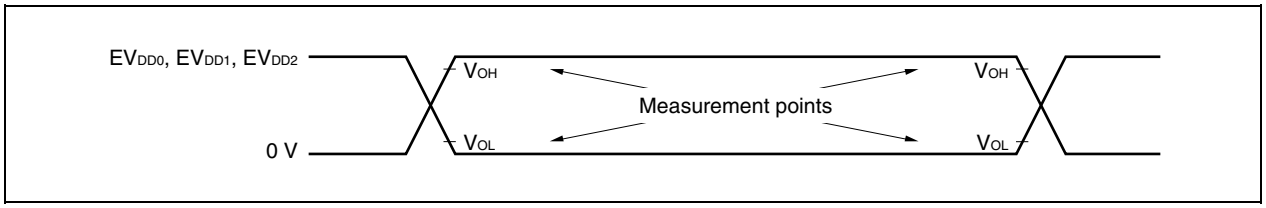


28.2.8 AC characteristics

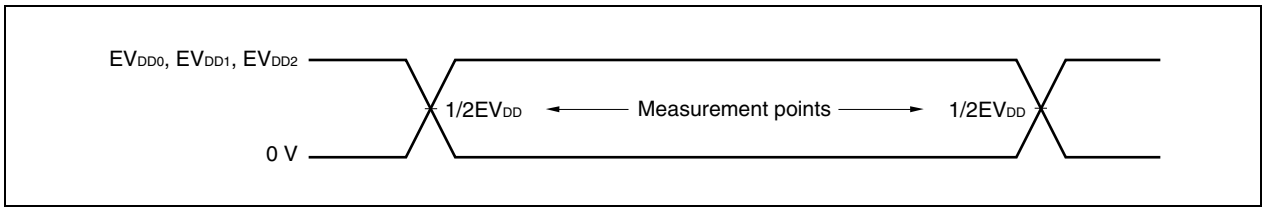
AC Test Input Measurement Points (External Bus (μPD70F3454GC-8EA-A Only), Pins Other than CSIB0 to CSIB2)



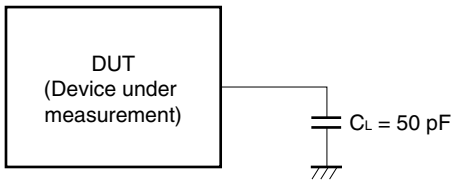
AC Test Output Measurement Points (External Bus (μPD70F3454GC-8EA-A Only), Pins Other than CSIB0 to CSIB2)



AC Test I/O Measurement Points (External Bus (μPD70F3454GC-8EA-A Only), CSIB0 to CSIB2 Pins)



Load Conditions

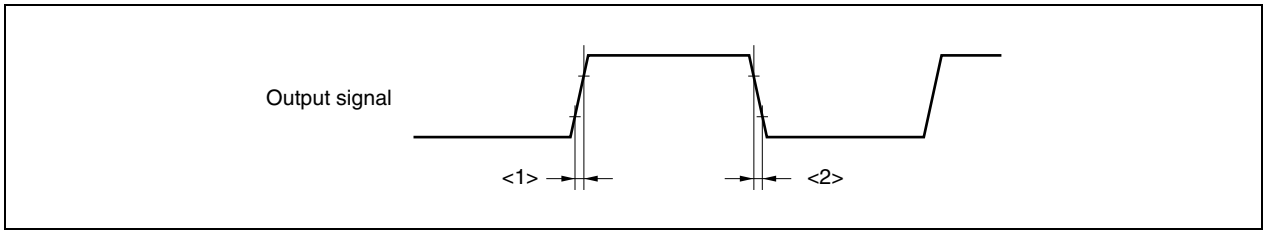


Caution If the load capacitance exceeds 50 pF due to the circuit configuration, bring the load capacitance of the device to 50 pF or less by inserting a buffer or by some other means.

(1) Output signal timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output rise time	t_{OR}	<1>	P07, PDL0 to PDL15, DDO		8	ns
			Other than above		15	ns
Output fall time	t_{OF}	<2>	P07, PDL0 to PDL15, DDO		8	ns
			Other than above		15	ns



(2) Reset, external interrupt timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

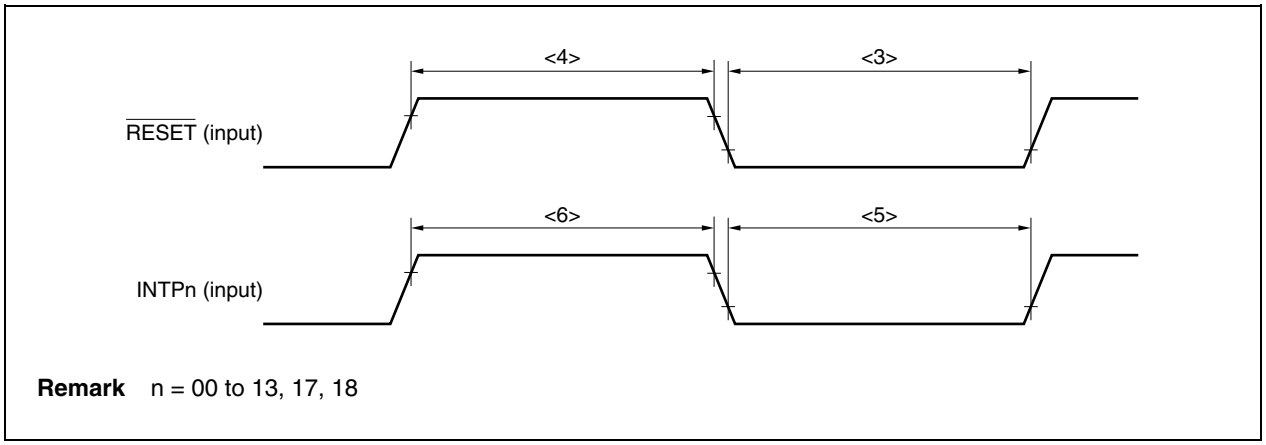
Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ low-level width	t_{WRSL}	<3>	Power is on, STOP mode is released	500 + T_{os}		ns
			Other than above	500		ns
$\overline{\text{RESET}}$ high-level width	t_{WRSH}	<4>		500		ns
INTPn low-level width	t_{WITL}	<5>	n = 00 to 13, 17, 18 (analog noise elimination)	500		ns
			n = 14 to 16 (digital noise elimination)	$4T_{smp}$		ns
INTPn high-level width	t_{WITH}	<6>	n = 00 to 13, 17, 18 (analog noise elimination)	500		ns
			n = 14 to 16 (digital noise elimination)	$4T_{smp}$		ns

Remarks 1. T_{os}: Oscillation stabilization time

T_{smp}: Noise elimination sampling clock cycle (set by INTNFCn register)

2. After reset release, a 1 ms oscillation stabilization time is internally secured when the oscillation frequency (f_X) = 8 MHz. The oscillation stabilization time is therefore (T_{os} + 1) ms. After STOP mode release, an oscillation stabilization time half the value set to the OSTS register is internally secured. Therefore, T_{os} = 0 ns is acceptable if sufficient stabilization time can be secured by the OSTS register setting.

Reset/Interrupt

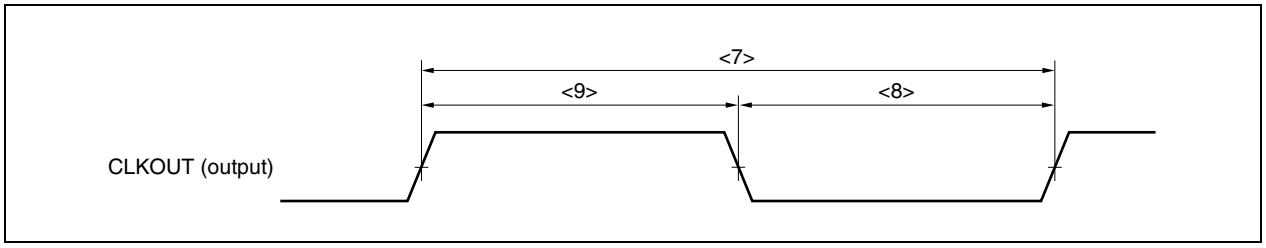


Remark n = 00 to 13, 17, 18

(3) CLKOUT output timing (μPD70F3454GC-8EA-A only)

(T_A = –40 to +85 °C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output cycle	t_{CYK}	<7>		31.25 ns	8 μs	
Low-level width	t_{WKH}	<8>		t_{CYK}/2 – 6.2		ns
High-level width	t_{WKL}	<9>		t_{CYK}/2 – 6.2		ns



(4) Bus Timing (μPD70F3454GC-8EA-A only)

(a) Read cycle (CLKOUT asynchronous)

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from address to ASTB↓	t_{DAST2}	<10>		(0.5 + w_{AS}) T – 20		ns
ASTB high-level width	t_{WSTH}	<11>		(1 + w_{AS} + i) T – 17		ns
Address hold time from ASTB↓	t_{HSTA}	<12>		(0.5 + w_{AH}) T – 19		ns
Address hold time from $\overline{RD}$↑	t_{HRDA2}	<13>		(1 + i) T – 29		ns
Delay time from address to $\overline{RD}$↓	t_{DARD2}	<14>		(1 + w_{AS} + w_{AH}) T – 36		ns
Delay time from $\overline{RD}$↓ to address float	t_{FRDA}	<15>			16	ns
Data input setup time from address	t_{DAID2}	<16>			(2 + w_D + w + w_{AS} + w_{AH}) T – 37	ns
Data input setup time from ASTB↓	t_{DSTID}	<17>			(1.5 + w_D + w + w_{AH}) T – 37	ns
Data input setup time from $\overline{RD}$↓	t_{DRDID2}	<18>			(1 + w_D + w) T – 37	ns
Delay time from ASTB↓ to $\overline{RD}$↓	t_{DSTRD3}	<19>		(0.5 + w_{AH}) T – 16		ns
Data input hold time (from $\overline{RD}$↑)	t_{HRDID2}	<20>		2		ns
Delay time from $\overline{RD}$↑ to bus output	t_{DRDOD2}	<21>		(1 + i) T – 19		ns
Delay time from $\overline{RD}$↑ to ASTB↑	t_{DRDST}	<22>		0.5T – 16		ns
$\overline{RD}$ low-level width	t_{WRDL2}	<23>		(1 + w_D + w) T – 20		ns
$\overline{RD}$ high-level width	t_{WRDH2}	<24>		(2 + i + w_{AS} + w_{AH}) T – 20		ns
High-level hold time from $\overline{RD}$↑ to $\overline{WRn}$	t_{HRDWR2}	<25>		(2 + i + w_{AS} + w_{AH}) T – 20		ns
$\overline{WAIT}$ setup time (to address)	t_{DAWT2}	<26>			(1.5 + w_D + w + w_{AS} + w_{AH}) T – 45	ns
$\overline{WAIT}$ hold time (from address)	t_{HAWT2}	<27>		(1.5 + w_D + w + w_{AS} + w_{AH}) T – 1		ns
$\overline{WAIT}$ setup time (to ASTB↓)	t_{DSTWT}	<28>			(1 + w_D + w + w_{AH}) T – 37	ns
$\overline{WAIT}$ hold time (from ASTB↓)	t_{HSTWT}	<29>		(1 + w_D + w + w_{AH}) T + 2		ns
$\overline{WAIT}$ setup time (to $\overline{RD}$↓)	t_{DRDWT2}	<30>			(0.5 + w_D + w) T – 37	ns
$\overline{WAIT}$ hold time (from $\overline{RD}$↓)	t_{HRDWT2}	<31>		(0.5 + w_D + w) T + 2		ns

Cautions 1. Set T in accordance with the following condition.

31.25 ns ≥ T

2. Be sure to insert the address setup waits and address hold waits.

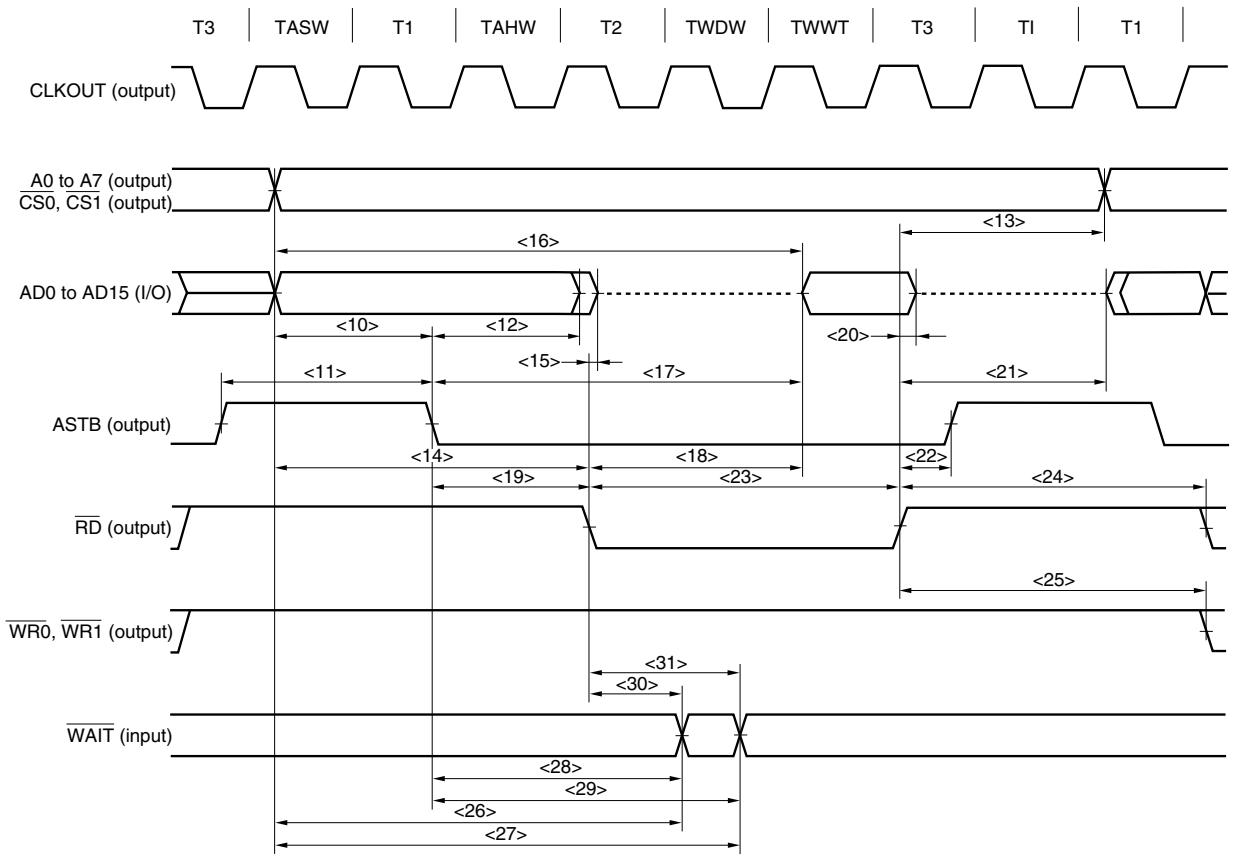
Remarks 1. w_{AS}: Number of address setup waits by the AWC register
w_{AH}: Number of address hold waits by the AWC register
w_D: Number of data waits by the DWC0 register
w: Number of external waits by the $\overline{WAIT}$ pin

2. T = 1/f_{CPU} (f_{CPU}: CPU clock frequency)

3. n = 0, 1

4. i: Number of idle states

Read cycle (CLKOUT asynchronous)



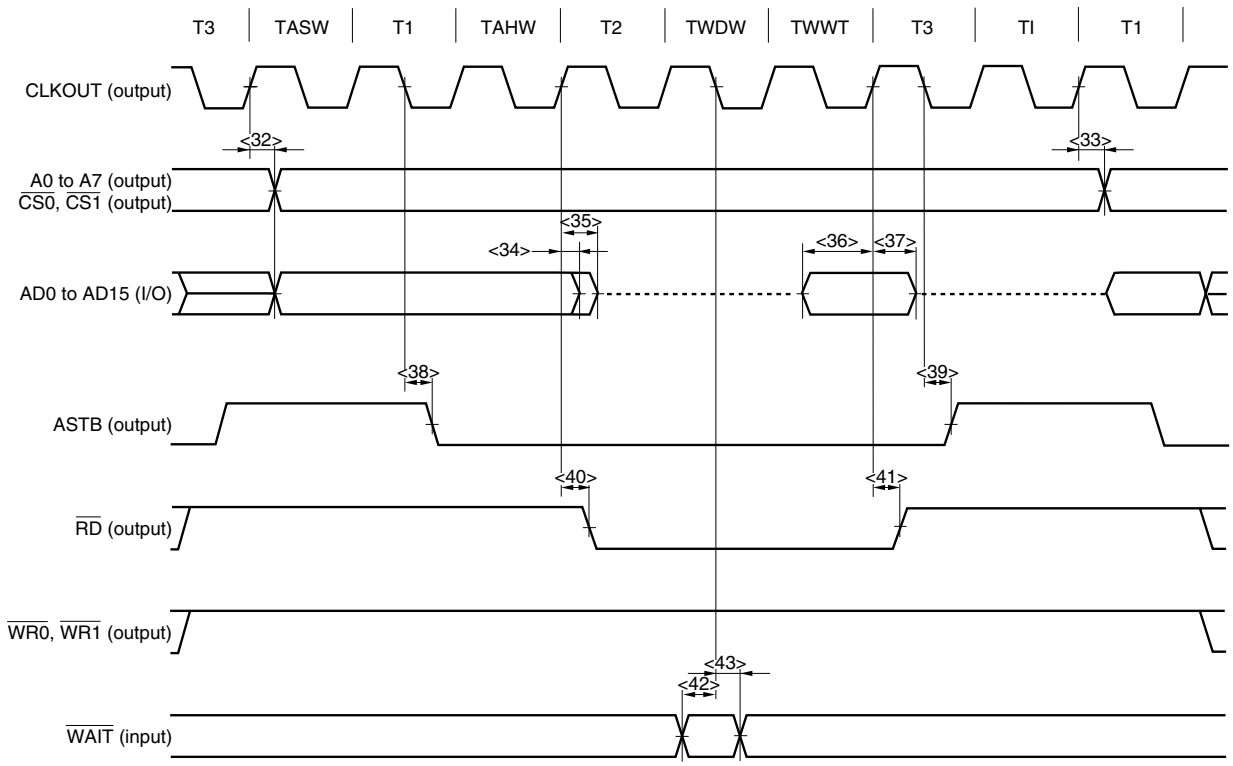
Remark The above timing chart shows the timing when the number of address setup waits is 1, number of address hold waits is 1, number of data waits is 1, number of waits by $\overline{\text{WAIT}}$ pin is 1 (when an active level (low level) is input for one cycle during the determined wait period), and number of idle states is 1.

(b) Read cycle (CLKOUT synchronous)

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address	t_{DKA2}	<32>			22	ns
Address hold time from CLKOUT↑	t_{HKA2}	<33>		–5		ns
Address hold time from CLKOUT↑	t_{HKA3}	<34>		–8		ns
Delay time from CLKOUT↑ to address float	t_{FKA}	<35>			15	ns
Data input setup time (to CLKOUT↑)	t_{SIDK2}	<36>		30		ns
Data input hold time (from CLKOUT↑)	t_{HKID2}	<37>		9		ns
Delay time from CLKOUT↓ to ASTB↓	t_{DKST3}	<38>		–8	18	ns
Delay time from CLKOUT↓ to ASTB↑	t_{DKST4}	<39>		–8	18	ns
Delay time from CLKOUT↑ to $\overline{RD}$↓	t_{DKRD3}	<40>		–10	17	ns
Delay time from CLKOUT↑ to $\overline{RD}$↑	t_{DKRD4}	<41>		–10	17	ns
$\overline{WAIT}$ setup time (to CLKOUT↓)	t_{SWTK2}	<42>		30		ns
$\overline{WAIT}$ hold time (from CLKOUT↓)	t_{HKWT2}	<43>		9		ns

Read cycle (CLKOUT synchronous)



Remark The above timing chart shows the timing when the number of address setup waits is 1, number of address hold waits is 1, number of data waits is 1, number of waits by $\overline{\text{WAIT}}$ pin is 1 (when an active level (low level) is input for one cycle during the determined wait period), and number of idle states is 1.

(c) Write cycle (CLKOUT asynchronous)

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from address to ASTB↓	t_{DAST2}	<10>		(0.5 + w_{AS}) T – 20		ns
ASTB high-level width	t_{WSTH}	<11>		(1 + w_{AS} + i) T – 16		ns
Address hold time from ASTB↓	t_{HSTA}	<12>		(0.5 + w_{AH}) T – 19		ns
Address hold time from $\overline{WRn}$↑	t_{HWRA2}	<44>		T – 19		ns
Delay time from address to $\overline{WRn}$↓	t_{DAWR2}	<45>		(1 + w_{AS} + w_{AH}) T – 36		ns
Delay time from $\overline{WRn}$↓ to data output	t_{DWROD3}	<46>			15	ns
Delay time from ASTB↓ to $\overline{WRn}$↓	t_{DSTWR3}	<47>		(0.5 + w_{AH}) T – 16		ns
Delay time from data output to $\overline{WRn}$↑	t_{DODWR2}	<48>		(1 + w_D + w) T – 25		ns
Data output hold time from $\overline{WRn}$↑	t_{HWROD2}	<49>		T – 19		ns
Delay time from $\overline{WRn}$↑ to ASTB↑	t_{DWRST}	<50>		0.5T – 16		ns
$\overline{WRn}$ low-level width	t_{WWRL2}	<51>		(1 + w_D + w) T – 20		ns
$\overline{WRn}$ high-level width	t_{WWRH2}	<52>		(2 + w_{AS} + w_{AH}) T – 20		ns
High-level hold time from $\overline{WRn}$↑ to $\overline{RD}$	t_{HWRRD2}	<53>		(2 + w_{AS} + w_{AH}) T – 20		ns
$\overline{WAIT}$ setup time (to address)	t_{DAWT2}	<26>			(1.5 + w_D + w + w_{AS} + w_{AH}) T – 45	ns
$\overline{WAIT}$ hold time (from address)	t_{HAWT2}	<27>		(1.5 + w_D + w + w_{AS} + w_{AH}) T – 1		ns
$\overline{WAIT}$ setup time (to ASTB↓)	t_{DSTWT}	<28>			(1 + w_D + w + w_{AH}) T – 37	ns
$\overline{WAIT}$ hold time (from ASTB↓)	t_{HSTWT}	<29>		(1 + w_D + w + w_{AH}) T + 2		ns
$\overline{WAIT}$ setup time (to $\overline{WRn}$↓)	t_{DWRWT2}	<54>			(0.5 + w_D + w) T – 37	ns
$\overline{WAIT}$ hold time (from $\overline{WRn}$↓)	t_{HWRWT2}	<55>		(0.5 + w_D + w) T + 2		ns

Cautions 1. Set T in accordance with the following condition.

31.25 ns ≥ T

2. Be sure to insert the address setup waits and address hold waits.

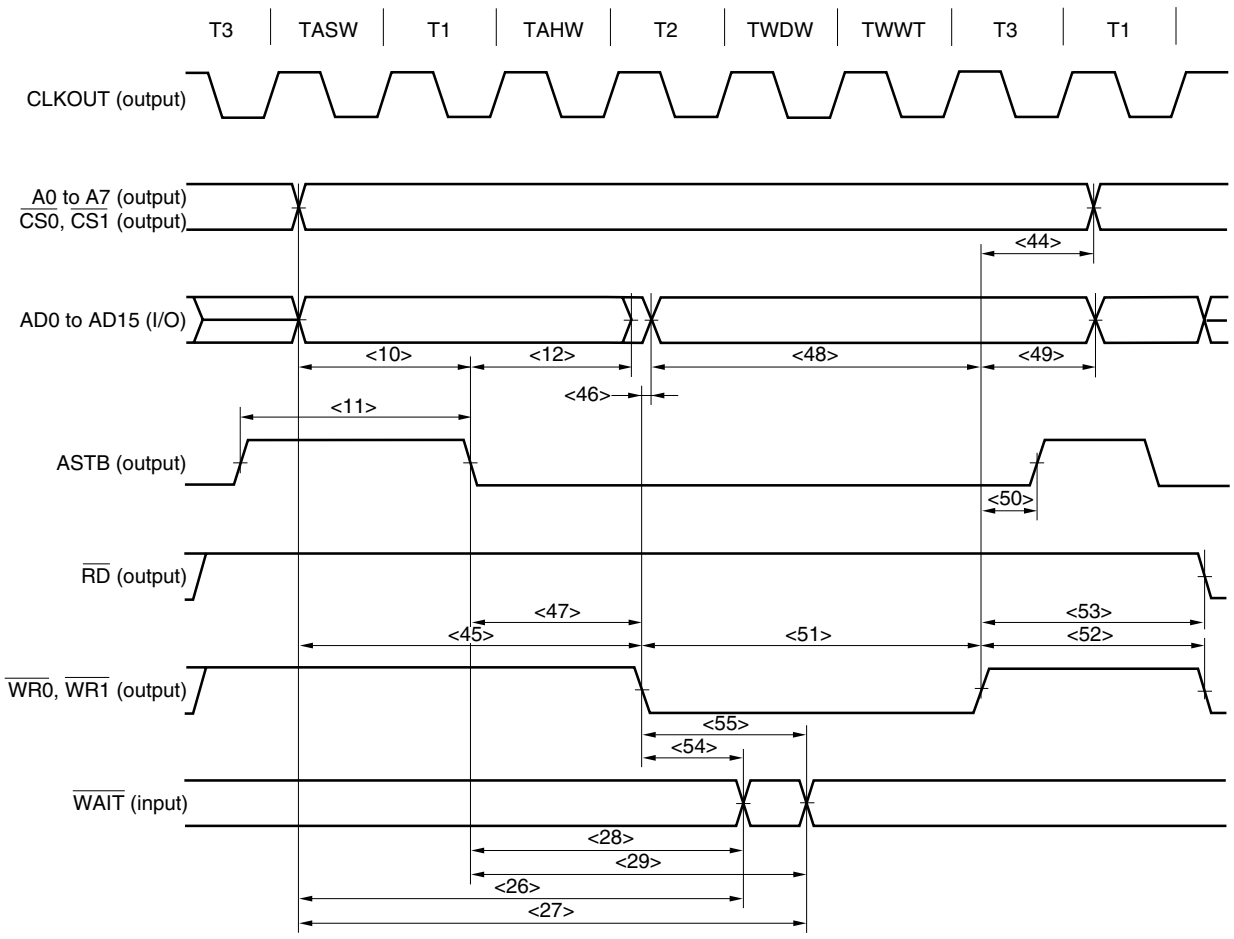
Remarks 1. w_{AS}: Number of address setup waits by the AWC register
w_{AH}: Number of address hold waits by the AWC register
w_D: Number of data waits by the DWC0 register
w: Number of external waits by the $\overline{WAIT}$ pin

2. T = 1/f_{CPU} (f_{CPU}: CPU operating clock frequency)

3. n = 0, 1

4. i: Number of idle states

Write cycle (CLKOUT asynchronous)



Remark The above timing chart shows the timing when the number of address setup waits is 1, number of address hold waits is 1, number of data waits is 1, and number of waits by $\overline{\text{WAIT}}$ pin is 1 (when an active level (low level) is input for one cycle during the determined wait period).

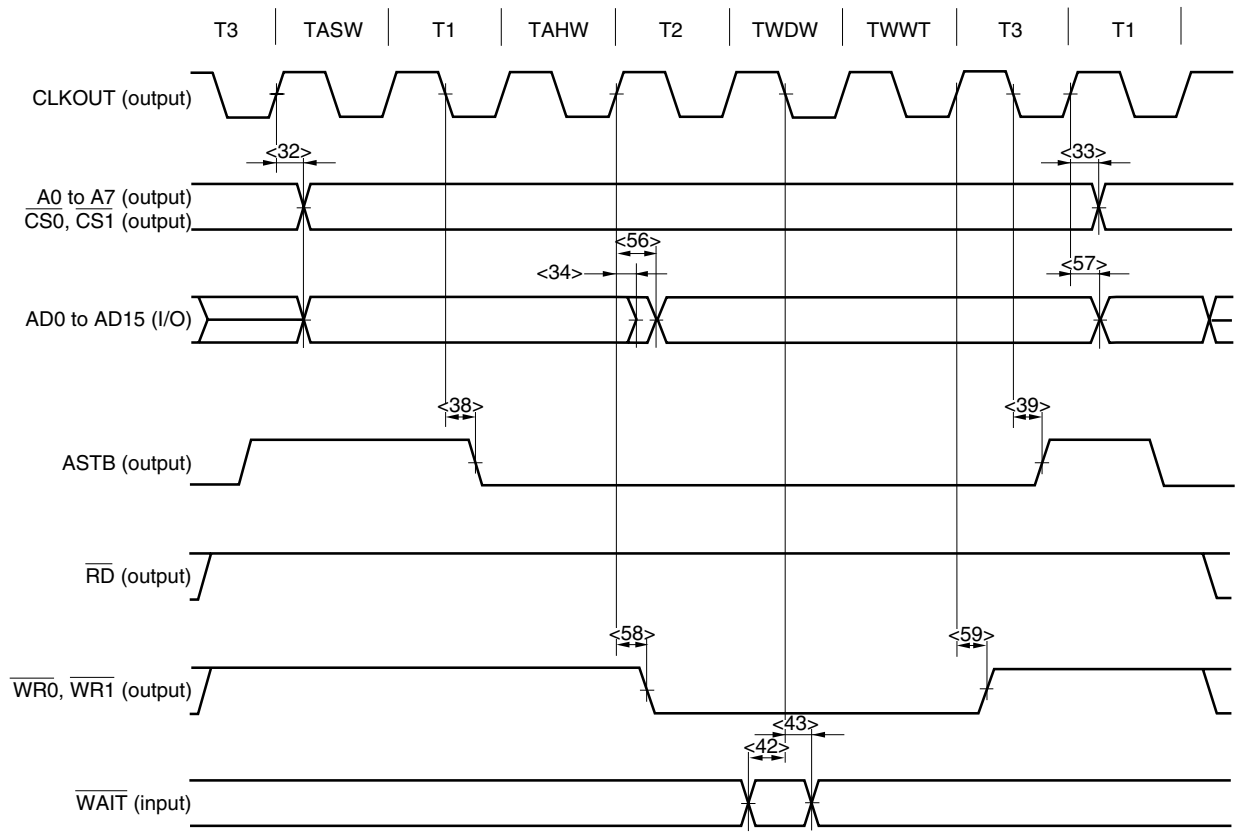
(d) Write cycle (CLKOUT synchronous)

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address	t_{DKA2}	<32>			22	ns
Address hold time from CLKOUT↑	t_{HKA2}	<33>		–5		ns
Address hold time from CLKOUT↑	t_{HKA3}	<34>		–8		ns
Delay time from CLKOUT↓ to ASTB↓	t_{DKST3}	<38>		–8	18	ns
Delay time from CLKOUT↓ to ASTB↑	t_{DKST4}	<39>		–8	18	ns
Delay time from CLKOUT↑ to data output	t_{DKOD3}	<56>			22	ns
Data output hold time from CLKOUT↑	t_{HKOD2}	<57>		–9		ns
Delay time from CLKOUT↑ to $\overline{WRn}$↓	t_{DKWR3}	<58>		–10	17	ns
Delay time from CLKOUT↑ to $\overline{WRn}$↑	t_{DKWR4}	<59>		–10	17	ns
$\overline{WAIT}$ setup time (to CLKOUT↓)	t_{SWTK2}	<42>		30		ns
$\overline{WAIT}$ hold time (from CLKOUT↓)	t_{HKWT2}	<43>		9		ns

Remark n = 0, 1

Write cycle (CLKOUT synchronous)



Remark The above timing chart shows the timing when the number of address setup waits is 1, number of address hold waits is 1, number of data waits is 1, and number of waits by $\overline{\text{WAIT}}$ pin is 1 (when an active level (low level) is input for one cycle during the determined wait period).

(5) Timer Timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
TIn high-level width[Notes 1, 2]	t_{WTIHn}	<60>	n = B00 to B03, B10 to B13	12T + 10		ns
			n = A20, A21, A30, A31, A40, A41	$3T_{smp1}$ + 10		ns
TIn low-level width[Notes 1, 2]	t_{WTILn}	<61>	n = B00 to B03, B10 to B13	12T + 10		ns
			n = A20, A21, A30, A31, A40, A41	$3T_{smp1}$ + 10		ns
EVTBm high-level width[Note 1]	t_{WEVBHm}	<62>	m = 0, 1	12T + 10		ns
EVTBm low-level width[Note 1]	t_{WEVBLm}	<63>	m = 0, 1	12T + 10		ns
TRGBm high-level width[Note 1]	t_{WTRHm}	<64>	m = 0, 1	12T + 10		ns
TRGBm low-level width[Note 1]	t_{WTRLm}	<65>	m = 0, 1	12T + 10		ns
TENCm0/TENCm1 high-level width[Note 3]	t_{WENCHm}	<66>	m = 0, 1	$3T_{smp2}$ + 10		ns
TENCm0/TENCm1 low-level width[Note 3]	t_{WENCLm}	<67>	m = 0, 1	$3T_{smp2}$ + 10		ns
TECRm high-level width[Note 3]	t_{WCRHm}	<68>	m = 0, 1	$3T_{smp2}$ + 10		ns
TECRm low-level width[Note 3]	t_{WCRLm}	<69>	m = 0, 1	$3T_{smp2}$ + 10		ns
TITm0/TITm1 high-level width[Note 3]	t_{WTITHm}	<70>	m = 0, 1	$3T_{smp2}$ + 10		ns
TITm0/TITm1 low-level width[Note 3]	t_{WTITLm}	<71>	m = 0, 1	$3T_{smp2}$ + 10		ns
EVTTm high-level width[Note 3]	t_{WEVTHm}	<72>	m = 0, 1	$3T_{smp2}$ + 10		ns
EVTTm low-level width[Note 3]	t_{WEVTLm}	<73>	m = 0, 1	$3T_{smp2}$ + 10		ns
TENCm0/TENCm1 input time differential[Note 3]	t_{PHUDm}	<74>	m = 0, 1	$3T_{smp2}$ + 10		ns

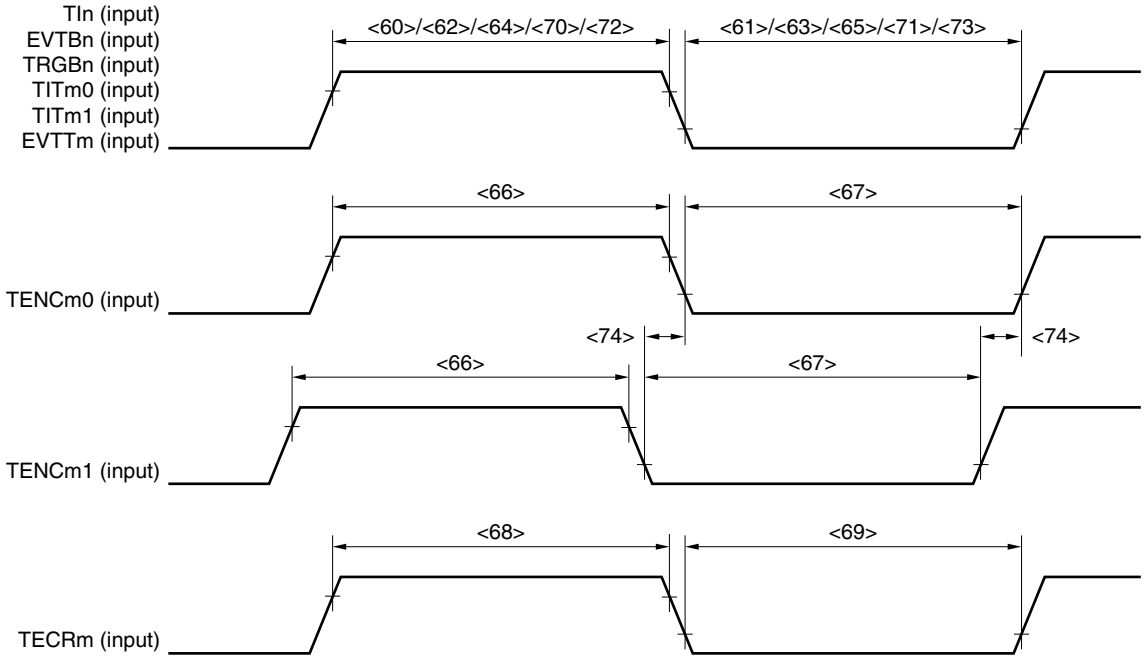
Notes 1. T = 1/fxx

2. T_{smp1}: Noise elimination sampling clock cycle (set by TANFC2 to TANFC4 registers)

3. T_{smp2}: Noise elimination sampling clock cycle (set by TTNFC0 and TTNFC1 registers)

Remark The above specification shows a pulse width that is accurately detected as a valid edge. Even if a pulse narrower than the above specification is input, therefore, it may be detected as a valid edge.

Timer Input Timing



Remark n = A20, A21, A30, A31, A40, A41, B00 to B03, B10 to B13
m = 0, 1

(6) CSIB Timing

(a) Master mode

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKBn}$ cycle	t_{KCYM}	<75>		125		ns
$\overline{SCKBn}$ high-/low-level width	t_{KWHM}, t_{KWLM}	<76>		t_{KCYM}/2 – 10		ns
SIBn setup time (to $\overline{SCKBn}$↑)	t_{SSIM}	<77>		30		ns
SIBn setup time (to $\overline{SCKBn}$↓)				30		ns
SIBn hold time (from $\overline{SCKBn}$↑)	t_{HSIM}	<78>		30		ns
SIBn hold time (from $\overline{SCKBn}$↓)				30		ns
SOBn output delay time (from $\overline{SCKBn}$↓)	t_{DSOM}	<79>			30	ns
SOBn output delay time (from $\overline{SCKBn}$↑)					30	ns
SOBn output hold time (from $\overline{SCKBn}$↑)	t_{HSOM}	<80>		t_{KCYM}/2 – 10		ns
SOBn output hold time (from $\overline{SCKBn}$↓)				t_{KCYM}/2 – 10		ns

Remark n = 0 to 2

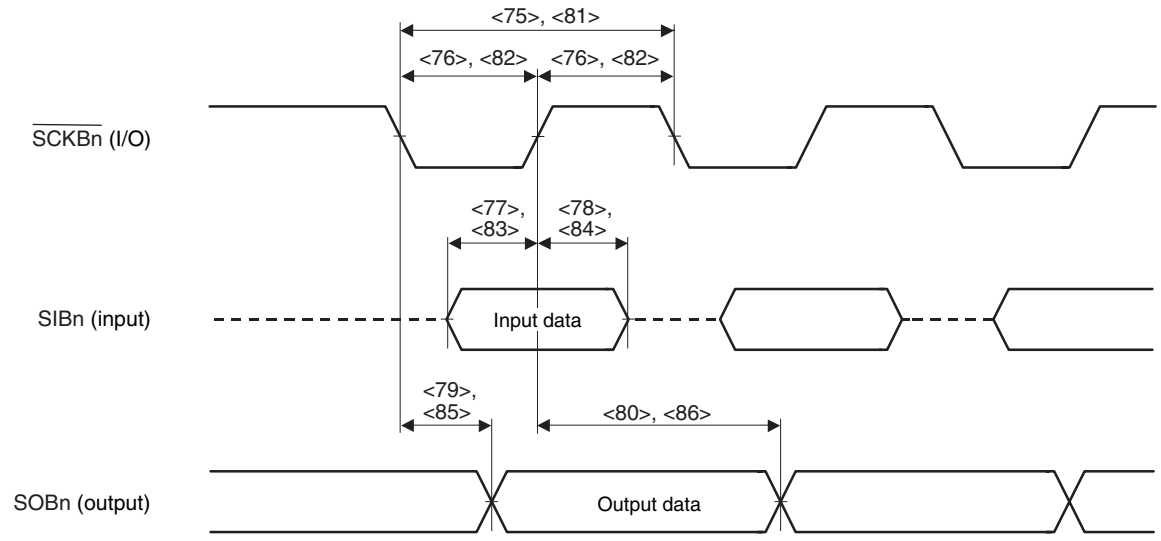
(b) Slave mode

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKBn}$ cycle	t_{KCYS}	<81>		125		ns
$\overline{SCKBn}$ high-/low-level width	t_{KWHS}, t_{KWLS}	<82>		t_{KCYS}/2 – 10		ns
SIBn setup time (to $\overline{SCKBn}$↑)	t_{SSIS}	<83>		30		ns
SIBn setup time (to $\overline{SCKBn}$↓)				30		ns
SIBn hold time (from $\overline{SCKBn}$↑)	t_{HSIS}	<84>		30		ns
SIBn hold time (from $\overline{SCKBn}$↓)				30		ns
SOBn output delay time (from $\overline{SCKBn}$↓)	t_{DSOS}	<85>			30	ns
SOBn output delay time (from $\overline{SCKBn}$↑)					30	ns
SOBn output hold time (from $\overline{SCKBn}$↑)	t_{HSOS}	<86>		t_{KCYS}/2 – 10		ns
SOBn output hold time (from $\overline{SCKBn}$↓)				t_{KCYS}/2 – 10		ns

Remark n = 0 to 2

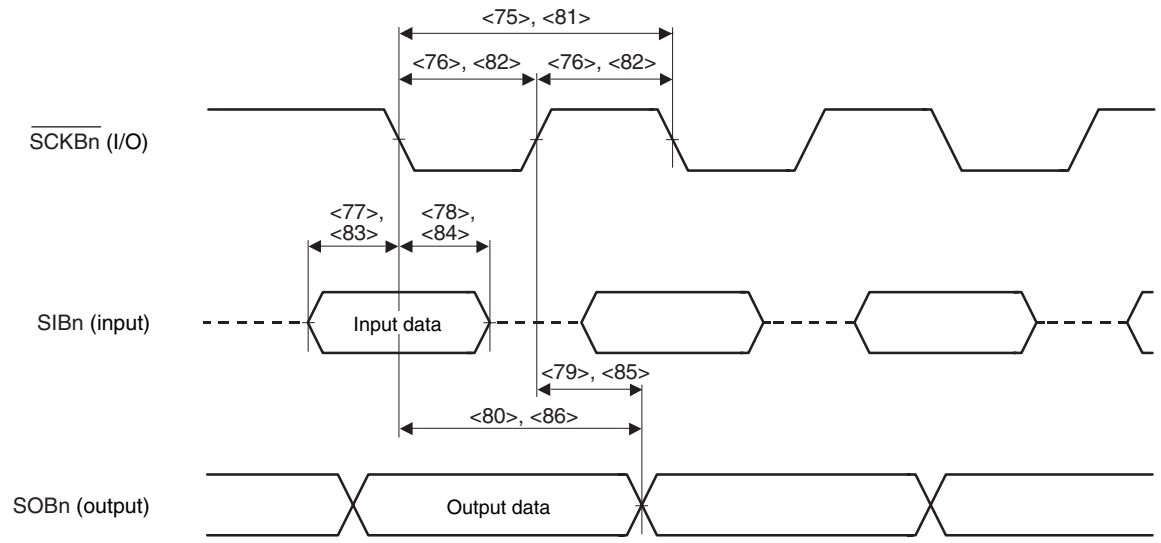
CSIB timing when CBnCKP and CBnDAP bits of CBnCTL1 register = 00



Remarks 1. Broken lines indicate high impedance.

2. n = 0 to 2

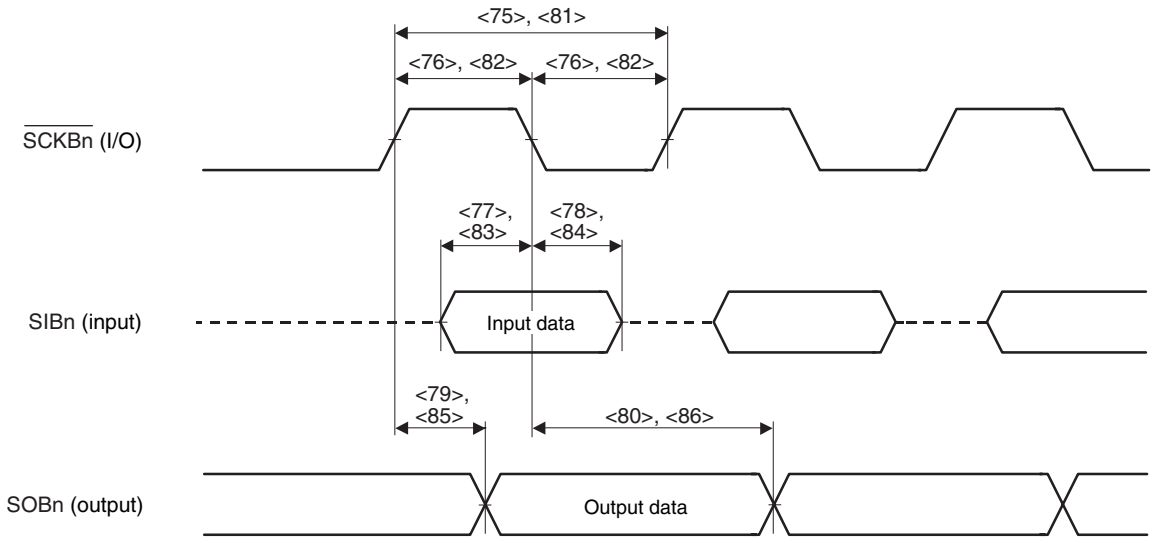
CSIB timing when CBnCKP and CBnDAP bits of CBnCTL1 register = 01



Remarks 1. Broken lines indicate high impedance.

2. n = 0 to 2

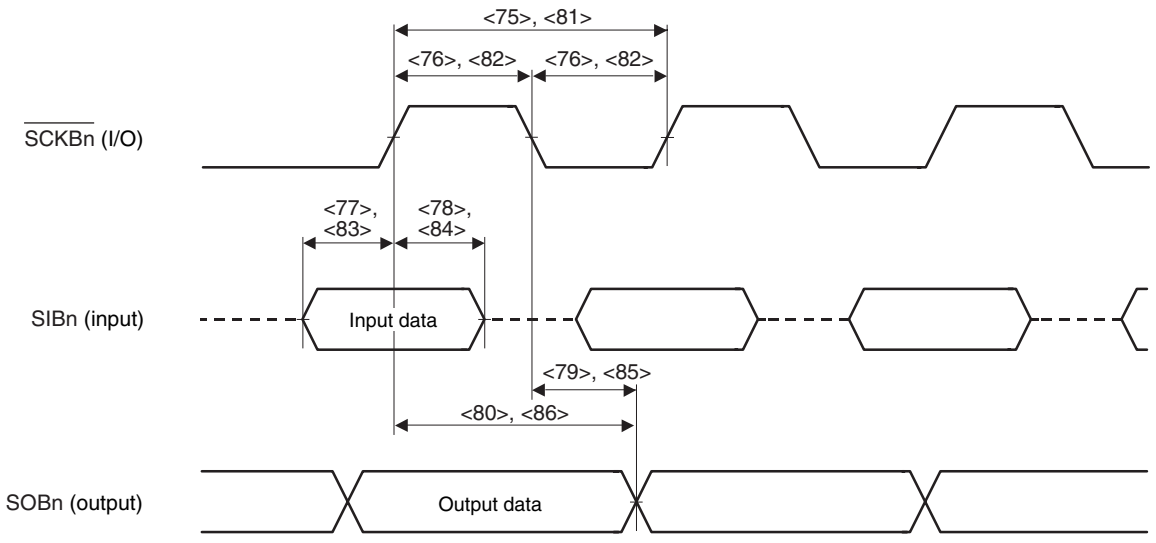
CSIB timing when CBnCKP and CBnDAP bits of CBnCTL1 register = 10



Remarks 1. Broken lines indicate high impedance.

2. n = 0 to 2

CSIB timing when CBnCKP and CBnDAP bits of CBnCTL1 register = 11



Remarks 1. Broken lines indicate high impedance.

2. n = 0 to 2

(7) I^2C bus timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter		Symbol		Standard Mode		High-Speed Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCL clock frequency		f_{CLK}	–	0	100	0	400	kHz
Bus free time (between stop condition and start condition)		t_{BUF}	<87>	4.7	–	1.3	–	μs
Hold time[Note 1]		$t_{HD:STA}$	<88>	4.0	–	0.6	–	μs
SCL clock low-level width		t_{LOW}	<89>	4.7	–	1.3	–	μs
SCL clock high-level width		t_{HIGH}	<90>	4.0	–	0.6	–	μs
Start/restart condition setup time		$t_{SU:STA}$	<91>	4.7	–	0.6	–	μs
Data hold time	CBUS-compatible master	$t_{HD:DAT}$	<92>	5.0	–	–	–	μs
	I^2C mode			0[Note 2]	–	0[Note 2]	0.9[Note 3]	μs
Data setup time		$t_{SU:DAT}$	<93>	250	–	100[Note 4]	–	ns
SDA, SCL signal rise time		t_R	<94>	–	1000	20 + 0.1Cb[Note 5]	300	ns
SDA, SCL signal fall time		t_F	<95>	–	300	20 + 0.1Cb[Note 5]	300	ns
Stop condition setup time		$t_{SU:STO}$	<96>	4.0	–	0.6	–	μs
Pulse width of spike suppressed by input filter		t_{SP}	<97>	–	–	0	50	ns
Each bus line capacitive load		Cb	–	–	400	–	400	pF

Notes 1. The first clock pulse is generated after a hold time during the start condition.

2. The system must internally supply a hold time of at least 300 ns for the SDA signal (at $V_{IHmin.}$ of SCL signal) to fill the undefined area at the falling edge of SCL.

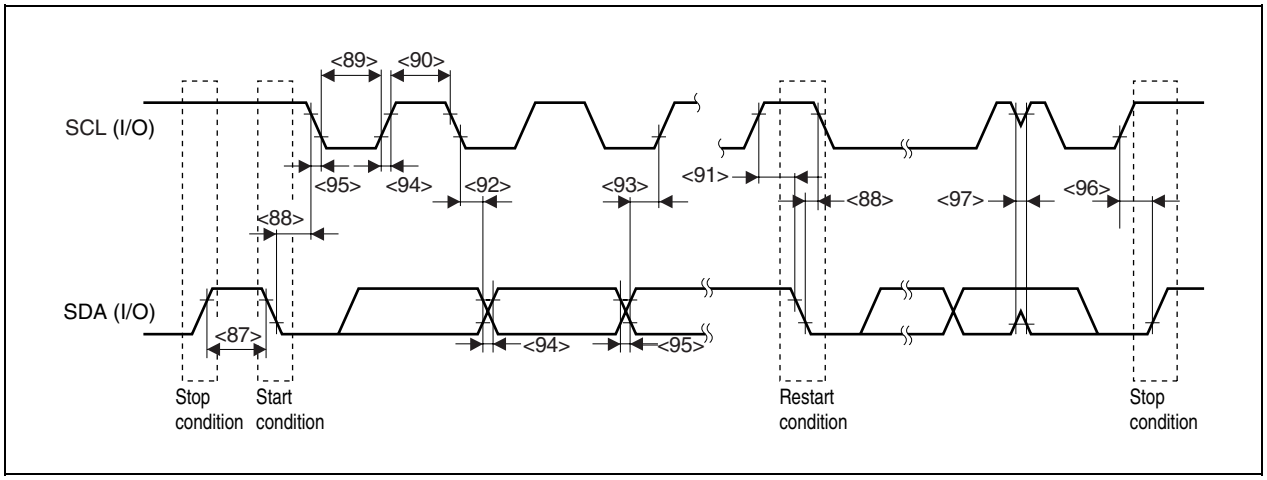
3. If the system does not extend the low hold time (t_{LOW}) of the SCL signal, the maximum data hold time ($t_{HD:DAT}$) must be satisfied.

4. The high-speed mode I^2C bus can be used in the standard mode I^2C bus system. In this case, make sure that the following conditions are satisfied.

- If system does not extend the low status hold time of the SCL signal
  $t_{SU:DAT} \geq 250$ ns
- If system extends the low status hold time of SCL signal
  Sends the next data bit to the SDA line before the SCL line is released ($t_{Rmax.} + t_{SU:DAT}$ = 1000 + 250 = 1250 ns: standard mode I^2C bus specification).

5. Cb: Total capacitance of one bus line (unit: pF)

I²C bus timing



(8) High-impedance control timing

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Oscillation stop → timer output high impedance	t_{CLM}	When clock monitor is operating		65	μs
Input to TOBnOFF → timer output high impedance	t_{HTQn}			300	ns
Input to TOAmOFF → timer output high impedance	t_{HTPm}			300	ns
Input to ANI00/ANI05 → timer output high impedance	t_{ANI0}			10	μs
Input to ANI10/ANI15 to ANI12/ANI17 → timer output high impedance	t_{ANI1}			10	μs

Remark n = 0, 1
m = 2, 3

28.2.9 Characteristics of A/D converters 0, 1

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = AV_{REFP0} = AV_{REFP1} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			12	12	12	bit
Overall error[Note 1]					±10	LSB
Conversion time	t_{CONV}	f_{AD01} = 16 MHz, ADAnCTC register = 0BH or 0CH	2.0			μs
		f_{AD01} = 12 MHz, ADAnCTC register = 00H			7.42	μs
Zero scale error[Note 1]					±10	LSB
Full-scale error[Note 1]					±10	LSB
Integral linearity error[Note 1]					±4	LSB
Differential linearity error[Note 1]					±2.5	LSB
Analog reference voltage	AV_{DD}		4.0		5.5	V
Analog input voltage	V_{IAN}		AV_{SS}		AV_{DD}	V
AV_{DD} supply current[Note 2]	AI_{DD}	During operation		4.5	7.5	mA
	AI_{DDS}	In STOP mode[Note 3]		3.5	17.5	μA

Notes 1. Excludes quantization error (±0.5 LSB).

2. This value is for only one A/D converter (A/D converter 0 or 1).

3. Stop the operation of A/D converters 0 and 1 (ADnSCM.ADnCE bit = 0) before setting STOP mode.

Remarks 1. LSB: Least Significant Bit

2. f_{AD01}: Base clock of A/D converters 0 and 1

3. n = 0, 1

28.2.10 Characteristics of A/D converter 2

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution			10	10	10	bit
Overall error[Note 1]					±4.0	LSB
Conversion time	t_{CONV}		3.88		10	μs
Zero scale error[Note 1]					±4.0	LSB
Full-scale error[Note 1]					±4.0	LSB
Integral linearity error[Note 1]					±4.0	LSB
Differential linearity error[Note 1]					±2.0	LSB
Analog reference voltage	AV_{DD}		4.0		5.5	V
Analog input voltage	V_{IAN}		AV_{SS}		AV_{DD}	V
AV_{DD} supply current	AI_{DD}	During operation		3.5	7	mA
	AI_{DDS}	In STOP mode[Note 2]		1	10	μA

Notes 1. Excludes quantization error (±0.5 LSB).

2. Stop the operation of A/D converter 2 (AD2M0.AD2CE bit = 0) before setting STOP mode.

Remark LSB: Least Significant Bit

28.2.11 Operational amplifier characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IO}				±9.0		mV
Input voltage range	V_I	Gain = 2.500		0.04AV_{DD}		0.36AV_{DD}	V
		Gain = 5.000		0.02AV_{DD}		0.18AV_{DD}	V
		Gain = 10.00		0.01AV_{DD}		0.085AV_{DD}	V
Slew rate[Note 1]	S_R			10	15		V/μs
Gain error		**Note 2**	Gain = 2.500 to 4.444		±1.0	±1.3	%
			Gain = 5.000 to 6.667		±1.0	±1.5	%
			Gain = 8.000, 10.00		±1.0	±1.7	%
		Note 3	Gain = 2.500 to 4.444		±1.0	±2.0	%
			Gain = 5.000 to 6.667		±1.0	±2.1	%
			Gain = 8.000, 10.00		±1.0	±2.2	%
Operating current[Note 4]	I_{OPDD}	During operation			1.8	2.6	mA
	A_{IDDS}	In STOP mode[Note 5]			1.0	10	μA

Notes 1. Inclination characteristic of 10% to 90% of output voltage

2. AV_{DD0} = AV_{DD1} = 4.5 to 5.5 V

3. AV_{DD0} = AV_{DD1} = 4.0 to 5.5 V

4. Four operational amplifiers are provided in total. The value shows the operating current per operational amplifier.

5. Stop operational amplifier operation (OP0CTL0.OP0EN bit = 0, OP1CTL0.OP12EN, OP11EN, and OP10EN bits = 0)) before setting STOP mode.

Remark Power supplies AV_{DD0} and AV_{DD1} are used for the operational amplifier.

28.2.12 Comparator characteristics

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 4.0 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IO}			±3.0		mV
Input voltage range	V_I		AV_{SS}		AV_{DD}	V
Comparator reference voltage (full range)	CREFnF		$0.02AV_{DD}$ + 0.1		$0.92AV_{DD}$ – 0.1	V
Comparator reference voltage (low range)	CREFnL		$0.02AV_{DD}$ + 0.1		$0.5AV_{DD}$ – 0.1	V
Response time	t_{CR}	Input amplitude = 100 mV, at rising edge[Note 1]		1.0		μs
	t_{CF}	Input amplitude = 100 mV, at falling edge[Note 2]		1.0		μs
Operating current[Note 3]	I_{CPDD}	During operation			250	μA
	A_{IDDS}	In STOP mode[Note 4]		2.0	20	nA

Notes 1. Characteristics of pulse response when ANIm input changes from the comparator reference voltage – 100 mV to the comparator reference voltage + 100 mV

2. Characteristics of pulse response when ANIm input changes from the comparator reference voltage + 100 mV to the comparator reference voltage – 100 mV

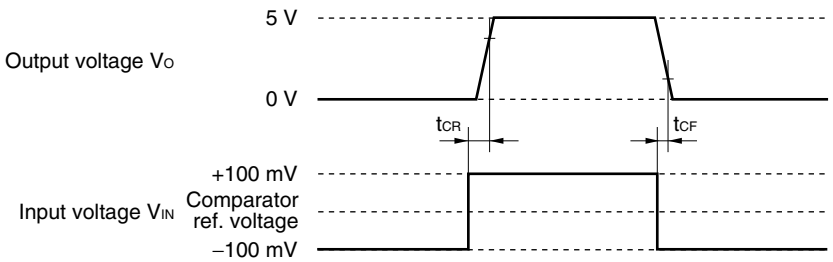
3. Four comparators are provided in total. The value shows the operating current per comparator.

4. Stop comparator operation (CMPnCTL0 register = 00H) before setting STOP mode.

Remarks 1. Power supplies for the comparators are AV_{DD0} and AV_{DD1}.

2. m = 05, 15 to 17
n = 0, 1

Comparator Characteristics



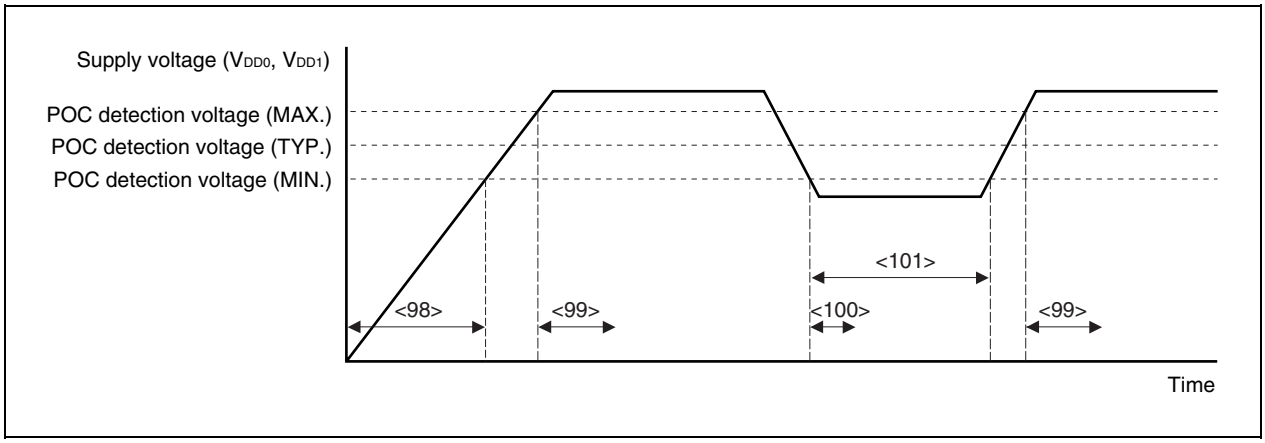
28.2.13 Power-on-clear circuit (POC)

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	TYP.	MAX.	Unit
POC detection voltage	V_{POC0}			3.5	3.7	3.9	V
Supply voltage rise time	t_{PTH}	<98>	V_{DD0}, V_{DD1} = 0 to 3.5 V	2.5 μs		1.8 s	
Response time 1[Note 1]	t_{PTHD}	<99>	After V_{DD0} and V_{DD1} reach 3.9 V on power application			3.0	ms
Response time 2[Note 2]	t_{PD}	<100>	After V_{DD0} and V_{DD1} drop to 3.5 V on power off			1.0	ms
Minimum width of V_{DD0}, V_{DD1}	t_{PW}	<101>		0.2			ms

Notes 1. The time required to release a reset signal (POCRES) after the POC detection voltage is detected.

2. The time required to output a reset signal (POCRES) after the POC detection voltage is detected.

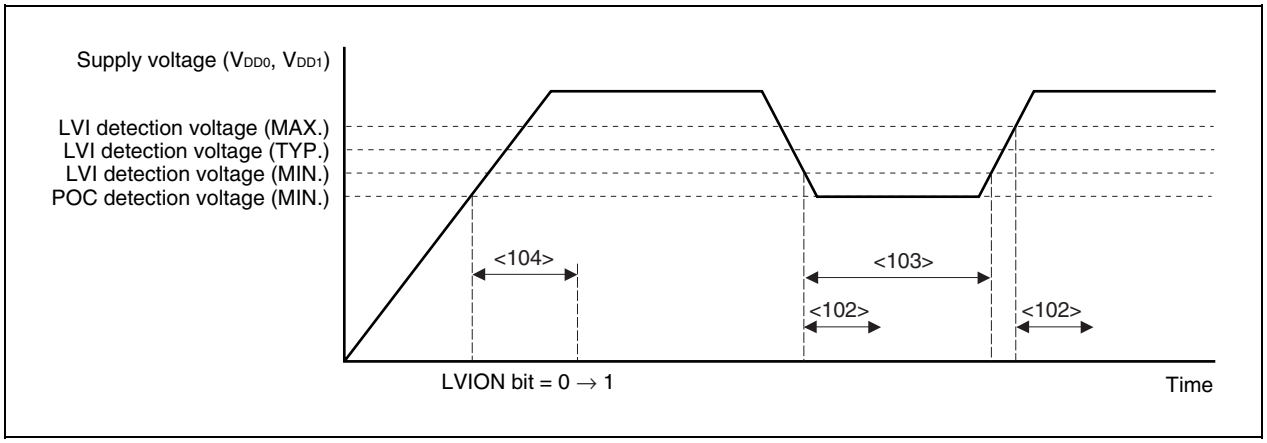


28.2.14 Low-voltage detector (LVI)

(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	TYP.	MAX.	Unit
LVI detection voltage	V_{LVI0}		LVIS.LVIS0 bit = 0	4.2	4.4	4.6	V
	V_{LVI1}		LVIS.LVIS0 bit = 1	4.0	4.2	4.4	V
Response time 1[Note]	t_{LD}	<102>	After V_{DD0} and V_{DD1} reach V_{LVI0}/V_{LVI1} (MAX.) or drop to V_{LVI0}/V_{LVI1} (MIN.)		0.2	2.0	ms
Minimum width of V_{DD0}, V_{DD1}	t_{LW}	<103>		0.2			ms
Reference voltage stabilization wait time	t_{LWAIT}	<104>	After V_{DD0} and V_{DD1} reach POC detection voltage (MIN.) and the LVIM.LVION bit is changed from 0 to 1		0.1		ms

Note The time required to output an interrupt request signal (INTLVIL, INTLVIH) or internal reset signal (LVIRES) after the LVI detection voltage is detected.



28.2.15 Flash memory programming characteristics

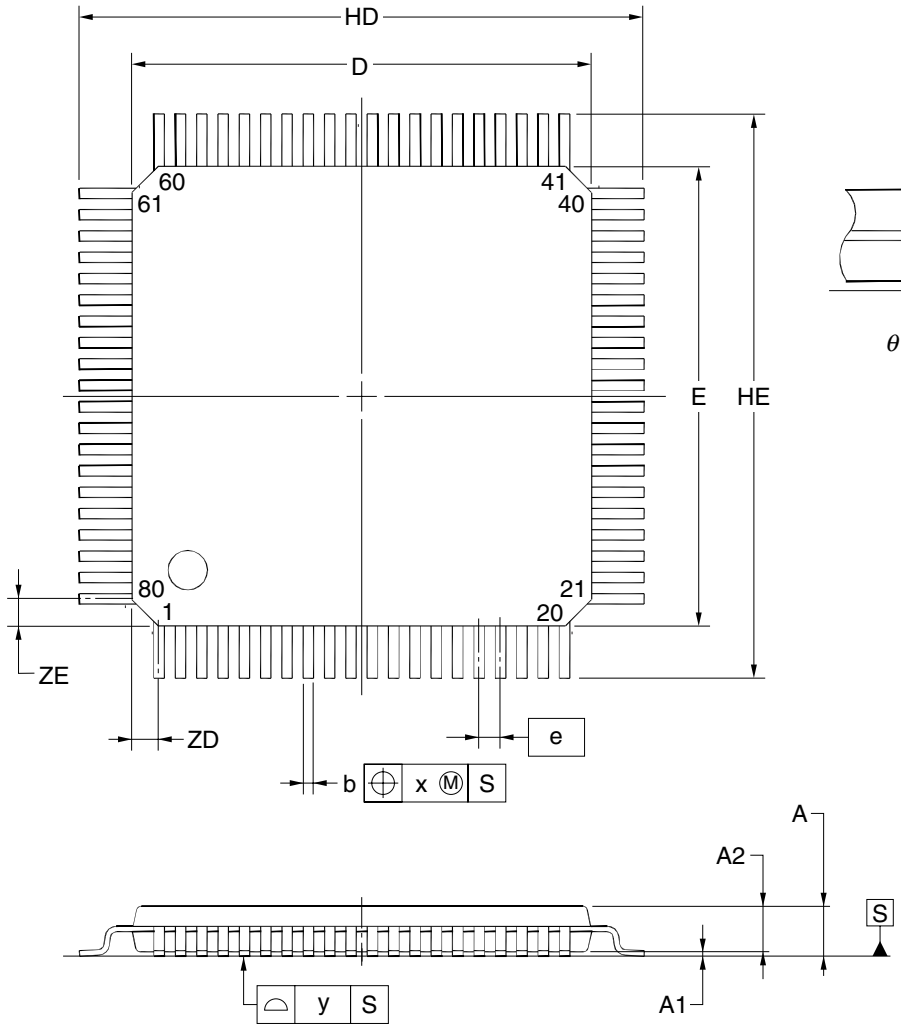
(T_A = –40 to +85°C, V_{DD0} = V_{DD1} = EV_{DD0} = EV_{DD1} = EV_{DD2} = AV_{DD0} = AV_{DD1} = AV_{DD2} = 3.5 to 5.5 V, V_{SS0} = V_{SS1} = EV_{SS0} = EV_{SS1} = EV_{SS2} = AV_{SS0} = AV_{SS1} = AV_{SS2} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Rewrite count	C_{ERWR}	**Note**		100		Times

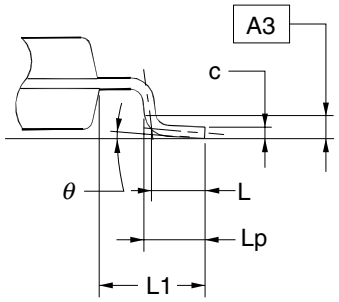
Note Rewrite as follows.

Example when three rewrites: Shipped product →E→P→E→P→E→P (P: Write, E: Erase)

80-PIN PLASTIC LQFP(14x14)



detail of lead end



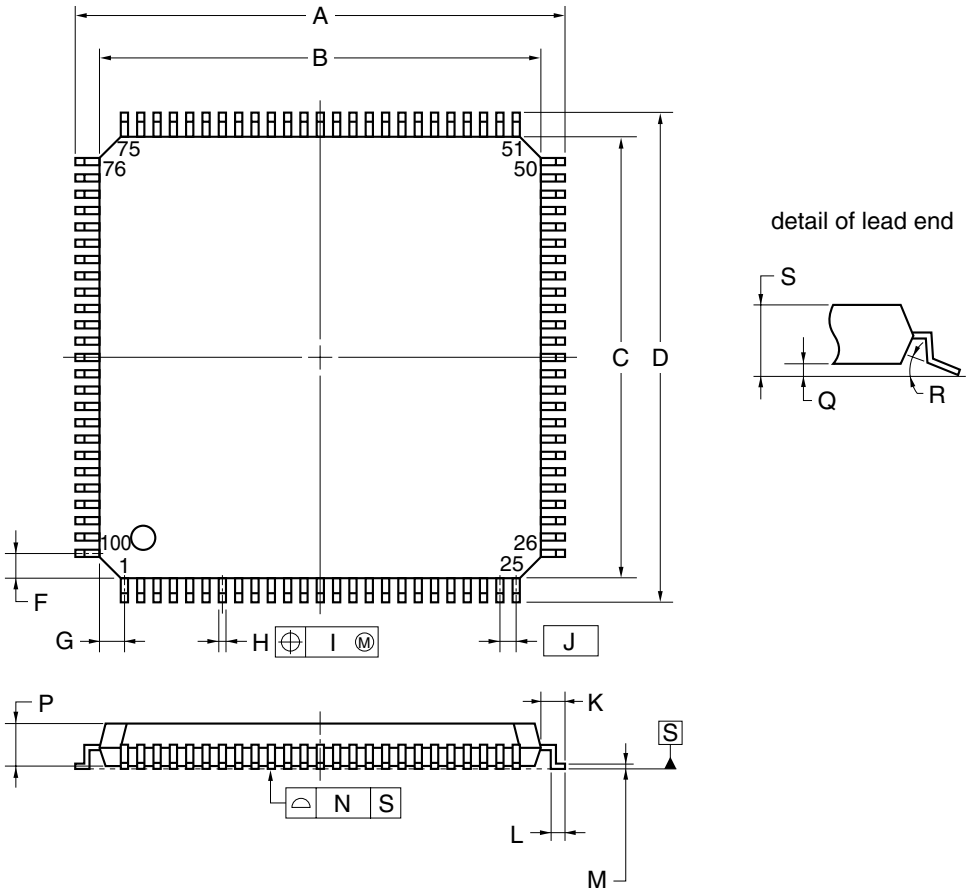
(UNIT:mm)

ITEM	DIMENSIONS
D	14.00±0.20
E	14.00±0.20
HD	17.20±0.20
HE	17.20±0.20
A	1.70 MAX.
A1	0.125±0.075
A2	1.40±0.05
A3	0.25
b	0.32±0.06
c	$0.17^{+0.03}_{-0.06}$
L	0.80
Lp	0.886±0.15
L1	1.60±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.65
x	0.13
y	0.10
ZD	0.825
ZE	0.825

P80GC-65-UBT

NOTE
Each lead centerline is located within 0.13 mm of its true position at maximum material condition.

100-PIN PLASTIC LQFP (FINE PITCH) (14x14)



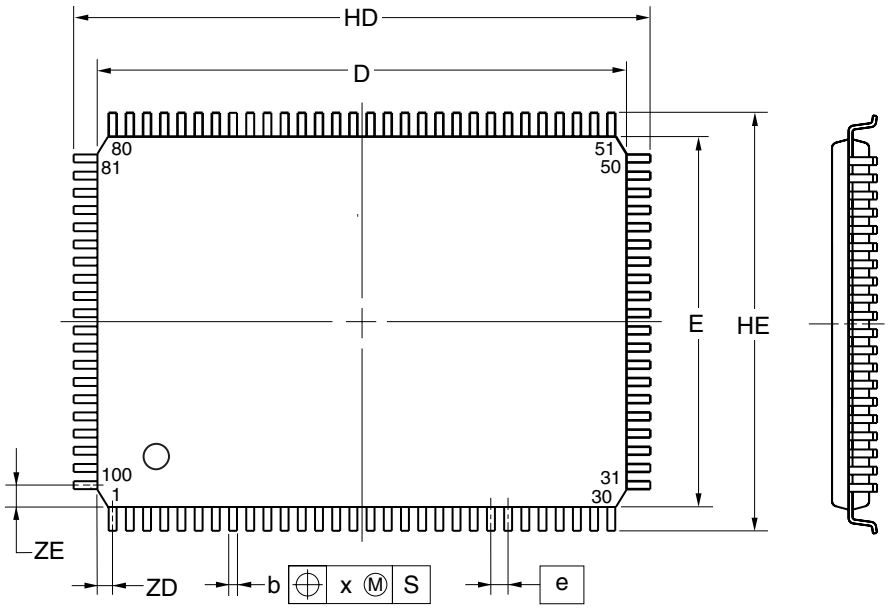
NOTE

Each lead centerline is located within 0.08 mm of its true position (T.P.) at maximum material condition.

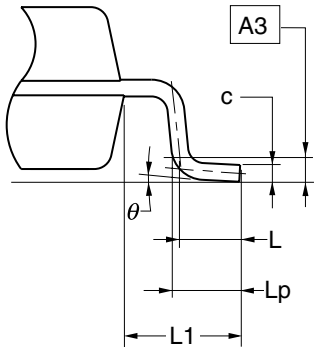
ITEM	MILLIMETERS
A	16.00±0.20
B	14.00±0.20
C	14.00±0.20
D	16.00±0.20
F	1.00
G	1.00
H	$0.22^{+0.05}_{-0.04}$
I	0.08
J	0.50 (T.P.)
K	1.00±0.20
L	0.50±0.20
M	$0.17^{+0.03}_{-0.07}$
N	0.08
P	1.40±0.05
Q	0.10±0.05
R	$3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$
S	1.60 MAX.

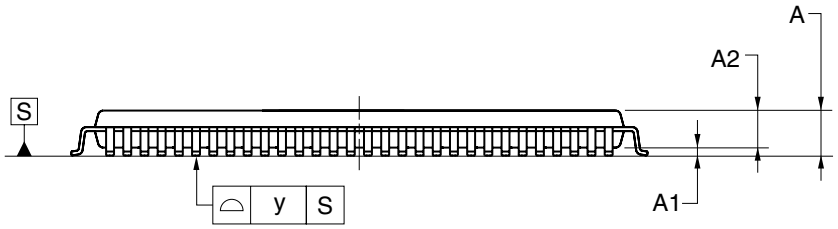
S100GC-50-8EU, 8EA-2

100-PIN PLASTIC LQFP (14x20)



detail of lead end





(UNIT:mm)

ITEM	DIMENSIONS
D	20.00±0.20
E	14.00±0.20
HD	22.00±0.20
HE	16.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
[A3]	0.25
b	$0.30^{+0.08}_{-0.04}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
[e]	0.65
x	0.13
y	0.10
ZD	0.575
ZE	0.825

P100GF-65-GAS

NOTE
Each lead centerline is located within 0.13 mm of its true position at maximum material condition.

CHAPTER 30 RECOMMENDED SOLDERING CONDITIONS

Undefined

APPENDIX A CAUTIONS

A.1 Restriction on Conflict Between sld Instruction and Interrupt Request

A.1.1 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.w 0x28, 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.

A.1.2 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.

APPENDIX B REGISTER INDEX

(1/13)

Symbol	Name	Unit	Page
AD0CH1	A/D converter 0 channel specification register 1	ADC0	625
AD0CH2	A/D converter 0 channel specification register 2	ADC0	627
AD0CHEN	A/D converter 0 conversion channel specification register	ADC0	617
AD0CHENH	A/D converter 0 conversion channel specification register H	ADC0	617
AD0CHENL	A/D converter 0 conversion channel specification register L	ADC0	617
AD0CR0	A/D0 conversion result register 0	ADC0	619
AD0CR0H	A/D0 conversion result register 0H	ADC0	619
AD0CR1	A/D0 conversion result register 1	ADC0	619
AD0CR10	A/D0 conversion result register 10	ADC0	619
AD0CR10H	A/D0 conversion result register 10H	ADC0	619
AD0CR11	A/D0 conversion result register 11	ADC0	619
AD0CR11H	A/D0 conversion result register 11H	ADC0	619
AD0CR12	A/D0 conversion result register 12	ADC0	619
AD0CR12H	A/D0 conversion result register 12H	ADC0	619
AD0CR13	A/D0 conversion result register 13	ADC0	619
AD0CR13H	A/D0 conversion result register 13H	ADC0	619
AD0CR14	A/D0 conversion result register 14	ADC0	619
AD0CR14H	A/D0 conversion result register 14H	ADC0	619
AD0CR15	A/D0 conversion result register 15	ADC0	619
AD0CR15H	A/D0 conversion result register 15H	ADC0	619
AD0CR1H	A/D0 conversion result register 1H	ADC0	619
AD0CR2	A/D0 conversion result register 2	ADC0	619
AD0CR2H	A/D0 conversion result register 2H	ADC0	619
AD0CR3	A/D0 conversion result register 3	ADC0	619
AD0CR3H	A/D0 conversion result register 3H	ADC0	619
AD0CR4	A/D0 conversion result register 4	ADC0	619
AD0CR4H	A/D0 conversion result register 4H	ADC0	619
AD0CR5	A/D0 conversion result register 5	ADC0	619
AD0CR5H	A/D0 conversion result register 5H	ADC0	619
AD0CR6	A/D0 conversion result register 6	ADC0	619
AD0CR6H	A/D0 conversion result register 6H	ADC0	619
AD0CR7	A/D0 conversion result register 7	ADC0	619
AD0CR7H	A/D0 conversion result register 7H	ADC0	619
AD0CR8	A/D0 conversion result register 8	ADC0	619
AD0CR8H	A/D0 conversion result register 8H	ADC0	619
AD0CR9	A/D0 conversion result register 9	ADC0	619
AD0CR9H	A/D0 conversion result register 9H	ADC0	619
AD0CTC	A/D converter 0 conversion time control register	ADC0	616
AD0CTL0	A/D converter 0 control register	ADC0	623

(2/13)

Symbol	Name	Unit	Page
AD0ECR0	A/D0 conversion result extension register 0	ADC0	629
AD0ECR0H	A/D0 conversion result extension register 0H	ADC0	629
AD0ECR1	A/D0 conversion result extension register 1	ADC0	629
AD0ECR1H	A/D0 conversion result extension register 1H	ADC0	629
AD0ECR2	A/D0 conversion result extension register 2	ADC0	629
AD0ECR2H	A/D0 conversion result extension register 2H	ADC0	629
AD0ECR3	A/D0 conversion result extension register 3	ADC0	629
AD0ECR3H	A/D0 conversion result extension register 3H	ADC0	629
AD0ECR4	A/D0 conversion result extension register 4	ADC0	629
AD0ECR4H	A/D0 conversion result extension register 4H	ADC0	629
AD0FLG	A/D converter 0 flag register	ADC0	631
AD0FLGB	A/D converter 0 flag buffer register	ADC0	632
AD0IC	Interrupt control register	ADC0	992
AD0OCKS	A/D converter 0 clock select register	ADC0	643
AD0SCM	A/D converter 0 scan mode register	ADC0	614
AD0SCMH	A/D converter 0 scan mode register H	ADC0	614
AD0SCML	A/D converter 0 scan mode register L	ADC0	614
AD0TSEL	A/D converter 0 trigger select register	ADC0	624
AD1CH1	A/D converter 1 channel specification register 1	ADC1	625
AD1CH2	A/D converter 1 channel specification register 2	ADC1	627
AD1CHEN	A/D converter 1 conversion channel specification register	ADC1	617
AD1CHENH	A/D converter 1 conversion channel specification register H	ADC1	617
AD1CHENL	A/D converter 1 conversion channel specification register L	ADC1	617
AD1CR0	A/D1 conversion result register 0	ADC1	619
AD1CR0H	A/D1 conversion result register 0H	ADC1	619
AD1CR1	A/D1 conversion result register 1	ADC1	619
AD1CR10	A/D1 conversion result register 10	ADC1	619
AD1CR10H	A/D1 conversion result register 10H	ADC1	619
AD1CR11	A/D1 conversion result register 11	ADC1	619
AD1CR11H	A/D1 conversion result register 11H	ADC1	619
AD1CR12	A/D1 conversion result register 12	ADC1	619
AD1CR12H	A/D1 conversion result register 12H	ADC1	619
AD1CR13	A/D1 conversion result register 13	ADC1	619
AD1CR13H	A/D1 conversion result register 13H	ADC1	619
AD1CR14	A/D1 conversion result register 14	ADC1	619
AD1CR14H	A/D1 conversion result register 14H	ADC1	619
AD1CR15	A/D1 conversion result register 15	ADC1	619
AD1CR15H	A/D1 conversion result register 15H	ADC1	619
AD1CR1H	A/D1 conversion result register 1H	ADC1	619
AD1CR2	A/D1 conversion result register 2	ADC1	619
AD1CR2H	A/D1 conversion result register 2H	ADC1	619
AD1CR3	A/D1 conversion result register 3	ADC1	619
AD1CR3H	A/D1 conversion result register 3H	ADC1	619
AD1CR4	A/D1 conversion result register 4	ADC1	619

(3/13)

Symbol	Name	Unit	Page
AD1CR4H	A/D1 conversion result register 4H	ADC1	619
AD1CR5	A/D1 conversion result register 5	ADC1	619
AD1CR5H	A/D1 conversion result register 5H	ADC1	619
AD1CR6	A/D1 conversion result register 6	ADC1	619
AD1CR6H	A/D1 conversion result register 6H	ADC1	619
AD1CR7	A/D1 conversion result register 7	ADC1	619
AD1CR7H	A/D1 conversion result register 7H	ADC1	619
AD1CR8	A/D1 conversion result register 8	ADC1	619
AD1CR8H	A/D1 conversion result register 8H	ADC1	619
AD1CR9	A/D1 conversion result register 9	ADC1	619
AD1CR9H	A/D1 conversion result register 9H	ADC1	619
AD1CTC	A/D converter 1 conversion time control register	ADC1	616
AD1CTL0	A/D converter 1 control register	ADC1	623
AD1ECR0	A/D1 conversion result extension register 0	ADC1	629
AD1ECR0H	A/D1 conversion result extension register 0H	ADC1	629
AD1ECR1	A/D1 conversion result extension register 1	ADC1	629
AD1ECR1H	A/D1 conversion result extension register 1H	ADC1	629
AD1ECR2	A/D1 conversion result extension register 2	ADC1	629
AD1ECR2H	A/D1 conversion result extension register 2H	ADC1	629
AD1ECR3	A/D1 conversion result extension register 3	ADC1	629
AD1ECR3H	A/D1 conversion result extension register 3H	ADC1	629
AD1ECR4	A/D1 conversion result extension register 4	ADC1	629
AD1ECR4H	A/D1 conversion result extension register 4H	ADC1	629
AD1FLG	A/D converter 1 flag register	ADC1	631
AD1FLGB	A/D converter 1 flag buffer register	ADC1	632
AD1IC	Interrupt control register	ADC1	992
AD1OCKS	A/D converter 1 clock select register	ADC1	643
AD1SCM	A/D converter 1 scan mode register	ADC1	614
AD1SCMH	A/D converter 1 scan mode register H	ADC1	614
AD1SCML	A/D converter 1 scan mode register L	ADC1	614
AD1TSEL	A/D converter 1 trigger select register	ADC1	624
AD2CR0	A/D2 conversion result register 0	ADC2	687
AD2CR0H	A/D2 conversion result register 0H	ADC2	687
AD2CR1	A/D2 conversion result register 1	ADC2	687
AD2CR1H	A/D2 conversion result register 1H	ADC2	687
AD2CR2	A/D2 conversion result register 2	ADC2	687
AD2CR2H	A/D2 conversion result register 2H	ADC2	687
AD2CR3	A/D2 conversion result register 3	ADC2	687
AD2CR3H	A/D2 conversion result register 3H	ADC2	687
AD2CR4	A/D2 conversion result register 4	ADC2	687
AD2CR4H	A/D2 conversion result register 4H	ADC2	687
AD2CR5	A/D2 conversion result register 5	ADC2	687
AD2CR5H	A/D2 conversion result register 5H	ADC2	687
AD2CR6	A/D2 conversion result register 6	ADC2	687

(4/13)

Symbol	Name	Unit	Page
AD2CR6H	A/D2 conversion result register 6H	ADC2	687
AD2CR7	A/D2 conversion result register 7	ADC2	687
AD2CR7H	A/D2 conversion result register 7H	ADC2	687
AD2IC	Interrupt control register	ADC2	992
AD2M0	A/D converter 2 mode register 0	ADC2	684
AD2M1	A/D converter 2 mode register 1	ADC2	685
AD2S	A/D converter 2 channel specification register	ADC2	686
ADLTS1	A/DLDTRG1 input select register	ADC0	633
ADLTS2	A/DLDTRG2 input select register	ADC1	633
ADT0IC	Interrupt control register	INTC	992
ADT1IC	Interrupt control register	INTC	992
ADTF	A/D trigger falling edge specification register	ADC0	645
ADTR	A/D trigger rising edge specification register	ADC0	645
AWC	Address wait control register	BCU	935
BCC	Bus cycle control register	BCU	938
BCT0	Bus cycle type configuration register 0	BCU	923
BSC	Bus size configuration register	BCU	925
CB0CTL0	CSIB0 control register 0	CSIB	804
CB0CTL1	CSIB0 control register 1	CSIB	807
CB0CTL2	CSIB0 control register 2	CSIB	808
CB0REIC	Interrupt control register	INTC	992
CB0RIC	Interrupt control register	INTC	992
CB0RX	CSIB0 receive data register	CSIB	803
CB0RXL	CSIB0 receive data register L	CSIB	803
CB0STR	CSIB0 status register	CSIB	810
CB0TIC	Interrupt control register	INTC	992
CB0TX	CSIB0 transmit data register	CSIB	803
CB0TXL	CSIB0 transmit data register L	CSIB	803
CB1CTL0	CSIB1 control register 0	CSIB	804
CB1CTL1	CSIB1 control register 1	CSIB	807
CB1CTL2	CSIB1 control register 2	CSIB	808
CB1REIC	Interrupt control register	CSIB	992
CB1RIC	Interrupt control register	CSIB	992
CB1RX	CSIB1 receive data register	CSIB	803
CB1RXL	CSIB1 receive data register L	CSIB	803
CB1STR	CSIB1 status register	CSIB	810
CB1TIC	Interrupt control register	CSIB	992
CB1TX	CSIB1 transmit data register	CSIB	803
CB1TXL	CSIB1 transmit data register L	CSIB	803
CB2CTL0	CSIB2 control register 0	CSIB	804
CB2CTL1	CSIB2 control register 1	CSIB	807
CB2CTL2	CSIB2 control register 2	CSIB	808
CB2REIC	Interrupt control register	CSIB	992
CB2RIC	Interrupt control register	CSIB	992

(5/13)

Symbol	Name	Unit	Page
CB2RX	CSIB2 receive data register	CSIB	803
CB2RXL	CSIB2 receive data register L	CSIB	803
CB2STR	CSIB2 status register	CSIB	810
CB2TIC	Interrupt control register	CSIB	992
CB2TX	CSIB2 transmit data register	CSIB	803
CB2TXL	CSIB2 transmit data register L	CSIB	803
CLM	Clock monitor mode register	CG	180
CMP0CTL0	Comparator 0 control register 0	ADC0	636
CMP0CTL1	Comparator 0 control register 1	ADC0	638
CMP0CTL2	Comparator 0 control register 2	ADC0	640
CMP0CTL3	Comparator 0 control register 3	ADC0	641
CMP1CTL0	Comparator 1 control register 0	ADC0	636
CMP1CTL1	Comparator 1 control register 1	ADC0	638
CMP1CTL2	Comparator 1 control register 2	ADC0	640
CMP1CTL3	Comparator 1 control register 3	ADC0	641
CMPIC0F	Interrupt control register	INTC	992
CMPIC0L	Interrupt control register	INTC	992
CMPIC1F	Interrupt control register	INTC	992
CMPIC1L	Interrupt control register	INTC	992
CMPNFC0F	Comparator output digital noise elimination register 0F	ADC0	644
CMPNFC0L	Comparator output digital noise elimination register 0L	ADC0	644
CMPNFC1F	Comparator output digital noise elimination register 1F	ADC0	644
CMPNFC1L	Comparator output digital noise elimination register 1L	ADC0	644
CMPOF	Comparator output interrupt falling edge specification register	ADC0	646
CMPOR	Comparator output interrupt rising edge specification register	ADC0	646
DADC0	DMA addressing control register 0	DMAC	958
DADC1	DMA addressing control register 1	DMAC	958
DADC2	DMA addressing control register 2	DMAC	958
DADC3	DMA addressing control register 3	DMAC	958
DBC0	DMA transfer count register 0	DMAC	957
DBC1	DMA transfer count register 1	DMAC	957
DBC2	DMA transfer count register 2	DMAC	957
DBC3	DMA transfer count register 3	DMAC	957
DCHC0	DMA channel control register 0	DMAC	959
DCHC1	DMA channel control register 1	DMAC	959
DCHC2	DMA channel control register 2	DMAC	959
DCHC3	DMA channel control register 3	DMAC	959
DDA0H	DMA destination address register 0H	DMAC	955
DDA0L	DMA destination address register 0L	DMAC	956
DDA1H	DMA destination address register 1H	DMAC	955
DDA1L	DMA destination address register 1L	DMAC	956
DDA2H	DMA destination address register 2H	DMAC	955
DDA2L	DMA destination address register 2L	DMAC	956
DDA3H	DMA destination address register 3H	DMAC	955

(6/13)

Symbol	Name	Unit	Page
DDA3L	DMA destination address register 3L	DMAC	956
DMAIC0	Interrupt control register	INTC	992
DMAIC1	Interrupt control register	INTC	992
DMAIC2	Interrupt control register	INTC	992
DMAIC3	Interrupt control register	INTC	992
DSA0H	DMA source address register 0H	DMAC	953
DSA0L	DMA source address register 0L	DMAC	954
DSA1H	DMA source address register 1H	DMAC	953
DSA1L	DMA source address register 1L	DMAC	954
DSA2H	DMA source address register 2H	DMAC	953
DSA2L	DMA source address register 2L	DMAC	954
DSA3H	DMA source address register 3H	DMAC	953
DSA3L	DMA source address register 3L	DMAC	954
DTFR0	DMA trigger factor register 0	DMAC	961
DTFR1	DMA trigger factor register 1	DMAC	961
DTFR2	DMA trigger factor register 2	DMAC	961
DTFR3	DMA trigger factor register 3	DMAC	961
DVC	Bus clock division control register	BCU	939
DWC0	Data wait control register 0	BCU	933
HZA0CTL0	High-impedance output control register 00	Timer	548
HZA0CTL1	High-impedance output control register 01	Timer	548
HZA1CTL0	High-impedance output control register 10	Timer	548
HZA1CTL1	High-impedance output control register 11	Timer	548
HZA2CTL0	High-impedance output control register 20	Timer	548
HZA2CTL1	High-impedance output control register 21	Timer	548
HZA3CTL0	High-impedance output control register 30	Timer	548
HZA3CTL1	High-impedance output control register 31	Timer	548
IIC0	IIC shift register 0	I^2C	864
IICC0	IIC control register 0	I^2C	852
IICCL0	IIC clock select register 0	I^2C	861
IICF0	IIC flag register 0	I^2C	859
IICIC	Interrupt control register	INTC	992
IICOCKS	IICOPS clock select register	I^2C	862
IICS0	IIC status register 0	I^2C	856
IICX0	IIC function expansion register 0	I^2C	862
IMR0	Interrupt mask register 0	INTC	997
IMR0H	Interrupt mask register 0H	INTC	997
IMR0L	Interrupt mask register 0L	INTC	997
IMR1	Interrupt mask register 1	INTC	997
IMR1H	Interrupt mask register 1H	INTC	997
IMR1L	Interrupt mask register 1L	INTC	997
IMR2	Interrupt mask register 2	INTC	997
IMR2H	Interrupt mask register 2H	INTC	997
IMR2L	Interrupt mask register 2L	INTC	997

(7/13)

Symbol	Name	Unit	Page
IMR3	Interrupt mask register 3	INTC	997
IMR3H	Interrupt mask register 3H	INTC	997
IMR3L	Interrupt mask register 3L	INTC	997
IMR4	Interrupt mask register 4	INTC	997
IMR4H	Interrupt mask register 4H	INTC	997
IMR4L	Interrupt mask register 4L	INTC	997
IMR5	Interrupt mask register 5	INTC	997
IMR5H	Interrupt mask register 5H	INTC	997
IMR5L	Interrupt mask register 5L	INTC	997
INTF0	External interrupt falling edge specification register 0	INTC	1003
INTF1	External interrupt falling edge specification register 1	INTC	1004
INTF2	External interrupt falling edge specification register 2	INTC	1005
INTNFC14	Digital noise elimination 0 control register 14	Port	166
INTNFC15	Digital noise elimination 0 control register 15	Port	166
INTNFC16	Digital noise elimination 0 control register 16	Port	166
INTR0	External interrupt rising edge specification register 0	INTC	1003
INTR1	External interrupt rising edge specification register 1	INTC	1004
INTR2	External interrupt rising edge specification register 2	INTC	1005
ISPR	In-service priority register	INTC	1000
LVIHIC	Interrupt control register	INTC	992
LVILIC	Interrupt control register	INTC	992
LVIM	Low-voltage detection register	LVI	1034
LVIS	Low-voltage detection level select register	LVI	1035
OP0CTL0	Operational amplifier 0 control register 0	ADC0	634
OP1CTL0	Operational amplifier 1 control register 0	ADC0	634
OSTS	Oscillation stabilization time select register	CG	179
P0	Port 0 register	Port	105
P1	Port 1 register	Port	111
P2	Port 2 register	Port	117
P3	Port 3 register	Port	123
P4	Port 4 register	Port	129
P7	Port 7 register	Port	135
PCC	Processor clock control register	CG	176
PDL	Port DL register	Port	137
PDLH	Port DLH register	Port	137
PDLL	Port DLL register	Port	137
PF3	Port 3 function register	Port	127
PFC0	Port 0 function control register	Port	107
PFC1	Port 1 function control register	Port	113
PFC2	Port 2 function control register	Port	119
PFC3	Port 3 function control register	Port	125
PFC4	Port 4 function control register	Port	131
PFCE0	Port 0 function control expansion register	Port	107
PFCE1	Port 1 function control expansion register	Port	113

(8/13)

Symbol	Name	Unit	Page
PFCE2	Port 2 function control expansion register	Port	119
PFCE3	Port 3 function control expansion register	Port	125
PFCE4	Port 4 function control expansion register	Port	131
PIC00	Interrupt control register	INTC	992
PIC01	Interrupt control register	INTC	992
PIC02	Interrupt control register	INTC	992
PIC03	Interrupt control register	INTC	992
PIC04	Interrupt control register	INTC	992
PIC05	Interrupt control register	INTC	992
PIC06	Interrupt control register	INTC	992
PIC07	Interrupt control register	INTC	992
PIC08	Interrupt control register	INTC	992
PIC09	Interrupt control register	INTC	992
PIC10	Interrupt control register	INTC	992
PIC11	Interrupt control register	INTC	992
PIC12	Interrupt control register	INTC	992
PIC13	Interrupt control register	INTC	992
PIC14	Interrupt control register	INTC	992
PIC15	Interrupt control register	INTC	992
PIC16	Interrupt control register	INTC	992
PIC17	Interrupt control register	INTC	992
PIC18	Interrupt control register	INTC	992
PLLCTL	PLL control register	CG	175
PM0	Port 0 mode register	Port	105
PM1	Port 1 mode register	Port	111
PM2	Port 2 mode register	Port	117
PM3	Port 3 mode register	Port	123
PM4	Port 4 mode register	Port	129
PMC0	Port 0 mode control register	Port	106
PMC1	Port 1 mode control register	Port	112
PMC2	Port 2 mode control register	Port	118
PMC3	Port 3 mode control register	Port	124
PMC4	Port 4 mode control register	Port	130
PMC7	Port 7 mode control register	Port	135
PMCDL	Port DL mode control register	Port	139
PMCDLH	Port DL mode control register H	Port	139
PMCDLL	Port DL mode control register L	Port	139
PMDL	Port DL mode register	Port	138
PMDLH	Port DL mode register H	Port	138
PMDLL	Port DL mode register L	Port	138
PRCMD	Command register	CPU	92
PSC	Power save control register	CPU	177, 1019
PSMR	Power save mode register	CPU	178, 1020
PU0	Pull-up resistor option register 0	Port	109

(9/13)

Symbol	Name	Unit	Page
PU1	Pull-up resistor option register 1	Port	115
PU2	Pull-up resistor option register 2	Port	121
PU3	Pull-up resistor option register 3	Port	127
PU4	Pull-up resistor option register 4	Port	133
PUDL	Pull-up resistor option register DL	Port	140
PUDLH	Pull-up resistor option register DLH	Port	140
PUDLL	Pull-up resistor option register DLL	Port	140
RESF	Reset source flag register	Reset	1029
SVA0	Slave address register 0	I^2C	864
SYS	System status register	CPU	93
TA0CCIC0	Interrupt control register	INTC	992
TA0CCIC1	Interrupt control register	INTC	992
TA0OVIC	Interrupt control register	INTC	992
TA1CCIC0	Interrupt control register	INTC	992
TA1CCIC1	Interrupt control register	INTC	992
TA1OVIC	Interrupt control register	INTC	992
TA2CCIC0	Interrupt control register	INTC	992
TA2CCIC1	Interrupt control register	INTC	992
TA2OVIC	Interrupt control register	INTC	992
TA3CCIC0	Interrupt control register	INTC	992
TA3CCIC1	Interrupt control register	INTC	992
TA3OVIC	Interrupt control register	INTC	992
TA4CCIC0	Interrupt control register	INTC	992
TA4CCIC1	Interrupt control register	INTC	992
TA4OVIC	Interrupt control register	INTC	992
TAA0CCR0	TAA0 capture/compare register 0	TAA	204
TAA0CCR1	TAA0 capture/compare register 1	TAA	206
TAA0CNT	TAA0 counter read buffer register	TAA	208
TAA0CTL0	TAA0 control register 0	TAA	196
TAA0CTL1	TAA0 control register 1	TAA	197
TAA0OPT0	TAA0 option register 0	TAA	203
TAA1CCR0	TAA1 capture/compare register 0	TAA	204
TAA1CCR1	TAA1 capture/compare register 1	TAA	206
TAA1CNT	TAA1 counter read buffer register	TAA	208
TAA1CTL0	TAA1 control register 0	TAA	196
TAA1CTL1	TAA1 control register 1	TAA	197
TAA1OPT0	TAA1 option register 0	TAA	203
TAA2CCR0	TAA2 capture/compare register 0	TAA	204
TAA2CCR1	TAA2 capture/compare register 1	TAA	206
TAA2CNT	TAA2 counter read buffer register	TAA	208
TAA2CTL0	TAA2 control register 0	TAA	196
TAA2CTL1	TAA2 control register 1	TAA	197
TAA2IOC0	TAA2 I/O control register 0	TAA	199
TAA2IOC1	TAA2 I/O control register 1	TAA	201

(10/13)

Symbol	Name	Unit	Page
TAA2IOC2	TAA2 I/O control register 2	TAA	202
TAA2OPT0	TAA2 option register 0	TAA	203
TAA3CCR0	TAA3 capture/compare register 0	TAA	204
TAA3CCR1	TAA3 capture/compare register 1	TAA	206
TAA3CNT	TAA3 counter read buffer register	TAA	208
TAA3CTL0	TAA3 control register 0	TAA	196
TAA3CTL1	TAA3 control register 1	TAA	197
TAA3IOC0	TAA3 I/O control register 0	TAA	199
TAA3IOC1	TAA3 I/O control register 1	TAA	201
TAA3IOC2	TAA3 I/O control register 2	TAA	202
TAA3OPT0	TAA3 option register 0	TAA	203
TAA4CCR0	TAA4 capture/compare register 0	TAA	204
TAA4CCR1	TAA4 capture/compare register 1	TAA	206
TAA4CNT	TAA4 counter read buffer register	TAA	208
TAA4CTL0	TAA4 control register 0	TAA	196
TAA4CTL1	TAA4 control register 1	TAA	197
TAA4IOC0	TAA4 I/O control register 0	TAA	199
TAA4IOC1	TAA4 I/O control register 1	TAA	201
TAA4IOC2	TAA4 I/O control register 2	TAA	202
TAA4OPT0	TAA4 option register 0	TAA	203
TAB0CCR0	TAB0 capture/compare register 0	TAB	298
TAB0CCR1	TAB0 capture/compare register 1	TAB	300
TAB0CCR2	TAB0 capture/compare register 2	TAB	302
TAB0CCR3	TAB0 capture/compare register 3	TAB	304
TAB0CNT	TAB0 counter read buffer register	TAB	306
TAB0CTL0	TAB0 control register 0	TAB	292
TAB0CTL1	TAB0 control register 1	TAB	293
TAB0DTC	TAB0 dead-time compare register	Timer	539
TAB0IOC0	TAB0 I/O control register 0	TAB	294
TAB0IOC1	TAB0 I/O control register 1	TAB	295
TAB0IOC2	TAB0 I/O control register 2	TAB	296
TAB0IOC3	TAB0 I/O control register 3	Timer	545
TAB0OPT0	TAB0 option register 0	TAB	297, 540
TAB0OPT1	TAB0 option register 1	Timer	541
TAB0OPT2	TAB0 option register 2	Timer	542
TAB0OPT3	TAB0 option register 3	Timer	544
TAB1CCR0	TAB1 capture/compare register 0	TAB	298
TAB1CCR1	TAB1 capture/compare register 1	TAB	300
TAB1CCR2	TAB1 capture/compare register 2	TAB	302
TAB1CCR3	TAB1 capture/compare register 3	TAB	304
TAB1CNT	TAB1 counter read buffer register	TAB	306
TAB1CTL0	TAB1 control register 0	TAB	292
TAB1CTL1	TAB1 control register 1	TAB	293
TAB1DTC	TAB1 dead-time compare register	TAB	539

(11/13)

Symbol	Name	Unit	Page
TAB1IOC0	TAB1 I/O control register 0	TAB	294
TAB1IOC1	TAB1 I/O control register 1	TAB	295
TAB1IOC2	TAB1 I/O control register 2	TAB	296
TAB1IOC3	TAB1 I/O control register 3	TAB	545
TAB1OPT0	TAB1 option register 0	TAB	297, 540
TAB1OPT1	TAB1 option register 1	TAB	541
TAB1OPT2	TAB1 option register 2	TAB	542
TAB1OPT3	TAB1 option register 3	TAB	544
TANFC2	Digital noise elimination 1 control register 2	Port	167
TANFC3	Digital noise elimination 1 control register 3	Port	167
TANFC4	Digital noise elimination 1 control register 4	Port	167
TB0CCIC0	Interrupt control register	INTC	992
TB0CCIC1	Interrupt control register	INTC	992
TB0CCIC2	Interrupt control register	INTC	992
TB0CCIC3	Interrupt control register	INTC	992
TB0OVIC	Interrupt control register	INTC	992
TB1CCIC0	Interrupt control register	INTC	992
TB1CCIC1	Interrupt control register	INTC	992
TB1CCIC2	Interrupt control register	INTC	992
TB1CCIC3	Interrupt control register	INTC	992
TB1OVIC	Interrupt control register	INTC	992
TM0CMP0	TMM0 compare register 0	TMM	528
TM0CTL0	TMM0 control register 0	TMM	529
TM0EQIC0	Interrupt control register	INTC	992
TM1CMP0	TMM1 compare register 0	TMM	528
TM1CTL0	TMM1 control register 0	TMM	529
TM1EQIC0	Interrupt control register	INTC	992
TM2CMP0	TMM2 compare register 0	TMM	528
TM2CTL0	TMM2 control register 0	TMM	529
TM2EQIC0	Interrupt control register	INTC	992
TM3CMP0	TMM3 compare register 0	TMM	528
TM3CTL0	TMM3 control register 0	TMM	529
TM3EQIC0	Interrupt control register	INTC	992
TT0CCIC0	Interrupt control register	INTC	992
TT0CCIC1	Interrupt control register	INTC	992
TT0CCR0	TMT0 capture/compare register 0	TMT	416
TT0CCR1	TMT0 capture/compare register 1	TMT	418
TT0CNT	TMT0 counter read buffer register	TMT	420
TT0CTL0	TMT0 control register 0	TMT	402
TT0CTL1	TMT0 control register 1	TMT	403
TT0CTL2	TMT0 control register 2	TMT	405
TT0IECIC	Interrupt control register	INTC	992
TT0IOC0	TMT0 I/O control register 0	TMT	407
TT0IOC1	TMT0 I/O control register 1	TMT	408

(12/13)

Symbol	Name	Unit	Page
TT0IOC2	TMT0 I/O control register 2	TMT	409
TT0IOC3	TMT0 I/O control register 3	TMT	410
TT0OPT0	TMT0 option register 0	TMT	412
TT0OPT1	TMT0 option register 1	TMT	413
TT0OVIC	Interrupt control register	INTC	992
TT0TCW	TMT0 counter write register	TMT	420
TT1CCIC0	Interrupt control register	INTC	992
TT1CCIC1	Interrupt control register	INTC	992
TT1CCR0	TMT1 capture/compare register 0	TMT	416
TT1CCR1	TMT1 capture/compare register 1	TMT	418
TT1CNT	TMT1 counter read buffer register	TMT	420
TT1CTL0	TMT1 control register 0	TMT	402
TT1CTL1	TMT1 control register 1	TMT	403
TT1CTL2	TMT1 control register 2	TMT	405
TT1IECIC	Interrupt control register	INTC	992
TT1IOC0	TMT1 I/O control register 0	TMT	407
TT1IOC1	TMT1 I/O control register 1	TMT	408
TT1IOC2	TMT1 I/O control register 2	TMT	409
TT1IOC3	TMT1 I/O control register 3	TMT	410
TT1OPT0	TMT1 option register 0	TMT	412
TT1OPT1	TMT1 option register 1	TMT	413
TT1OVIC	Interrupt control register	INTC	992
TT1TCW	TMT1 counter write register	TMT	420
TTISL0	TMT0 capture input select register	TMT	415
TTISL1	TMT1 capture input select register	TMT	415
TTNFC0	Digital noise elimination 2 control register 0	Port	168
TTNFC1	Digital noise elimination 2 control register 1	Port	168
UA0CTL0	UARTA0 control register 0	UARTA	715
UA0CTL1	UARTA0 control register 1	UARTA	731
UA0CTL2	UARTA0 control register 2	UARTA	732
UA0OPT0	UARTA0 option control register 0	UARTA	717
UA0REIC	Interrupt control register	INTC	992
UA0RIC	Interrupt control register	INTC	992
UA0RX	UARTA0 receive data register	UARTA	719
UA0STR	UARTA0 status register	UARTA	717
UA0TIC	Interrupt control register	INTC	992
UA0TX	UARTA0 transmit data register	UARTA	719
UA1CTL0	UARTA1 control register 0	UARTA	715
UA1CTL1	UARTA1 control register 1	UARTA	731
UA1CTL2	UARTA1 control register 2	UARTA	732
UA1OPT0	UARTA1 option control register 0	UARTA	717
UA1REIC	Interrupt control register	INTC	992
UA1RIC	Interrupt control register	INTC	992
UA1RX	UARTA1 receive data register	UARTA	719

(13/13)

Symbol	Name	Unit	Page
UA1STR	UARTA1 status register	UARTA	717
UA1TIC	Interrupt control register	INTC	992
UA1TX	UARTA1 transmit data register	UARTA	719
UA2CTL0	UARTA2 control register 0	UARTA	715
UA2CTL1	UARTA2 control register 1	UARTA	731
UA2CTL2	UARTA2 control register 2	UARTA	732
UA2OPT0	UARTA2 option control register 0	UARTA	717
UA2REIC	Interrupt control register	INTC	992
UA2RIC	Interrupt control register	INTC	992
UA2RX	UARTA2 receive data register	UARTA	719
UA2STR	UARTA2 status register	UARTA	717
UA2TIC	Interrupt control register	INTC	992
UA2TX	UARTA2 transmit data register	UARTA	719
UBCTL0	UARTB control register 0	UARTB	744
UBCTL2	UARTB control register 2	UARTB	749
UBFIC0	UARTB FIFO control register 0	UARTB	753
UBFIC1	UARTB FIFO control register 1	UARTB	755
UBFIC2	UARTB FIFO control register 2	UARTB	756
UBFIC2H	UARTB FIFO control register 2H	UARTB	756
UBFIC2L	UARTB FIFO control register 2L	UARTB	756
UBFIS0	UARTB FIFO status register 0	UARTB	758
UBFIS1	UARTB FIFO status register 1	UARTB	759
UBRX	UARTB receive data register	UARTB	751
UBRXAP	UARTB receive data register AP	UARTB	751
UBSTR	UARTB status register	UARTB	747
UBTX	UARTB transmit data register	UARTB	750
UIFIC	Interrupt control register	INTC	992
UREIC	Interrupt control register	INTC	992
URIC	Interrupt control register	INTC	992
UTIC	Interrupt control register	INTC	992
UTOIC	Interrupt control register	INTC	992
VSWC	System wait control register	BCU	94
WDTE	Watchdog timer enable register	WDT	601
WDTM	Watchdog timer mode register	WDT	600

APPENDIX C INSTRUCTION SET LIST

C.1 Conventions

(1) Register symbols used to describe operands

Register Symbol	Explanation
reg1	General-purpose registers: Used as source registers.
reg2	General-purpose registers: Used mainly as destination registers. Also used as source register in some instructions.
reg3	General-purpose registers: Used mainly to store the remainders of division results and the higher order 32 bits of multiplication results.
bit#3	3-bit data for specifying the bit number
immX	X bit immediate data
dispX	X bit displacement data
regID	System register number
vector	5-bit data that specifies the trap vector (00H to 1FH)
cccc	4-bit data that shows the conditions code
sp	Stack pointer (SP)
ep	Element pointer (r30)
listX	X item register list

(2) Register symbols used to describe opcodes

Register Symbol	Explanation
R	1-bit data of a code that specifies reg1 or regID
r	1-bit data of the code that specifies reg2
w	1-bit data of the code that specifies reg3
d	1-bit displacement data
I	1-bit immediate data (indicates the higher bits of immediate data)
i	1-bit immediate data
cccc	4-bit data that shows the condition codes
CCCC	4-bit data that shows the condition codes of Bcond instruction
bbb	3-bit data for specifying the bit number
L	1-bit data that specifies a program register in the register list
S	1-bit data that specifies a system register in the register list

(3) Register symbols used in operations

Register Symbol	Explanation
←	Input for
GR [ ]	General-purpose register
SR [ ]	System register
zero-extend (n)	Expand n with zeros until word length.
sign-extend (n)	Expand n with signs until word length.
load-memory (a, b)	Read size b data from address a.
store-memory (a, b, c)	Write data b into address a in size c.
load-memory-bit (a, b)	Read bit b of address a.
store-memory-bit (a, b, c)	Write c to bit b of address a.
saturated (n)	Execute saturated processing of n (n is a 2's complement). If, as a result of calculations, n ≥ 7FFFFFFFH, let it be 7FFFFFFFH. n ≤ 80000000H, let it be 80000000H.
result	Reflects the results in a flag.
Byte	Byte (8 bits)
Halfword	Half word (16 bits)
Word	Word (32 bits)
+	Addition
–	Subtraction
ll	Bit concatenation
×	Multiplication
÷	Division
%	Remainder from division results
AND	Logical product
OR	Logical sum
XOR	Exclusive OR
NOT	Logical negation
logically shift left by	Logical shift left
logically shift right by	Logical shift right
arithmetically shift right by	Arithmetic shift right

(4) Register symbols used in execution clock

Register Symbol	Explanation
i	If executing another instruction immediately after executing the first instruction (issue).
r	If repeating execution of the same instruction immediately after executing the first instruction (repeat).
l	If using the results of instruction execution in the instruction immediately after the execution (latency).

(5) Register symbols used in flag operations

Identifier	Explanation
(Blank)	No change
0	Clear to 0
X	Set or cleared in accordance with the results.
R	Previously saved values are restored.

(6) Condition codes

Condition Name (cond)	Condition Code (cccc)	Condition Formula	Explanation
V	0 0 0 0	OV = 1	Overflow
NV	1 0 0 0	OV = 0	No overflow
C/L	0 0 0 1	CY = 1	Carry Lower (Less than)
NC/NL	1 0 0 1	CY = 0	No carry Not lower (Greater than or equal)
Z/E	0 0 1 0	Z = 1	Zero Equal
NZ/NE	1 0 1 0	Z = 0	Not zero Not equal
NH	0 0 1 1	(CY or Z) = 1	Not higher (Less than or equal)
H	1 0 1 1	(CY or Z) = 0	Higher (Greater than)
N	0 1 0 0	S = 1	Negative
P	1 1 0 0	S = 0	Positive
T	0 1 0 1	–	Always (Unconditional)
SA	1 1 0 1	SAT = 1	Saturated
LT	0 1 1 0	(S xor OV) = 1	Less than signed
GE	1 1 1 0	(S xor OV) = 0	Greater than or equal signed
LE	0 1 1 1	((S xor OV) or Z) = 1	Less than or equal signed
GT	1 1 1 1	((S xor OV) or Z) = 0	Greater than signed

C.2 Instruction Set (in Alphabetical Order)

(1/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock i	Execution Clock r	Execution Clock l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	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	rrrrr110000RRRRR 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	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]		1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1]AND zero-extend(imm16)		1	1	1		0	0	×	
Bcond	disp9	ddddd1011dddcccc **Note 1**	if conditions are satisfied then PC←PC+sign-extend(disp9)	When conditions are satisfied	3 Note 2	3 Note 2	3 Note 2					
				When conditions are not satisfied	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 wwwww01101000010	GR[reg3]←GR[reg2] (23:16) ll GR[reg2] (31:24) ll GR[reg2] (7:0) ll GR[reg2] (15:8)		1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 wwwww01101000000	GR[reg3]←GR[reg2] (7:0) ll GR[reg2] (15:8) ll GR [reg2] (23:16) ll GR[reg2] (31:24)		1	1	1	×	0	×	×	
CALLT	imm6	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]	10bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)		3 Note 3	3 Note 3	3 Note 3				×	
CMOV	cccc,imm5,reg2,reg3	r r r r r 1 1 1 1 1 i i i i i wwwww011000cccc0	if conditions are satisfied then GR[reg3]←sign-extended(imm5) else GR[reg3]←GR[reg2]		1	1	1					
	cccc,reg1,reg2,reg3	rrrrr111111RRRR wwwww011001cccc0	if conditions are satisfied then GR[reg3]←GR[reg1] else GR[reg3]←GR[reg2]		1	1	1					
CMP	reg1,reg2	rrrrr001111RRRRR	result←GR[reg2]–GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	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		0000011111100000 0000000101000100	PC←CTPC PSW←CTPSW		4	4	4	R	R	R	R	R
DBRET		0000011111100000 0000000101000110	PC←DBPC PSW←DBPSW		4	4	4	R	R	R	R	R

(2/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
DBTRAP		1111100001000000	DBPC←PC+2(return PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	4	4	4					
DI		0000011111100000 0000000101100000	PSW.ID←1	1	1	1					
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 all regs in list12 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 all regs in list12 is loaded PC←GR[reg1]	n+3 Note 4	n+3 Note 4	n+3 Note 4					
DIV	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000000	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVH	reg1,reg2	rrrrr000010RRRRR	GR[reg2]←GR[reg2]÷GR[reg1]Note 6	35	35	35		×	×	×	
	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000000	GR[reg2]←GR[reg2]÷GR[reg1]Note 6 GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVHU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000010	GR[reg2]←GR[reg2]÷GR[reg1]Note 6 GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
DIVU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000010	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
EI		1000011111100000 0000000101100000	PSW.ID←0	1	1	1					
HALT		0000011111100000 0000000100100000	Stop	1	1	1					
HSW	reg2,reg3	rrrrr111111100000 wwwww01101000100	GR[reg3]←GR[reg2](15:0) ll GR[reg2] (31:16)	1	1	1	×	0	×	×	
JARL	disp22,reg2	rrrrr11110ddddddd ddddddddddddddd0 **Note 7**	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	3	3	3					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	4	4	4					
JR	disp22	0000011110ddddddd ddddddddddddddd0 **Note 7**	PC←PC+sign-extend(disp22)	3	3	3					
LD.B	disp16[reg1],reg2	rrrrr111000RRRRR ddddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 11					
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRRR ddddddddddddddd1 **Notes 8, 10**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 11					

(3/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
LD.H	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Halfword))		1	1	Note 11					
LDSR	reg2,regID	rrrrr111111RRRRR 0000000000100000 **Note 12**	SR[regID]←GR[reg2]	Other than regID = PSW	1	1	1					
				regID = PSW	1	1	1	×	×	×	×	×
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Halfword)		1	1	Note 11					
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←Load-memory(adr,Word)		1	1	Note 11					
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]		1	1	1					
	imm5,reg2	rrrrr010000iiiii	GR[reg2]←sign-extend(imm5)		1	1	1					
	imm32,reg1	00000110001RRRRR iiiiiiiiiiiiiiii lllllllllllllll	GR[reg1]←imm32		2	2	2					
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1					
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+(imm16 ll 0^{16})		1	1	1					
MUL Note 22	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1]		1	2 Note 14	2					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII00 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xsign-extend(imm9)		1	2 Note 14	2					
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2] Note 6 xGR[reg1] Note 6		1	1	2					
	imm5,reg2	rrrrr010111iiiii	GR[reg2]←GR[reg2] Note 6 xsign-extend(imm5)		1	1	2					
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] Note 6 ximm16		1	1	2					
MULU Note 22	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1]		1	2 Note 14	2					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII10 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xzero-extend(imm9)		1	2 Note 14	2					
NOP		0000000000000000	Pass at least one clock cycle doing nothing.		1	1	1					
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])		1	1	1		0	×	×	
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,Z flag)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,Z flag)		3 Note 3	3 Note 3	3 Note 3				×	

(4/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
OR	reg1,reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×	
ORI	imm16,reg1,reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×	
PREPARE	list12,imm5	0000011110iiiiiL LLLLLLLLLLL00001	Store-memory(sp–4,GR[reg in list12],Word) sp←sp–4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5)	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	list12,imm5, sp/imm[Note 15]	0000011110iiiiiL LLLLLLLLLLLff011 imm16/imm32 **Note 16**	Store-memory(sp–4,GR[reg in list12],Word) GR[reg in list 12]←Load-memory(sp,Word) sp←sp+4 repeat 2 step above until all 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					
RETI		0000011111100000 0000000101000000	if PSW.EP=1 then PC ←EIPC PSW ←EIPSW else if PSW.NP=1 then PC ←FEPC PSW ←FEPSW else PC ←EIPC PSW ←EIPSW	4	4	4	R	R	R	R	R
SAR	reg1,reg2	rrrrr111111RRRRR 0000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010101iiiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SASF	cccc,reg2	rrrrr1111110cccc 0000001000000000	if conditions are satisfied then GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000001H else GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000000H	1	1	1					
SATADD	reg1,reg2	rrrrr000110RRRRR	GR[reg2]←saturated(GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×
	imm5,reg2	rrrrr010001iiiii	GR[reg2]←saturated(GR[reg2]+sign-extend(imm5)	1	1	1	×	×	×	×	×
SATSUB	reg1,reg2	rrrrr000101RRRRR	GR[reg2]←saturated(GR[reg2]–GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16,reg1,reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiii	GR[reg2]←saturated(GR[reg1]–sign-extend(imm16)	1	1	1	×	×	×	×	×
SATSUBR	reg1,reg2	rrrrr000100RRRRR	GR[reg2]←saturated(GR[reg1]–GR[reg2])	1	1	1	×	×	×	×	×
SETF	cccc,reg2	rrrrr1111110cccc 0000000000000000	If conditions are satisfied then GR[reg2]←00000001H else GR[reg2]←00000000H	1	1	1					

(5/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SET1	bit#3,disp16[reg1]	00bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,1)	3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 Note 3	3 Note 3	3 Note 3				×	
SHL	reg1,reg2	rrrrr111111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010110iiiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×	
SHR	reg1,reg2	rrrrr111111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010100iiiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SLD.B	disp7[ep],reg2	rrrrr0110ddddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.BU	disp4[ep],reg2	rrrrr0000110dddd **Note 18**	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.H	disp8[ep],reg2	rrrrr1000ddddddd **Note 19**	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	Note 9					
SLD.HU	disp5[ep],reg2	rrrrr0000111dddd **Notes 18, 20**	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,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 dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1					
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],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 0000000001000000	GR[reg2]←SR[regID]	1	1	1					

(6/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1,reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]–GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]–GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	00000000010RRRRR	adr←(PC+2) + (GR[reg1] logically shift left by 1) PC←(PC+2) + (sign-extend (Load-memory(adr,Halfword)) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←sign-extend (GR[reg1] (7:0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←sign-extend (GR[reg1] (15:0))	1	1	1					
TRAP	vector	0 0 0 0 0 1 1 1 1 1 1 i i i i i 0000000100000000	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 0000000011100110	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2))	3 Note 3	3 Note 3	3 Note 3				×	
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16,reg1,reg2	rrrrr110101RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1] XOR zero-extend(imm16)	1	1	1		0	×	×	
ZXB	reg1	00000000100RRRRR	GR[reg1]←zero-extend(GR[reg1] (7:0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←zero-extend(GR[reg1] (15:0))	1	1	1					

Notes 1. ddddddddd: Higher 8 bits of disp9.

2. 4 if there is an instruction that rewrites the contents of the PSW immediately before.

3. If there is no wait state (3 + the number of read access wait states).

4. n is the total number of list12 load registers. (According to the number of wait states. Also, if there are no wait states, n is the total number of list12 registers. If n = 0, same operation as when n = 1)

5. RRRRR: other than 00000.

6. The lower halfword data only are valid.

7. ddddddddddddddddddddd: The higher 21 bits of disp22.

8. ddddddddddddddd: The higher 15 bits of disp16.

9. According to the number of wait states (1 if there are no wait states).

10. b: bit 0 of disp16.

11. According to the number of wait states (2 if there are no wait states).

Notes 12. In this instruction, for convenience of mnemonic description, the source register is made reg2, but the reg1 field is used in the opcode. Therefore, the meaning of register specification in the mnemonic description and in the opcode differs from other instructions.

rrrrr = regID specification

RRRRR = reg2 specification

13. iiiii: Lower 5 bits of imm9.

IIII: Higher 4 bits of imm9.

14. In the case of reg2 = reg3 (the lower 32 bits of the results are not written in the register) or reg3 = r0 (the higher 32 bits of the results are not written in the register), shortened by 1 clock.

15. sp/imm: specified by bits 19 and 20 of the sub-opcode.

16. ff = 00: Load sp in ep.

01: Load sign expanded 16-bit immediate data (bits 47 to 32) in ep.

10: Load 16-bit logically left shifted 16-bit immediate data (bits 47 to 32) in ep.

11: Load 32-bit immediate data (bits 63 to 32) in ep.

17. If imm = imm32, n + 3 clocks.

18. rrrrr: Other than 00000.

19. ddddddd: Higher 7 bits of disp8.

20. dddd: Higher 4 bits of disp5.

21. dddddd: Higher 6 bits of disp8.

22. Do not make a combination that satisfies all the following conditions when using the “MUL reg1, reg2, reg3” instruction and “MULU reg1, reg2, reg3” instruction. Operation is not guaranteed when an instruction that satisfies the following conditions is executed.

- Reg1 = reg3
- Reg1 ≠ reg2
- Reg1 ≠ r0
- Reg3 ≠ r0

For further information,
please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbielskistrasse 166 B
30177 Hannover
Tel: 0 511 33 40 2-0

Munich Office
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Succursale Française
9, rue Paul Dautier, B.P. 52
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands
Steijgerweg 6
5616 HS Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
Tel: 010-8235-1155
http://www.cn.necel.com/

Shanghai Branch
Room 2509-2510, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai, P.R.China P.C:200120
Tel:021-5888-5400
http://www.cn.necel.com/

Shenzhen Branch
Unit 01, 39/F, Excellence Times Square Building,
No. 4068 Yi Tian Road, Futian District, Shenzhen,
P.R.China P.C:518048
Tel:0755-8282-9800
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
Unit 1601-1613, 16/F., Tower 2, Grand Century Place,
193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-8175-9600
http://www.tw.necel.com/

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/

NEC Electronics Korea Ltd.
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737
http://www.kr.necel.com/

G0706

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for 8-bit Microcontrollers - MCU ***category:***

Click to view products by Renesas ***manufacturer:***

Other Similar products are found below :

CY8C20524-12PVXIT CY8C28433-24PVXIT MB95F012KPFT-G-SNE2 MB95F013KPMC-G-SNE2 MB95F263KPF-G-SNE2 MB95F264KPFT-G-SNE2 MB95F398KPMC-G-SNE2 MB95F478KPMC2-G-SNE2 MB95F562KPF-G-SNE2 MB95F564KPF-G-SNE2 MB95F634KPMC-G-SNE2 MB95F636KWQN-G-SNE1 MB95F696KPMC-G-SNE2 MB95F698KPMC1-G-SNE2 MB95F698KPMC2-G-SNE2 MB95F698KPMC-G-SNE2 MB95F818KPMC1-G-SNE2 MC908JK1ECDWER MC9S08PA32AVLD MC9S08PT60AVLD R5F1076CMSPV0 R5F5631ECDFBV0 C8051F389-B-GQ C8051F392-A-GMR ISD-ES1600_USB_PROG 901015X SC705C8AE0VFBE STM8TL53G4U6 PIC16F877-04/P-B R5F10Y17ASP#30 CY8C3MFIDOCK-125 403708R MB95F354EPF-G-SNE2 MB95F564KPFT-G-SNE2 MB95F564KWQN-G-SNE1 MB95F636KP-G-SH-SNE2 MB95F636KPMC-G-SNE2 MB95F694KPMC-G-SNE2 MB95F778JPMC1-G-SNE2 MB95F818KPMC-G-SNE2 MC908QY8CDWER MC9S08PT16AVLD MC9S08PT32AVLH MC9S08PT60AVLC MC9S08PT60AVLH C8051F500-IQR LC87F0G08AUJA-AH CP8361BT STM8S207C6T3 CG8421AF