NEC

User's Manual

V850ES/IE2

32-bit Single-Chip Microcontrollers

Hardware

μPD70F3713
μPD70F3714

Document No. U17716EJ2V0UD00 (2nd edition)
Date Published February 2008 N

Printed in Japan

[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.

- **The information in this document is current as of January, 2008. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".

  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

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

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

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

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

(Note)

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

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

M8E 02. 11-1

PREFACE

Readers

This manual is intended for users who wish to understand the functions of the V850ES/IE2 to design application systems using the V850ES/IE2.

Purpose

This manual is intended to give users an understanding of the hardware functions.

Organization

The V850ES/IE2 User's Manual is divided into two parts: Hardware (this manual) and Architecture (V850ES Architecture User's Manual). The organization of each manual is as follows:

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

How to Read This Manual

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

- To understand the overall functions of the V850ES/IE2
  →Read this manual according to the **CONTENTS**.

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

- How to interpret the register format
  →For a bit whose bit number is enclosed in angle brackets < >, its bit name is defined as a reserved word in the device file.

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

- To know the electrical specifications of the V850ES/IE2
  →See **CHAPTER 19 ELECTRICAL SPECIFICATIONS**.

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

The mark <R> shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what:" field.

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 V850ES/IE2

Document Name	Document No.
V850ES Architecture User's Manual	U15943E
V850ES/IE2 Hardware User's Manual	This manual

Documents related to development tools (user's manuals)

Document Name		Document No.
QB-V850ESIX2 (in-circuit emulator)		U17909E
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 Directive	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
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
PG-FP5 Flash Memory Programmer		U18865E

CONTENTS

CHAPTER 1 INTRODUCTION

The V850ES/IE2 is one of the low-power operation products in the NEC Electronics V850 Series of single-chip microcontrollers designed for real-time control applications.

1.1 General

The V850ES/IE2 is a 32-bit single-chip microcontroller that includes the V850ES CPU core and peripheral functions such as ROM/RAM, a timer/counter, serial interfaces, a watchdog timer, and an A/D converter.

In addition to high real-time response characteristics and 1-clock-pitch basic instructions, the V850ES/IE2 features 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 V850ES/IE2 enables an extremely high cost-performance for applications such as motor inverter control.

1.2 Features

○ Minimum instruction execution time:
50 ns (at internal 20 MHz operation)

○ General-purpose registers: 32 bits × 32

○ CPU features:
Signed multiplication (16 × 16 → 32): 1 to 2 clocks
Signed multiplication (32 × 32 → 64): 1 to 5 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
μPD70F3713	64 KB (flash memory)	6 KB
μPD70F3714	128 KB (flash memory)	6 KB

○ Interrupts/exceptions:
Non-maskable interrupts: 1 source (external: none, internal: 1)
Maskable interrupts: 42 sources (external: 7, internal: 35)
Software exceptions: 32 sources
Exception traps: 2 sources

○ I/O lines:
I/O ports: 39

○ Timer/counter function:
16-bit interval timer M (TMM): 1 channel
16-bit timer/event counter Q (TMQ): 2 channels
16-bit timer/event counter P (TMP): 4 channels
Motor control function (uses timer TMQ: 1 channel (TMQ1), TMP: 1 channel (TMP1))
16-bit accuracy 6-phase PWM function with dead time: 1 channel
High-impedance output control function
Timer tuning operation function
Arbitrary cycle setting function
Arbitrary dead-time setting function
Watchdog timer: 1 channel

○ Serial interfaces:
Asynchronous serial interface A (UARTA)
3-wire variable length serial I/O (CSIB)
CSIB: 1 channel
UARTA: 2 channels

○ A/D converter:
10-bit resolution A/D converters (A/D converters 0 and 1): 4 channels × 2 units

- Clock generator: 2.5 MHz resonator connectable (external clock input prohibited)
  Multiplication function by PLL clock synthesizer (fixed to multiplication by eight, f_{XX} = 20 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
- Self programming: Supported only in the μPD70F3714 (not supported in the μPD70F3713)
- Package: 64-pin plastic LQFP (14 × 14)
- Operation supply voltage: $V_{DD} = EV_{DD}$ = 3.5 to 5.5 V
  AV_{DD0}, AV_{DD1} = 4.5 to 5.5 V
- Operation ambient temperature:
  T_A = –40 to +85°C

1.3 Applications

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

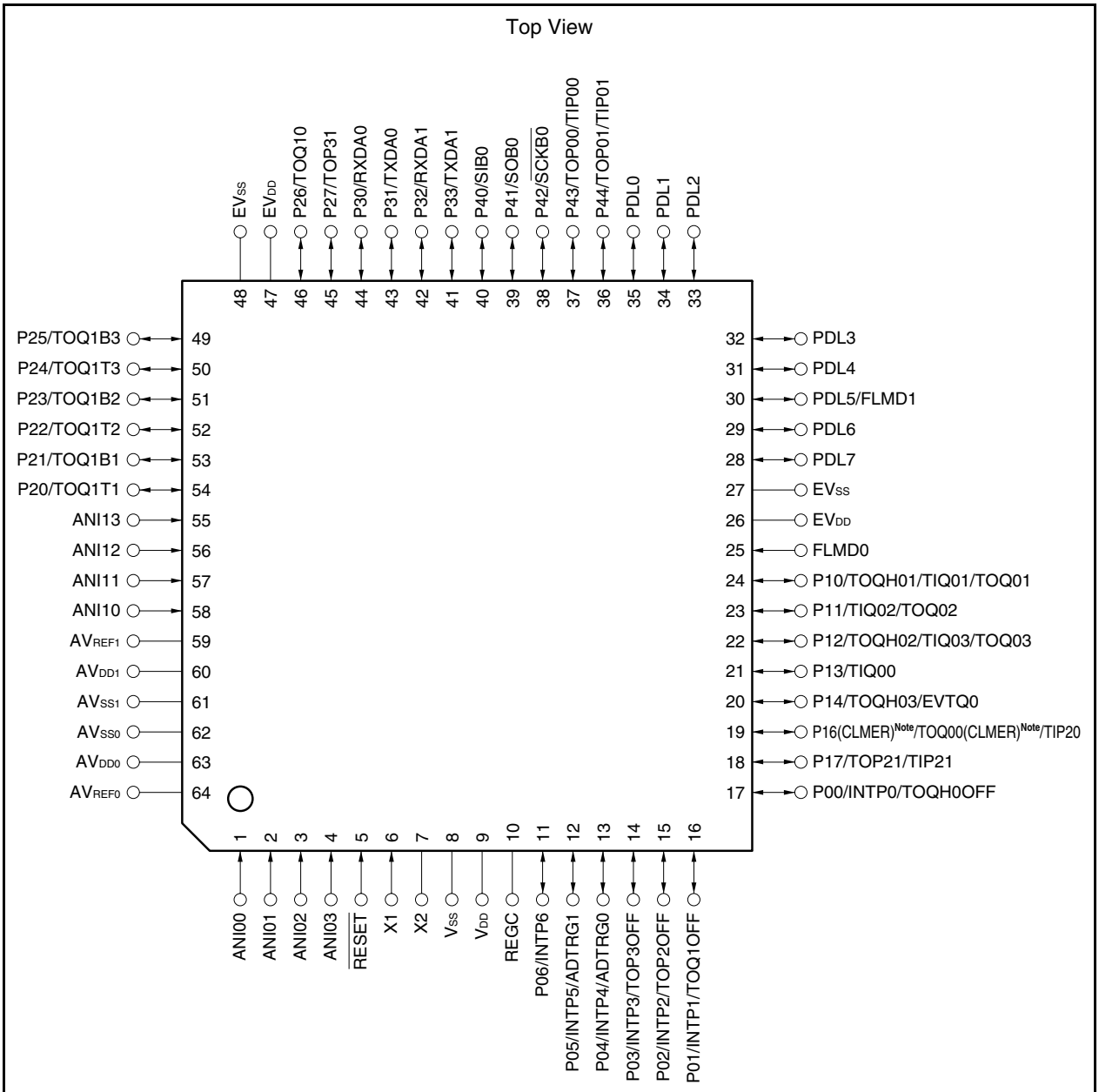
1.4 Ordering Information

Part Number	Package	Internal ROM
μPD70F3713GC-8BS-A	64-pin plastic LQFP (14 × 14)	Flash memory (64 KB)
μPD70F3714GC-8BS-A	64-pin plastic LQFP (14 × 14)	Flash memory (128 KB)

Remark Products with -A at the end of the part number are lead-free products.

1.5 Pin Configuration

- **64-pin plastic LQFP (14 × 14)**
  μPD70F3713GC-8BS-A
  μPD70F3714GC-8BS-A



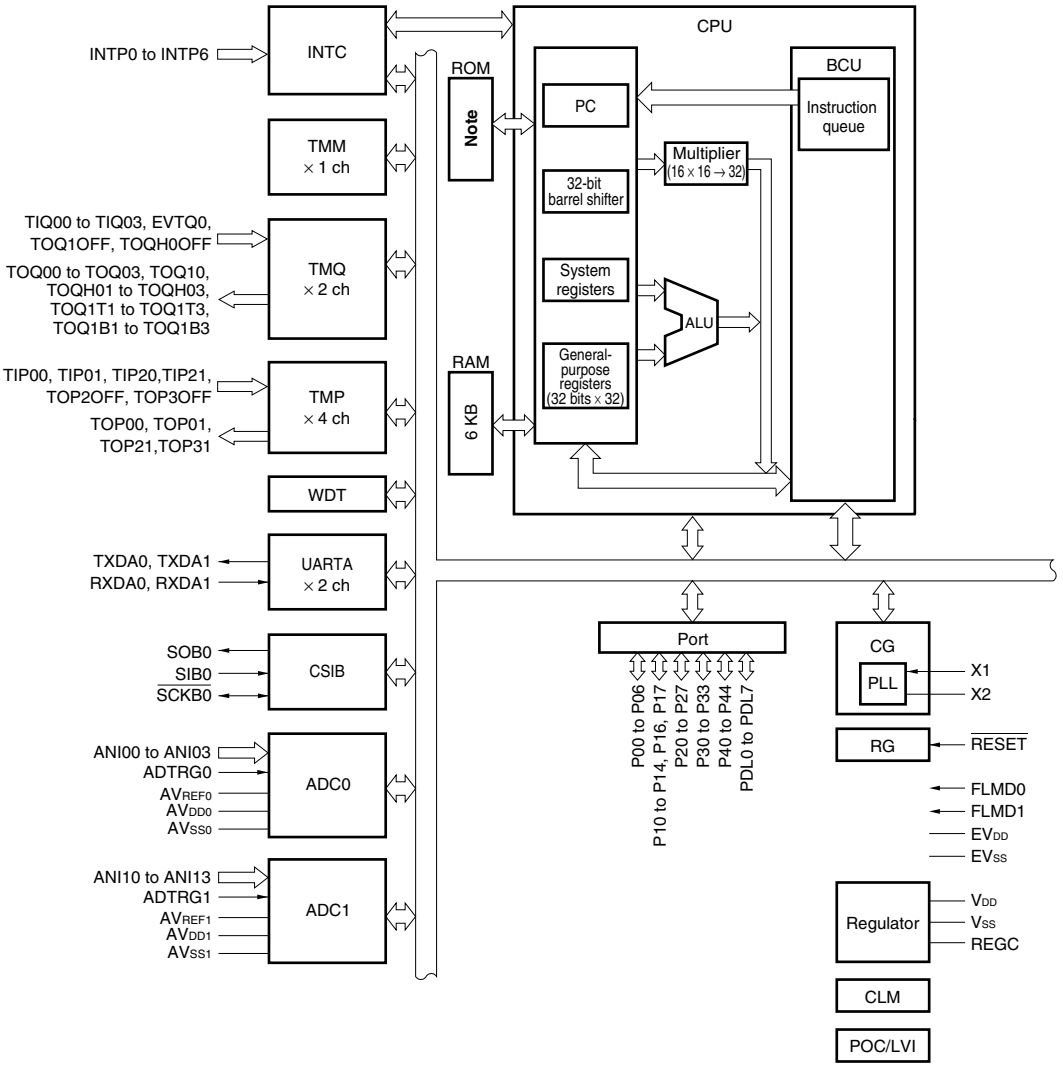
Note The CLMER signal is enabled only when P16 is specified as an output port or the output function of TOQ00. When an error (oscillator stop) is detected by the clock monitor, a low level is forcibly output. Low-level output is released by reset signal. For details, see **Table 4-5 Alternate-Function Pins of Port 1**.

Pin Identification

ADTRG0, ADTRG1:	A/D trigger input	$\overline{\text{SCKB0}}$:	Serial clock
ANI00 to ANI03,		SIB0:	Serial input
ANI10 to ANI13:	Analog input	SOB0:	Serial output
AV_{DD0}, AV_{DD1}:	Analog power supply	TIP00, TIP01,	
AV_{REF0}, AV_{REF1}:	Analog reference voltage	TIP20, TIP21,	
AV_{SS0}, AV_{SS1}:	Analog ground	TIQ00 to TIQ03:	Timer trigger input
EV_{DD}:	Power supply for port	TOP00, TOP01,	
EV_{SS}:	Ground for port	TOP21, TOP31,	
EVTQ0:	Timer event count input	TOQ1B1 to TOQ1B3,	
FLMD0, FLMD1:	Flash programming mode	TOQ1T1 to TOQ1T3,	
INTP0 to INTP6:	External interrupt input	TOQ00 to TOQ03,	
P00 to P06:	Port 0	TOQ10,	
P10 to P14, P16, P17:	Port 1	TOQH01 to TOQH03:	Timer output
P20 to P27:	Port 2	TOP2OFF, TOP3OFF,	
P30 to P33:	Port 3	TOQ1OFF, TOQH0OFF:	Timer output off
P40 to P44:	Port 4	TXDA0, TXDA1:	Transmit data
PDL0 to PDL7:	Port DL	V_{DD}:	Power supply
REGC:	Regulator control	V_{SS}:	Ground
$\overline{\text{RESET}}$:	Reset	X1, X2:	Clock oscillator pin
RXDA0, RXDA1:	Receive data		

1.6 Function Blocks

(1) Internal block diagram



Note μPD70F3713: 64 KB (flash memory)
μPD70F3714: 128 KB (flash memory)

(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 (16 bits × 16 bits → 32 bits) and a barrel shifter (32 bits), help accelerate complex processing.

(b) Bus control unit (BCU)

The BCU controls the internal bus.

(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
μPD70F3713	64 KB (flash memory)	xn000000H to xn00FFFFH
μPD70F3714	128 KB (flash memory)	xn000000H to xn01FFFFH

Remark n = xx11B

(d) RAM

This is a 6 KB internal RAM that is mapped to the addresses xnFFD800H to xnFFEFFFH.

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

Remark n = xx11B

(e) Interrupt controller (INTC)

This controller handles hardware interrupt requests (INTP0 to INTP6) 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

This unit incorporates one 16-bit interval timer M (TMM) channel, two 16-bit timer/event counter Q (TMQ) channels, and four 16-bit timer/event counter P (TMP) 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 V850ES/IE2 includes two asynchronous serial interface A (UARTA) channels and one 3-wire variable length serial I/O (CSIB) channel as the serial interface.

For UARTA, data is transferred via the TXDAn and RXDAn pins (n = 0, 1).

For CSIB, data is transferred via the SOB0, SIB0, and $\overline{\text{SCKB0}}$ pins.

(j) A/D converter (ADC)

The V850ES/IE2 includes two-channel 10-bit A/D converters (ADC0 and ADC1) with four analog input pins.

(k) Ports

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

Port	I/O	Alternate Function
Port 0	7-bit I/O	Timer/counter input, external interrupt input, external trigger input of A/D converter
Port 1	7-bit I/O	Timer/counter I/O
Port 2	8-bit I/O	Timer/counter output
Port 3	4-bit I/O	Serial interface I/O
Port 4	5-bit I/O	Serial interface I/O, timer/counter I/O
Port DL	8-bit I/O	–

CHAPTER 2 PIN FUNCTIONS

2.1 List of Pin Functions

The names and functions of the pins in the V850ES/IE2 are listed below. These pins can be divided into port pins and non-port pins according to their function.

There are two power supplies for the I/O buffer of a pin: power supply for A/D converter (AV_{DD0} and AV_{DD1}) and power supply for external pin (EV_{DD}). The relationship between each power supply and the pins is shown below.

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

Power Supply	Corresponding Pins
AV_{DD0}, AV_{DD1}	ANI00 to ANI03, ANI10 to ANI13
EV_{DD}	Ports 0 to 4, port DL, $\overline{\text{RESET}}$

(1) Port pins

(1/2)

Pin Name	Pin No.	I/O	Function	Alternate Function
P00	17	I/O	Port 0 7-bit I/O port Input data read/output data write is enabled in 1-bit units. Use of an on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected only in the input mode of the port mode and when the alternate function of the pin is used).	INTP0/TOQH0OFF
P01	16			INTP1/TOQ1OFF
P02	15			INTP2/TOP2OFF
P03	14			INTP3/TOP3OFF
P04	13			INTP4/ADTRG0
P05	12			INTP5/ADTRG1
P06	11			INTP6
P10	24	I/O	Port 1 7-bit I/O port Input data read/output data write is enabled in 1-bit units. Use of an on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected only in the input mode of the port mode, when the input mode of alternate function of the pin is used, and when TOP21 and TOQH01 to TOQH03 pins, which function as output pins when their alternate function is used, go into a high-impedance state).	TOQH01/TIQ01/TOQ01
P11	23			TIQ02/TOQ02
P12	22			TOQH02/TIQ03/TOQ03
P13	21			TIQ00
P14	20			TOQH03/EVTQ0
P16 (CLMER)[Note]	19			TOQ00 (CLMER)[Note]/TIP20
P17	18			TOP21/TIP21

Note The CLMER signal is enabled only when P16 is specified as an output port or the output function of TOQ00. When an error (oscillator stop) is detected by the clock monitor, a low level is forcibly output. Low-level output is released by reset signal. For details, see **Table 4-5 Alternate-Function Pins of Port 1**.

(2/2)

Pin Name	Pin No.	I/O	Function	Alternate Function
P20	54	I/O	Port 2 8-bit I/O port Input data read/output data write is enabled in 1-bit units. Use of an on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected only in the input mode of the port mode, or when TOQ1T1 to TOQ1T3 and TOQ1B1 to TOQ1B3 and TOP31 pins, which function as output pins when their alternate function is used, go into a high-impedance state).	TOQ1T1
P21	53			TOQ1B1
P22	52			TOQ1T2
P23	51			TOQ1B2
P24	50			TOQ1T3
P25	49			TOQ1B3
P26	46			TOQ10
P27	45			TOP31
P30	44	I/O	Port 3 4-bit I/O port Input data read/output data write is enabled in 1-bit units. Use of an on-chip pull-up resistor can be specified in 1-bit units (the on-chip pull-up resistor can be connected only in the input mode of the port mode and when the input mode of the alternate function of the pin is used).	RXDA0
P31	43			TXDA0
P32	42			RXDA1
P33	41			TXDA1
P40	40	I/O	Port 4 5-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 in the input mode of the port mode, including the case where the $\overline{\text{SCKB0}}$ pin in the slave mode, and when the alternate function of the pin is used in the input mode).	SIB0
P41	39			SOB0
P42	38			$\overline{\text{SCKB0}}$
P43	37			TOP00/TIP00
P44	36			TOP01/TIP01
PDL0	35	I/O	Port DL 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).	–
PDL1	34			–
PDL2	33			–
PDL3	32			–
PDL4	31			–
PDL5	30			FLMD1
PDL6	29			–
PDL7	28			–

(2) Non-port pins

(1/2)

Pin Name	Pin No.	I/O	Function	Alternate Function
ADTRG0	13	Input	External trigger input for A/D converters 0, 1	INTP4/P04
ADTRG1	12	Input		INTP5/P05
ANI00	1	Input	Analog input to A/D converters 0, 1	–
ANI01	2	Input		–
ANI02	3	Input		–
ANI03	4	Input		–
ANI10	58	Input		–
ANI11	57	Input		–
ANI12	56	Input		–
ANI13	55	Input		–
AV_{DD0}	63	–	Positive power supply for A/D converters 0, 1 (same potential as V_{DD})	–
AV_{DD1}	60	–		–
AV_{REF0}	64	–	Reference voltage input for A/D converters 0, 1 (same potential as AV_{DD0} and AV_{DD1})	–
AV_{REF1}	59	–		–
AV_{SS0}	62	–	Ground potential for A/D converters 0, 1 (same potential as V_{SS})	–
AV_{SS1}	61	–		–
EV_{DD}	26, 47	–	Positive power supply for external pin	–
EV_{SS}	27, 48	–	Ground potential for external pin	–
EVTQ0	20	Input	External event count input of TMQ0	TOQH03/P14
FLMD0	25	Input	Pin for setting flash memory programming mode	–
FLMD1	30	Input		PDL5
INTP0	17	Input	External maskable interrupt request input	TOQH0OFF/P00
INTP1	16			TOQ1OFF/P01
INTP2	15			TOP2OFF/P02
INTP3	14			TOP3OFF/P03
INTP4	13			ADTRG0/P04
INTP5	12			ADTRG1/P05
INTP6	11			P06
REGC	10	–	Regulator output stabilization capacitance connection	–
$\overline{\text{RESET}}$	5	Input	System reset input	–
RXDA0	44	Input	Serial receive data input of UARTA0, UARTA1	P30
RXDA1	42			P32
$\overline{\text{SCKB0}}$	38	I/O	Serial clock I/O of CSIB0	P42
SIB0	40	Input	Serial receive data input of CSIB0	P40
SOB0	39	Output	Serial transmit data output of CSIB0	P41

(2/2)

Pin Name	Pin No.	I/O	Function	Alternate Function
TIP00	37	Input	External event count input/external trigger input/capture trigger input of TMP0	TOP00/P43
TIP01	36		Capture trigger input of TMP0	TOP01/P44
TIP20	19		External event count input/external trigger input/capture trigger input of TMP2	TOQ00 (CLMER)[Note]/ P16 (CLMER)[Note]
TIP21	18		Capture trigger input of TMP2	TOP21/P17
TIQ00	21	Input	Capture trigger input of TMQ0	P13
TIQ01	24			TOQH01/TOQ01/P10
TIQ02	23			TOQ02/P11
TIQ03	22			TOQH02/TOQ03/P12
TOP00	37	Output	Pulse signal output of TMP0, TMP2	TIP00/P43
TOP01	36			TIP01/P44
TOP21	18			TIP21/P17
TOP2OFF	15	Input	High-impedance output control signal input	INTP2/P02
TOP31	45	Output	Pulse signal output of TMP3	P27
TOP3OFF	14	Input	High-impedance output control signal input	INTP3/P03
TOQ00 (CLMER)[Note]	19	Output	Pulse signal output of TMQ0	TIP20/P16 (CLMER)[Note]
TOQ01	24			TOQH01/TIQ01/P10
TOQ02	23			TIQ02/P11
TOQ03	22			TOQH02/TIQ03/P12
TOQ10	46	Output	Pulse signal output of TMQ1	P26
TOQ1B1	53	Output	Pulse signal output for 6-phase PWM	P21
TOQ1B2	51			P23
TOQ1B3	49			P25
TOQ1OFF	16	Input	High-impedance output control signal input	INTP1/P01
TOQ1T1	54	Output	Pulse signal output for 6-phase PWM	P20
TOQ1T2	52			P22
TOQ1T3	50			P24
TOQH01	24	Output	High-impedance output by TMQ0 pulse signal output and valid edge of TOQH0OFF pin input	TIQ01/TOQ01/P10
TOQH02	22			TIQ03/TOQ03/P12
TOQH03	20			EVTQ0/P14
TOQH0OFF	17	Input	High-impedance output control signal input	INTP0/P00
TXDA0	43	Output	Serial transmit data output of UARTA0, UARTA1	P31
TXDA1	41			P33
V_{DD}	9	–	Positive power supply for internal unit	–
V_{SS}	8	–	Ground potential for internal unit	–
X1	6	Input	Resonator connection pin for system clock	–
X2	7	–		–

Note The CLMER signal is enabled only when P16 is specified as an output port or the output function of TOQ00. When an error (oscillator stop) is detected by the clock monitor, a low level is forcibly output. Low-level output is released by reset signal. For details, see **Table 4-5 Alternate-Function Pins of Port 1**.

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

(1/2)

Pin Name	Alternate-Function Pin Name	Pin No.	I/O Circuit Type	Recommended Connection
P00	INTP0/TOQH0OFF	17	8-P	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P01	INTP1/TOQ1OFF	16		
P02	INTP2/TOP2OFF	15		
P03	INTP3/TOP3OFF	14		
P04	INTP4/ADTRG0	13		
P05	INTP5/ADTRG1	12		
P06	INTP6	11		
P10	TOQH01/TIQ01/TOQ01	24		
P11	TIQ02/TOQ02	23		
P12	TOQH02/TIQ03/TOQ03	22		
P13	TIQ00	21		
P14	TOQH03/EVTQ0	20		
P16 (CLMER)[Note]	TOQ00 (CLMER)[Note]/TIP20	19		
P17	TOP21/TIP21	18		
P20	TOQ1T1	54	5-AG	
P21	TOQ1B1	53		
P22	TOQ1T2	52		
P23	TOQ1B2	51		
P24	TOQ1T3	50		
P25	TOQ1B3	49		
P26	TOQ10	46		
P27	TOP31	45		
P30	RXDA0	44	8-P	
P31	TXDA0	43	5-AG	
P32	RXDA1	42	8-P	
P33	TXDA1	41	5-AG	
P40	SIB0	40	8-P	
P41	SOB0	39	5-AG	
P42	$\overline{\text{SCKB0}}$	38	8-P	
P43	TOP00/TIP00	37		
P44	TOP01/TIP01	36		

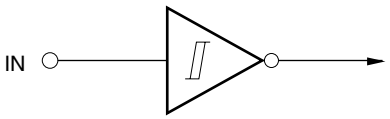
Note The CLMER signal is enabled only when P16 is specified as an output port or the output function of TOQ00. When an error (oscillator stop) is detected by the clock monitor, a low level is forcibly output. Low-level output is released by reset signal. For details, see **Table 4-5 Alternate-Function Pins of Port 1**.

(2/2)

Pin Name	Alternate-Function Pin Name	Pin No.	I/O Circuit Type	Recommended Connection
PDL0	–	35	5-AG	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
PDL1	–	34		
PDL2	–	33		
PDL3	–	32		
PDL4	–	31		
PDL5	FLMD1	30		
PDL6	–	29		
PDL7	–	28		
ANI00	–	1	7	Independently connect to AV_{DD0}, AV_{DD1}, AV_{SS0}, or AV_{SS1} via a resistor.
ANI01	–	2		
ANI02	–	3		
ANI03	–	4		
ANI10	–	58		
ANI11	–	57		
ANI12	–	56		
ANI13	–	55		
$\overline{\text{RESET}}$	–	5	2	–
FLMD0	–	25		–

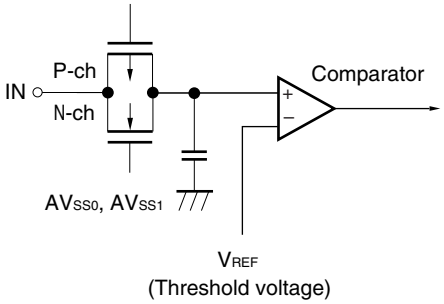
2.3 Pin I/O Circuits

Type 2

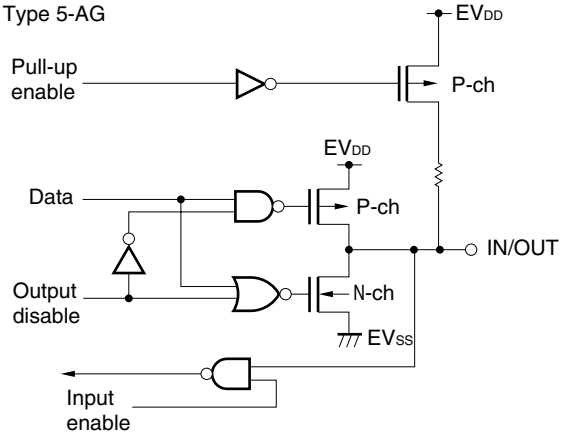


Schmitt-triggered input with hysteresis characteristics

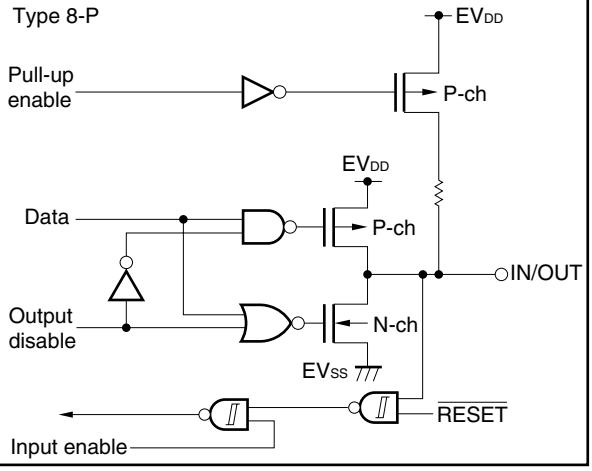
Type 7



Type 5-AG



Type 8-P



CHAPTER 3 CPU FUNCTION

The CPU of the V850ES/IE2 is based on the RISC architecture and executes most instructions in one clock cycle by using 5-stage pipeline control.

3.1 Features

- Minimum instruction execution time: 50 ns
  (@ 20 MHz operation: 4.5 to 5.5 V (when using A/D converter),
  3.5 to 5.5 V (when not using A/D converter))
- Memory space Program (physical address) space: 64 MB linear
  Data (logical address) space: 4 GB linear
- General-purpose registers: 32 bits × 32
- Internal 32-bit architecture
- 5-stage pipeline control
- Multiply/divide instructions
- Saturated operation instructions
- 32-bit shift instruction: 1 clock
- Load/store instruction with long/short format
- Four types of bit manipulation instructions
  - SET1
  - CLR1
  - NOT1
  - TST1

3.2 CPU Register Set

The CPU registers of the V850ES/IE2 can be classified into two categories: a general-purpose program register set and a dedicated system register set. All the registers have 32-bit width.

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

(1) Program register set

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

31		0
PC	(Program counter)	

(2) System register set

31		0
EIPC	(Interrupt status saving register)	
EIPSW	(Interrupt status saving register)	
FEPC	(NMI status saving register)	
FEPSW	(NMI status saving register)	
ECR	(Interrupt source register)	
PSW	(Program status word)	
CTPC	(CALLT execution status saving register)	
CTPSW	(CALLT execution status saving register)	
DBPC	(Exception/debug trap status saving register)	
DBPSW	(Exception/debug trap status saving register)	
CTBP	(CALLT base pointer)	

3.2.1 Program register set

The program 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. All 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 always holds 0 and is used for operations that use 0 and offset 0 addressing. r30 is used as a base pointer when performing memory access with the SLD and SST instructions.

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, and they must be restored to the registers after the registers have been used. There are cases when r2 is used by the real-time OS. If r2 is not used by the real-time OS, r2 can be used as a variable register.

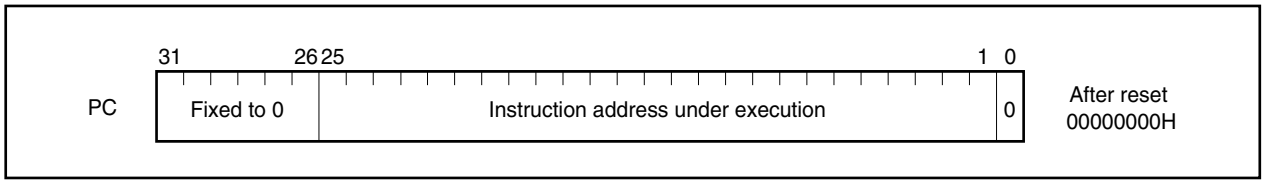
Table 3-1. General-Purpose Registers

Name	Usage	Operation
r0	Zero register	Always holds 0
r1	Assembler-reserved register	Working register for generating 32-bit immediate
r2	Address/data variable register (when r2 is not used by the real-time OS to be used)	
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 (area for placing program code)
r6 to r29	Address/data variable register	
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 address of the instruction under 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 bit 26, it is ignored.

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



3.2.2 System register set

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

Read from and write to system registers are performed by setting the system register numbers shown below with the system register load/store instructions (LDSR, STSR instructions).

Table 3-2. System Register Numbers

System Register No.	System Register Name	Operand Specification Enabled	
		LDSR Instruction	STSR Instruction
0	Interrupt status saving register (EIPC)[Note 1]	Yes	Yes
1	Interrupt status saving register (EIPSW)[Note 1]	Yes	Yes
2	NMI status saving register (FEPC)	Yes	Yes
3	NMI status saving register (FEPSW)	Yes	Yes
4	Interrupt source register (ECR)	No	Yes
5	Program status word (PSW)	Yes	Yes
6 to 15	Reserved numbers for future function expansion (The operation is not guaranteed if accessed.)	No	No
16	CALLT execution status saving register (CTPC)	Yes	Yes
17	CALLT execution status saving register (CTPSW)	Yes	Yes
18	Exception/debug trap status saving register (DBPC)	Yes[Note 2]	Yes[Note 2]
19	Exception/debug trap status saving register (DBPSW)	Yes[Note 2]	Yes[Note 2]
20	CALLT base pointer (CTBP)	Yes	Yes
21 to 31	Reserved numbers for future function expansion (The operation is not guaranteed if accessed.)	No	No

Notes 1. Since only one set of these registers is available, the contents of this register must be saved by the program when multiple interrupt servicing is enabled.

2. Can be accessed only after the DBTRAP instruction or illegal opcode is executed and before the DBRET instruction is executed.

Caution Even if bit 0 of EIPC, FEPC, or CTPC is set (1) by the LDSR instruction, bit 0 is ignored during return with the RETI instruction following interrupt servicing (because bit 0 of PC is fixed to 0). When setting a value to EIPC, FEPC, and CTPC, set an even number (bit 0 = 0).

(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)).

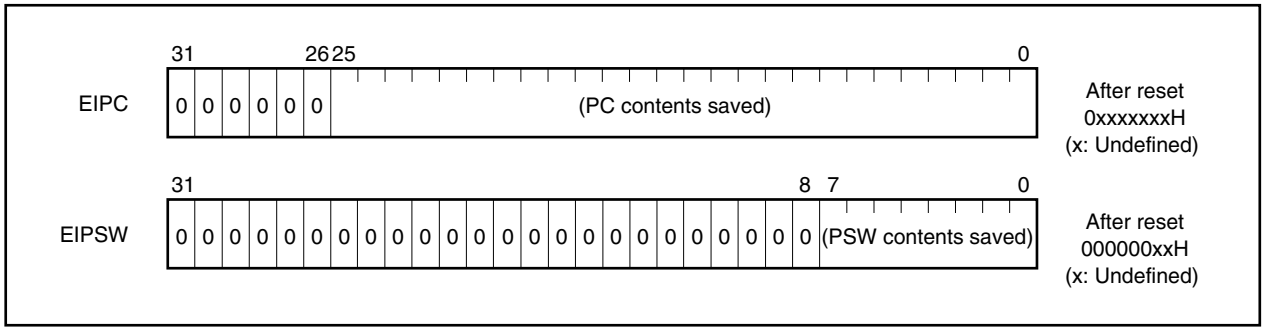
The address of the next instruction following the instruction executed when a software exception or maskable interrupt occurs is saved to EIPC, except for some instructions (see **14.9 Periods in Which CPU Does Not Acknowledge Interrupts**).

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.

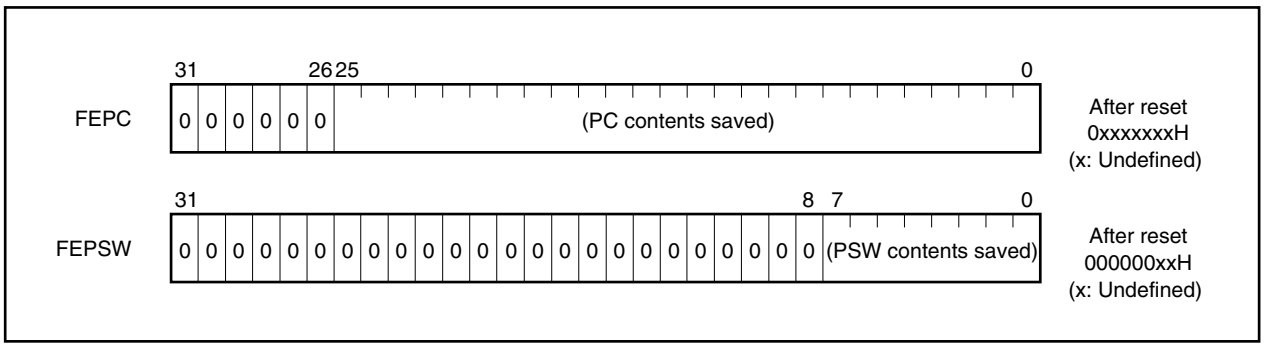
Upon occurrence of a non-maskable interrupt (NMI), the contents of the program counter (PC) are saved to FEPC and the contents of the program status word (PSW) are saved to FEPSW.

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.

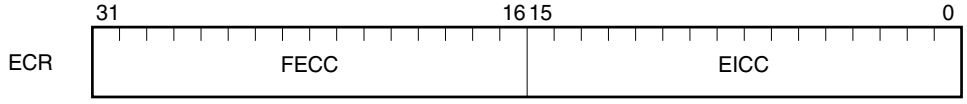
Bits 31 to 26 of FEPC and bits 31 to 8 of FEPSW are reserved (fixed to 0) for future function expansion.

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

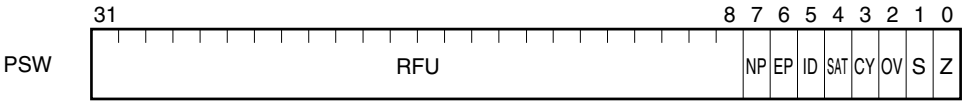
(4) Program status word (PSW)

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)

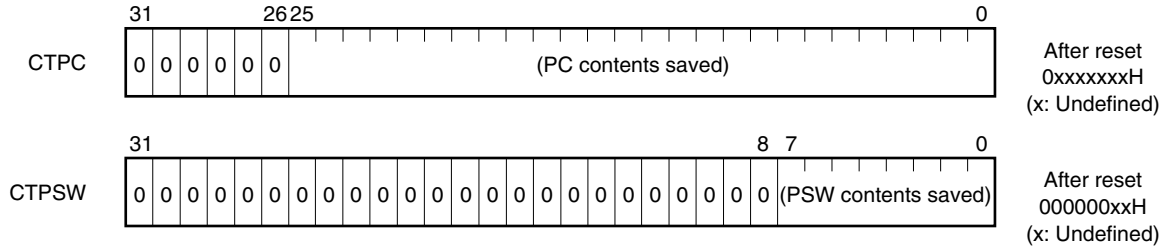
There are two CALLT execution status saving registers, CTPC and CTPSW.

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.

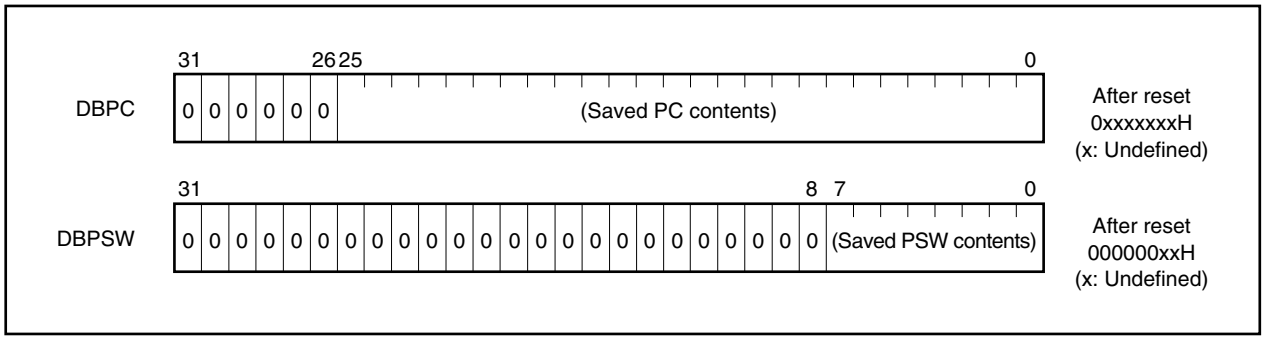
The contents saved to DBPC consist of the address of the next instruction after the instruction executed when an exception trap or debug trap occurs.

The current PSW contents are saved to DBPSW.

<R> 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.

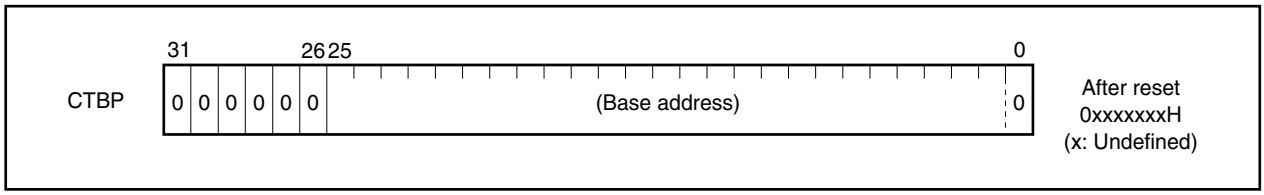
When the DBRET instruction has been executed, the values of DBPC and DBPSW are restored to the PC and PSW, respectively.



(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

The V850ES/IE2 has the following operating modes.

(1) Normal operating mode

After the system has been released from the reset state, the pins related to the bus interface are set to the port mode, execution branches to the reset entry address of the internal ROM, and instruction processing is started.

(2) Flash memory programming mode

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

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

Fix the specifications of these pins in the application system, and do not change then during operation.

FLMD0	FLMD1	Operating Mode
L	×	Normal operating mode
H	L	Flash memory programming mode
H	H	Setting prohibited

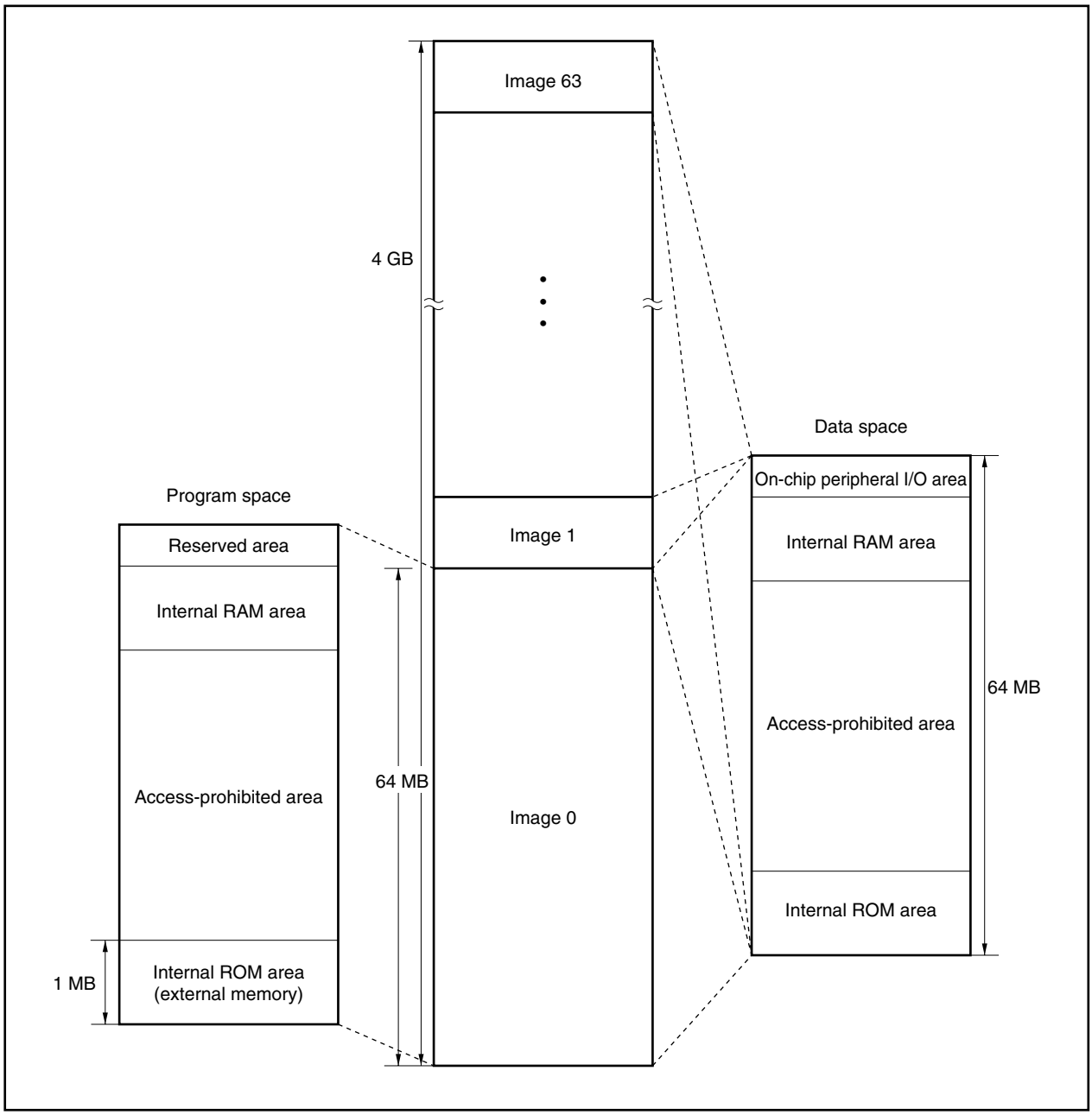
H: High level
L: Low level
×: Don't care

3.4 Address Space

3.4.1 CPU address space

For instruction addressing, an internal ROM area of up to 1 MB, and an internal RAM area are supported in a linear address space (program space) of up to 64 MB. For operand addressing (data access), up to 4 GB of a linear address space (data space) is supported. The 4 GB address space, however, is viewed as 64 images of a 64 MB physical address space. This means that the same 64 MB physical address space is accessed regardless of the value of bits 31 to 26.

Figure 3-1. Address Space Image



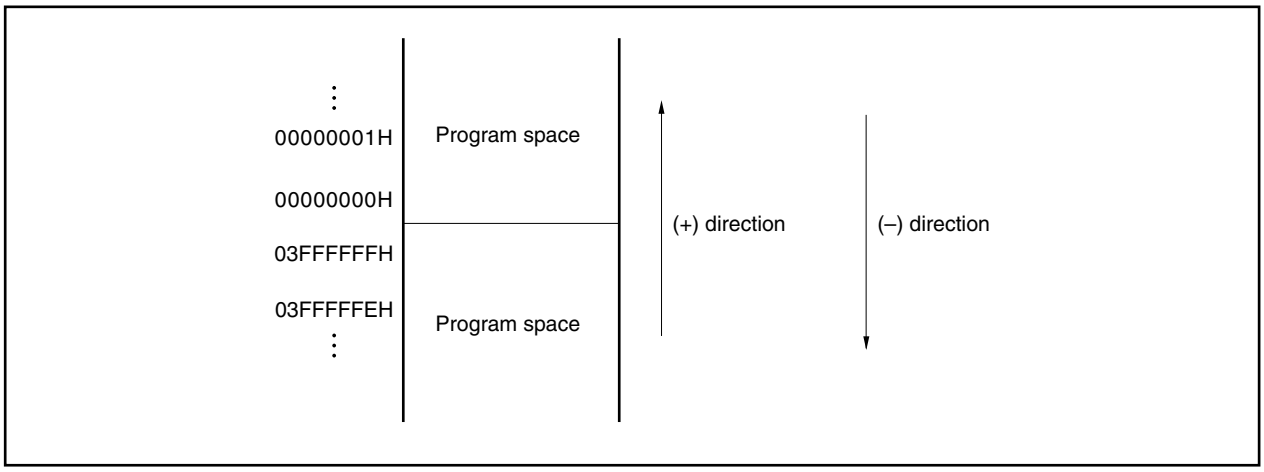
3.4.2 Wraparound of CPU address space

(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. Even if a carry or borrow occurs from bit 25 to bit 26 as a result of branch address calculation, the higher 6 bits ignore this and remain 0.

Therefore, the upper-limit address of the program space, 03FFFFFFH, and the lower-limit address, 00000000H, are contiguous addresses, and the program space is wrapped around at the boundary of these addresses.

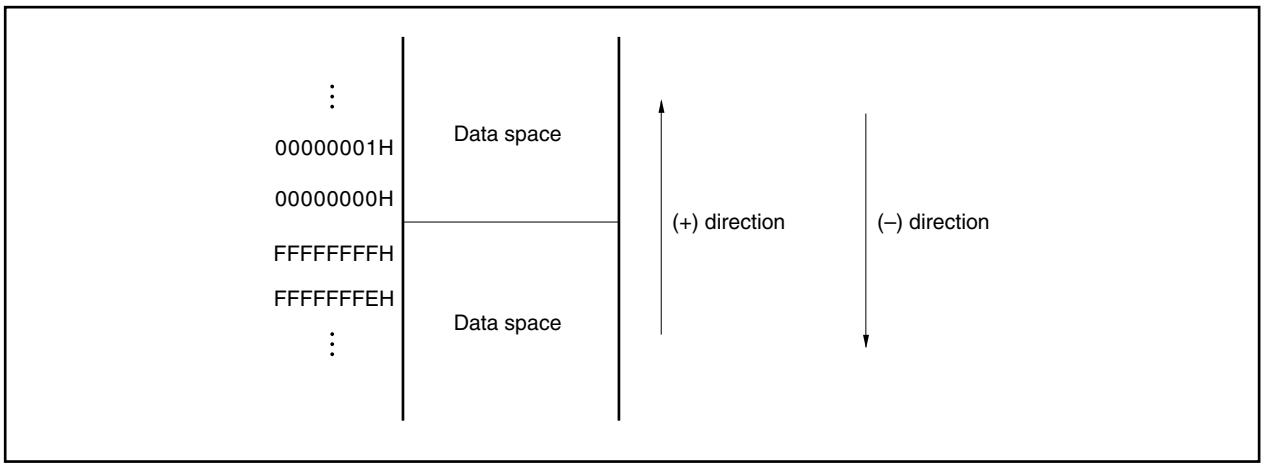
Caution No instructions can be fetched from the 4 KB area of 03FFF000H to 03FFFFFFH because this area is an on-chip peripheral I/O area. Therefore, do not execute any branch operation instructions in which the destination address will reside in any part of this area.



(2) Data space

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

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



3.4.3 Memory map

The V850ES/IE2 has reserved areas as shown below.

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

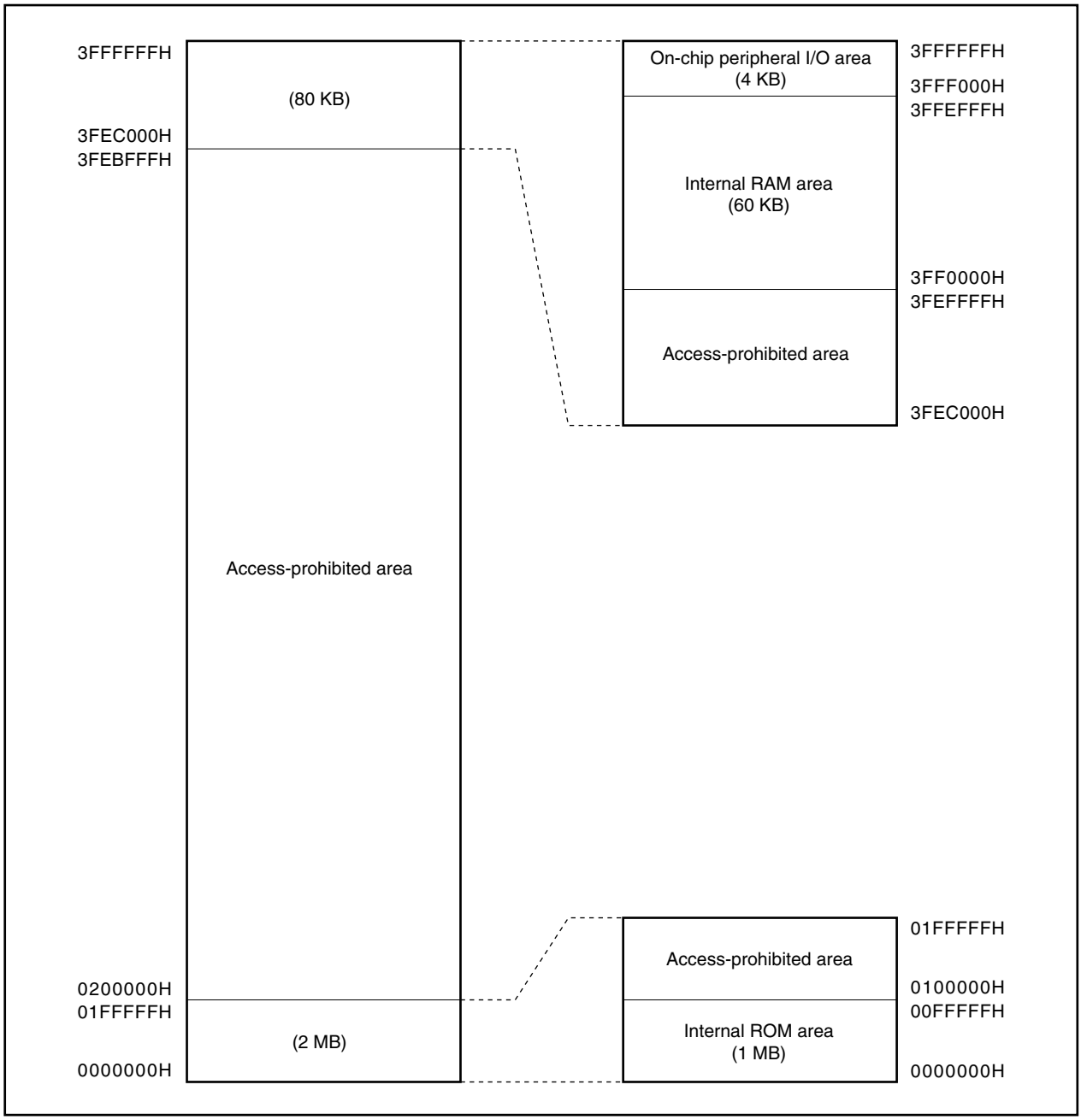
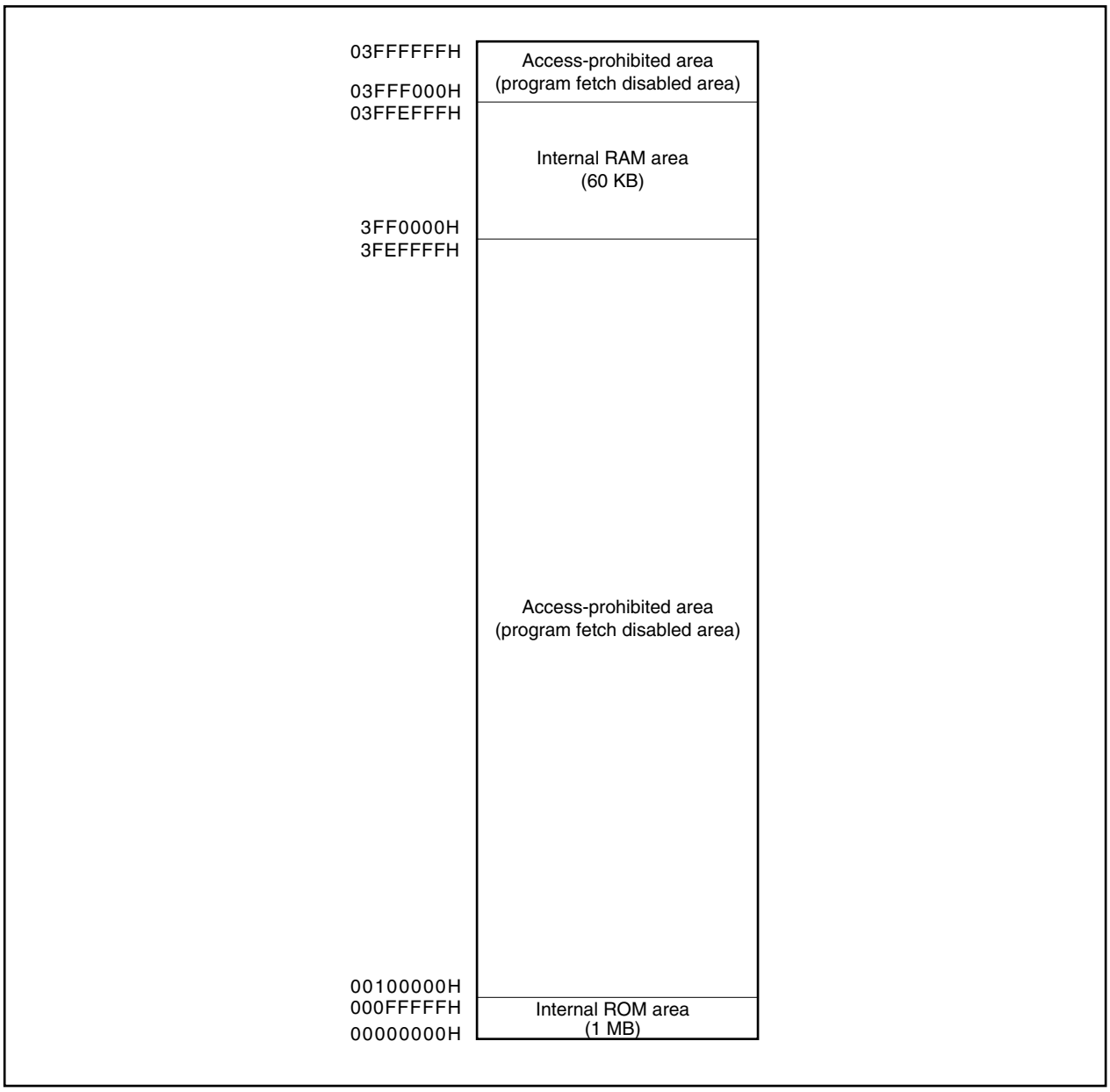


Figure 3-3. Program Memory Map



3.4.4 Areas

(1) Internal ROM area

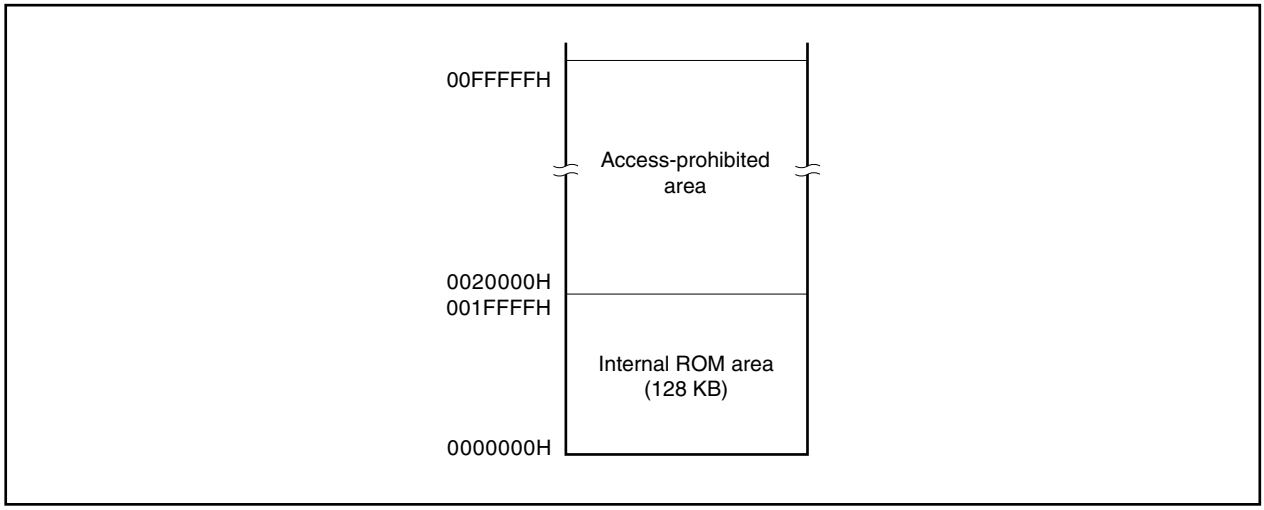
An area of 1 MB from 0000000H to 00FFFFFH is reserved for the internal ROM area.

(a) Internal ROM (128 KB)

A 128 KB area from 0000000H to 001FFFFH is provided in the μPD70F3714.

Addresses 0020000H to 00FFFFFH are an access-prohibited area.

Figure 3-4. Internal ROM Area (128 KB)

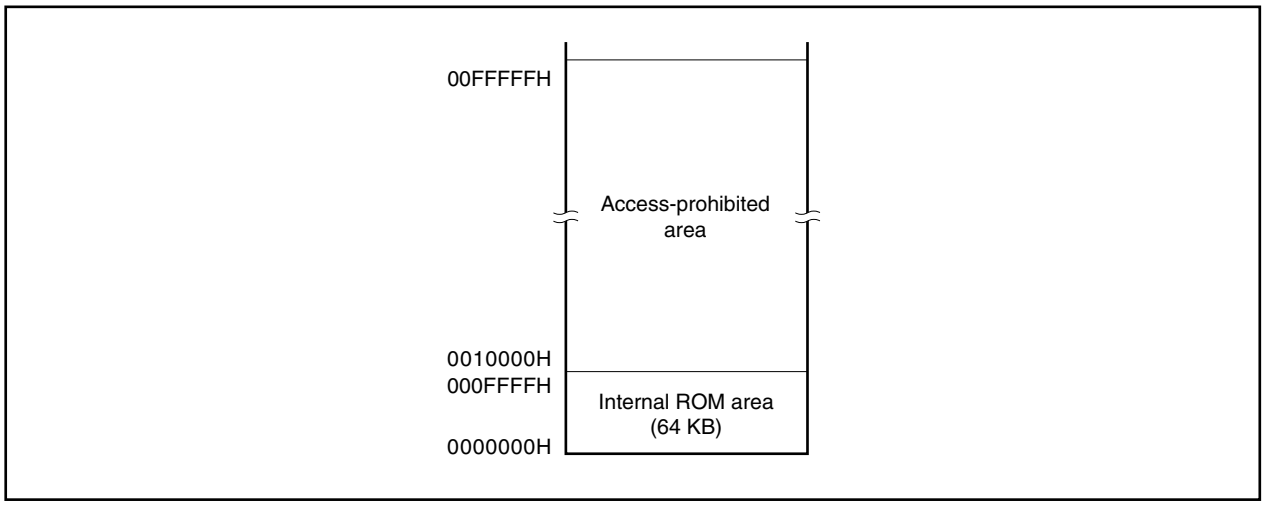


(b) Internal ROM (64 KB)

A 64 KB area from 000000H to 000FFFFH is provided in the μPD70F3713.

Addresses 0010000 to 00FFFFFH are an access-prohibited area.

Figure 3-5. Internal ROM Area (64 KB)



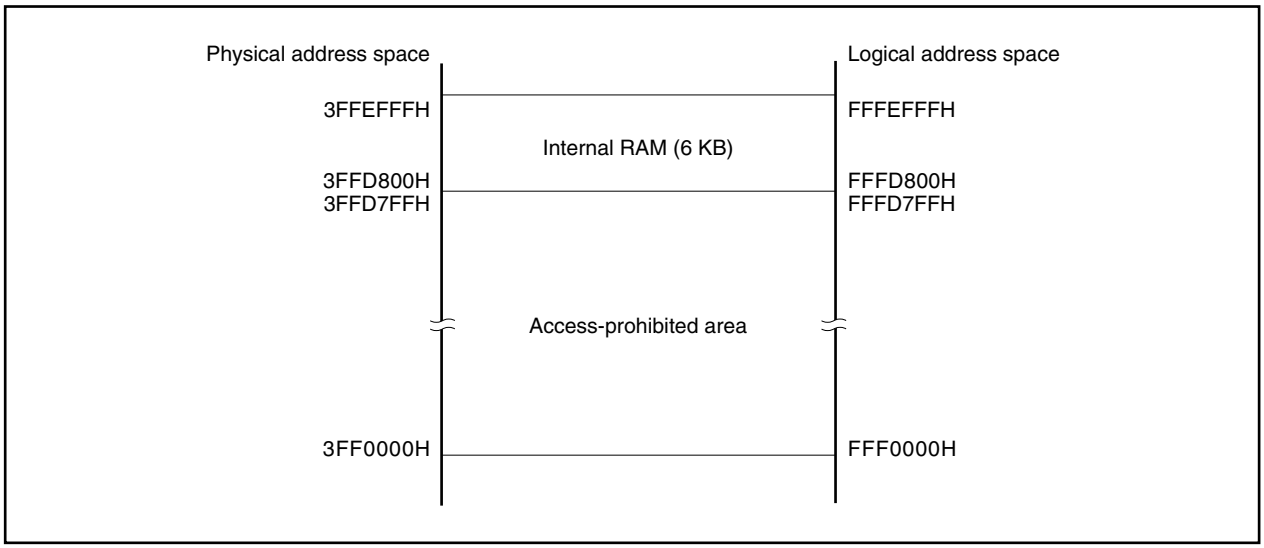
(2) Internal RAM area

An area of 60 KB maximum from 3FF0000H to 3FFEFFFH is reserved for the internal RAM area.

A 6 KB area from 3FFD800H to 3FFEFFFH is provided as physical internal RAM for the V850ES/IE2.

Addresses 3FF0000H to 3FFD7FFH are an access-prohibited area.

Figure 3-6. Internal RAM Area (6 KB)



(c) Internal memory size setting register (IMS)

The IMS register is used to set the internal RAM size of the V850ES/IE2.

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

Reset sets this register to 00H.

Cautions 1. Write the IMS register before the internal RAM is accessed. This register can be written only once after reset has been released.

2. Be sure to write 01H to the IMS register.

3. The sample startup routine supplied with the CA850 includes a code that clears the internal RAM area to 0. Therefore, setting the IMS register is required before the zero-clear routine is executed.

When using the sample startup routine, add instructions <2> to <5> shown in [Description example] below immediately after the __START label in the startup routine.

"0x11" of instruction <2> is the set value of the VSWC register and "0x01" of instruction <4> is the set value of the IMS register.

[Description example]

```
<1>_ _START:
<2>mov     0x11,      r13
<3>st.b    r13,       VSWC
<4>mov     0x01,      r12
<5>st.b    r12,       IMS
<6>mov     #_tp_TEXT, tp
           :
           :
```

(Add)

Remark When using a partner tool, make the setting in accordance with the contents of **Cautions 1** to **3.** Moreover, describe as follows to define the IMS register.

```
#define  IMS  (*((volatile unsigned char *)0xfffff9f0))
```

After reset: 00H W Address: FFFFF9F0H

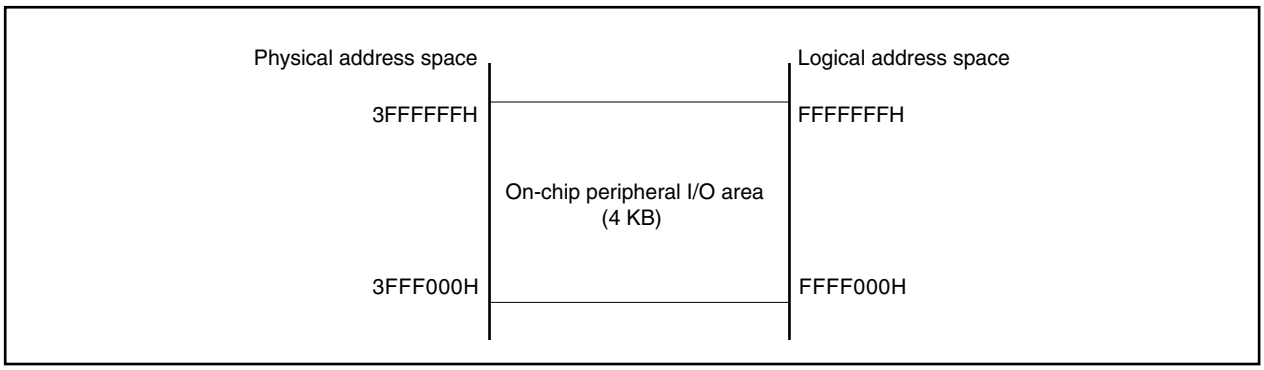
	7	6	5	4	3	2	1	0
IMS	0	0	0	0	0	0	0	RAM0

RAM0	Specification of internal RAM size
0	Undefined
1	6 KB (3FFD800H to 3FFEFFFH (RAM size of V850ES/IE2))

(3) On-chip peripheral I/O area

A 4 KB area from 3FFF000H to 3FFFFFFH is reserved as the on-chip peripheral I/O area.

Figure 3-7. On-Chip Peripheral I/O Area



On-chip peripheral I/O registers assigned with functions such as on-chip peripheral I/O operation mode specification and state monitoring are mapped to the on-chip peripheral I/O area. Program fetches are not allowed in this area.

Cautions 1. If word access of a register is attempted, halfword access to the word area is performed twice, first for the lower bits, then for the higher bits, ignoring the lower 2 address bits.

2. If a register that can be accessed in byte units is accessed in halfword units, the higher 8 bits become undefined if the access is a read operation. If a write access is performed, only the data in the lower 8 bits is written to the register.

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.

3.4.5 Recommended use of address space

The architecture of the V850ES/IE2 requires that a register that serves as a pointer be secured for address generation when operand data in the data space is accessed. The ±32 KB area of addresses stored in this pointer can be directly accessed by an instruction for operand data. Because the number of general-purpose registers that can be used as a pointer is limited, however, by keeping the performance from dropping during address calculation when a pointer value is changed, as many general-purpose registers as possible can be secured for variables, and the program size can be reduced.

(1) Program space

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

To use the internal RAM area as the program space, access addresses 3FFD800H to 3FFEFFFH (6 KB).

Caution When a branch instruction is positioned at the upper limit of the internal RAM area, a prefetch operation (invalid fetch) that will be located in the on-chip peripheral I/O area is not generated.

(2) Data space

With the V850ES/IE2 it seems that there are sixty-four 64 MB address spaces on the 4 GB CPU address space. Therefore, the least significant bit (bit 25) of a 26-bit address is sign-extended to 32 bits and allocated as an address.

(a) Application example of wraparound

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

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

Example: μPD70F3714

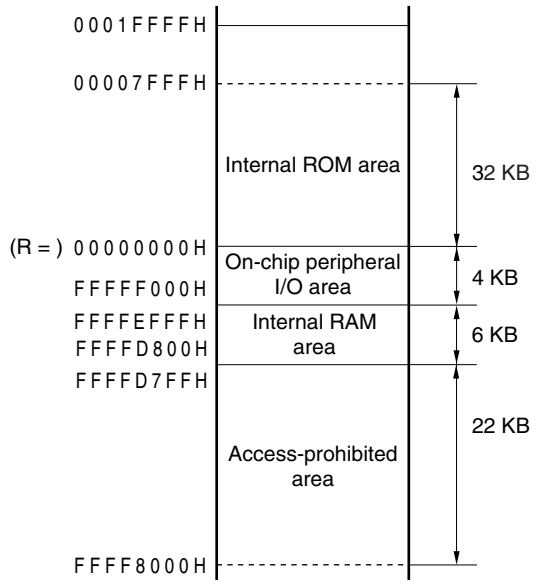
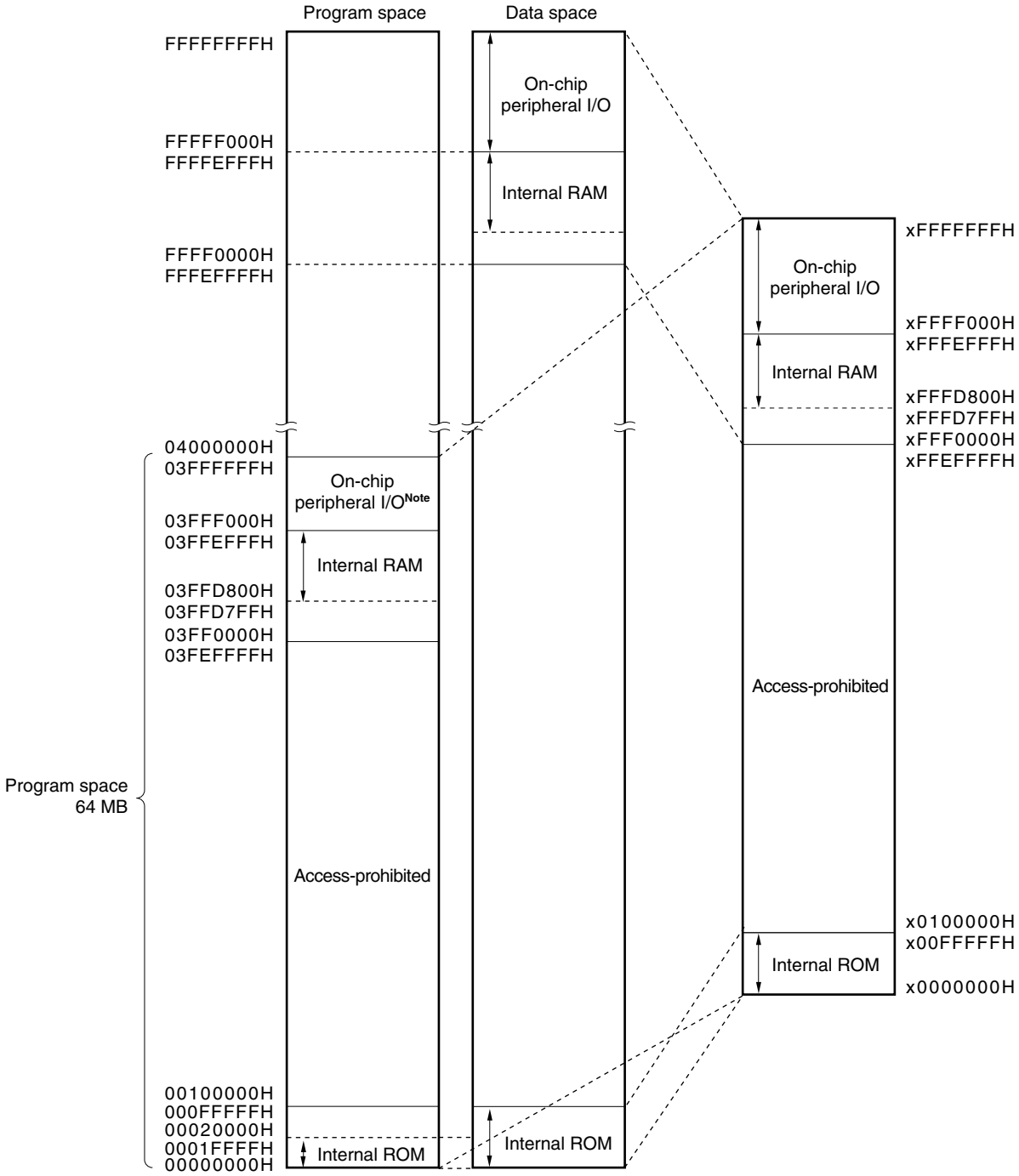


Figure 3-8. Recommended Memory Map



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

Remarks 1. ↕ indicates the recommended area.

2. This figure is the recommended memory map of the μPD70F3714.

3.4.6 On-chip peripheral I/O registers

(1/6)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF004H	Port DL register L	PDLL	R/W	√	√		Undefined
FFFFF024H	Port DL mode register L	PMDL		√	√		FFH
FFFFF06EH	System wait control register	VSWC			√		77H
FFFFF100H	Internal mask register 0	IMR0				√	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
FFFFF110H	Interrupt control register	PIC0		√	√		47H
FFFFF112H	Interrupt control register	PIC1		√	√		47H
FFFFF114H	Interrupt control register	PIC2		√	√		47H
FFFFF116H	Interrupt control register	PIC3		√	√		47H
FFFFF118H	Interrupt control register	PIC4		√	√		47H
FFFFF11AH	Interrupt control register	PIC5		√	√		47H
FFFFF11CH	Interrupt control register	PIC6		√	√		47H
FFFFF11EH	Interrupt control register	LVIIC		√	√		47H
FFFFF124H	Interrupt control register	TQ0OVIC		√	√		47H
FFFFF126H	Interrupt control register	TQ0CCIC0		√	√		47H
FFFFF128H	Interrupt control register	TQ0CCIC1		√	√		47H
FFFFF12AH	Interrupt control register	TQ0CCIC2		√	√		47H
FFFFF12CH	Interrupt control register	TQ0CCIC3		√	√		47H
FFFFF12EH	Interrupt control register	TQ1OVIC		√	√		47H
FFFFF130H	Interrupt control register	TQ1CCIC0		√	√		47H
FFFFF132H	Interrupt control register	TQ1CCIC1		√	√		47H
FFFFF134H	Interrupt control register	TQ1CCIC2		√	√		47H
FFFFF136H	Interrupt control register	TQ1CCIC3		√	√		47H
FFFFF148H	Interrupt control register	TP0OVIC		√	√		47H
FFFFF14AH	Interrupt control register	TP0CCIC0		√	√		47H
FFFFF14CH	Interrupt control register	TP0CCIC1		√	√		47H
FFFFF14EH	Interrupt control register	TP1OVIC		√	√		47H
FFFFF150H	Interrupt control register	TP1CCIC0		√	√		47H
FFFFF152H	Interrupt control register	TP1CCIC1		√	√		47H
FFFFF154H	Interrupt control register	TP2OVIC		√	√		47H
FFFFF156H	Interrupt control register	TP2CCIC0		√	√		47H

(2/6)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF158H	Interrupt control register	TP2CCIC1	R/W	√	√		47H
FFFFF15AH	Interrupt control register	TP3OVIC		√	√		47H
FFFFF15CH	Interrupt control register	TP3CCIC0		√	√		47H
FFFFF15EH	Interrupt control register	TP3CCIC1		√	√		47H
FFFFF168H	Interrupt control register	UA0REIC		√	√		47H
FFFFF16AH	Interrupt control register	UA0RIC		√	√		47H
FFFFF16CH	Interrupt control register	UA0TIC		√	√		47H
FFFFF16EH	Interrupt control register	CB0REIC		√	√		47H
FFFFF170H	Interrupt control register	CB0RIC		√	√		47H
FFFFF172H	Interrupt control register	CB0TIC		√	√		47H
FFFFF174H	Interrupt control register	UA1REIC		√	√		47H
FFFFF176H	Interrupt control register	UA1RIC		√	√		47H
FFFFF178H	Interrupt control register	UA1TIC		√	√		47H
FFFFF180H	Interrupt control register	AD0IC		√	√		47H
FFFFF182H	Interrupt control register	AD1IC		√	√		47H
FFFFF186H	Interrupt control register	TM0EQIC0		√	√		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/D converter 0 mode register 0	ADA0M0		√	√		00H
FFFFF201H	A/D converter 0 mode register 1	ADA0M1		√	√		00H
FFFFF202H	A/D converter 0 channel specification register	ADA0S		√	√		00H
FFFFF203H	A/D converter 0 mode register 2	ADA0M2		√	√		00H
FFFFF210H	A/D0 conversion result register 0	ADA0CR0	R			√	Undefined
FFFFF211H	A/D0 conversion result register 0H	ADA0CR0H			√		Undefined
FFFFF212H	A/D0 conversion result register 1	ADA0CR1				√	Undefined
FFFFF213H	A/D0 conversion result register 1H	ADA0CR1H			√		Undefined
FFFFF214H	A/D0 conversion result register 2	ADA0CR2				√	Undefined
FFFFF215H	A/D0 conversion result register 2H	ADA0CR2H			√		Undefined
FFFFF216H	A/D0 conversion result register 3	ADA0CR3				√	Undefined
FFFFF217H	A/D0 conversion result register 3H	ADA0CR3H			√		Undefined
FFFFF220H	A/D converter 1 mode register 0	ADA1M0	R/W	√	√		00H
FFFFF221H	A/D converter 1 mode register 1	ADA1M1		√	√		00H
FFFFF222H	A/D converter 1 channel specification register	ADA1S		√	√		00H
FFFFF223H	A/D converter 1 mode register 2	ADA1M2		√	√		00H
FFFFF230H	A/D1 conversion result register 0	ADA1CR0	R			√	Undefined
FFFFF231H	A/D1 conversion result register 0H	ADA1CR0H			√		Undefined
FFFFF232H	A/D1 conversion result register 1	ADA1CR1				√	Undefined
FFFFF233H	A/D1 conversion result register 1H	ADA1CR1H			√		Undefined
FFFFF234H	A/D1 conversion result register 2	ADA1CR2				√	Undefined
FFFFF235H	A/D1 conversion result register 2H	ADA1CR2H			√		Undefined
FFFFF236H	A/D1 conversion result register 3	ADA1CR3				√	Undefined
FFFFF237H	A/D1 conversion result register 3H	ADA1CR3H			√		Undefined

(3/6)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF310H	External interrupt noise elimination control register	INTPNRC	R/W	√	√		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
FFFFF462H	Port 1 function control register	PFC1		√	√		00H
FFFFF466H	Port 3 function control register	PFC3		√	√		00H
FFFFF468H	Port 4 function control register	PFC4		√	√		00H
FFFFF540H	TMM0 control register 0	TM0CTL0		√	√		00H
FFFFF544H	TMM0 compare register 0	TM0CMP0				√	0000H
FFFFF5C0H	TMQ0 control register 0	TQ0CTL0		√	√		00H
FFFFF5C1H	TMQ0 control register 1	TQ0CTL1		√	√		00H
FFFFF5C2H	TMQ0 I/O control register 0	TQ0IOC0		√	√		00H
FFFFF5C3H	TMQ0 I/O control register 1	TQ0IOC1		√	√		00H
FFFFF5C4H	TMQ0 I/O control register 2	TQ0IOC2		√	√		00H
FFFFF5C5H	TMQ0 option register 0	TQ0OPT0		√	√		00H
FFFFF5C6H	TMQ0 capture/compare register 0	TQ0CCR0				√	0000H
FFFFF5C8H	TMQ0 capture/compare register 1	TQ0CCR1				√	0000H
FFFFF5CAH	TMQ0 capture/compare register 2	TQ0CCR2				√	0000H
FFFFF5CCH	TMQ0 capture/compare register 3	TQ0CCR3				√	0000H
FFFFF5CEH	TMQ0 counter read buffer register	TQ0CNT	R			√	0000H
FFFFF5F0H	High-impedance output control register 00	HZA0CTL0	R/W	√	√		00H
FFFFF5F1H	High-impedance output control register 01	HZA0CTL1		√	√		00H
FFFFF600H	TMQ1 control register 0	TQ1CTL0		√	√		00H
FFFFF601H	TMQ1 control register 1	TQ1CTL1		√	√		00H
FFFFF602H	TMQ1 I/O control register 0	TQ1IOC0		√	√		00H
FFFFF605H	TMQ1 option register 0	TQ1OPT0		√	√		00H
FFFFF606H	TMQ1 capture/compare register 0	TQ1CCR0				√	0000H
FFFFF608H	TMQ1 capture/compare register 1	TQ1CCR1				√	0000H
FFFFF60AH	TMQ1 capture/compare register 2	TQ1CCR2				√	0000H

(4/6)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF60CH	TMQ1 capture/compare register 3	TQ1CCR3	R/W			√	0000H
FFFFF60EH	TMQ1 counter read buffer register	TQ1CNT	R			√	0000H
FFFFF620H	TMQ1 option register 1	TQ1OPT1	R/W	√	√		00H
FFFFF621H	TMQ1 option register 2	TQ1OPT2		√	√		00H
FFFFF622H	TMQ1 I/O control register 3	TQ1IOC3		√	√		A8H
FFFFF623H	TMQ1 option register 3	TQ1OPT3		√	√		00H
FFFFF624H	TMQ1 dead-time compare register	TQ1DTC				√	0000H
FFFFF630H	High-impedance output control register 10	HZA1CTL0		√	√		00H
FFFFF631H	High-impedance output control register 11	HZA1CTL1		√	√		00H
FFFFF640H	TMP0 control register 0	TP0CTL0		√	√		00H
FFFFF641H	TMP0 control register 1	TP0CTL1		√	√		00H
FFFFF642H	TMP0 I/O control register 0	TP0IOC0		√	√		00H
FFFFF643H	TMP0 I/O control register 1	TP0IOC1		√	√		00H
FFFFF644H	TMP0 I/O control register 2	TP0IOC2		√	√		00H
FFFFF645H	TMP0 option register 0	TP0OPT0		√	√		00H
FFFFF646H	TMP0 capture/compare register 0	TP0CCR0				√	0000H
FFFFF648H	TMP0 capture/compare register 1	TP0CCR1				√	0000H
FFFFF64AH	TMP0 counter read buffer register	TP0CNT	R			√	0000H
FFFFF660H	TMP1 control register 0	TP1CTL0	R/W	√	√		00H
FFFFF661H	TMP1 control register 1	TP1CTL1		√	√		00H
FFFFF665H	TMP1 option register 0	TP1OPT0		√	√		00H
FFFFF666H	TMP1 capture/compare register 0	TP1CCR0				√	0000H
FFFFF668H	TMP1 capture/compare register 1	TP1CCR1				√	0000H
FFFFF66AH	TMP1 counter read buffer register	TP1CNT	R			√	0000H
FFFFF680H	TMP2 control register 0	TP2CTL0	R/W	√	√		00H
FFFFF681H	TMP2 control register 1	TP2CTL1		√	√		00H
FFFFF682H	TMP2 I/O control register 0	TP2IOC0		√	√		00H
FFFFF683H	TMP2 I/O control register 1	TP2IOC1		√	√		00H
FFFFF684H	TMP2 I/O control register 2	TP2IOC2		√	√		00H
FFFFF685H	TMP2 option register 0	TP2OPT0		√	√		00H
FFFFF686H	TMP2 capture/compare register 0	TP2CCR0				√	0000H
FFFFF688H	TMP2 capture/compare register 1	TP2CCR1				√	0000H
FFFFF68AH	TMP2 counter read buffer register	TP2CNT	R			√	0000H
FFFFF6A0H	TMP3 control register 0	TP3CTL0	R/W	√	√		00H
FFFFF6A1H	TMP3 control register 1	TP3CTL1		√	√		00H
FFFFF6A2H	TMP3 I/O control register 0	TP3IOC0		√	√		00H
FFFFF6A5H	TMP3 option register 0	TP3OPT0		√	√		00H
FFFFF6A6H	TMP3 capture/compare register 0	TP3CCR0				√	0000H
FFFFF6A8H	TMP3 capture/compare register 1	TP3CCR1				√	0000H
FFFFF6AAH	TMP3 counter read buffer register	TP3CNT	R			√	0000H
FFFFF6C0H	Oscillation stabilization time select register	OSTS	R/W		√		06H
FFFFF6D0H	Watchdog timer mode register	WDTM			√		67H
FFFFF6D1H	Watchdog timer enable register	WDTE			√		1AH
FFFFF702H	Port 1 function control expansion register	PFCE1		√	√		00H

(5/6)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1	8	16	32	
FFFFF802H	System status register	SYS	R/W	√	√			00H
FFFFF820H	Power save mode register	PSMR		√	√			00H
FFFFF822H	Clock control register	CKC			√			0AH
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/11H
FFFFF890H	Low-voltage detection register	LVIM		√	√			00H/82H
FFFFF891H	Low-voltage detection level select register	LVIS			√			00H
FFFFF892H	Internal RAM data status register	RAMS		√	√			01H
FFFFF8A8H	Reset source flag register 2	RESF2	R		√			00H/01H/ 10H/11H
FFFFF8AAH	System status register	SYS2	R/W	√	√			00H
FFFFF9ECH	Command register 2	PRCMD2	W		√			Undefined
FFFFF9F0H	Internal memory size setting register	IMS			√			00H
FFFFFA00H	UARTA0 control register 0	UA0CTL0	R/W	√	√			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
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
FFFFFC00H	External interrupt falling edge specification register 0	INTF0		√	√			00H
FFFFFC20H	External interrupt rising edge specification register 0	INTR0		√	√			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
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

(6/6)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
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
FFFFFF44H	Pull-up resistor option register DLL	PUDLL		√	√		00H

3.4.7 Special registers

Special registers are registers that are protected from being written with illegal data due to a program hang-up. The V850ES/IE2 has the following seven special registers divided into two types.

[Special registers subject to error report by SYS.PRERR bit]

- Power save control register (PSC)
- Clock control register (CKC)
- Processor clock control register (PCC)
- Clock monitor mode register (CLM)
- Reset source flag register (RESF)

<R> **Caution When writing to any of the above five special registers, use command register PRCMD.**

[Special registers subject to error report by SYS.PRERR2 bit]

- Low-voltage detection register (LVIM)
- Internal RAM data status register (RAMS)

<R> **Caution When writing to either of the above two special registers, use command register PRCMD2.**

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

(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 0, RESF[r0]
```

(2) Command register

PRCMD is an 8-bit register that protects the registers that may seriously affect the application system from being written, so that the system does not inadvertently stop due to a program hang-up. This register can be used as PRCMD or PRCMD2 via a special register setting. The first write access to a special register is valid after data has been written in advance to the command 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.

(a) Command register (PRCMD)

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

<R> **Caution After writing to the PRCMD register, writing to special registers PSC, CKC, PCC, CLM, and RESF is enabled.**

After reset: Undefined W Address: FFFFF1FCH

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

(b) Command register 2 (PRCMD2)

The PRCMD2 register is write-only, in 8-bit units (undefined data is read when this register is read).
Reset makes this register undefined.

<R> **Caution After writing to the PRCMD2 register, writing to special registers LVIM and RAMS is enabled.**

After reset: Undefined W Address: FFFFF9ECH

	7	6	5	4	3	2	1	0
PRCMD2	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

(3) System status register

Status flags that indicate the operation status of the overall system are allocated to this register. This register can be used as SYS or SYS2 via a special register setting.

(a) System status register (SYS)

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.

(b) System status register 2 (SYS2)

If this register is not written in the correct sequence including an access to the PRCMD2 register, data is not written to the intended register, a protection error occurs, and the PRERR2 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: FFFFF8AAH

	7	6	5	4	3	2	1	<0>
SYS2	0	0	0	0	0	0	0	PRERR2

PRERR2	Protection error detection
0	Protection error did not occur.
1	Protection error occurred.

The operating conditions of the PRERR flag are shown below. For the operating conditions of the PRERR2 flag, read PRCMD and SYS as PRCMD2 and SYS2 in the following explanation.

(i) 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 **3.4.7 (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 **3.4.7 (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.

(ii) Clear condition (PRERR flag = 0)

- When 0 is written to the PRERR flag
- When the system is reset

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

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

3.4.8 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 V850ES CPU core is basically made in 3 clocks; however, in the V850ES/IE2, a wait is required in addition to those 3 clocks. Set 11H (set wait for 2 clocks) to the VSWC register.

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

CPU Clock Frequency (f_{CPU})	VSWC Set Value
312.5 kHz ≤ f_{CPU} ≤ 20 MHz	11H

Caution When using the V850ES/IE2, the VSWC register must be set first.
Set other registers as needed after setting the VSWC register.

Remark When a register includes status flags that indicate the statuses of the on-chip peripheral functions (register such as the STATUS1n register) or a register that indicates the count value of a timer 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

- ○ I/O ports: 39
- ○ 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, the on-chip pull-up resistor can be connected when a port is in the input port mode and when the alternate-function pin of the port functions as an input pin. The on-chip pull-up resistor can be connected to the TOP21, TOQ1T1 to TOQ1T3, TOQ1B1 to TOQ1B3, TOP31, and TOQH01 to TOQH03 pins, which function as output pins when the alternate function of the corresponding port is used, when these pins go into a high-impedance state because of processing of the TOP2OFF, TOQ1OFF, TOP3OFF, and TOQH0OFF pins or software.

4.2 Basic Port Configuration

The V850ES/IE2 incorporates a total of 39 I/O ports labeled ports 0 to 4 and DL. The port configuration is shown below.

Figure 4-1. Port Configuration

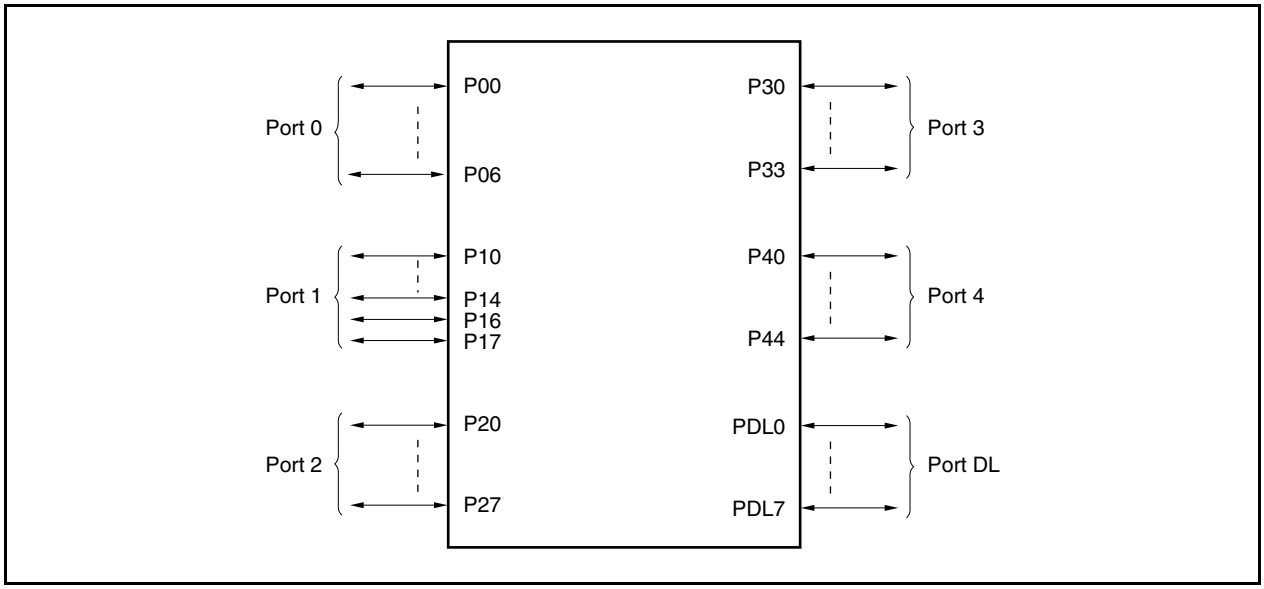


Table 4-1. Power Supplies for I/O Buffer of Each Pin

Power Supply	Corresponding Pins
AV_{DD0}, AV_{DD1}	ANI00 to ANI03, ANI10 to ANI13
EV_{DD}	Ports 0 to 4, port DL, $\overline{RESET}$

4.3 Port Configuration

Table 4-2. Port Configuration

Item	Configuration
Control registers	Port n register (Pn: n = 0 to 4, DLL) Port n mode register (PMn: n = 0 to 4, DLL) Port n mode control register (PMCn: n = 0 to 4) Port n function control register (PFCn: n = 1, 3, 4) Port 1 function control expansion register (PFCE1) Pull-up resistor option register (PUn: n = 0 to 4, DLL)
Ports	I/O: 39
Pull-up resistor	Software control: 39

(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 4-3. 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 input/output 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

	7	6	5	4	3	2	1	0
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

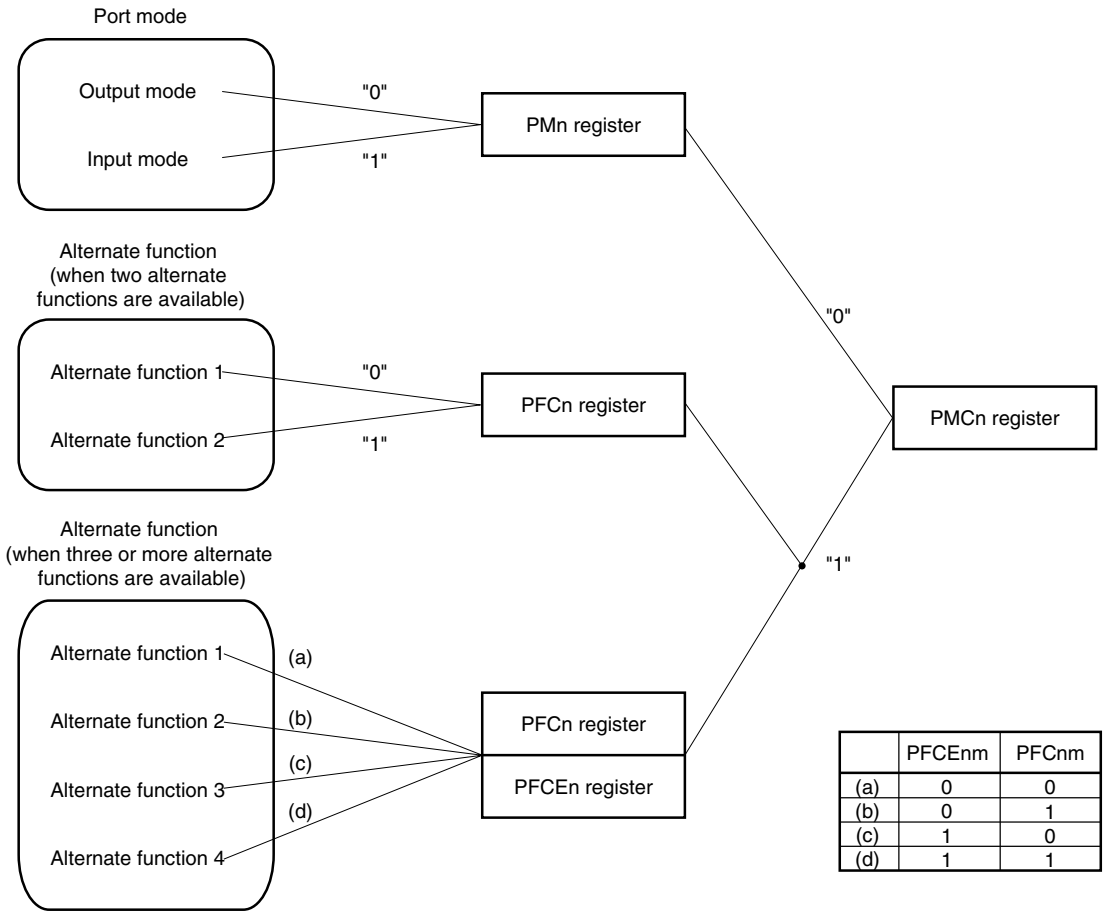
	7	6	5	4	3	2	1	0
PUn	PUn7	PUn6	PUn5	PUn4	PUn3	PUn2	PUn1	PUn0

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

(7) Port settings

Set the ports as follows.

Figure 4-2. 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 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 (INTPn), and then specify the valid edge (n = 0 to 6).

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

Remark Switch to the alternate function using the following procedure.

<1> Set the PFCn and PFCEn registers.

<2> Set the PMCn register.

<3> Set the INTRn/INTFn register (when external interrupt pin is set).

If the PFCn register is set before setting the PMCn 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.

Port 0 has an alternate function as the following pins.

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

Pin Name	Pin No.	Alternate-Function Pin	I/O	Pull-Up[Note 1]
P00[Note 2]	17	INTP0/TOQH0OFF[Note 3]	Input	Provided
P01[Note 2]	16	INTP1/TOQ1OFF[Note 3]	Input	
P02[Note 2]	15	INTP2/TOP2OFF[Note 3]	Input	
P03[Note 2]	14	INTP3/TOP3OFF[Note 3]	Input	
P04[Note 2]	13	INTP4/ADTRG0[Note 3]	Input	
P05[Note 2]	12	INTP5/ADTRG1[Note 3]	Input	
P06[Note 2]	11	INTP6	Input	

Notes 1. Software pull-up function

2. These pins operate as Schmitt trigger inputs when they are read in the port mode.

3. The TOQH0OFF, TOQ1OFF, TOP2OFF, TOP3OFF, ADTRG0, and ADTRG1 signals are input to the high-impedance output controller (see **CHAPTER 9 MOTOR CONTROL FUNCTION**) and A/D converters 0 and 1 (see **CHAPTER 11 A/D CONVERTERS 0 AND 1**) after noise is eliminated by a port (analog delay). In addition, a signal whose edge was detected is input to the interrupt controller (INTC) as INTPn (n = 0 to 5). Edge detection is performed by the high-impedance output controller and A/D converters 0 and 1.

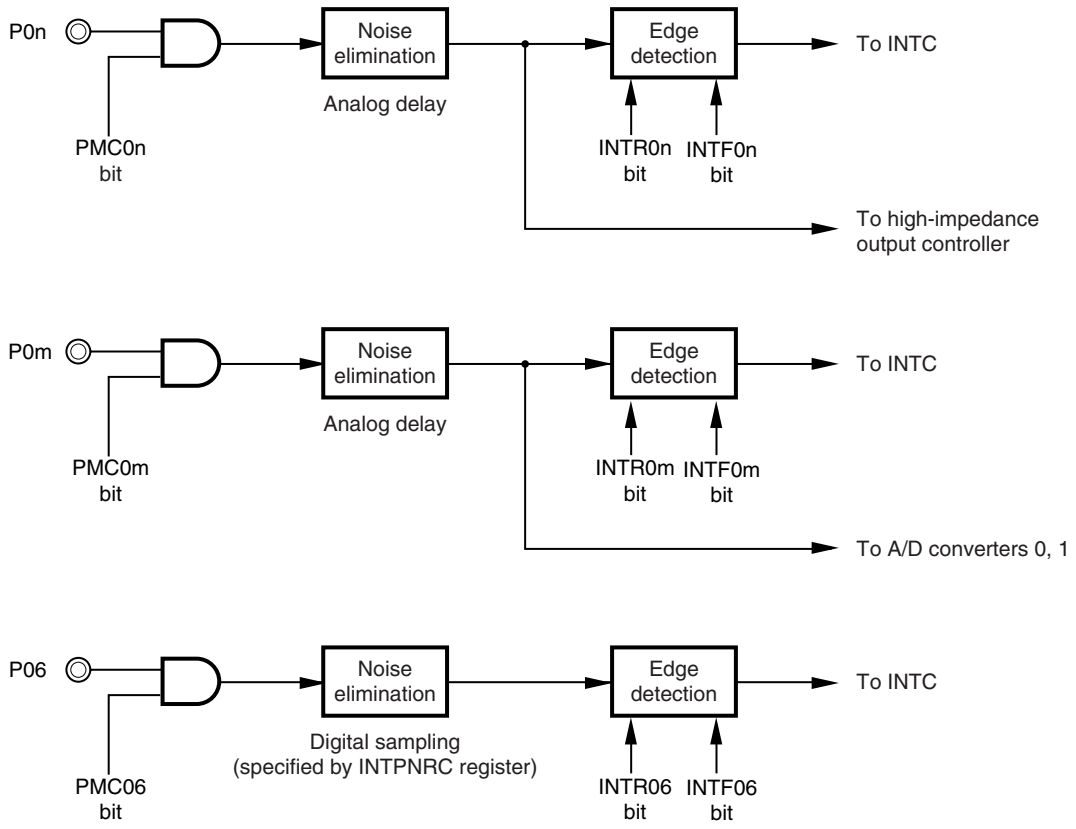
Cautions 1. To control the high-impedance output of a timer for motor control, be sure to set the PMC0.PMC0n bit to 1 and then specify the edge to be detected and enable the operation of the high-impedance output controller (n = 1 to 3), because the output of the motor control timer may go into a high-impedance state if a wrong valid edge is detected.

2. To input an A/D trigger to A/D converter 0 or 1, be sure to set the PMC0.PMC0n bit to 1 and then specify the edge to be detected and enable the operation of A/D converter 0 or 1 because the trigger may be input if a wrong valid edge is detected (n = 4, 5).

3. To switch to external interrupt input (INTPn) 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 (INTPn), and then specify the valid edge (n = 0 to 6).

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

A noise elimination function is included as an alternate function of port 0.



Caution To control high-impedance output of the external interrupt function and motor output control function, set the PMC0a bit to 1 (a = 0 to 6).

Remark n = 0 to 3
m = 4, 5

(1) Registers

(a) Port 0 register (P0)

After reset: Undefined R/W Address: FFFFF400H

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

P0n	Control of output data (in output mode) (n = 0 to 6)
0	Output 0.
1	Output 1.

(b) Port 0 mode register (PM0)

After reset: FFH R/W Address: FFFFF420H

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

PM0n	Control of input/output mode (in port mode) (n = 0 to 6)
0	Output mode
1	Input mode

(c) Port 0 mode control register (PMC0)

After reset: 00H R/W Address: FFFFF440H

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

PMC06	Specification of operating mode of P06 pin
0	I/O port
1	INTP6 input

PMC05	Specification of operating mode of P05 pin
0	I/O port
1	INTP5 input/ADTRG1 input

PMC04	Specification of operating mode of P04 pin
0	I/O port
1	INTP4 input/ADTRG0 input

PMC03	Specification of operating mode of P03 pin
0	I/O port
1	INTP3 input/TOP3OFF input

PMC02	Specification of operating mode of P02 pin
0	I/O port
1	INTP2 input/TOP2OFF input

PMC01	Specification of operating mode of P01 pin
0	I/O port
1	INTP1 input/TOQ1OFF input

PMC00	Specification of operating mode of P00 pin
0	I/O port
1	INTP0 input/TOQH0OFF input

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

After reset: 00H R/W Address: FFFFFC40H

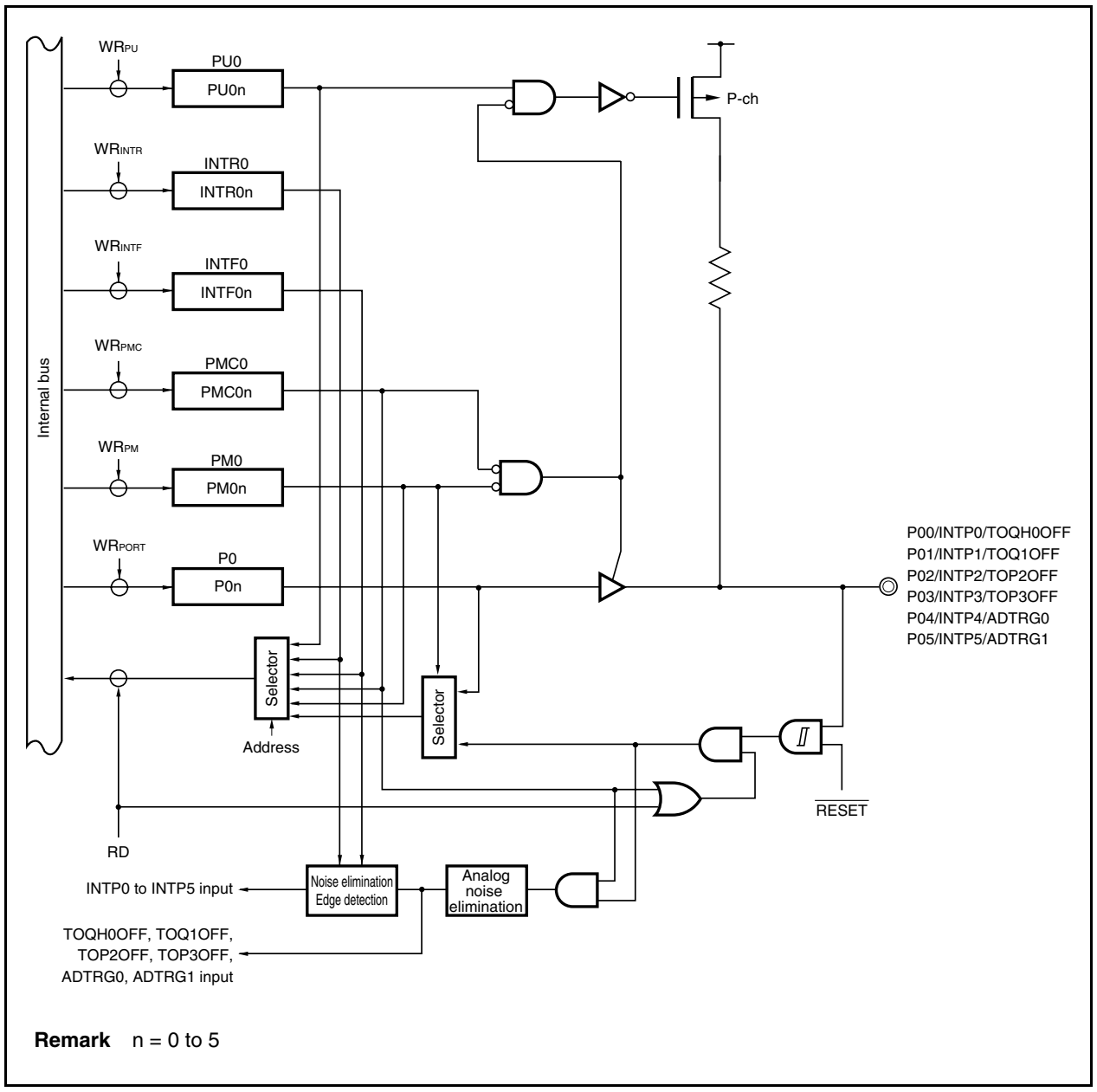
	7	6	5	4	3	2	1	0
PU0	0	PU06	PU05	PU04	PU03	PU02	PU01	PU00

PU0n	Control of on-chip pull-up resistor connection (n = 0 to 6)
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 alternate function. The pull-up resistor cannot be connected when the pin is in the output state.

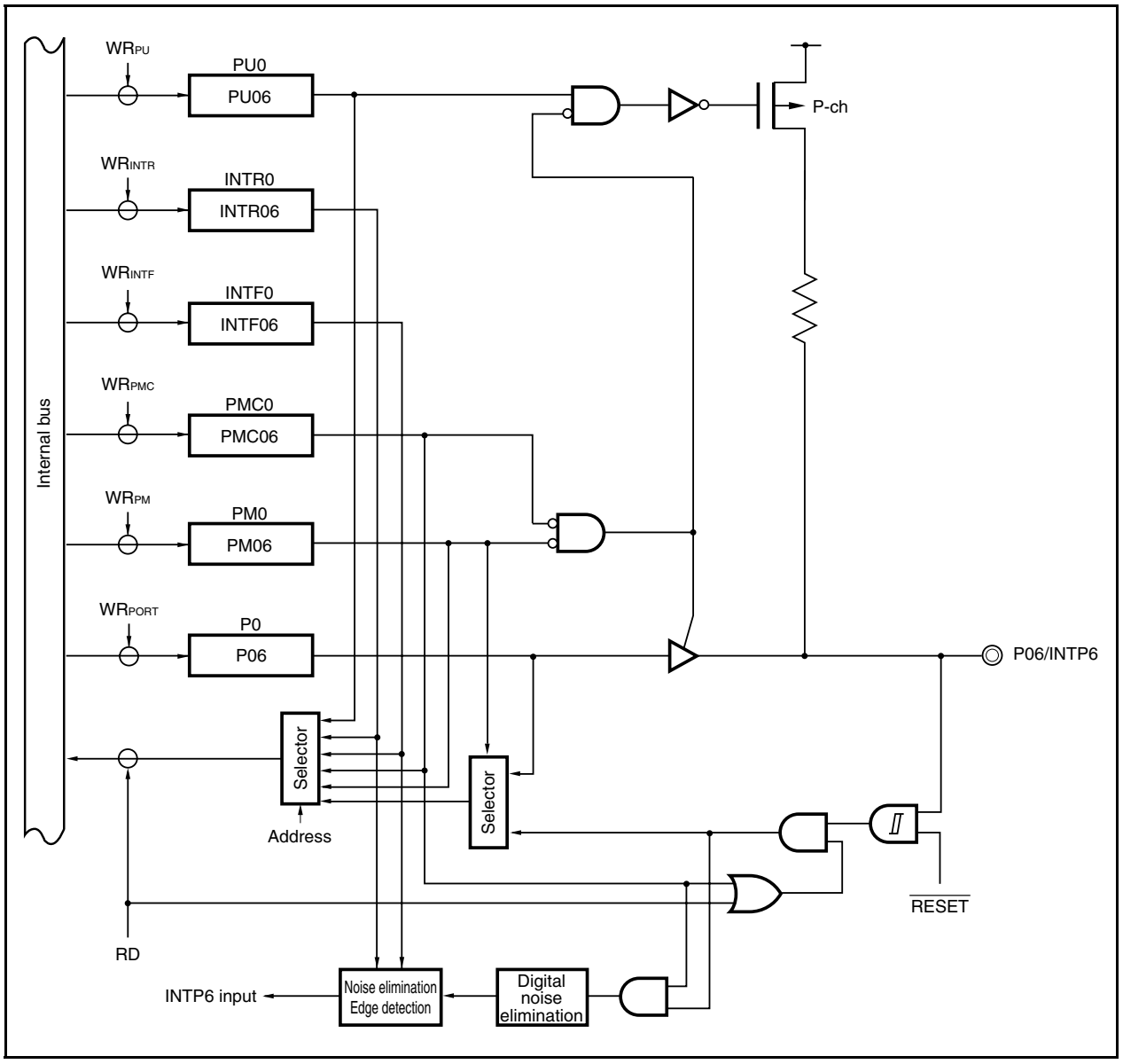
(2) Block diagrams

Figure 4-3. Block Diagram of P00 to P05 Pins



Remark n = 0 to 5

Figure 4-4. Block Diagram of P06 Pin



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 4-5. Alternate-Function Pins of Port 1

Pin Name	Pin No.	Alternate-Function Pin	I/O	Pull-Up[Note 1]
P10[Note 2]	24	TOQH01/TIQ01/TOQ01	I/O	Provided
P11[Note 2]	23	TIQ02/TOQ02	I/O	
P12[Note 2]	22	TOQH02/TIQ03/TOQ03	I/O	
P13[Note 2]	21	TIQ00	Input	
P14[Note 2]	20	TOQH03/EVTQ0	Input	
P16[Note 2] (CLMER)	19	TOQ00(CLMER)/TIP20	I/O	
P17[Note 2]	18	TOP21/TIP21	I/O	

Notes 1. Software pull-up function

2. These pins operate as Schmitt trigger inputs when they are read in the port mode.

Cautions 1. When P10, P12, P14, and P17 are used as TOQH01 to TOQH03 and TOP21, output is stopped when the following signals are asserted.

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

2. If P16 is set as an output port or the TOQ00 output function, when an error (oscillator stop) is detected by the clock monitor, the CLMER signal (low level) is output from P16. If P16 is set to as an input port or the TIP20 input function, the CLMER signal is not output. When the CLMER signal is output, the CLMER signal output is held until reset signal is generated. While the CLMER signal is active, the P16, PM16, PMC16, PFC16, and PU16 bits cannot be written.

(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	0	P14	P13	P12	P11	P10

P1n	Control of output data (in output mode) (n = 0 to 4, 6, 7)
0	Output 0.
1	Output 1.

(b) Port 1 mode register (PM1)

After reset: FFH R/W Address: FFFFF422H

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

PM1n	Control of input/output mode (in port mode) (n = 0 to 4, 6, 7)
0	Output mode[Note]
1	Input mode

Note If P16 is used as an output port, when an error (oscillator stop) is detected by the clock monitor, the CLMER signal (low level) is output from P16. Low-level output is released by reset signal. For details, see **Table 4-5 Alternate-Function Pins of Port 1**.

(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	0	PMC14	PMC13	PMC12	PMC11	PMC10

PMC17	Specification of operating mode of P17 pin
0	I/O port
1	TOP21 output/TIP21 input

PMC16	Specification of operating mode of P16 pin
0	I/O port
1	TOQ00 (CLMER) output/TIP20 input

PMC14	Specification of operating mode of P14 pin
0	I/O port
1	TOQH03 output/EVTQ0 input

PMC13	Specification of operating mode of P13 pin
0	I/O port
1	TIQ00 input

PMC12	Specification of operating mode of P12 pin
0	I/O port
1	TOQH02 output/TIQ03 input/TOQ03 output

PMC11	Specification of operating mode of P11 pin
0	I/O port
1	TIQ02 input/TOQ02 output

PMC10	Specification of operating mode of P10 pin
0	I/O port
1	TOQH01 output/TIQ01 input/TOQ01 output

(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	0	PFC14	PFC13	PFC12	PFC11	PFC10

Remark For the specification of alternate function, see **4.3.2 (1) (f) Setting of alternate function 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	0	0	0	0	0	PFCE12	PFCE11	PFCE10

Remark For the specification of alternate function, see **4.3.2 (1) (f) Setting of alternate function of port 1**.

(f) Setting of alternate function of port 1

PFC17	Specification of alternate function of P17 pin
0	TOP21 output
1	TIP21 input

PFC16	Specification of alternate function of P16 pin
0	TOQ00 (CLMER) output[Note 1]
1	TIP20 input

PFC14	Specification of alternate function of P14 pin
0	TOQH03 output[Note 2]
1	EVTQ0 input

PFC13	Specification of alternate function of P13 pin
0	Setting prohibited
1	TIQ00 input

PFCE12	PFC12	Specification of alternate function of P12 pin
0	0	TOQH02 output[Note 2]
0	1	TIQ03 input
1	0	TOQ03 output
1	1	Setting prohibited

PFCE11	PFC11	Specification of alternate function of P11 pin
0	0	Setting prohibited
0	1	TIQ02 input
1	0	TOQ02 output
1	1	Setting prohibited

PFCE10	PFC10	Specification of alternate function of P10 pin
0	0	TOQH01 output[Note 2]
0	1	TIQ01 input
1	0	TOQ01 output
1	1	Setting prohibited

Notes 1. If P16 is used as the TOQ00 output pin, when an error (oscillator stop) is detected by the clock monitor, the CLMER signal (low level) is output from P16. Low-level output is released by reset signal. For details, see **Table 4-5 Alternate-Function Pins of Port 1**.

2. These are setting prohibited when TMQ0 is in other than PWM output mode.

(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	0	PU14	PU13	PU12	PU11	PU10

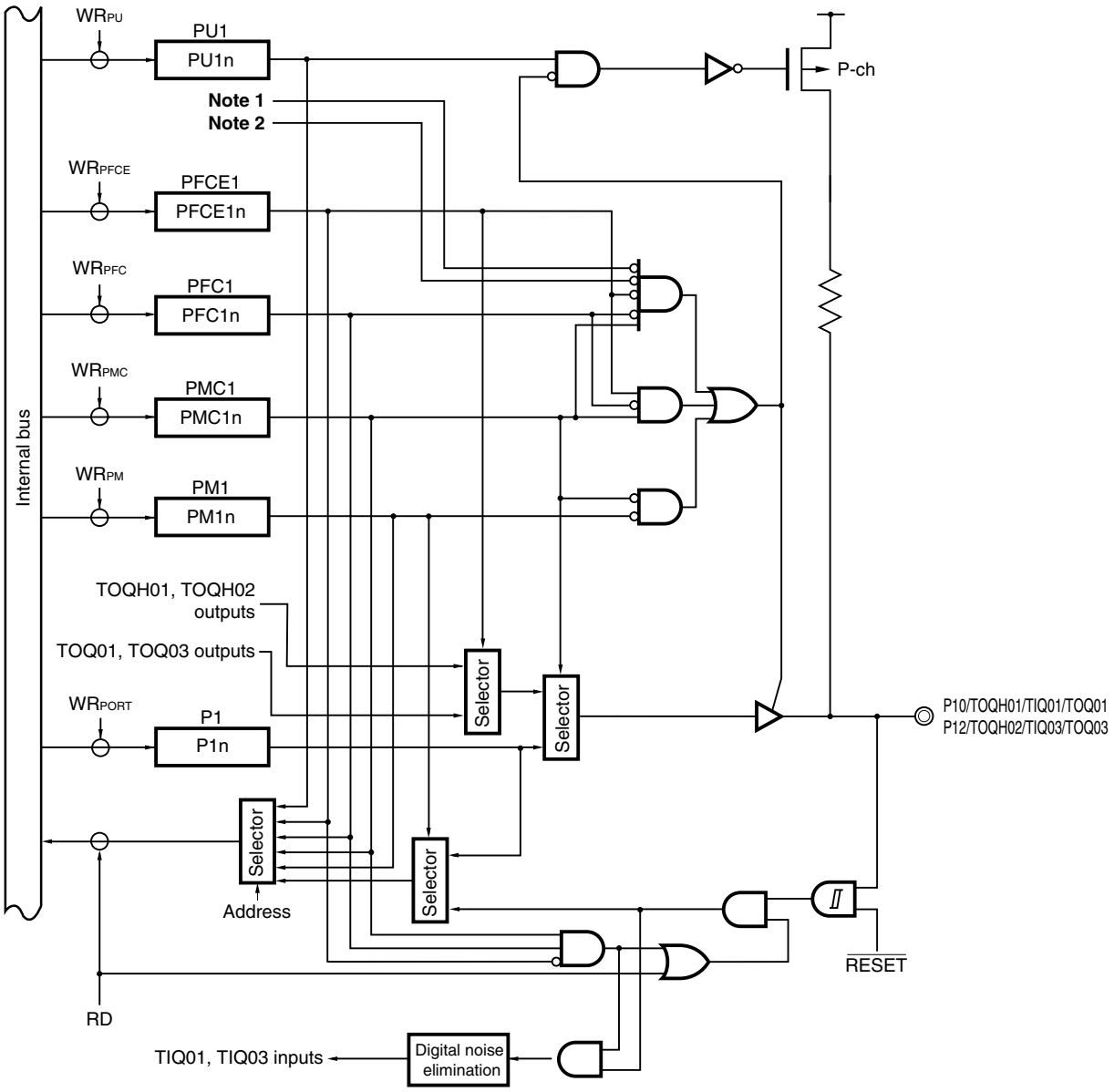
PU1n	Control of on-chip pull-up resistor connection (n = 0 to 4, 6, 7)
0	Do not connect
1	Connect[Note]

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 only be connected to the TOQH01 to TOQH03 and TOP21 pins when the pin goes into a high-impedance state in the alternate-function mode due to the TOQH0OFF and TOP2OFF pins, or software processing.

The resistor cannot be connected when the pin is in the output state.

(2) Block diagrams

Figure 4-5. Block Diagram of P10 and P12 Pins



Notes 1. Output of high impedance setting signal from high impedance output controller

2. Output of clock stop detection signal from clock monitor

Remark n = 0, 2

Figure 4-6. Block Diagram of P11 Pin

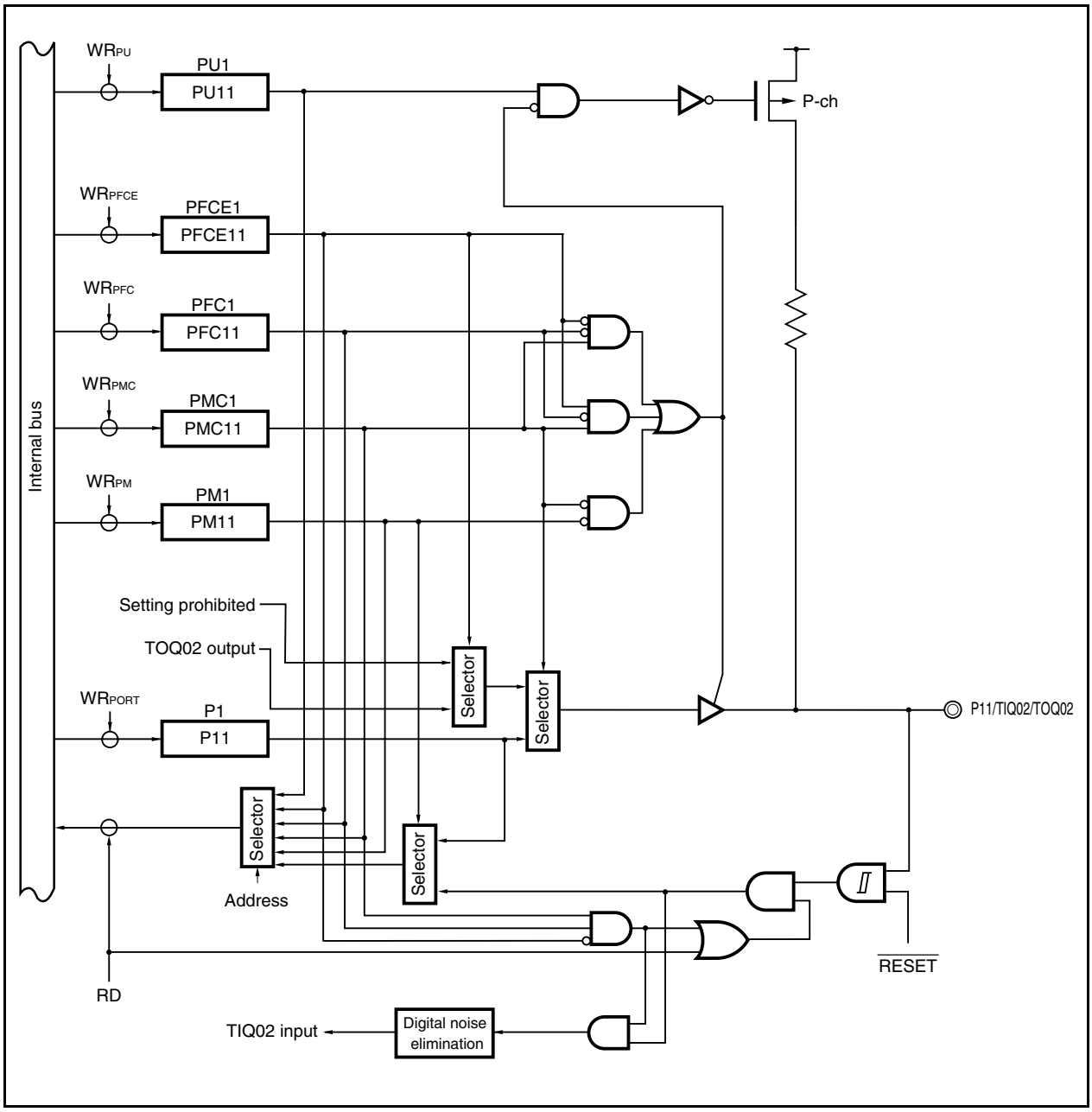


Figure 4-7. Block Diagram of P13 Pin

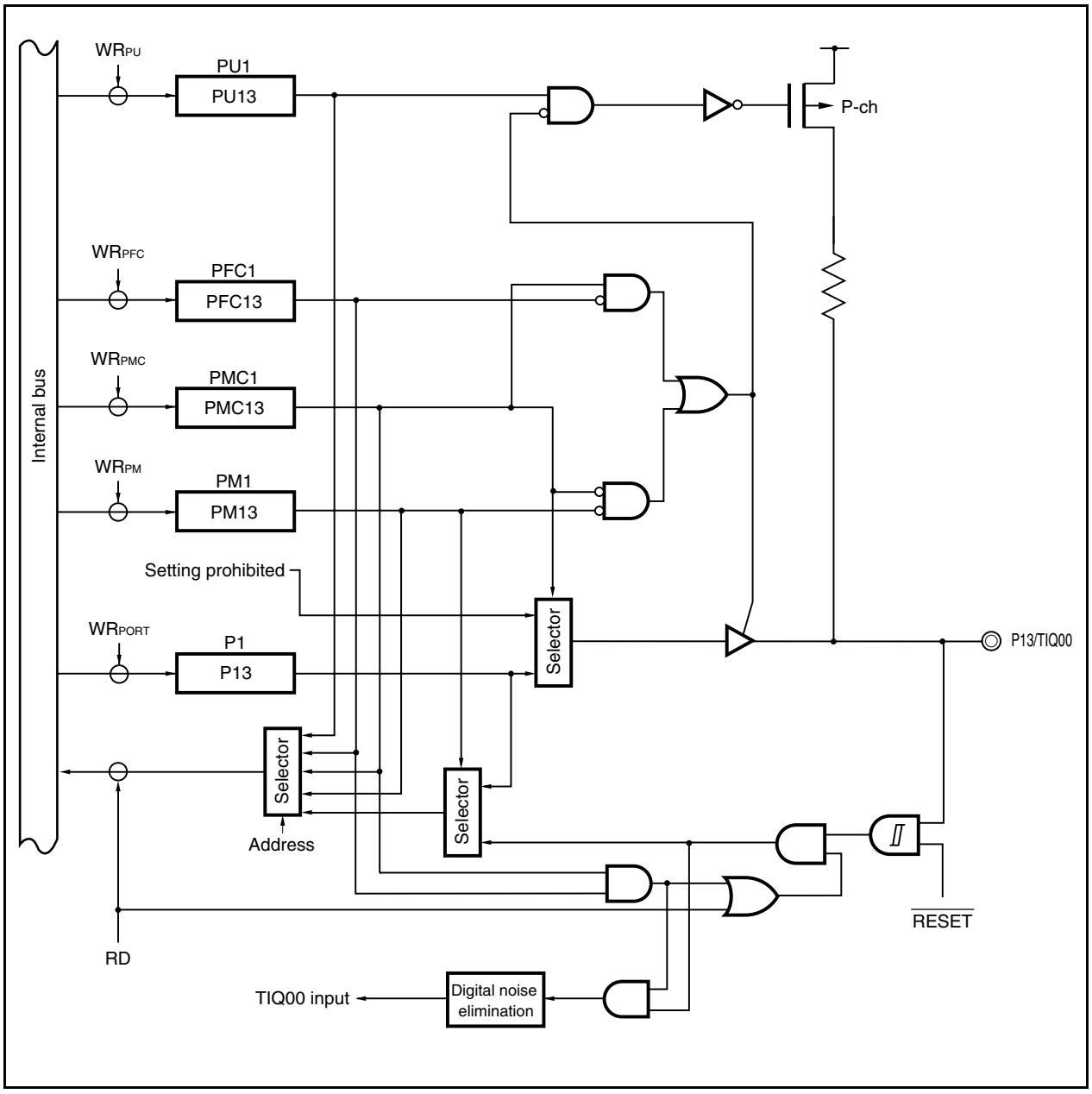
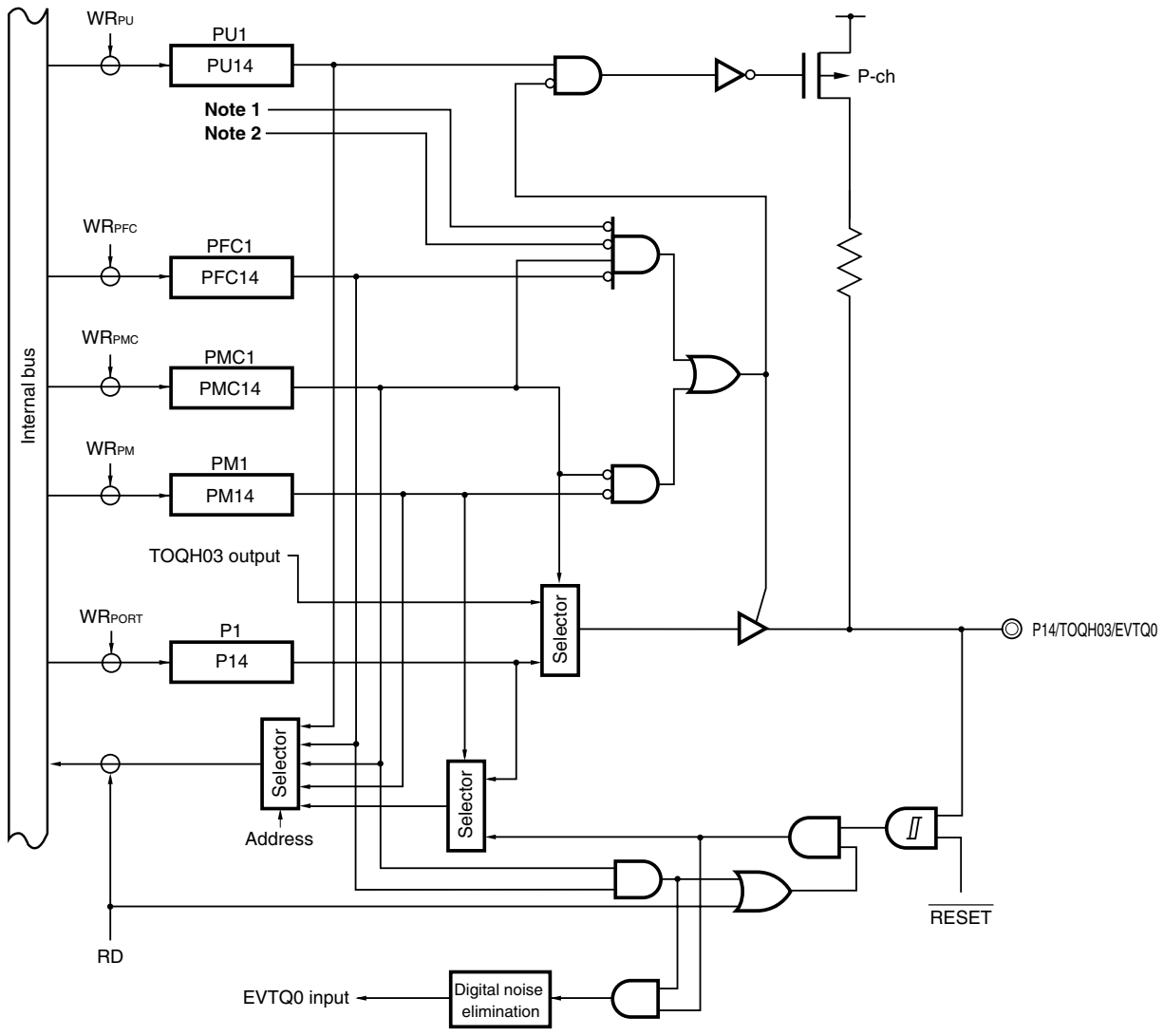


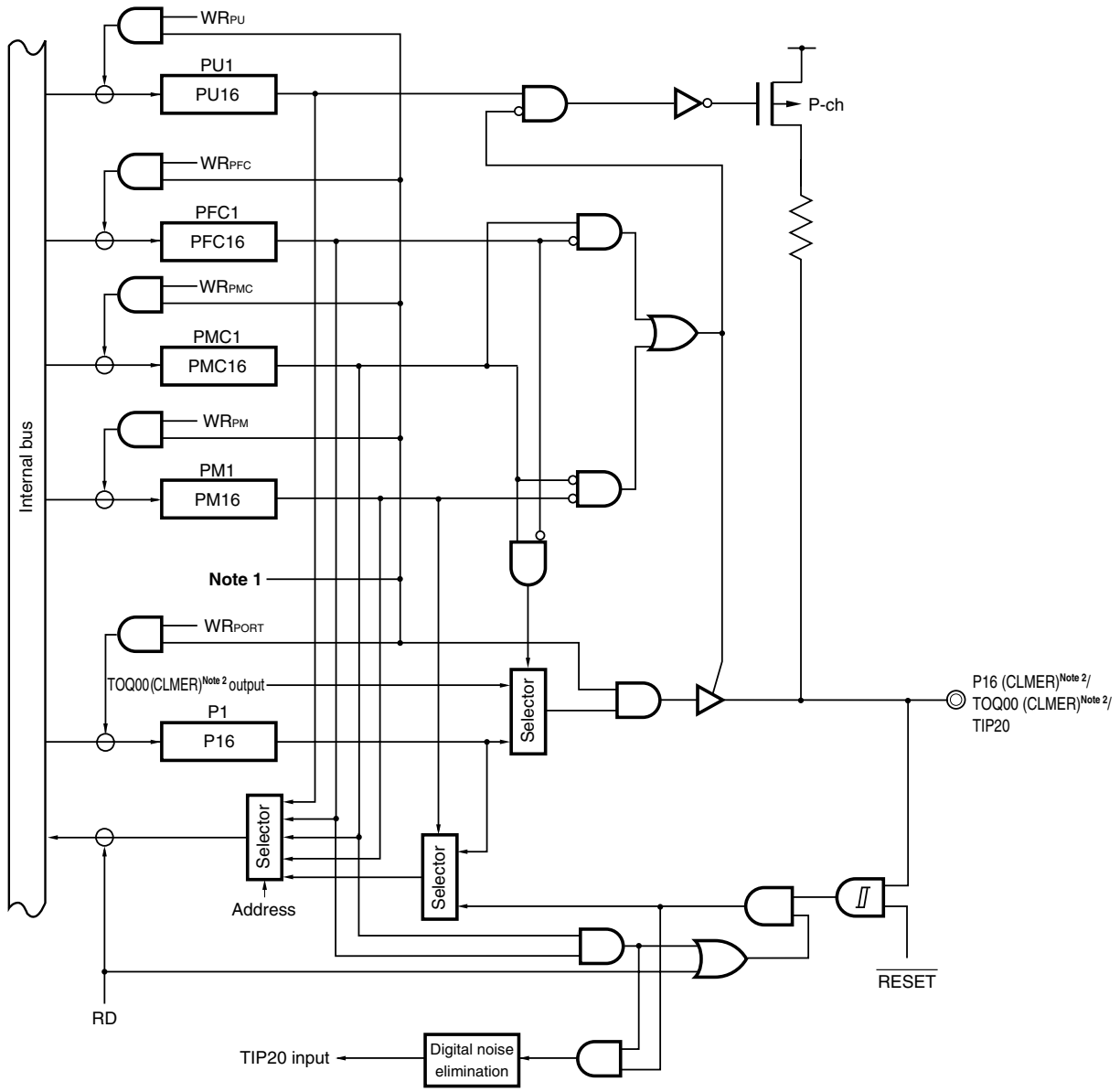
Figure 4-8. Block Diagram of P14 Pin



Notes 1. Output of high impedance setting signal from high impedance output controller

2. Output of clock stop detection signal from clock monitor

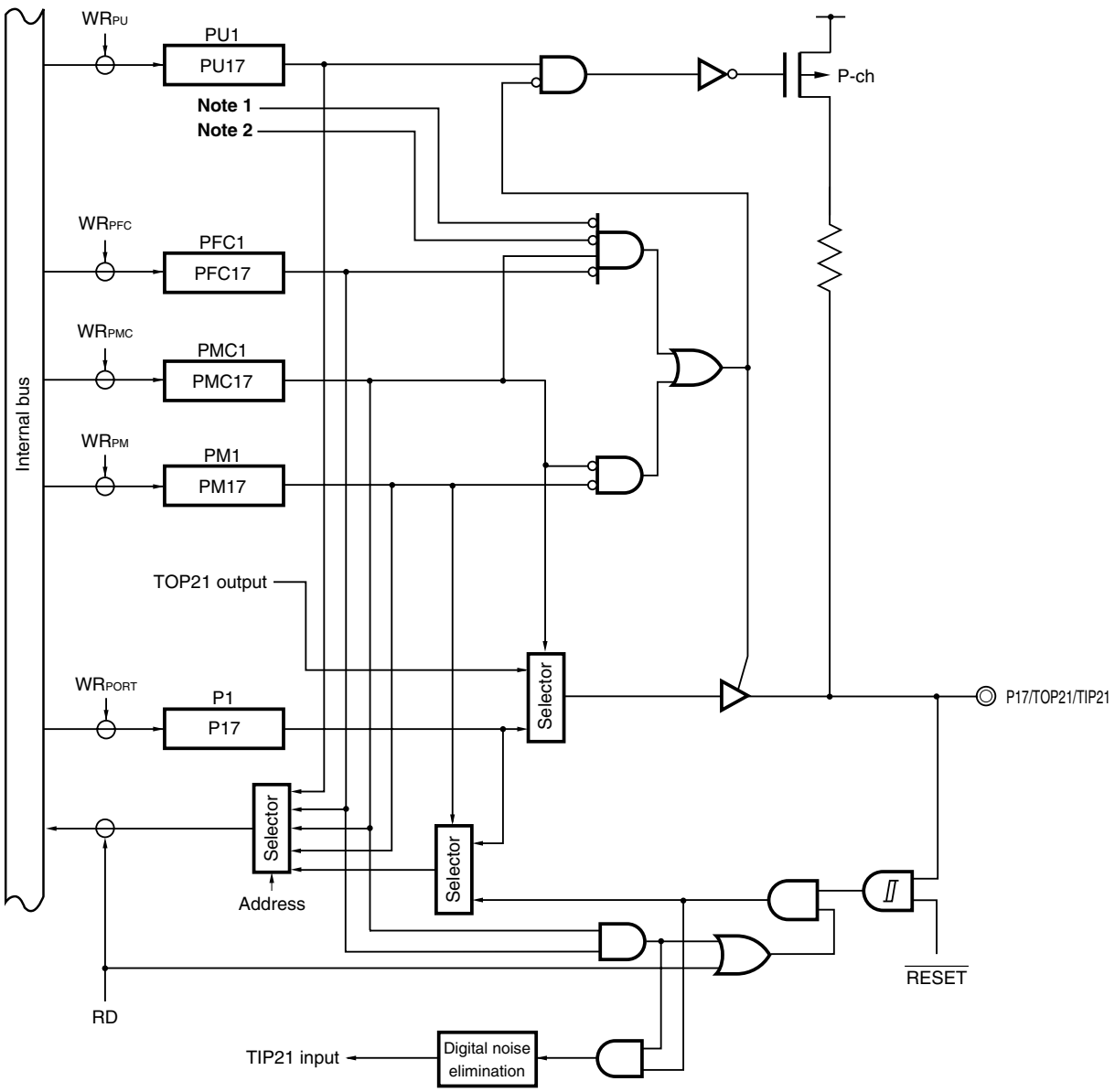
Figure 4-9. Block Diagram of P16 Pin



Notes 1. Output of clock stop detection signal from clock monitor

2. When P16 is set as the output port or TOQ00 output function, if an error (oscillator stop) is detected by the clock monitor, the CLMER signal (low level) is output from P16. Low-level output is released by reset signal. For details, see **Table 4-5 Alternate-Function Pins of Port 1**.

Figure 4-10. Block Diagram of P17 Pin



Notes 1. Output of high impedance setting signal from high impedance output controller

2. Output of clock stop detection signal from clock monitor

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 4-6. Alternate-Function Pins of Port 2

Pin Name	Pin No.	Alternate-Function Pin	I/O	Pull-Up[Note]
P20	54	TOQ1T1	Output	Provided
P21	53	TOQ1B1	Output	
P22	52	TOQ1T2	Output	
P23	51	TOQ1B2	Output	
P24	50	TOQ1T3	Output	
P25	49	TOQ1B3	Output	
P26	46	TOQ10	Output	
P27	45	TOP31	Output	

Note Software pull-up function

Caution When P20 to P25 and P27 are used as TOQ1T1 to TOQ1T3, TOQ1B1 to TOQ1B3, and TOP31, output is stopped when the following signals are asserted.

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

(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) (n = 0 to 7)
0	Output 0.
1	Output 1.

(b) Port 2 mode register (PM2)

After reset: FFH R/W Address: FFFFF424H

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

PM2n	Control of input/output mode (in port mode) (n = 0 to 7)
0	Output mode
1	Input mode

(c) Port 2 mode control register (PMC2)

After reset: 00H R/W Address: FFFFF444H

	7	6	5	4	3	2	1	0
PMC2	PMC27	PMC26	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20

PMC27	Specification of operating mode of P27 pin
0	I/O port
1	TOP31 output

PMC26	Specification of operating mode of P26 pin
0	I/O port
1	TOQ10 output

PMC25	Specification of operating mode of P25 pin
0	I/O port
1	TOQ1B3 output

PMC24	Specification of operating mode of P24 pin
0	I/O port
1	TOQ1T3 output

PMC23	Specification of operating mode of P23 pin
0	I/O port
1	TOQ1B2 output

PMC22	Specification of operating mode of P22 pin
0	I/O port
1	TOQ1T2 output

PMC21	Specification of operating mode of P21 pin
0	I/O port
1	TOQ1B1 output

PMC20	Specification of operating mode of P20 pin
0	I/O port
1	TOQ1T1 output

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

After reset: 00H R/W Address: FFFFFC44H

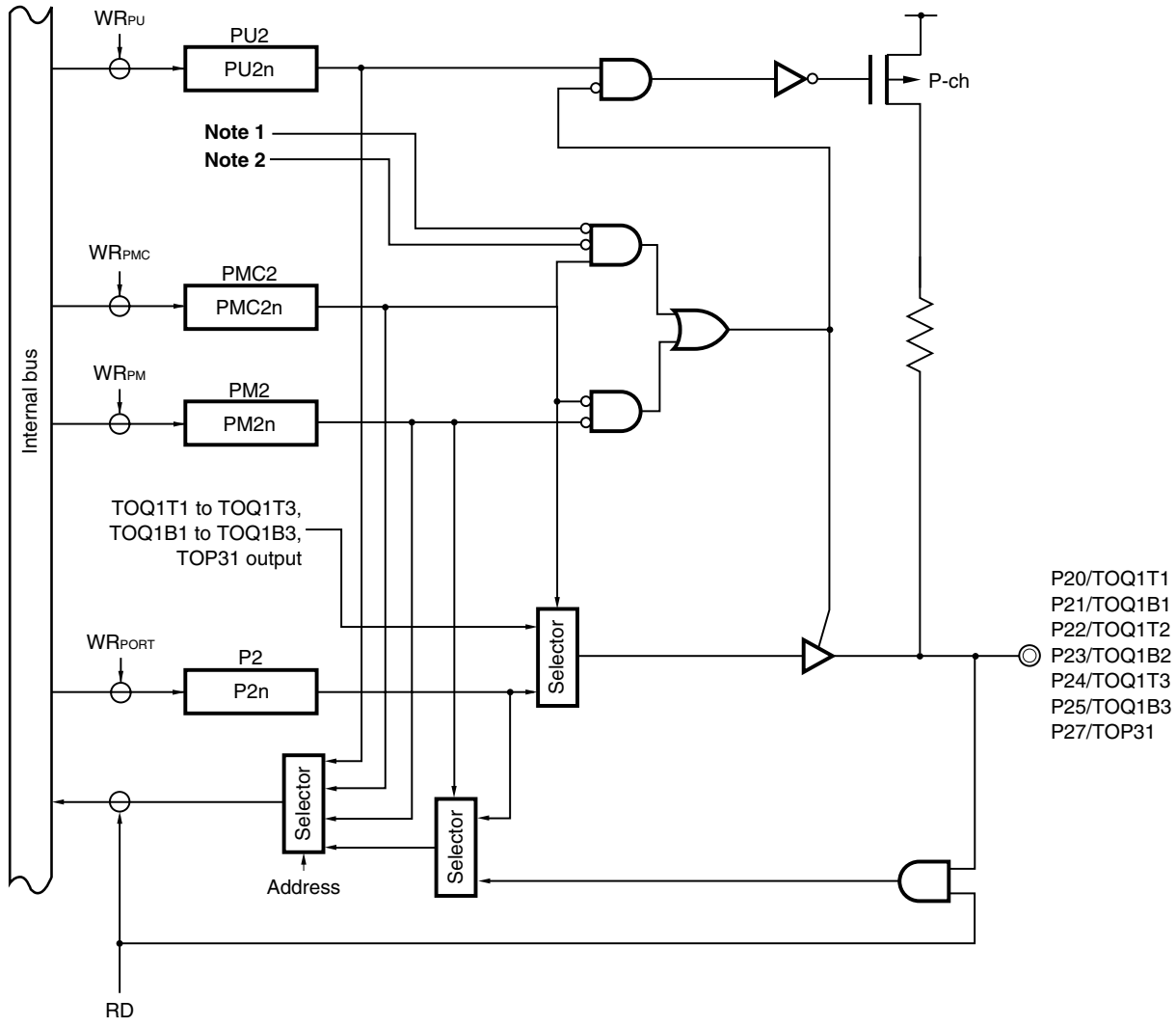
	7	6	5	4	3	2	1	0
PU2	PU27	PU26	PU25	PU24	PU23	PU22	PU21	PU20

PU2n	Control of on-chip pull-up resistor connection (n = 0 to 7)
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. Moreover, an on-chip pull-up resistor can only be connected to the TOQ1T1 to TOQ1T3, TOQ1B1 to TOQ1B3, and TOP31 pins when these pins go into a high-impedance state in the alternate-function mode due to the TOQ1OFF or TOP3OFF pin, or software processing. The resistor cannot be connected when the pin is in the output state.

(2) Block diagram

Figure 4-11. Block Diagram of P20 to P25 and P27 Pins

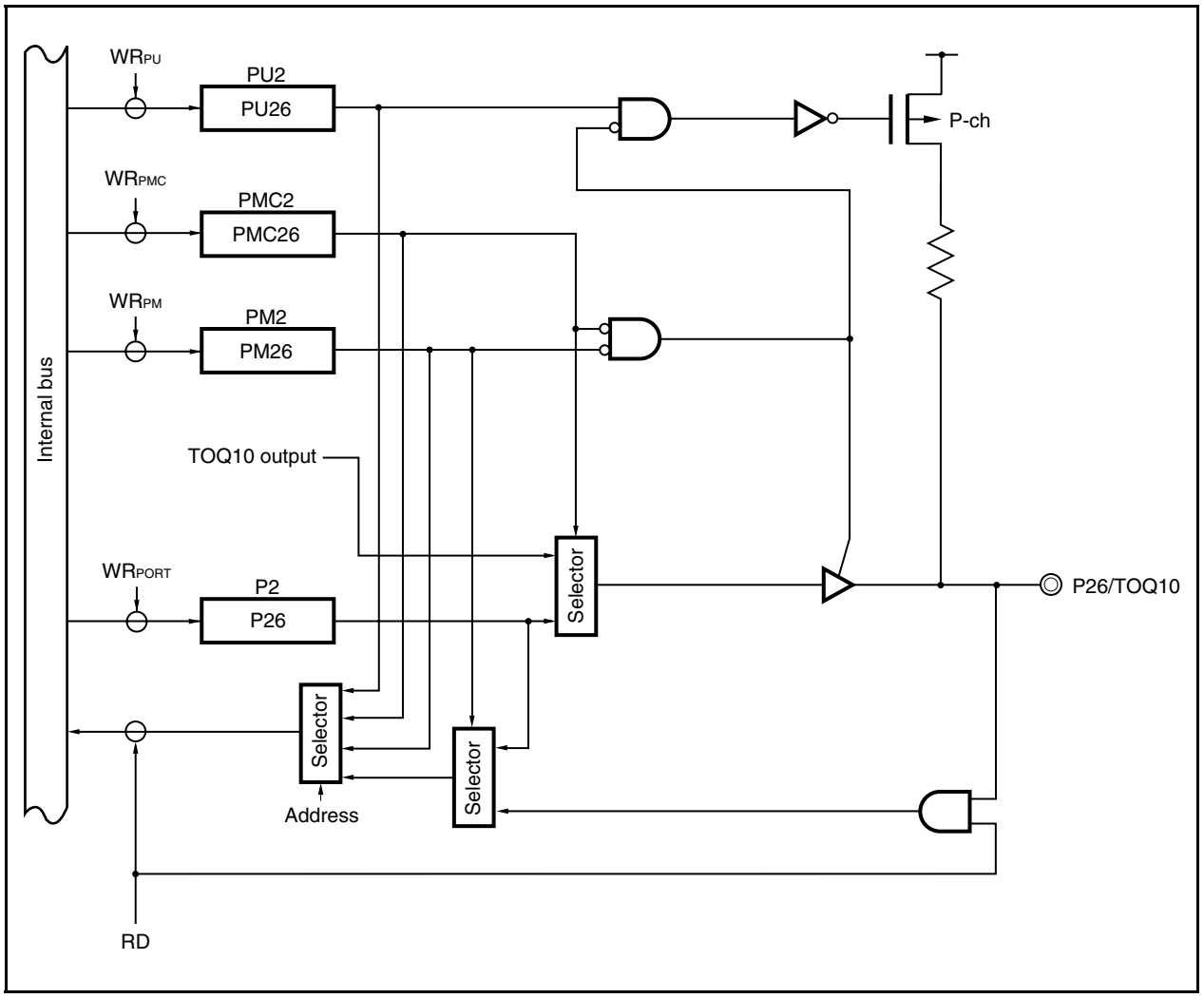


Notes 1. Output of high impedance setting signal from high impedance output controller

2. Output of clock stop detection signal from clock monitor

Remark n = 0 to 5, 7

Figure 4-12. Block Diagram of P26 Pin



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 4-7. Alternate-Function Pins of Port 3

Pin Name	Pin No.	Alternate-Function Pin	I/O	Pull-Up[Note 1]
P30[Note 2]	44	RXDA0	Input	Provided
P31	43	TXDA0	Output	
P32[Note 2]	42	RXDA1	Input	
P33	41	TXDA1	Output	

Notes 1. Software pull-up function

2. These pins operate as Schmitt trigger inputs when they are read in the port mode.

(1) Registers

(a) Port 3 register (P3)

After reset: Undefined R/W Address: FFFFF406H

	7	6	5	4	3	2	1	0
P3	0	0	0	0	P33	P32	P31	P30

P3n	Control of output data (in output mode) (n = 0 to 3)
0	Output 0.
1	Output 1.

(b) Port 3 mode register (PM3)

After reset: FFH R/W Address: FFFFF426H

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

PM3n	Control of input/output mode (in port mode) (n = 0 to 3)
0	Output mode
1	Input mode

(c) Port 3 mode control register (PMC3)

After reset: 00H R/W Address: FFFFF446H

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

PMC33	Specification of operating mode of P33 pin
0	I/O port
1	TXDA1 output

PMC32	Specification of operating mode of P32 pin
0	I/O port
1	RXDA1 input

PMC31	Specification of operating mode of P31 pin
0	I/O port
1	TXDA0 output

PMC30	Specification of operating mode of P30 pin
0	I/O port
1	RXDA0 input

(d) Port 3 function control register (PFC3)

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3	0	0	0	0	PFC33	PFC32	0	0

PFC33	Specification of alternate function of P33 pin
0	Setting prohibited
1	TXDA1 output

PFC32	Specification of alternate function of P32 pin
0	Setting prohibited
1	RXDA1 input

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

After reset: 00H R/W Address: FFFFFC46H

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

PU3n	Control of on-chip pull-up resistor connection (n = 0 to 3)
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. The resistor cannot be connected when the pin is in the output state.

(2) Block diagram

Figure 4-13. Block Diagram of P30 Pin

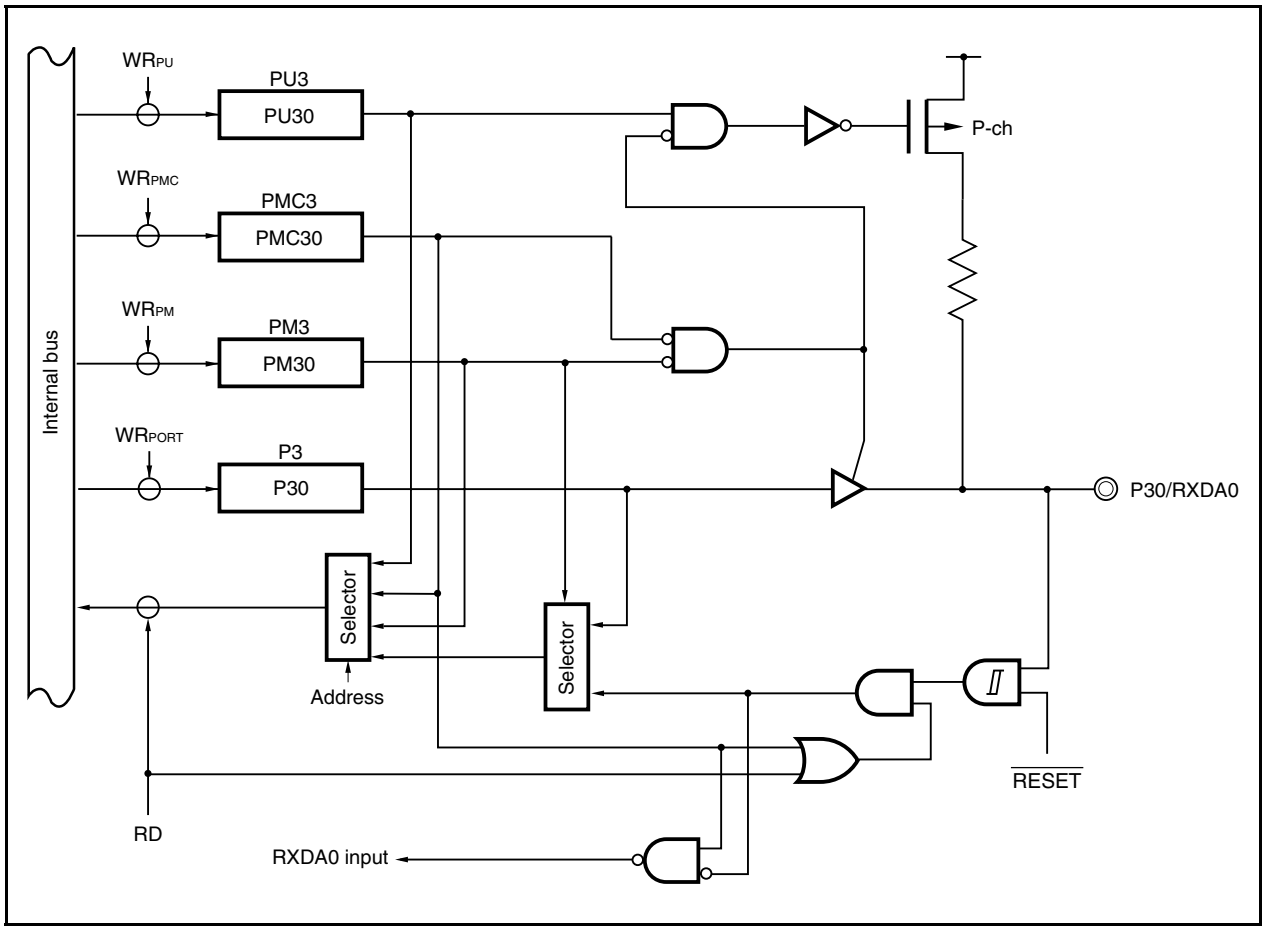


Figure 4-14. Block Diagram of P31 Pin

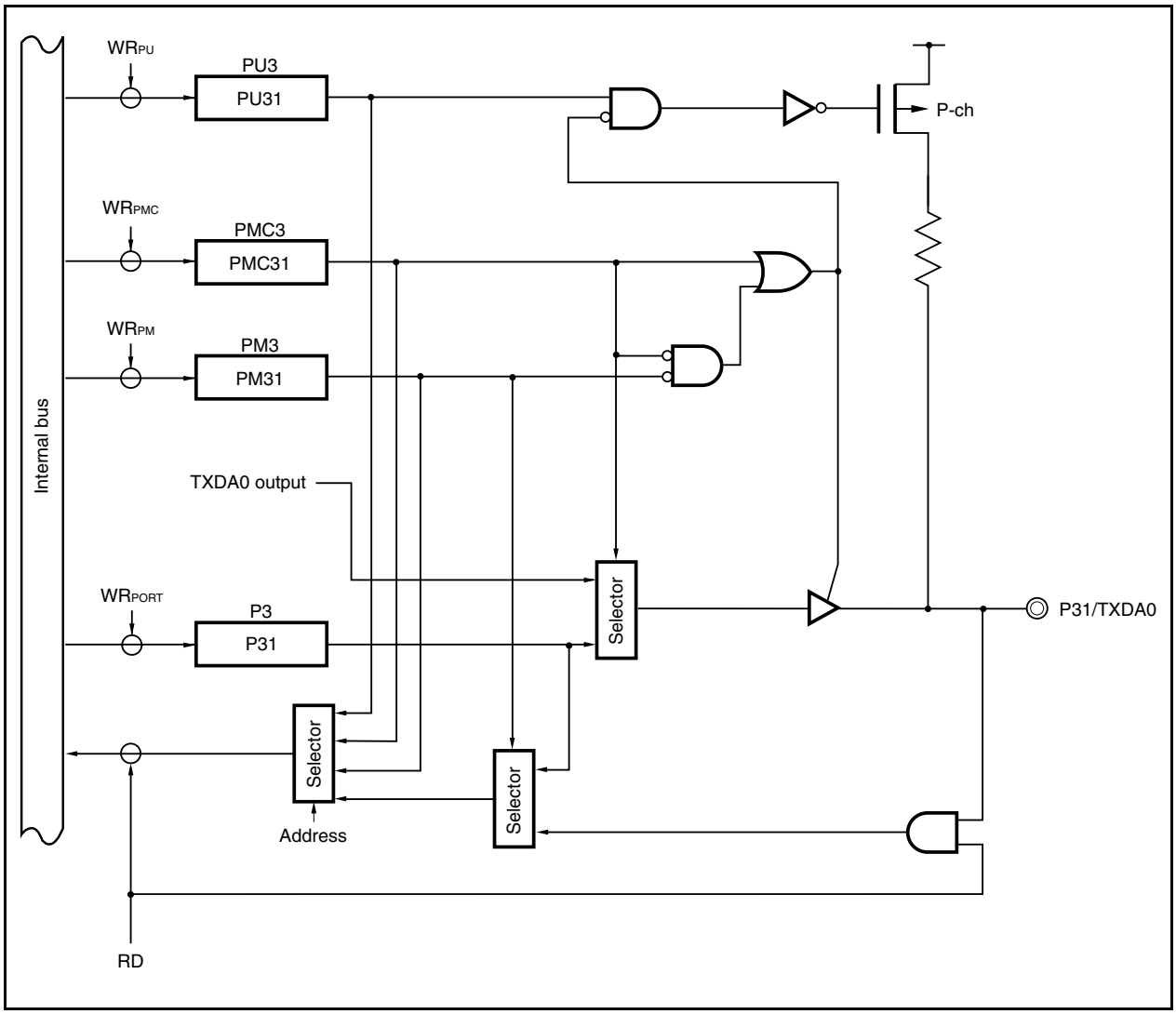


Figure 4-15. Block Diagram of P32 Pin

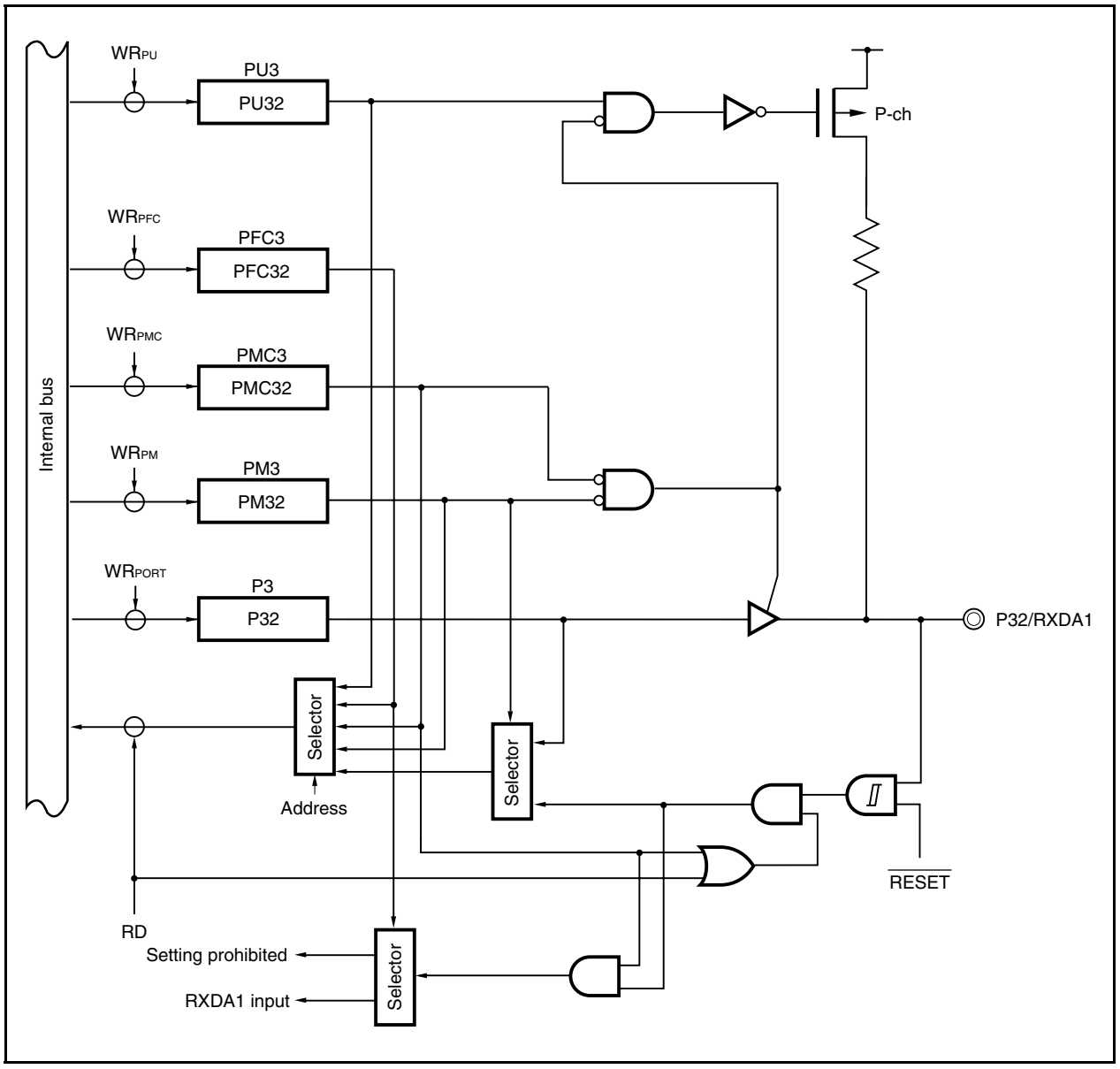
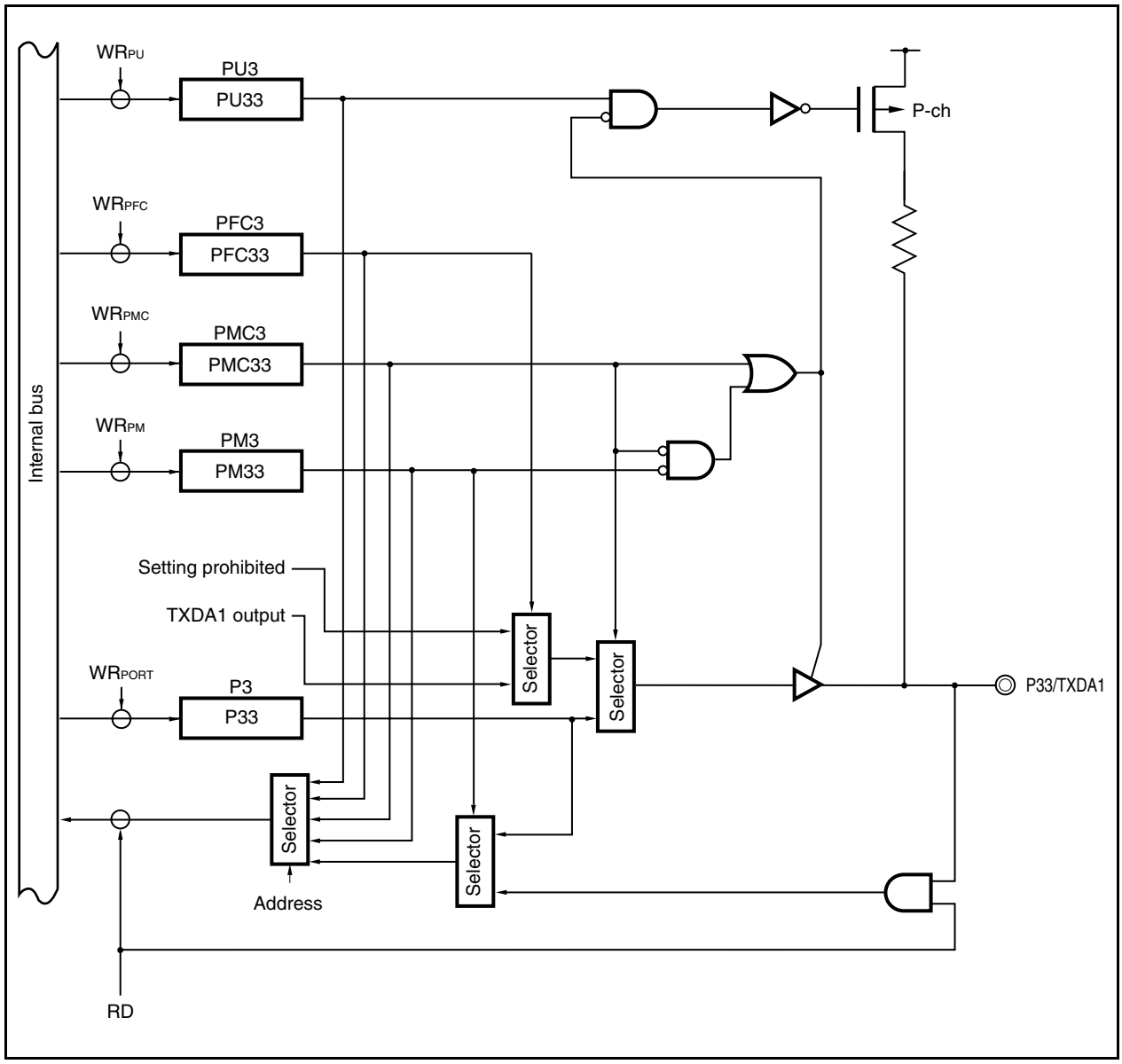


Figure 4-16. Block Diagram of P33 Pin



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 4-8. Alternate-Function Pins of Port 4

Pin Name	Pin No.	Alternate-Function Pin	I/O	Pull-Up[Note 1]
P40[Note 2]	40	SIB0	Input	Provided
P41	39	SOB0	Output	
P42[Note 2]	38	$\overline{\text{SCKB0}}$	I/O	
P43[Note 2]	37	TOP00/TIP00	I/O	
P44[Note 2]	36	TOP01/TIP01	I/O	

Notes 1. Software pull-up function

2. These pins operate as Schmitt trigger inputs when they are read in the port mode.

(1) Registers

(a) Port 4 register (P4)

After reset: Undefined R/W Address: FFFFF408H

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

P4n	Control of output data (in output mode) (n = 0 to 4)
0	Output 0.
1	Output 1.

(b) Port 4 mode register (PM4)

After reset: FFH R/W Address: FFFFF428H

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

PM4n	Control of input/output mode (in port mode) (n = 0 to 4)
0	Output mode
1	Input mode

(c) Port 4 mode control register (PMC4)

After reset: 00H R/W Address: FFFFF448H

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

PMC44	Specification of operating mode of P44 pin
0	I/O port
1	TOP01 output/TIP01 input

PMC43	Specification of operating mode of P43 pin
0	I/O port
1	TOP00 output/TIP00 input

PMC42	Specification of operating mode of P42 pin
0	I/O port
1	$\overline{\text{SCKB0}}$ I/O

PMC41	Specification of operating mode of P41 pin
0	I/O port
1	SOB0 output

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

(d) Port 4 function control register (PFC4)

After reset: 00H R/W Address: FFFFF468H

	7	6	5	4	3	2	1	0
PFC4	0	0	0	PFC44	PFC43	0	0	0

PFC44	Specification of alternate function of P44 pin
0	TOP01 output
1	TIP01 input

PFC43	Specification of alternate function of P43 pin
0	TOP00 output
1	TIP00 input

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

After reset: 00H R/W Address: FFFFFC48H

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

PU4n	Control of on-chip pull-up resistor connection (n = 0 to 4)
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 when in the $\overline{\text{SCKB0}}$ pin slave mode). The resistor cannot be connected when the pin is in the output state.

(2) Block diagram

Figure 4-17. Block Diagram of P40 Pin

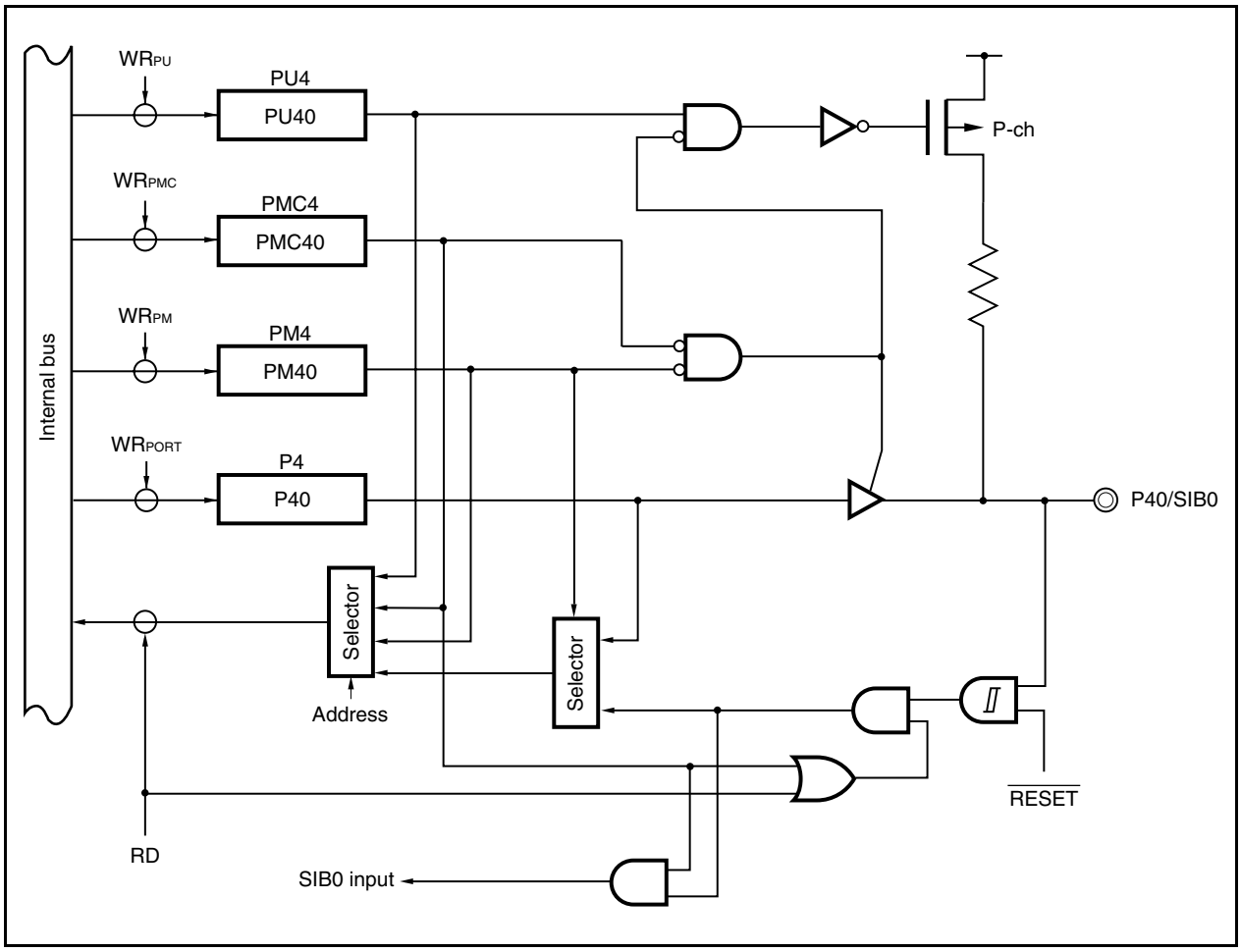


Figure 4-18. Block Diagram of P41 Pin

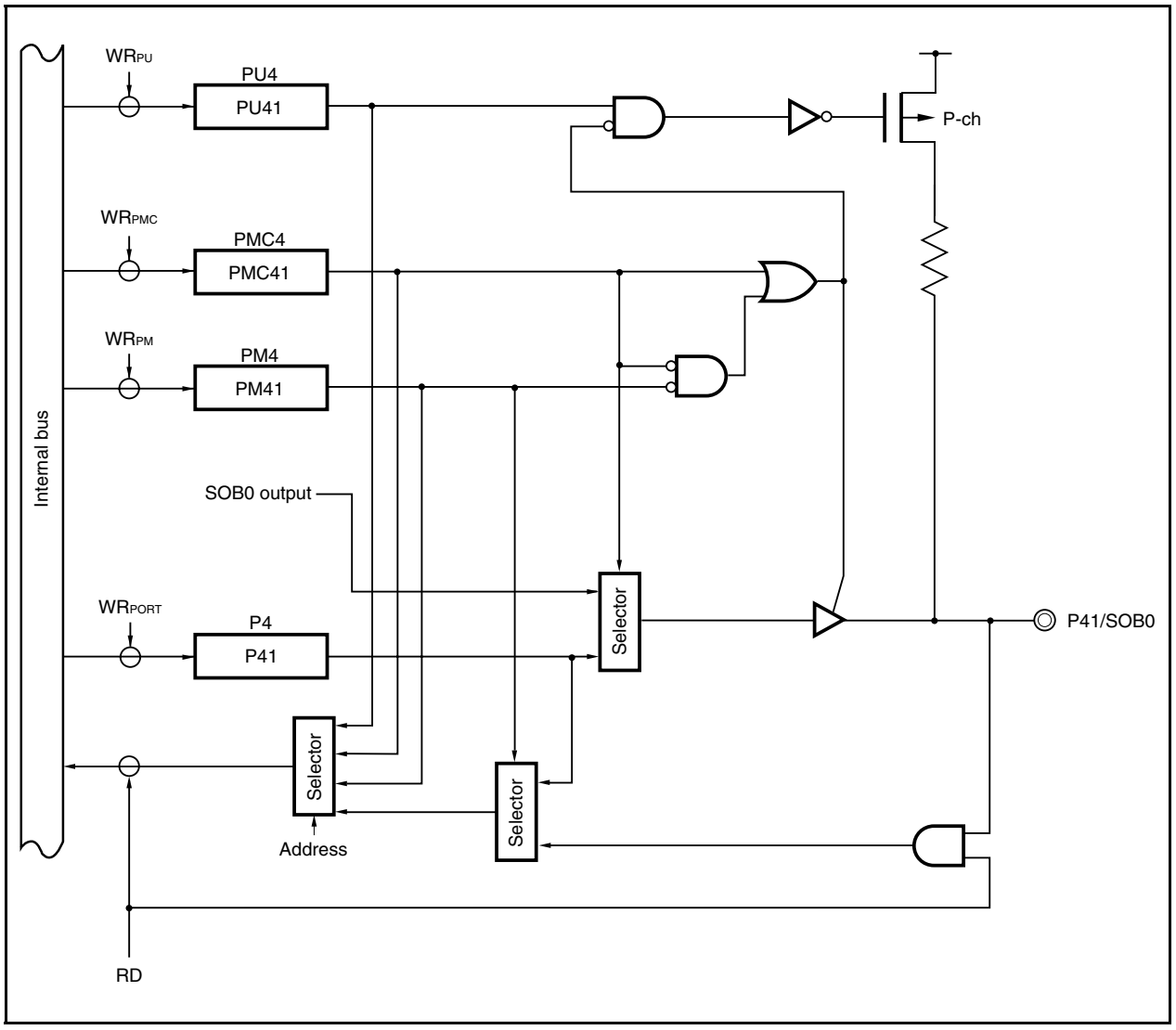


Figure 4-19. Block Diagram of P42 Pin

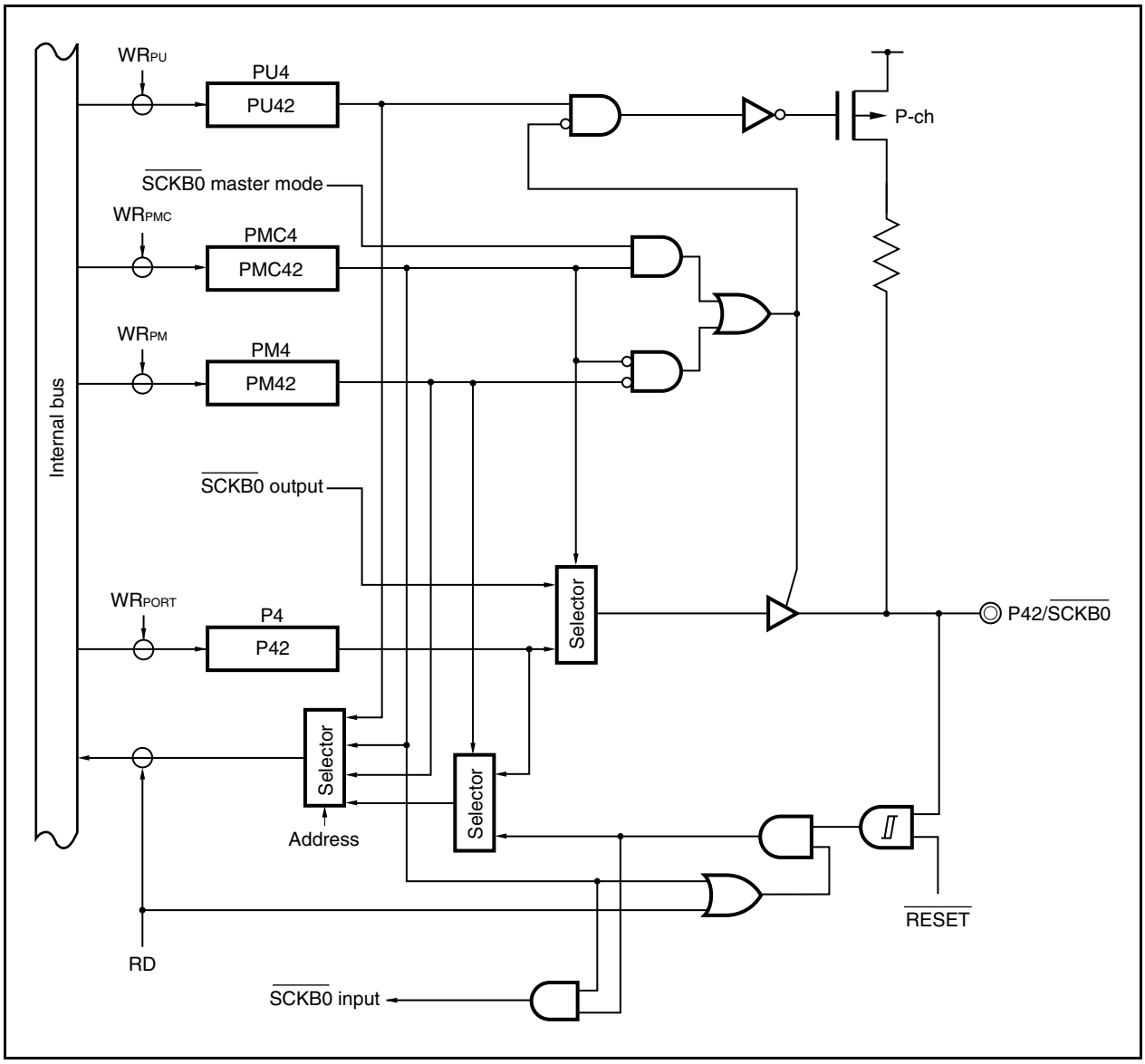
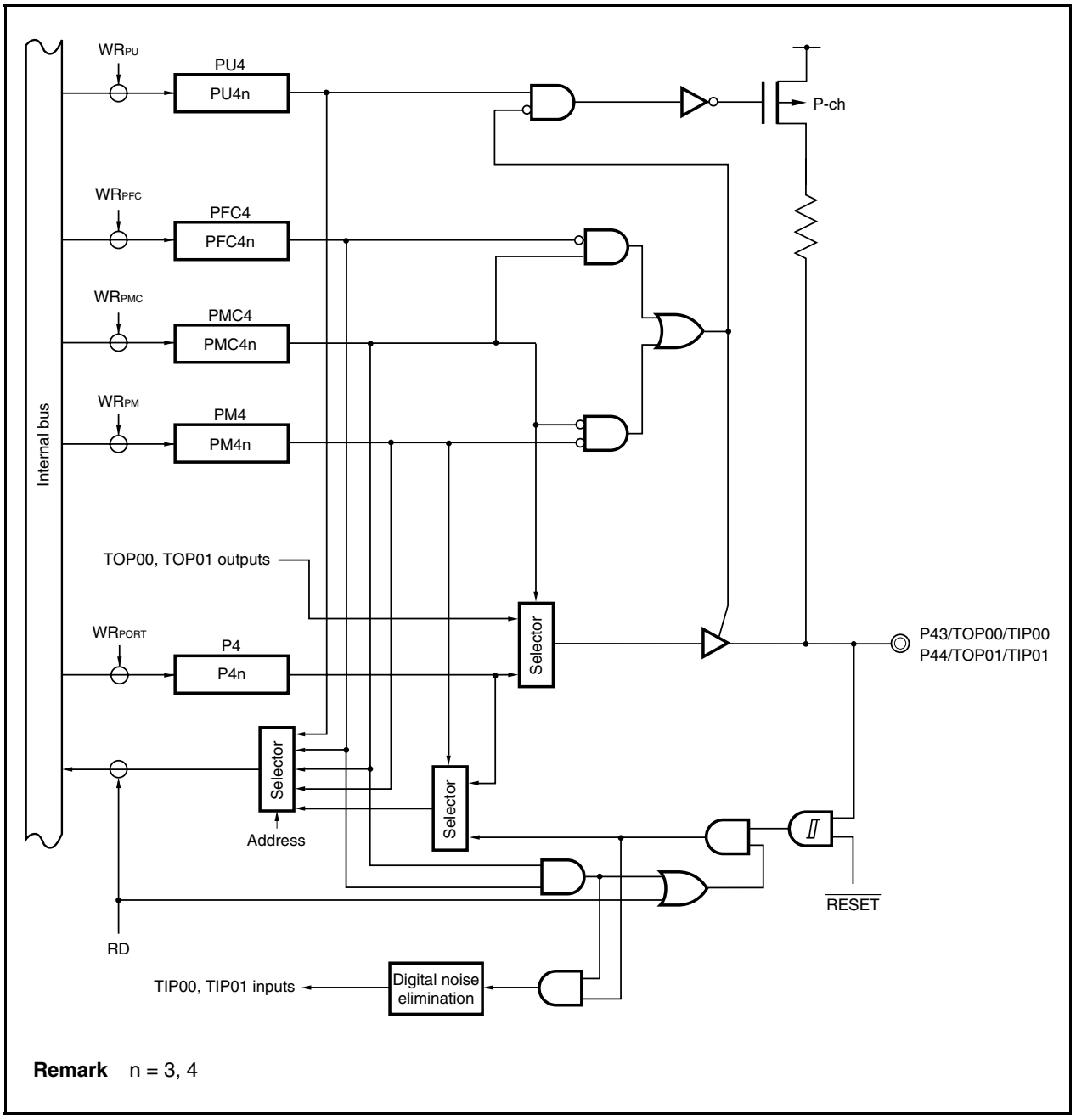


Figure 4-20. Block Diagram of P43 and P44 Pins



Remark n = 3, 4

4.3.6 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 4-9. Alternate-Function Pins of Port DL

Pin Name	Pin No.	Alternate-Function Pin	I/O	Pull-Up[Note 1]
PDL0	35	–	–	Provided
PDL1	34	–	–	
PDL2	33	–	–	
PDL3	32	–	–	
PDL4	31	–	–	
PDL5	30	FLMD1[Note 2]	–	
PDL6	29	–	–	
PDL7	28	–	–	

Notes 1. Software pull-up function

2. 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 18 FLASH MEMORY**.

(1) Registers

(a) Port DL register L (PDLL)

After reset: Undefined R/W Address: FFFFF004H

	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) (n = 0 to 7)
0	Output 0.
1	Output 1.

(b) Port DL mode register L (PMDLL)

After reset: FFH R/W Address: FFFFF024H

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

PMDLn	Control of input/output mode (in port mode) (n = 0 to 7)
0	Output mode
1	Input mode

(c) Pull-up resistor option register DLL (PUDLL)

After reset: 00H R/W Address: FFFFFF44H

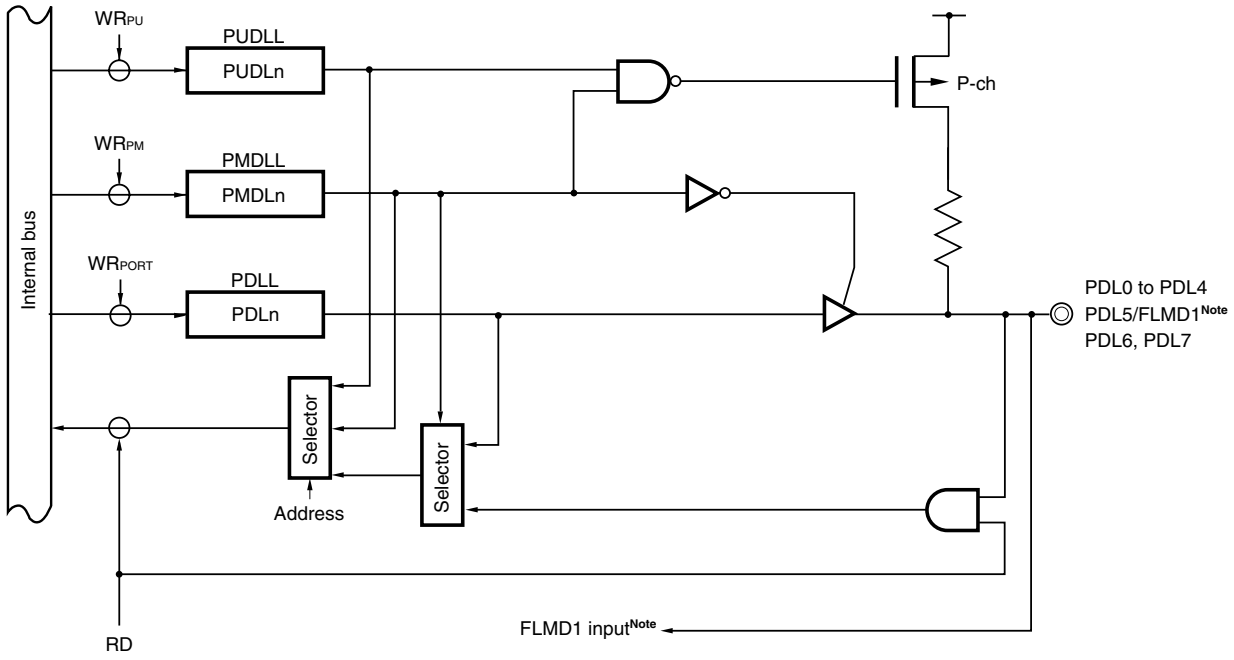
	7	6	5	4	3	2	1	0
PUDLL	PUDL7	PUDL6	PUDL5	PUDL4	PUDL3	PUDL2	PUDL1	PUDL0

PUDLn	Control of on-chip pull-up resistor connection (n = 0 to 7)
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. The resistor cannot be connected when the pin is in the output state.

(2) Block diagram

Figure 4-21. Block Diagram of PDL0 to PDL7 Pins



Note 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 18 FLASH MEMORY**.

4.4 Output Data and Read Value of Port for Each Setting

The following shows the values used to select the alternate function of the respective pins, output data and read value of each port for each setting. In addition to the settings shown here, setting of each peripheral function control register is necessary.

Table 4-10. Output Data and Port Read Value for Each Setting (1/4)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	Remark
P00 to P06	Output port	0	None	None	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	INTP0 to INTP5[Note], INTP6	1	None	None	0	–	Port latch	Necessary to specify valid edge
					1		Pin level	
P10, P12	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOQH01, TOQH02	1	0	0	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	TIQ01, TOQ03	1	0	1	0	–	Port latch	Alternate input (timer input)
					1		Pin level	
	TOQ01, TOQ03	1	1	0	0	Alternate output (timer output)	Port latch	
					1		Pin level	
P11	Output port	0	×	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TIQ02	1	0	1	0	–	Port latch	Alternate input (timer input)
					1		Pin level	
	TOQ02	1	1	0	0	Alternate output (timer output)	Port latch	
					1		Pin level	
P13	Output port	0	None	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TIQ00	1	None	1	0	–	Port latch	Alternate input (timer input)
					1		Pin level	

Note Including TOQH0OFF, TOQ1OFF, TOP2OFF, TOP3OFF, ADTRG0, and ADTRG1.

Remark ×: Don't care

Table 4-10. Output Data and Port Read Value for Each Setting (2/4)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	Remark
P14	Output port	0	None	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOQH03	1	None	0	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	EVTQ0	1	None	1	0	–	Port latch	Alternate input (timer input)
					1		Pin level	
P16 (CLMER)[Note], P17	Output port[Note]	0	None	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOQ00 (CLMER)[Note], TOP21	1	None	0	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	TIP20, TIP21	1	None	1	0	–	Port latch	Alternate input (timer input)
					1		Pin level	
P20 to P27	Output port	0	None	None	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOQ1T1 to TOQ1T3, TOQ1B1 to TOQ1B3, TOQ10, TOP31	1	None	None	0	Alternate output (timer output)	Port latch	
					1		Pin level	
P30	Output port	0	None	None	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	RXDA0	1	None	None	0	–	Port latch	Alternate input (serial input)
					1		Pin level	
P31	Output port	0	None	None	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TXDA0	1	None	None	0	Alternate output (serial output)	Port latch	
					1		Pin level	

Note When P16 is set as the output port or TOQ00 output function, if an error (oscillator stop) is detected by the clock monitor, the CLMER signal (low level) is output from P16. Low-level output is released by reset signal. For details, see **Table 4-5 Alternate-Function Pins of Port 1**.

Remark ×: Don't care

Table 4-10. Output Data and Port Read Value for Each Setting (3/4)

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	Remark
P32	Output port	0	None	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	RXDA1	1	None	1	0	–	Port latch	Alternate input (serial input)
					1		Pin level	
P33	Output port	0	None	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TXDA1	1	None	1	0	Alternate output (serial output)	Port latch	
					1		Pin level	
P40	Output port	0	None	None	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	SIB0	1	None	None	0	–	Port latch	Alternate input (serial input)
					1		Pin level	
P41	Output port	0	None	None	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	SOB0	1	None	None	0	Alternate output (serial output)	Port latch	
					1		Pin level	
P42	Output port	0	None	None	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	$\overline{\text{SCKB0}}$	1	None	None	0	Alternate output (serial output)	Port latch	Output in master mode Input in slave mode
					1		Pin level	

Remark ×: Don't care

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

Port Name	Function	PMCmn	PFCEmn	PFCmn	PMmn	Output Data	Pmn Read Value	Remark
P43, P44	Output port	0	None	×	0	Port latch	Port latch	
	Input port				1	–	Pin level	
	TOP00, TOP01	1	None	0	0	Alternate output (timer output)	Port latch	
					1		Pin level	
	TIP00, TIP01	1	None	1	0	–	Port latch	Alternate input (timer input)
					1		Pin level	
PDL0 to PDL4, PDL5[Note], PDL6, PDL7	Output port	None	None	None	0	Port latch	Port latch	
	Input port				1	–	Pin level	

Note The PDL5 pin also functions as a pin to be set in the flash programming mode. This pin does not need to be manipulated using the port control register. For details, see **CHAPTER 18 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 4-11. Using Port Pin as Alternate-Function Pin (1/3)

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	Other Bit (Register)
	Name	I/O						
P00	INTP0	Input	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	–	–	
	TOQH0OFF	Input	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	–	–	
P01	INTP1	Input	P01 = Setting not required	PM01 = Setting not required	PMC01 = 1	–	–	
	TOQ1OFF	Input	P01 = Setting not required	PM01 = Setting not required	PMC01 = 1	–	–	
P02	INTP2	Input	P02 = Setting not required	PM02 = Setting not required	PMC02 = 1	–	–	
	TOQ2OFF	Input	P02 = Setting not required	PM02 = Setting not required	PMC02 = 1	–	–	
P03	INTP3	Input	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	–	–	
	TOQ3OFF	Input	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	–	–	
P04	INTP4	Input	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	–	–	
	ADTRG0	Input	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	–	–	
P05	INTP5	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	–	–	
	ADTRG1	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	–	–	
P06	INTP6	Input	P06 = Setting not required	PM06 = Setting not required	PMC06 = 1	–	–	
P10	TOQH01	Output	P10 = Setting not required	PM10 = Setting not required	PMC10 = 1	PFCE10 = 0	PFC10 = 0	
	TIQ01	Input	P10 = Setting not required	PM10 = Setting not required	PMC10 = 1	PFCE10 = 0	PFC10 = 1	
	TOQ01	Output	P10 = Setting not required	PM10 = Setting not required	PMC10 = 1	PFCE10 = 1	PFC10 = 0	
P11	TIQ02	Input	P11 = Setting not required	PM11 = Setting not required	PMC11 = 1	PFCE11 = 0	PFC11 = 1	
	TOQ02	Output	P11 = Setting not required	PM11 = Setting not required	PMC11 = 1	PFCE11 = 1	PFC11 = 0	
P12	TOQH02	Output	P12 = Setting not required	PM12 = Setting not required	PMC12 = 1	PFCE12 = 0	PFC12 = 0	
	TIQ03	Input	P12 = Setting not required	PM12 = Setting not required	PMC12 = 1	PFCE12 = 0	PFC12 = 1	
	TOQ03	Output	P12 = Setting not required	PM12 = Setting not required	PMC12 = 1	PFCE12 = 1	PFC12 = 0	
P13	TIQ00	Input	P13 = Setting not required	PM13 = Setting not required	PMC13 = 1	–	PFC13 = 1	
P14	TOQH03	Output	P14 = Setting not required	PM14 = Setting not required	PMC14 = 1	–	PFC14 = 0	
	EVTQ0	Input	P14 = Setting not required	PM14 = Setting not required	PMC14 = 1	–	PFC14 = 1	

Table 4-11. Using Port Pin as Alternate-Function Pin (2/3)

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	Other Bit (Register)
	Name	I/O						
P16 (CLMER)[Note]	TOQ00 (CLMER)[Note]	Output	P16 = Setting not required	PM16 = Setting not required	PMC16 = 1	–	PFC16 = 0	
	TIP20	Input	P16 = Setting not required	PM16 = Setting not required	PMC16 = 1	–	PFC16 = 1	
P17	TOP21	Output	P17 = Setting not required	PM17 = Setting not required	PMC17 = 1	–	PFC17 = 0	
	TIP21	Input	P17 = Setting not required	PM17 = Setting not required	PMC17 = 1	–	PFC17 = 1	
P20	TOQ1T1	Output	P20 = Setting not required	PM20 = Setting not required	PMC20 = 1	–	–	
P21	TOQ1B1	Output	P21 = Setting not required	PM21 = Setting not required	PMC21 = 1	–	–	
P22	TOQ1T2	Output	P22 = Setting not required	PM22 = Setting not required	PMC22 = 1	–	–	
P23	TOQ1B2	Output	P23 = Setting not required	PM23 = Setting not required	PMC23 = 1	–	–	
P24	TOQ1T3	Output	P24 = Setting not required	PM24 = Setting not required	PMC24 = 1	–	–	
P25	TOQ1B3	Output	P25 = Setting not required	PM25 = Setting not required	PMC25 = 1	–	–	
P26	TOQ10	Output	P26 = Setting not required	PM26 = Setting not required	PMC26 = 1	–	–	
P27	TOP31	Output	P27 = Setting not required	PM27 = Setting not required	PMC27 = 1	–	–	
P30	RXDA0	Input	P30 = Setting not required	PM30 = Setting not required	PMC30 = 1	–	–	
P31	TXDA0	Output	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	–	–	
P32	RXDA1	Input	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	–	PFC32 = 1	
P33	TXDA1	Output	P33 = Setting not required	PM33 = Setting not required	PMC33 = 1	–	PFC33 = 1	
P40	SIB0	Input	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	–	–	
P41	SOB0	Output	P41 = Setting not required	PM41 = Setting not required	PMC41 = 1	–	–	
P42	$\overline{\text{SCKB0}}$	I/O	P42 = Setting not required	PM42 = Setting not required	PMC42 = 1	–	–	

Note When P16 is set as the output port or TOQ00 output function, if an error (oscillator stop) is detected by the clock monitor, the CLMER signal (low level) is output from P16. Low-level output is released by reset signal. For details, see **Table 4-5 Alternate-Function Pins of Port 1**.

Table 4-11. Using Port Pin as Alternate-Function Pin (3/3)

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	Other Bit (Register)
	Name	I/O						
P43	TOP00	Output	P43 = Setting not required	PM43 = Setting not required	PMC43 = 1	–	PFC43 = 0	
	TIP00	Input	P43 = Setting not required	PM43 = Setting not required	PMC43 = 1	–	PFC43 = 1	
P44	TOP01	Output	P44 = Setting not required	PM44 = Setting not required	PMC44 = 1	–	PFC44 = 0	
	TIP01	Input	P44 = Setting not required	PM44 = Setting not required	PMC44 = 1	–	PFC44 = 1	
PDL0	–	–	PDL0 = Setting not required	PMDL0 = Setting not required	–	–	–	
PDL1	–	–	PDL1 = Setting not required	PMDL1 = Setting not required	–	–	–	
PDL2	–	–	PDL2 = Setting not required	PMDL2 = Setting not required	–	–	–	
PDL3	–	–	PDL3 = Setting not required	PMDL3 = Setting not required	–	–	–	
PDL4	–	–	PDL4 = Setting not required	PMDL4 = Setting not required	–	–	–	
PDL5[Note]	FLMD1[Note]	Input	PDL5 = Setting not required	PMDL5 = Setting not required	–	–	–	
PDL6	–	–	PDL6 = Setting not required	PMDL6 = Setting not required	–	–	–	
PDL7	–	–	PDL7 = Setting not required	PMDL7 = Setting not required	–	–	–	

Note The PDL5 pin also functions as a pin to be set in the flash programming mode (FLMD1). This pin does not need to be manipulated using the port control register. For details, see **CHAPTER 18 FLASH MEMORY**.

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.

<R>

Table 4-12. Noise Eliminator

Unit	Target Pin	Delay Type	Noise Elimination Width	Sampling Clock
Reset	$\overline{\text{RESET}}$	Analog delay	Several 10 ns (TYP.)	–
Mode pin	FLMD0			
• Interrupt (INTC)[Note] • High impedance output control function of timer for motor control • A/D converter (ADC)	INTP0/TOQH0OFF INTP1/TOQ1OFF INTP2/TOP2OFF INTP3/TOP3OFF INTP4/ADTRG0 INTP5/ADTRG1		500 ns (MIN.)	
	INTP6	Digital delay	4 to 5 clocks (400 ns (at 20 MHz)) (800 ns (at 20 MHz)) (1.6 μs (at 20 MHz)) (3.2 μs (at 20 MHz))	f_{XX}/2 (100 ns (at 20 MHz)) f_{XX}/4 (200 ns (at 20 MHz)) f_{XX}/8 (400 ns (at 20 MHz)) f_{XX}/16 (800 ns (at 20 MHz)) selectable
Timer Q (TMQ)	TIQ00 TIQ01 TIQ02 TIQ03 EVTQ0		4 to 5 clocks (200 ns (at 20 MHz))	f_{XX} (50 ns (at 20 MHz))
Timer P (TMP)	TIP00 TIP01 TIP20 TIP21			

Note A maskable interrupt input other than INTP6 can be used as the release source of IDLE or STOP mode.

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

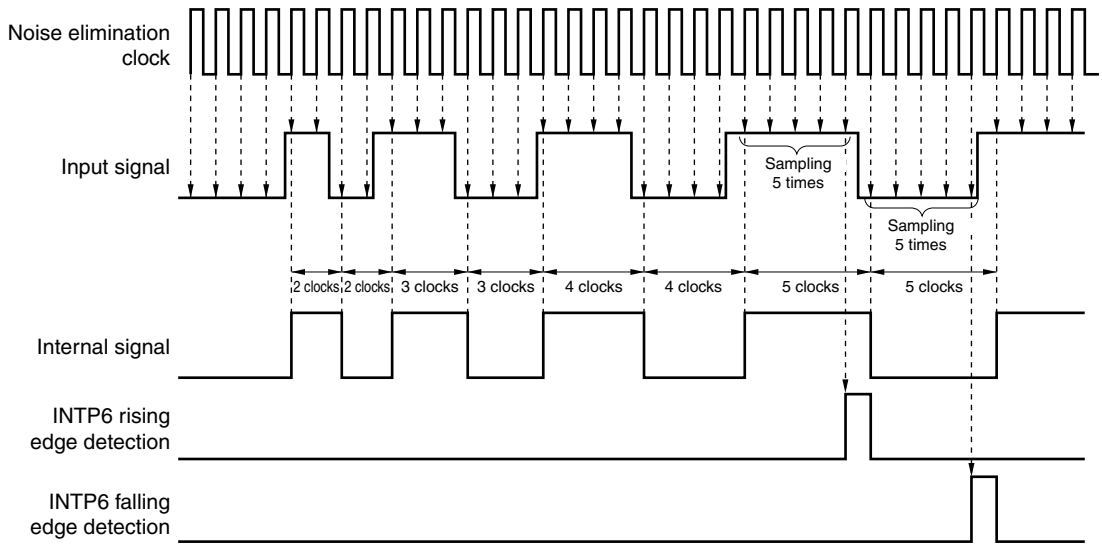
2. The noise filter of the digital delay pin uses clock sampling and therefore cannot acknowledge an input signal when the peripheral clock (f_{XX}) is stopped.

3. The noise eliminator operates only when the alternate function is used.

The following shows an example of digital noise elimination timing of the INTP6 pin.

<R>

Figure 4-22. Example of Noise Elimination Timing



Caution If there are four or less noise elimination clocks while the INTP6 input signal is high level (or low level), the input pulse is eliminated as noise. If it is sampled at least five times, the edge is detected as valid input.

<R> **(1) External interrupt noise elimination control register (INTPNRC)**

The INTPNRC register is used to select the sampling clock that is used to eliminate digital noise on the INTP6 pin. If the same level is not detected five times in a row, the signal is eliminated as noise.

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

Reset sets this register to 00H.

Cautions 1. If the input pulse lasts for the duration of 4 to 5 clocks, it is undefined whether the pulse is detected as a valid edge or eliminated as noise. So that the pulse is actually detected as a valid edge, the same pulse level must be input for the duration of 5 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: FFFFF310H

	7	6	5	4	3	2	1	0
INTPNRC	0	0	0	0	0	0	INTPNRC1	INTPNRC0

INTPNRC1	INTPNRC0	Selection of sampling clock
0	0	$f_{XX}/16$
0	1	$f_{XX}/8$
1	0	$f_{XX}/4$
1	1	$f_{XX}/2$

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 PFCn register is set before setting the PMCn 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, an on-chip pull-up resistor can be connected to the TOP21, TOQ1T1 to TOQ1T3, TOQ1B1 to TOQ1B3, TOP31, and TOQH01 to TOQH03 output pins when these pins go into a high-impedance state due to the TOQ1OFF, TOP2OFF, TOP3OFF, or TOQH0OFF pin or software processing.

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 P90 pin is changed from low level to high level via a bit manipulation instruction, the value of the port latch is FFH.

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

A bit manipulation instruction is executed in the following order in the V850ES/IE2.

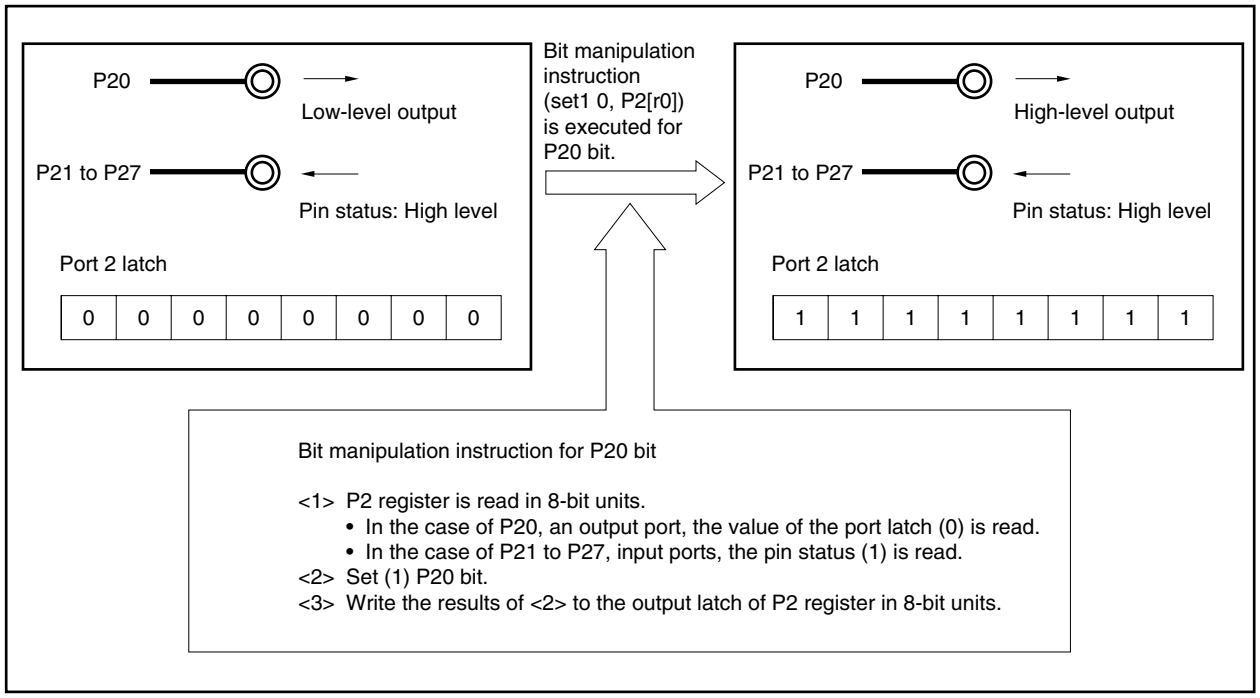
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

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 4-23. Bit Manipulation Instruction (P20 Pin)



CHAPTER 5 CLOCK GENERATOR

5.1 Overview

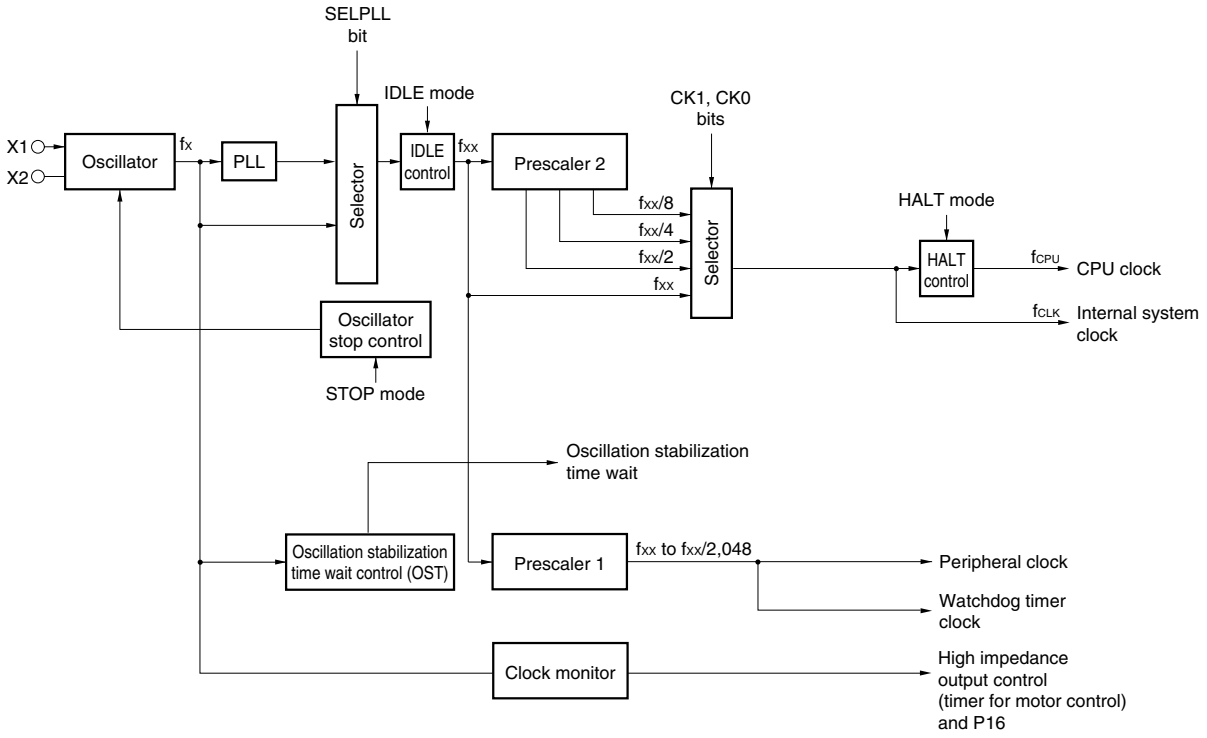
The features of clock generator are as follows.

- ○ Oscillator
  - In PLL mode: f_X = 2.5 MHz (f_{XX} = 20 MHz)
  - In clock-through mode: f_X = 2.5 MHz (f_{XX} = 2.5 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

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

5.2 Configuration

Figure 5-1. Clock Generator



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 = 2.5 MHz, f_{CPU} and f_{CLK} are 312.5 kHz.

Remark f_X: Oscillation frequency
f_{XX}: System clock frequency
f_{CPU}: CPU clock frequency
f_{CLK}: Internal system 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)
Interrupt controller	f_{CLK} (Selected from f_{XX} to $f_{XX}/8$ by PCC register)
Timer (excluding watchdog timer)	f_{XX}
Watchdog timer	$f_{XX}/512$
UARTA	f_{UCLK} (Selected from f_{XX} to $f_{XX}/2,048$ by UAnCTL1 register)
CSIB	f_{CCLK} (Selected from $f_{XX}/2$ to $f_{XX}/128$ and external clock by CB0CTL1 register)
A/D converters 0, 1	f_{XX}

Remarks 1. f_{XX}: Peripheral clock
2. n = 0, 1

(1) Oscillator

The main resonator oscillates the following frequencies (f_X):

- In PLL mode (×8 fixed): f_X = 2.5 MHz (f_{XX} = 20 MHz)
- In clock-through mode: f_X = 2.5 MHz (f_{XX} = 2.5 MHz)

(2) IDLE control

All functions other than the oscillator, PLL, clock monitor operation, and CSIB in slave mode 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.

The output frequency of PLL is 20 MHz in the PLL mode.

(5) Prescaler 1

This prescaler generates the clock (f_{XX} to f_{XX}/2,048) 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 $2^{16}/f_X$.

(8) Clock monitor

The clock monitor samples the clock generated by the oscillator (f_X), by using the internal oscillation clock. When it detects an error (stop of oscillation), the output of the motor control timer goes into a high-impedance state. The CLMER signal (low level)[Note] is output from P16 (for details, see **CHAPTER 4 PORT FUNCTIONS** and **CHAPTER 9 MOTOR CONTROL FUNCTION)**. Low-level output is released by reset signal.

Note The CLMER signal (low level) is valid only when P16 is specified as an output port or the TOQ00 output function.

5.3 Control Registers

The clock generator is controlled by the following seven registers.

- PLL control register (PLLCTL)
- Clock control register (CKC)
- 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 clear 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)
```

3. Before using the PLL mode (SELPLL bit = 1), be sure to set the CKC register to 0BH. Unless the CKC register has been set, the operation is not guaranteed.

(2) Clock control register (CKC)

The CKC register is used to control the PLL mode. Before using the PLL mode (PLLCTL.SELPLL bit = 1), be sure to set the CKC register to 0BH. Unless the CKC register has been set, the operation is not guaranteed. The CKC register is a special register. Data can be written to this register only in a combination of specific sequences (see **3.4.7 Special registers**).

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

Reset sets this register to 0AH.

After reset: 0AH R/W Address: FFFFF822H

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

Cautions 1. The default value of the CKC register is 0AH. However, be sure to set 0BH to this register during initial settings. The default value of an in-circuit emulator is undefined, but also set 0BH to the register when an in-circuit emulator is used.

<R> **2. When writing to the CKC register, use command register PRCMD.**

(3) 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 **3.4.7 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 clear bits 2 to 7 to "0".

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

<R> **3. When writing to the PCC register, use command register PRCMD.**

(4) Power save control register (PSC)

The PSC register is a special register. Data can be written to this register only in a combination of specific sequences (see **3.4.7 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: FFFFF1FEH

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

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

STB	Operation mode setting
0	Normal mode
1	Standby mode

Note For details, see **Table 14-1 Interrupt Source List**.

Cautions 1. Be sure to clear 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.

<R> **4. When writing to the PSC register, use command register PRCMD.**

(5) 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	Specification of operation in software standby mode
0	IDLE mode
1	STOP mode

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

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

(6) 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.

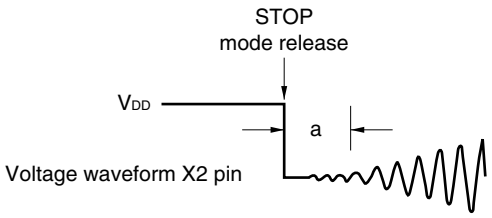
Reset sets this register to 06H.

After reset: 06H R/W Address: FFFFF6C0H

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

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time (f_X = 2.5 MHz)
1	0	0	$2^{14}/f_X$ (6.55 ms)
1	0	1	$2^{15}/f_X$ (13.1 ms)
1	1	0	$2^{16}/f_X$ (26.2 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. 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. The oscillation stabilization time after reset is $2^{16}/f_X$ (because the default value of the OSTS register is 06H).

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

Remark f_X: Oscillation frequency

(7) Clock monitor mode register (CLM)

The CLM register sets the clock monitor operation mode. It can be written only in a combination of specific sequences (see **3.4.7 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**

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 9-4 for the target timer output.

<R> **4. When writing to the CLM register, use command register PRCMD.**

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) = 2.5 MHz, output clock (f_{XX}) = 20 MHz
Clock-through mode: Input clock (f_X) = 2.5 MHz, output clock (f_{XX}) = 2.5 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 V850ES/IE2, the lockup time after release of reset is secured automatically.

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.

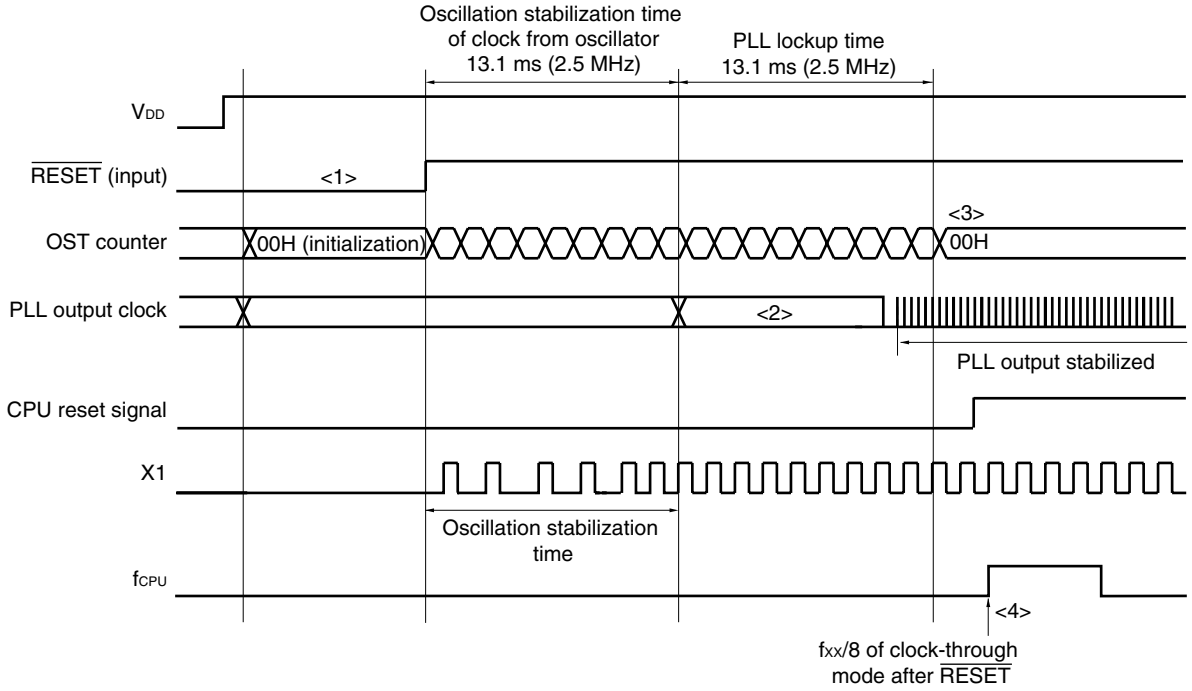
Table 5-2. Operation Status of Each Clock

Power Save Mode	Oscillator (f_X)	PLL (f_{XX})	Internal System Clock (f_{CLK})	CPU Clock (f_{CPU})	Peripheral Clock (f_{XX} to $f_{XX}/2,048$)	Watchdog Timer Clock ($f_{XX}/512$)
Normal operation	√	√	√	√	√	√
During $\overline{\text{RESET}}$ pin input	×	×	×	×	×	×
During oscillation stabilization time count	√	√	×	×	×	×
HALT mode	√	√	√	×	√	√
IDLE mode	√	√	×	×	×	×
STOP mode	×	×	×	×	×	×

Remark √: Operating
×: Stopped

5.5.2 Operation timing

(1) Power on (power-on reset)



<1> The oscillator is activated by $\overline{\text{RESET}}$ release power application.
PLL stops during the $\overline{\text{RESET}}$ period and the oscillation stabilization time set using the OSTS register.

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

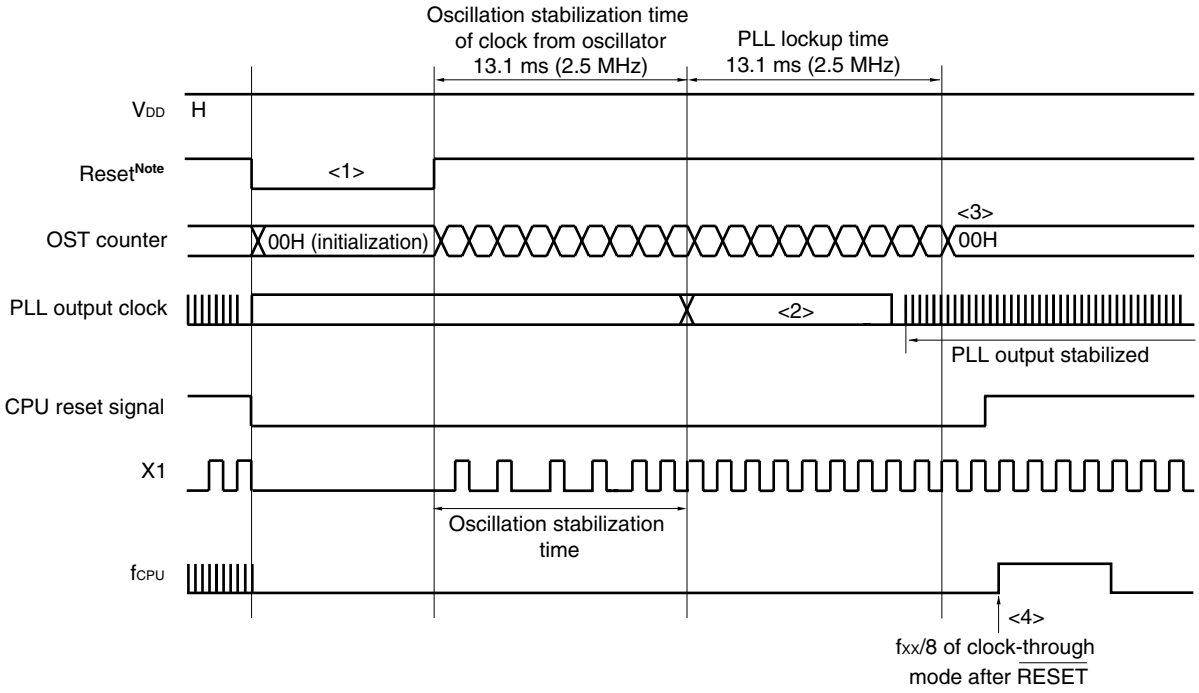
<3> PLL is locked when counting of the lockup time is over. The OST counter is initialized to 00H.

<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. After the $\overline{\text{RESET}}$ signal is released, a specific wait time (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> PLL stops during the reset period and the oscillation stabilization time set using the OSTS register.

<2> When the 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.

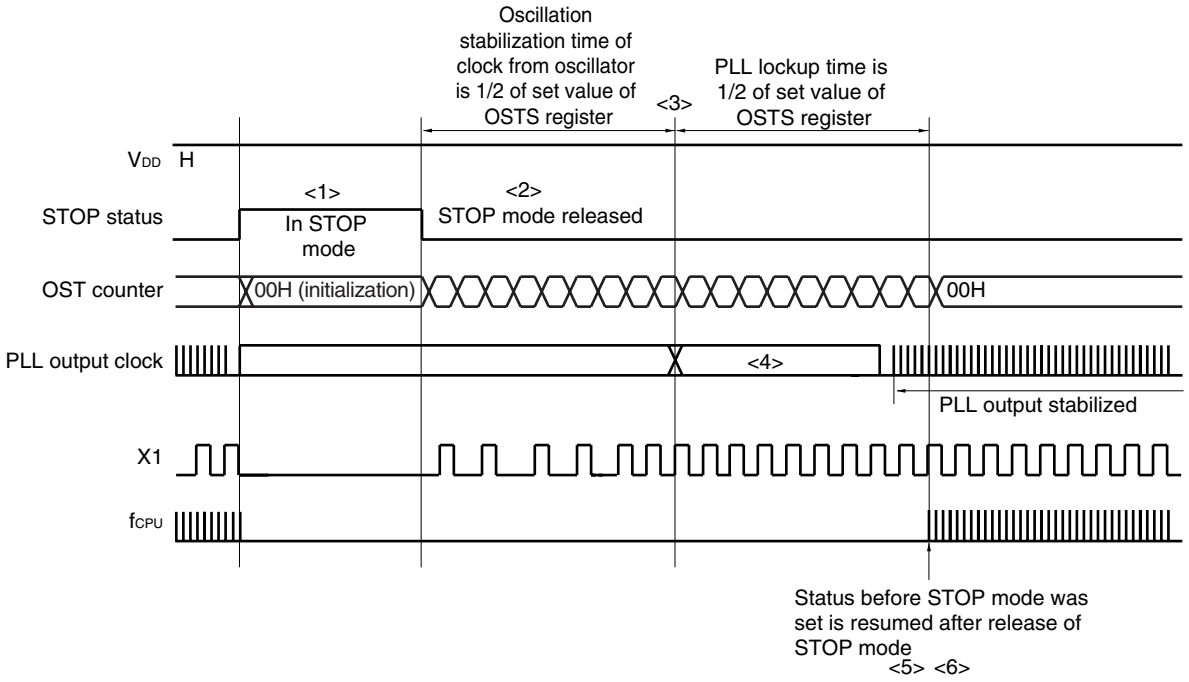
<4> When the lockup time expires, the CPU releases the reset signal and operates in the clock-through mode (f_X). The CPU operation clock (f_{CPU}) is $f_{XX}/8$. The PLL mode can be set by software.

Note $\overline{\text{RESET}}$ pin input, or WDTRES, LVIRES, or POCRES signal generation

Cautions 1. After the reset signal is released, a specific wait time (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 $\overline{\text{RESET}}$ input or LVIRES or POCRES signal generation is the same as that in **(1) Power on (power-on reset)** and **(2) Reset input with power on**.

5.6 Clock Monitor

(1) Function

The clock monitor samples the clock generated by the oscillator (fx), by using the internal oscillation clock. When it detects an error (stop of oscillation), the output of the motor control timer goes into a high-impedance state. The CLMER signal (low level)[Note] is output from P16 (for details, see **CHAPTER 4 PORT FUNCTIONS** and **CHAPTER 9 MOTOR CONTROL FUNCTION)**. Low-level output is released by reset signal.

Note The CLMER signal (low level) is valid only when P16 is set to the output port or TOQ00 output function.

(2) Operation

The following explains the functions of the clock monitor. The operation start and operation stop conditions are as follows.

<Operation start condition>

Setting the CLM.CLME bit to enable (1)

<Operation stop condition>

While oscillation stabilization time is being counted after STOP mode is released

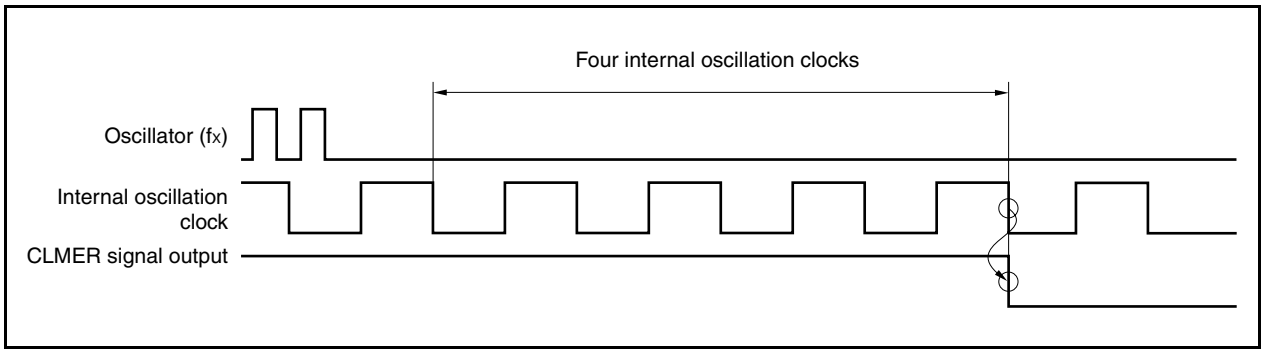
Table 5-3. Operation Status of Clock Monitor (When CLM.CLME Bit = 1)

Operation Mode	Status of Clock (fx) Generated by Oscillator	Status of Internal Oscillation Clock	Status of Clock Monitor
Normal operation mode	Oscillates	Oscillates	Operates
HALT mode	Oscillates	Oscillates	Operates
IDLE mode	Oscillates	Oscillates	Operates
STOP mode	Stops	Oscillates	Stops
During reset	Stops	Stops	Stops

(a) Operation when oscillator is stopped (CLM.CLME bit = 1)

If the oscillator is stopped when the CLME bit = 1, the CLMER signal is output from P16.

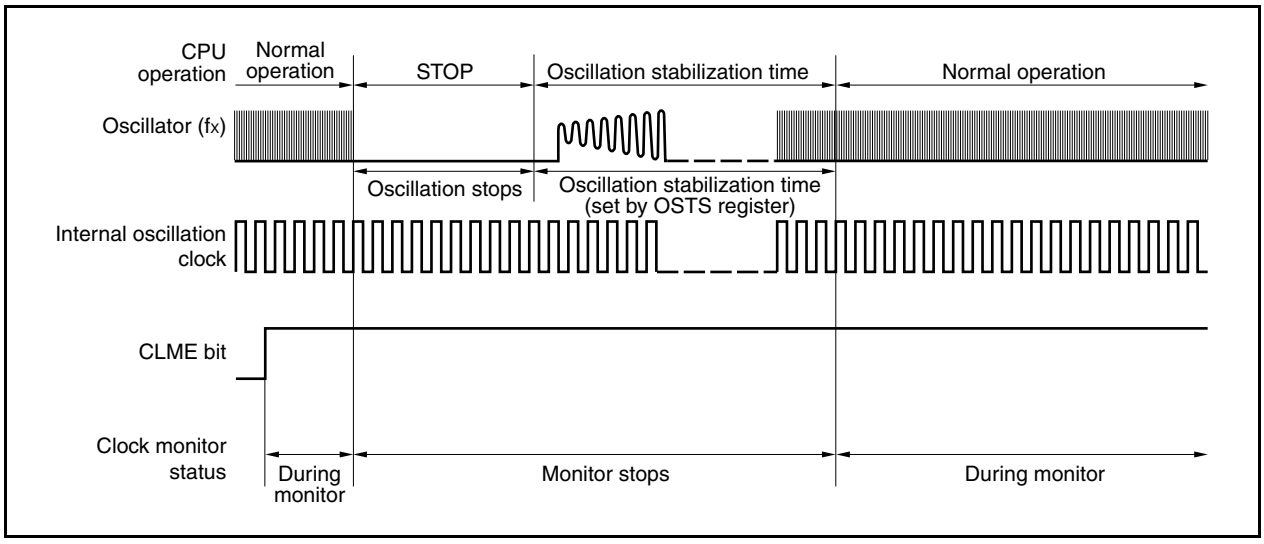
Figure 5-2. When Oscillation of Main Clock Is Stopped



(b) Operation in STOP mode or after STOP mode is released

If the STOP mode is set with the CLME bit = 1, the monitor operation is stopped in the STOP mode and while the oscillation stabilization time is being counted. After the oscillation stabilization time, the monitor operation is automatically started.

Figure 5-3. Operation in STOP Mode or After STOP Mode Is Released



CHAPTER 6 16-BIT TIMER/EVENT COUNTER P (TMP)

Timer P (TMP) is a 16-bit timer/event counter.
The V850ES/IE2 incorporates TMP0 to TMP3.

6.1 Overview

The TMPn of channels are outlined below (n = 0 to 3).

Table 6-1. TMPn Overview

Item	TMP0	TMP1	TMP2	TMP3
Clock selection	8 ways	8 ways	8 ways	8 ways
Capture trigger input pin	2	None	2	None
External event count input pin	1	None	1	None
External trigger input pin	1	None	1	None
Timer counter	1	1	1	1
Capture/compare register	2	2 Note	2	2 Note
Capture/compare match interrupt request signal	2	2 Note	2	2 Note
<R> Overflow interrupt request signal	1	1	1	1
Timer output pin	2	None	1	1

Note Compare function only

6.2 Functions

The functions of TMPn that can be realized differ from one channel to another, as shown in the table below (n = 0 to 3).

Table 6-2. TMPn Functions

Function	TMP0	TMP1	TMP2	TMP3
Interval timer	√	√	√	√
External event counter	√	×	√	×
External trigger pulse output	√	×	√	√ Note
One-shot pulse output	√	×	√	√ Note
PWM output	√	×	√	√
Free-running timer	√	√	√	√
Pulse width measurement	√	×	√	×
Timer tuning operation	×	√ (TMQ1)	×	×

Note Realized by software trigger only. External trigger input cannot be used.

6.3 Configuration

TMPn includes the following hardware.

Table 6-3. Configuration of TMPn

Item	Configuration
Timer register	16-bit counter × 1
Registers	TMPn capture/compare registers 0, 1 (TPnCCR0, TPnCCR1) TMPn counter read buffer register (TPnCNT) CCR0 and CCR1 buffer registers
Timer input	Total 4 (TIP00[Note 1], TIP01, TIP20[Note 1], TIP21 pins)[Note 2]
Timer output	Total 4 (TOP00, TOP01, TOP21, TOP31 pins)[Note 3]
Control registers	TMPn control registers 0, 1 (TPnCTL0, TPnCTL1) TMPm I/O control register 0 (TPmIOC0) TMPk I/O control registers 1, 2 (TPkIOC1, TPkIOC2) TMPn option register 0 (TPnOPT0)

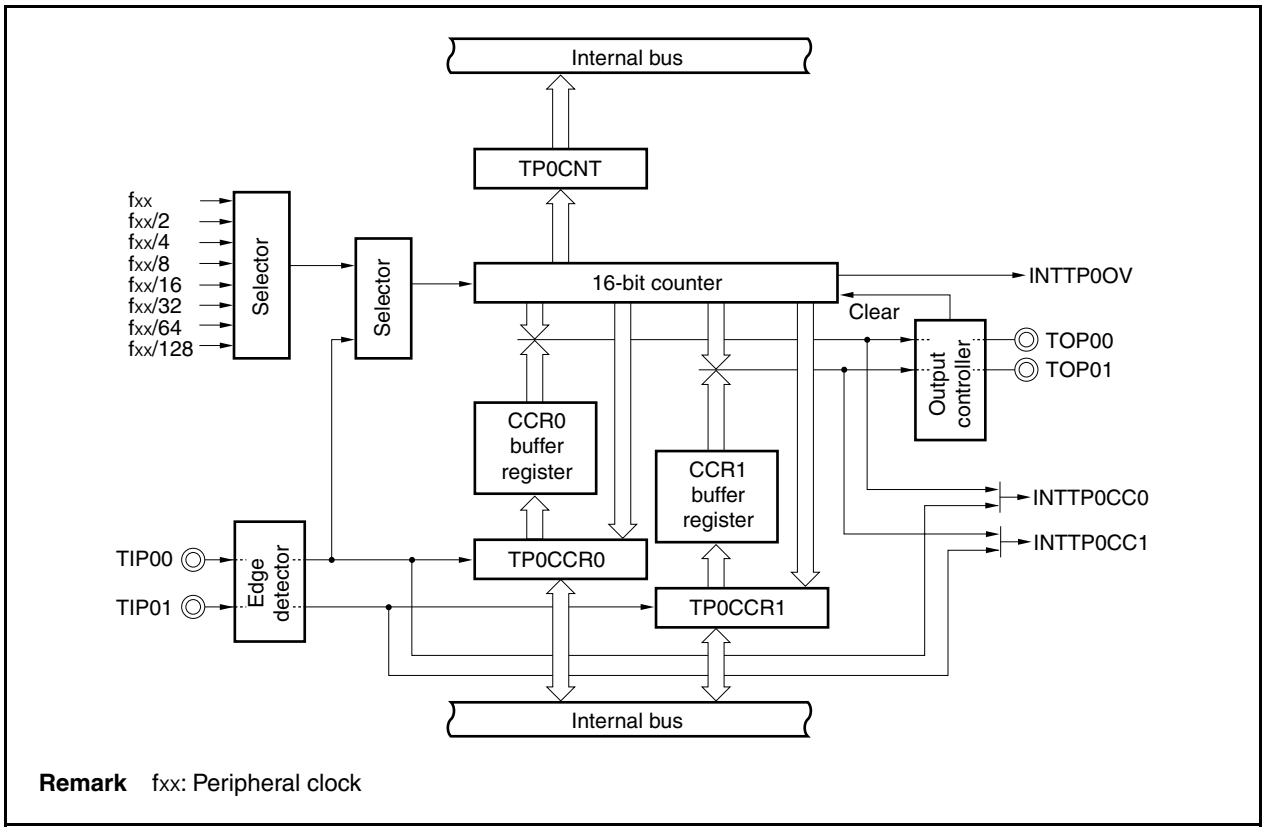
Notes 1. The TIP00 and TIP20 pins function alternately as a capture input signal, external event count input signal, and external trigger input signal.

2. Not provided for TMP1 and TMP3

3. Not provided for TMP1

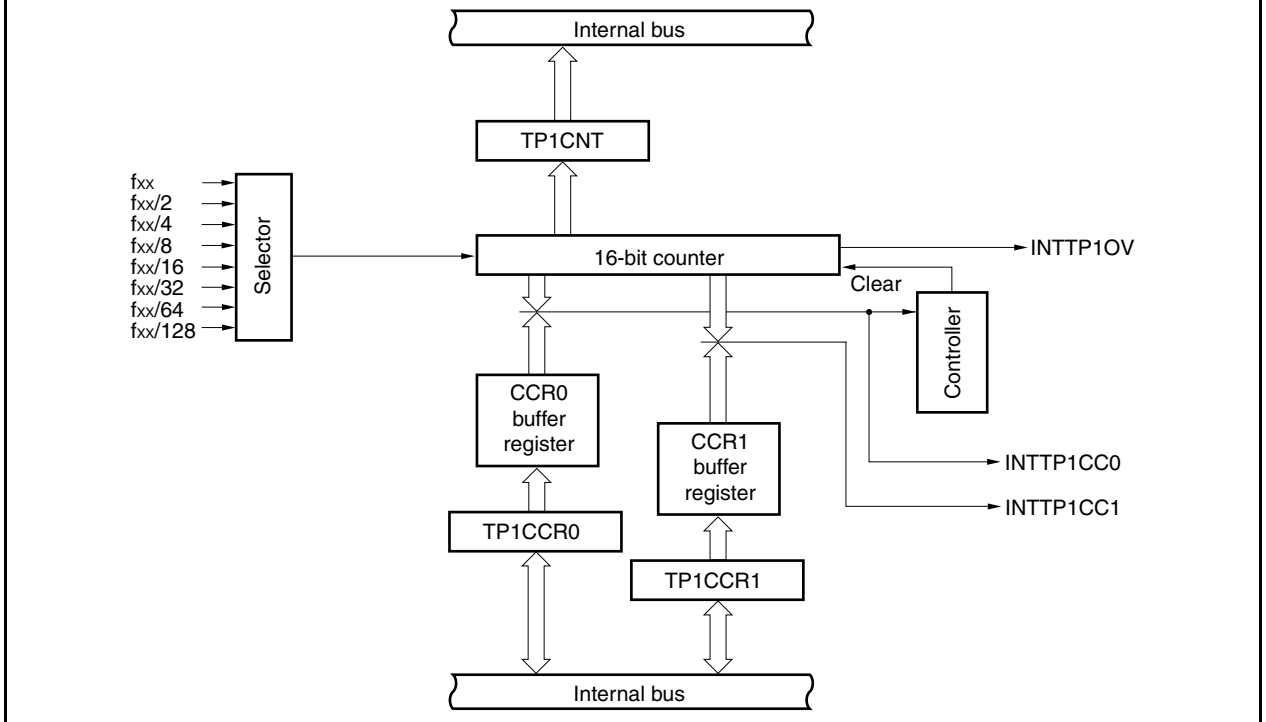
Remark n = 0 to 3
m = 0, 2, 3
k = 0, 2

Figure 6-1. TMP0 Block Diagram



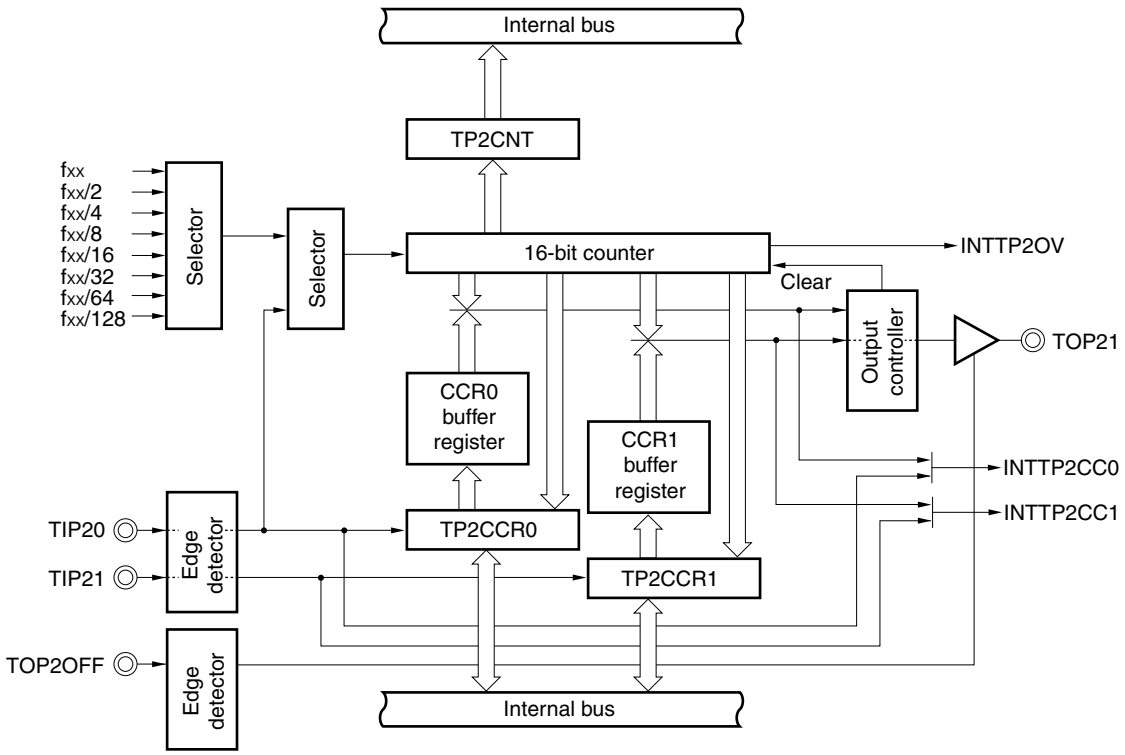
Remark fxx: Peripheral clock

Figure 6-2. TMP1 Block Diagram



Remark fxx: Peripheral clock

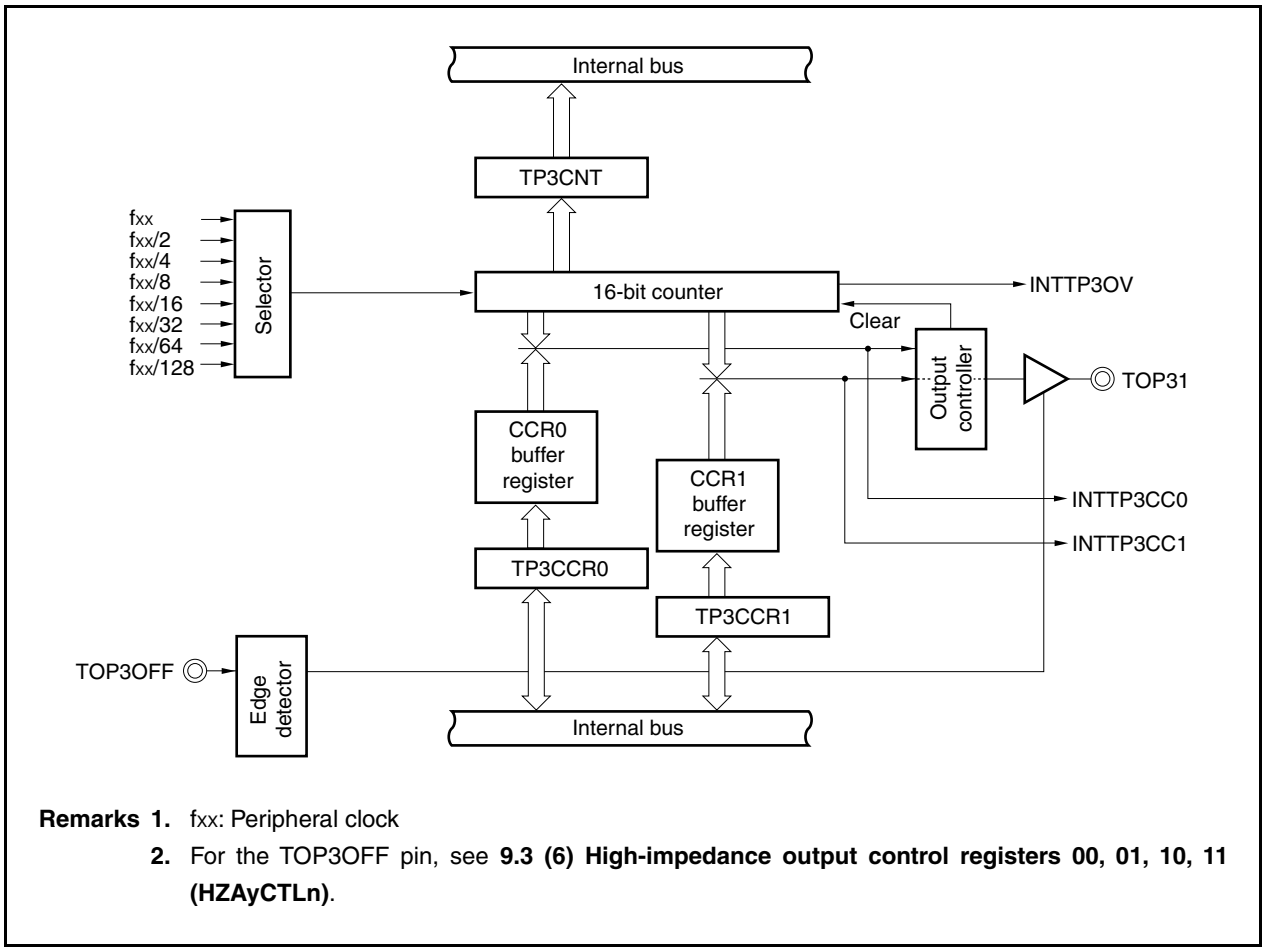
Figure 6-3. TMP2 Block Diagram



Remarks 1. fxx: Peripheral clock

2. For the TOP2OFF pin, see **9.3 (6) High-impedance output control registers 00, 01, 10, 11 (HZAyCTLn)**.

Figure 6-4. TMP3 Block Diagram



Remarks 1. fxx: Peripheral clock

2. For the TOP3OFF pin, see **9.3 (6) High-impedance output control registers 00, 01, 10, 11 (HZAyCTLn)**.

(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 TPnCNT register.

When the TPnCTL0.TPnCE bit = 0, the value of the 16-bit counter is FFFFH. If the TPnCNT register is read at this time, 0000H is read.

The TPnCE bit is cleared to 0 after reset.

(2) CCR0 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TPnCCR0 register is used as a compare register, the value written to the TPnCCR0 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 (INTTPnCC0) is generated.

The CCR0 buffer register cannot be read or written directly.

The TPnCCR0 register is cleared to 0000H after reset, and the CCR0 buffer 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 TPnCCR1 register is used as a compare register, the value written to the TPnCCR1 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 (INTTPnCC1) is generated.

The CCR1 buffer register cannot be read or written directly.

The TPnCCR1 register is cleared to 0000H after reset, and the CCR1 buffer register is cleared to 0000H.

(4) Edge detector

This circuit detects the valid edges input to the TIP00, TIP01, TIP20, and TIP21 pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TP0IOC1, TP2IOC1, TP0IOC2, and TP2IOC2 registers.

(5) Output controller

This circuit controls the output of the TOP00, TOP01, TOP21, and TOP31 pins. The output of the TOP00, TOP01, TOP21, and TOP31 pins is controlled by the TP0IOC0, TP2IOC0, and TP3IOC0 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) TMPn control register 0 (TPnCTL0)

The TPnCTL0 register is an 8-bit register that controls the operation of TMPn.

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 TPnCTL0 register by software.

After reset: 00H R/W Address: TP0CTL0 FFFFF640H, TP1CTL0 FFFFF660H, TP2CTL0 FFFFF680H, TP3CTL0 FFFFF6A0H

TPnCTL0 (n = 0 to 3, m = 0, 2, 3)

<7>	6	5	4	3	2	1	0
TPnCE	0	0	0	0	TPnCKS2	TPnCKS1	TPnCKS0

TPnCE	TMPn operation control
0	TMPn operation disabled (TMPn reset asynchronously[Note])
1	TMPn operation enabled. TMPn operation start

TPnCKS2	TPnCKS1	TPnCKS0	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 TPnOPT0.TPnOVF bit and 16-bit counter are reset simultaneously. Moreover, timer outputs (TOP00, TOP01, TOP21, and TOP31 pins) are reset to the TPmIOC0 register set status at the same time as the 16-bit counter.

Cautions 1. Set the TPnCKS2 to TPnCKS0 bits when the TPnCE bit = 0.
When the value of the TPnCE bit is changed from 0 to 1, the TPnCKS2 to TPnCKS0 bits can be set simultaneously.
2. Be sure to clear bits 3 to 6 to "0".

Remark f_{XX}: Peripheral clock

(2) TMPn control register 1 (TPnCTL1)

The TPnCTL1 register is an 8-bit register that controls the TMPn 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: TP0CTL1 FFFFF641H, TP1CTL1 FFFFF661H, TP2CTL1 FFFFF681H, TP3CTL1 FFFFF6A1H

	7	6	5	4	3	2	1	0
TPnCTL1	TP1SYE[Note 1]	TPmEST[Note 2]	TPkEEE[Note 3]	0	0	TPnMD2	TPnMD1	TPnMD0

(n = 0 to 3
m = 0, 2, 3
k = 0, 2)

TP1SYE[Note 1]	Operation mode selection
0	TMP1 single mode
1	Tuning operation mode (see **9.4.5**)
TMP1 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 TMQ1.	

TPmEST[Note 2]	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 TPmEST bit as the trigger. • In external trigger pulse output mode: A PWM waveform is output with writing 1 to the TPmEST bit as the trigger.
Read value of the TPmEST bit is always 0.	

TPkEEE[Note 3]	Count clock selection
0	Disable operation with external event count input (TIPk0 pin). (Perform counting with the count clock selected by the TPkCTL0.TPkCKS0 to TPkCTL0.TPkCKS2 bits.)
1	Enable operation with external event count input (TIPk0 pin)[Note 4]. (Perform counting at the valid edge of the external event count input signal.)
The TPkEEE bit selects whether counting is performed with the internal count clock or the valid edge of the external event count input.	

Notes 1. This bit can only be set in TMP1. Be sure to clear bit 7 of TMP0, TMP2, and TMP3 to 0. For details of tuning operation mode, see **CHAPTER 9 MOTOR CONTROL FUNCTION**.

2. This bit can only be set in TMP0, TMP2, and TMP3. Be sure to clear bit 6 of TMP1 to 0.

3. This bit can only be set in TMP0 and TMP2. Be sure to clear bit 5 of TMP1 and TMP3 to 0.

4. Set the valid edge selection of capture trigger input (TIPk0 pin) to “No edge detection”.

(2/2)

TPnMD2	TPnMD1	TPnMD0	Timer mode selection[Note]
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	Setting prohibited

Note The settings that can be realized differ from one channel to another. For details, see **Tables 6-8** to **6-11**.

Cautions 1. The TPmEST 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 TPkEEE bit.

<R>

3. Set the TP1SYE, TPkEEE and TPnMD2 to TPnMD0 bits when the TPnCTL0.TPnCE bit = 0. (The same value can be written when the TPnCE bit = 1.) The operation is not guaranteed when rewriting is performed with the TPnCE bit = 1. If rewriting was mistakenly performed, clear the TPnCE bit to 0 and then set the bits again.

4. Be sure to clear bits 3 and 4 to "0".

(3) TMPm I/O control register 0 (TPmIOC0)

The TPmIOC0 register is an 8-bit register that controls the timer output (TOP00, TOPm1 pins).

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

Reset sets this register to 00H.

Remark TMP1 does not have the TP1IOC0 register.

(1/2)

After reset: 00H R/W Address: TP0IOC0 FFFFF642H, TP2IOC0 FFFFF682H, TP3IOC0 FFFFF6A2H

	7	6	5	4	3	<2>	1	<0>
TPmIOC0	0	0	0	0	TPmOL1	TPmOE1	TP0OL0Note 1	TP0OE0Note 1

(m = 0, 2, 3)

TPmOL1	TOPm1 pin output level settingNote 2
0	TOPm1 pin starts output at high level.
1	TOPm1 pin starts output at low level.

TPmOE1	TOPm1 pin output setting
0	Timer output prohibited • Low level is output from the TOPm1 pin when the TPmOL1 bit = 0. • High level is output from the TOPm1 pin when the TPmOL1 bit = 1.
1	Timer output enabled (A pulse is output from the TOPm1 pin.)

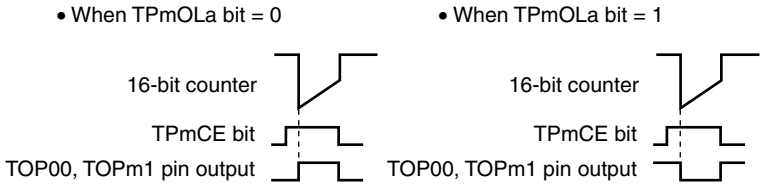
TP0OL0Note 1	TOP00 pin output level settingNote 2
0	TOP00 pin starts output at high level.
1	TOP00 pin starts output at low level.

TP0OE0Note 1	TOP00 pin output setting
0	Timer output prohibited • Low level is output from the TOP00 pin when the TP0OL0 bit = 0. • High level is output from the TOP00 pin when the TP0OL0 bit = 1.
1	Timer output enabled (A pulse is output from the TOP00 pin.)

Notes 1. Valid only for TMP0. Be sure to clear bits 1 and 0 of TMP2 and TMP3 to 0.

2. The output level of the timer output pins (TOP00 and TOPm1) specified by the TPmOLa (a = 0, 1) bit is shown below (a = 0, 1).

<R>



(2/2)

<R> **Cautions 1. If the setting of the TPmIOC0 register is changed when TOP00 and TOPm1 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 TPmOL1, TPmOE1, TP0OL0, and TP0OE0 bits when the TPmCTL0.TPnCE bit = 0. (The same value can be written when the TPmCE bit = 1.) If rewriting was mistakenly performed, clear the TPmCE bit to 0 and then set the bits again.

3. Even if the TP0OL0 or TPmOL1 bit is manipulated when the TPmCE, TP0OE0, and TPmOE1 bits are 0, the output level of the TOP00 and TOPm1 pins changes.

(4) TMPk I/O control register 1 (TPkIOC1)

The TPkIOC1 register is an 8-bit register that controls the valid edge for the capture trigger input signals (TIPk0, TIPk1 pins).

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

Reset sets this register to 00H.

Remark TMP1 and TMP3 do not have the TP1IOC1 and TP3IOC1 registers.

After reset: 00H R/W Address: TP0IOC1 FFFFF643H, TP2IOC1 FFFFF683H

	7	6	5	4	3	2	1	0
TPkIOC1	0	0	0	0	TPkIS3	TPkIS2	TPkIS1	TPkIS0

(k = 0, 2)

TPkIS3	TPkIS2	Capture trigger input signal (TIPk1 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

TPkIS1	TPkIS0	Capture trigger input signal (TIPk0 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 TPkIS3 to TPkIS0 bits when the TPkCTL0.TPkCE bit = 0. (The same value can be written when the TPkCE bit = 1.) If rewriting was mistakenly performed, clear the TPkCE bit to 0 and then set the bits again.

2. The TPkIS3 to TPkIS0 bits are valid only in the free-running timer mode (only when the TPnOPT0.TPkCCS1 and TPkCCS0 bits = 11) and the pulse width measurement mode. In all other modes, a capture operation is not possible (TMP0, TMP2 only).

(5) TMPk I/O control register 2 (TPkIOC2)

The TPkIOC2 register is an 8-bit register that controls the valid edge for the external event count input signal (TIPk0 pin) and external trigger input signal (TIPk0 pin).

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

Reset sets this register to 00H.

Remark TMP1 and TMP3 do not have the TP1IOC2 and TP3IOC2 registers.

After reset: 00H R/W Address: TP0IOC2 FFFFF644H, TP2IOC2 FFFFF684H

	7	6	5	4	3	2	1	0
TPkIOC2	0	0	0	0	TPkEES1	TPkEES0	TPkETS1	TPkETS0

(k = 0, 2)

TPkEES1	TPkEES0	External event count input signal (TIPk0 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

TPkETS1	TPkETS0	External trigger input signal (TIPk0 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 TPkEES1, TPkEES0, TPkETS1, and TPkETS0 bits when the TPkCTL0.TPkCE bit = 0. (The same value can be written when the TPkCE bit = 1.) If rewriting was mistakenly performed, clear the TPkCE bit to 0 and then set the bits again.

2. The TPkEES1 and TPkEES0 bits are valid only when the TPkCTL1.TPkEEE bit = 1 or when the external event count mode (TPkCTL1.TPkMD2 to TPkCTL1.TPkMD0 bits = 001) has been set.

3. The TPkETS1 and TPkETS0 bits are valid only in the external trigger pulse output mode or one-shot pulse output mode.

(6) TMPn option register 0 (TPnOPT0)

The TPnOPT0 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: TP0OPT0 FFFFF645H, TP1OPT0 FFFFF665H, TP2OPT0 FFFFF685H, TP3OPT0 FFFFF6A5H

TPnOPT0 (n = 0 to 3, k = 0, 2)

7	6	5	4	3	2	1	<0>
0	0	TPkCCS1[Note]	TPkCCS0[Note]	0	0	0	TPnOVF

TPkCCS1[Note]	TPkCCR1 register capture/compare selection
0	Compare register selected
1	Capture register selected (cleared by TPkCTL0.TPkCE bit = 0)
The TPkCCS1 bit setting is valid only in the free-running timer mode.	

TPkCCS0[Note]	TPkCCR0 register capture/compare selection
0	Compare register selected
1	Capture register selected (cleared by TPkCTL0.TPkCE bit = 0)
The TPkCCS0 bit setting is valid only in the free-running timer mode.	

TPnOVF	TMPn overflow detection flag
Set (1)	Overflow occurred
Reset (0)	0 written to TPnOVF bit or TPnCTL0.TPnCE bit = 0
• The TPnOVF 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 (INTTPnOV) is generated at the same time that the TPnOVF bit is set to 1. The INTTPnOV signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode. • The TPnOVF bit is not cleared to 0 even when the TPnOVF bit or the TPnOPT0 register is read when the TPnOVF bit = 1. • Before clearing the TPnOVF bit to 0 after generation of the INTTPnOV signal, be sure to confirm (by reading) that the TPnOVF bit is set to 1. • The TPnOVF bit can be both read and written, but the TPnOVF bit cannot be set to 1 by software. Writing 1 has no effect on the operation of TMPn.	

Note Valid only for TMP0 and TMP2. Be sure to clear bits 5 and 4 of TMP1 and TMP3 to 0.

Cautions 1. Rewrite the TPkCCS1 and TPkCCS0 bits when the TPkCE bit = 0. (The same value can be written when the TPkCE bit = 1.) If rewriting was mistakenly performed, clear the TPkCE bit to 0 and then set the bits again.

2. Be sure to clear bits 1 to 3, 6, and 7 to "0".

(7) TMPn capture/compare register 0 (TPnCCR0)

The TP0CCR0 and TP2CCR0 registers are 16-bit registers that can be used as capture registers or compare registers depending on the mode. The TP1CCR0 and TP3CCR0 registers are 16-bit registers that can only be used as compare registers.

The TP0CCR0 and TP2CCR0 registers can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TP0OPT0.TP0CCS0 or TP2OPT0.TP2CCS0 bit. In the pulse width measurement mode, the TPnCCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TPnCCR0 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: TP0CCR0 FFFFF646H, TP1CCR0 FFFFF666H, TP2CCR0 FFFFF686H, TP3CCR0 FFFFF6A6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCCR0																

(n = 0 to 3)

(a) Function as compare register

The TPnCCR0 register can be rewritten even when the TPnCTL0.TPnCE bit = 1.

The set value of the TPnCCR0 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 (INTTPnCC0) is generated. If TOP00 pin output is enabled at this time, the output of the TOP00 pin is inverted (TOP10, TOP20, and TOP30 pins are not provided in the V850ES/IE2).

When the TPnCCR0 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 the TPnCTL0.TPnCE bit = 0.

(b) Function as capture register (TP0CCR0 and TP2CCR0 registers only)

When the TPkCCR0 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 TPkCCR0 register if the valid edge of the capture trigger input pin (TIPk0 pin) is detected. In the pulse width measurement mode, the count value of the 16-bit counter is stored in the TPkCCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIPk0 pin) is detected.

Even if the capture operation and reading the TPkCCR0 register conflict, the correct value of the TPkCCR0 register can be read.

The capture register is cleared by the TPkCTL0.TPkCE bit = 0.

Remark k = 0, 2

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 2]	Compare register	Batch write[Note 4]
One-shot pulse output[Note 2]	Compare register	Anytime write
PWM output[Note 3]	Compare register	Batch write[Note 4]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None

Notes 1. TMP0 and TMP2 only

2. TMP0 and TMP2 only (also TMP3 in software trigger mode)

3. TMP0, TMP2, and TMP3 only

4. Writing to the TPnCCR1 register is the trigger.

Remark For anytime write and batch write, see **6.6 (2) Anytime write and batch write**.

(8) TMPn capture/compare register 1 (TPnCCR1)

The TP0CCR1 and TP2CCR1 registers are 16-bit registers that can be used as capture registers or compare registers depending on the mode. The TP1CCR1 and TP3CCR1 registers are 16-bit registers that can only be used as compare registers.

The TP0CCR1 and TP2CCR1 registers can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TP0OPT0.TP0CCS1 or TP2OPT0.TP2CCS1 bit. In the pulse width measurement mode, the TPnCCR1 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TPnCCR1 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: TP0CCR1 FFFFF648H, TP1CCR1 FFFFF668H, TP2CCR1 FFFFF688H, TP3CCR1 FFFFF6A8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCCR1																

(n = 0 to 3)

(a) Function as compare register

The TPnCCR1 register can be rewritten even when the TPnCTL0.TPnCE bit = 1.

The set value of the TPnCCR1 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 (INTTPnCC1) is generated. If TOPm1 pin output is enabled at this time, the output of the TOPm1 pin is inverted (the TOP11 pin is not provided in the V850ES/IE2).

The compare register is not cleared by the TPnCTL0.TPnCE bit = 0.

Remark m = 0, 2, 3

(b) Function as capture register (TP0CCR1 and TP2CCR1 registers only)

When the TPkCCR1 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 TPkCCR1 register if the valid edge of the capture trigger input pin (TIPk1 pin) is detected. In the pulse width measurement mode, the count value of the 16-bit counter is stored in the TPkCCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIPk1 pin) is detected.

Even if the capture operation and reading the TPkCCR1 register conflict, the correct value of the TPkCCR1 register can be read.

The capture register is cleared by the TPkCTL0.TPkCE bit = 0.

Remark k = 0, 2

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 2]	Compare register	Batch write[Note 4]
One-shot pulse output[Note 2]	Compare register	Anytime write
PWM output[Note 3]	Compare register	Batch write[Note 4]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None

Notes 1. TMP0 and TMP2 only

2. TMP0 and TMP2 only (also TMP3 in software trigger mode)

3. TMP0, TMP2, and TMP3 only

4. Writing to the TPnCCR1 register is the trigger.

Remark For anytime write and batch write, see **6.6 (2) Anytime write and batch write**.

(9) TMPn counter read buffer register (TPnCNT)

The TPnCNT register is a read buffer register that can read the count value of the 16-bit counter.

If this register is read when the TPnCTL0.TPnCE bit = 1, the count value of the 16-bit counter can be read.

This register is read-only, in 16-bit units.

The value of the TPnCNT register is cleared to 0000H when the TPnCE bit = 0. If the TPnCNT 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 TPnCE bit is cleared to 0 after reset, and the TPnCNT register is cleared to 0000H.

After reset: 0000H R Address: TP0CNT FFFFF64AH, TP1CNT FFFFF66AH, TP2CNT FFFFF68AH, TP3CNT FFFFF6AAH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPnCNT																

(n = 0 to 3)

6.5 Timer Output Operations

The following table shows the operations and output levels of the TOP00 and TOPm1 pins.

Table 6-6. Timer Output Control in Each Mode

Operation Mode	TOPm1 Pin	TOP00 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 m = 0, 2, 3

Table 6-7. Truth Table of TOP00 and TOPm1 Pins Under Control of Timer Output Control Bits

TPmIOC0.TPmOLa Bit	TPmIOC0.TPmOEa Bit	TPmCTL0.TPmCE Bit	Level of TOPma 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 a = 0, 1 when m = 0
a = 1 when m = 2, 3

6.6 Operation

The functions of TMPn that can be realized differ from one channel to another. The functions of each channel are shown below.

Table 6-8. TMP0 Specifications in Each Mode

Operation	TP0CTL1.TP0EST Bit (Software Trigger Bit)	TIP00 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	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 TIP00 pin capture trigger input valid edge selection to “No edge detection”. (Clear the TP0IOC1.TP0IS1 and TP0IOC1.TP0IS0 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 TP0CTL1.TP0EEE bit to 0).

Remark The TIP00 pin functions alternately as a capture trigger input, external event count input, and external trigger input.

Table 6-9. TMP1 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. TMP1 does not have timer input pins (TIP10, TIP11) and timer output pins (TOP10, TOP11). It has interrupt request signals (INTTP1CC0, INTTP1CC1) indicating a match between the value of the 16-bit counter and the values of the TP1CCR0 and TP1CCR1 registers.

2. TMP1 has a function to execute tuning with TMQ1. For details, see **CHAPTER 9 MOTOR CONTROL FUNCTION**.

Table 6-10. TMP2 Specifications in Each Mode

Operation	TP2CTL1.TP2EST Bit (Software Trigger Bit)	TIP20 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	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 TIP20 pin capture trigger input valid edge selection to “No edge detection”. (Clear the TP2IOC1.TP2IS1 and TO2IOC1.TP2ISO bits to 00.)

2. When using the external trigger pulse output mode and one-shot pulse output mode, select the internal clock as the count clock (by clearing the TP2CTL1.TP2EEE bit to 0).

Remark The TIP20 pin functions alternately as a capture trigger input, external event count input, and external trigger input.

Table 6-11. TMP3 Specifications in Each Mode

Operation	TP3CTL1.TP3EST Bit (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[Note]	Valid	Invalid	Compare only	Batch write
One-shot pulse output mode[Note]	Valid	Invalid	Compare only	Anytime write
PWM output mode	Invalid	Invalid	Compare only	Batch write
Free-running mode	Invalid	Invalid	Compare only	Anytime write
Pulse width measurement mode	None			

Note When using the external trigger pulse output mode, one-shot pulse output mode, an external trigger cannot be input. Only a software trigger (set by the TP3CTL1.TP3EST bit) can be used.

Remark TMP3 does not have timer input pins (TIP30, TIP31) and timer output pin (TOP30). The match interrupt request signals (INTTP3CC0, INTTP3CC1) of the 16-bit counter and the TP3CCR0 and TP3CCR1 registers are provided.

(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 to 3
k = 0, 2

<R>

(a) Counting start operation

- In external event count mode
  When the TPkCTL0.TPkCE bit is set from 0 to 1, the 16-bit counter is set to 0000H.
  After that, it counts up to 0001H, 0002H, 0003H, … each time the valid edge of external event count input (TIPk0) is detected.
- In modes other than the above
  The 16-bit counter of TMPn 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 INTTPnCC0 and INTTPnCC1 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 timer mode or pulse width measurement mode. If the counter overflows, the TPnOPT0.TPnOVF bit is set to 1 and an interrupt request signal (INTTPnOV) is generated. Note that the INTTPnOV 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 (INTTPnOV) has been generated, be sure to check that the overflow flag (TPnOVF bit) is set to 1.

(d) Counter read operation during counting operation

The value of the 16-bit counter of TMPn can be read by using the TPnCNT register during the count operation. When the TPnCTL0.TPnCE bit = 1, the value of the 16-bit counter can be read by reading the TPnCNT register. However, when the TPnCE bit = 0, the 16-bit counter is FFFFH and the TPnCNT register is 0000H.

(e) Interrupt operation

TMPn generates the following three types of interrupt request signals.

- INTTPnCC0 interrupt: This signal functions as a match interrupt request signal of the CCR0 buffer register and as a capture interrupt request signal to the TPnCCR0 register.
- INTTPnCC1 interrupt: This signal functions as a match interrupt request signal of the CCR1 buffer register and as a capture interrupt request signal to the TPnCCR1 register.
- INTTPnOV interrupt: This signal functions as an overflow interrupt request signal.

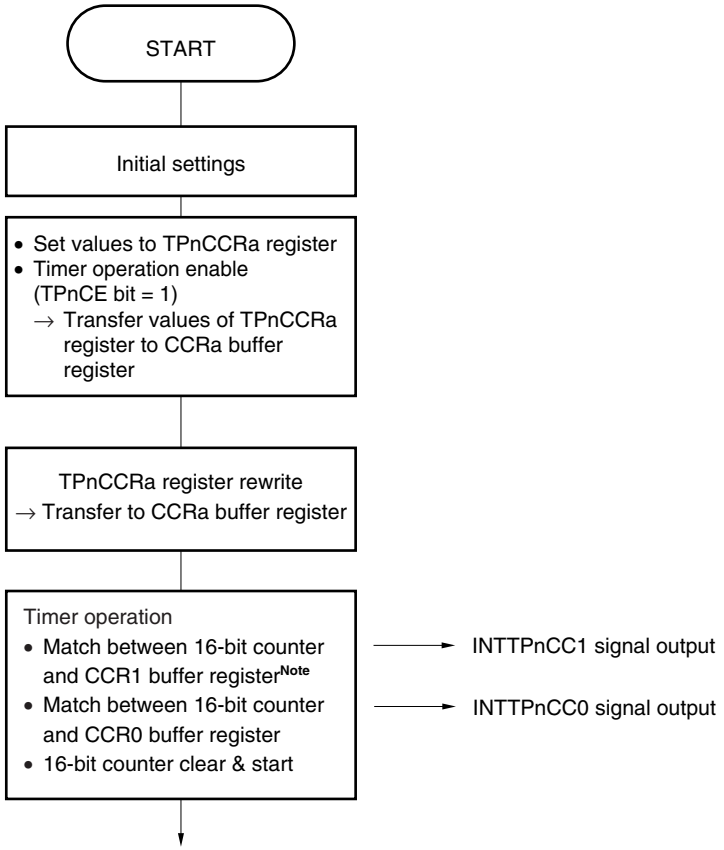
(2) Anytime write and batch write

The TPnCCR0 and TPnCCR1 registers in TMPn can be rewritten during timer operation (TPnCTL0.TPnCE 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 TPnCCR0 and TPnCCR1 registers to the CCR0 and CCR1 buffer registers during timer operation (n = 0 to 3).

Figure 6-5. Flowchart of Basic Operation for Anytime Write

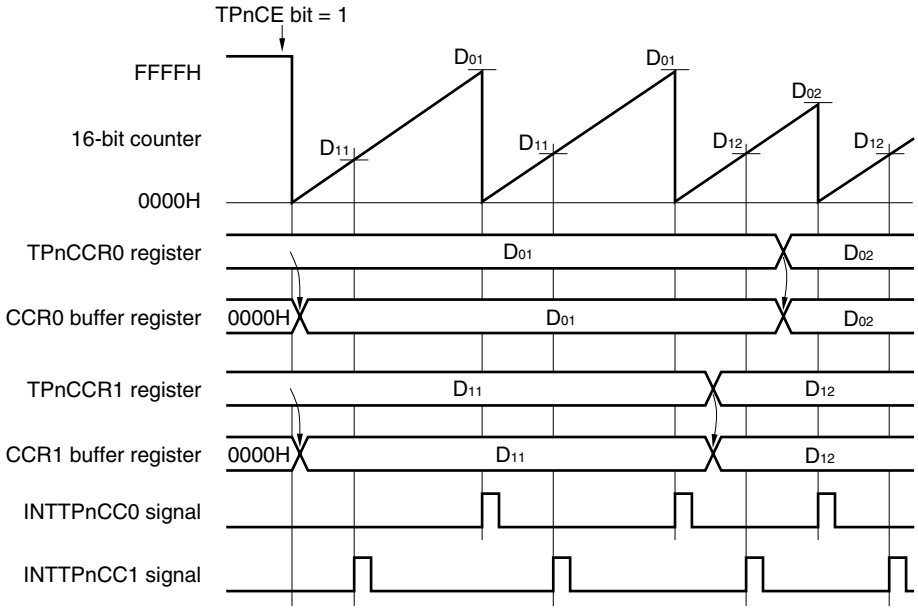


Note The 16-bit counter is not cleared upon a match between the value of the 16-bit counter and the value of the CCR1 buffer register. It is cleared upon a match between the value of the 16-bit counter and the value of the CCR0 buffer register.

Remarks 1. The above flowchart illustrates an example of the operation in the interval timer mode.

2. n = 0 to 3, a = 0, 1

Figure 6-6. Timing of Anytime Write



Remarks 1. D_{01}, D_{02}: Setting values of TPnCCR0 register

D_{11}, D_{12}: Setting values of TPnCCR1 register

2. The above timing chart illustrates an example of the operation in the interval timer mode.

3. n = 0 to 3

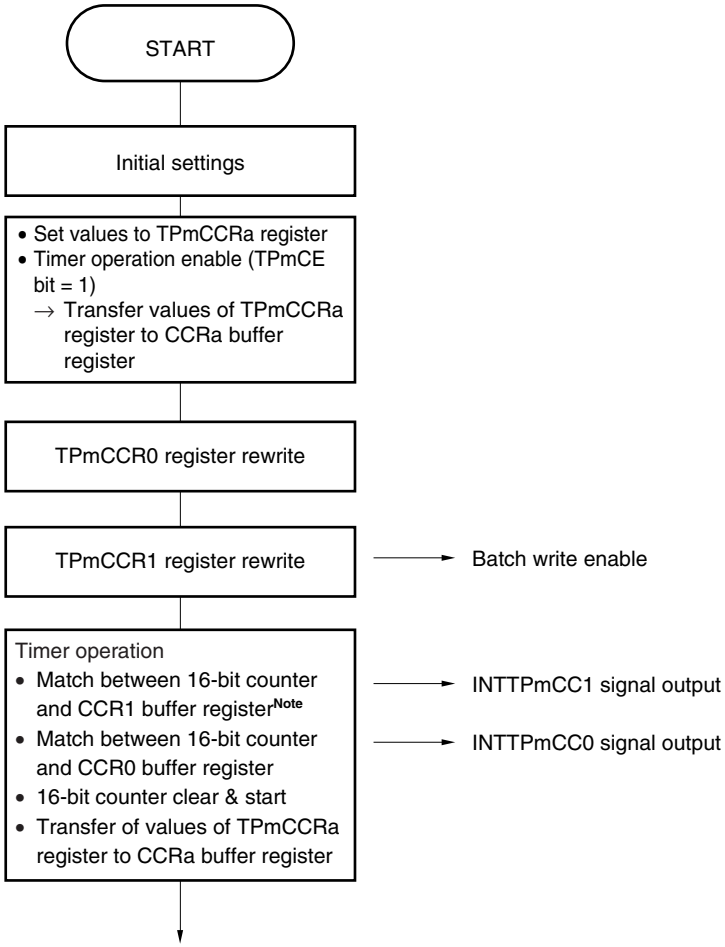
(b) Batch write

In this mode, data is transferred all at once from the TPmCCR0 and TPmCCR1 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 TPmCCR1 register.

Whether to enable or disable the next transfer timing is controlled by writing or not writing to the TPmCCR1 register.

In order for the setting value when the TPmCCR0 and TPmCCR1 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 TPmCCR0 register and then write to the TPmCCR1 register before the 16-bit counter value and the CCR0 buffer register value match. Therefore, the values of the TPmCCR0 and TPmCCR1 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 TPmCCR0 register, also write the same value (same as preset value of the TPmCCR1 register) to the TPmCCR1 register.

Figure 6-7. Flowchart of Basic Operation for Batch Write



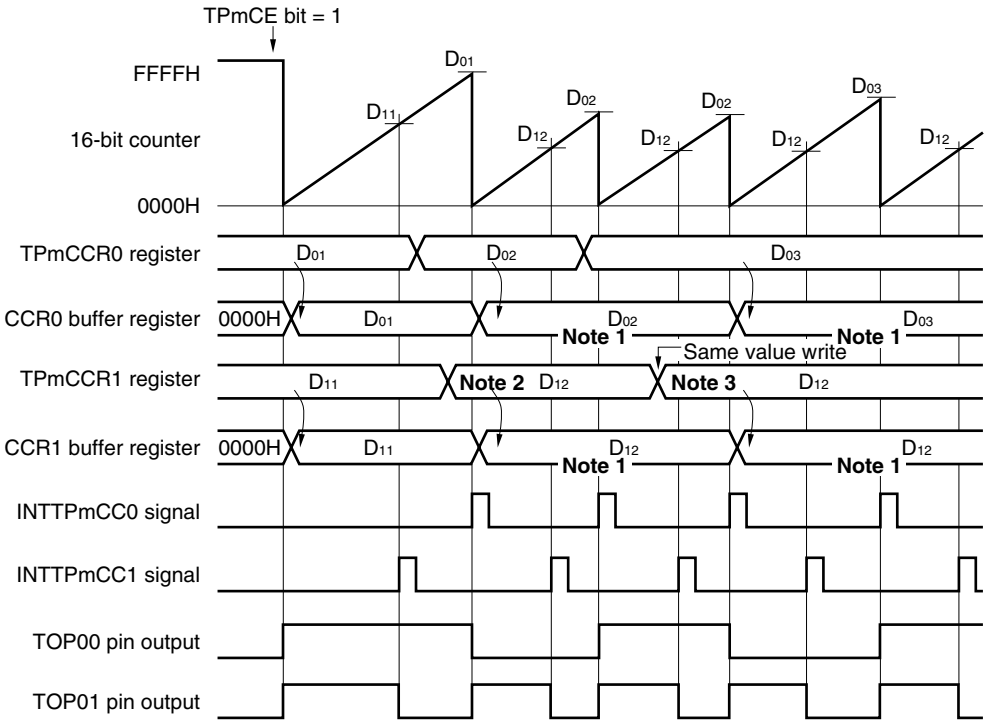
Note The 16-bit counter is not cleared upon a match between the value of the 16-bit counter and the value of the CCR1 buffer register. It is cleared upon a match between the value of the 16-bit counter and the value of the CCR0 buffer register.

Caution Writing to the TPmCCR1 register includes enabling of batch write. Thus, rewrite the TPmCCR1 register after rewriting the TPmCCR0 register.

Remarks 1. The above flowchart illustrates an example of the operation in the PWM output mode.

2. m = 0, 2, 3
a = 0, 1

Figure 6-8. Timing of Batch Write



Notes 1. Because the TPmCCR1 register was not rewritten, D_{03} is not transferred.

2. Because the TPmCCR1 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 TPmCCR0 register (D_{01}).

3. Because the TPmCCR1 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 TPmCCR0 register (D_{02}).

Remarks 1. D_{01}, D_{02}, D_{03}: Setting values of TPmCCR0 register
D_{11}, D_{12}: Setting values of TPmCCR1 register

2. The above timing chart illustrates an example of the operation in the PWM output mode.

3. m = 0, 2, 3

6.6.1 Interval timer mode (TPnMD2 to TPnMD0 bits = 000)

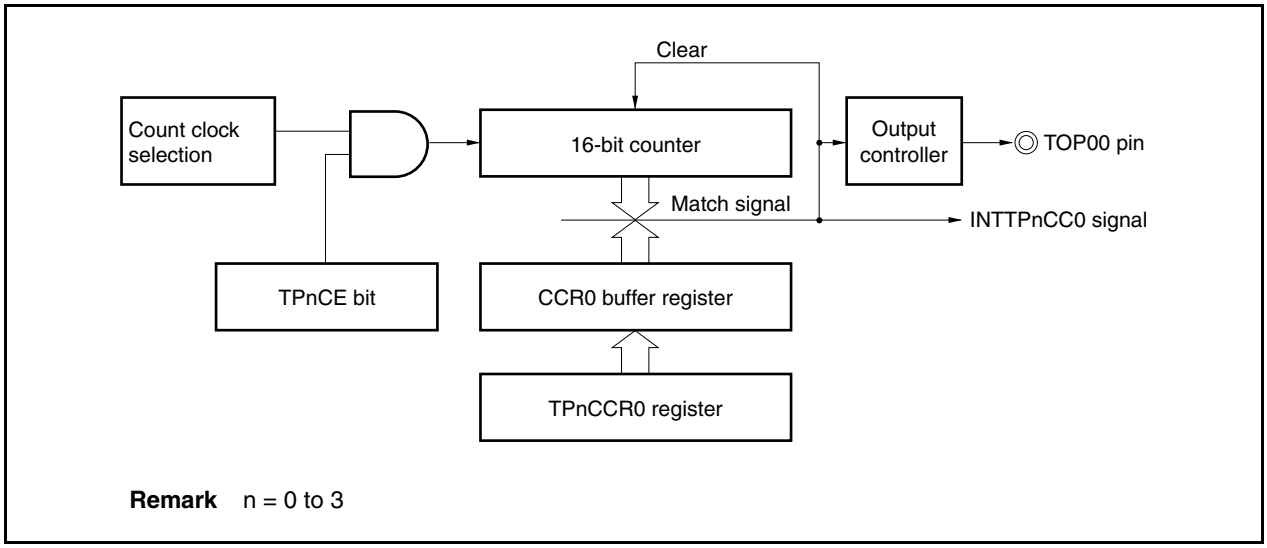
In the interval timer mode, an interrupt request signal (INTTPnCC0) is generated at the interval set by the TPnCCR0 register if the TPnCTL0.TPnCE bit is set to 1. A PWM waveform with a duty factor of 50% whose half cycle is equal to the interval can be output from the TOP00 pin (TMP0 only).

The TPnCCR1 register is not used in the interval timer mode. However, the set value of the TPnCCR1 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 (INTTPnCC1) is generated. In addition, a PWM waveform with a duty factor of 50%, which is inverted when the INTTPmCC1 signal is generated, can be output from the TOPm1 pin.

The value of the TPnCCR0 and TPnCCR1 registers can be rewritten even while the timer is operating.

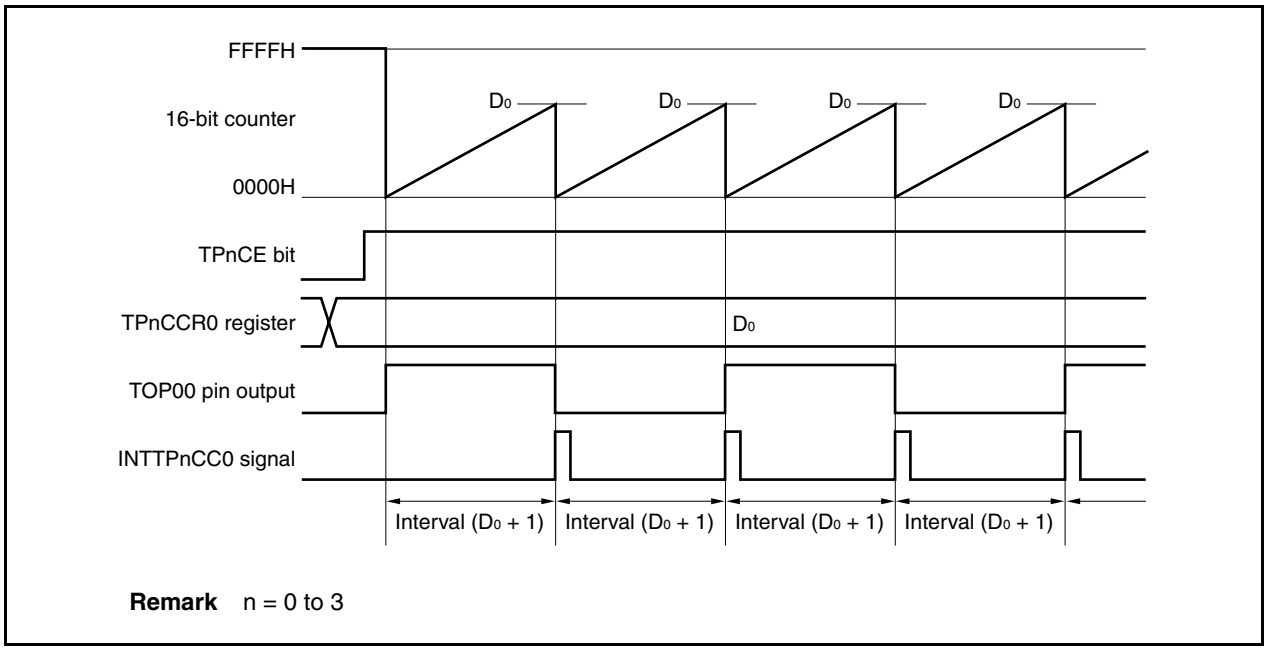
Remark m = 0, 2, 3

Figure 6-9. Configuration of Interval Timer



Remark n = 0 to 3

Figure 6-10. Basic Timing of Operation in Interval Timer Mode



Remark n = 0 to 3

When the TPnCE 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 TOP00 pin is inverted. Additionally, the set value of the TPnCCR0 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 TOP00 pin is inverted, and a compare match interrupt request signal (INTTPnCC0) is generated.

The interval can be calculated by the following expression.

Interval = (Set value of TPnCCR0 register + 1) × Count clock cycle

Remark n = 0 to 3

Figure 6-11. Register Setting for Interval Timer Mode Operation (1/3)

(a) TMPn control register 0 (TPnCTL0)

	TPnCE					TPnCKS2	TPnCKS1	TPnCKS0
TPnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock

0: Stop counting
1: Enable counting

(b) TMPn control register 1 (TPnCTL1)

	TP1SYE	TPmEST	TPkEEE			TPnMD2	TPnMD1	TPnMD0
<R> TPnCTL1	0	0	0/1[Note]	0	0	0	0	0

0, 0, 0:
Interval timer mode

0: Operate at count clock selected in TPkCKS0 to TPkCKS2 bits
1: Count operation by external event count input signal

<R> **Note** Enable setting of the TPkEEE bit to 1 only when timer output (TOPk1) is used. In this case, set the TPkCCR0 and TPkCCR1 registers to the same value.

Figure 6-11. Register Setting for Interval Timer Mode Operation (2/3)

(c) TMPm I/O control register 0 (TPmIOC0)

	TPmOL1	TPmOE1	TP0OL0	TP0OE0
TPmIOC0	0/1	0/1	0/1	0/1

- TP0OE0: 0: Disable TOP00 pin output; 1: Enable TOP00 pin output
- TP0OL0: Setting of TOP00 pin output level before count operation; 0: Low level; 1: High level
- TPmOE1: 0: Disable TOPm1 pin output; 1: Enable TOPm1 pin output
- TPmOL1: Setting of TOPm1 pin output level before count operation; 0: Low level; 1: High level

<R> **(d) TMPk I/O control register 2 (TPkIOC2)**

	TPkEES1	TPkEES0	TPkETS1	TPkETS0
TPkIOC2	0/1 Note	0/1 Note	0	0

- TPkEES1, TPkEES0: Selection of valid edge of external event count input (TIPk0 pin)

Note Enable setting of the TPkEES1 and TPkEES0 bits only when timer output (TOPk1) is used. In this case, set the TPkCCR0 and TPkCCR1 registers to the same value.

(e) TMPn counter read buffer register (TPnCNT)

By reading the TPnCNT register, the count value of the 16-bit counter can be read.

(f) TMPn capture/compare register 0 (TPnCCR0)

If the TPnCCR0 register is set to D_0, the interval is as follows.

Interval = $(D_0 + 1) \times$ Count clock cycle

Figure 6-11. Register Setting for Interval Timer Mode Operation (3/3)

(g) TMPn capture/compare register 1 (TPnCCR1)

The TPnCCR1 register is not used in the interval timer mode. However, the set value of the TPnCCR1 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 TOPm1 pin output is inverted and a compare match interrupt request signal (INTTPnCC1) is generated.

By setting this register to the same value as the value set in the TPmCCR0 register, a PWM waveform with a duty factor of 50% can be output from the TOPm1 pin.

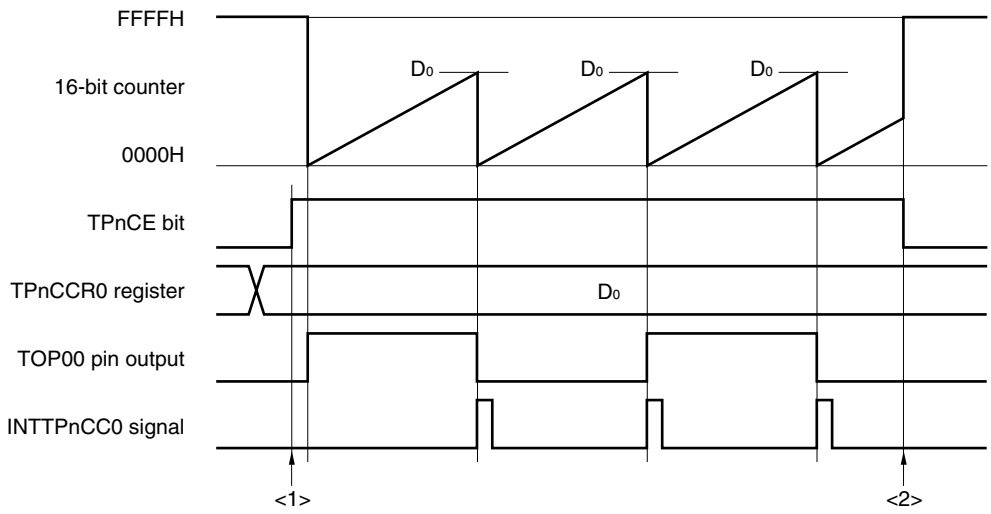
When the TPnCCR1 register is not used, it is recommended to set its value to FFFFH. Also mask the register by the interrupt mask flag (TPnCCIC1.TPnCCMK1).

<R> **Remarks 1.** TMPk I/O control register 1 (TPkIOC1) and TMPn option register 0 (TPnOPT0) are not used in the interval timer mode.

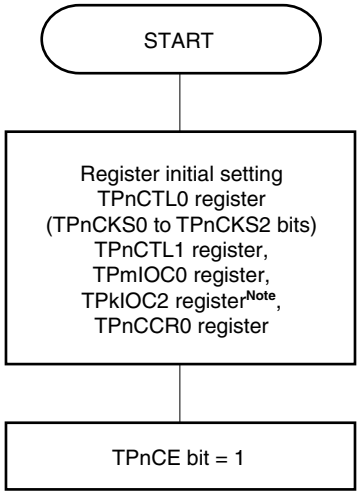
2. n = 0 to 3,
m = 0, 2, 3
k = 0, 2

(1) Interval timer mode operation flow

Figure 6-12. Software Processing Flow in Interval Timer Mode (1/2)



<1> Count operation start flow



Initial setting of these registers is performed before setting the TPnCE bit to 1.

The TPnCKS0 to TPnCKS2 bits can be set at the same time when counting has been started (TPnCE bit = 1).

<R>

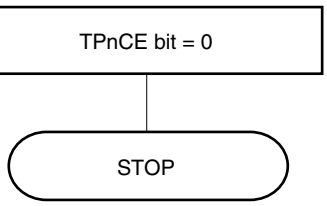
<R>

Note Enable setting of the TPkEES1 and TPkEES0 bits only when timer output (TOPk1) is used. In this case, set the TPkCCR0 and TPkCCR1 registers to the same value.

Remark n = 0 to 3
m = 0, 2, 3

Figure 6-12. 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 TPnCE bit to 0. The output level of the TOP00 pin is as specified by the TPmIOC0 register.

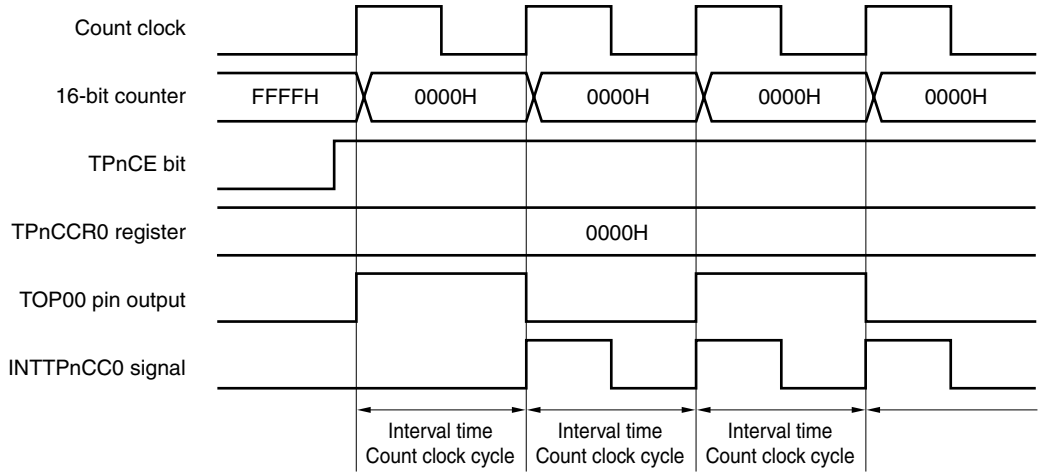
Remark n = 0 to 3
m = 0, 2, 3

(2) Interval timer mode operation timing

(a) Operation if TPnCCR0 register is set to 0000H

If the TPnCCR0 register is set to 0000H, the INTTPnCC0 signal is generated at each count clock, and the output of the TOP00 pin is inverted.

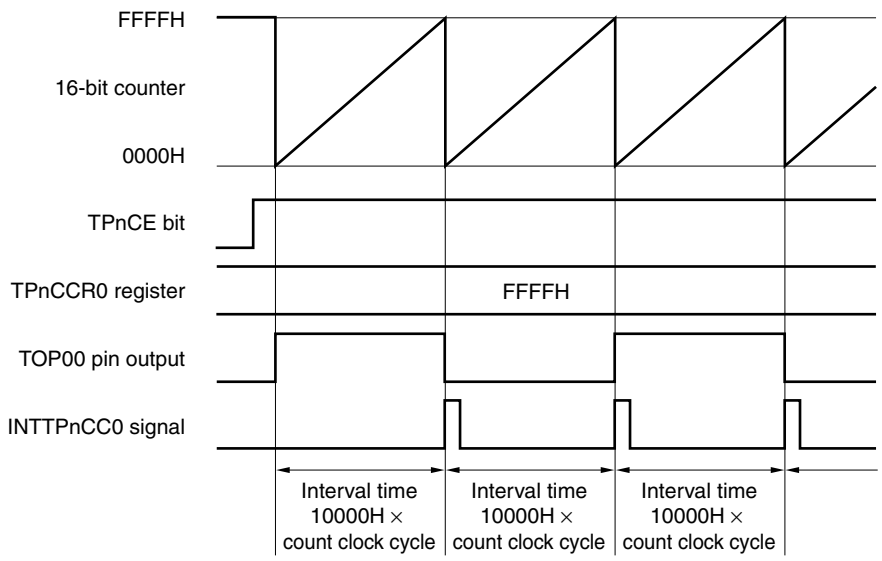
The value of the 16-bit counter is always 0000H.



Remark n = 0 to 3

(b) Operation if TPnCCR0 register is set to FFFFH

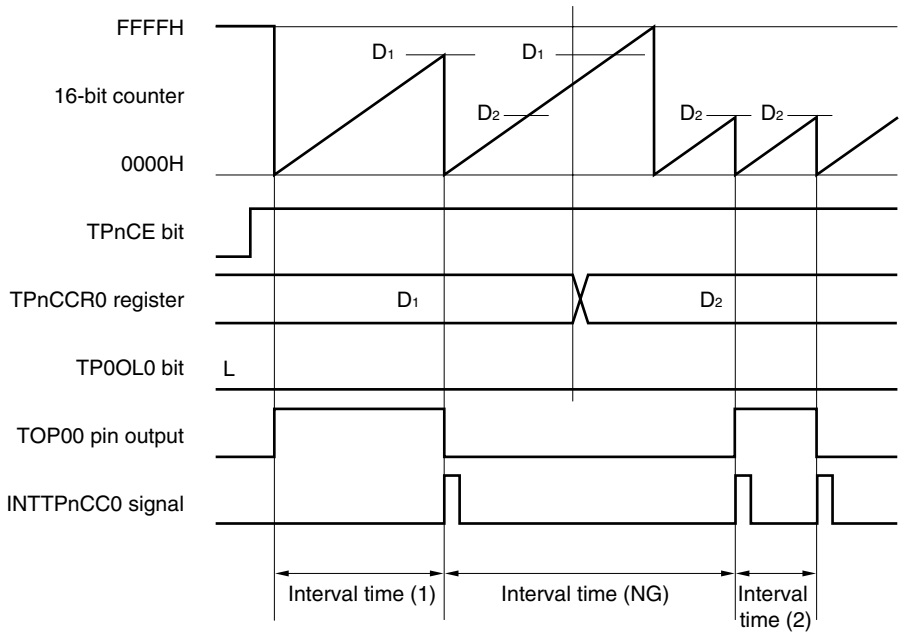
If the TPnCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTPnCC0 signal is generated and the output of the TOP00 pin is inverted. At this time, an overflow interrupt request signal (INTTPnOV) is not generated, nor is the overflow flag (TPnOPT0.TPnOVF bit) set to 1.



Remark n = 0 to 3

(c) Notes on rewriting TPnCCR0 register

If the value of the TPnCCR0 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. n = 0 to 3

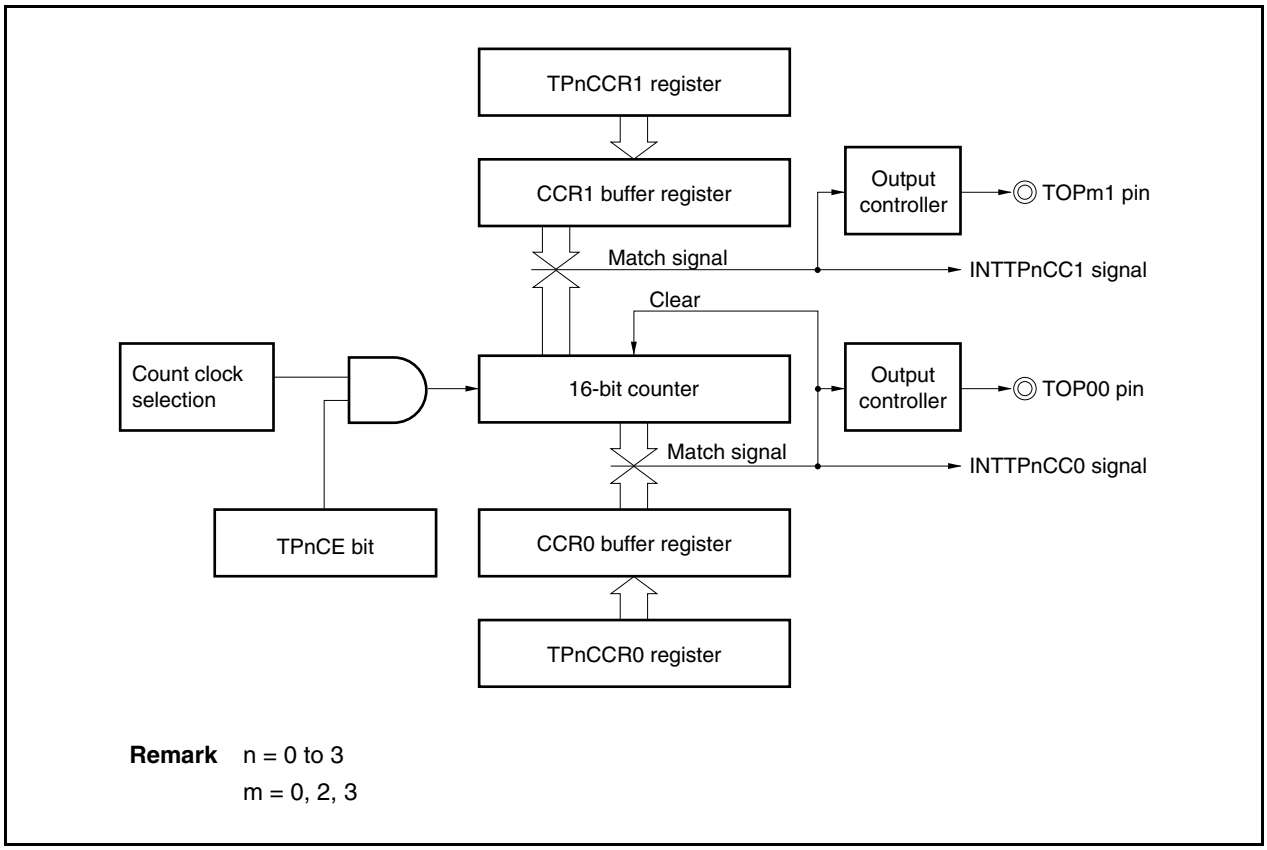
If the value of the TPnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TPnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is 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 INTTPnCC0 signal is generated and the output of the TOP00 pin is inverted.

Therefore, the INTTPnCC0 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 TPnCCR1 register

Figure 6-13. Configuration of TPnCCR1 Register



Remark n = 0 to 3
m = 0, 2, 3

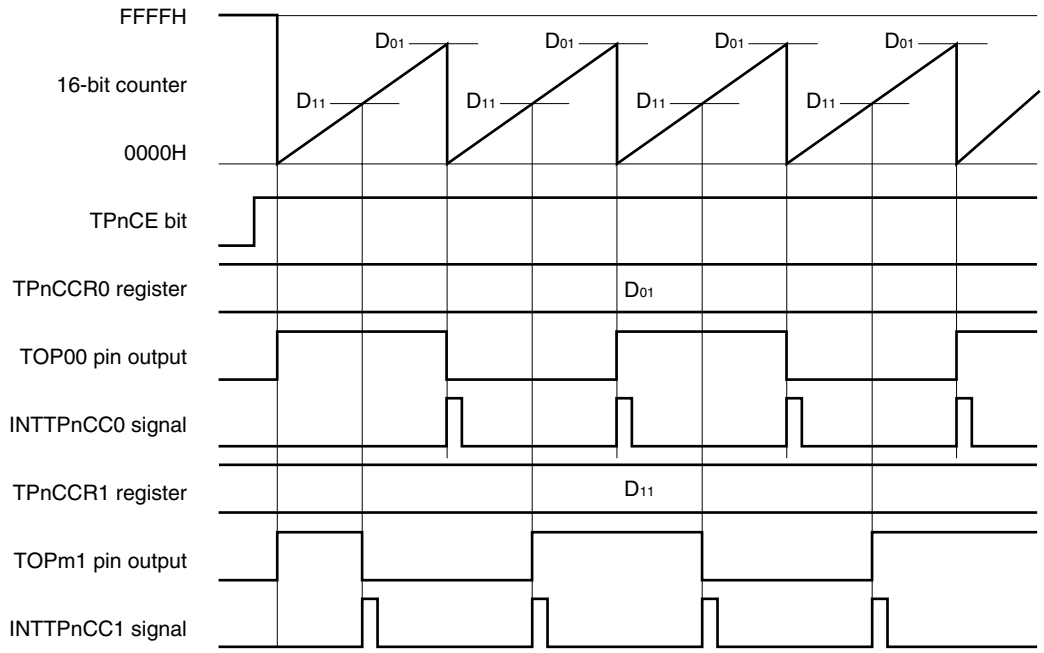
When the TPnCCR1 register is set to the same value as the TPnCCR0 register, the INTTPnCC1 signal is generated at the same timing as the INTTPnCC0 signal and the TOPm1 pin output is inverted. In other words, a PWM waveform with a duty factor of 50% can be output from the TOPm1 pin.

The following shows the operation when the TPnCCR1 register is set to other than the value set in the TPnCCR0 register.

If the set value of the TPnCCR1 register is less than the set value of the TPnCCR0 register, the INTTPnCC1 signal is generated once per cycle. At the same time, the output of the TOPm1 pin is inverted.

The TOPm1 pin outputs a PWM waveform with a duty factor of 50% after outputting a short-width pulse.

Figure 6-14. Timing Chart When $D_{01} \geq D_{11}$

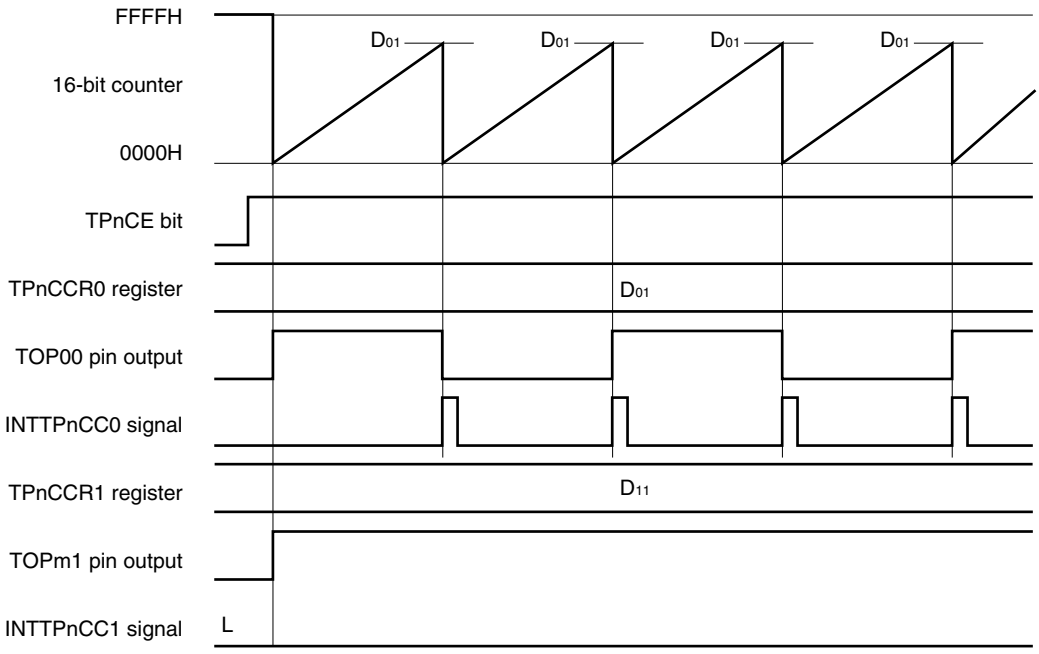


Remark n = 0 to 3
m = 0, 2, 3

If the set value of the TPnCCR1 register is greater than the set value of the TPnCCR0 register, the count value of the 16-bit counter does not match the value of the TPnCCR1 register. Consequently, the INTTPnCC1 signal is not generated, nor is the output of the TOPm1 pin changed.

When the TPnCCR1 register is not used, it is recommended to set its value to FFFFH.

Figure 6-15. Timing Chart When $D_{01} < D_{11}$



Remark n = 0 to 3
m = 0, 2, 3

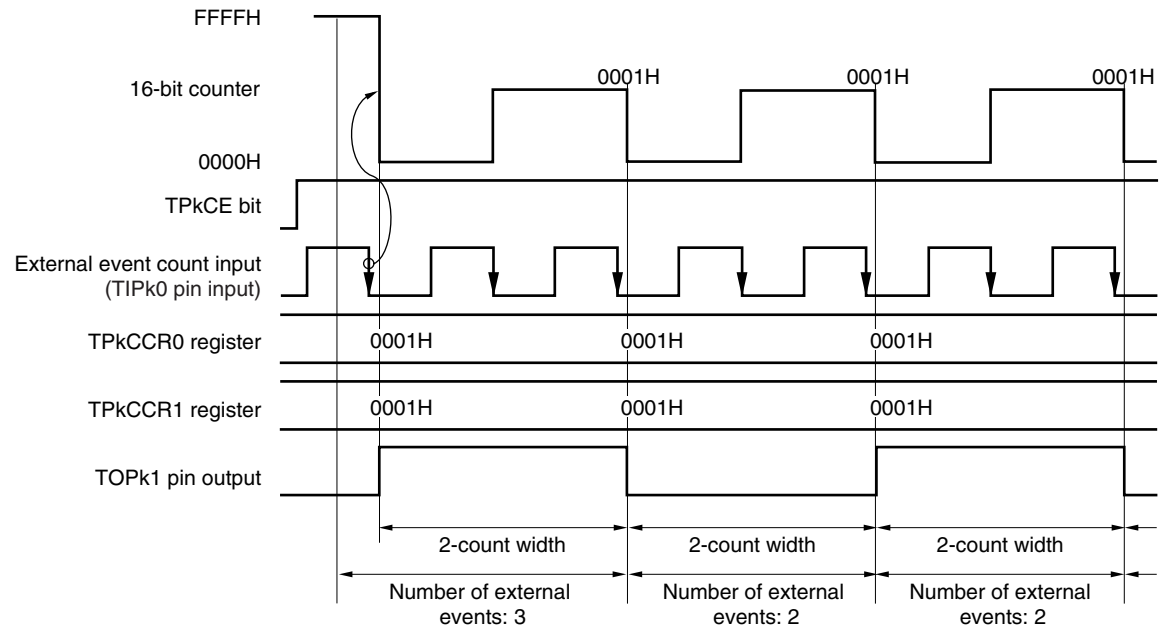
<R> **(3) Operation by external event count input (TIPk0)**

(a) Operation

To count the 16-bit counter at the valid edge of external event count input (TIPk0) in the interval timer mode, clear the 16-bit counter from FFFFH to 0000H at the valid edge of the first external event count input after the TPkCE bit is set from 0 to 1.

When both the TPkCCR0 and TPkCCR1 registers are set to 0001H, the TOPk1 pin output is inverted each time the 16-bit counter counts twice.

The TPkCTL1.TPkEEE bit can be set to 1 in the interval timer mode only when the timer output (TOPk1) is used with the external event count input.



Remark k = 0, 2

6.6.2 External event count mode (TPkMD2 to TPkMD0 bits = 001)

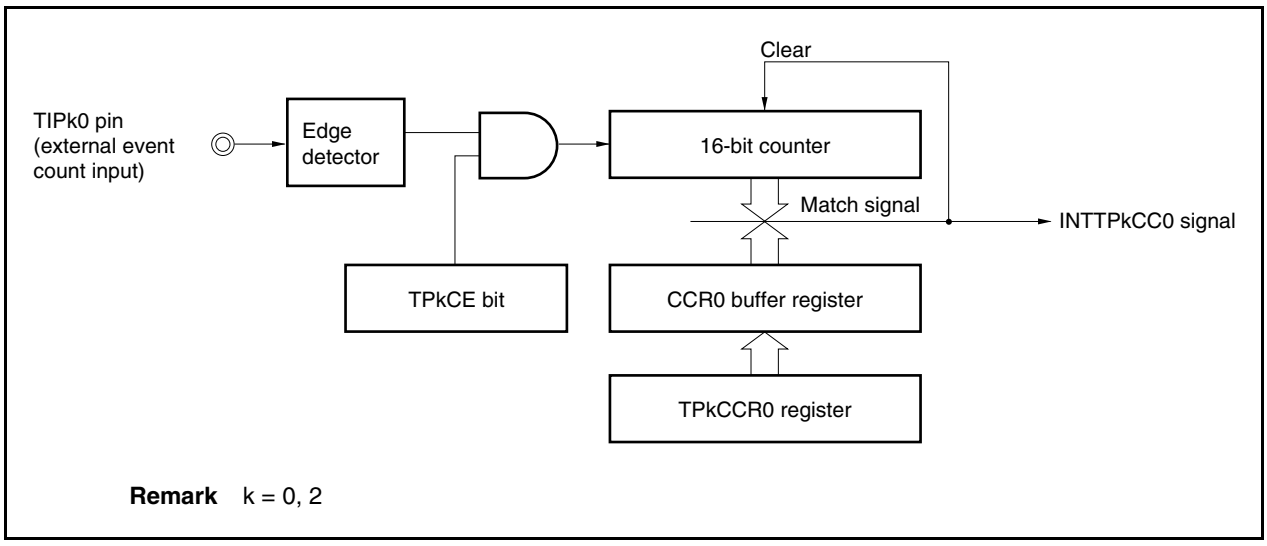
This mode is valid only in TMP0 and TMP2.

In the external event count mode, the valid edge of the external event count input (TIPk0) is counted when the TPkCTL0.TPkCE bit is set to 1, and an interrupt request signal (INTTPkCC0) is generated each time the number of <R> edges set by the TPkCCR0 register have been counted. The TOP00 and TOPk1 pins cannot be used. When using the TOPk1 pin in the external event count input mode, set the TPkCTL1.TPkEEE bit to 1 in the interval timer mode (see **6.6.1 (3) Operation by external event count input (TIPk0)**).

The TPkCCR1 register is not used in the external event count mode.

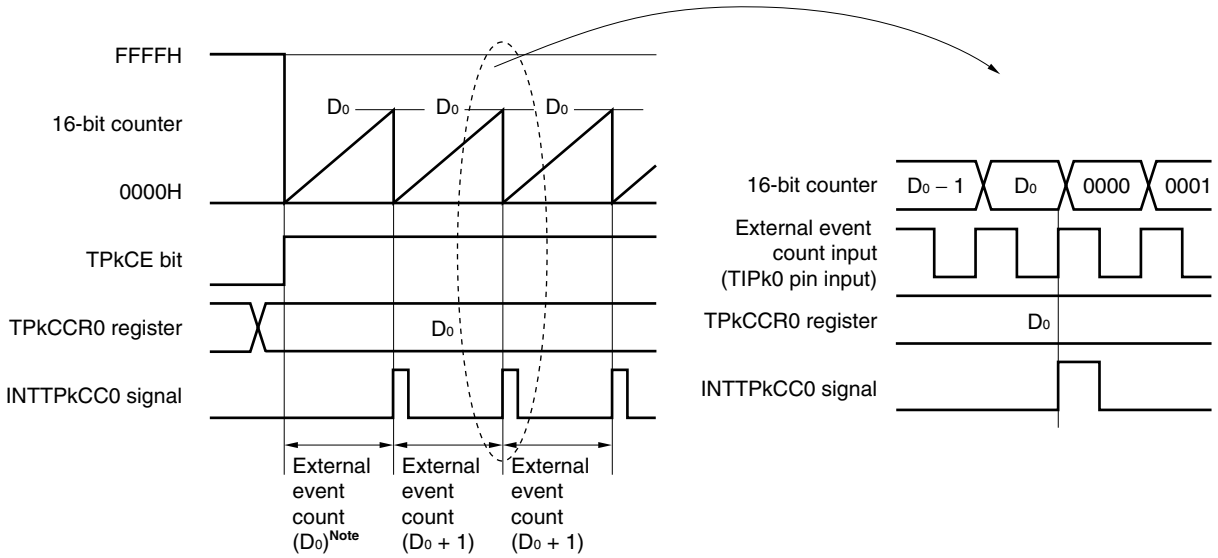
Caution In the external event count mode, the TPkCCR0 and TPkCCR1 registers must not be cleared to 0000H.

Figure 6-16. Configuration in External Event Count Mode



Remark k = 0, 2

Figure 6-17. Basic Timing in External Event Count Mode



Note In the external event count mode, when the TPkCTL0.TPkCE bit is set to 1 (operation starts), the 16-bit counter is cleared from FFFFH to 0000H at the same time. The first count operation starts from the 0001H every time the valid edge of the external event count input is detected. Therefore, the count of the first count operation is one number smaller than the count of second or subsequent count operation.

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. k = 0, 2

When the TPkCE 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 TPkCCR0 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 (INTTPkCC0) is generated.

The INTTPkCC0 signal is generated for the first time when the valid edge of the external event count input has been detected "value set to TPkCCR0 register" times. After that, the INTTPkCC0 signal is generated each time the valid edge of the external event count has been detected "value set to TPkCCR0 register + 1" times.

Figure 6-18. Register Setting for Operation in External Event Count Mode (1/2)

(a) TMPk control register 0 (TPkCTL0)

	TPkCE					TPkCKS2	TPkCKS1	TPkCKS0
TPkCTL0	0/1	0	0	0	0	0	0	0

0: Stop counting
1: Enable counting

(b) TMPk control register 1 (TPkCTL1)

		TPkEST	TPkEEE			TPkMD2	TPkMD1	TPkMD0
TPkCTL1	0	0	0	0	0	0	0	1

0, 0, 1:
External event count mode

(c) TMPk I/O control register 2 (TPkIOC2)

					TPkEES1	TPkEES0	TPkETS1	TPkETS0
TPkIOC2	0	0	0	0	0/1	0/1	0	0

Select valid edge
of external event
count input (TIPk0 pin)

(d) TMPk counter read buffer register (TPkCNT)

The count value of the 16-bit counter can be read by reading the TPkCNT register.

(e) TMPk capture/compare register 0 (TPkCCR0)

If the TPkCCR0 register is set to D_0, the count is cleared when the number of external event counts has reached (D_0) and the first compare match interrupt request signal (INTTPkCC0) is generated. The second compare match interrupt request signal (INTTPkCC0) is generated when the number of external event counts has reached (D_0 + 1).

Figure 6-18. Register Setting for Operation in External Event Count Mode (2/2)

(f) TMPk capture/compare register 1 (TPkCCR1)

The TPkCCR1 register is not used in the external event count mode. However, the set value of the TPkCCR1 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 (INTTPkCC1) is generated.

When the TPkCCR1 register is not used, it is recommended to set its value to FFFFH. Also mask the register by the interrupt mask flag (TPkCCIC1.TPkCCMK1).

<R> **Cautions 1. Set the TPkIOC0 register to 00H.**

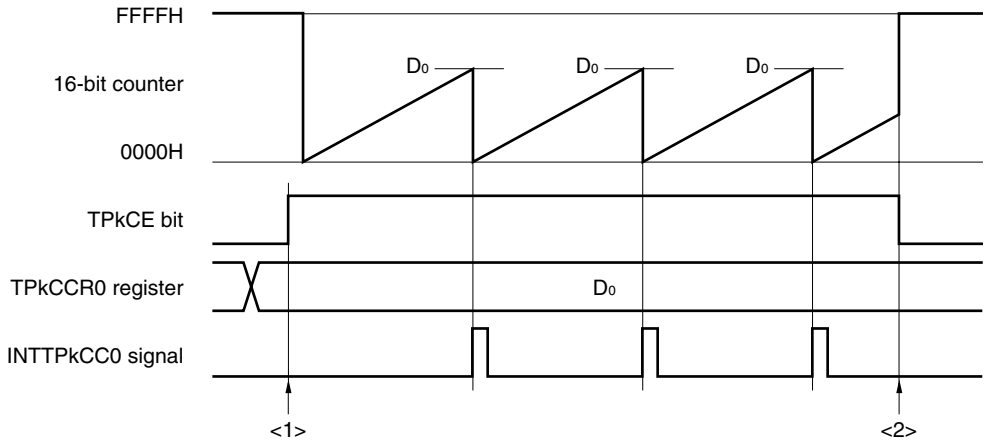
<R> **2. When an external clock is used as the count clock, the external clock can be input only from the TIPk0 pin. At this time, set the TPkIOC1.TPkIS1 and TPkIOC1.TPkIS0 bits to 00 (capture trigger input (TIPk0 pin): no edge detection).**

Remarks 1. TMPk I/O control register 0 (TPkIOC0), TMPk I/O control register 1 (TPkIOC1), and TMPk option register 0 (TPkOPT0) are not used in the external event count mode.

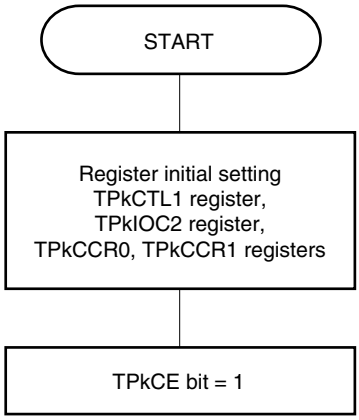
2. k = 0, 2

(1) External event count mode operation flow

Figure 6-19. Flow of Software Processing in External Event Count Mode

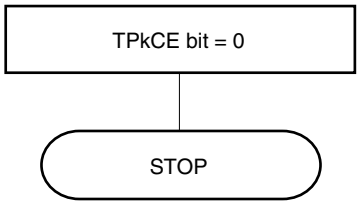


<1> Count operation start flow



Initial setting of these registers is performed before setting the TPkCE bit to 1.

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TPkCE bit to 0.

Remark k = 0, 2

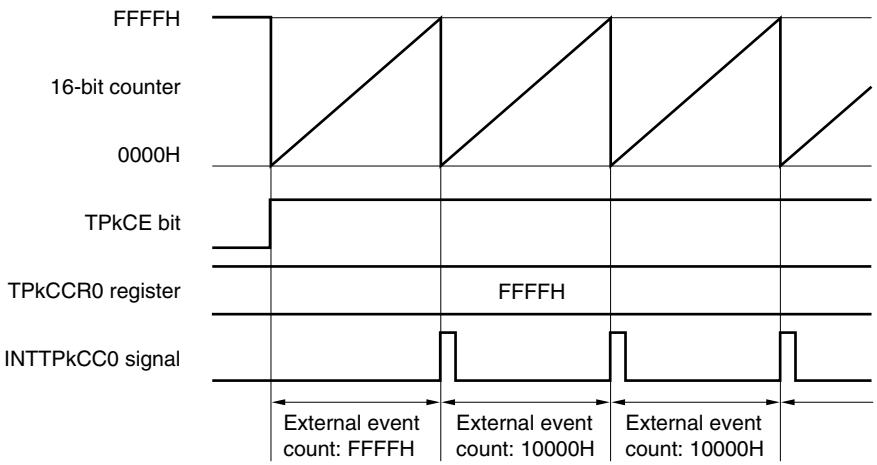
(2) Operation timing in external event count mode

Cautions 1. In the external event count mode, the TPkCCR0 and TPkCCR1 registers must not be cleared to 0000H.

<R> **2. In the external event count mode, use of the timer output (TOP00, TOPk1) is disabled. If using timer output (TOPk1) with external event count input (TIPk0), set the interval timer mode, and enable the count clock operation with the external event count input (TPkCTL1.TPkEEE bit = 1) (see 6.6.1 (3) Operation by external event count input (TIPk0)).**

(a) Operation if TPkCCR0 register is set to FFFFH

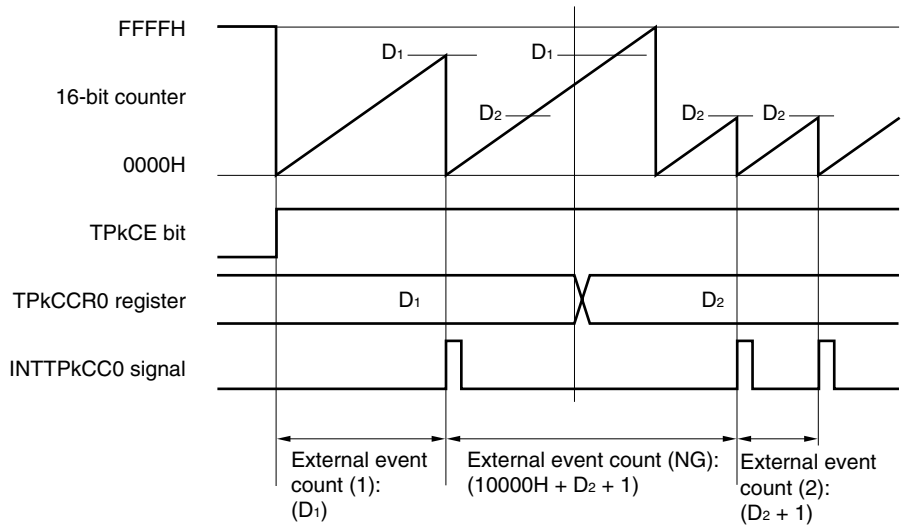
If the TPkCCR0 register is set to FFFFH, the 16-bit counter counts to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTTPkCC0 signal is generated. At this time, the TPkOPT0.TPkOVF bit is not set.



Remark k = 0, 2

(b) Notes on rewriting the TPkCCR0 register

If the value of the TPkCCR0 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 k = 0, 2

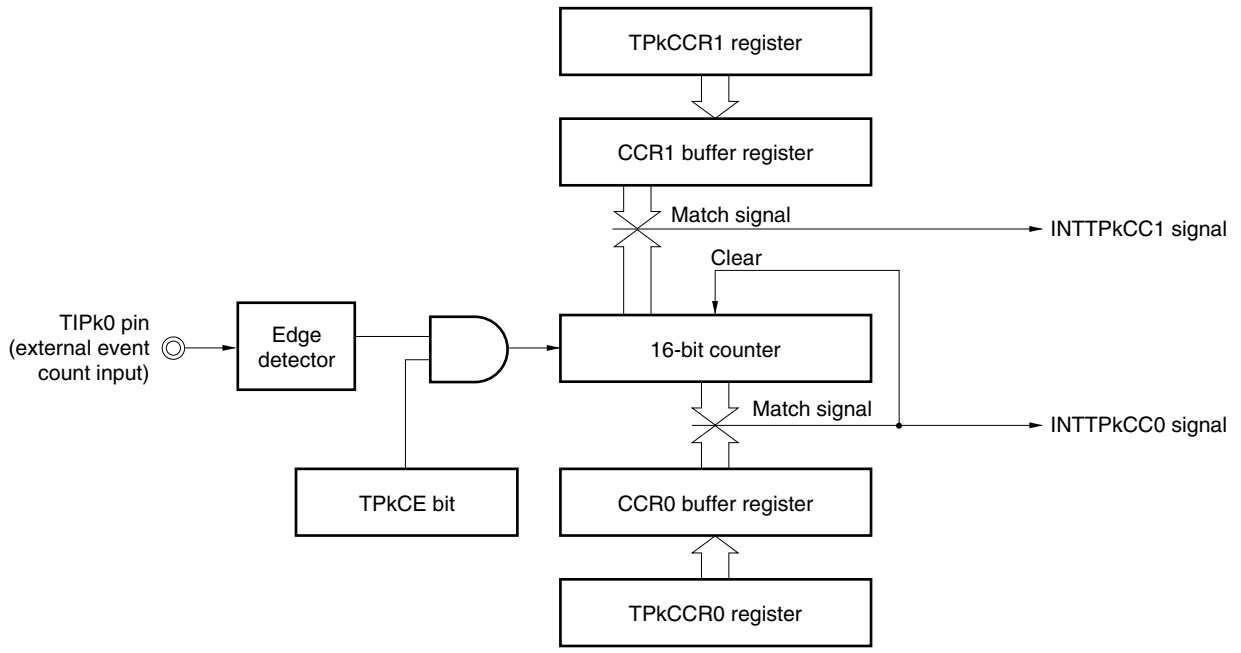
If the value of the TPkCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TPkCCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter 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 INTTPkCC0 signal is generated.

Therefore, the INTTPkCC0 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".

(c) Operation of TPkCCR1 register

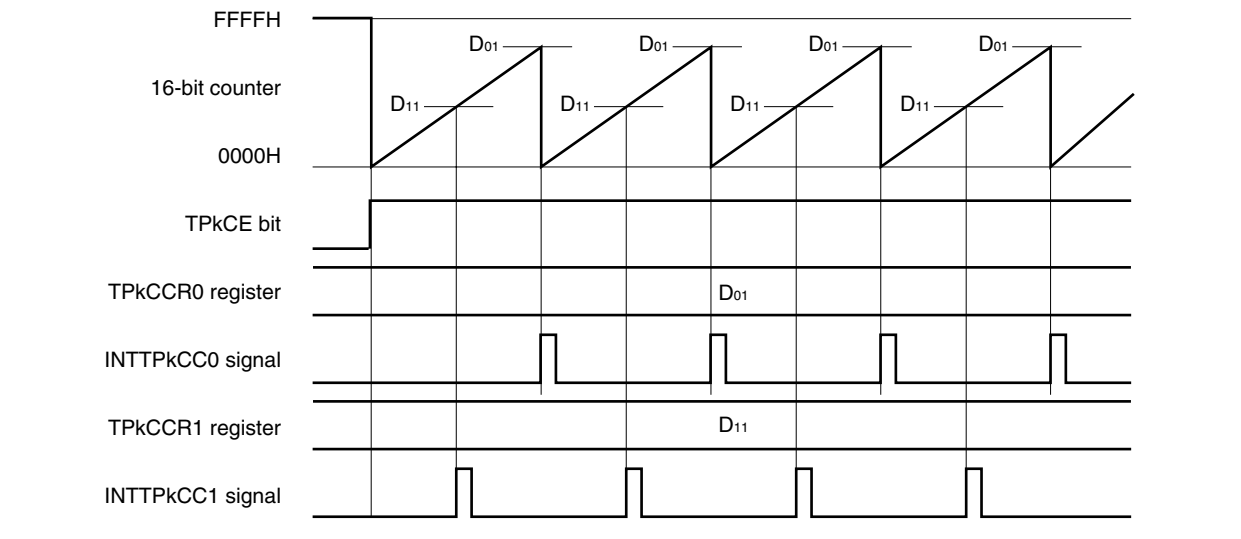
Figure 6-20. Configuration of TPkCCR1 Register



Remark k = 0, 2

If the set value of the TPkCCR1 register is smaller than the set value of the TPkCCR0 register, the INTTPkCC1 signal is generated once per cycle.

Figure 6-21. Timing Chart When $D_{01} \geq D_{11}$

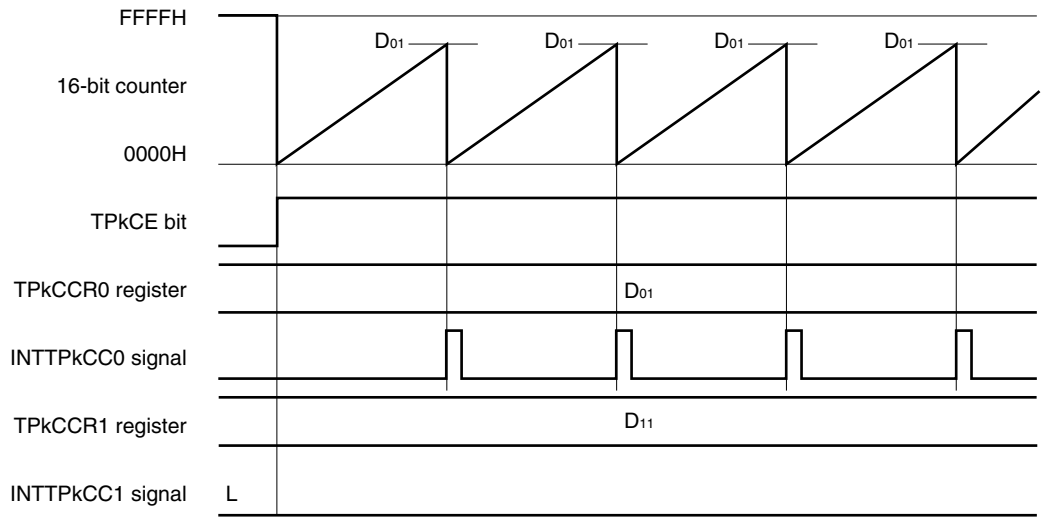


Remark k = 0, 2

If the set value of the TPkCCR1 register is greater than the set value of the TPkCCR0 register, the INTTPkCC1 signal is not generated because the count value of the 16-bit counter and the value of the TPkCCR1 register do not match.

When the TPkCCR1 register is not used, it is recommended to set its value to FFFFH.

Figure 6-22. Timing Chart When $D_{01} < D_{11}$



Remark k = 0, 2

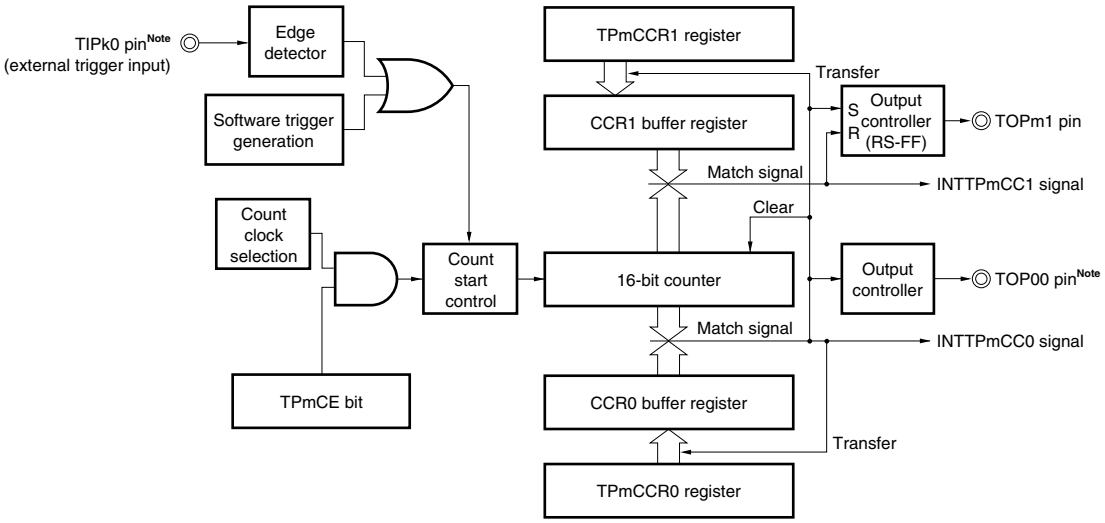
6.6.3 External trigger pulse output mode (TPmMD2 to TPmMD0 bits = 010)

This mode is valid only in TMP0, TMP2, and TMP3 (software trigger only for TMP3).

In the external trigger pulse output mode, 16-bit timer/event counter P waits for a trigger when the TPmCTL0.TPmCE bit is set to 1. When the valid edge of an external trigger input (TIPk0) is detected, 16-bit timer/event counter P starts counting, and outputs a PWM waveform from the TOPm1 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 TPmCCR0 register + 1 as half its cycle can also be output from the TOP00 pin.

<R> **Figure 6-23. Configuration in External Trigger Pulse Output Mode**

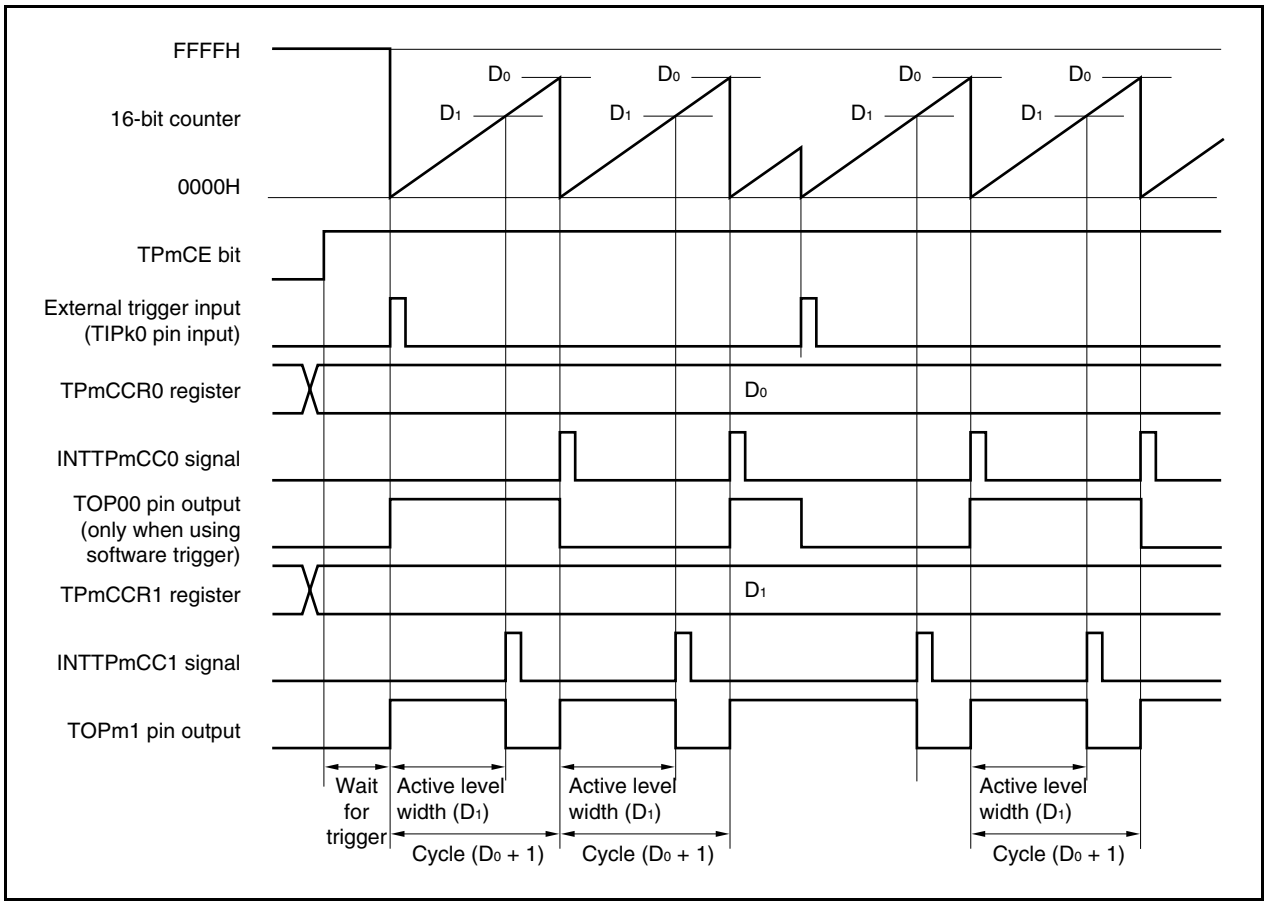


Note Because the external trigger input pin (TIPk0) and timer output pin (TOP00) share the same alternate-function pin, two or more functions cannot be used at the same time.

Caution In the external trigger pulse output mode, select the internal clock as the count clock (by clearing the TPkCTL1.TPkEEE bit to 0).

Remark m = 0, 2, 3
k = 0, 2

Figure 6-24. Basic Timing in External Trigger Pulse Output Mode



16-bit timer/event counter P waits for a trigger when the TPmCE 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 TOPm1 pin. If the trigger is generated again while the counter is operating, the counter is cleared to 0000H and <R> restarted. (The output of the TOP00 pin is inverted. The TOPm1 pin outputs 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 TPmCCR1 register) × Count clock cycle
Cycle = (Set value of TPmCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TPmCCR1 register)/(Set value of TPmCCR0 register + 1)

The compare match interrupt request signal INTTPmCC0 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 INTTPmCC1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TPmCCRa 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 (TIPk0) or setting the software trigger (TPmCTL1.TPmEST bit) to 1 is used as the trigger.

Remark m = 0, 2, 3
k = 0, 2
a = 0, 1

Figure 6-25. Setting of Registers in External Trigger Pulse Output Mode (1/2)

(a) TMPm control register 0 (TPmCTL0)

<R>

	TPmCE					TPmCKS2	TPmCKS1	TPmCKS0
TPmCTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock

0: Stop counting
1: Enable counting

(b) TMPm control register 1 (TPmCTL1)

<R>

	TPmEST	TPkEEE				TPmMD2	TPmMD1	TPmMD0
TPmCTL1	0	0/1	0	0	0	0	1	0

0, 1, 0:
External trigger pulse output mode

0: Operate on count clock selected by TPkCKS0 to TPkCKS2 bits

Generate software trigger when 1 is written

(c) TMPm I/O control register 0 (TPmIOC0)

<R>

					TPmOL1	TPmOE1	TP0OL0	TP0OE0
TPmIOC0	0	0	0	0	0/1	0/1	0/1	0/1[Note]

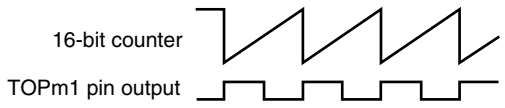
0: Disable TOP00 pin output
1: Enable TOP00 pin output

Settings of TOP00 pin output level in external trigger waiting state
0: Low level
1: High level

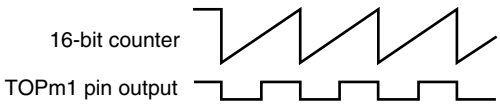
0: Disable TOPm1 pin output
1: Enable TOPm1 pin output

Settings of TOPm1 pin output level in external trigger waiting state
0: Low level
1: High level

- When TPmOL1 bit = 0



- When TPmOL1 bit = 1



<R> **Note** Clear this bit to 0 when the TOP00 pin is not used in the external trigger pulse output mode.

Figure 6-25. Setting of Registers in External Trigger Pulse Output Mode (2/2)

(d) TMPk I/O control register 2 (TPkIOC2)

<R>

					TPkEES1	TPkEES0	TPkETS1	TPkETS0
TPkIOC2	0	0	0	0	0	0	0/1	0/1

Select valid edge of external trigger input (TIPk0 pin)

(e) TMPm counter read buffer register (TPmCNT)

The value of the 16-bit counter can be read by reading the TPmCNT register.

(f) TMPm capture/compare registers 0 and 1 (TPmCCR0 and TPmCCR1)

If D_0 is set to the TPmCCR0 register and D_1 to the TPmCCR1 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

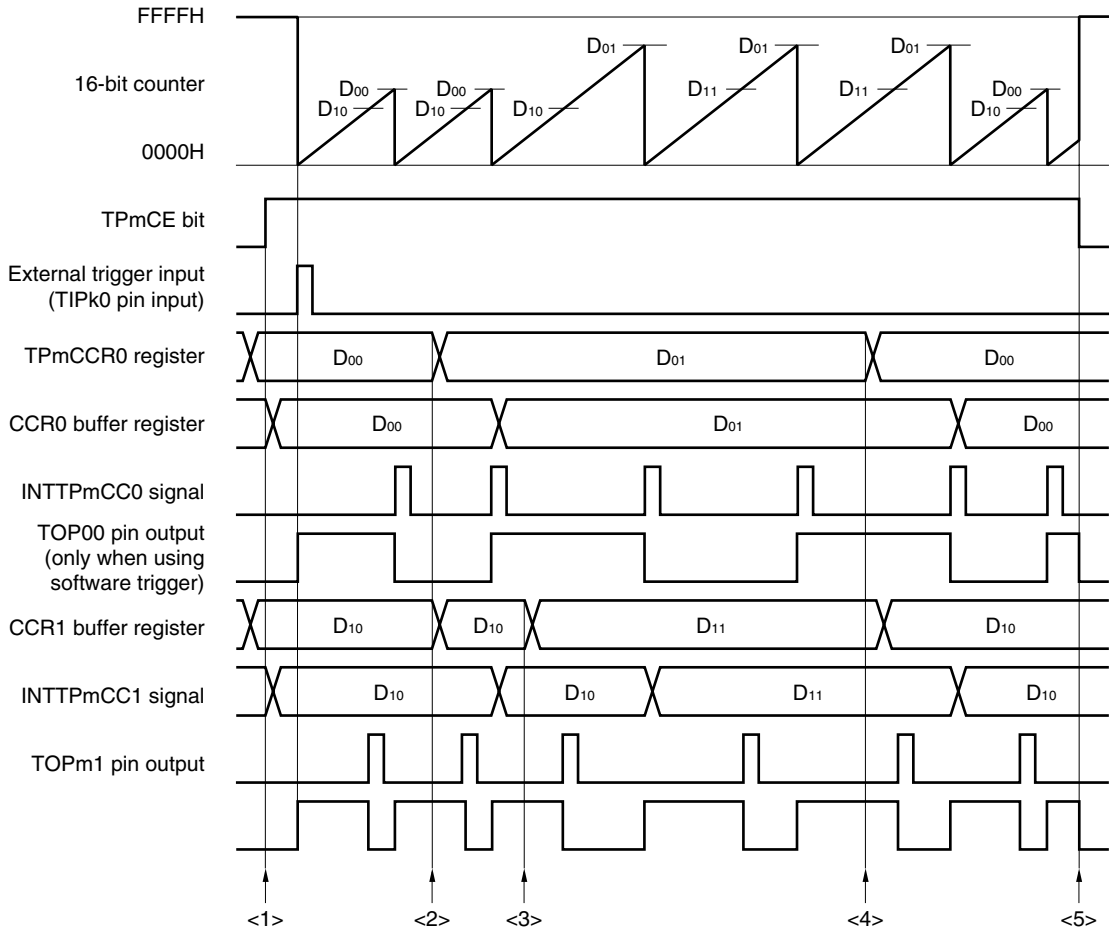
Remarks 1. TMPk I/O control register 1 (TPkIOC1) and TMPm option register 0 (TPmOPT0) are not used in the external trigger pulse output mode.

2. m = 0, 2, 3

k = 0, 2

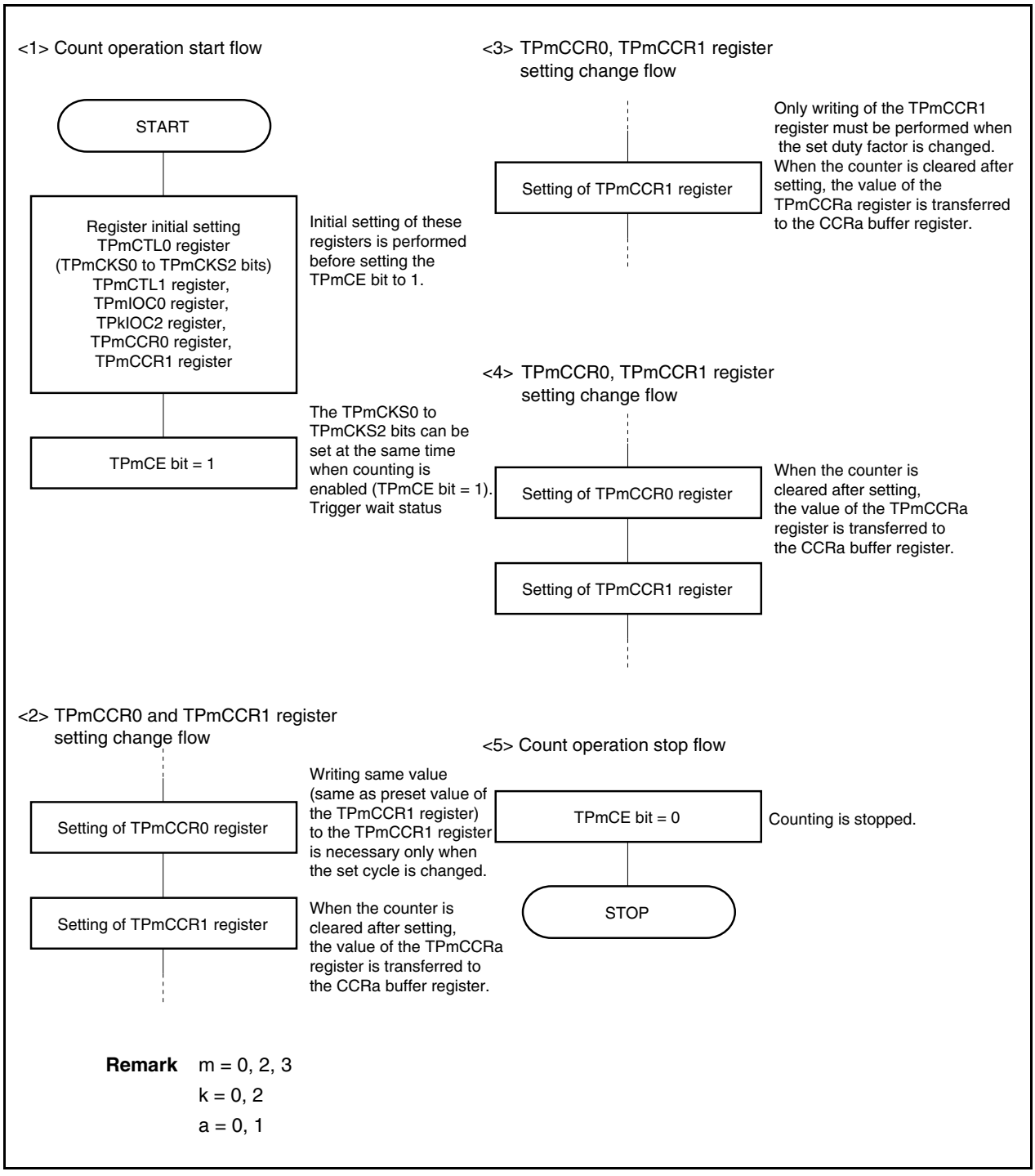
(1) Operation flow in external trigger pulse output mode

Figure 6-26. Software Processing Flow in External Trigger Pulse Output Mode (1/2)



Remark m = 0, 2, 3
k = 0, 2

Figure 6-26. Software Processing Flow in External Trigger Pulse Output Mode (2/2)



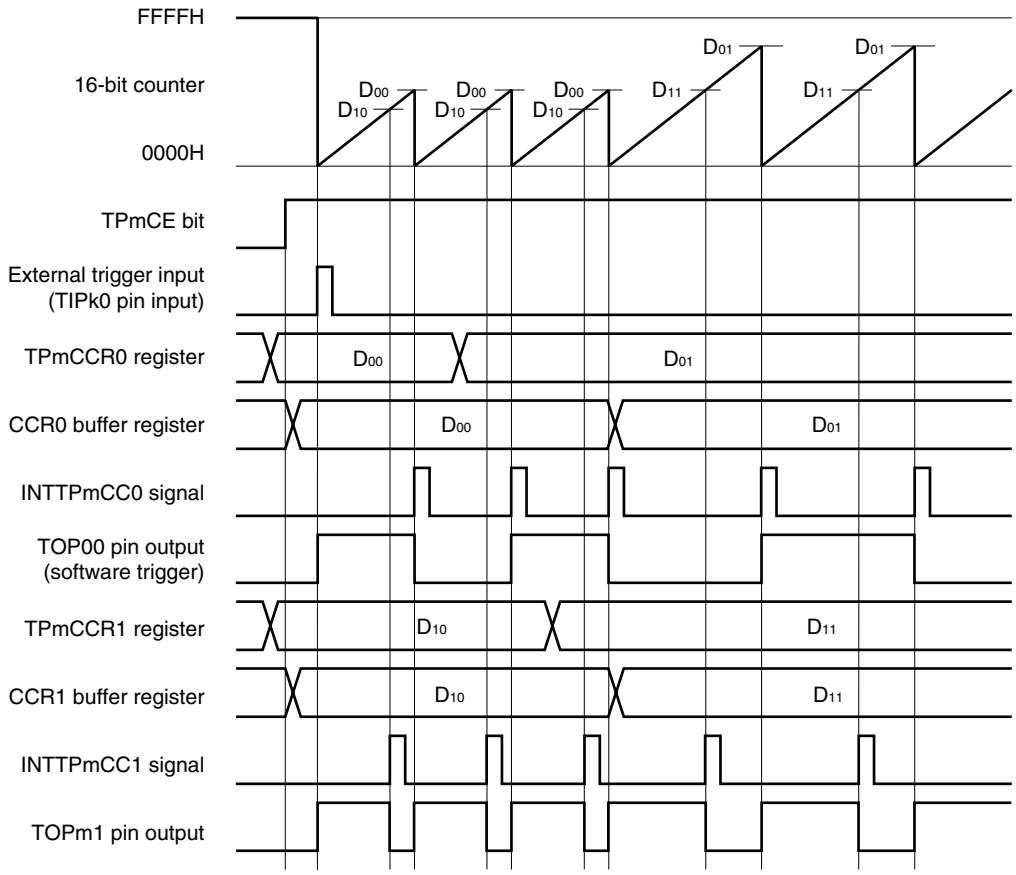
Remark m = 0, 2, 3
k = 0, 2
a = 0, 1

(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 TPmCCR1 register last.

Rewrite the TPmCCRa register after writing the TPmCCR1 register after the INTTPmCC0 signal is detected.



Remark m = 0, 2, 3
k = 0, 2

In order to transfer data from the TPmCCRa register to the CCRa buffer register, the TPmCCR1 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 TPmCCR0 register and then set the active level width to the TPmCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TPmCCR0 register, and then write the same value (same as preset value of the TPmCCR1 register) to the TPmCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TPmCCR1 register has to be set.

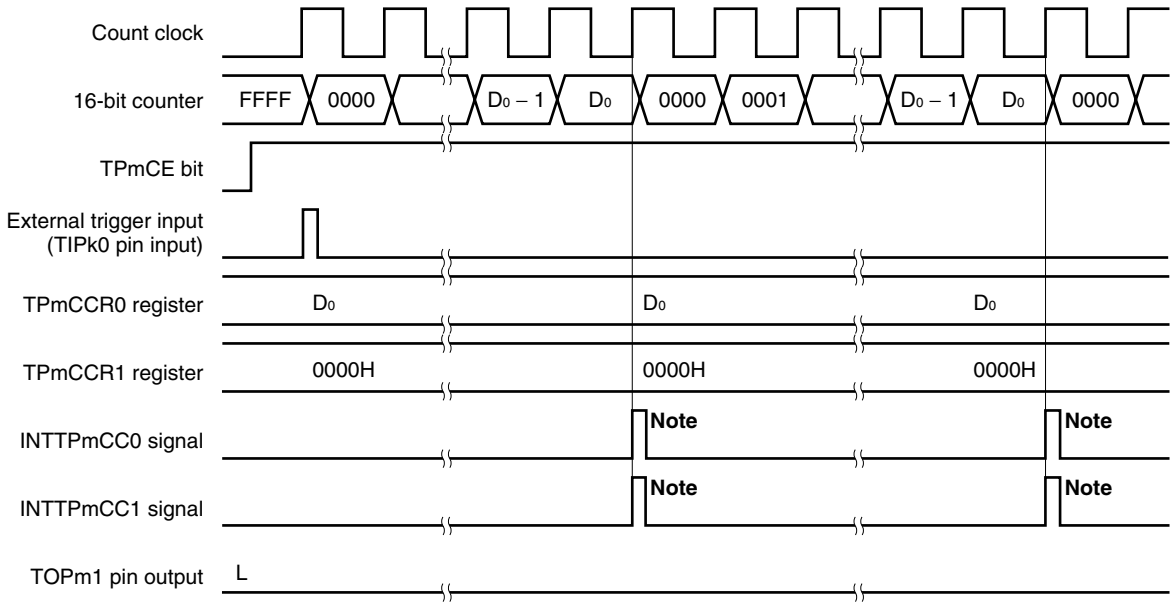
After data is written to the TPmCCR1 register, the value written to the TPmCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TPmCCR0 or TPmCCR1 register again after writing the TPmCCR1 register once, do so after the INTTPmCC0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TPmCCRa register to the CCRa buffer register conflicts with writing the TPmCCRa register.

Remark m = 0, 2, 3
a = 0, 1

(b) 0%/100% output of PWM waveform

To output a 0% waveform, set the TPmCCR1 register to 0000H. The 16-bit counter is cleared to 0000H and the INTTPmCC0 and INTTPmCC1 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.

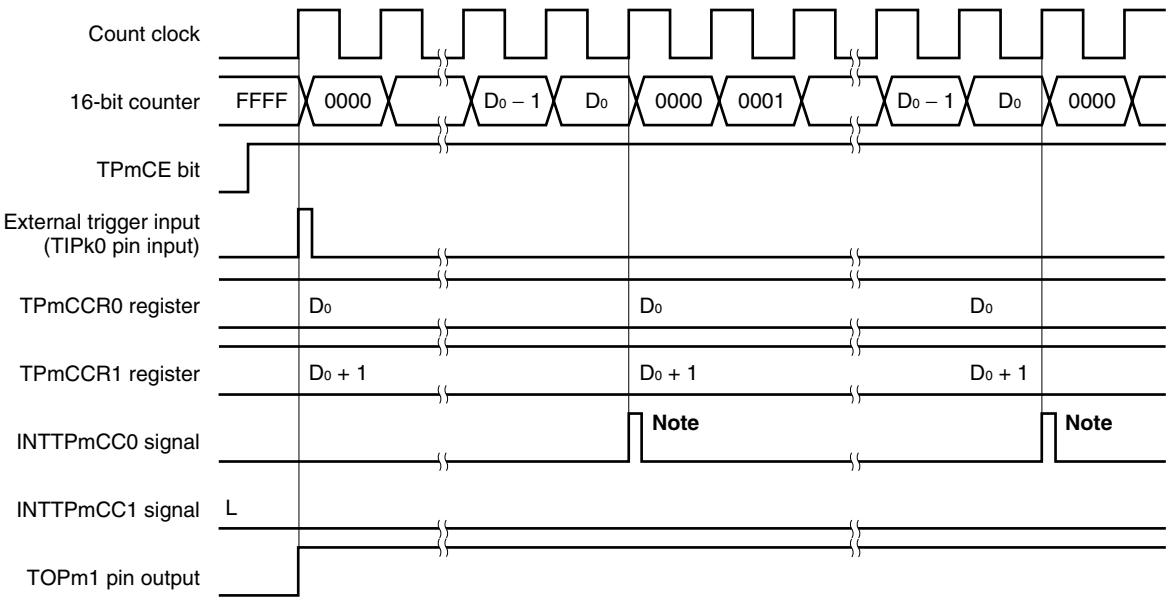


Note Actually, the timing is delayed by one operating clock (f_{XX}).

Remark m = 0, 2, 3
k = 0, 2

To output a 100% waveform, set a value of (set value of TPmCCR0 register + 1) to the TPmCCR1 register. If the set value of the TPmCCR0 register is FFFFH, 100% output cannot be produced.

<R>

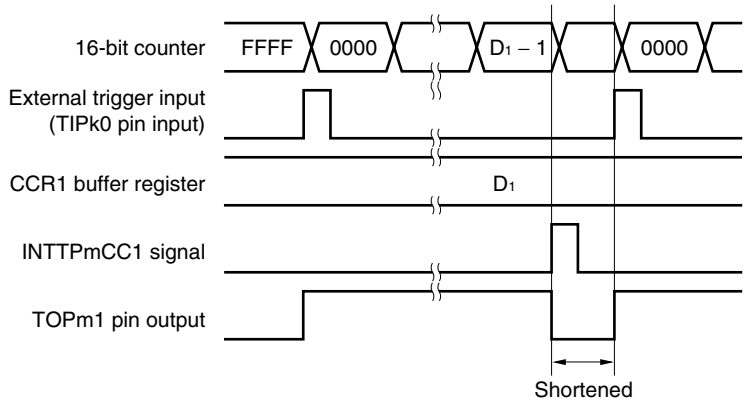


Note Actually, the timing is delayed by one operating clock (fxx).

Remark m = 0, 2, 3
k = 0, 2

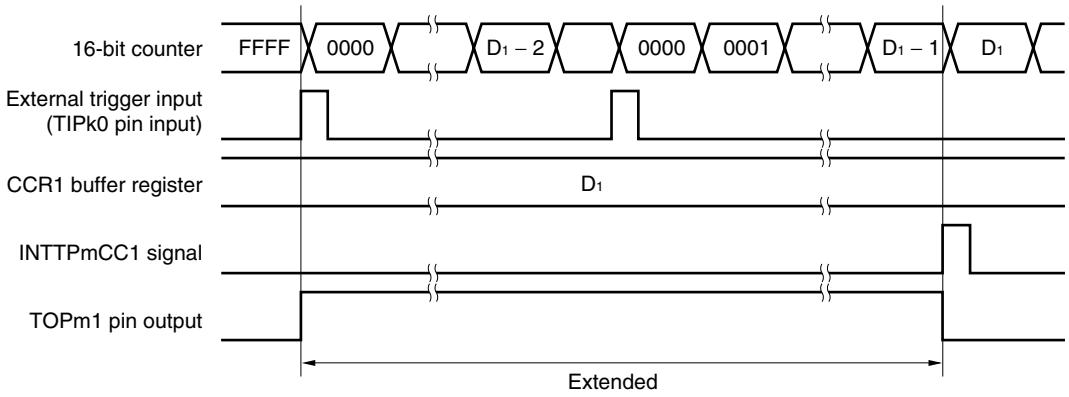
(c) Conflict between trigger detection and match with CCR1 buffer register

If the trigger is detected immediately after the INTTPmCC1 signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TOPm1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark m = 0, 2, 3
k = 0, 2

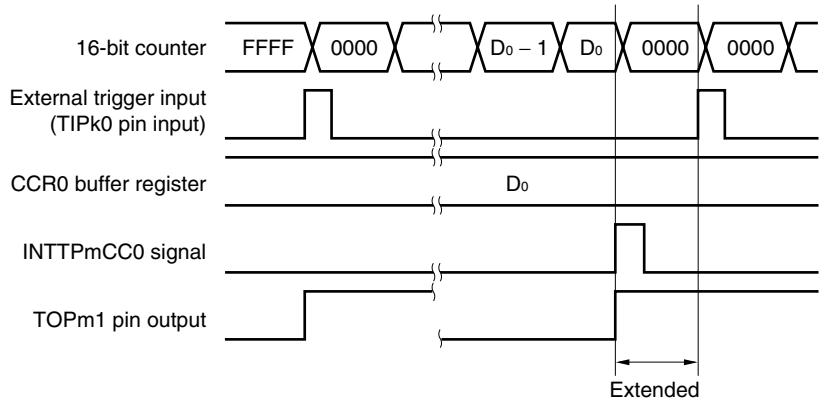
If the trigger is detected immediately before the INTTPmCC1 signal is generated, the INTTPmCC1 signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOPm1 pin remains active. Consequently, the active period of the PWM waveform is extended.



Remark m = 0, 2, 3
k = 0, 2

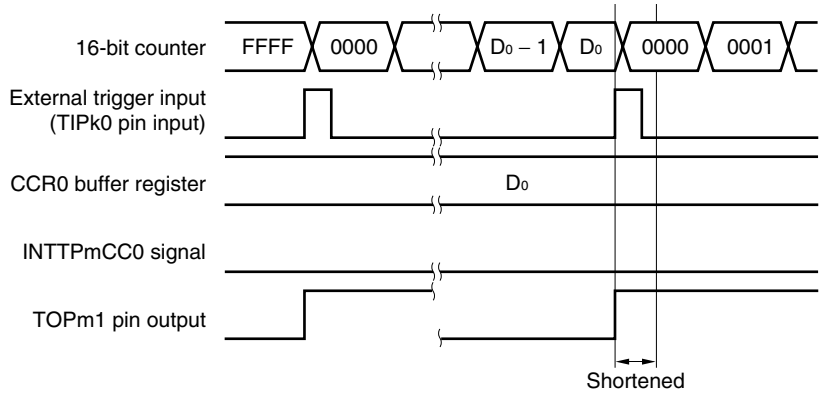
(d) Conflict between trigger detection and match with CCR0 buffer register

If the trigger is detected immediately after the INTTPmCC0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOPm1 pin is extended by time from generation of the INTTPmCC0 signal to trigger detection.



Remark m = 0, 2, 3
k = 0, 2

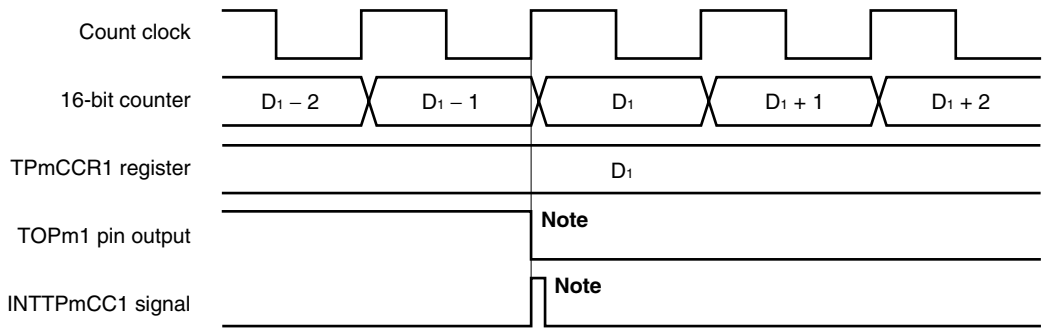
If the trigger is detected immediately before the INTTPmCC0 signal is generated, the INTTPmCC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOPm1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark m = 0, 2, 3
k = 0, 2

(e) Generation timing of compare match interrupt request signal (INTTPmCC1)

The timing of generation of the INTTPmCC1 signal in the external trigger pulse output mode differs from the timing of INTTPmCC1 signals in other mode; the INTTPmCC1 signal is generated when the count value of the 16-bit counter matches the value of the TPmCCR1 register.



Note Actually, the timing is delayed by one operating clock (fxx).

Remark m = 0, 2, 3

Usually, the INTTPmCC1 signal is generated in synchronization with the next count up, after the count value of the 16-bit counter matches the value of the TPmCCR1 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 TOPm1 pin.

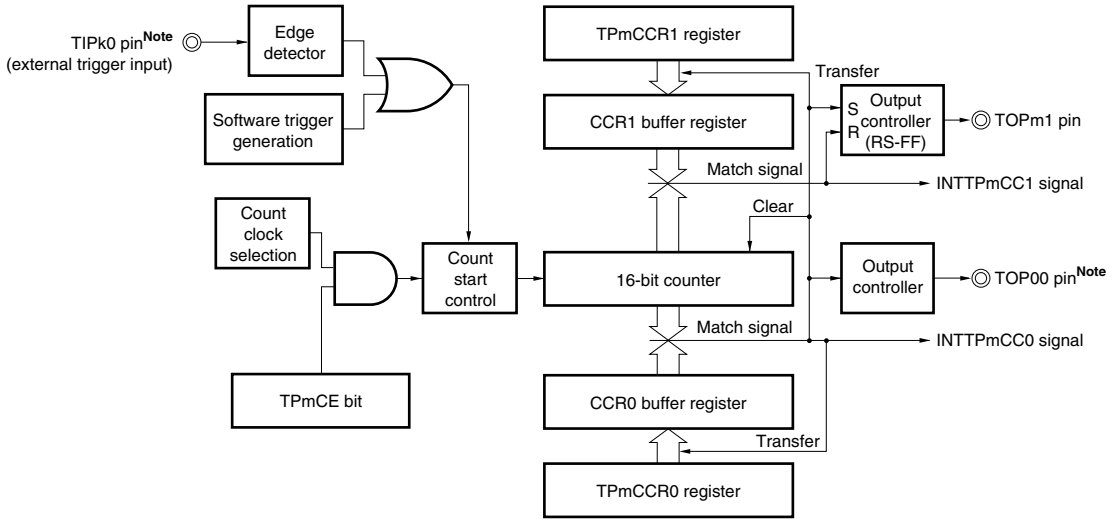
6.6.4 One-shot pulse output mode (TPmMD2 to TPmMD0 bits = 011)

This mode is valid only in TMP0, TMP2, and TMP3 (software trigger only for TMP3).

In the one-shot pulse output mode, 16-bit timer/event counter P waits for a trigger when the TPmCTL0.TPmCE bit is set to 1. When the valid edge of an external trigger input (TIPk0) is detected, 16-bit timer/event counter P starts counting, and outputs a one-shot pulse from the TOPm1 pin.

Instead of the external trigger, a software trigger can also be generated to output the pulse. When the software trigger is used, the TOP00 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).

<R> **Figure 6-27. Configuration in One-Shot Pulse Output Mode**

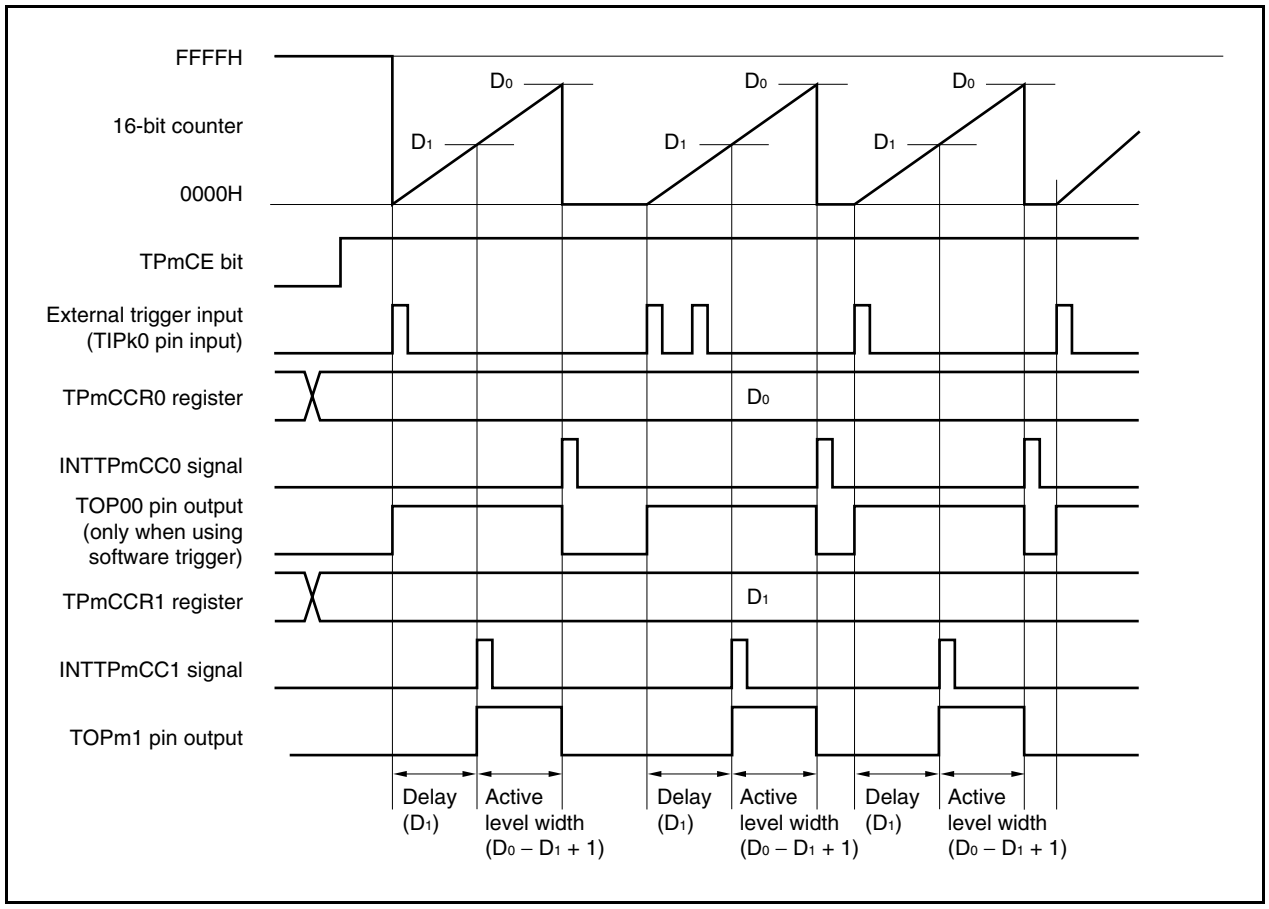


Note Because the external trigger input pin (TIPk0) and timer output pin (TOP00) share the same alternate-function pin, two functions cannot be used at the same time.

Caution In the one-shot pulse output mode, select the internal clock as the count clock (by clearing the TPkCTL1.TPkEEE bit to 0).

Remark m = 0, 2, 3
k = 0, 2

Figure 6-28. Basic Timing in One-Shot Pulse Output Mode



When the TPmCE bit is set to 1, 16-bit timer/event counter P 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 TOPm1 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 TPmCCR1 register) × Count clock cycle

Active level width = (Set value of TPmCCR0 register − Set value of TPmCCR1 register + 1) × Count clock cycle

The compare match interrupt request signal INTTPmCC0 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 INTTPmCC1 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 (TIPk0 pin) or setting the software trigger (TPmCTL1.TPmEST bit) to 1 is used as the trigger.

Remark m = 0, 2, 3
k = 0, 2

Figure 6-29. Setting of Registers in One-Shot Pulse Output Mode (1/2)

(a) TMPm control register 0 (TPmCTL0)

<R>

	TPmCE					TPmCKS2	TPmCKS1	TPmCKS0
TPmCTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock

0: Stop counting
1: Enable counting

(b) TMPm control register 1 (TPmCTL1)

<R>

		TPmEST	TPkEEE			TPmMD2	TPmMD1	TPmMD0
TPmCTL1	0	0/1	0	0	0	0	1	1

0, 1, 1:
One-shot pulse output mode

0: Operate on count clock selected by TPkCKS0 to TPkCKS2 bits

Generate software trigger when 1 is written

(c) TMPm I/O control register 0 (TPmIOC0)

<R>

					TPmOL1	TPmOE1	TP0OL0	TP0OE0
TPmIOC0	0	0	0	0	0/1	0/1	0/1	0/1[Note]

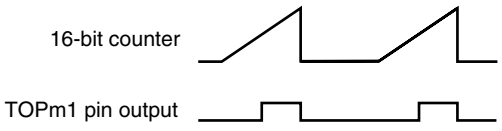
0: Disable TOP00 pin output
1: Enable TOP00 pin output

Setting of TOP00 pin output level in external trigger waiting state
0: Low level
1: High level

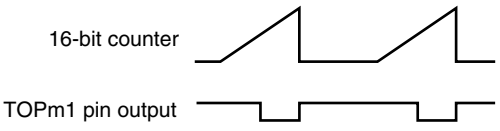
0: Disable TOPm1 pin output
1: Enable TOPm1 pin output

Setting of TOPm1 pin output level in external trigger waiting state
0: Low level
1: High level

• When TPmOL1 bit = 0



• When TPmOL1 bit = 1



<R>

Note Clear this bit to 0 when the TOP00 pin is not used in the one-shot pulse output mode.

Figure 6-29. Setting of Registers in One-Shot Pulse Output Mode (2/2)

(d) TMPk I/O control register 2 (TPkIOC2)

<R>

							TPkEES1	TPkEES0	TPkETS1	TPkETS0
TPkIOC2	0	0	0	0	0	0	0/1	0/1		

Select valid edge of external trigger input (TIPk0 pin)

(e) TMPm counter read buffer register (TPmCNT)

The value of the 16-bit counter can be read by reading the TPmCNT register.

(f) TMPm capture/compare registers 0 and 1 (TPmCCR0 and TPmCCR1)

If D_0 is set to the TPmCCR0 register and D_1 to the TPmCCR1 register, the active level width and output delay period of the one-shot pulse are as follows.

<R> Active level width = $(D_0 - D_1 + 1) \times$ Count clock cycle

Output delay period = $D_1 \times$ Count clock cycle

<R> **Caution One-shot pulses are not output even in the one-shot pulse output mode, if the value set in the TPmCCR1 register is greater than that set in the TPmCCR0 register.**

Remarks 1. TMPk I/O control register 1 (TPkIOC1) and TMPm option register 0 (TPmOPT0) are not used in the one-shot pulse output mode.

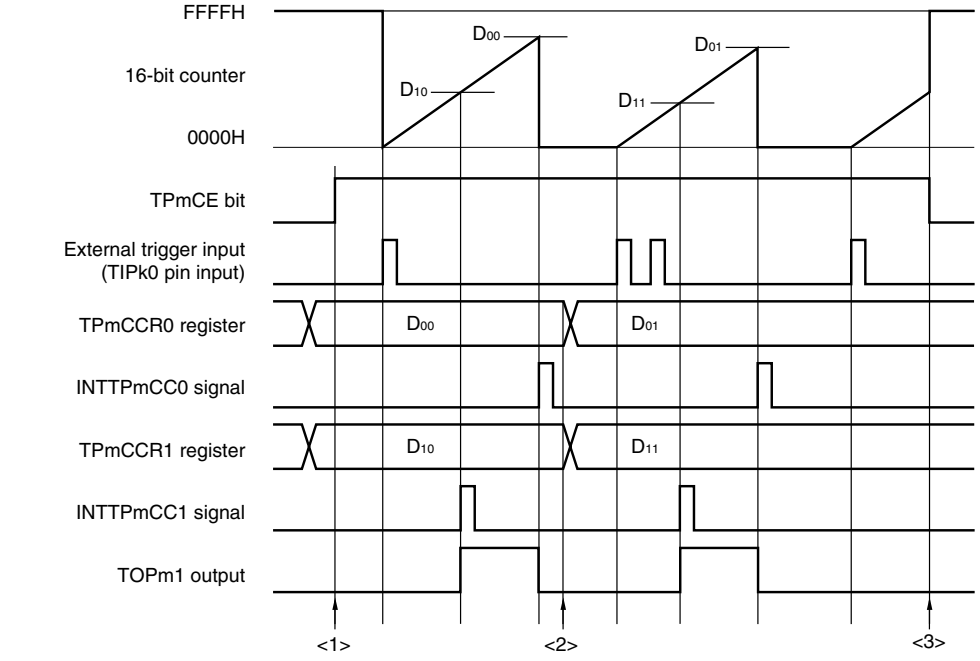
2. m = 0, 2, 3

k = 0, 2

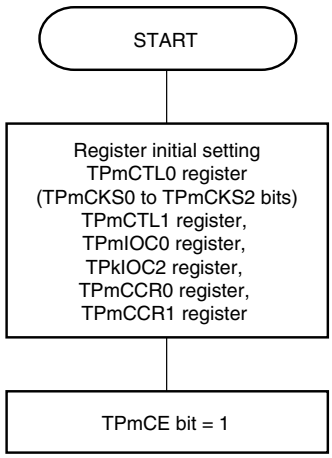
(1) Operation flow in one-shot pulse output mode

<R>

Figure 6-30. Software Processing Flow in One-Shot Pulse Output Mode



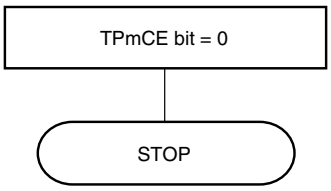
<1> Count operation start flow



Initial setting of these registers is performed before setting the TPmCE bit to 1.

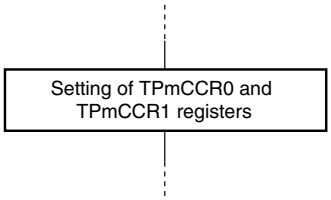
The TPmCKS0 to TPmCKS2 bits can be set at the same time when counting has been started (TPmCE bit = 1). Trigger wait status

<3> Count operation stop flow



Count operation is stopped

<2> TPmCCR0, TPmCCR1 register setting change flow



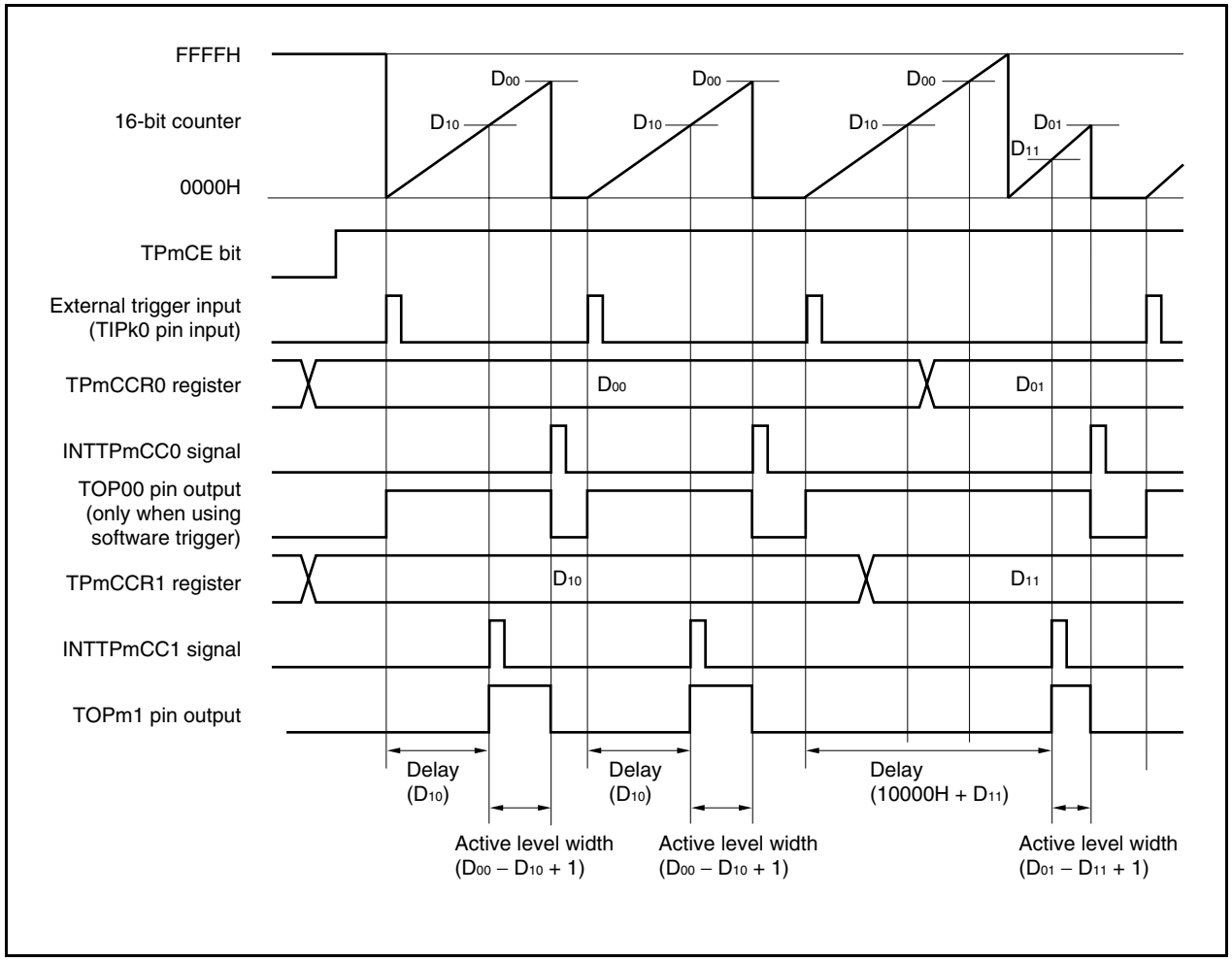
As rewriting the TPmCCRa register immediately forwards to the CCRa buffer register, rewriting immediately after the generation of the INTTPmCC0 signal is recommended.

Remark m = 0, 2, 3, k = 0, 2, a = 0, 1

(2) Operation timing in one-shot pulse output mode

(a) Note on rewriting TPmCCRa register

If the value of the TPmCCRa 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.



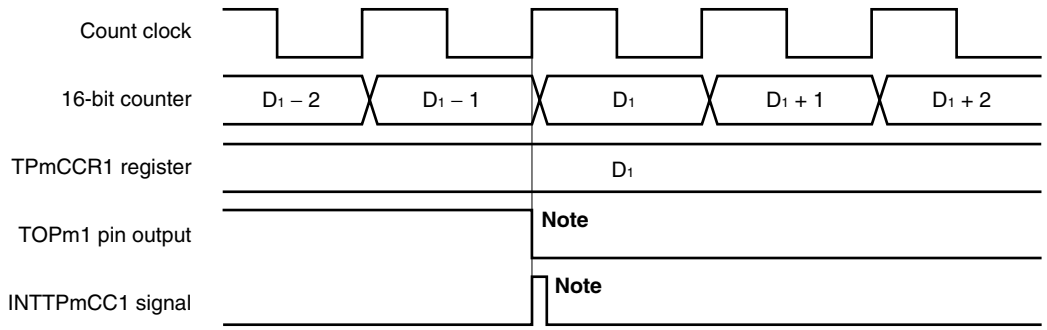
When the TPmCCR0 register is rewritten from D_{00} to D_{01} and the TPmCCR1 register from D_{10} to D_{11} where $D_{00} > D_{01}$ and $D_{10} > D_{11}$, if the TPmCCR1 register is rewritten when the count value of the 16-bit counter is greater than D_{11} and less than D_{10} and if the TPmCCR0 register 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 INTTPmCC1 signal and asserts the TOPm1 pin. When the count value matches D_{01}, the counter generates the INTTPmCC0 signal, deasserts the TOPm1 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 m = 0, 2, 3, k = 0, 2, a = 0, 1

(b) Generation timing of compare match interrupt request signal (INTTPmCC1)

The generation timing of the INTTPmCC1 signal in the one-shot pulse output mode is different from INTTPmCC1 signals in other mode; the INTTPmCC1 signal is generated when the count value of the 16-bit counter matches the value of the TPmCCR1 register.



Note Actually, the timing is delayed by one operating clock (fxx).

Remark m = 0, 2, 3

Usually, the INTTPmCC1 signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TPmCCR1 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 TOPm1 pin.

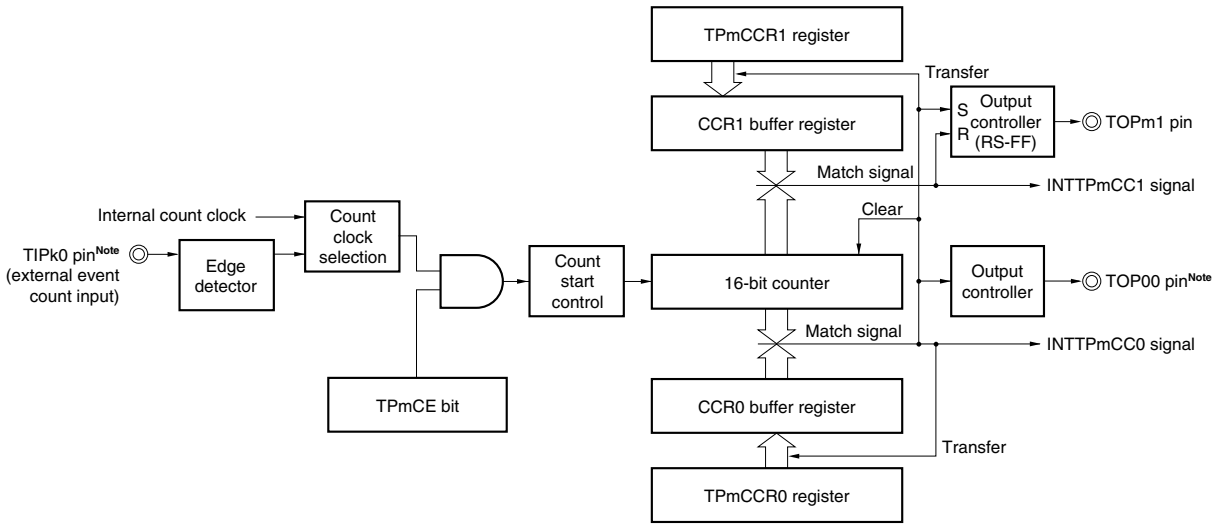
6.6.5 PWM output mode (TPmMD2 to TPmMD0 bits = 100)

This mode is valid only in TMP0, TMP2, and TMP3.

In the PWM output mode, a PWM waveform is output from the TOPm1 pin when the TPmCTL0.TPmCE bit is set to 1.

In addition, a PWM waveform with a duty factor of 50% with the set value of the TPmCCR0 register + 1 as half its cycle is output from the TOP00 pin.

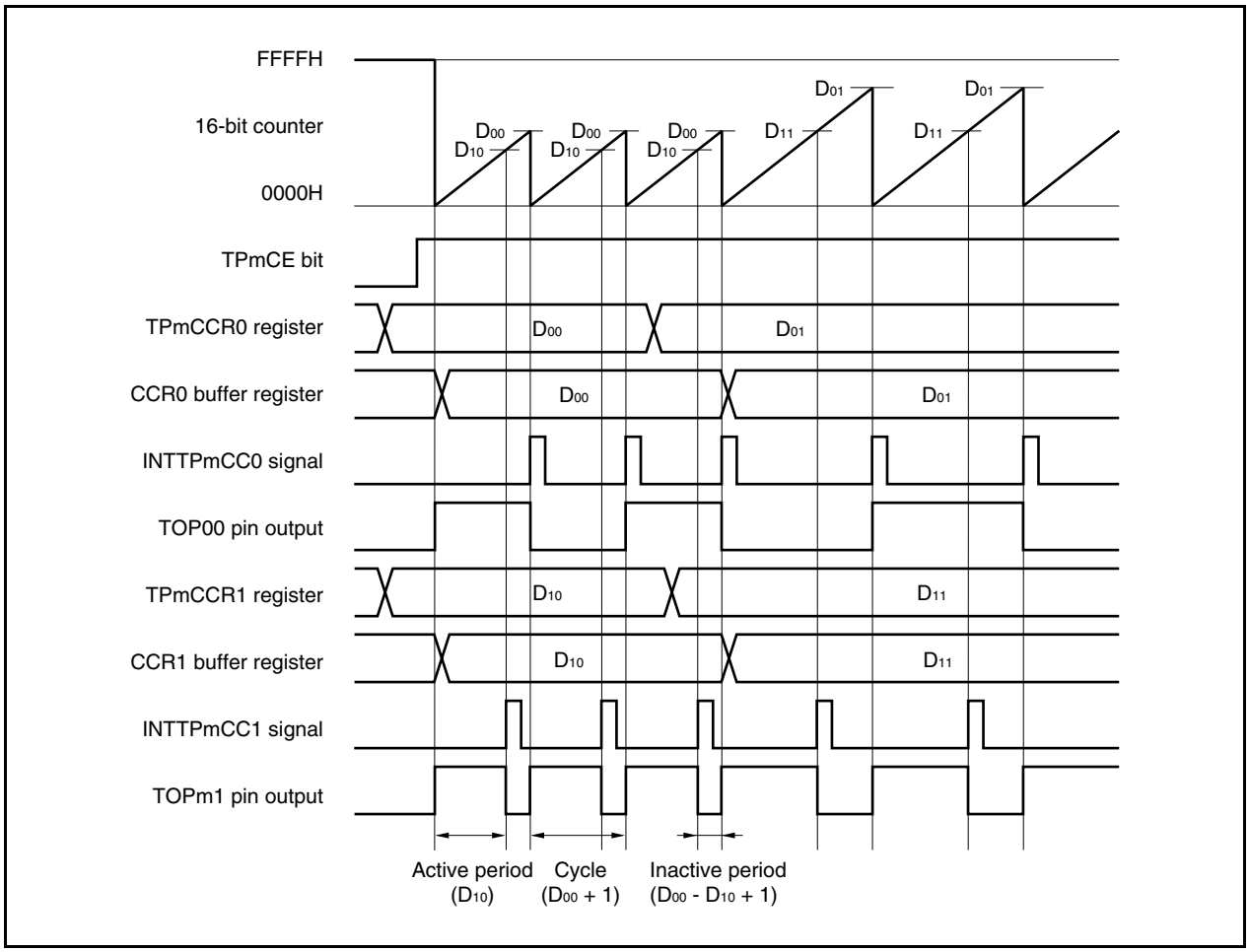
Figure 6-31. Configuration in PWM Output Mode



Note Because the external event count input pin (TIP00) and timer output pin (TOP00) share the same alternate-function pin, two functions cannot be used at the same time.

Remark m = 0, 2, 3
k = 0, 2

Figure 6-32. Basic Timing in PWM Output Mode



When the TPmCE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a PWM waveform from the TOPm1 pin.

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TPmCCR1 register) × Count clock cycle
Cycle = (Set value of TPmCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TPmCCR1 register)/(Set value of TPmCCR0 register + 1)

The PWM waveform can be changed by rewriting the TPmCCRa 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 INTTPmCC0 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 INTTPmCC1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TPmCCRa 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 m = 0, 2, 3, a = 0, 1

Figure 6-33. Register Setting in PWM Output Mode (1/2)

(a) TMPm control register 0 (TPmCTL0)

	TPmCE					TPmCKS2	TPmCKS1	TPmCKS0
TPmCTL0	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 TPmCTL1.TPkEEE bit = 1.

(b) TMPm control register 1 (TPmCTL1)

		TPmEST	TPkEEE			TPmMD2	TPmMD1	TPmMD0
TPmCTL1	0	0	0/1	0	0	1	0	0

1, 0, 0:
PWM output mode

0: Operate on count clock selected by TPkCKS0 to TPkCKS2 bits
1: Count with external event count input signal

(c) TMPm I/O control register 0 (TPmIOC0)

<R>

					TPmOL1	TPmOE1	TP0OL0	TP0OE0
TPmIOC0	0	0	0	0	0/1	0/1	0/1	0/1[Note]

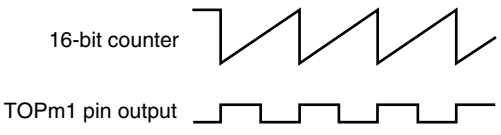
0: Disable TOP00 pin output
1: Enable TOP00 pin output

Setting of TOP00 pin output level before count operation
0: Low level
1: High level

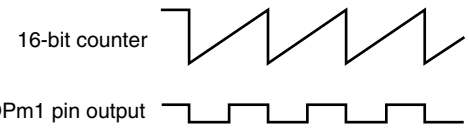
0: Disable TOPm1 pin output
1: Enable TOPm1 pin output

Setting of TOPm1 pin output level before count operation
0: Low level
1: High level

- When TPmOL1 bit = 0



- When TPmOL1 bit = 1

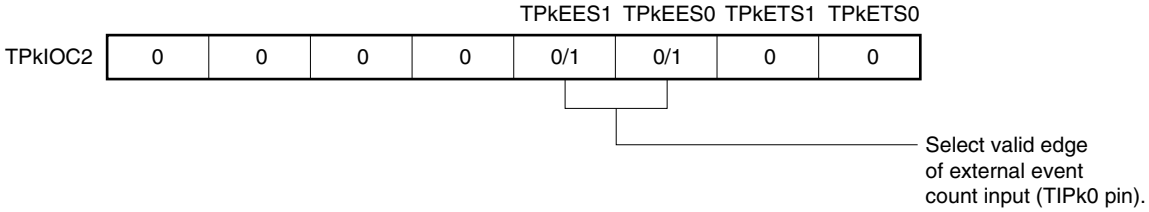


<R>

Note Clear this bit to 0 when the TOP00 pin is not used in the PWM output mode.

Figure 6-33. Register Setting in PWM Output Mode (2/2)

(d) TMPk I/O control register 2 (TPkIOC2)



(e) TMPm counter read buffer register (TPmCNT)

The value of the 16-bit counter can be read by reading the TPmCNT register.

(f) TMPm capture/compare registers 0 and 1 (TPmCCR0 and TPmCCR1)

If D_0 is set to the TPmCCR0 register and D_1 to the TPmCCR1 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

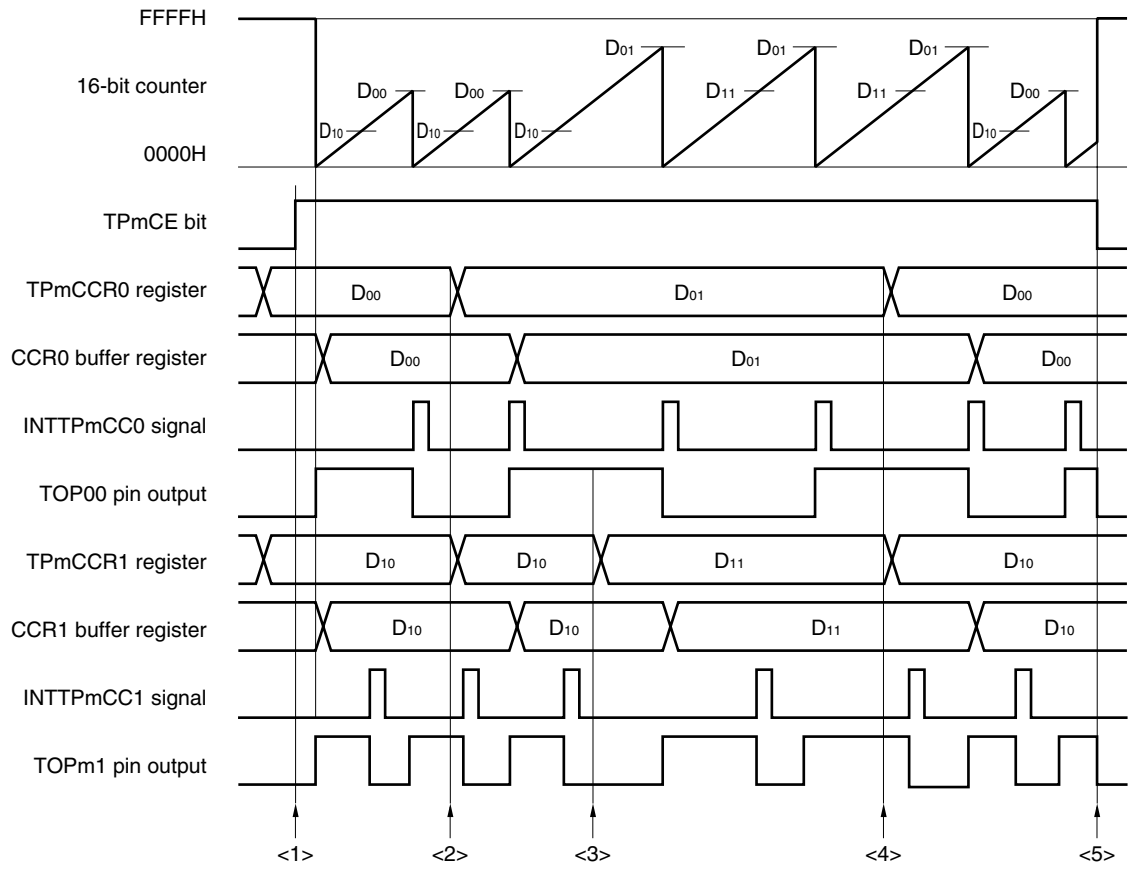
Remarks 1. TMPk I/O control register 1 (TPkIOC1) and TMPk option register 0 (TPkOPT0) are not used in the PWM output mode.

2. m = 0, 2, 3

k = 0, 2

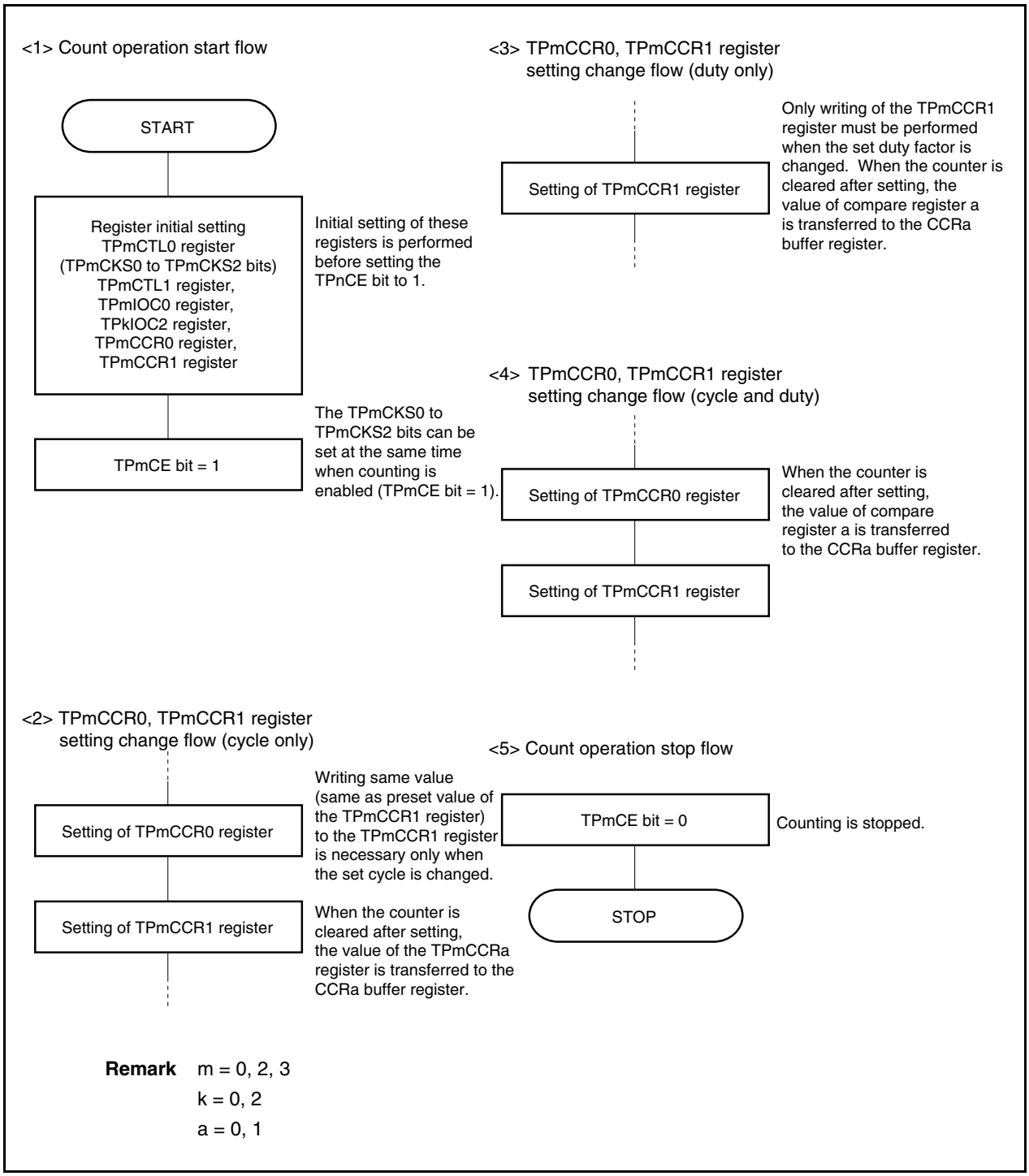
(1) Operation flow in PWM output mode

Figure 6-34. Software Processing Flow in PWM Output Mode (1/2)



Remark m = 0, 2, 3
k = 0, 2

Figure 6-34. Software Processing Flow in PWM Output Mode (2/2)



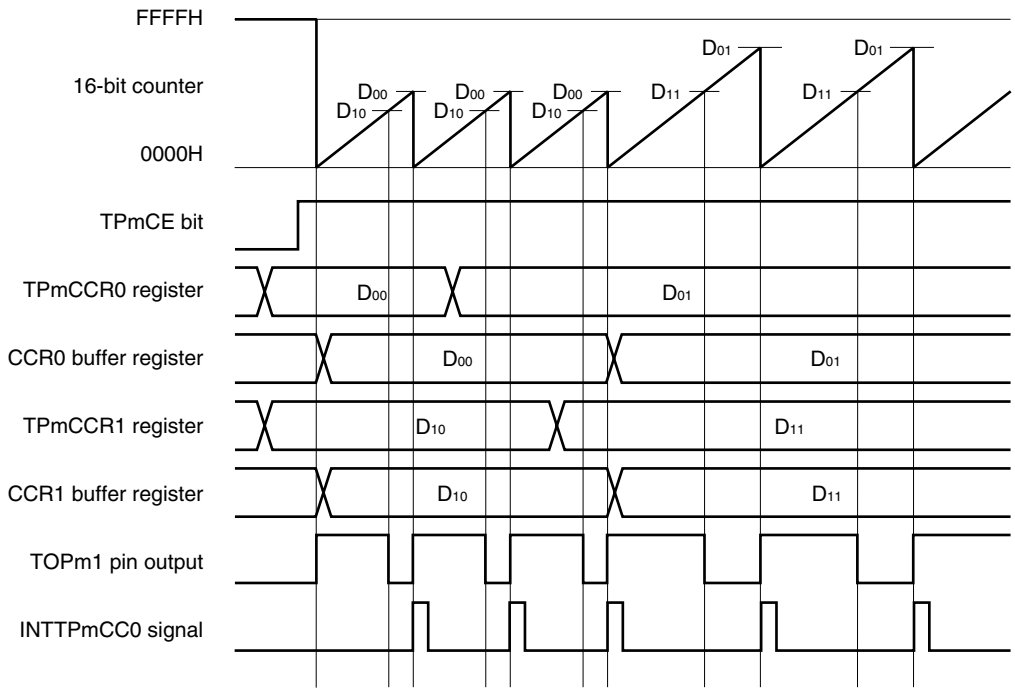
Remark m = 0, 2, 3
k = 0, 2
a = 0, 1

(2) PWM output mode operation timing

(a) Changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TPmCCR1 register last.

Rewrite the TPmCCRa register after writing the TPmCCR1 register after the INTTPmCC1 signal is detected.



To transfer data from the TPmCCRa register to the CCRa buffer register, the TPmCCR1 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 TPmCCR0 register and then set the active level to the TPmCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TPmCCR0 register, and then write the same value (same as preset value of the TPmCCR1 register) to the TPmCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TPmCCR1 register has to be set.

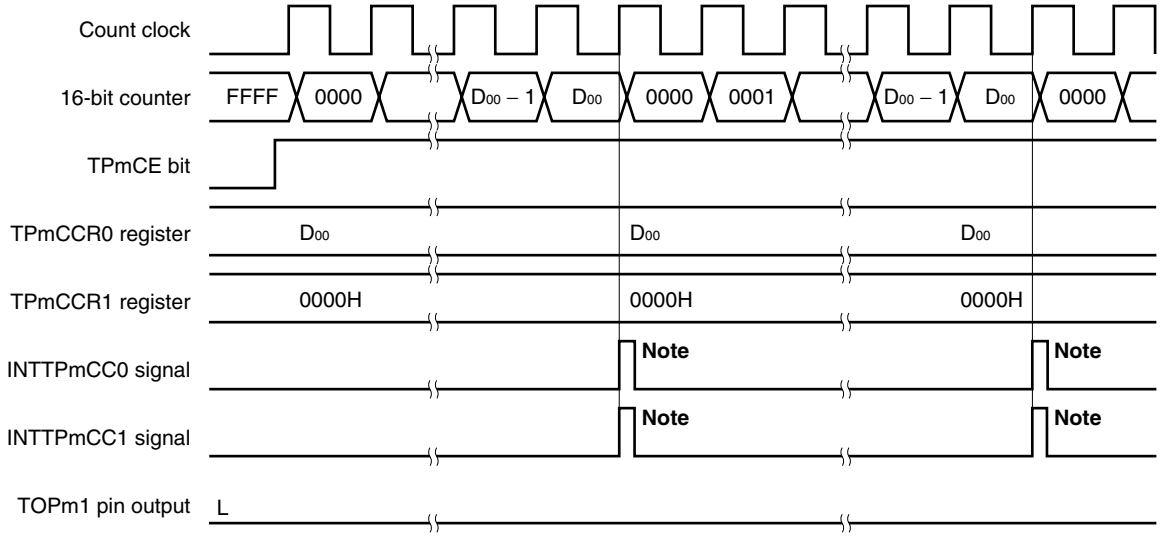
After data is written to the TPmCCR1 register, the value written to the TPmCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TPmCCR0 or TPmCCR1 register again after writing the TPmCCR1 register once, do so after the INTTPmCC0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TPmCCRa register to the CCRa buffer register conflicts with writing the TPmCCRa register.

Remark m = 0, 2, 3
a = 0, 1

(b) 0%/100% output of PWM waveform

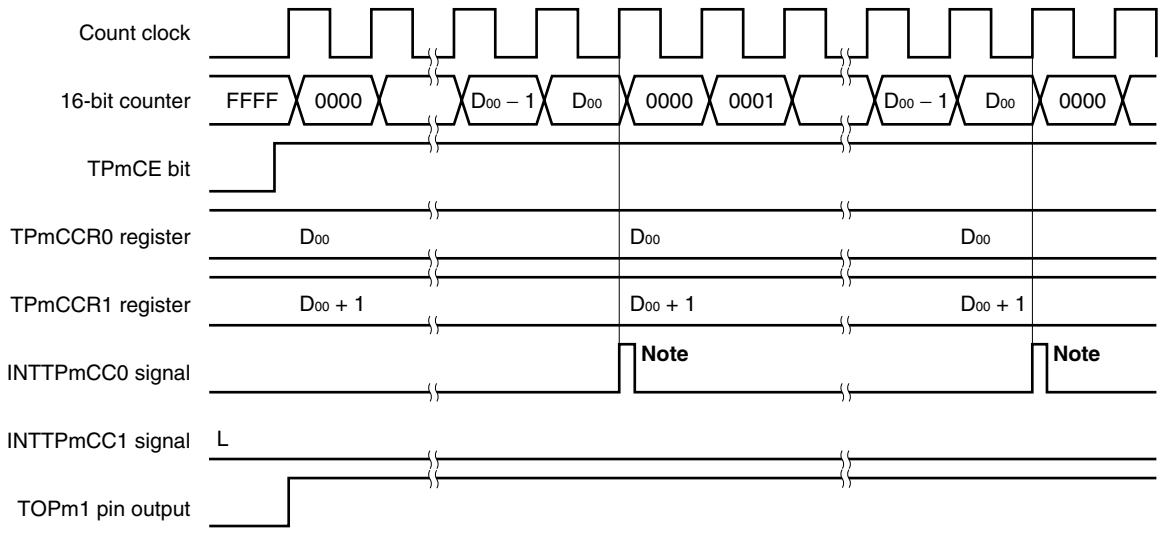
To output a 0% waveform, set the TPmCCR1 register to 0000H. The 16-bit counter is cleared to 0000H and the INTTPmCC0 and INTTPmCC1 signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.



Note Actually, the timing is delayed by one operating clock (fxx).

Remark m = 0, 2, 3

To output a 100% waveform, set a value of (set value of TPmCCR0 register + 1) to the TPmCCR1 register. If the set value of the TPmCCR0 register is FFFFH, 100% output cannot be produced.

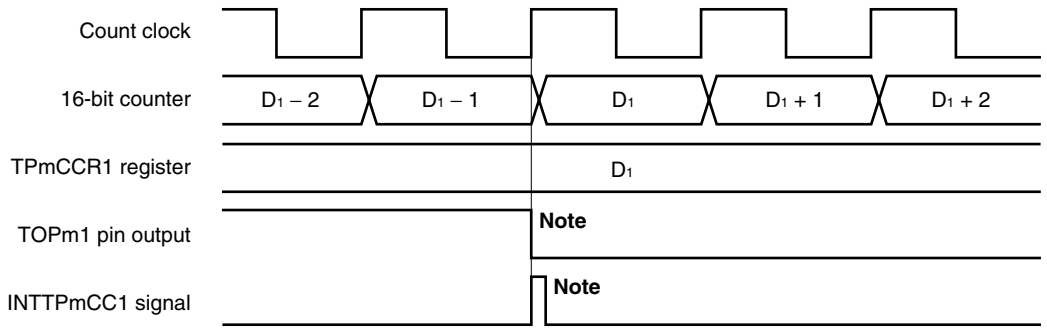


Note Actually, the timing is delayed by one operating clock (fxx).

Remark m = 0, 2, 3

(c) Generation timing of compare match interrupt request signal (INTTPmCC1)

The timing of generation of the INTTPmCC1 signal in the PWM output mode differs from the timing of INTTPmCC1 signals in other modes; the INTTPmCC1 signal is generated when the count value of the 16-bit counter matches the value of the TPmCCR1 register.



Note Actually, the timing is delayed by one operating clock (fxx).

Remark m = 0, 2, 3

Usually, the INTTPmCC1 signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TPmCCR1 register.

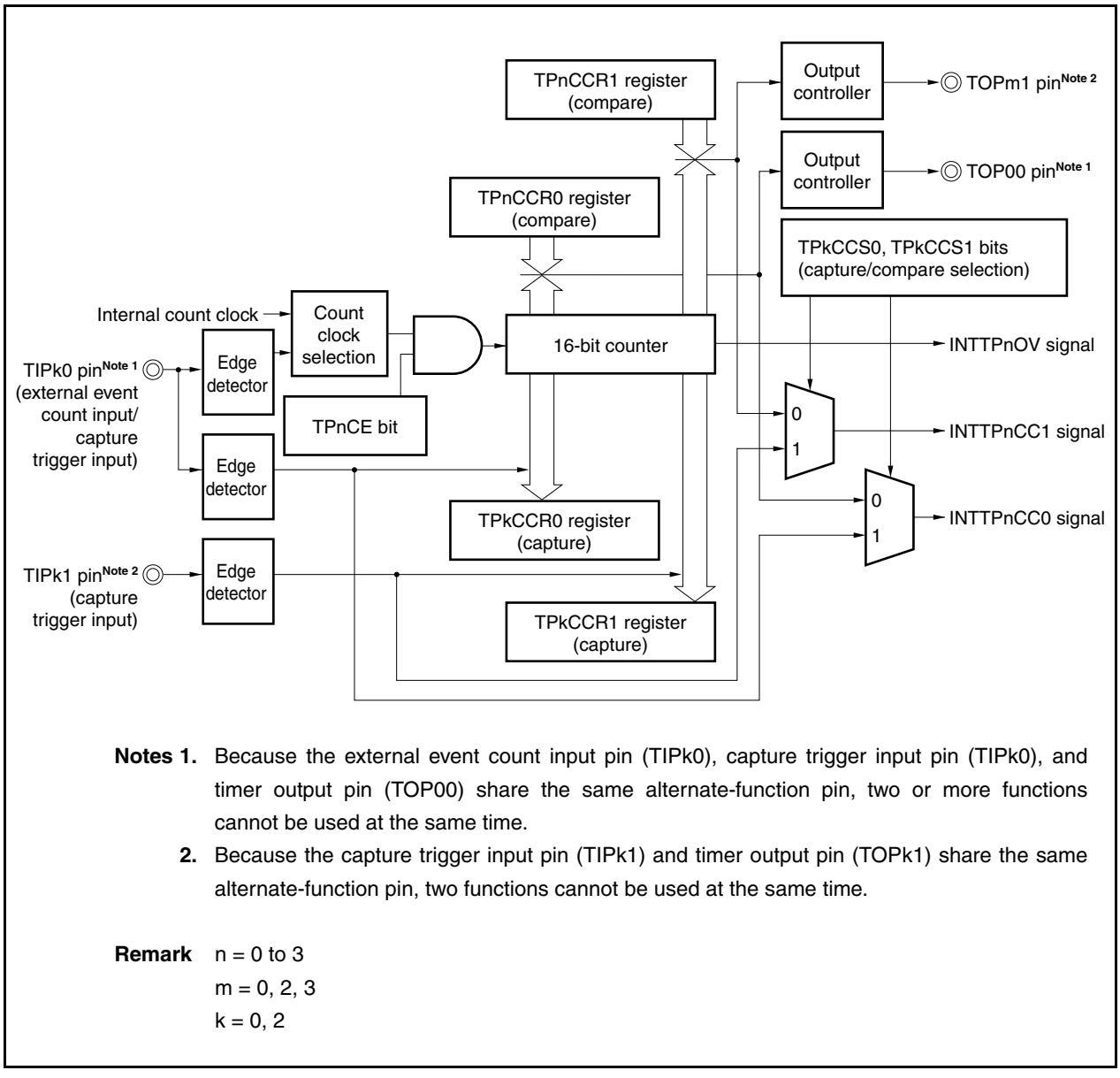
In the PWM output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the output signal of the TOPm1 pin.

6.6.6 Free-running timer mode (TPnMD2 to TPnMD0 bits = 101)

The compare function is valid for all of TMP0 to TMP3. The capture function is valid only for TMP0 and TMP2.

In the free-running timer mode, 16-bit timer/event counter P starts counting when the TPnCTL0.TPnCE bit is set to 1. At this time, the TPkCCR0 and TPkCCR1 registers can be used as compare registers or capture registers, depending on the setting of the TPkOPT0.TPkCCS0 and TPkOPT0.TPkCCS1 bits.

Figure 6-35. Configuration in Free-Running Timer Mode



Notes 1. Because the external event count input pin (TIPk0), capture trigger input pin (TIPk0), and timer output pin (TOP00) share the same alternate-function pin, two or more functions cannot be used at the same time.

2. Because the capture trigger input pin (TIPk1) and timer output pin (TOPk1) share the same alternate-function pin, two functions cannot be used at the same time.

Remark n = 0 to 3
m = 0, 2, 3
k = 0, 2

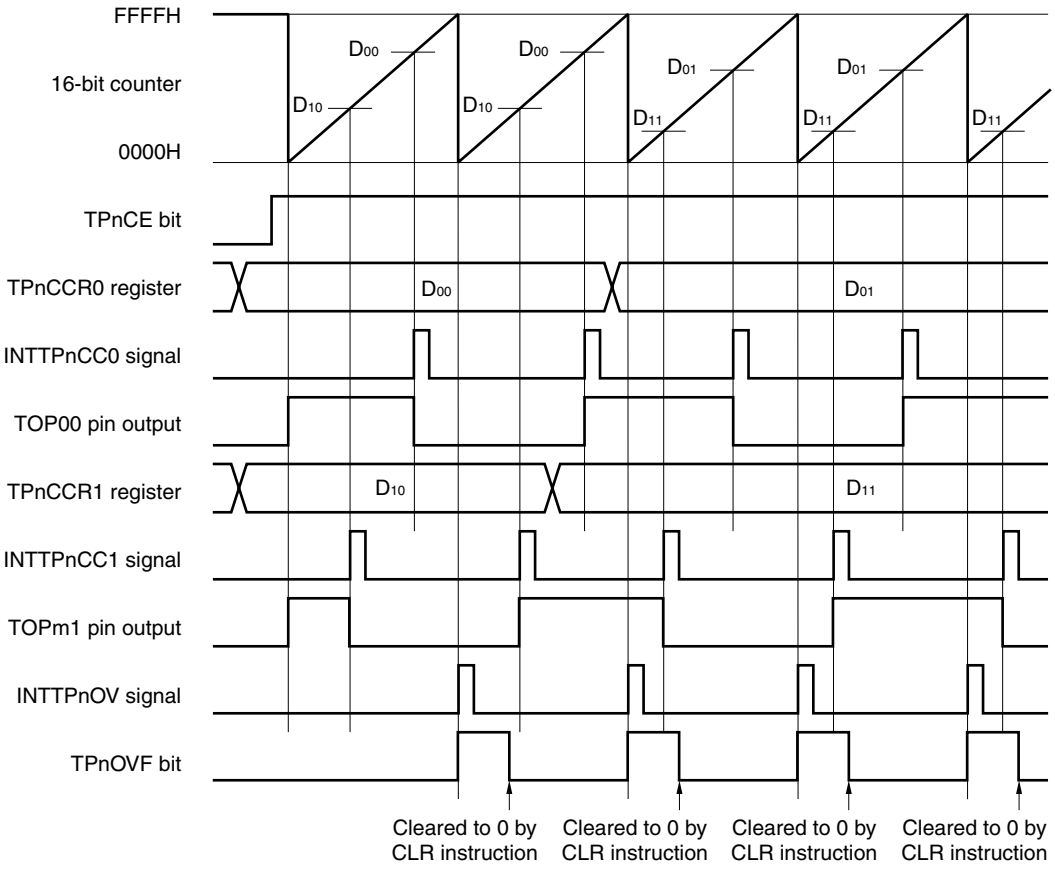
- Compare operation

  When the TPnCE bit is set to 1, 16-bit timer/event counter P starts counting, and the output signals of the TOP00 and TOPm1 pins are inverted. When the count value of the 16-bit counter later matches the set value of the TPnCCRa register, a compare match interrupt request signal (INTTPnCCa) is generated, and the output signals of the TOP00 and TOPm1 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 (INTTPnOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TPnOPT0.TPnOVF 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 TPnCCRa 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 6-36. Basic Timing in Free-Running Timer Mode (Compare Function)



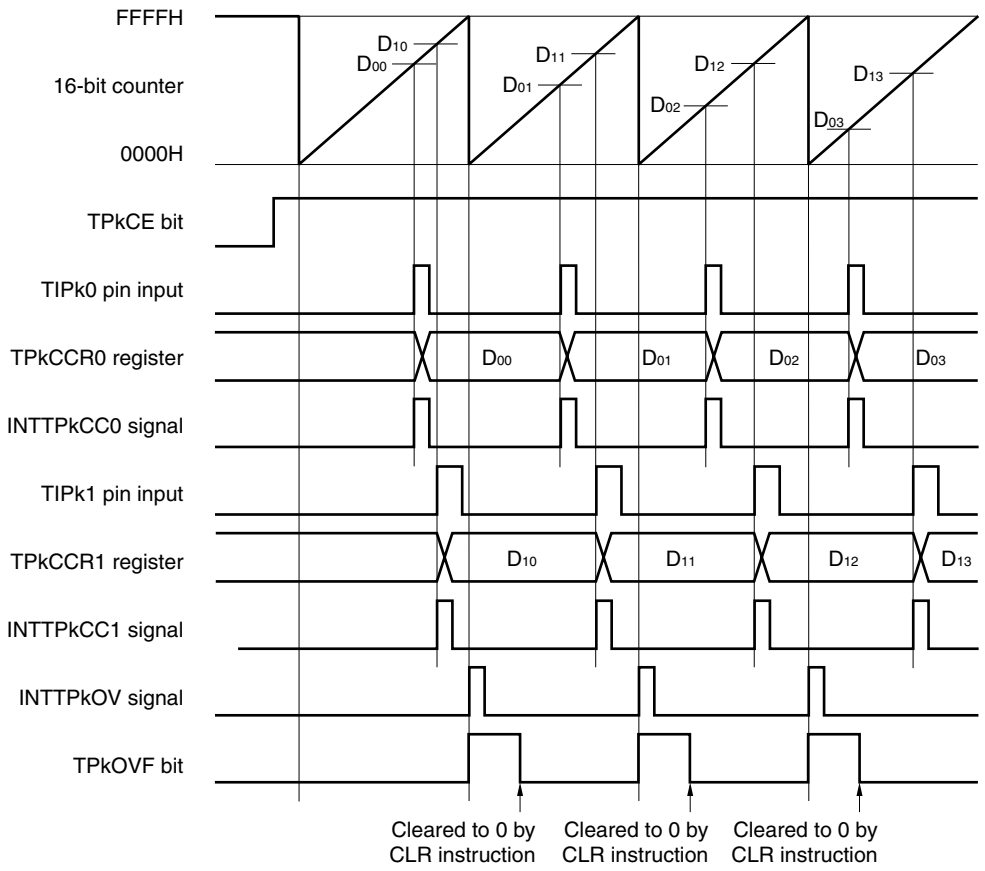
Remark n = 0 to 3
m = 0, 2, 3
a = 0, 1

- Capture operation

  When the TPkCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIPka pin is detected, the count value of the 16-bit counter is stored in the TPkCCRa register, and a capture interrupt request signal (INTTPkCCa) 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 (INTTPkOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TPkOPT0.TPkOVF 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.

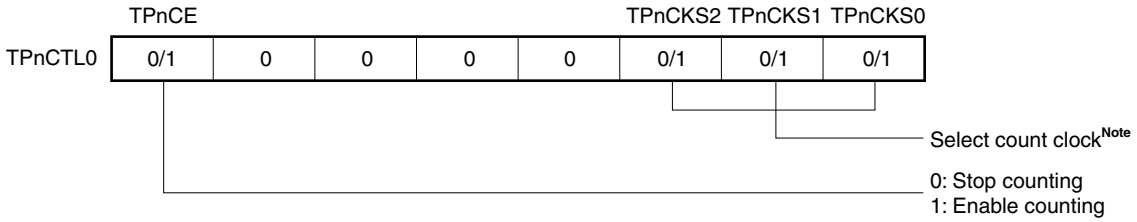
Figure 6-37. Basic Timing in Free-Running Timer Mode (Capture Function)



Remark k = 0, 2

Figure 6-38. Register Setting in Free-Running Timer Mode (1/2)

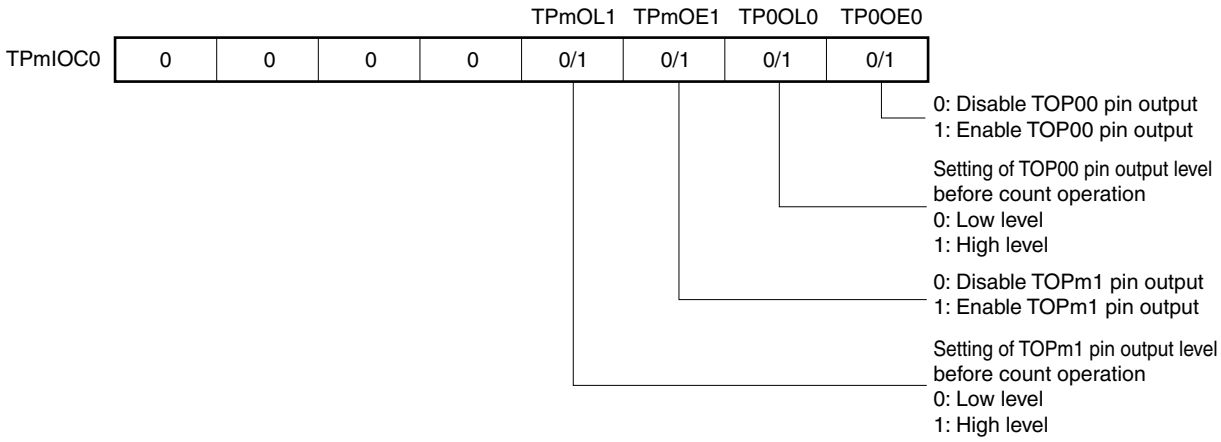
(a) TMPn control register 0 (TPnCTL0)



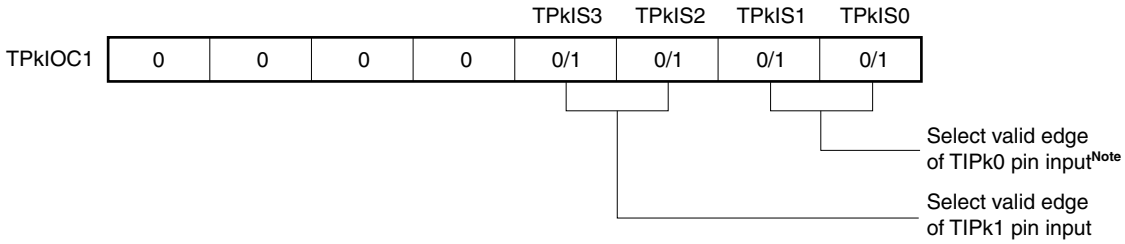
Note The setting is invalid when the TPkCTL1.TPkEEE bit = 1.

(b) TMPn control register 1 (TPnCTL1)

(c) TMPm I/O control register 0 (TPmIOC0)



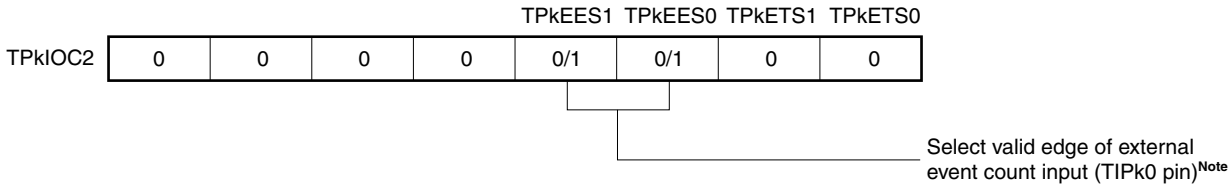
(d) TMPk I/O control register 1 (TPkIOC1)



Note Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

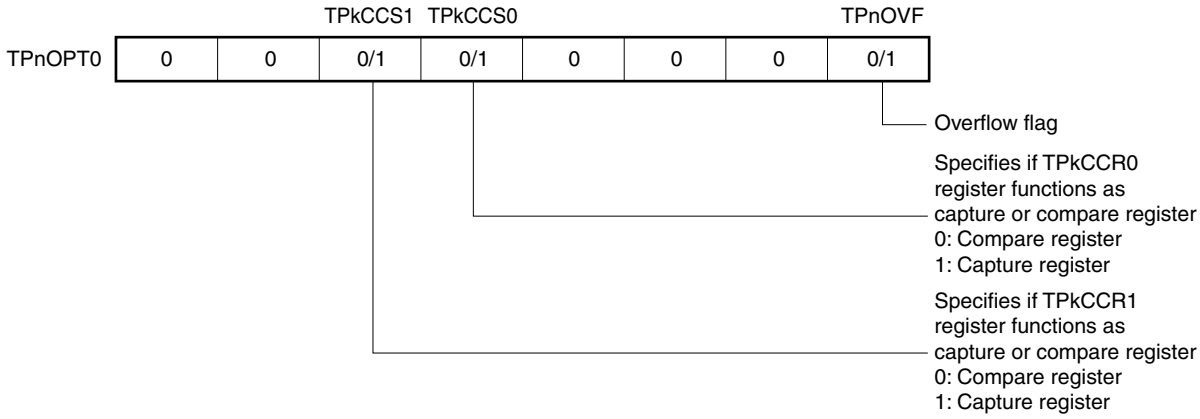
Figure 6-38. Register Setting in Free-Running Timer Mode (2/2)

(e) TMPk I/O control register 2 (TPkIOC2)



Note Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

(f) TMPn option register 0 (TPnOPT0)



(g) TMPn counter read buffer register (TPnCNT)

The value of the 16-bit counter can be read by reading the TPnCNT register.

(h) TMPn capture/compare registers 0 and 1 (TPnCCR0 and TPnCCR1)

These registers function as capture registers or compare registers depending on the setting of the TPkOPT0.TPkCCSa 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 TIPka pin is detected.

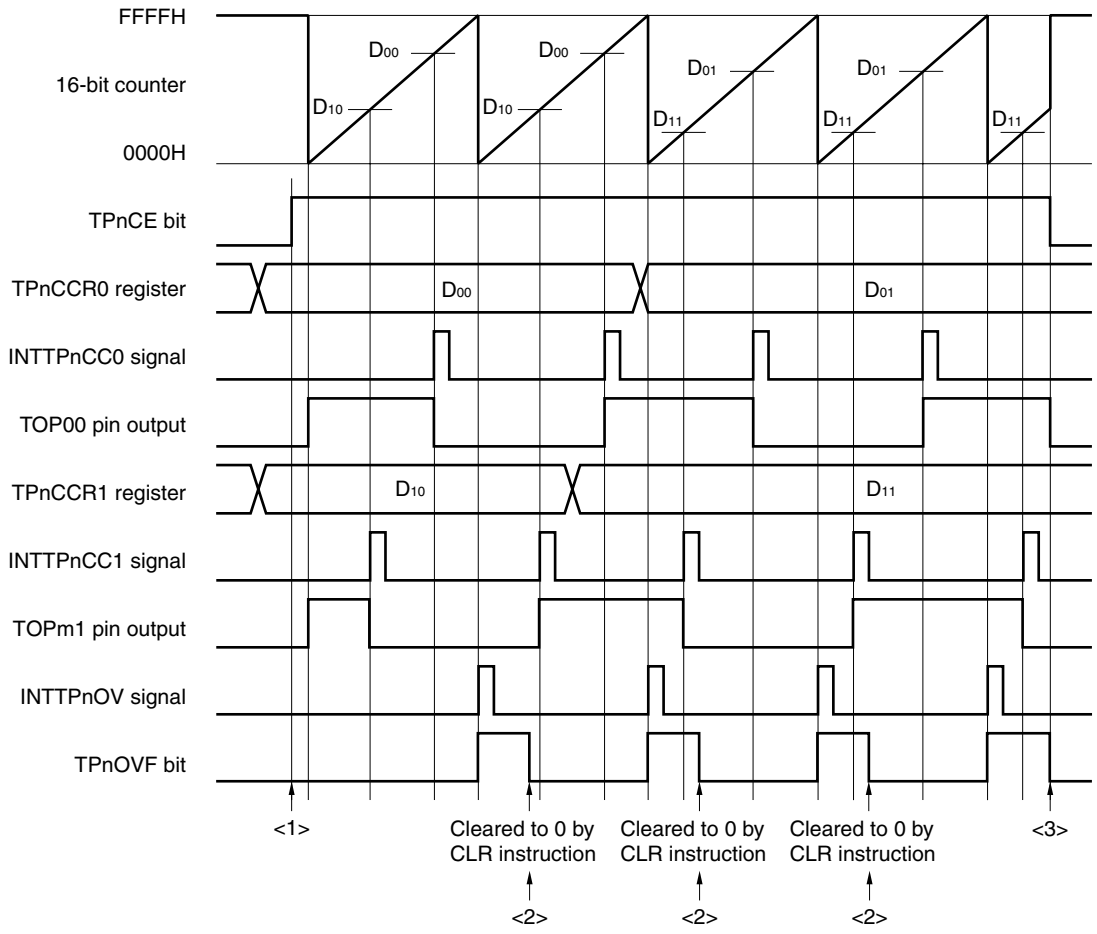
When the registers function as compare registers and when D_a is set to the TPnCCRa register, the INTTPnCCa signal is generated when the counter reaches (D_a + 1), and the output signals of the TOP00 and TOPm1 pins are inverted.

Remark n = 0 to 3, m = 0, 2, 3, k = 0, 2, a = 0, 1

(1) Operation flow in free-running timer mode

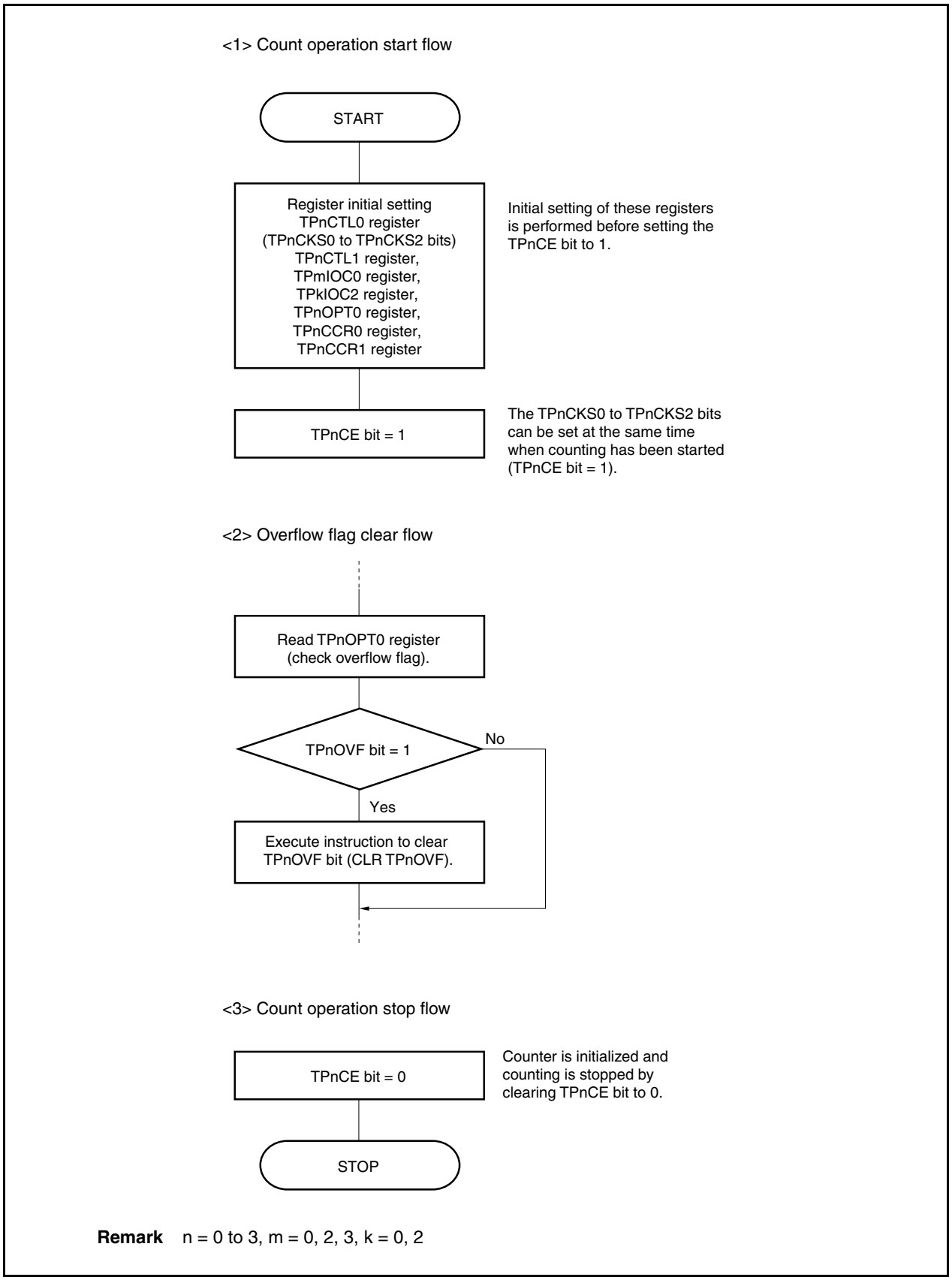
(a) When using capture/compare register as compare register

Figure 6-39. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)



Remark n = 0 to 3
m = 0, 2, 3

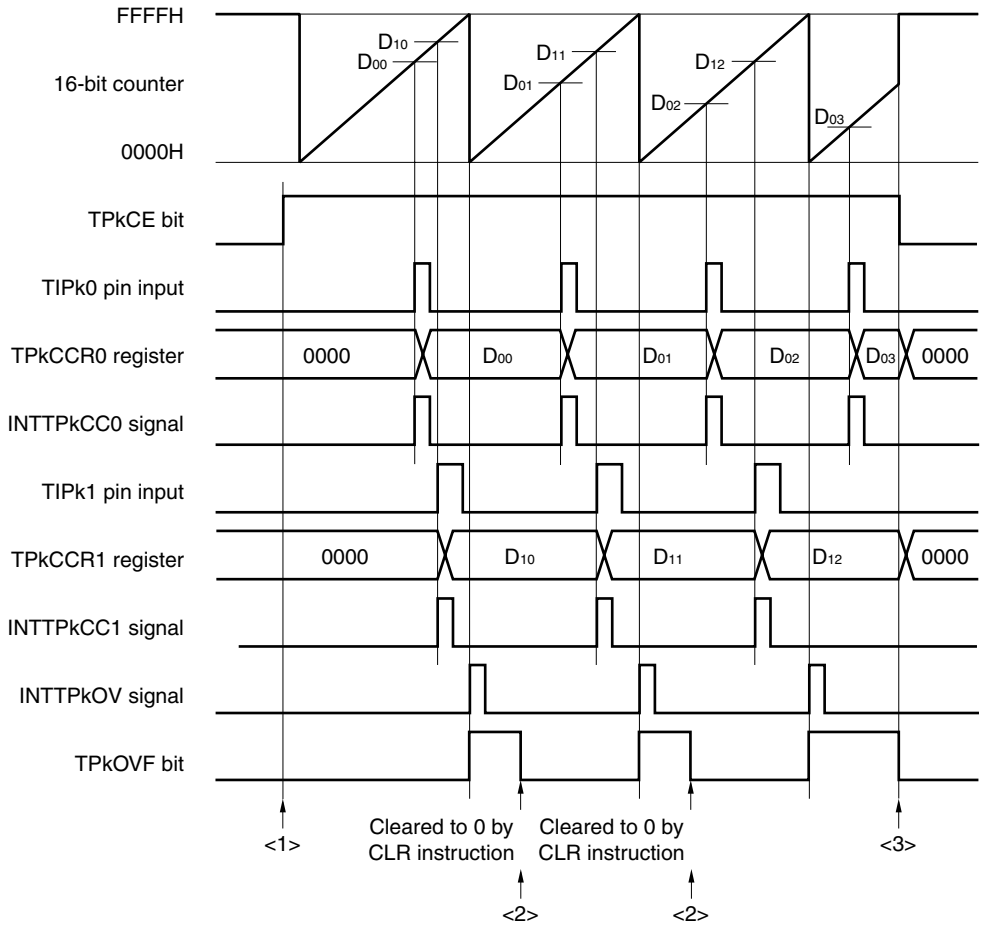
Figure 6-39. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)



Remark n = 0 to 3, m = 0, 2, 3, k = 0, 2

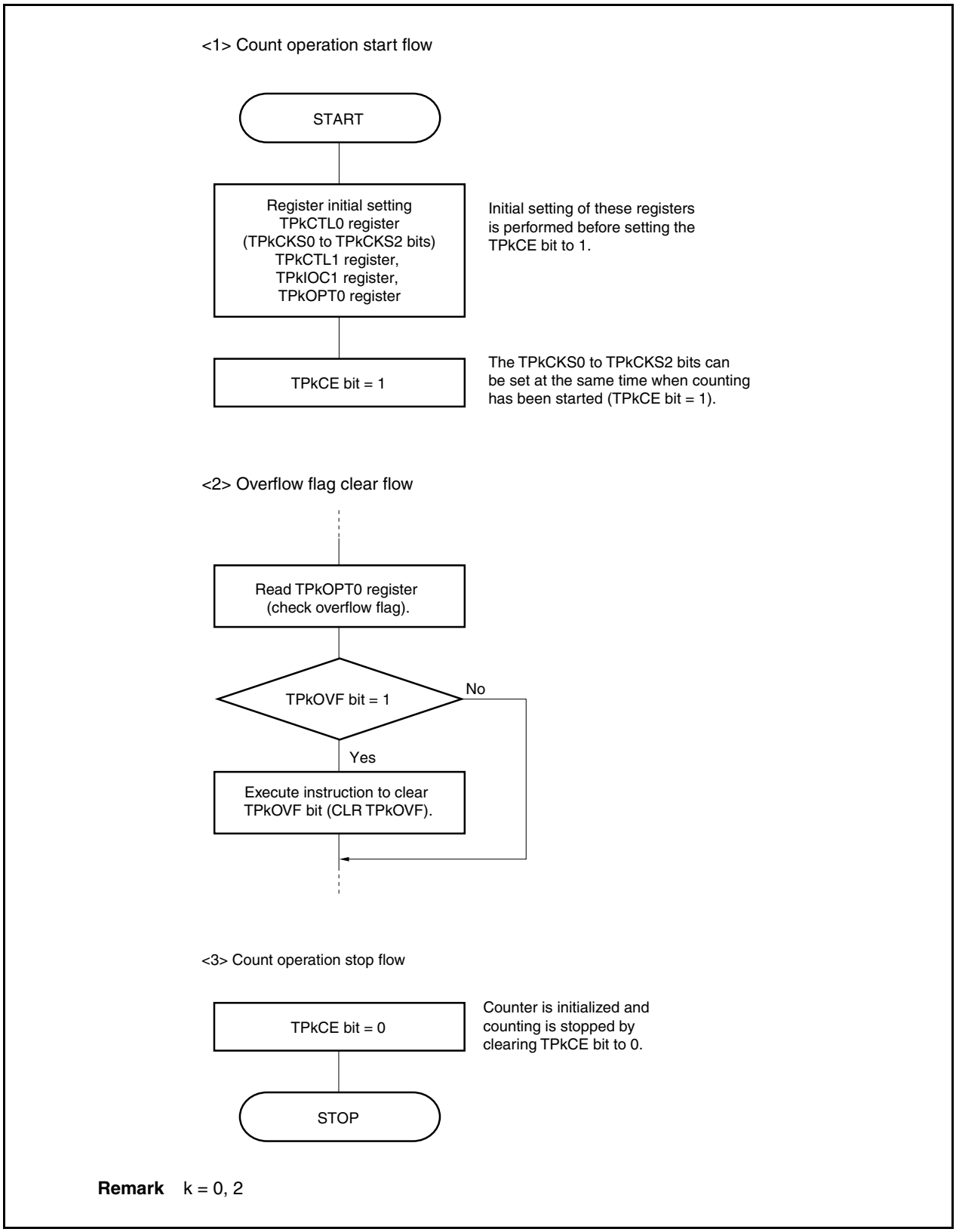
(b) When using capture/compare register as capture register

Figure 6-40. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)



Remark k = 0, 2

Figure 6-40. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)

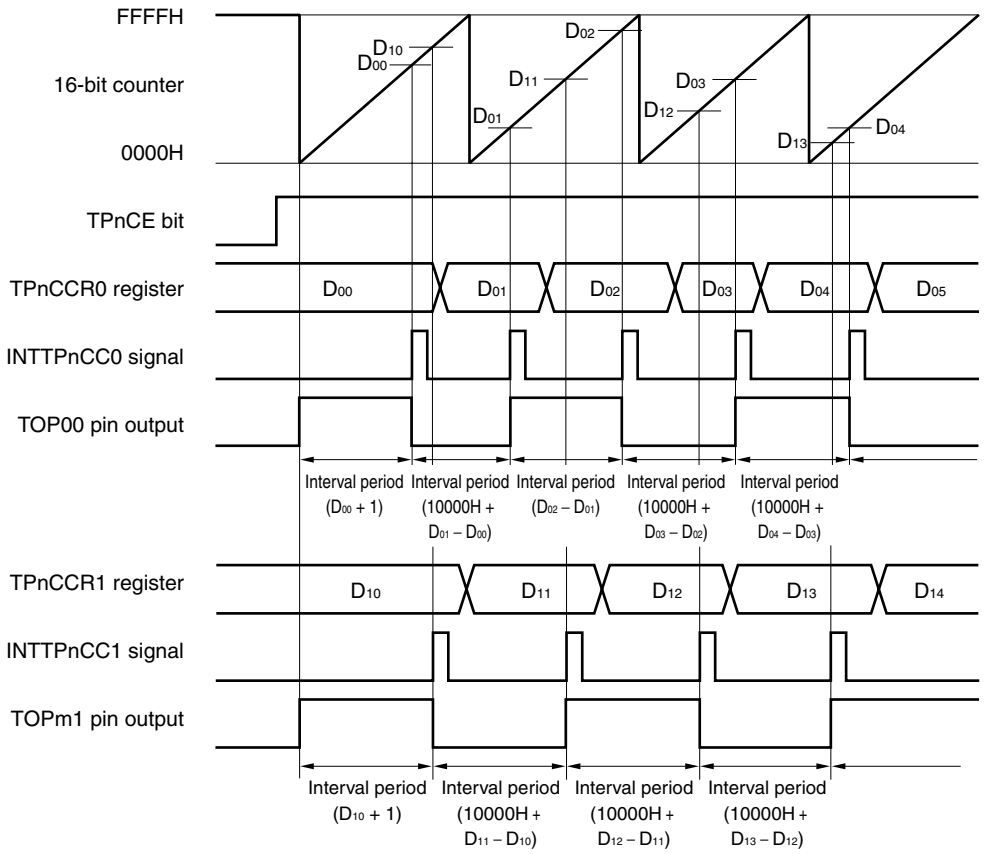


Remark k = 0, 2

(2) Operation timing in free-running timer mode

(a) Interval operation with compare register

When 16-bit timer/event counter P is used as an interval timer with the TPnCCRa 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 INTTPnCCa 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 TPnCCRa register must be re-set in the interrupt servicing that is executed when the INTTPnCCa signal is detected.

The set value for re-setting the TPnCCRa register can be calculated by the following expression, where "D_a" is the interval period.

Compare register default value: $D_a - 1$

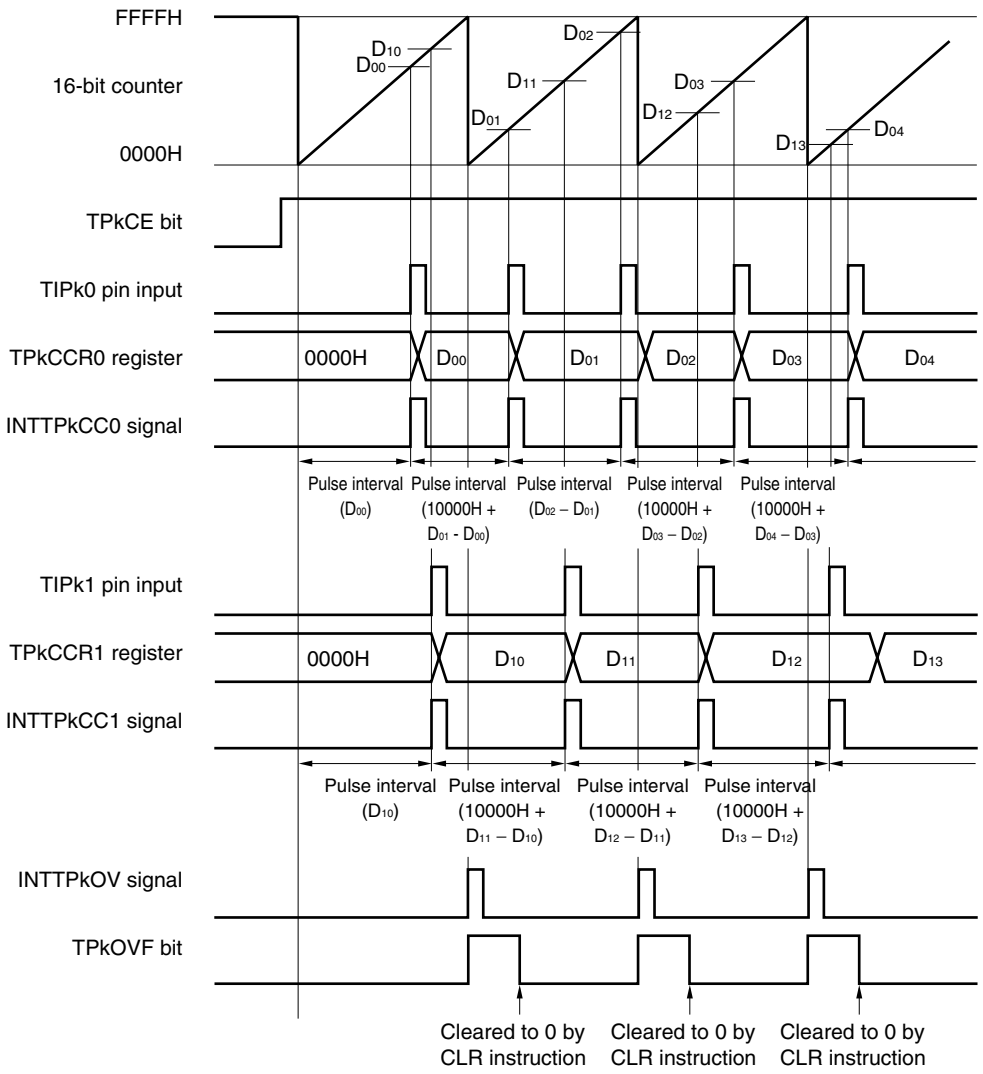
Value set to compare register second and subsequent time: Previous set value + D_a

(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

Remark n = 0 to 3
m = 0, 2, 3
a = 0, 1

(b) Pulse width measurement with capture register

When pulse width measurement is performed with the TPkCCRa register used as a capture register, software processing is necessary for reading the capture register each time the INTTPkCCa 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.

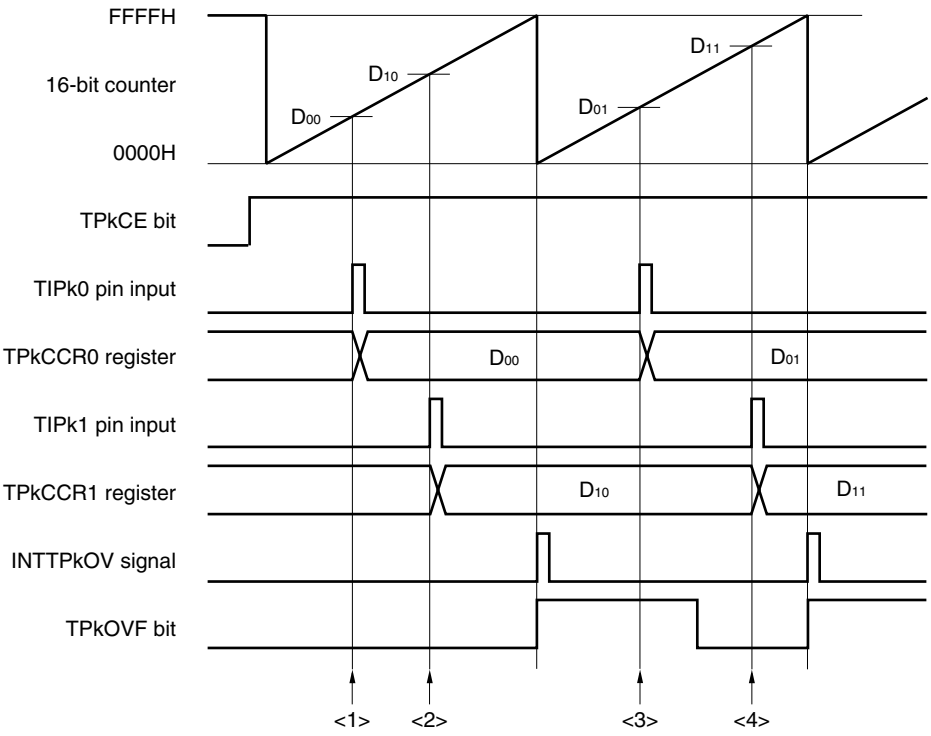
To measure a pulse width, the pulse width can be calculated by reading the value of the TPkCCRa register in synchronization with the INTTPkCCa signal, and calculating the difference between the read value and the previously read value.

Remark k = 0, 2
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.

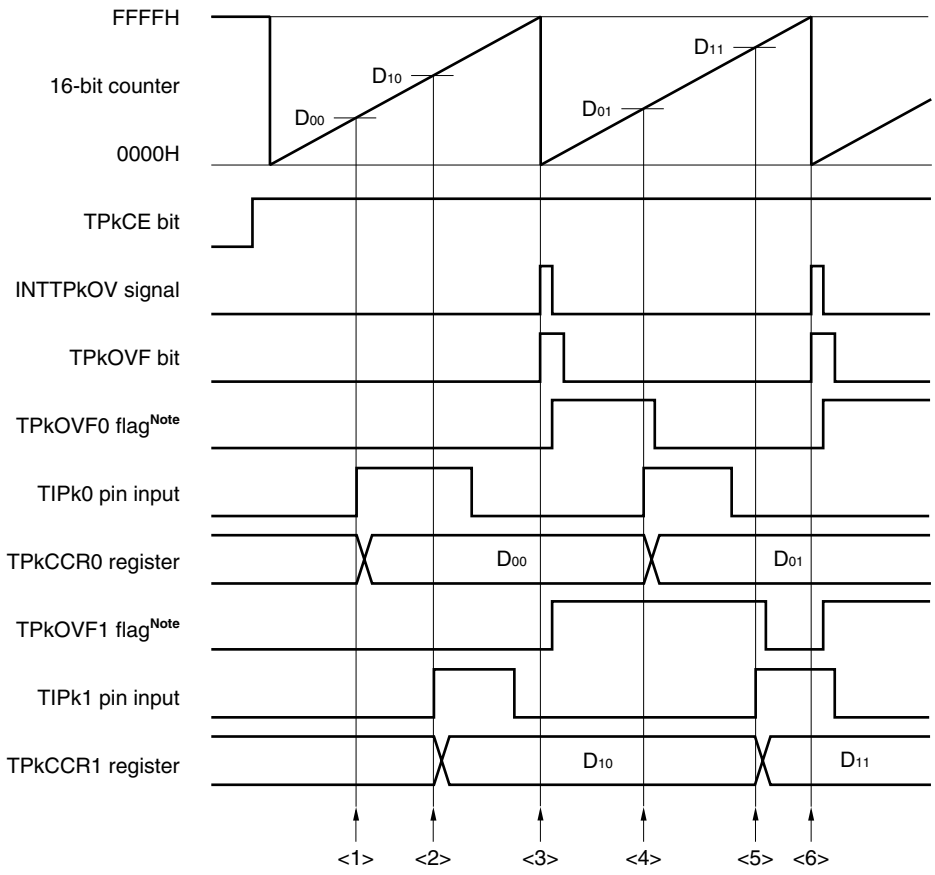
<1> Read the TPkCCR0 register (setting of the default value of the TIPk0 pin input).
<2> Read the TPkCCR1 register (setting of the default value of the TIPk1 pin input).
<3> Read the TPkCCR0 register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} − D_{00}).
<4> Read the TPkCCR1 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 k = 0, 2

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)



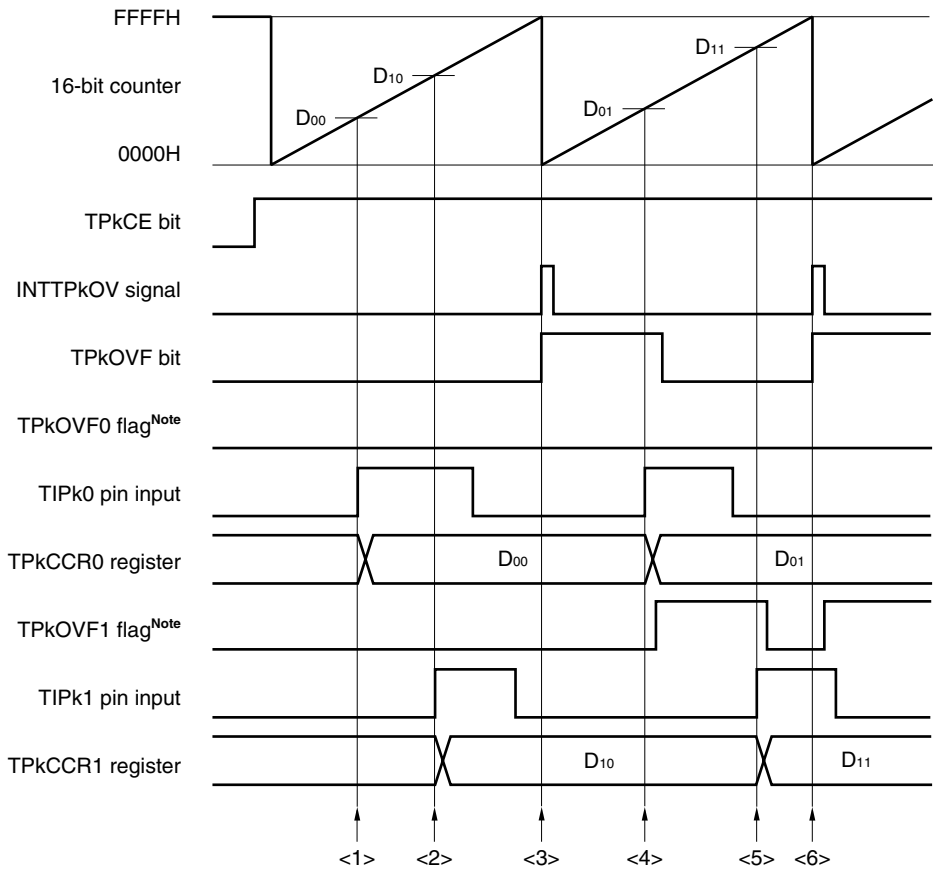
Note The TPkOVF0 and TPkOVF1 flags are set on the internal RAM by software.

<1> Read the TPkCCR0 register (setting of the default value of the TIk0 pin input).
<2> Read the TPkCCR1 register (setting of the default value of the TIPk1 pin input).
<3> An overflow occurs. Set the TPkOVF0 and TPkOVF1 flags to 1 in the overflow interrupt servicing, and clear the overflow flag to 0.
<4> Read the TPkCCR0 register.
Read the TPkOVF0 flag. If the TPkOVF0 flag is 1, clear it to 0.
Because the TPkOVF0 flag is 1, the pulse width can be calculated by (10000H + $D_{01} - D_{00}$).
<5> Read the TPkCCR1 register.
Read the TPkOVF1 flag. If the TPkOVF1 flag is 1, clear it to 0 (the TPkOVF0 flag is cleared in <4>, and the TPkOVF1 flag remains 1).
Because the TPkOVF1 flag is 1, the pulse width can be calculated by (10000H + $D_{11} - D_{10}$) (correct).
<6> Same as <3>

Remark k = 0, 2

(2/2)

Example when two capture registers are used (without using overflow interrupt)



Note The TPkOVF0 and TPkOVF1 flags are set on the internal RAM by software.

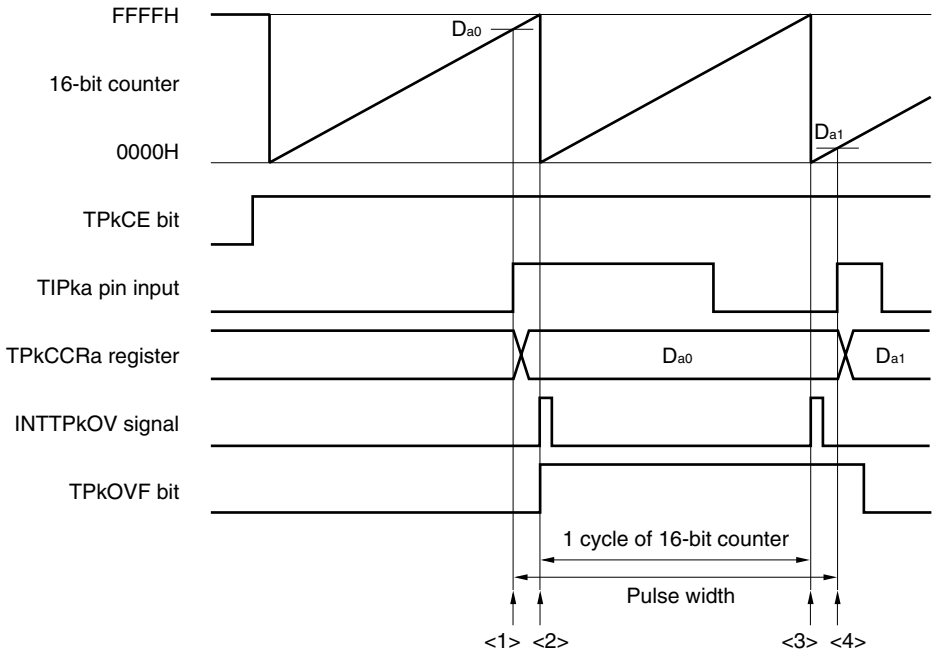
<1> Read the TPkCCR0 register (setting of the default value of the TIPk0 pin input).
<2> Read the TPkCCR1 register (setting of the default value of the TIPk1 pin input).
<3> An overflow occurs. Nothing is done by software.
<4> Read the TPkCCR0 register.
Read the overflow flag. If the overflow flag is 1, set only the TPkOVF1 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 TPkCCR1 register.
Read the overflow flag. Because the overflow flag is cleared in <4>, 0 is read.
Read the TPkOVF1 flag. If the TPkOVF1 flag is 1, clear it to 0.
Because the TPkOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} − D_{10}) (correct).
<6> Same as <3>

Remark k = 0, 2

(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



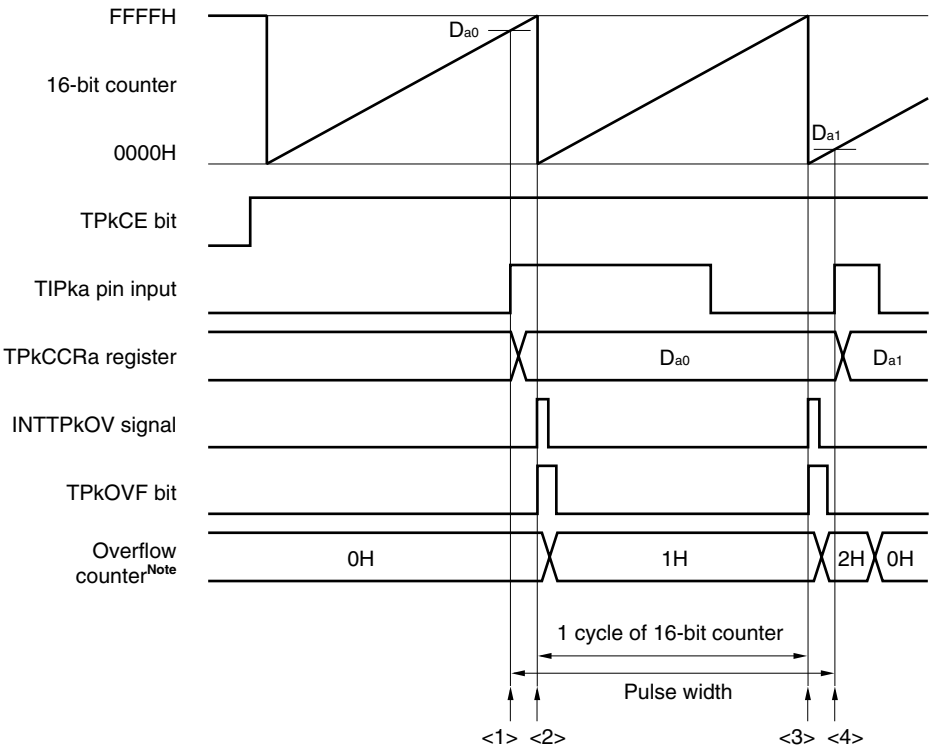
The following problem may occur when long pulse width is measured in the free-running timer mode.

<1> Read the TPkCCRa register (setting of the default value of the TIPka 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 TPkCCRa 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 k = 0, 2
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 TPkCCRa register (setting of the default value of the TIPka 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 TPkCCRa 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 k = 0, 2

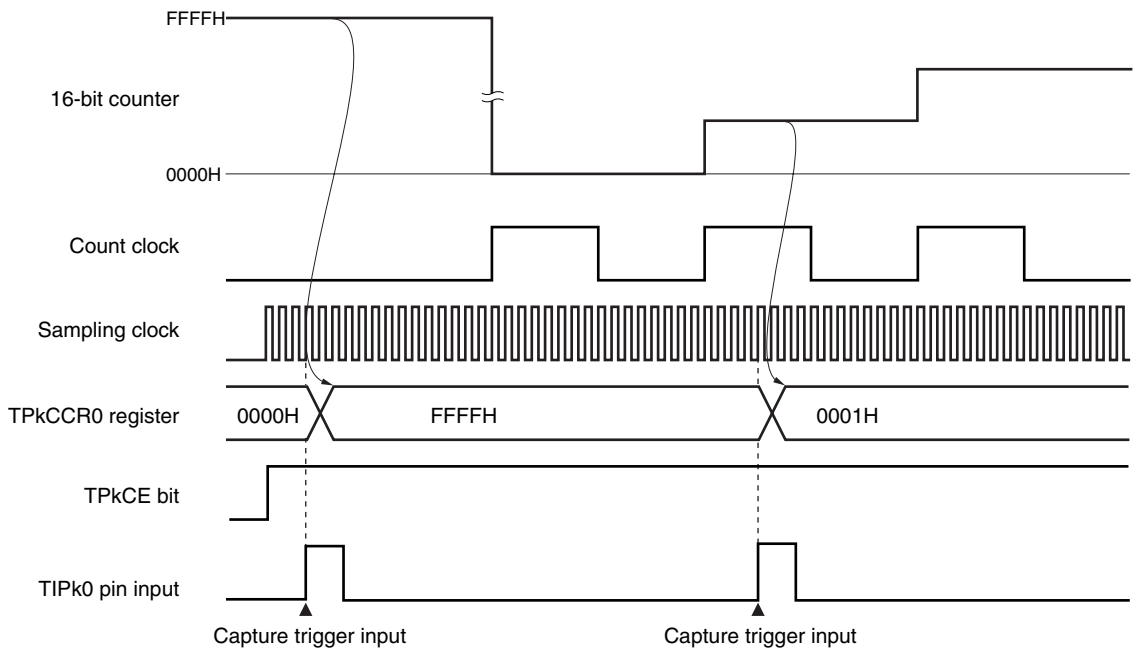
a = 0, 1

(e) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TPnOVF bit to 0 with the CLR instruction after reading the TPnOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TPnOPT0 register after reading the TPnOVF bit when it is 1.

(3) Note on capture operation

If the capture operation is used and if a slow clock is selected as the count clock, FFFFH, not 0000H, may be captured to the TPkCCRa register if the capture trigger is input immediately after the TPkCTL0.TPkCE bit is set to 1.



Remark k = 0, 2
a = 0, 1

6.6.7 Pulse width measurement mode (TPkMD2 to TPkMD0 bits = 110)

The mode is valid only in TMP0 and TMP2.

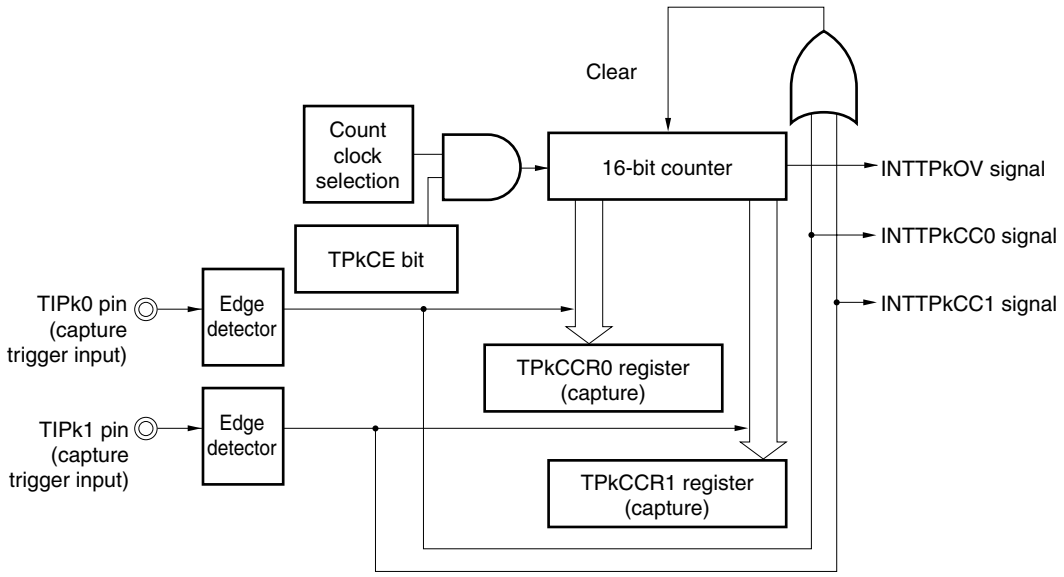
In the pulse width measurement mode, 16-bit timer/event counter P starts counting when the TPkCTL0.TPkCE bit is set to 1. Each time the valid edge input to the TIPka pin has been detected, the count value of the 16-bit counter is stored in the TPkCCRa register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TPkCCRa register after a capture interrupt request signal (INTTPkCCa) occurs.

As shown in Figure 6-42, select either the TIPk0 or TIPk1 pin as the capture trigger input pin and set the unused pins to “No edge detection” by using the TPkIOC1 register.

<R>

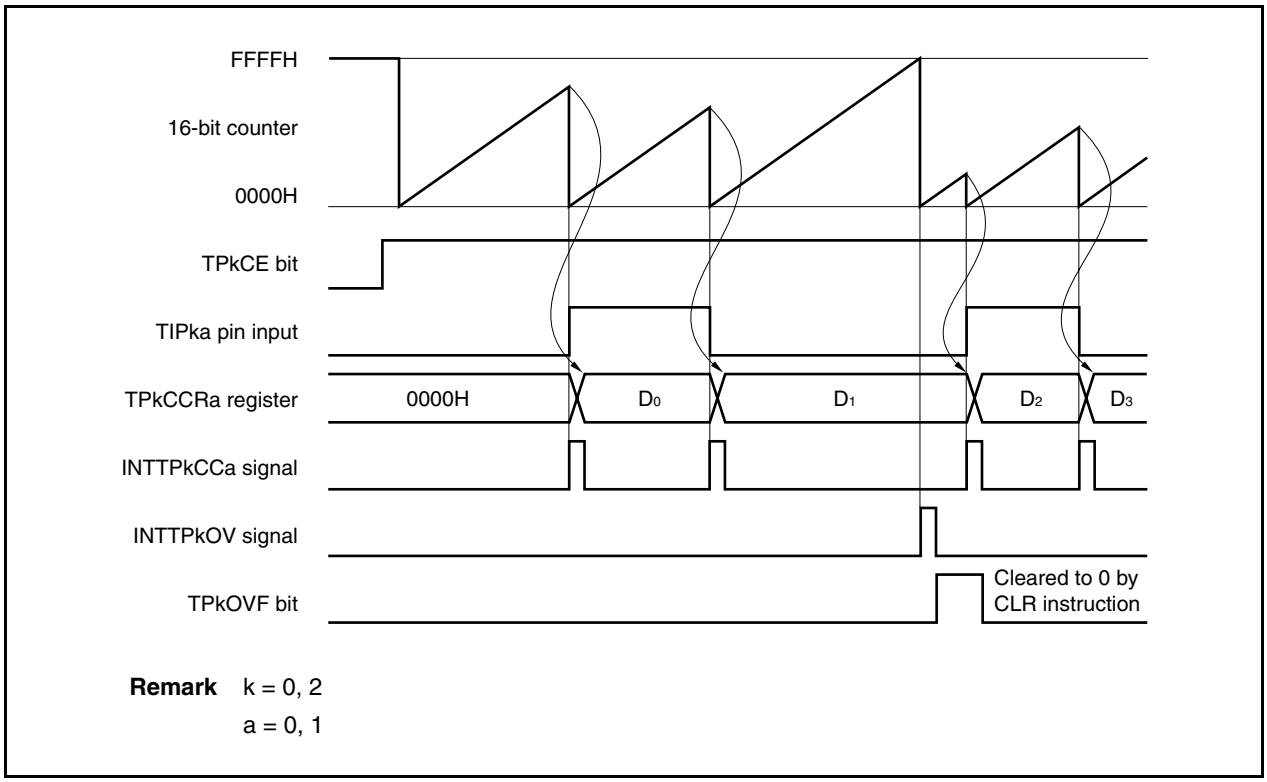
Figure 6-41. Configuration in Pulse Width Measurement Mode



Caution In the pulse width measurement mode, select the internal clock as the count clock (by clearing the TPkCTL1.TPkEEE bit to 0).

Remark k = 0, 2
a = 0, 1

Figure 6-42. Basic Timing in Pulse Width Measurement Mode



Remark k = 0, 2
a = 0, 1

When the TPkCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIPka pin is later detected, the count value of the 16-bit counter is stored in the TPkCCRa register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTTPkCCa) is generated.

The pulse width is calculated as follows.

<R> Pulse width = Captured value × Count clock cycle

If the valid edge is not input to the TIPnm pin even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTPkOV) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TPkOPT0.TPkOVF 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.

<R> Pulse width = (10000H × Number of times for which TPkOVF bit is set to 1 + Captured value) × Count clock cycle

Remark k = 0, 2
a = 0, 1

Figure 6-43. Register Setting in Pulse Width Measurement Mode (1/2)

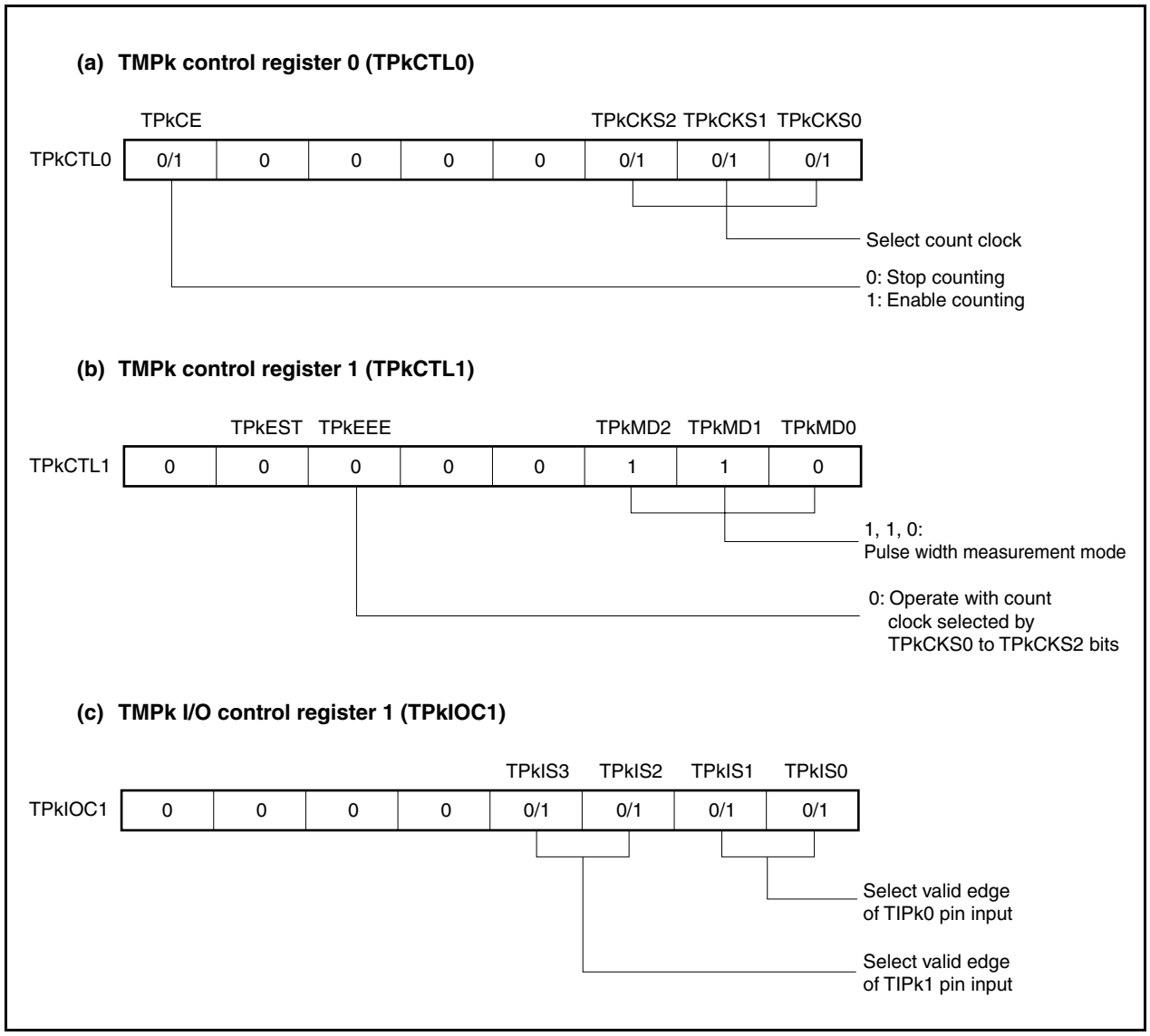


Figure 6-43. Register Setting in Pulse Width Measurement Mode (2/2)

(d) TMPk option register 0 (TPkOPT0)

	TPkCCS1	TPkCCS0					TPkOVF
TPkOPT0	0	0	0	0	0	0	0/1

Overflow flag

(e) TMPk counter read buffer register (TPkCNT)

The value of the 16-bit counter can be read by reading the TPkCNT register.

(f) TMPk capture/compare registers 0 and 1 (TPkCCR0 and TPkCCR1)

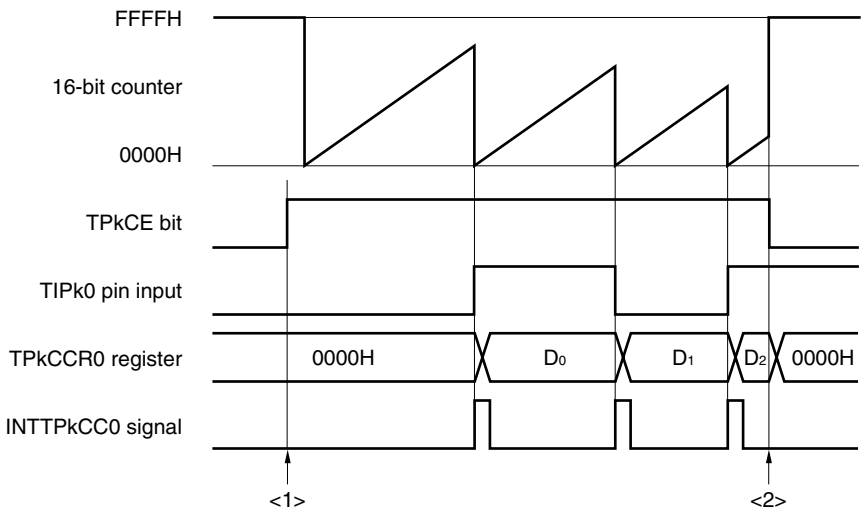
These registers store the count value of the 16-bit counter when the valid edge input to the TIPk0 and TIPk1 pins is detected.

<R> **Remarks 1.** TMPk I/O control register 0 (TPkIOC0) and TMPk I/O control register 2 (TPkIOC2) are not used in the pulse width measurement mode.

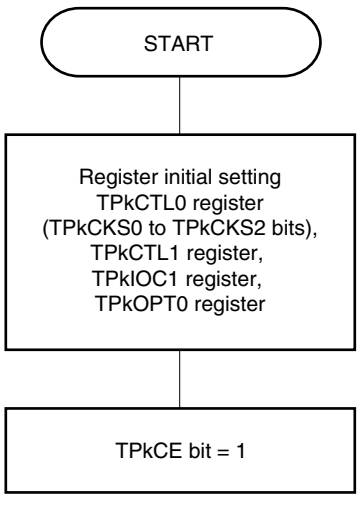
2. k = 0, 2

(1) Operation flow in pulse width measurement mode

Figure 6-44. Software Processing Flow in Pulse Width Measurement Mode



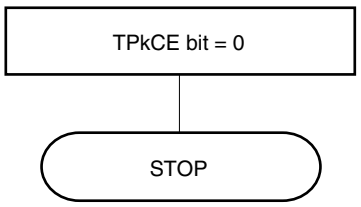
<R> <1> Count operation start flow



Initial setting of these registers is performed before setting the TPkCE bit to 1.

The TPkCKS0 to TPkCKS2 bits can be set at the same time when counting has been started (TPkCE bit = 1).

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TPkCE bit to 0.

Remark k = 0, 2

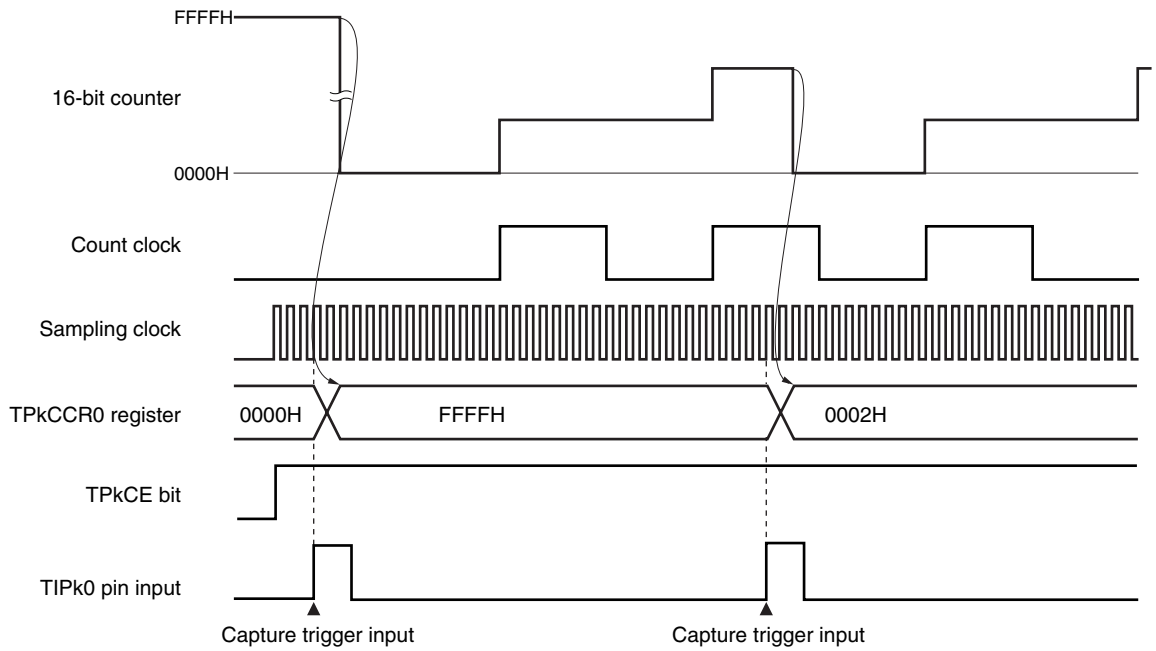
(2) Operation timing in pulse width measurement mode

(a) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TPkOVF bit to 0 with the CLR instruction after reading the TPkOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TPkOPT0 register after reading the TPkOVF bit when it is 1.

(3) Note

If a slow clock is selected as the count clock, FFFFH, not 0000H, may be captured to the TPkCCRa register if the capture trigger is input immediately after the TPkCTL0.TPkCE bit has been set to 1.



Remark k = 0, 2
a = 0, 1

CHAPTER 7 16-BIT TIMER/EVENT COUNTER Q (TMQ)

Timer Q (TMQ) is a 16-bit timer/event counter.
The V850ES/IE2 incorporates TMQ1 and TMQ0.

7.1 Overview

The TMQn channels are outlined below (n = 0, 1).

Caution If P16 is used as the TOQ00 output pin or an output port, when an error (oscillator stop) is detected by the clock monitor, the CLMER signal (low level) is output from P16. Low-level output is released by reset signal. For details, see Table 4-5 Alternate-Function Pins of Port 1.

Table 7-1. TMQn Overview

Item	TMQ0	TMQ1
Clock selection	8 ways	8 ways
Capture trigger input pin	4	None
External event count input pin	1	None
External trigger input pin	None	None
Timer counter	1	1
Capture/compare register	4	4 Note 1
Capture/compare match interrupt request signal	4	4 Note 1
<R> Overflow interrupt request signal	1	1
Timer output pin Note 2	4	1

Notes 1. Compare function only

2. This is the number of output pins of TMQn; it does not include the output pins of TMQOP1. For details of the output pins of TMQOP1, see **CHAPTER 9 MOTOR CONTROL FUNCTION**.

7.2 Functions

The TMQn functions that can be realized differ from one channel to another, as shown in the table below (n = 0, 1).

Table 7-2. TMQn Functions

Function	TMQ0	TMQ1
6-phase PWM output[Note 1]	×	√
Interval timer	√	√
External event counter	√	×
External trigger pulse output	√[Note 2]	×
One-shot pulse output	√[Note 2]	×
PWM output	√	×
Free-running timer	√	√
Pulse width measurement	√	×

Notes 1. This is connected to TMQOP1. For details, see **CHAPTER 9 MOTOR CONTROL FUNCTION**.

2. Can be realized only by software trigger. No external trigger input pins are provided.

7.3 Configuration

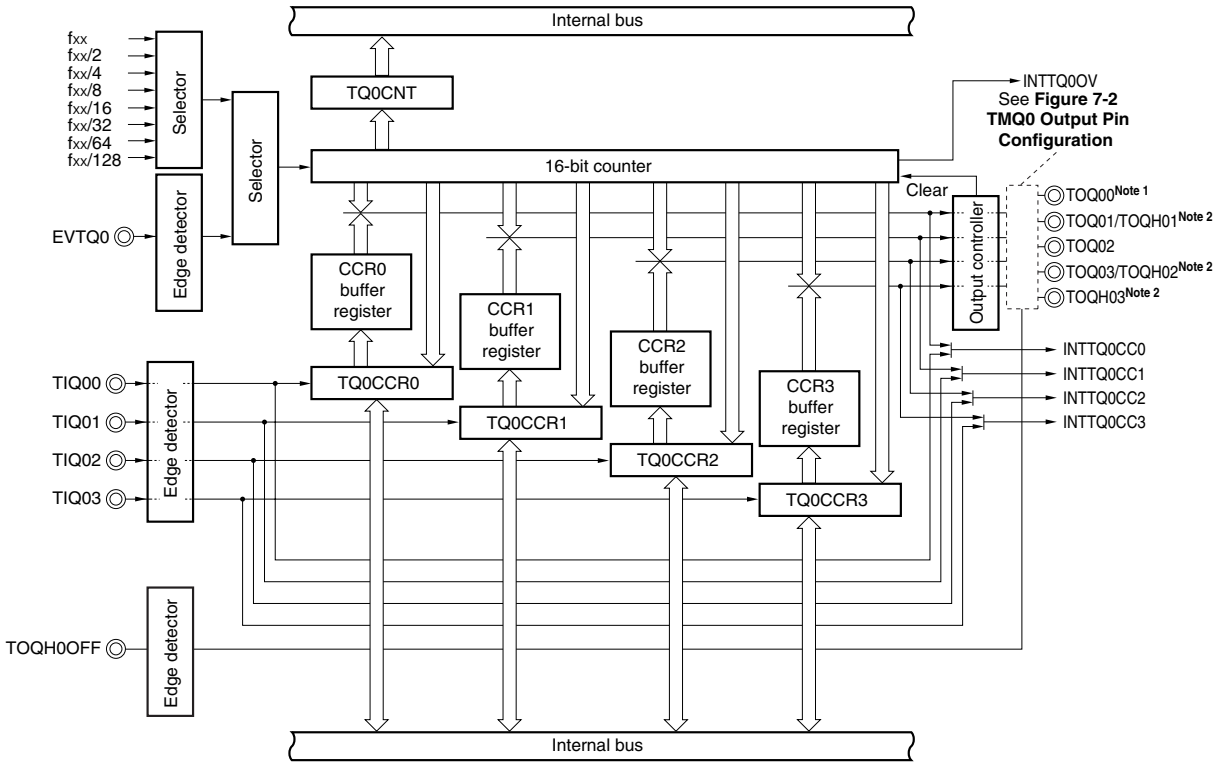
TMQn includes the following hardware.

Table 7-3. TMQn Configuration

Item	Configuration
Timer register	16-bit counter × 1
Registers	TMQn counter read buffer register (TQnCNT): Total of 2 TMQn capture/compare registers 0 to 3 (TQnCCR0 to TQnCCR3): Total of 8 CCR0 to CCR3 buffer registers: Total of 8
Timer input	Total of 5 (TIQ00 to TIQ03, EVTQ0 pins)
Timer output	Total of 6 (TOQ00, TOQ01/TOQH01, TOQ02, TOQ03/TOQH02, TOQH03, TOQ10 pins)
Control registers	TMQn control registers 0, 1 (TQnCTL0, TQnCTL1) TMQn I/O control register 0 (TQnIOC0) TMQ0 I/O control registers 1, 2 (TQ0IOC1, TQ0IOC2) TMQn option register 0 (TQnOPT0)

Remark n = 0, 1

Figure 7-1. TMQ0 Block Diagram



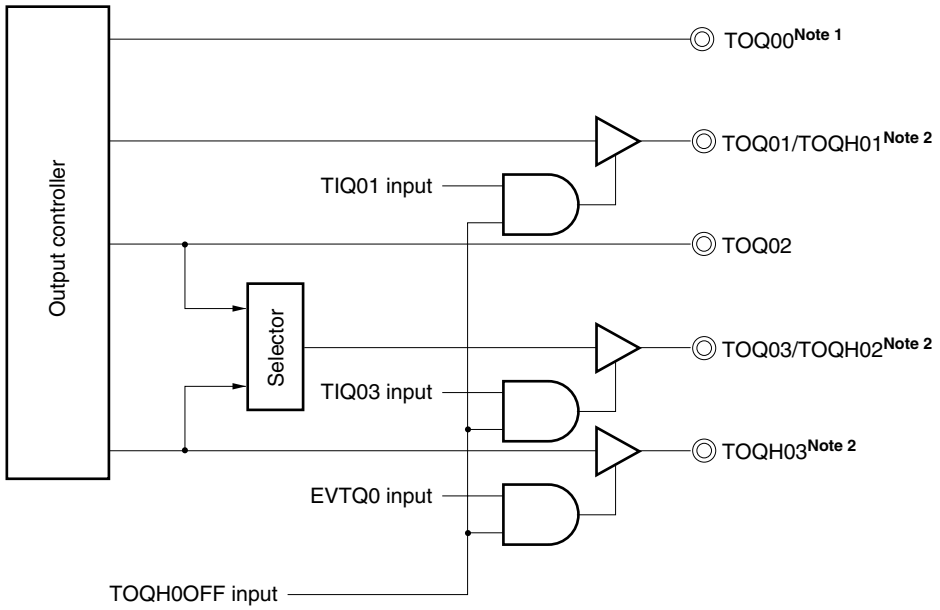
Notes 1. If P16 is used as the TOQ00 output pin or an output port, when an error (oscillator stop) is detected by the clock monitor, the CLMER signal (low level) is output from P16. Low-level output is released by reset signal. For details, see **Table 4-5 Alternate-Function Pins of Port 1**.

2. When the TOQH01 to TOQH03 pins are selected, these pins can be set in the high-impedance state by the TOQH0OFF pin input. For details, see **9.3 (6) High-impedance output control registers 00, 01, 10, 11 (HZAyCTLn)**.

Caution The TOQH01 to TOQH03 pins are valid only in the PWM output mode.

Remark fxx: Peripheral clock

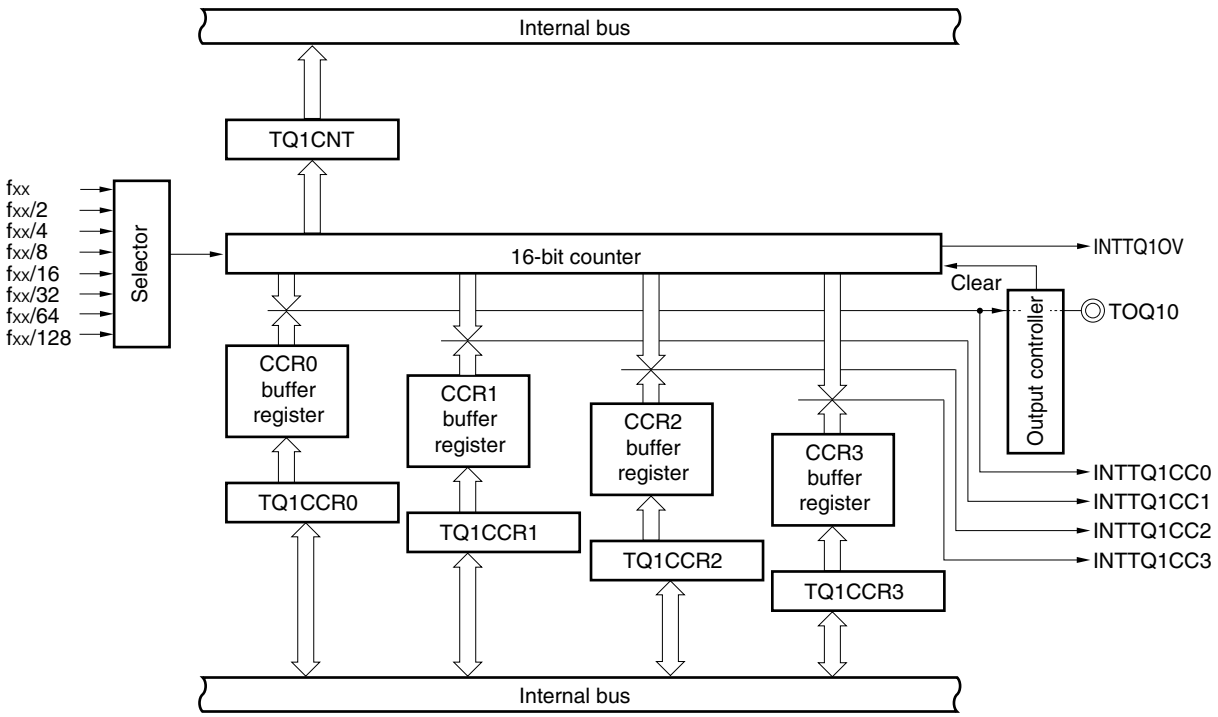
Figure 7-2. TMQ0 Output Pin Configuration



Notes 1. If P16 is used as the TOQ00 output pin or an output port, when an error (oscillator stop) is detected by the clock monitor, the CLMER signal (low level) is output from P16. Low-level output is released by reset signal. For details, see **Table 4-5 Alternate-Function Pins of Port 1**.

2. When the TOQH01 to TOQH03 pins are selected, these pins can be set in the high-impedance state by the TOQH0OFF pin input.

Figure 7-3. TMQ1 Block Diagram



Remark fxx: Peripheral clock

(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 TQnCNT register.

When the TQnCTL0.TQnCE bit = 0, the value of the 16-bit counter is FFFFH. If the TQnCNT register is read at this time, 0000H is read.

The TQnCE bit is cleared to 0 after reset.

(2) CCR0 buffer register

This is a 16-bit compare register that compares the count value of the 16-bit counter.

When the TQnCCR0 register is used as a compare register, the value written to the TQnCCR0 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 (INTTQnCC0) is generated.

The CCR0 buffer register cannot be read or written directly.

The TQnCCR0 register is cleared to 0000H after reset, and the CCR0 buffer 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 TQnCCR1 register is used as a compare register, the value written to the TQnCCR1 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 (INTTQnCC1) is generated.

The CCR1 buffer register cannot be read or written directly.

The TQnCCR1 register is cleared to 0000H after reset, and the CCR1 buffer 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 TQnCCR2 register is used as a compare register, the value written to the TQnCCR2 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 (INTTQnCC2) is generated.

The CCR2 buffer register cannot be read or written directly.

The TQnCCR2 register is cleared to 0000H after reset, and the CCR2 buffer 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 TQnCCR3 register is used as a compare register, the value written to the TQnCCR3 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 (INTTQnCC3) is generated.

The CCR3 buffer register cannot be read or written directly.

The TQnCCR3 register is cleared to 0000H after reset, and the CCR3 buffer register is cleared to 0000H.

(6) Edge detector

This circuit detects the valid edges input to the TIQ00 to TIQ03 and EVTQ0 pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TQ0IOC1 and TQ0IOC2 registers.

(7) Output controller

This circuit controls the output of the TOQ00, TOQ01/TOQH01, TOQ02, TOQ03/TOQH02, TOQH03, and TOQ10 pins. The output of the TOQ00, TOQ01/TOQH01, TOQ02, TOQ03/TOQH02, and TOQH03 pins is controlled by the TQ0IOC0 register. The output of the TOQ10 pin is controlled by the TQ1IOC0 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) TMQn control register 0 (TQnCTL0)

The TQnCTL0 register is an 8-bit register that controls the TMQn 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 TQnCTL0 register by software.

After reset: 00H R/W Address: TQ0CTL0 FFFFF5C0H, TQ1CTL0 FFFFF600H

	<7>	6	5	4	3	2	1	0
TQnCTL0 (n = 0, 1)	TQnCE	0	0	0	0	TQnCKS2	TQnCKS1	TQnCKS0

TQnCE	TMQn operation control
0	TMQn operation disabled (TMQn reset asynchronously[Note])
1	TMQn operation enabled. Start TMQn operation

TQnCKS2	TQnCKS1	TQnCKS0	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 TQnOPT0.TQnOVF bit and 16-bit counter are reset simultaneously. Moreover, timer outputs (TOQ00, TOQ01/TOQH01, TOQ02, TOQ03/TOQH02, TOQH03, and TOQ10 pins) are reset to the TQnIOC0 register set status at the same time as the 16-bit counter.

Cautions 1. Set the TQnCKS2 to TQnCKS0 bits when the TQnCE bit = 0.
When the value of the TQnCE bit is changed from 0 to 1, the TQnCKS2 to TQnCKS0 bits can be set simultaneously.
2. Be sure to clear bits 3 to 6 to "0".

Remark f_{XX}: Peripheral clock

(2) TMQn control register 1 (TQnCTL1)

The TQnCTL1 register is an 8-bit register that controls the TMQn operation.

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: TQ0CTL1 FFFFF5C1H, TQ1CTL1 FFFFF601H

TQnCTL1 (n = 0, 1)

7	6	5	4	3	2	1	0
0	TQ0EST[Note 1]	TQ0EEE[Note 1]	0	0	TQnMD2	TQnMD1	TQnMD0

TQ0EST[Note 1]	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 TQ0EST bit as the trigger. • In external trigger pulse output mode: A PWM waveform is output with writing 1 to the TQ0EST bit as the trigger.
Read value of the TQ0EST bit is always 0.	

TQ0EEE[Note 1]	Count clock selection
0	Disable operation with external event count input (EVTQ0 pin). (Perform counting with the count clock selected by the TQ0CTL0.TQ0CKS0 to TQ0CTL0.TQ0CKS2 bits.)
1	Enable operation with external event count input (EVTQ0 pin). (Perform counting at the valid edge of the external event count input signal.)
The TQ0EEE bit selects whether counting is performed with the internal count clock or the valid edge of the external event count input.	

TQnMD2	TQnMD1	TQnMD0	Timer mode selection
0	0	0	Interval timer mode
0	0	1	External event count mode[Note 2]
0	1	0	External trigger pulse output mode[Note 2]
0	1	1	One-shot pulse output mode[Note 2]
1	0	0	PWM output mode[Note 2]
1	0	1	Free-running timer mode
1	1	0	Pulse width measurement mode[Note 2]
1	1	1	6-phase PWM output mode[Note 3]

Notes 1. These bits can be set only in TMQ0. Be sure to clear bits 5 and 6 of TMQ1 to 0.

2. These modes can be set only in TMQ0. Do not set them in TMQ1.

3. These modes can be set only in TMQ1. Do not set them in TMQ0. This mode cannot be used when only TMQ1 is used. For details, see **CHAPTER 9 MOTOR CONTROL FUNCTION**.

Cautions 1. The TQ0EST 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 TQ0EEE bit.

3. Set the TQnEEE and TQnMD2 to TQnMD0 bits when the TQnCTL0.TQnCE bit = 0. (The same value can be written when the TQnCE bit = 1.) The operation is not guaranteed when rewriting is performed with the TQnCE bit = 1. If rewriting was mistakenly performed, clear the TQnCE bit to 0 and then set the bits again.

4. Be sure to clear bits 3, 4, and 7 to "0".

(3) TMQn I/O control register 0 (TQnIOC0)

The TQnIOC0 register is an 8-bit register that controls the timer output (TOQ00, TOQ01/TOQH01, TOQ02, TOQ03/TOQH02, TOQH03, and TOQ1T1 to TOQ1T3 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: TQ0IOC0 FFFFF5C2H, TQ1IOC0 FFFFF602H

TQnIOC0 (n = 0, 1; b = 1 to 3)

7	<6>	5	<4>	3	<2>	1	<0>
TQ0OL3 Note 1	TQ0OE3 Note 1	TQ0OL2 Note 1	TQ0OE2 Note 1	TQ0OL1 Note 1	TQ0OE1 Note 1	TQnOL0	TQnOE0

TQnOLm	Output level setting of TOQnm, TOQH0b, TOQ1Tb pins Note 2 (TMQ0: m = 0 to 3, TMQ1: m = 0)
0	TOQnm, TOQH0b, and TOQ1Tb pins start output at high level.
1	TOQnm, TOQH0b, and TOQ1Tb pins start output at low level.

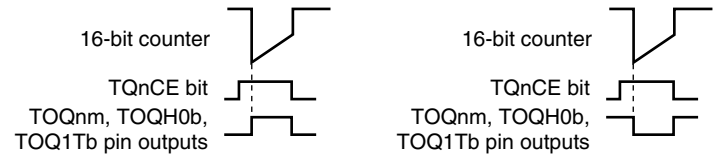
TQnOEm	Output setting of TOQnm, TOQH0b, TOQ1Tb pins (TMQ0: m = 0 to 3, TMQ1: m = 0)
0	Timer output prohibited • Low level is output from the TOQnm, TOQH0b, and TOQ1Tb pins when the TQnOLm bit = 0. • High level is output from the TOQnm, TOQH0b, and TOQ1Tb pins when the TQnOLm bit = 1.
1	Timer output enabled (A pulse is output from the TOQnm, TOQH0b, and TOQ1Tb pins.)

Notes 1. Be sure to clear bits 2 to 7 of the TQ1IOC0 register to 0 when using TMQ1 as an interval timer or a free-running timer.

In addition, set bits 2, 4, and 6 of the TQ1IOC0 register to 1, and bits 3, 5, and 7 to 0 or 1 when using the functions of TOQ1T1 to TOQ1T3 and TOQ1B1 to TOQ1B3 with TMQ1 as a 6-phase PWM output.

2. The output level of the timer output pins (TOQnm, TOQH0b, and TOQ1Tb) specified by the TQnOLm bit is shown below.

<R>



<R> **Cautions 1. If the setting of the TQnIOC0 register is changed when TOQnm, TOQH0b, and TOQ1Tb are set to 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.**

(2/2)

Cautions 2. Rewrite the TQnOLm and TQnOEm bits when the TQnCTL0.TQnCE bit = 0. (The same value can be written when the TQnCE bit = 1.) If rewriting was mistakenly performed, clear (0) the TQnCE bit and then set the bits again.

3. If the TQnOLm bit is manipulated when the TQnCE and TQnOE bits are 0, the output level of the TOQnm, TOQH0b, and TOQ1Tb pins changes.

4. To generate the TOQ1Tb 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 TOQ1Tb pin output mode using the TQ1IOC0 register. At this time, be sure to clear the TQ1OL0 bit to 0 and set the TQ1OE0 bit to 1 (b = 1 to 3).

(4) TMQ0 I/O control register 1 (TQ0IOC1)

The TQ0IOC1 register is an 8-bit register that controls the valid edge for the capture trigger input signals (TIQ00 to TIQ03 pins).

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

Reset sets this register to 00H.

Remark TMQ1 does not have the TQ1IOC1 register.

After reset: 00H R/W Address: FFFFF5C3H

	7	6	5	4	3	2	1	0
TQ0IOC1	TQ0IS7	TQ0IS6	TQ0IS5	TQ0IS4	TQ0IS3	TQ0IS2	TQ0IS1	TQ0IS0

TQ0IS7	TQ0IS6	Capture trigger input signal (TIQ03 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

TQ0IS5	TQ0IS4	Capture trigger input signal (TIQ02 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

TQ0IS3	TQ0IS2	Capture trigger input signal (TIQ01 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

TQ0IS1	TQ0IS0	Capture trigger input signal (TIQ00 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 TQ0IS7 to TQ0IS0 bits when the TQ0CTL0.TQ0CE bit = 0. (The same value can be written when the TQ0CE bit = 1.) If rewriting was mistakenly performed, clear (0) the TQ0CE bit and then set the bits again.

2. The TQ0IS7 to TQ0IS0 bits are valid only in the free-running timer mode (only when TQ0OPT0.TQ0CCS3 to TQ0OPT0.TQ0CCS0 bits = 1111) and pulse width measurement mode. In all other modes, a capture operation is not performed.

(5) TMQ0 I/O control register 2 (TQ0IOC2)

The TQ0IOC2 register is an 8-bit register that controls the valid edge for the external event count input signal (EVTQ0 pin).

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

Reset sets this register to 00H.

Remark TMQ1 does not have the TQ1IOC2 register.

After reset: 00H R/W Address: FFFFF5C4H

	7	6	5	4	3	2	1	0
TQ0IOC2	0	0	0	0	TQ0EES1	TQ0EES0	0	0

TQ0EES1	TQ0EES0	External event count input signal (EVTQ0 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

Cautions 1. Rewrite the TQ0EES1 and TQ0EES0 bits when the TQ0CTL0.TQ0CE bit = 0. (The same value can be written when the TQ0CE bit = 1.) If rewriting was mistakenly performed, clear (0) the TQ0CE bit and then set the bits again.

2. The TQ0EES1 and TQ0EES0 bits are valid only when the TQ0CTL1.TQ0EEE bit = 1 or when the external event count mode is set (TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 bits = 001).

3. Be sure to clear bits 0, 1, and 4 to 7 to "0".

(6) TMQn option register 0 (TQnOPT0)

The TQnOPT0 register is an 8-bit register used to set the capture/compare operation and detect 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: TQ0OPT0 FFFFF5C5H, TQ1OPT0 FFFFF605H

TQnOPT0 (n = 0, 1)

<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
TQ0CCS3[Note 1]	TQ0CCS2[Note 1]	TQ0CCS1[Note 1]	TQ0CCS0[Note 1]	0	TQ1CMS[Note 2]	TQ1CUF[Note 2]	TQnOVF

TQ0CCSm	TQ0CCRm register capture/compare selection (m = 0 to 3)
0	Compare register selected
1	Capture register selected (cleared by TQ0CTL0.TQ0CE bit = 0)
The TQ0CCSm bit setting is valid only in the free-running timer mode.	

TQnOVF	TMQn overflow flag
Set (1)	Overflow occurred
Reset (0)	0 written to TQnOVF bit or TQnCTL0.TQnCE bit = 0

- The TQnOVF bit is set (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 (INTTQnOV) is generated at the same time that the TQnOVF bit is set (1). The INTTQnOV signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode.
- The TQnOVF bit is not cleared to 0 even when the TQnOVF bit or the TQnOPT0 register is read when the TQnOVF bit = 1.
- Before clearing the TQnOVF bit to 0 after generation of the INTTQnOV signal, be sure to confirm (by reading) that the TQnOVF bit is set to 1.
- The TQnOVF bit can be read or written, but the TQnOVF bit cannot be set (1) by software. Writing 1 has no effect on the TMQn operation.

Notes 1. Valid only in TMQ0. Be sure to clear bits 7 to 4 in TMQ1 to 0.

2. Valid only in TMQ1. Be sure to clear bits 2 and 1 of TMQ0 to 0. For details of the TQ1CMS and TQ1CUF bits, see **CHAPTER 9 MOTOR CONTROL FUNCTION**.

Cautions 1. Rewrite the TQ0CCS3 to TQ0CCS0 bits when the TQ0CE bit = 0. (The same value can be written when the TQ0CE bit = 1.) If rewriting was mistakenly performed, clear (0) the TQ0CE bit = 0 and then set the bits again.

2. Be sure to clear bit 3 to "0".

(7) TMQn capture/compare register 0 (TQnCCR0)

The TQ0CCR0 register is a 16-bit register that can be used as a capture register or a compare register depending on the mode. The TQ1CCR0 register is a 16-bit register that can only be used as a compare register.

The TQ0CCR0 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 TQ0OPT0.TQ0CCS0 bit. In the pulse width measurement mode, the TQ0CCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TQnCCR0 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: TQ0CCR0 FFFFF5C6H, TQ1CCR0 FFFFF606H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQnCCR0 (n = 0, 1)																

(a) Function as compare register

The TQnCCR0 register can be rewritten even when the TQnCTL0.TQnCE bit = 1.

The set value of the TQnCCR0 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 (INTTQnCC0) is generated. If TOQn0 pin output is enabled at this time, the output of the TOQn0 pin is inverted.

When the TQnCCR0 register is used as a cycle register in the interval timer mode, external event count mode[Note], external trigger pulse output mode[Note], one-shot pulse output mode[Note], or PWM output mode[Note], 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 the TQnCTL0.TQnCE bit = 0.

Note These modes can be set only in TMQ0. They cannot be set in TMQ1.

(b) Function as capture register (TQ0CCR0 register only)

When the TQ0CCR0 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 TQ0CCR0 register if the valid edge of the capture trigger input pin (TIQ00 pin) is detected. In the pulse width measurement mode, the count value of the 16-bit counter is stored in the TQ0CCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIQ00 pin) is detected.

Even if the capture operation and reading the TQ0CCR0 register conflict, the correct value of the TQ0CCR0 register can be read.

The capture register is cleared by the TQ0CTL0.TQ0CE 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 7-4. 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[Notes 1, 2]	Compare register	Batch write[Note 3]
One-shot pulse output[Notes 1, 2]	Compare register	Anytime write
PWM output[Note 1]	Compare register	Batch write[Note 3]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None

Notes 1. TMQ0 only

2. This mode can be set only with the software trigger. No external trigger input pin is available.

3. Writing to the TQ0CCR1 register is the trigger.

Remark For anytime write and batch write, see **7.6 (2) Anytime write and batch write**.

(8) TMQn capture/compare register 1 (TQnCCR1)

The TQ0CCR1 register is a 16-bit register that can be used as a capture register or a compare register depending on the mode. The TQ1CCR1 register is a 16-bit register that can only be used as a compare register.

The TQ0CCR1 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 TQ0OPT0.TQ0CCS1 bit. In the pulse width measurement mode, the TQ0CCR1 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TQnCCR1 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: TQ0CCR1 FFFFF5C8H, TQ1CCR1 FFFFF608H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQnCCR1 (n = 0, 1)																

(a) Function as compare register

The TQnCCR1 register can be rewritten even when the TQnCTL0.TQnCE bit = 1.

The set value of the TQnCCR1 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 (INTTQnCC1) is generated. If TOQ01/TOQH01 pin output is enabled at this time, the output of the TOQ01/TOQH01 pin is inverted (the TOQ11 and TOQH11 pins are not provided).

The compare register is not cleared by the TQnCTL0.TQnCE bit = 0.

(b) Function as capture register (TQ0CCR1 register only)

When the TQ0CCR1 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 TQ0CCR1 register if the valid edge of the capture trigger input pin (TIQ01 pin) is detected. In the pulse width measurement mode, the count value of the 16-bit counter is stored in the TQ0CCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIQ01 pin) is detected.

Even if the capture operation and reading the TQ0CCR1 register conflict, the correct value of the TQ0CCR1 register can be read.

The capture register is cleared by the TQ0CTL0.TQ0CE 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 7-5. 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[Notes 1, 2]	Compare register	Batch write[Note 3]
One-shot pulse output[Notes 1, 2]	Compare register	Anytime write
PWM output[Note 1]	Compare register	Batch write[Note 3]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None

Notes 1. TMQ0 only

2. This mode can be set only with the software trigger. No external trigger input pin is available.

3. Writing to the TQ0CCR1 register is the trigger.

Remark For anytime write and batch write, see **7.6 (2) Anytime write and batch write**.

(9) TMQn capture/compare register 2 (TQnCCR2)

The TQ0CCR2 register is a 16-bit register that can be used as a capture register or a compare register depending on the mode. The TQ1CCR2 register is a 16-bit register that can only be used as a compare register.

The TQ0CCR2 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 TQ0OPT0.TQ0CCS2 bit. In the pulse width measurement mode, the TQ0CCR2 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TQnCCR2 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: TQ0CCR2 FFFFF5CAH, TQ1CCR2 FFFFF60AH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQnCCR2 (n = 0, 1)																

(a) Function as compare register

The TQnCCR2 register can be rewritten even when the TQnCTL0.TQnCE bit = 1.

The set value of the TQnCCR2 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 (INTTQnCC2) is generated. If TOQ02/TOQH02 pin output is enabled at this time, the output of the TOQ02/TOQH02 pin is inverted (the TOQ12 and TOQH12 pins are not provided).

The compare register is not cleared by the TQnCTL0.TQnCE bit = 0.

(b) Function as capture register (TQ0CCR2 register only)

When the TQ0CCR2 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 TQ0CCR2 register if the valid edge of the capture trigger input pin (TIQ02 pin) is detected. In the pulse width measurement mode, the count value of the 16-bit counter is stored in the TQ0CCR2 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIQ02 pin) is detected.

Even if the capture operation and reading the TQ0CCR2 register conflict, the correct value of the TQ0CCR2 register can be read.

The capture register is cleared by the TQ0CTL0.TQ0CE 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 7-6. 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[Notes 1, 2]	Compare register	Batch write[Note 3]
One-shot pulse output[Notes 1, 2]	Compare register	Anytime write
PWM output[Note 1]	Compare register	Batch write[Note 3]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None

Notes 1. TMQ0 only

2. This mode can be set only with the software trigger. No external trigger input pin is available.

3. Writing to the TQ0CCR1 register is the trigger.

Remark For anytime write and batch write, see **7.6 (2) Anytime write and batch write**.

(10) TMQn capture/compare register 3 (TQnCCR3)

The TQ0CCR3 register is a 16-bit register that can be used as a capture register or a compare register depending on the mode. The TQ1CCR3 register is a 16-bit register that can only be used as a compare register.

The TQ0CCR3 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 TQ0OPT0.TQ0CCS3 bit. In the pulse width measurement mode, the TQ0CCR3 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TQnCCR3 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: TQ0CCR3 FFFFF5CCH, TQ1CCR3 FFFFF60CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQnCCR3 (n = 0, 1)																

(a) Function as compare register

The TQnCCR3 register can be rewritten even when the TQnCTL0.TQnCE bit = 1.

The set value of the TQnCCR3 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 (INTTQnCC3) is generated. If TOQ03/TOQH03 pin output is enabled at this time, the output of the TOQ03/TOQH03 pin is inverted (the TOQ13 and TOQH13 pins are not provided).

The compare register is not cleared by the TQnCTL0.TQnCE bit = 0.

(b) Function as capture register (TQ0CCR3 register only)

When the TQ0CCR3 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 TQ0CCR3 register if the valid edge of the capture trigger input pin (TIQ03 pin) is detected. In the pulse width measurement mode, the count value of the 16-bit counter is stored in the TQ0CCR3 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIQ03 pi) is detected.

Even if the capture operation and reading the TQ0CCR3 register conflict, the correct value of the TQ0CCR3 register can be read.

The capture register is cleared by the TQ0CTL0.TQ0CE 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 7-7. 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[Notes 1, 2]	Compare register	Batch write[Note 3]
One-shot pulse output[Notes 1, 2]	Compare register	Anytime write
PWM output[Note 1]	Compare register	Batch write[Note 3]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement[Note 1]	Capture register	None

Notes 1. TMQ0 only

2. This mode can be set only with the software trigger. No external trigger input pin is available.

3. Writing to the TQ0CCR1 register is the trigger.

Remark For anytime write and batch write, see **7.6 (2) Anytime write and batch write**.

(11) TMQn counter read buffer register (TQnCNT)

The TQnCNT register is a read buffer register that can read the count value of the 16-bit counter.

If this register is read when the TQnCTL0.TQnCE 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 TQnCNT register is cleared to 0000H when the TQnCE bit = 0. If the TQnCNT 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 TQnCE bit is cleared to 0 after reset, and the TQnCNT register is cleared to 0000H.

After reset: 0000H R Address: TQ0CNT FFFFF5CEH, TQ1CNT FFFFF60EH

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

7.5 Timer Output Operations

The following table shows the operations and output levels of the TOQ00 to TOQ03, TOQ10, and TOQH01 to TOQH03 pins.

Table 7-8. Timer Output Control in Each Mode

Operation Mode	TOQn0 Pin	TOQ01 to TOQ03 Pins	TOQH01 to TOQH03 Pins
Interval timer mode	PWM output		None
External event count mode	None		
External trigger pulse output mode	PWM output[Note]	External trigger pulse output	None
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)		None
Pulse width measurement mode	None		

Note TOQ00 pin only

Remark n = 0, 1

Table 7-9. Truth Table of TOQ00 to TOQ03, TOQ10, and TOQH01 to TOQH03 Pins Under Control of Timer Output Control Bits

TQnIOC0.TQnOLa Bit	TQnIOC0.TQnOEa Bit	TQnCTL0.TQnCE Bit	Level of TOQna and TOQH1b Pins
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 a = 0 to 3 when n = 0
a = 0 when n = 1
b = 1 to 3

7.6 Operation

The functions that can be realized differ between TMQ0 and TMQ1. The functions of each channel are shown below.

Table 7-10. TMQ0 Specifications in Each Mode

Operation	TQ0CTL1.TQ0EST Bit (Software Trigger Bit)	Capture/Compare Register Setting	Compare Register Write Method
Interval timer mode	Invalid	Compare only	Anytime write
External event count mode	Invalid	Compare only	Anytime write
External trigger pulse output mode	Valid	Compare only	Batch write
One-shot pulse output mode	Valid	Compare only	Anytime write
PWM output mode	Invalid	Compare only	Batch write
Free-running timer mode	Invalid	Switchable	Anytime write
Pulse width measurement mode	Invalid	Capture only	Not applicable

Table 7-11. TMQ1 Specifications in Each Mode

Operation	Software Trigger Bit	Capture/Compare Register Setting	Compare Register Write Method
Interval timer mode	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	Compare only	Anytime write
Pulse width measurement mode	None		

Remark TMQ1 has a function to execute tuning with TMP1. For details, see **CHAPTER 9 MOTOR CONTROL FUNCTION**.

(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

<R> **(a) Counting start operation**

- In external event count mode
  When the TQ0CTL0.TQ0CE bit is set from 0 to 1, the 16-bit counter is set to 0000H.
  After that, it counts up to 0001H, 0002H, 0003H, … each time the valid edge of external event count input (EVTQ0) is detected.
- In modes other than the above
  The 16-bit counter of TMQn 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 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 INTTQnCCa 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 timer mode or pulse width measurement mode. If the counter overflows, the TQnOPT0.TQnOVF bit is set to 1 and an interrupt request signal (INTTQnOV) is generated. Note that the INTTQnOV 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 in the pulse width measurement mode and the counter counts up from FFFFH to 0000H

Caution After the overflow interrupt request signal (INTTQnOV) has been generated, be sure to check that the overflow flag (TQnOVF bit) is set to 1.

(d) Counter read operation during counting operation

The value of the 16-bit counter of TMQn can be read by using the TQnCNT register during the count operation. When the TQnCTL0.TQnCE bit = 1, the value of the 16-bit counter can be read by reading the TQ0CNT register. However, when the TQnCE bit = 0, the 16-bit counter is FFFFH and the TQnCNT register is 0000H.

(e) Interrupt operation

TMQn generates the following five interrupt request signals.

- INTTQnCC0 interrupt: This signal functions as a match interrupt request signal of the CCR0 buffer register and as a capture interrupt request signal to the TQnCCR0 register.
- INTTQnCC1 interrupt: This signal functions as a match interrupt request signal of the CCR1 buffer register and as a capture interrupt request signal to the TQnCCR1 register.
- INTTQnCC2 interrupt: This signal functions as a match interrupt request signal of the CCR2 buffer register and as a capture interrupt request signal to the TQnCCR2 register.
- INTTQnCC3 interrupt: This signal functions as a match interrupt request signal of the CCR3 buffer register and as a capture interrupt request signal to the TQnCCR3 register.
- INTTQnOV interrupt: This signal functions as an overflow interrupt request signal.

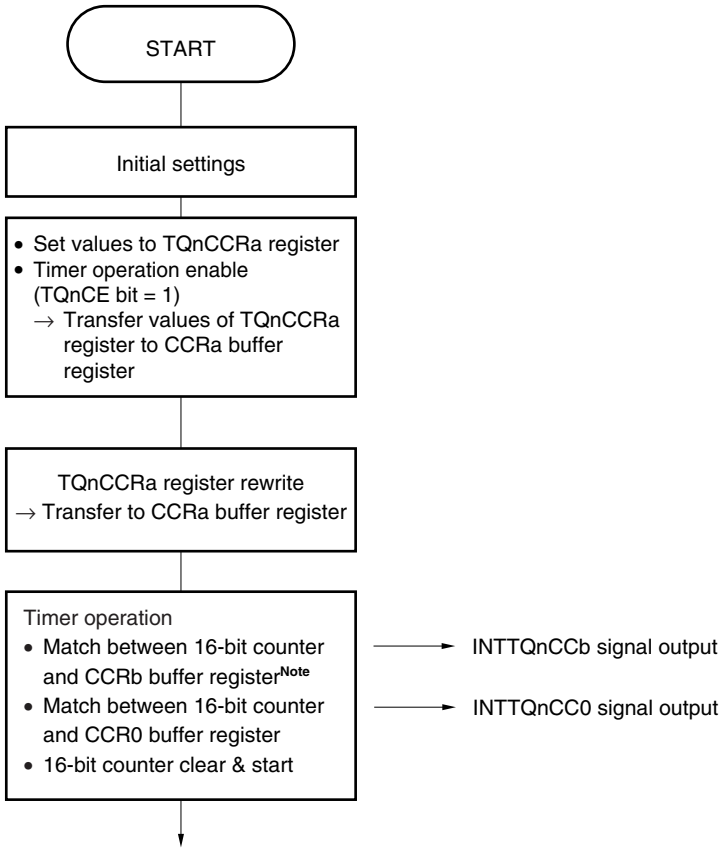
(2) Anytime write and batch write

The TQnCCR0 to TQnCCR3 registers can be rewritten in the TMQn during timer operation (TQnCTL0.TQnCE 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 TQnCCR0 to TQnCCR3 registers to the CCR0 to CCR3 buffer registers during the timer operation (n = 0, 1).

Figure 7-4. Flowchart of Basic Operation for Anytime Write

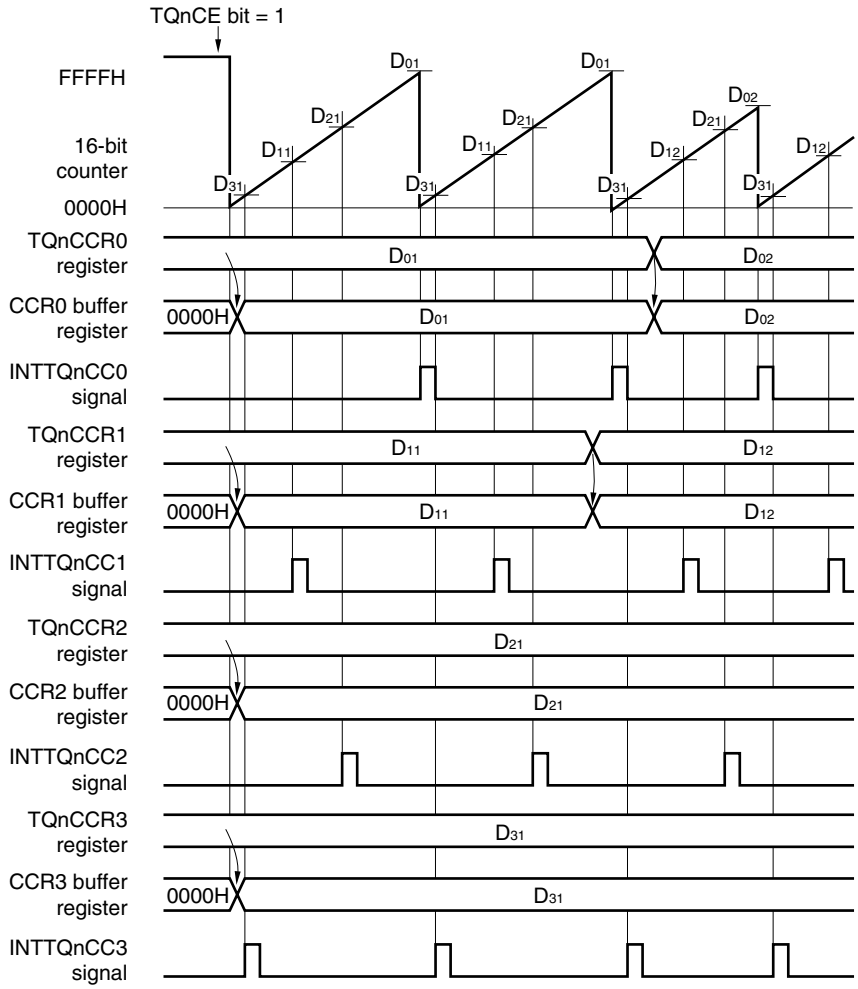


Note The 16-bit counter is not cleared upon a match between the value of the 16-bit counter and the value of the CCRb buffer register. It is cleared upon a match between the value of the 16-bit counter and the value of the CCR0 buffer register.

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 7-5. Timing of Anytime Write



Remarks 1. D_{01}, D_{02}: Setting values of TQnCCR0 register

D_{11}, D_{12}: Setting values of TQnCCR1 register

D_{21}: Setting value of TQnCCR2 register

D_{31}: Setting value of TQnCCR3 register

2. The above timing chart illustrates an example of the operation in the interval timer mode.

3. n = 0, 1

(b) Batch write

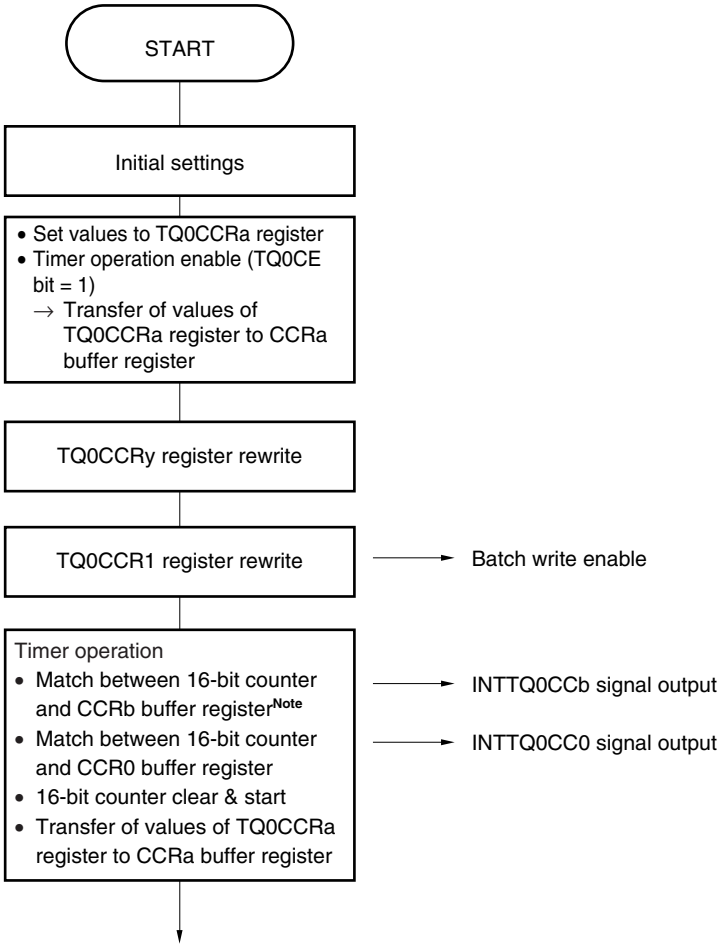
In this mode, data is transferred all at once from the TQ0CCR0 to TQ0CCR3 registers to the CCR0 to CCR3 buffer registers during timer operation. This data is transferred upon a match between the value of the CCR0 buffer register and the value of the 16-bit counter. Transfer is enabled by writing to the TQ0CCR1 register.

Whether to enable or disable the next transfer timing is controlled by writing or not writing to the TQ0CCR1 register.

In order for the setting value when the TQ0CCR0 to TQ0CCR3 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 TQ0CCR0 and finally write to the TQ0CCR1 register before the 16-bit counter value and the CCR0 buffer register value match. The values of the TQ0CCR0 to TQ0CCR3 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 TQ0CCR0, TQ0CCR2, or TQ0CCR3 register, also write the same value (same as preset value of the TQ0CCR1 register) to the TQ0CCR1 register.

Remark TMQ1 cannot be set in a mode in which it can be rewritten by batch write.

Figure 7-6. Flowchart of Basic Operation for Batch Write



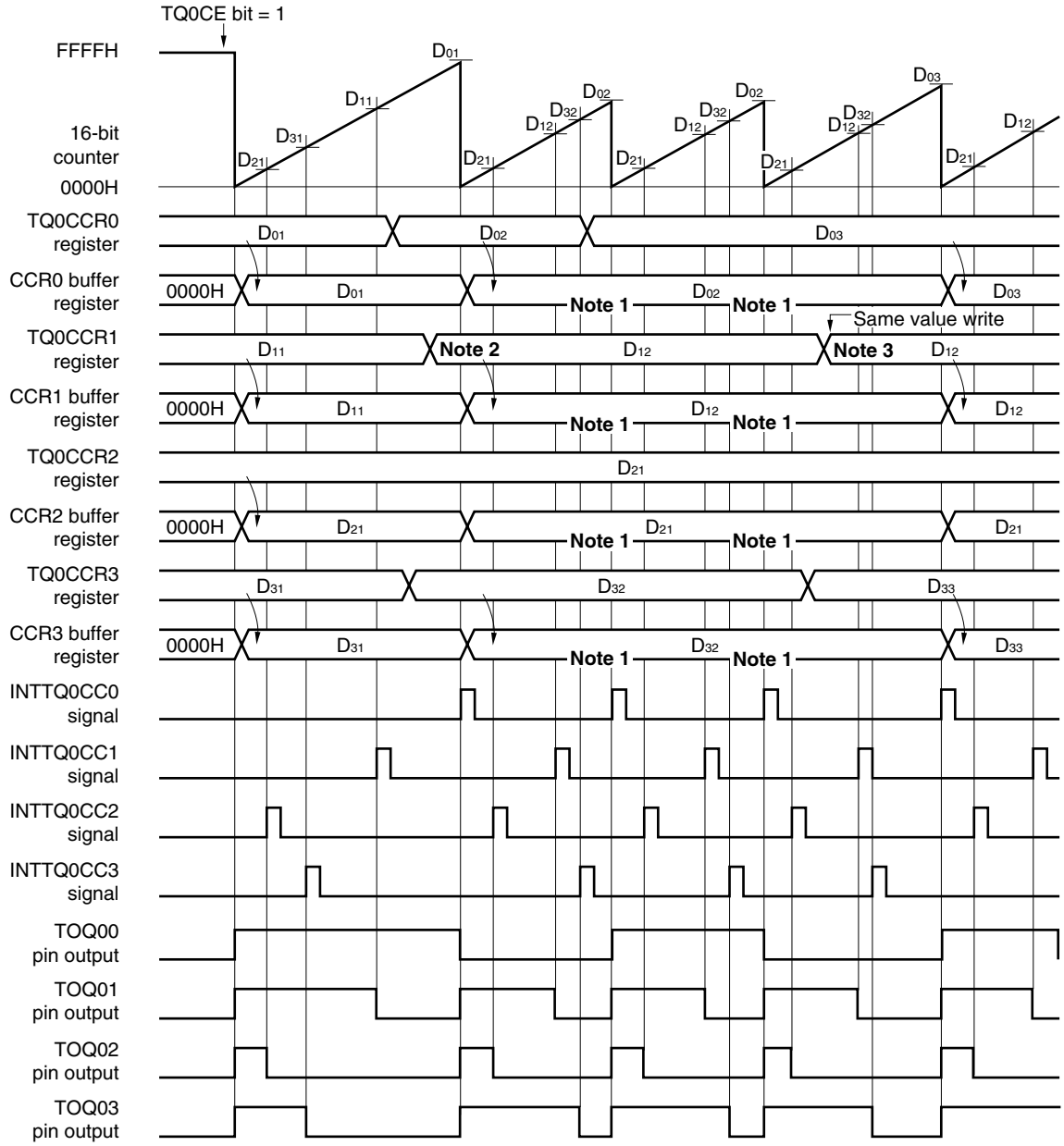
Note The 16-bit counter is not cleared upon a match between the value of the 16-bit counter and the value of the CCRb buffer register. It is cleared upon a match between the value of the 16-bit counter and the value of the CCR0 buffer register.

Caution Writing to the TQ0CCR1 register includes enabling of batch write. Thus, rewrite the TQ0CCR1 register after rewriting the TQ0CCR0, TQ0CCR2, and TQ0CCR3 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 7-7. Timing of Batch Write



Notes 1. Because the TQ0CCR1 register was not rewritten, D_{02} is not transferred.

2. Because TQ0CCR1 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 TQ0CCR0 register (D_{01}).

3. Because TQ0CCR1 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 TQ0CCR0 register (D_{12}).

Remarks 1. D_{01}, D_{02}, D_{03}: Setting values of TQ0CCR0 register
D_{11}, D_{12}: Setting values of TQ0CCR1 register
D_{21}: Setting value of TQ0CCR2 register
D_{31}, D_{32}, D_{33}: Setting values of TQ0CCR3 register

2. The above timing chart illustrates an example of the operation in the PWM output mode.

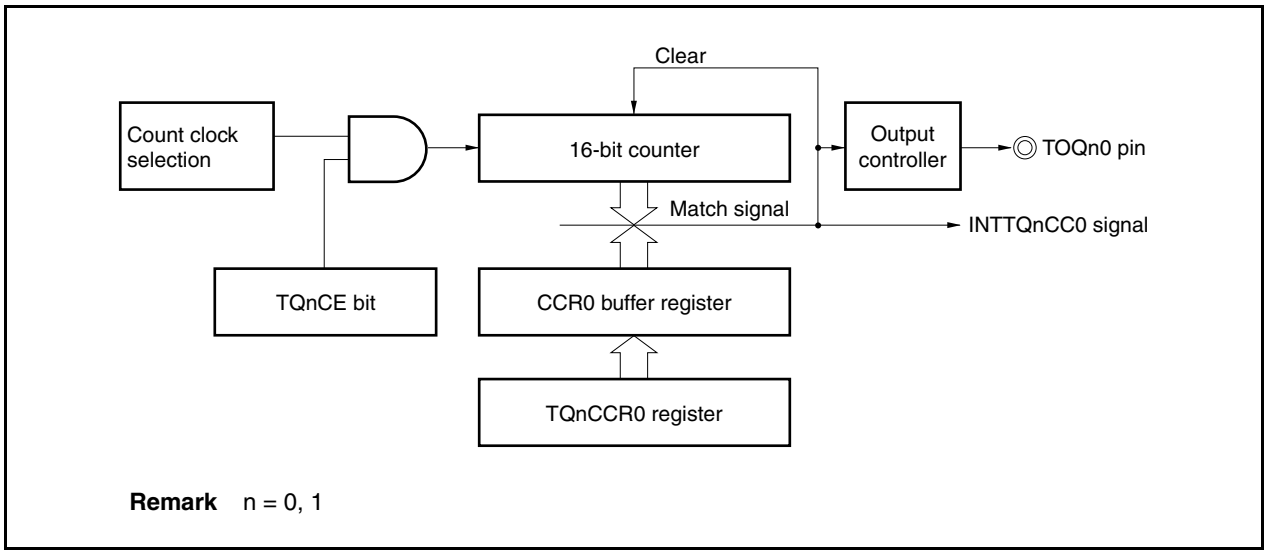
7.6.1 Interval timer mode (TQnMD2 to TQnMD0 bits = 000)

In the interval timer mode, an interrupt request signal (INTTQnCC0) is generated at the interval set by the TQnCCR0 register if the TQnCTL0.TQnCE 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 TOQn0 pin.

The TQnCCR1 to TQnCCR3 registers are not used in the interval timer mode. However, the set value of the TQnCCR1 to TQnCCR3 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 (INTTQnCC1 to INTTQnCC3) are generated. In addition, a PWM waveform with a duty factor of 50%, which is inverted when the INTTQ0CC1 to INTTQ0CC3 signals are generated, can be output from the TOQ01 to TOQ03 pins.

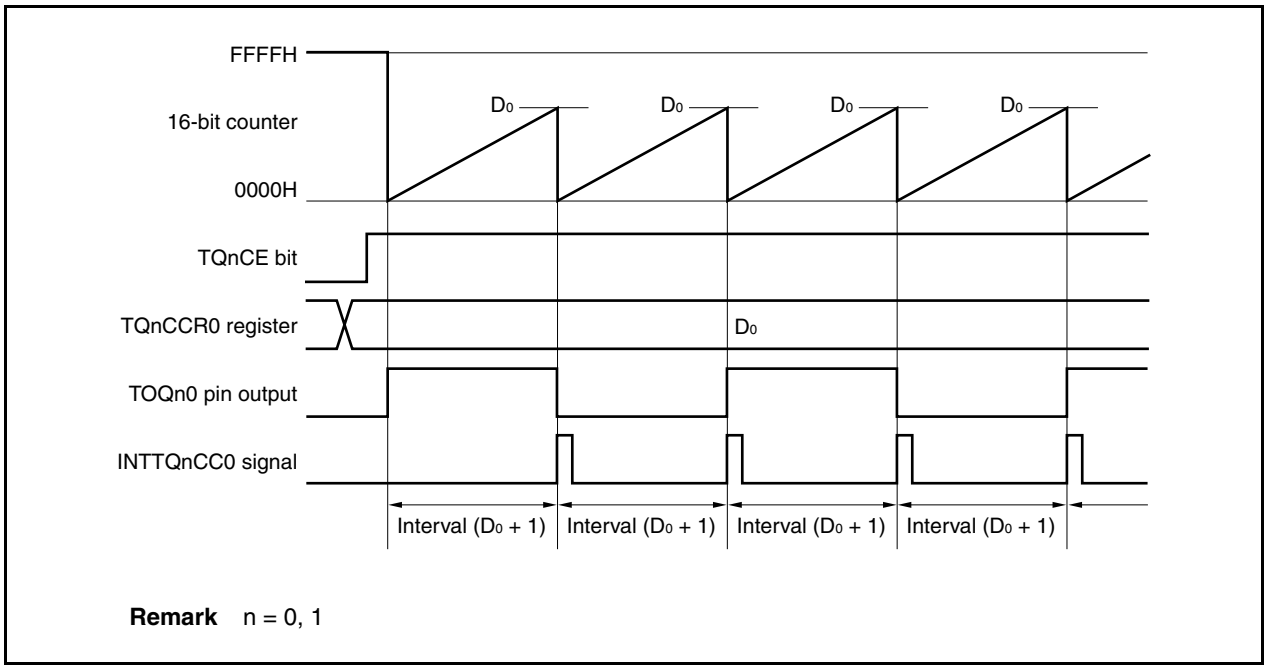
The value of the TQnCCR1 to TQnCCR3 registers can be rewritten even while the timer is operating.

Figure 7-8. Interval Timer Configuration



Remark n = 0, 1

Figure 7-9. Basic Timing of Operation in Interval Timer Mode



Remark n = 0, 1

When the TQnCE 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 TOQn0 pin is inverted. Additionally, the set value of the TQnCCR0 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 TOQn0 pin is inverted, and a compare match interrupt request signal (INTTQnCC0) is generated.

The interval can be calculated by the following expression.

Interval = (Set value of TQnCCR0 register + 1) × Count clock cycle

Remark n = 0, 1

Figure 7-10. Register Setting for Interval Timer Mode Operation (1/3)

(a) TMQn control register 0 (TQnCTL0)

	TQnCE					TQnCKS2	TQnCKS1	TQnCKS0
TQnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock

0: Stop counting
1: Enable counting

(b) TMQn control register 1 (TQnCTL1)

		TQ0EST	TQ0EEE			TQnMD2	TQnMD1	TQnMD0
<R>	TQnCTL1	0	0	0/1[Note]	0	0	0	0	0

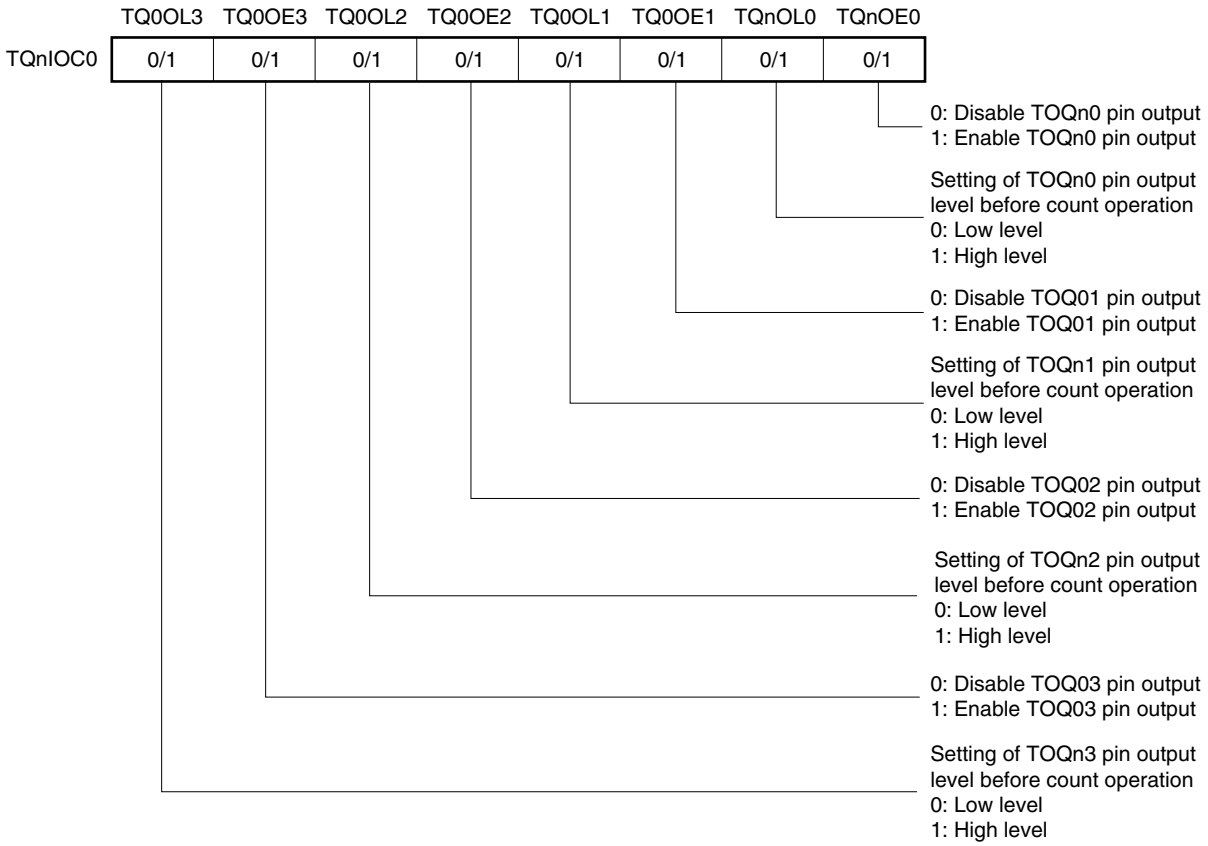
0, 0, 0:
Interval timer mode

0: Operate on count clock selected by TQ0CKS0 to TQ0CKS2 bits
1: Count with external event count input signal

<R> **Note** Enable setting of the TQ0EEE bit to 1 only when timer output (TOQ00, TOQ0b) is used. In this case, set the TQ0CCR0 and TQ0CCRb registers to the same value (b = 1 to 3).

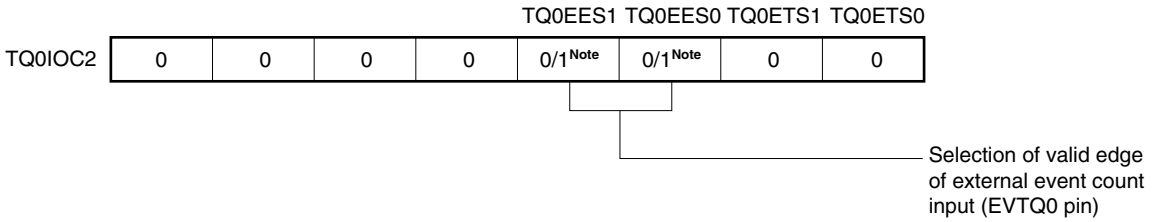
Figure 7-10. Register Setting for Interval Timer Mode Operation (2/3)

(c) TMQn I/O control register 0 (TQnIOC0)



<R>

(d) TMQ0 I/O control register 2 (TQ0IOC2)



Note Enable setting of the TQ0EES1 and TQ0EES0 bits only when timer output (TOQ00 to TOQ03) is used. In this case, set the TQ0CCR0 to TQ0CCR3 registers to the same value.

(e) TMQn counter read buffer register (TQnCNT)

By reading the TQnCNT register, the count value of the 16-bit counter can be read.

(f) TMQn capture/compare register 0 (TQnCCR0)

If the TQnCCR0 register is set to D_0, the interval is as follows.

Interval = $(D_0 + 1) \times$ Count clock cycle

Figure 7-10. Register Setting for Interval Timer Mode Operation (3/3)

(g) TMQn capture/compare registers 1 to 3 (TQnCCR1 to TQnCCR3)

The TQnCCR1 to TQnCCR3 registers are not used in the interval timer mode, but the set values of the TQnCCR1 to TQnCCR3 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 TOQ01 to TOQ03 pin outputs are inverted and the compare match interrupt request signals (INTTQnCC1 to INTTQnCC3) are generated.

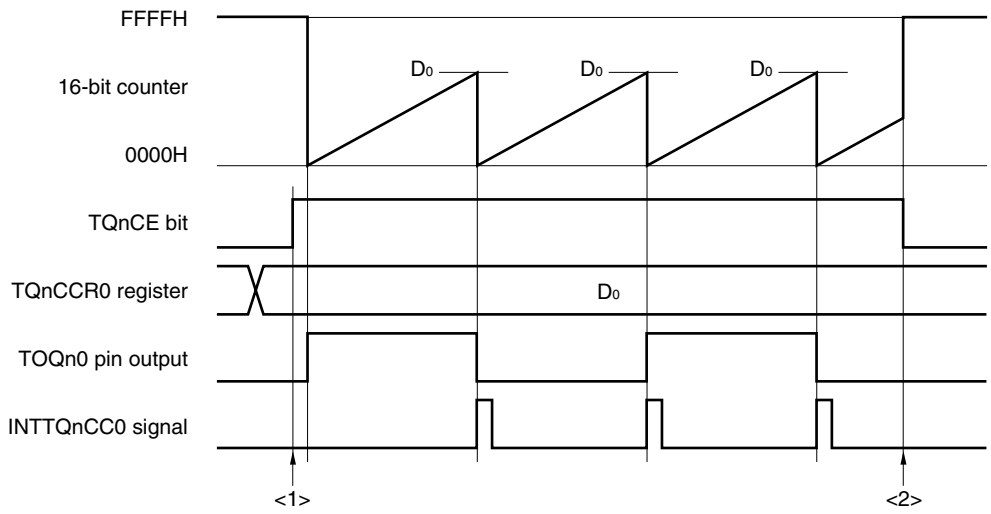
When the TQnCCR1 to TQnCCR3 registers are not used, it is recommended to set their values to FFFFH. Also mask the registers by the interrupt mask flags (TQnCCIC1.TQnCCMK1 to TQnCCIC3.TQnCCMK3).

<R> **Remarks 1.** TMQ0 I/O control register 1 (TQ0IOC1) and TMQn option register 0 (TQnOPT0) are not used in the interval timer mode.

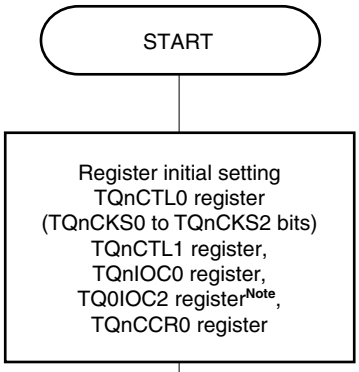
2. n = 0, 1

(1) Interval timer mode operation flow

Figure 7-11. Software Processing Flow in Interval Timer Mode



<1> Count operation start flow

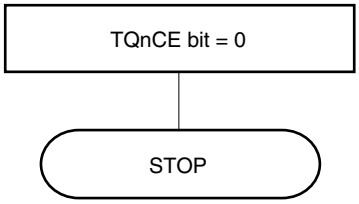


<R>

Initial setting of these registers is performed before setting the TQnCE bit to 1.

The TQnCKS0 to TQnCKS2 bits can be set at the same time when counting has been started (TQnCE bit = 1).

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TQnCE bit to 0. The output level of the TOQn0 pin is as specified by the TQnIOC0 register.

<R> **Note** Enable setting of the TQ0EES1 and TQ0EES0 bits only when timer output (TOQ00 to TOQ03) is used. In this case, set the TQ0CCR0 to TQ0CCR3 registers to the same value.

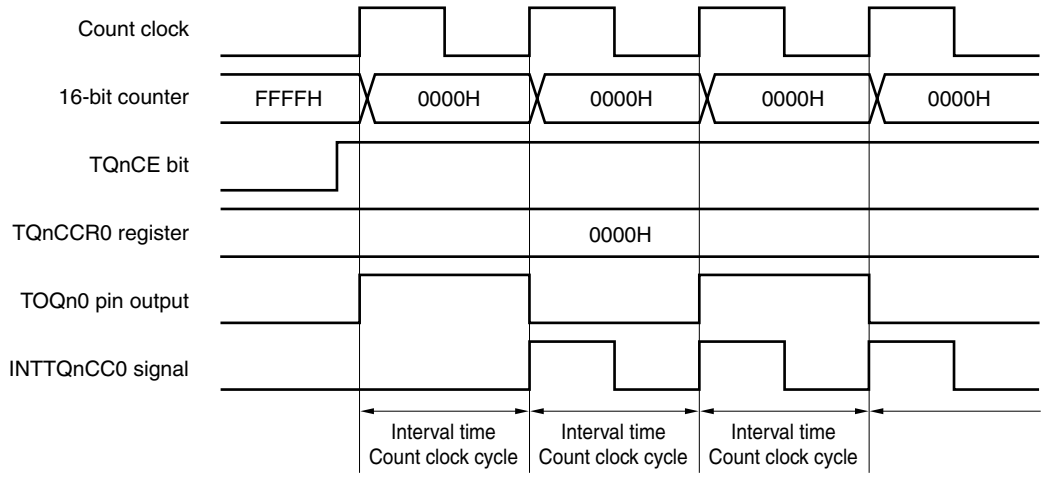
Remark n = 0, 1

(2) Interval timer mode operation timing

(a) Operation if TQnCCR0 register is set to 0000H

If the TQnCCR0 register is set to 0000H, the INTTQnCC0 signal is generated at each count clock, and the output of the TOQn0 pin is inverted.

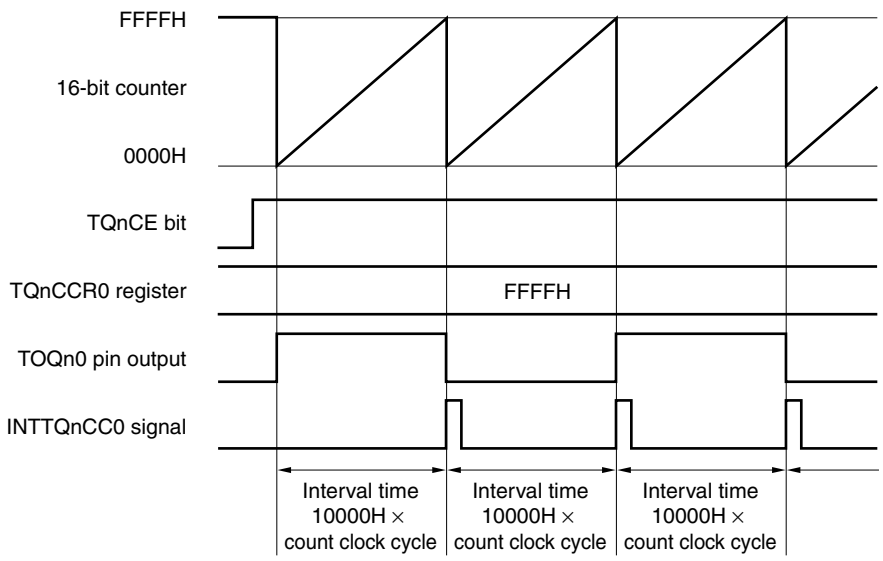
The value of the 16-bit counter is always 0000H.



Remark n = 0, 1

(b) Operation if TQnCCR0 register is set to FFFFH

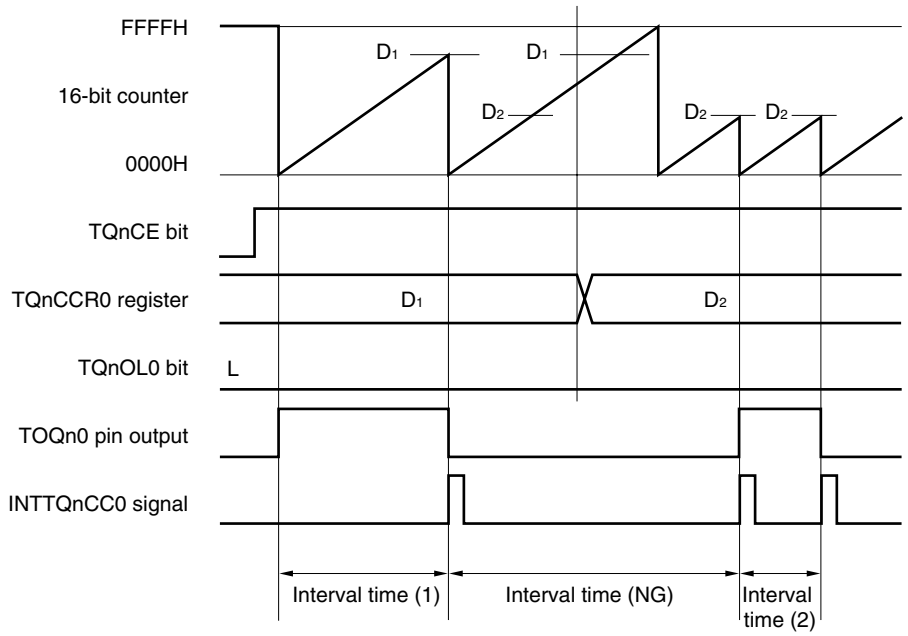
If the TQnCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTQnCC0 signal is generated and the output of the TOQn0 pin is inverted. At this time, an overflow interrupt request signal (INTTQnOV) is not generated, nor is the overflow flag (TQnOPT0.TQnOVF bit) set to 1.



Remark n = 0, 1

(c) Notes on rewriting TQnCCR0 register

If the value of the TQnCCR0 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

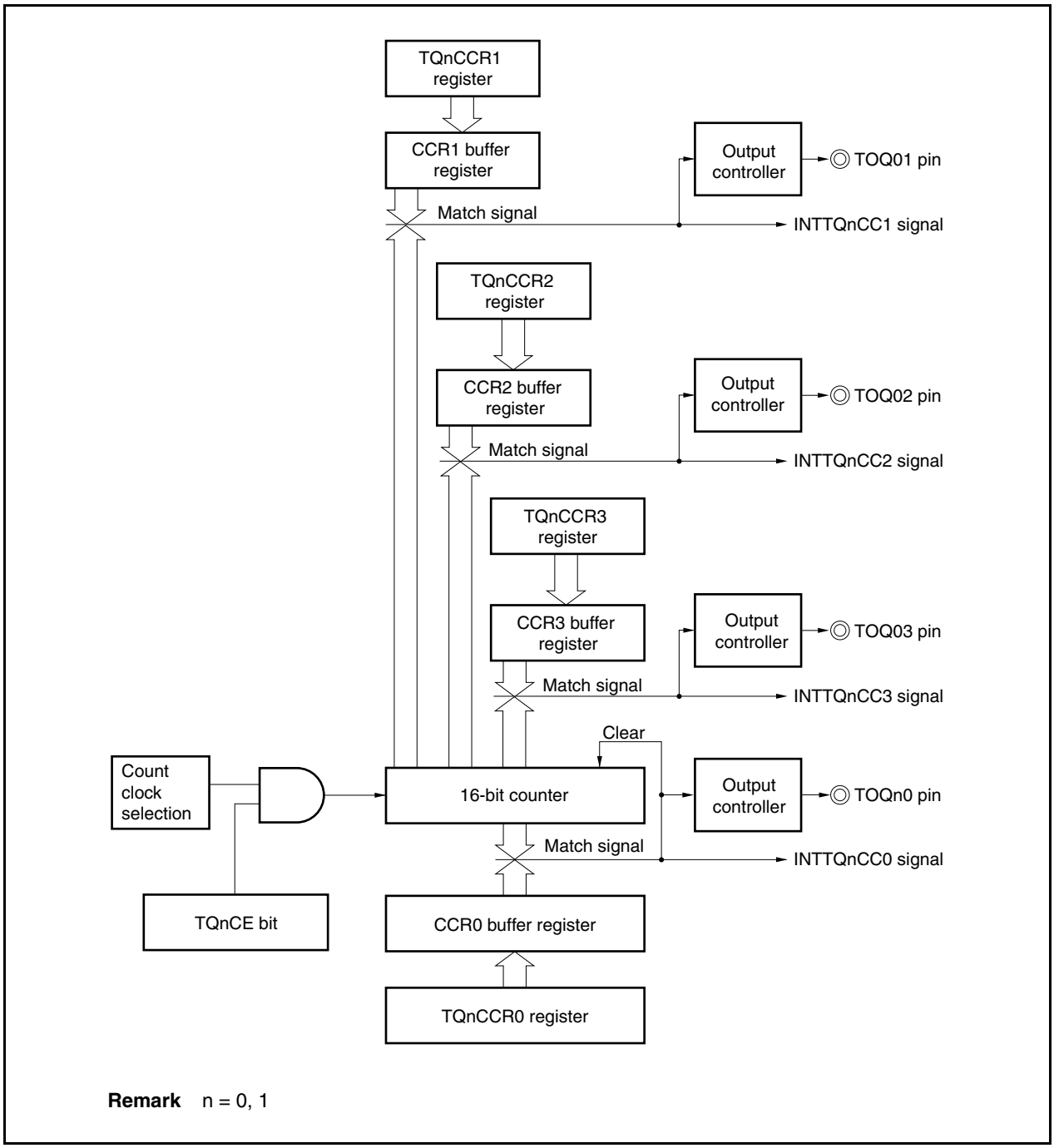
If the value of the TQnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TQnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is 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 INTTQnCC0 signal is generated and the output of the TOQn0 pin is inverted.

Therefore, the INTTQnCC0 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 TQnCCR1 to TQnCCR3 registers

Figure 7-12. Configuration of TQnCCR1 to TQnCCR3 Registers



Remark n = 0, 1

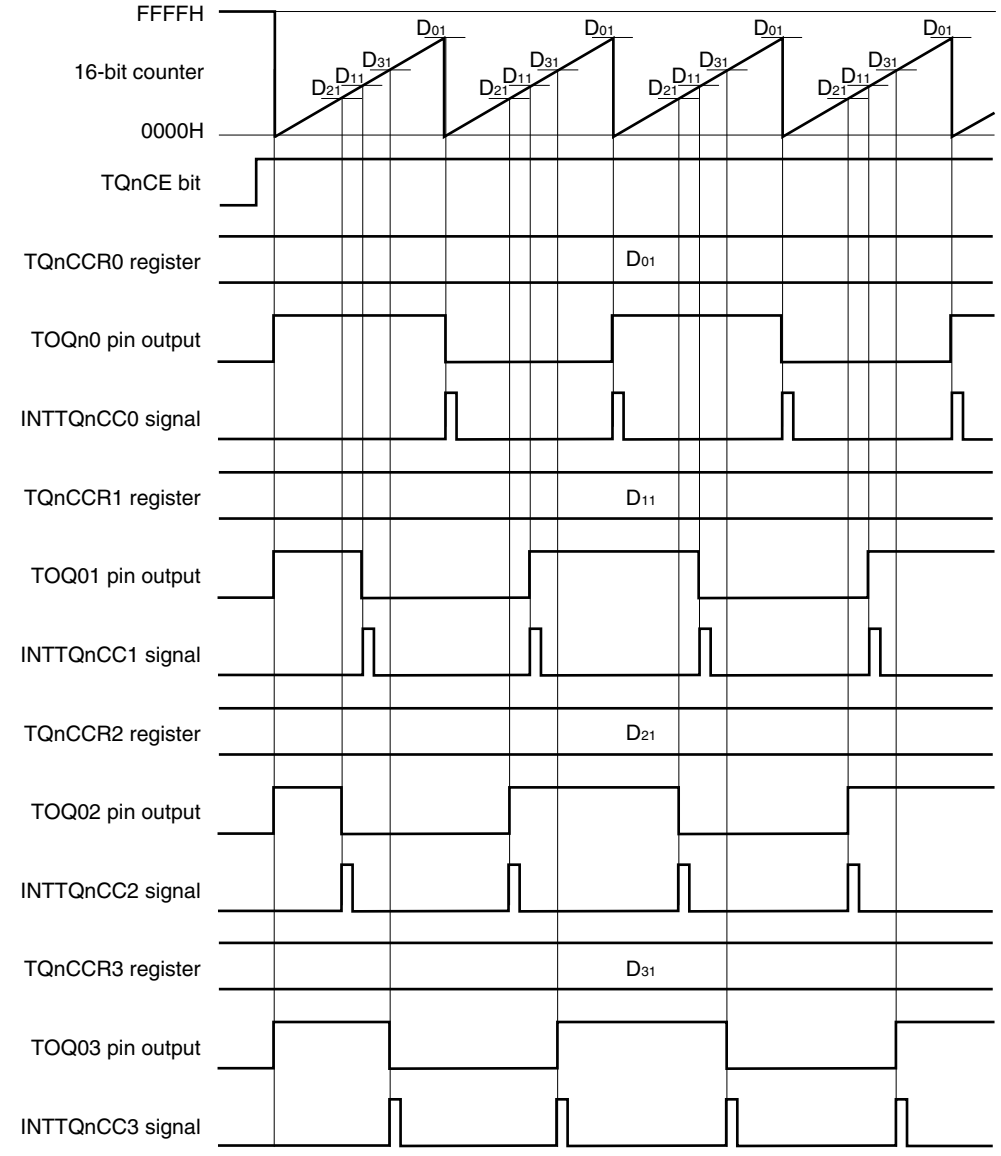
<R> When the TQnCCRb register is set to the same value as that of the TQnCCR0 register, the INTTQnCCb signal is generated at the same timing as the INTTQnCC0 signal is generated, and the TOQ0b pin output is inverted. In other words, a PWM waveform with a duty factor of 50% can be output from the TOQ0b pin.

The following shows the operation when the TQnCCRb register is set to other than the value set in the TQnCCR0 register.

If the set value of the TQnCCRb register is less than the set value of the TQnCCR0 register, the INTTQnCCb signal is generated once per cycle. At the same time, the output of the TOQ0b pin is inverted.

<R> After outputting the short-width pulse first, the TOQ0b pin outputs a PWM waveform with a duty factor of 50%.

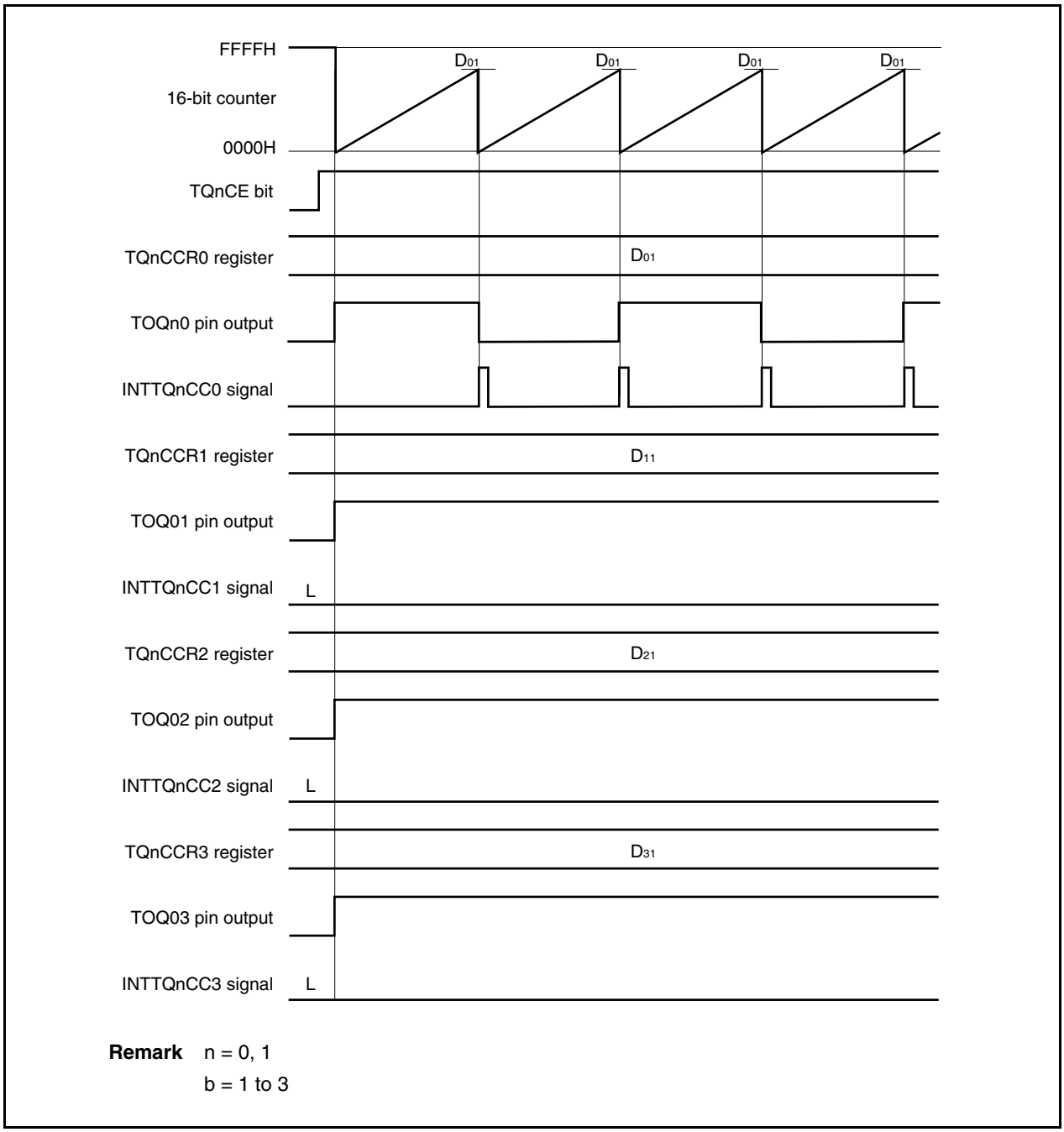
Figure 7-13. Timing Chart When $D_{01} \geq D_{b1}$



Remark n = 0, 1
b = 1 to 3

If the set value of the TQnCCRb register is greater than the set value of the TQnCCR0 register, the count value of the 16-bit counter does not match the value of the TQnCCRb register. Consequently, the INTTQnCCb signal is not generated, nor is the output of the TOQ0b pin changed.
When the TQnCCRb register is not used, it is recommended to set its value to FFFFH.

Figure 7-14. Timing Chart When $D_{01} < D_{b1}$



Remark n = 0, 1
b = 1 to 3

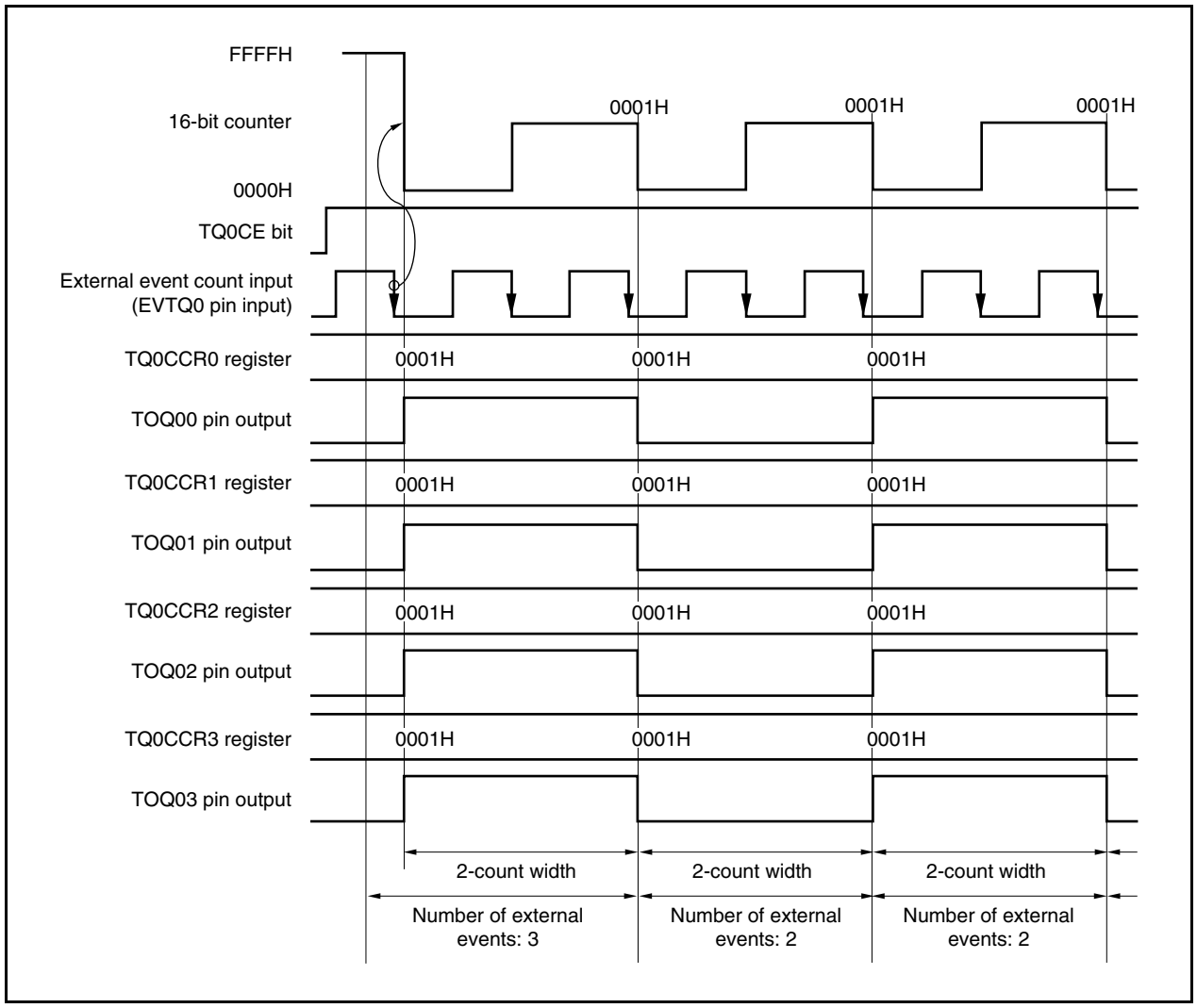
<R> **(3) Operation by external event count input (EVTQ0)**

(a) Operation

To count the 16-bit counter at the valid edge of external event count input (EVTQ0) in the interval timer mode, clear the 16-bit counter from FFFFH to 0000H at the valid edge of the first external event count input after the TQ0CE bit is set from 0 to 1.

When both the TQ0CCR0 and TQ0CCRb registers are set to 0001H, the output of the TOQ00 and TOQ0b pins is inverted each time the 16-bit counter counts twice (b = 1 to 3).

The TQ0CTL0.TQ0EEE bit can be set to 1 in the interval timer mode only when the timer output (TOQ00, TOQ0b) is used with the external event count input.



7.6.2 External event count mode (TQ0MD2 to TQ0MD0 bits = 001)

This mode is valid only in TMQ0.

In the external event count mode, the valid edge of the external event count input (EVTQ0) is counted when the TQ0CTL0.TQ0CE bit is set to 1, and an interrupt request signal (INTTQ0CC0) is generated each time the specified number of edges set by the TQ0CCR0 register have been counted. The TOQ00 to TOQ03 pins cannot be used.

<R> When using the TOQ00 to TOQ03 pins in the external event count input mode, set the TQ0CTL1.TQ0EEE bit to 1 in the interval timer mode (see **7.6.1 (3) Operation by external event count input (EVTQ0)**).

The TQ0CCR1 to TQ0CCR3 registers are not used in the external event count mode.

Caution In the external event count mode, the TQ0CCR0 to TQ0CCR3 registers must not be cleared to 0000H.

Figure 7-15. Configuration in External Event Count Mode

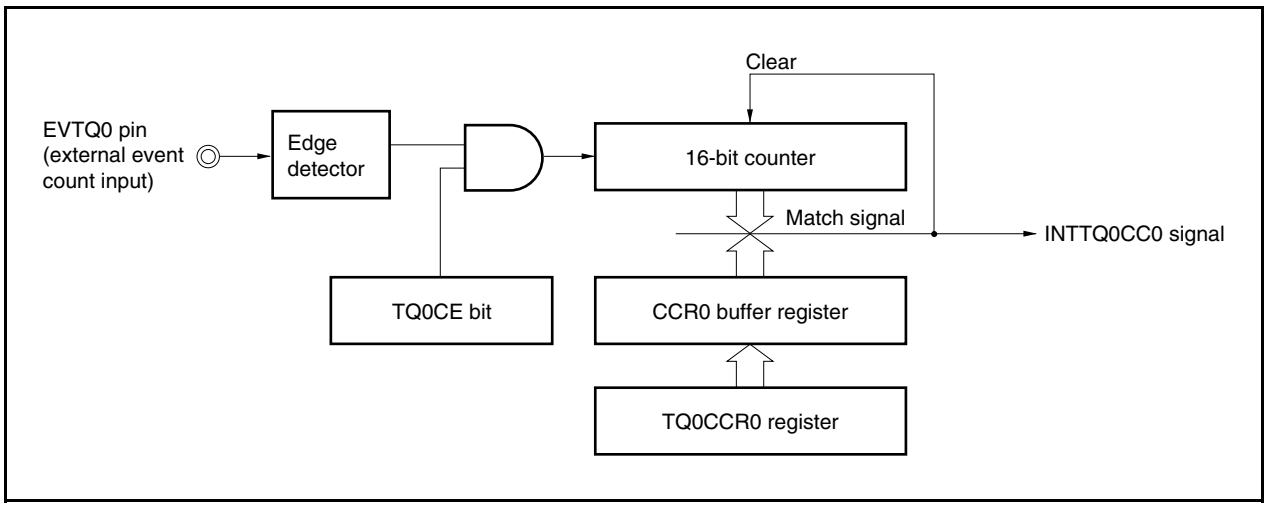
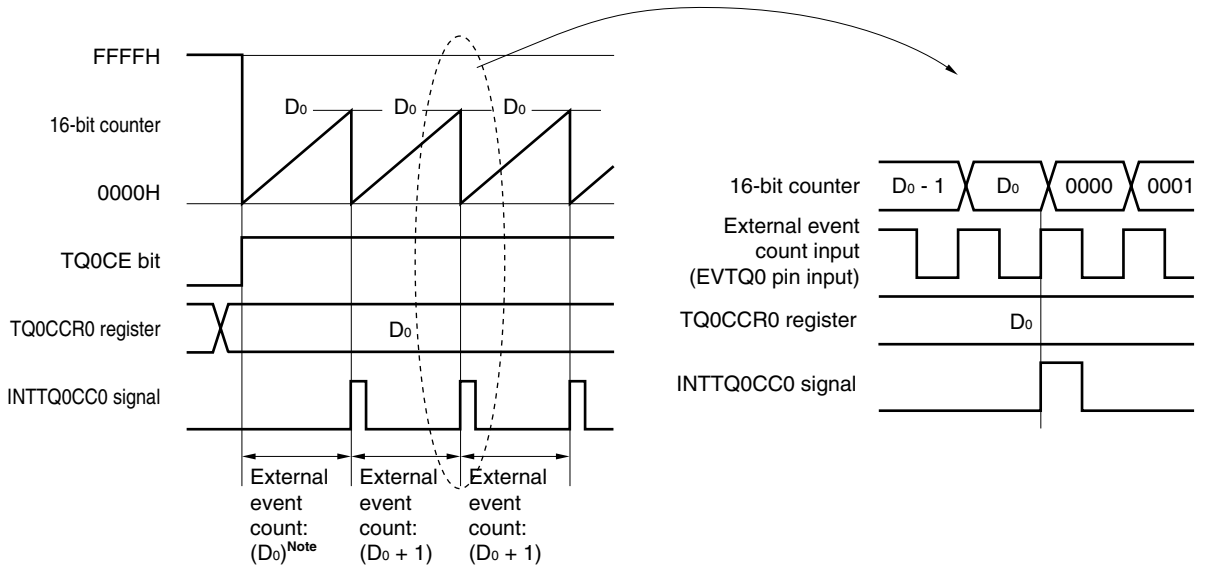


Figure 7-16. Basic Timing in External Event Count Mode



Note In the external event count mode, when the TQ0CTL0.TQ0CE bit is set to 1 (operation starts), the 16-bit counter is cleared from FFFFH to 0000H at the same time. The first count operation starts from 0001H every time the valid edge of the external event count input is detected. Therefore, the count of the first count operation is one number smaller than the count of second or subsequent count operation.

Remark This figure shows the basic timing when the rising edge is specified as the valid edge of the external event count input.

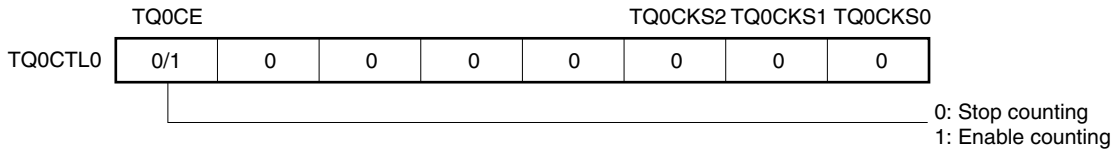
When the TQ0CE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H. The counter counts each time the valid edge of external event count input is detected. Additionally, the set value of the TQ0CCR0 register is transferred to the CCR0 buffer register.

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 (INTTQ0CC0) is generated.

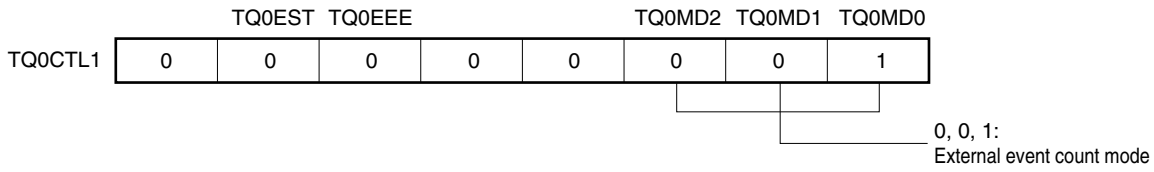
The INTTQ0CC0 signal is generated for the first time when the valid edge of the external event count input has been detected "value set to TQ0CCR0 register" times. After that, the INTTQ0CC0 signal is generated each time the valid edge of the external event count has been detected "value set to TQ0CCR0 register + 1" times.

Figure 7-17. Register Setting for Operation in External Event Count Mode (1/2)

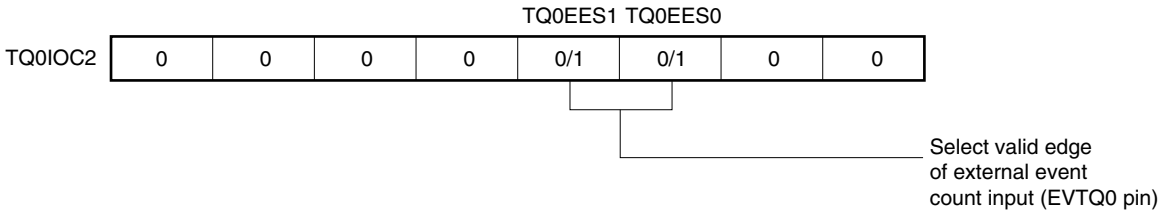
(a) TMQ0 control register 0 (TQ0CTL0)



(b) TMQ0 control register 1 (TQ0CTL1)



(c) TMQ0 I/O control register 2 (TQ0IOC2)



(d) TMQ0 counter read buffer register (TQ0CNT)

The count value of the 16-bit counter can be read by reading the TQ0CNT register.

(e) TMQ0 capture/compare register 0 (TQ0CCR0)

If the TQ0CCR0 register is set to D_0, the count is cleared when the number of external event counts has reached (D_0) and the first compare match interrupt request signal (INTTQ0CC0) is generated. The second compare match interrupt request signal (INTTQ0CC0) is generated when the number of external event counts has reached ($D_0 + 1$).

Figure 7-17. Register Setting for Operation in External Event Count Mode (2/2)

(f) TMQ0 capture/compare registers 1 to 3 (TQ0CCR1 to TQ0CCR3)

The TQ0CCR1 to TQ0CCR3 registers are not used in the external event count mode. However, the set values of the TQ0CCR1 to TQ0CCR3 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 (INTTQ0CC1 to INTTQ0CC3) are generated.

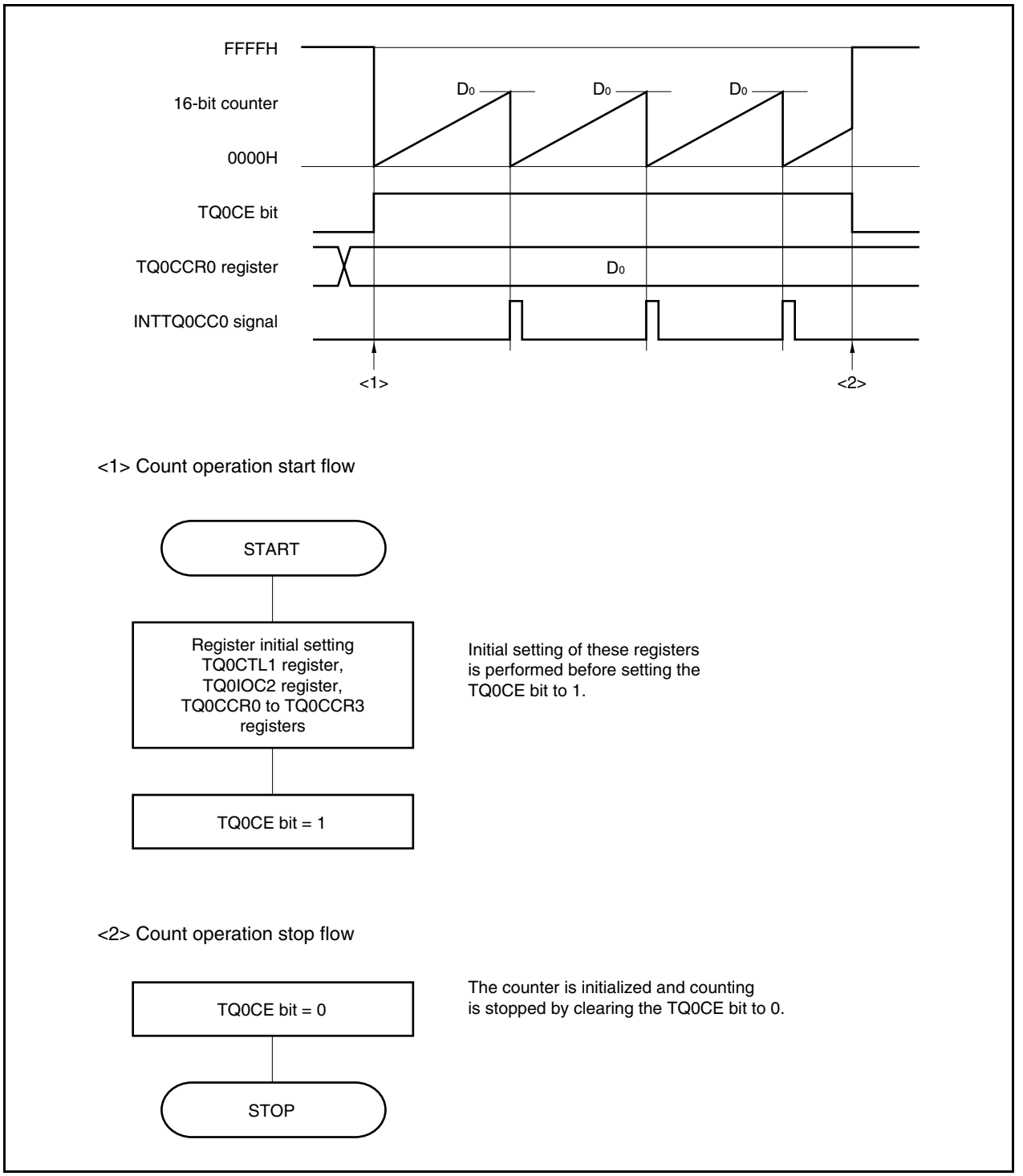
When the TQ0CCR1 to TQ0CCR3 registers are not used, it is recommended to set their values to FFFFH. Also mask the registers by the interrupt mask flags (TQ0CCIC1.TQ0CCMK1 to TQ0CCIC3.TQ0CCMK3).

<R> **Caution Set the TQ0IOC0 register to 00H.**

Remark TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the external event count mode.

(1) External event count mode operation flow

Figure 7-18. Flow of Software Processing in External Event Count Mode



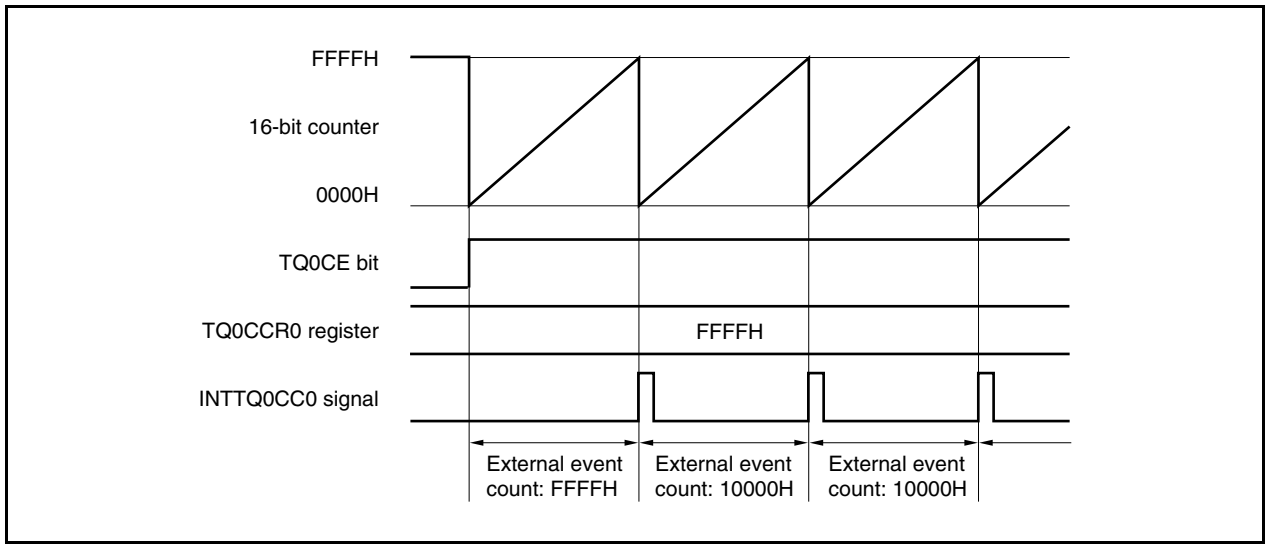
(2) Operation timing in external event count mode

Cautions 1. In the external event count mode, the TQ0CCR0 to TQ0CCR3 registers must not be cleared to 0000H.

<R> **2. In the external event count mode, use of the timer output (TOQ00 to TOQ03) is disabled. If using timer output (TOQ00 to TOQ03) with external event count input (EVTQ0), set the interval timer mode, and enable the count clock operation with the external event count input (TQ0CTL1.TQ0EEE bit = 1) (see 7.6.1 (3) Operation by external event count input (EVTQ0)).**

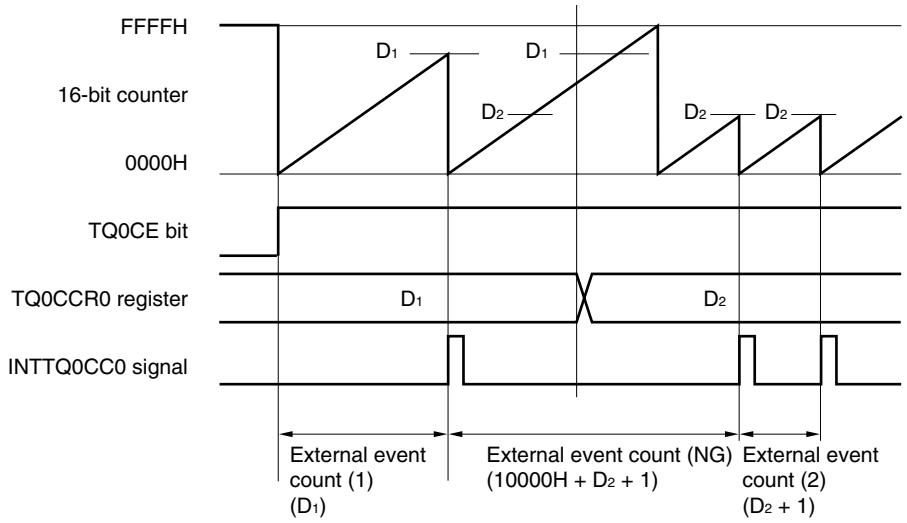
(a) Operation if TQ0CCR0 register is set to FFFFH

If the TQ0CCR0 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 INTTQ0CC0 signal is generated. At this time, the TQ0OPT0.TQ0OVF bit is not set.



(b) Notes on rewriting the TQ0CCR0 register

If the value of the TQ0CCR0 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.



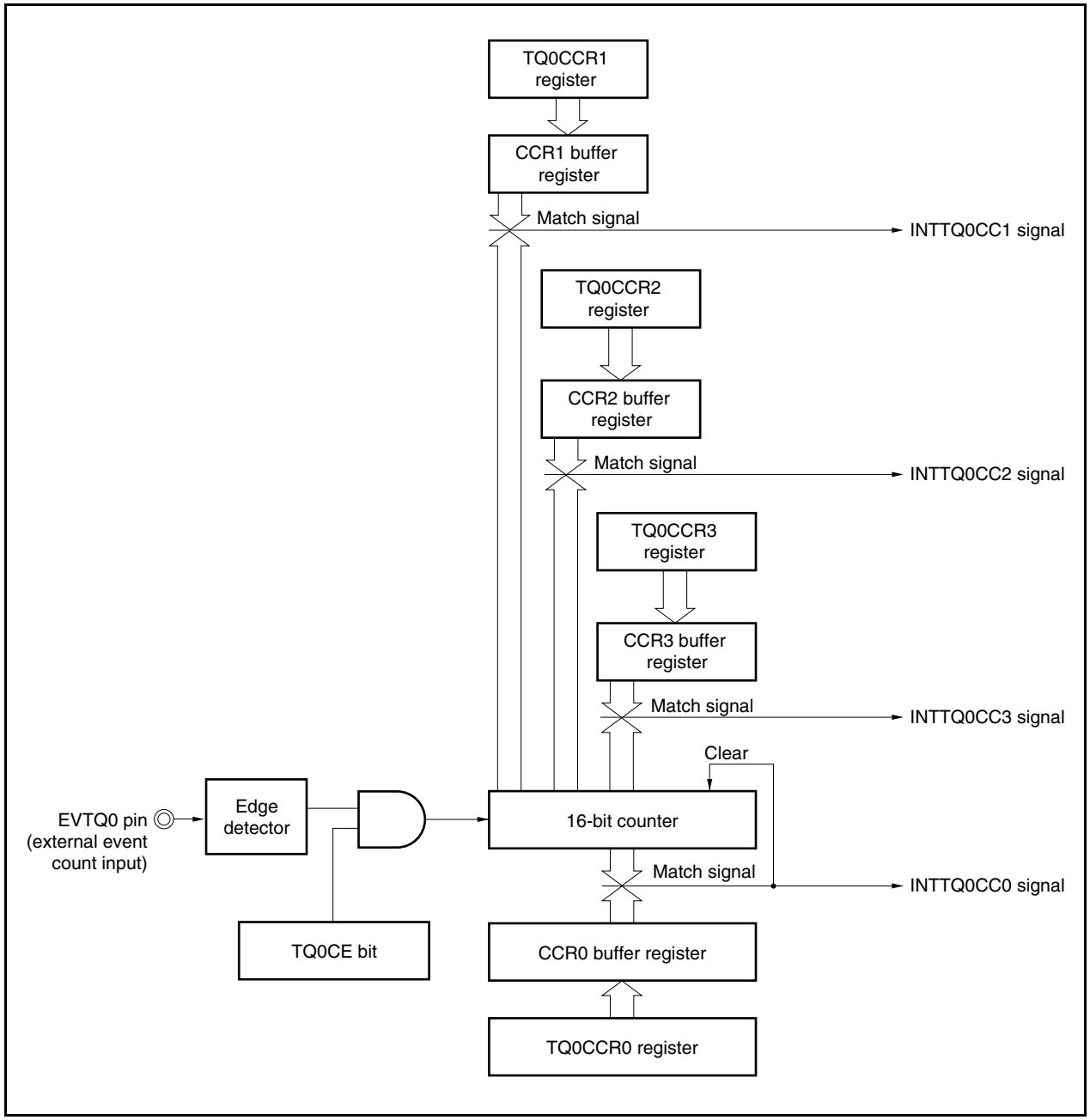
If the value of the TQ0CCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TQ0CCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter 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 INTTQ0CC0 signal is generated.

Therefore, the INTTQ0CC0 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".

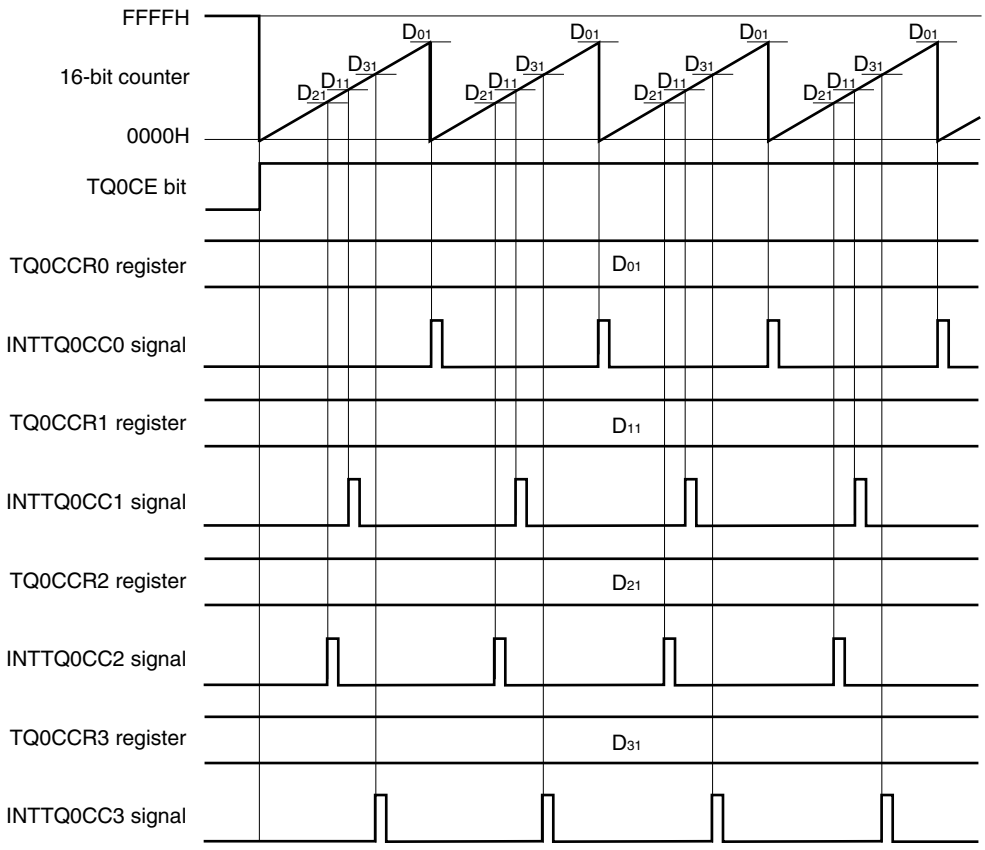
(c) Operation of TQ0CCR1 to TQ0CCR3 registers

Figure 7-19. Configuration of TQ0CCR1 to TQ0CCR3 Registers



If the set value of the TQ0CCRb register is smaller than the set value of the TQ0CCR0 register, the INTTQ0CCb signal is generated once per cycle.

Figure 7-20. Timing Chart When $D_{01} \geq D_{b1}$

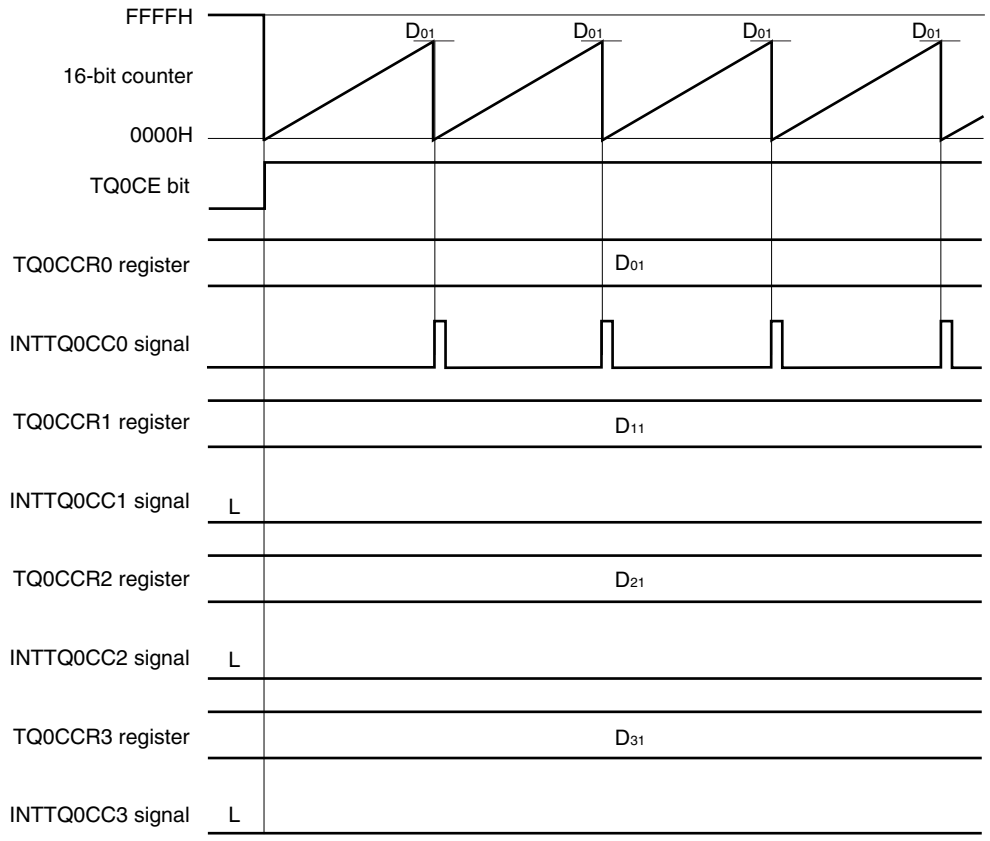


Remark b = 1 to 3

If the set value of the TQ0CCRb register is greater than the set value of the TQ0CCR0 register, the INTTQ0CCb signal is not generated because the count value of the 16-bit counter and the value of the TQ0CCRb register do not match.

When the TQ0CCRb register is not used, it is recommended to set its value to FFFFH.

Figure 7-21. Timing Chart When $D_{01} < D_{b1}$



Remark b = 1 to 3

7.6.3 External trigger pulse output mode (TQ0MD2 to TQ0MD0 bits = 010)

This mode is valid only in TMQ0.

In the external trigger pulse output mode, 16-bit timer/event counter Q waits for a trigger when the TQ0CTL0.TQ0CE bit is set to 1. When a software trigger is generated, 16-bit timer/event counter Q starts counting, and outputs a PWM waveform from the TOQ01 to TOQ03 pins. A PWM waveform with a duty factor of 50% whose half cycle is the set value of the TQ0CCR0 register + 1 can also be output from the TOQ00 pin.

External trigger input pins are not provided.

Figure 7-22. Configuration in External Trigger Pulse Output Mode

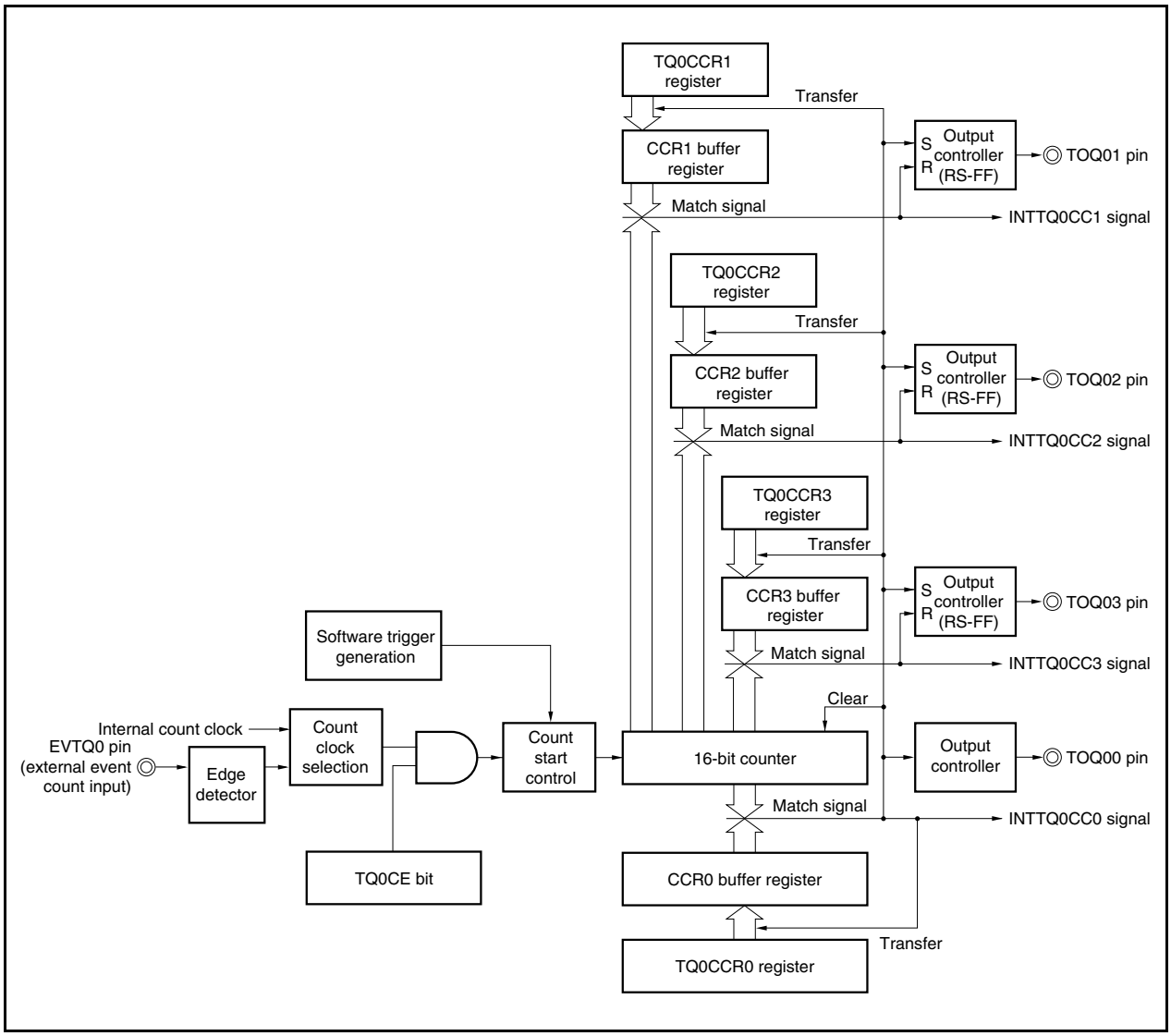
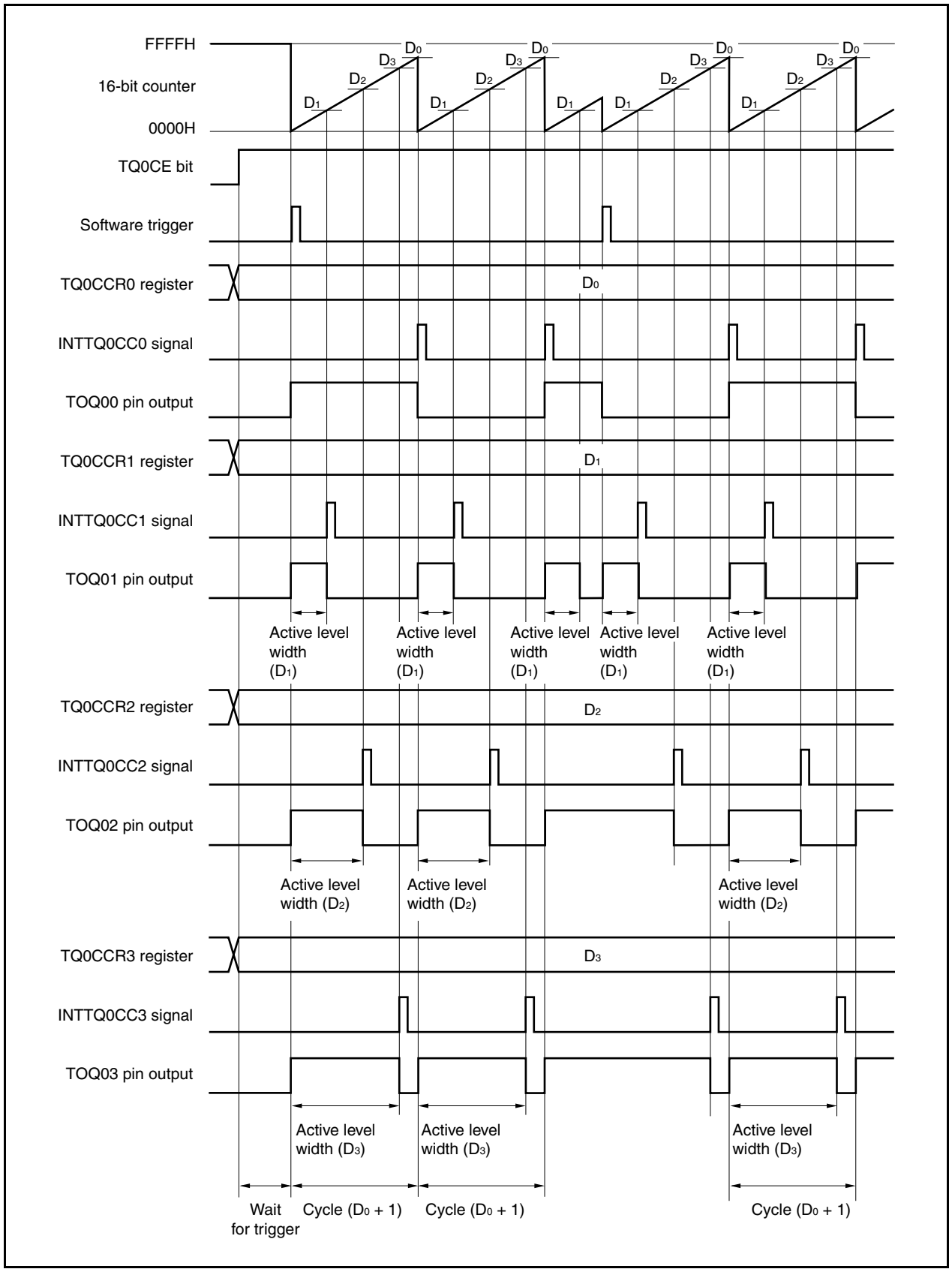


Figure 7-23. Basic Timing in External Trigger Pulse Output Mode



16-bit timer/event counter Q waits for a trigger when the TQ0CE 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 TOQ0b pin. If the trigger is generated again while the counter is operating, the counter is cleared to 0000H and <R> restarted. (The output of the TOQ00 pin is inverted. The TOQ0b pin outputs 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 TQ0CCRb register) × Count clock cycle
Cycle = (Set value of TQ0CCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TQ0CCRb register)/(Set value of TQ0CCR0 register + 1)

The compare match interrupt request signal INTTQ0CC0 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 INTTQ0CCb is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.

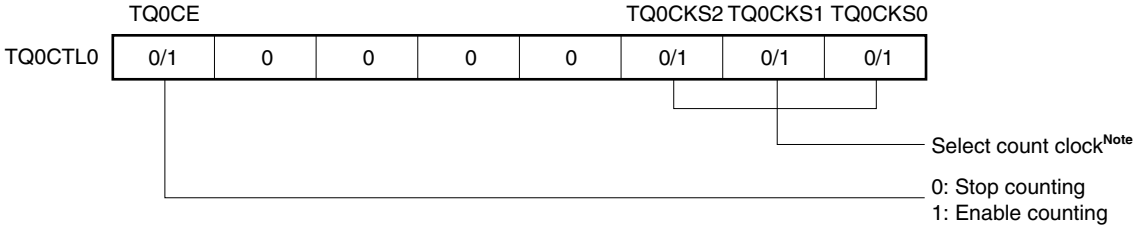
The value set to the TQ0CCRa register is transferred to the CCRa buffer register when the count value of the 16-bit counter matches the value of the CCRa buffer register and the 16-bit counter is cleared to 0000H.

Only setting the software trigger (TQ0CTL1.TQ0EST bit) to 1 is used as the trigger.

Remark a = 0 to 3
b = 1 to 3

Figure 7-24. Setting of Registers in External Trigger Pulse Output Mode (1/3)

(a) TMQ0 control register 0 (TQ0CTL0)



Note The setting is invalid when the TQ0CTL1.TQ0EEE bit = 1.

Figure 7-24. Setting of Registers in External Trigger Pulse Output Mode (2/3)

(b) TMQ0 control register 1 (TQ0CTL1)

	TQ0EST	TQ0EEE			TQ0MD2	TQ0MD1	TQ0MD0
TQ0CTL1	0	0/1	0/1	0	0	0	1	0

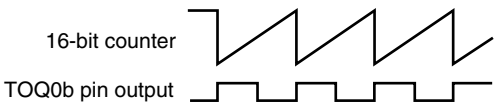
- 0, 1, 0: External trigger pulse output mode
- 0: Operate on count clock selected by TQ0CKS0 to TQ0CKS2 bits
  1: Count with external event count input signal
- Generate software trigger when 1 is written

(c) TMQ0 I/O control register 0 (TQ0IOC0)

	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0
TQ0IOC0	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

- 0: Disable TOQ00 pin output
  1: Enable TOQ00 pin output
- Setting of TOQ00 pin output level in external trigger waiting state
  0: Low level
  1: High level
- 0: Disable TOQ01 pin output
  1: Enable TOQ01 pin output
- Setting of TOQ01 pin output level in external trigger waiting state
  0: Low level
  1: High level
- 0: Disable TOQ02 pin output
  1: Enable TOQ02 pin output
- Setting of TOQ02 pin output level in external trigger waiting state
  0: Low level
  1: High level
- 0: Disable TOQ03 pin output
  1: Enable TOQ03 pin output
- Setting of TOQ03 pin output level in external trigger waiting state
  0: Low level
  1: High level

- When TQ0OLb bit = 0



- When TQ0OLb bit = 1

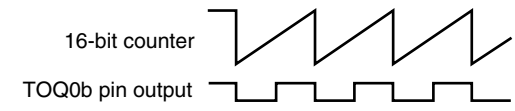
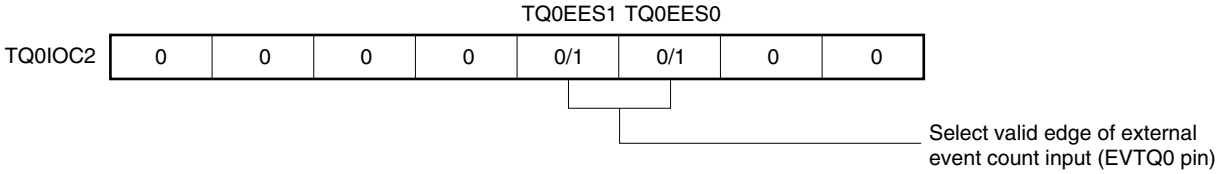


Figure 7-24. Setting of Registers in External Trigger Pulse Output Mode (3/3)

(d) TMQ0 I/O control register 2 (TQ0IOC2)



(e) TMQ0 counter read buffer register (TQ0CNT)

The value of the 16-bit counter can be read by reading the TQ0CNT register.

(f) TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3)

If D_0 is set to the TQ0CCR0 register, D_1 to the TQ0CCR1 register, D_2 to the TQ0CCR2 register, and D_3, to the TQ0CCR3 register, the cycle and active level of the PWM waveform are as follows.

PWM waveform cycle = $(D_0 + 1) \times$ Count clock cycle
TOQ01 pin PWM waveform active level width = $D_1 \times$ Count clock cycle
TOQ02 pin PWM waveform active level width = $D_2 \times$ Count clock cycle
TOQ03 pin PWM waveform active level width = $D_3 \times$ Count clock cycle

Remarks 1. TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the external trigger pulse output mode.

2. b = 1 to 3

(1) Operation flow in external trigger pulse output mode

Figure 7-25. Software Processing Flow in External Trigger Pulse Output Mode (1/2)

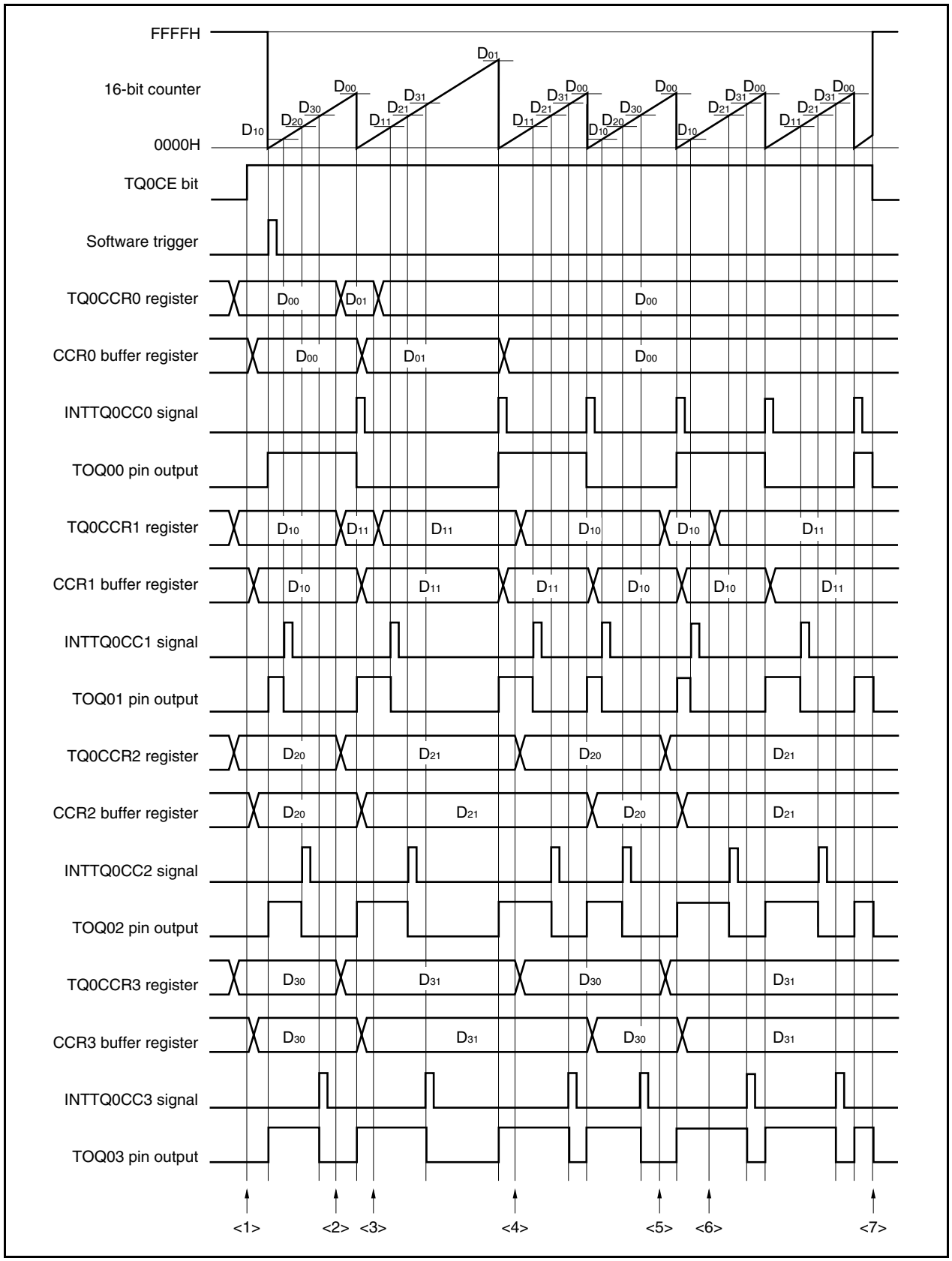
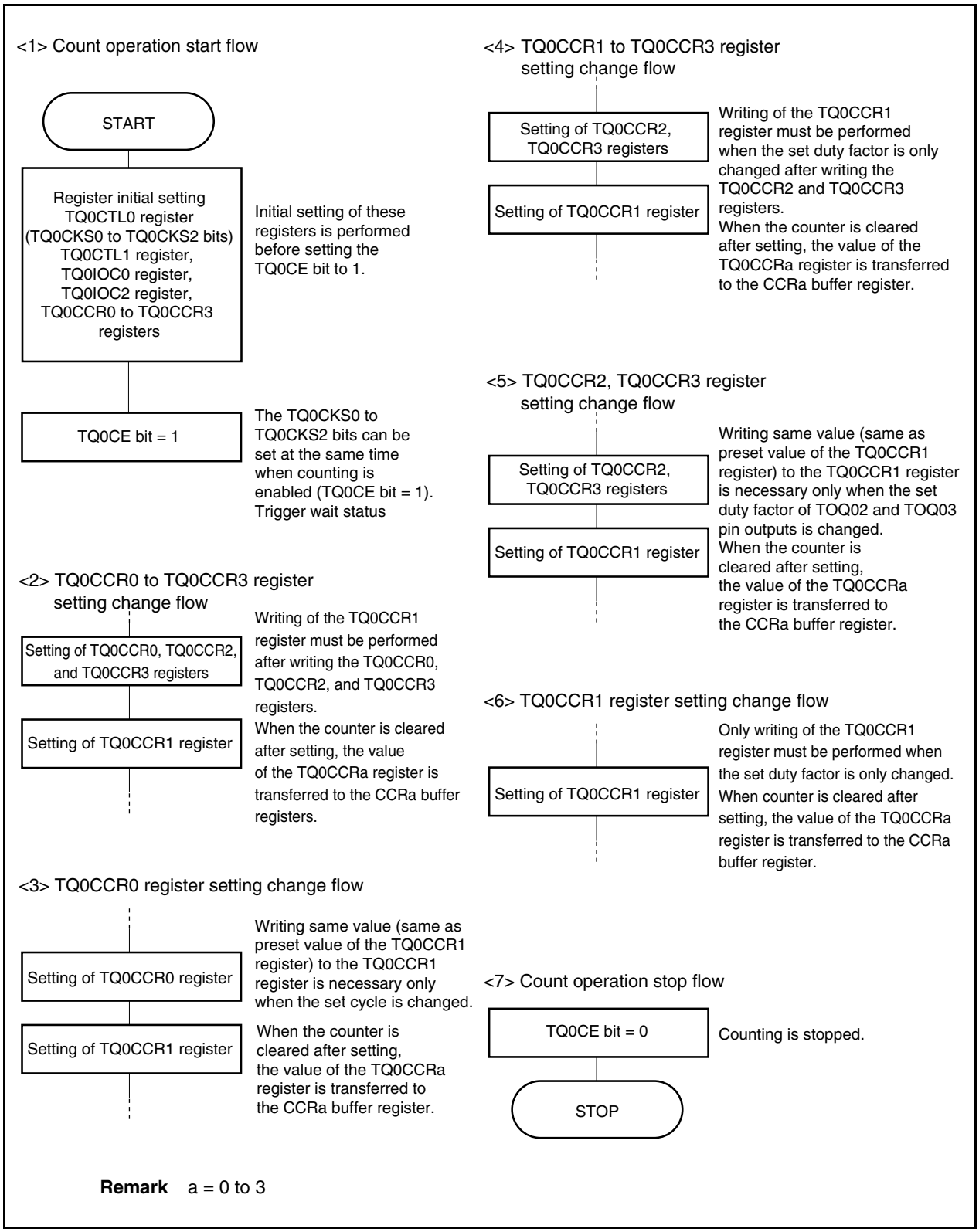


Figure 7-25. Software Processing Flow in External Trigger Pulse Output Mode (2/2)



Remark a = 0 to 3

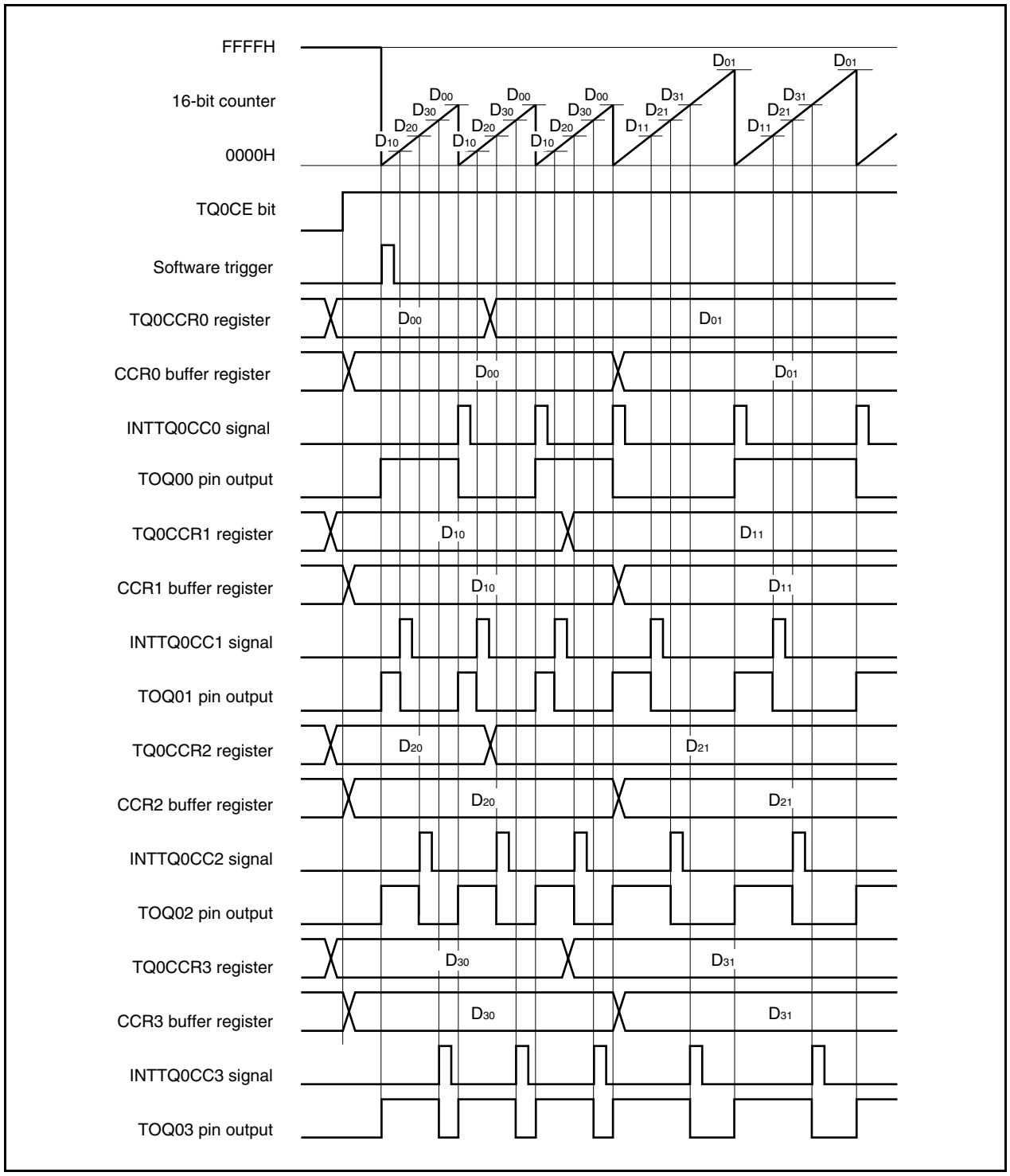
(2) External trigger pulse output mode operation timing

(a) Note on changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TQ0CCR1 register last.

Rewrite the TQ0CCRb register after writing the TQ0CCR1 register after the INTTQ0CC0 signal is detected.

Remark b = 1 to 3



In order to transfer data from the TQ0CCRa register to the CCRa buffer register, the TQ0CCR1 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 TQ0CCR0 register, set the active level width to the TQ0CCR2 and TQ0CCR3 registers, and then set an active level to the TQ0CCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TQ0CCR0 register, and then write the same value (same as preset value of the TQ0CCR1 register) to the TQ0CCR1 register.

To change only the active level width (duty factor) of the PWM waveform, first set an active level to the TQ0CCR2 and TQ0CCR3 registers and then set an active level to the TQ0CCR1 register.

To change only the active level width (duty factor) of the PWM waveform output by the TOQ01 pin, only the TQ0CCR1 register has to be set.

To change only the active level width (duty factor) of the PWM waveform output by the TOQ02 and TOQ03 pins, first set an active level width to the TQ0CCR2 and TQ0CCR3 registers, and then write the same value (same as preset value of the TQ0CCR1 register) to the TQ0CCR1 register.

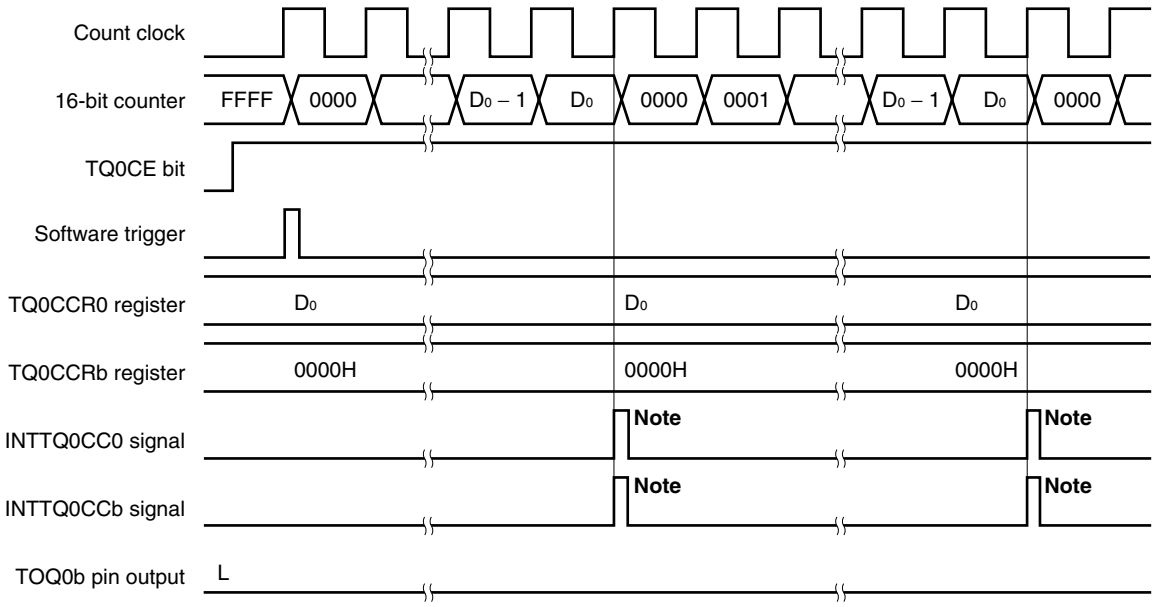
After data is written to the TQ0CCR1 register, the value written to the TQ0CCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TQ0CCR0 to TQ0CCR3 registers again after writing the TQ0CCR1 register once, do so after the INTTQ0CC0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because timing of transferring data from the TQ0CCRa register to the CCRa buffer register conflicts with writing the TQ0CCRa register.

Remark a = 0 to 3

(b) 0%/100% output of PWM waveform

To output a 0% waveform, set the TQ0CCRb register to 0000H. The 16-bit counter is cleared to 0000H and the INTTQ0CC0 and INTTQ0CCb 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.

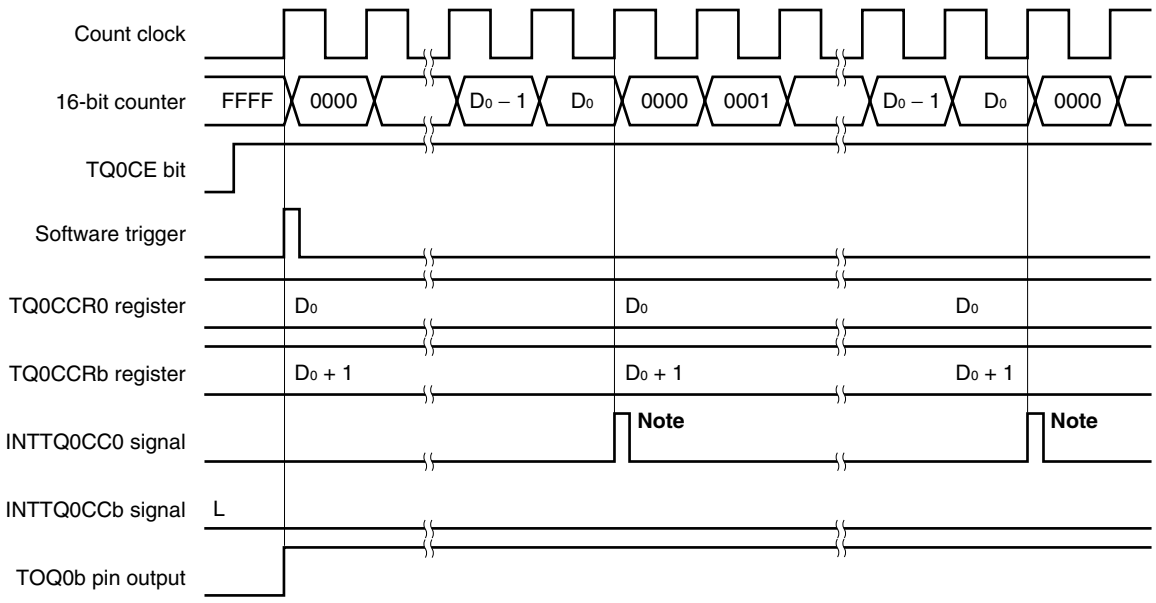


Note Actually, the timing is delayed by one operating clock (f_{XX}).

Remark b = 1 to 3

To output a 100% waveform, set a value of (set value of TQ0CCR0 register + 1) to the TQ0CCRb register. If the set value of the TQ0CCR0 register is FFFFH, 100% output cannot be produced.

<R>

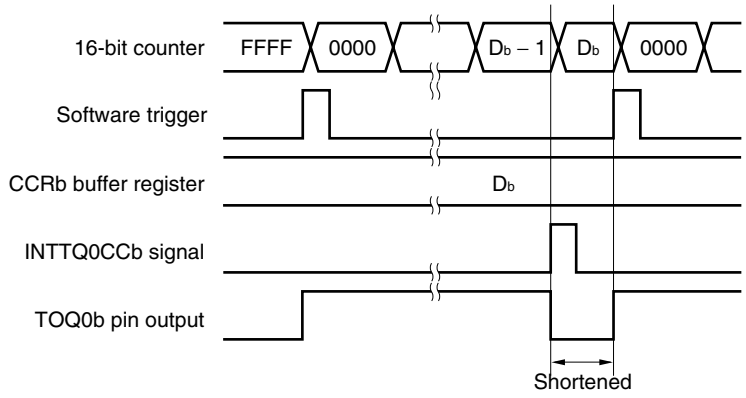


Note Actually, the timing is delayed by one operating clock (fxx).

Remark b = 1 to 3

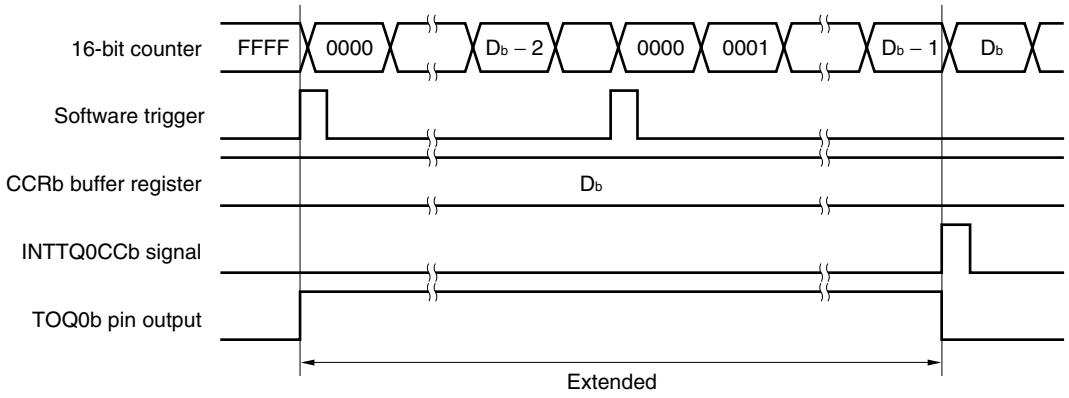
(c) Conflict between trigger detection and match with CCRb buffer register

If the trigger is detected immediately after the INTTQ0CCb signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TOQ0b pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark b = 1 to 3

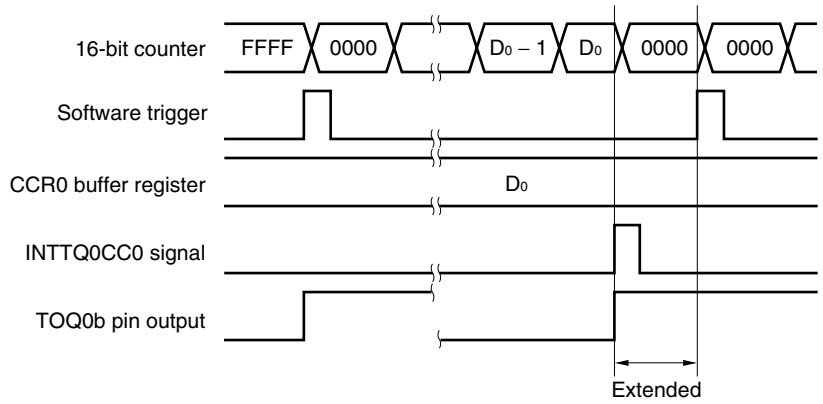
If the trigger is detected immediately before the INTTQ0CCb signal is generated, the INTTQ0CCb signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOQ0b pin remains active. Consequently, the active period of the PWM waveform is extended.



Remark b = 1 to 3

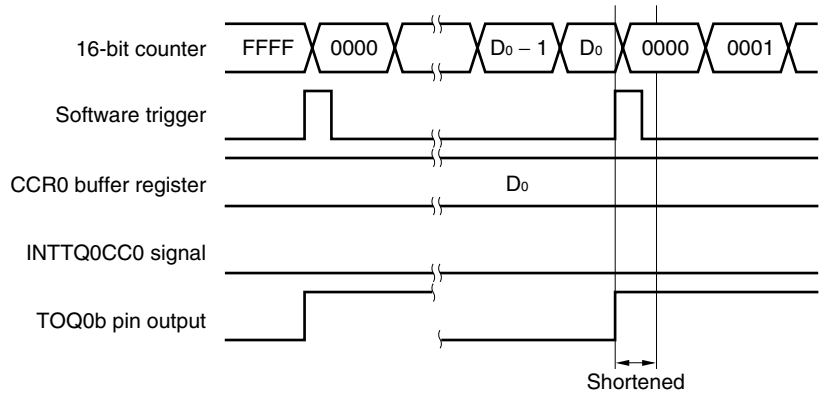
(d) Conflict between trigger detection and match with CCR0 buffer register

If the trigger is detected immediately after the INTTQ0CC0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOQ0b pin is extended by time from generation of the INTTQ0CC0 signal to trigger detection.



Remark b = 1 to 3

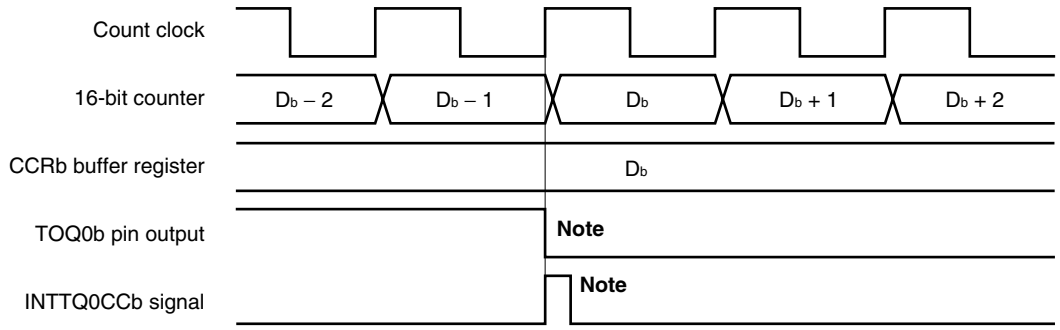
If the trigger is detected immediately before the INTTQ0CC0 signal is generated, the INTTQ0CC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOQ0b pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark b = 1 to 3

(e) Generation timing of compare match interrupt request signal (INTTQ0CCb)

The timing of generation of the INTTQ0CCb signal in the external trigger pulse output mode differs from the timing of INTTQ0CCb signals in other mode; the INTTQ0CCb signal is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.



Note Actually, the timing is delayed by one operating clock (fxx).

Remark b = 1 to 3

Usually, the INTTQ0CCb 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 TOQ0b pin.

7.6.4 One-shot pulse output mode (TQ0MD2 to TQ0MD0 bits = 011)

This mode is valid only in TMQ0.

In the one-shot pulse output mode, 16-bit timer/event counter Q waits for a trigger when the TQ0CTL0.TQ0CE bit is set to 1. When a software trigger is generated, 16-bit timer/event counter Q starts counting, and outputs a one-shot pulse from the TOQ01 to TOQ03 pins. When the software trigger is used, the TOQ00 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 7-26. Configuration in One-Shot Pulse Output Mode

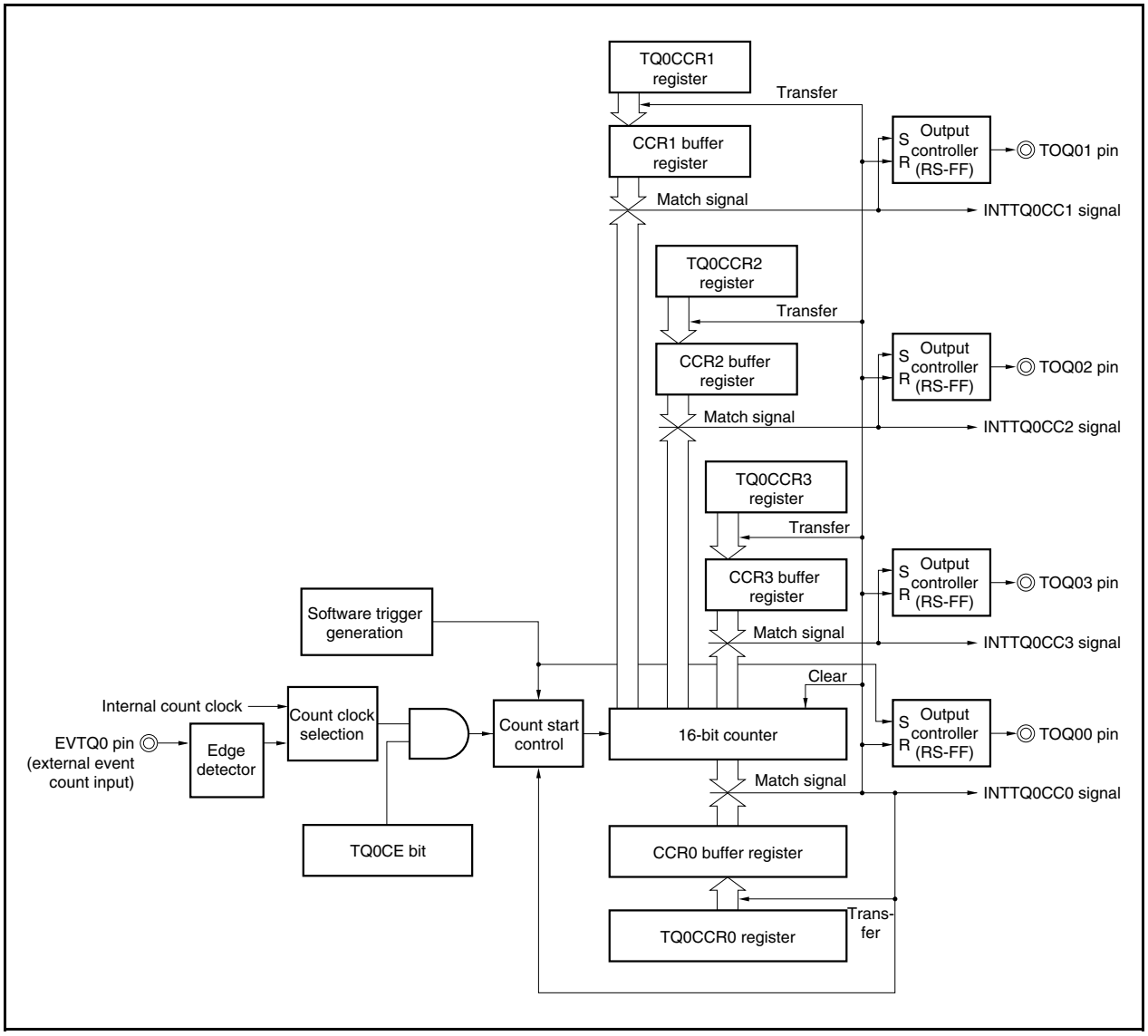
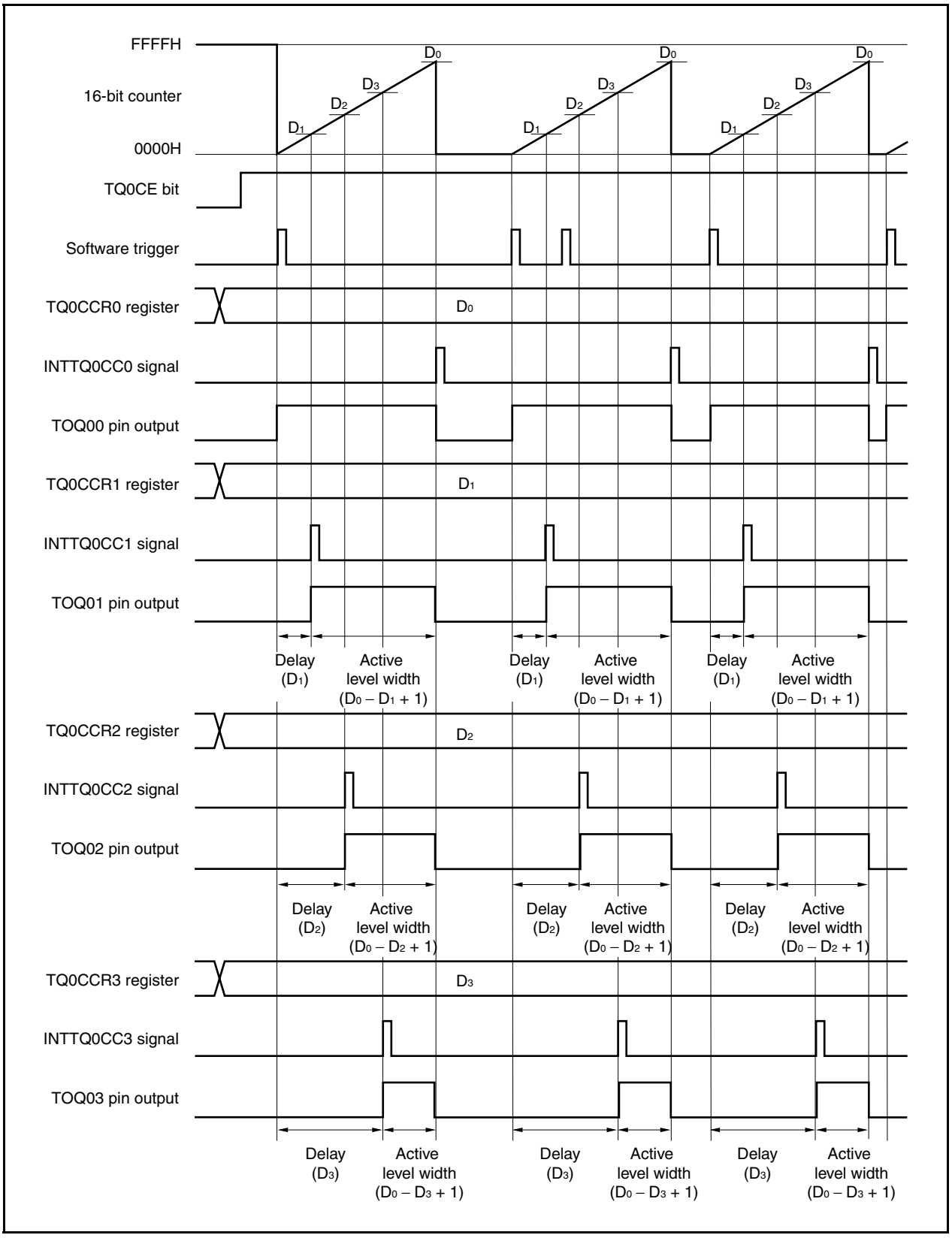


Figure 7-27. Basic Timing in One-Shot Pulse Output Mode



When the TQ0CE bit is set to 1, 16-bit timer/event counter Q 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 TOQ0b 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 TQ0CCRb register) × Count clock cycle

Active level width = (Set value of TQ0CCR0 register − Set value of TQ0CCRb register + 1) × Count clock cycle

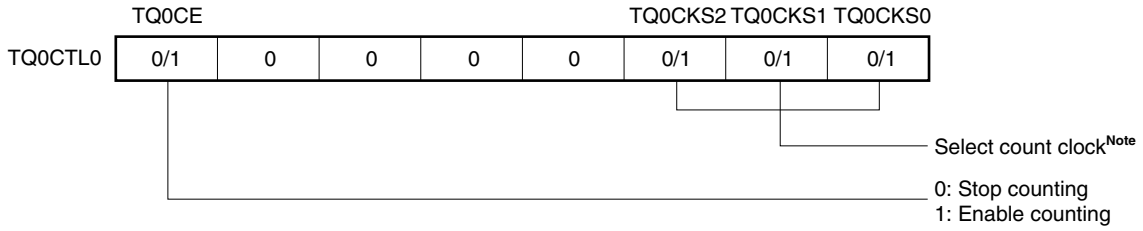
The compare match interrupt request signal INTTQ0CC0 is generated when the 16-bit counter counts after its count value matches the value of the CCR0 buffer register. The compare match interrupt request signal INTTQ0CCb is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.

Only setting the software trigger (TQ0CTL1.TQ0EST bit) to 1 is used as the trigger.

Remark b = 1 to 3

Figure 7-28. Register Setting in One-Shot Pulse Output Mode (1/3)

(a) TMQ0 control register 0 (TQ0CTL0)



Note The setting is invalid when the TQ0CTL1.TQ0EEE bit = 1.

(b) TMQ0 control register 1 (TQ0CTL1)

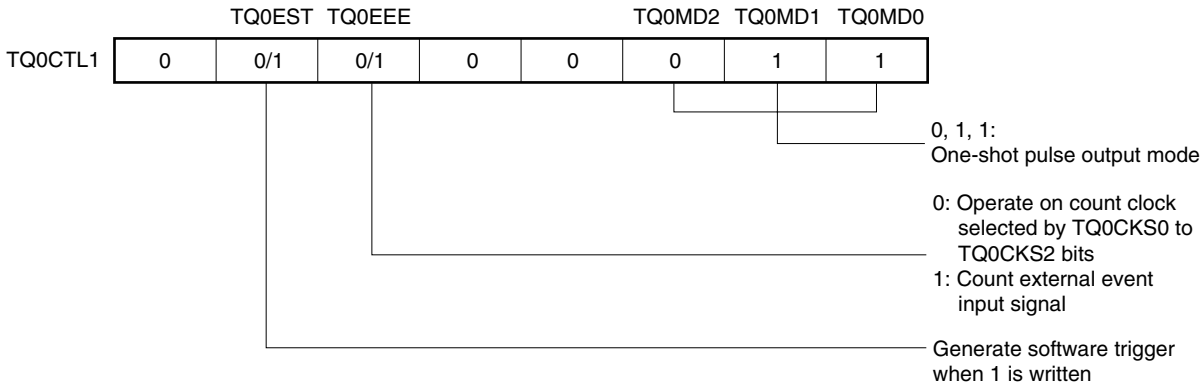
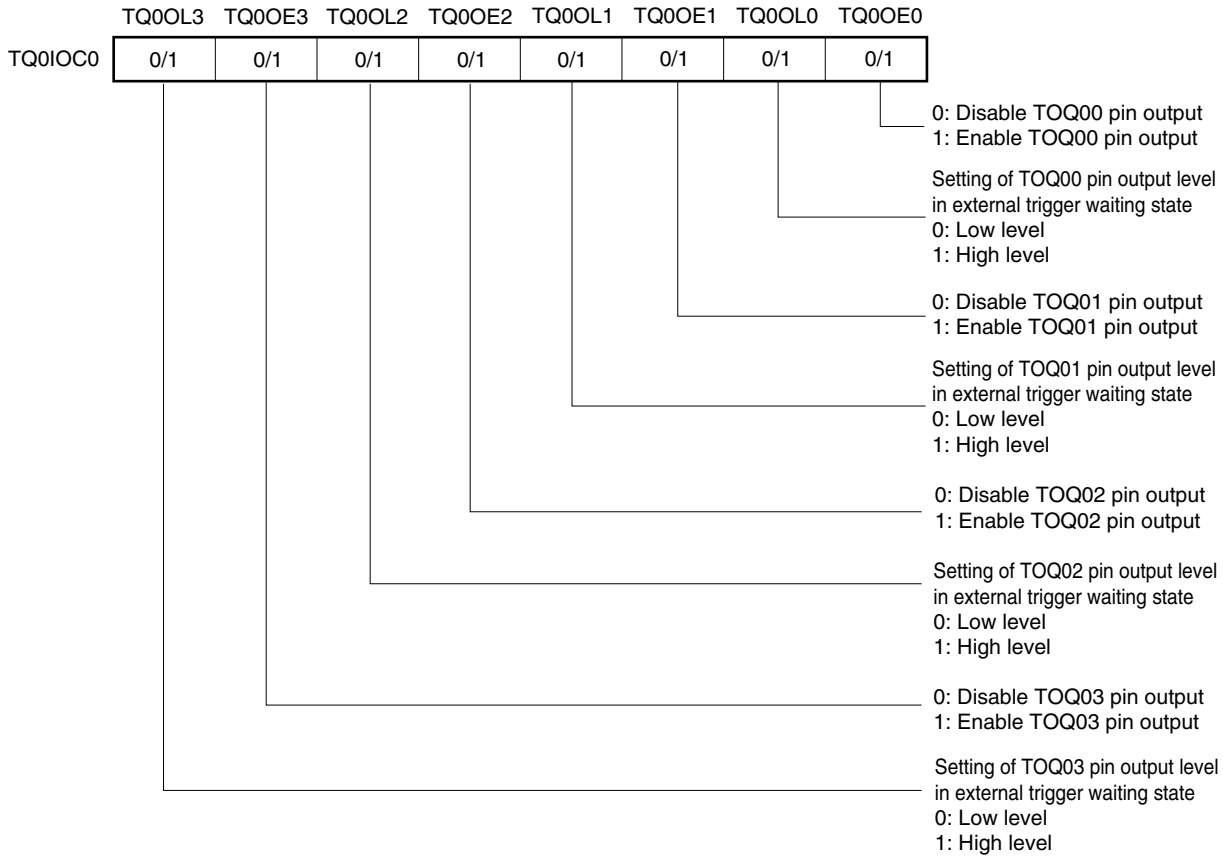
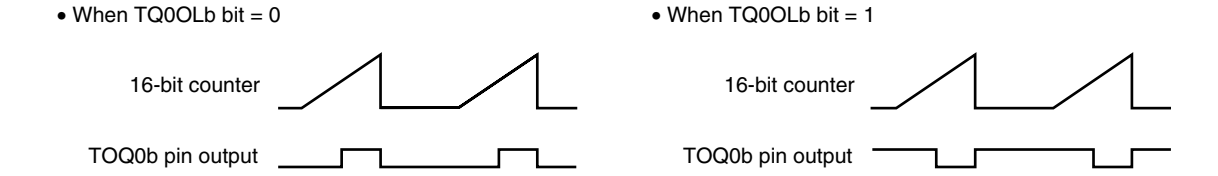


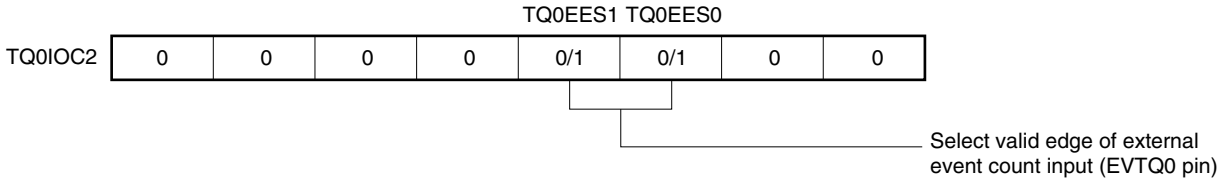
Figure 7-28. Register Setting in One-Shot Pulse Output Mode (2/3)

(c) TMQ0 I/O control register 0 (TQ0IOC0)





(d) TMQ0 I/O control register 2 (TQ0IOC2)



(e) TMQ0 counter read buffer register (TQ0CNT)

The value of the 16-bit counter can be read by reading the TQ0CNT register.

Figure 7-28. Register Setting in One-Shot Pulse Output Mode (3/3)

(f) TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3)

If D_0 is set to the TQ0CCR0 register and D_b to the TQ0CCRb register, the active level width and output delay period of the one-shot pulse are as follows.

<R> Active level width = $(D_0 - D_b + 1) \times$ Count clock cycle

Output delay period = $D_b \times$ Count clock cycle

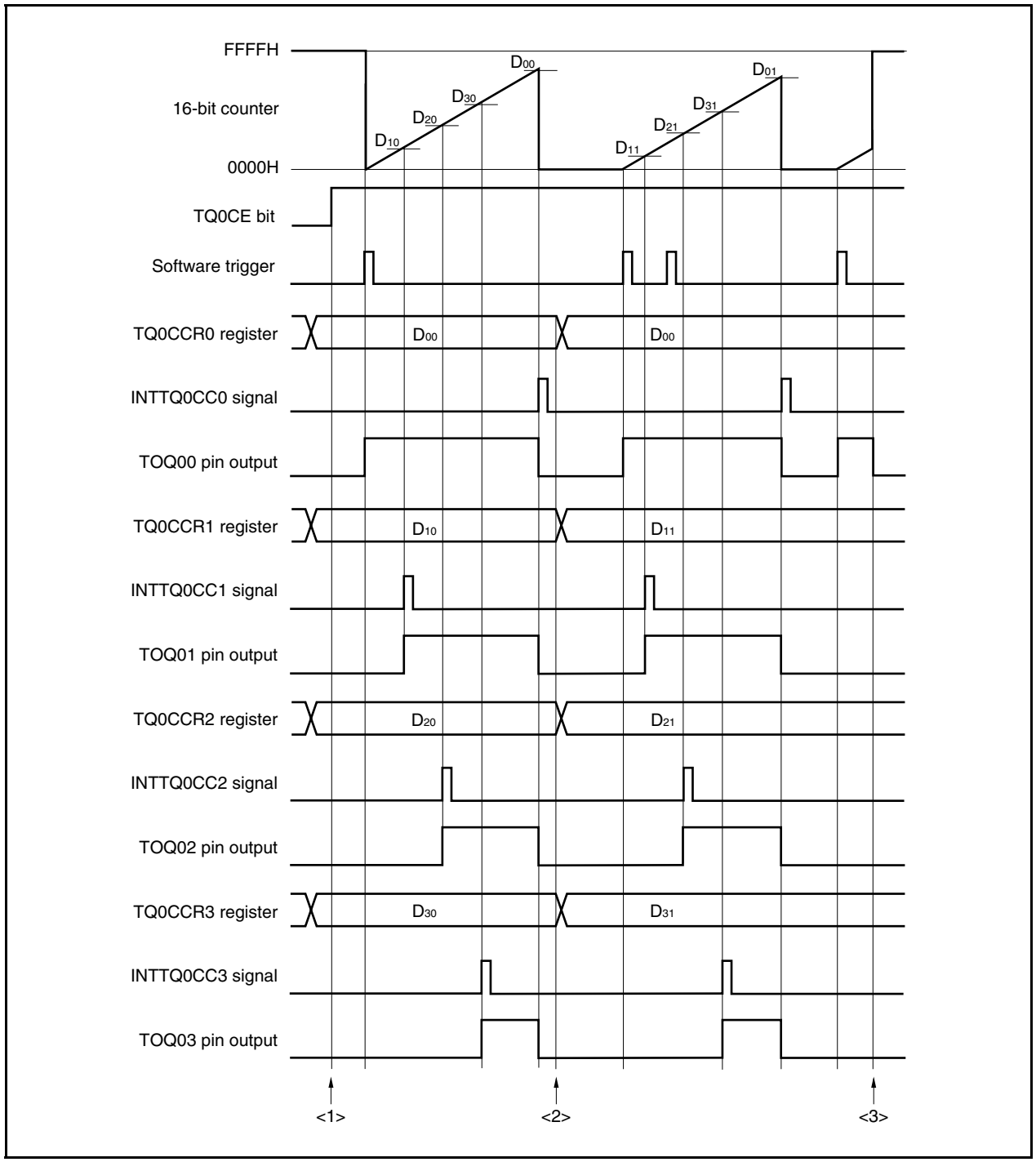
<R> **Caution One-shot pulses are not output even in the one-shot pulse output mode, if the value set in the TQ0CCRb register is greater than that set in the TQ0CCR0 register.**

Remarks 1. TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the one-shot pulse output mode.

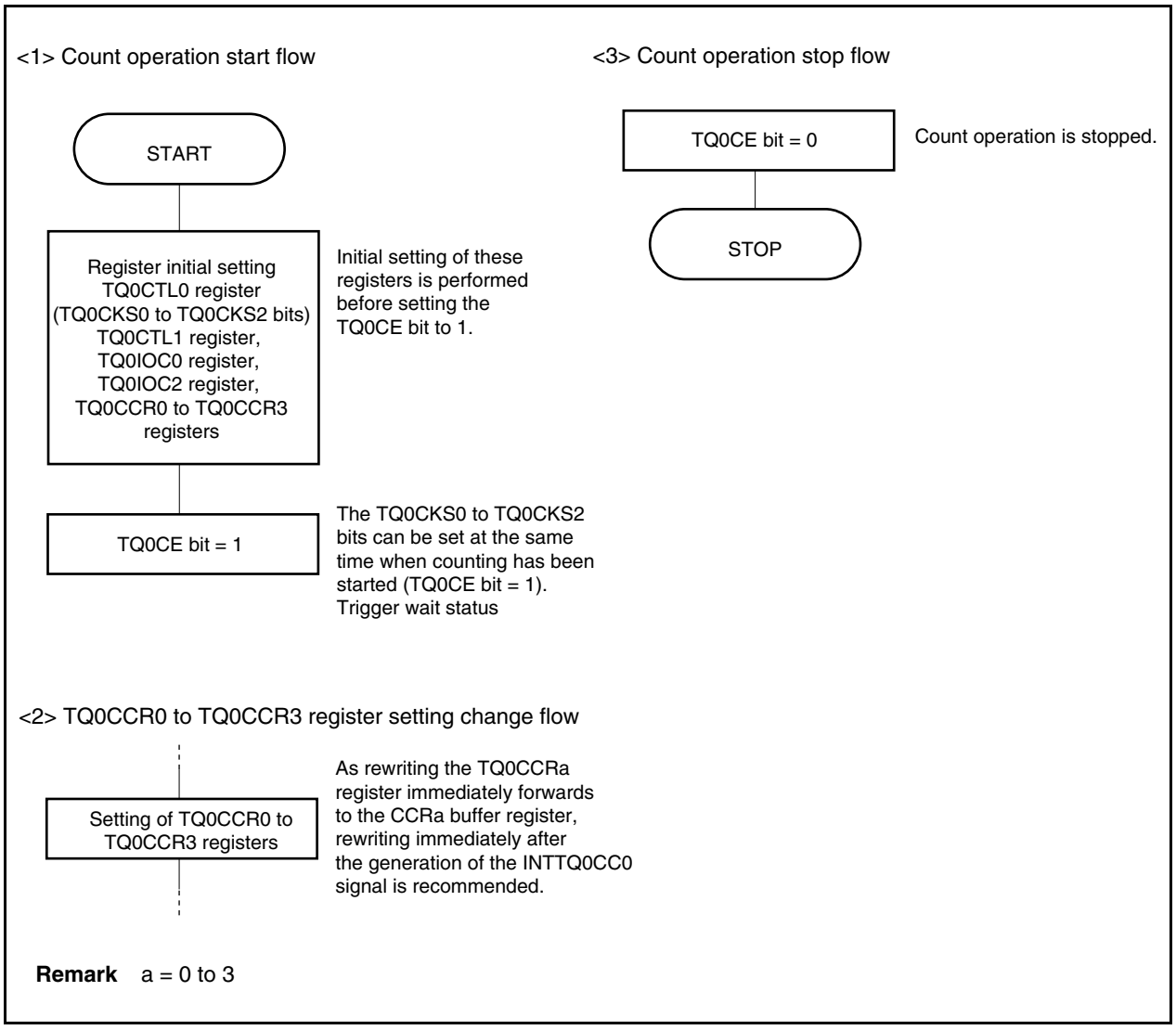
2. b = 1 to 3

(1) Operation flow in one-shot pulse output mode

<R> **Figure 7-29. Software Processing Flow in One-Shot Pulse Output Mode (1/2)**



<R> **Figure 7-29. Software Processing Flow in One-Shot Pulse Output Mode (2/2)**

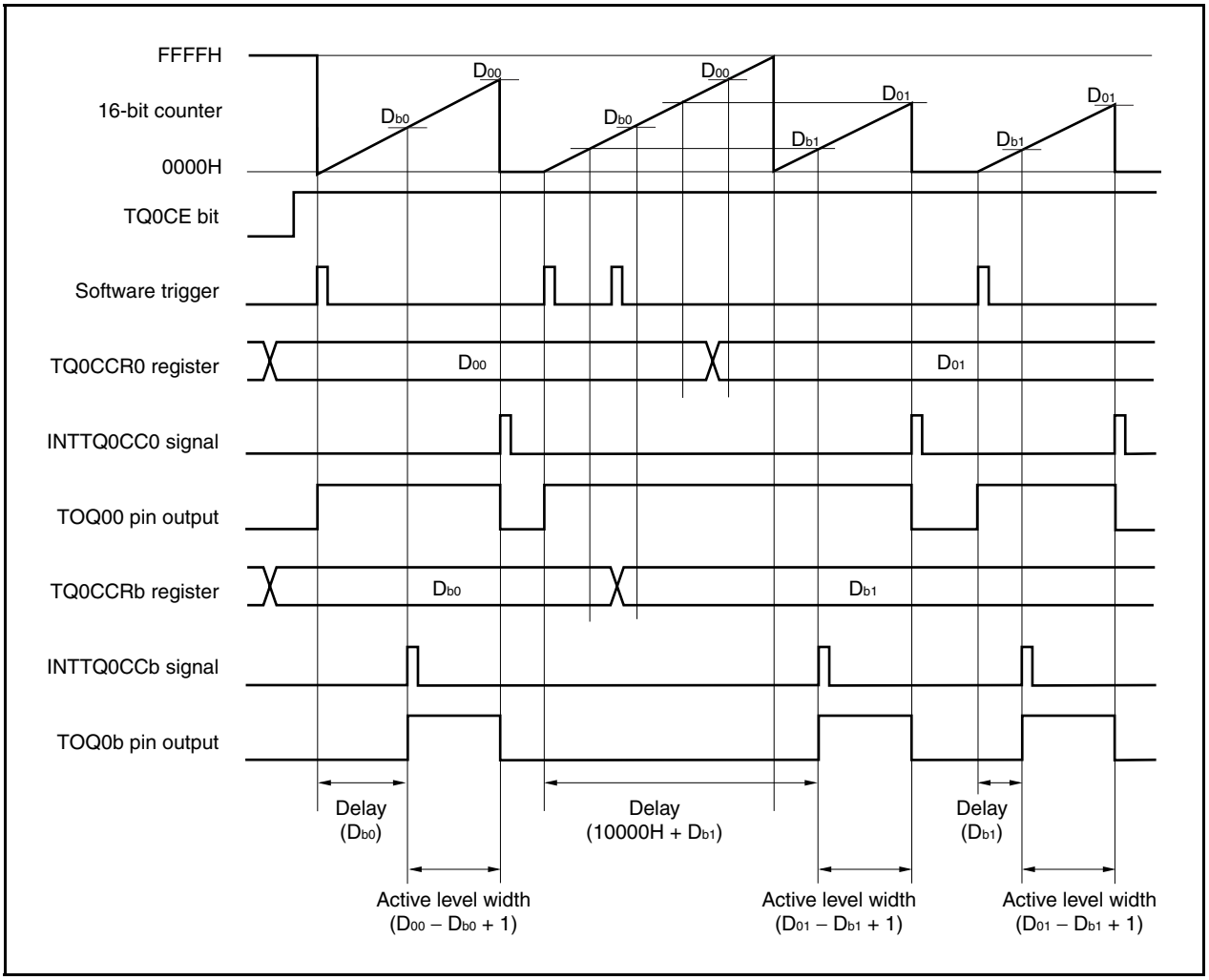


Remark a = 0 to 3

(2) Operation timing in one-shot pulse output mode

(a) Note on rewriting TQ0CCRa register

If the value of the TQ0CCRa 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.



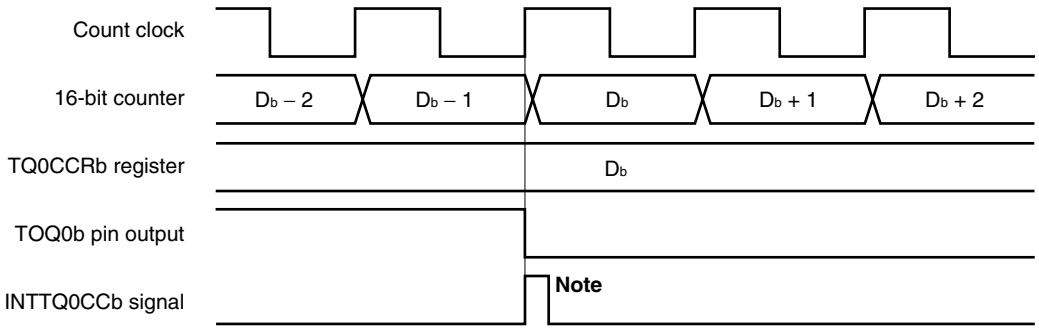
When the TQ0CCR0 register is rewritten from D_{00} to D_{01} and the TQ0CCRb register from D_{b0} to D_{b1} where $D_{00} > D_{01}$ and $D_{b0} > D_{b1}$, if the TQ0CCRb register is rewritten when the count value of the 16-bit counter is greater than D_{b1} and less than D_{b0} and if the TQ0CCR0 register 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 INTTQ0CCb signal and asserts the TOQ0b pin. When the count value matches D_{01}, the counter generates the INTTQ0CC0 signal, deasserts the TOQ0b 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 a = 0 to 3, b = 1 to 3

(b) Generation timing of compare match interrupt request signal (INTTQ0CCb)

The generation timing of the INTTQ0CCb signal in the one-shot pulse output mode is different from INTTQ0CCb signals in other mode; the INTTQ0CCb signal is generated when the count value of the 16-bit counter matches the value of the TQ0CCRb register.



Note Actually, the timing is delayed by one operating clock (fxx).

Remark b = 1 to 3

Usually, the INTTQ0CCb signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TQ0CCRb 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 TOQ0b pin.

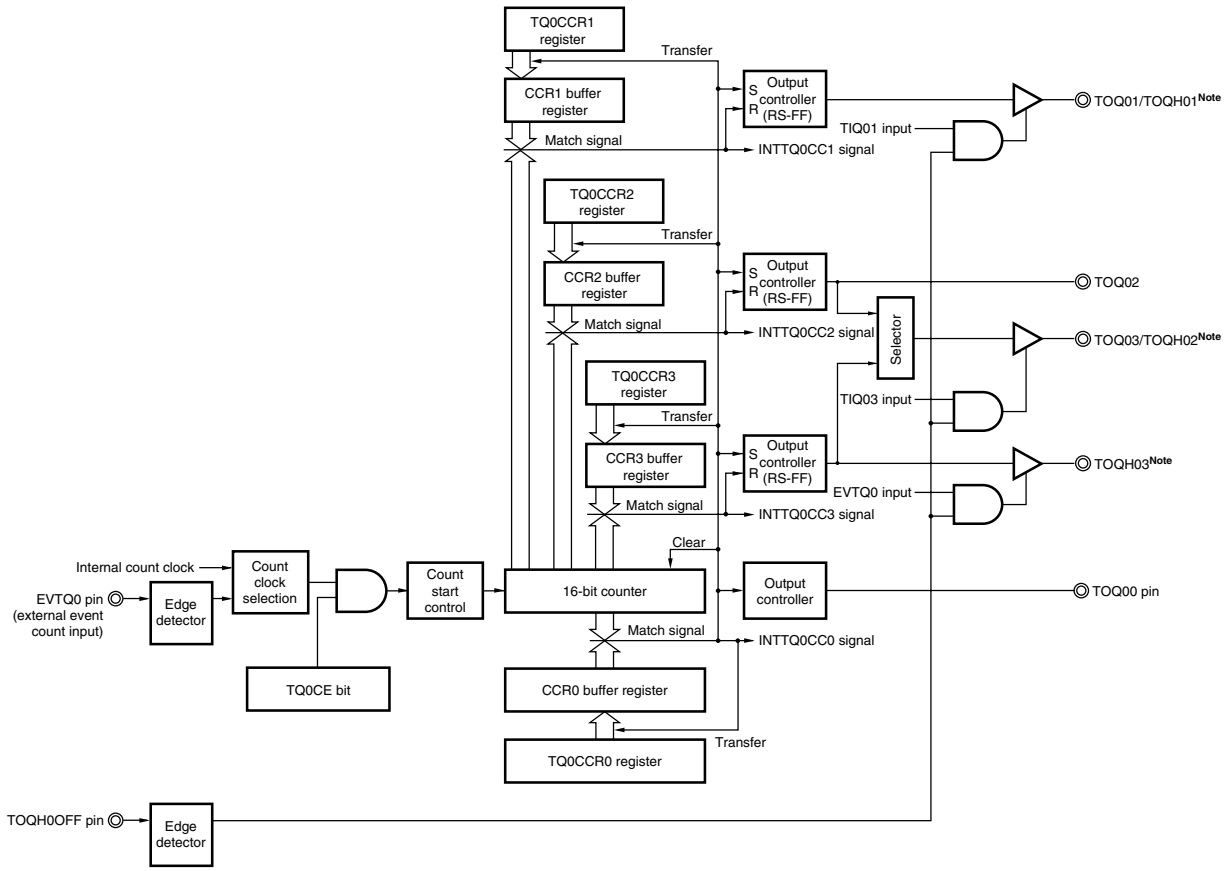
7.6.5 PWM output mode (TQ0MD2 to TQ0MD0 bits = 100)

This mode is valid only in TMQ0.

In the PWM output mode, a PWM waveform is output from the TOQ01 to TOQ03 (TOQH01 to TOQH03) pins when the TQ0CTL0.TQ0CE bit is set to 1.

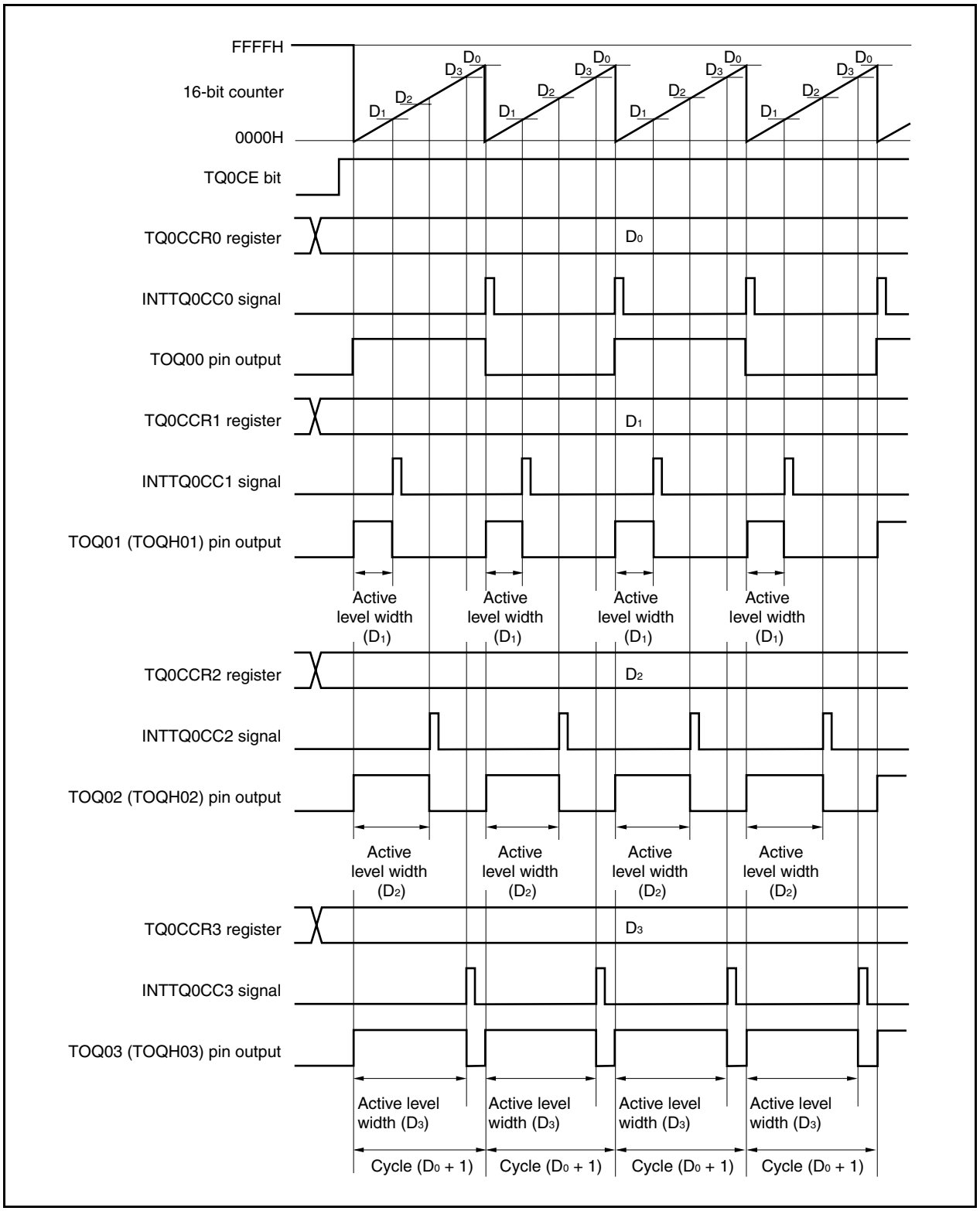
In addition, a PWM waveform with a duty factor of 50% with the set value of the TQ0CCR0 register + 1 as half its cycle is output from the TOQ00 pin.

Figure 7-30. Configuration in PWM Output Mode



Note When the TOQH01 to TOQH03 pins are selected, these pins can be set in the high-impedance state by the TOQH0OFF pin input.

Figure 7-31. Basic Timing in PWM Output Mode



When the TQ0CE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs PWM waveform from the TOQ0b pin (TOQH0b).

The active level width, cycle, and duty factor of the PWM waveform can be calculated as follows.

Active level width = (Set value of TQ0CCRb register) × Count clock cycle
Cycle = (Set value of TQ0CCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TQ0CCRb register)/(Set value of TQ0CCR0 register + 1)

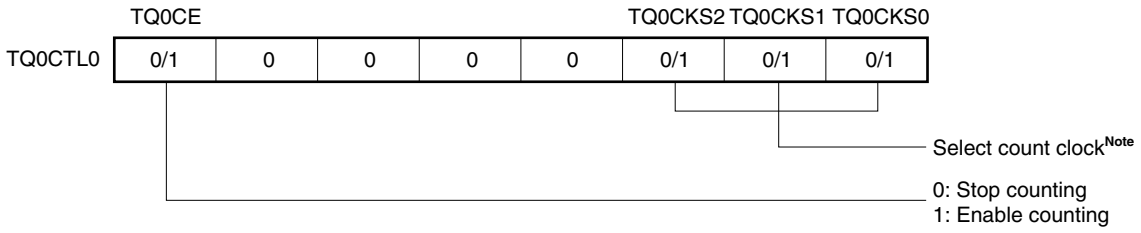
The PWM waveform can be changed by rewriting the TQ0CCRa register while the counter is operating. The newly written value is reflected when the count value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

The compare match interrupt request signal INTTQ0CC0 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 INTTQ0CCb is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.

Remark a = 0 to 3
b = 1 to 3

Figure 7-32. Register Setting in PWM Output Mode (1/3)

(a) TMQ0 control register 0 (TQ0CTL0)



Note The setting is invalid when the TQ0CTL1.TQ0EEE bit = 1.

(b) TMQ0 control register 1 (TQ0CTL1)

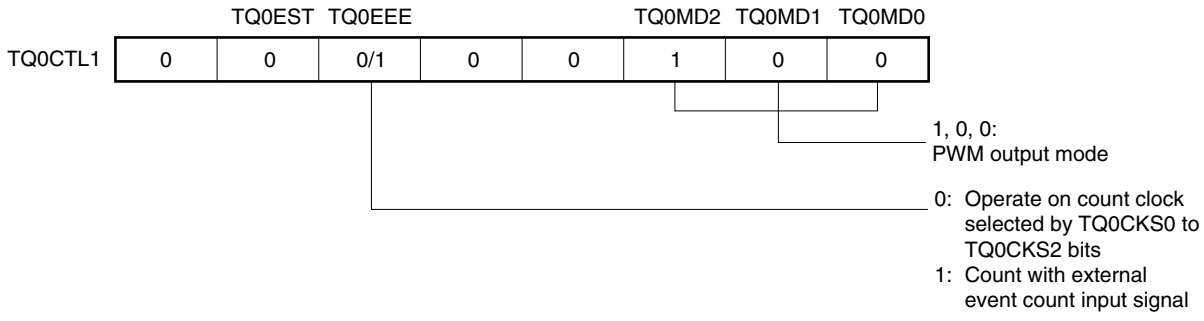
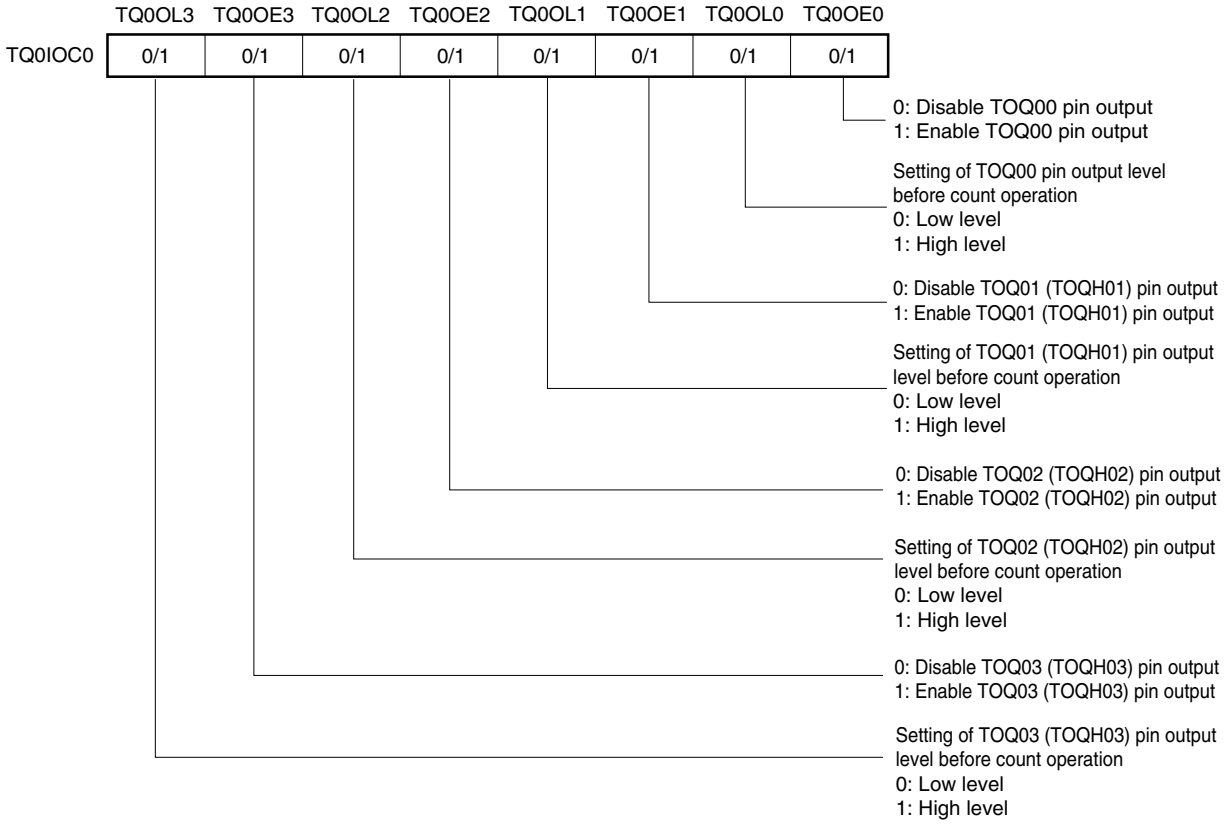
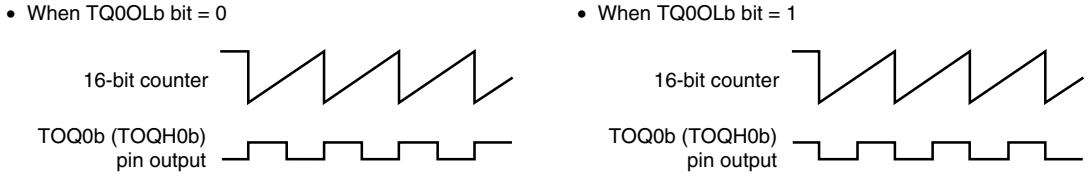


Figure 7-32. Register Setting in PWM Output Mode (2/3)

(c) TMQ0 I/O control register 0 (TQ0IOC0)





(d) TMQ0 I/O control register 2 (TQ0IOC2)

(e) TMQ0 counter read buffer register (TQ0CNT)

The value of the 16-bit counter can be read by reading the TQ0CNT register.

Figure 7-32. Register Setting in PWM Output Mode (3/3)

(f) TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3)

If D_0 is set to the TQ0CCR0 register and D_b to the TQ0CCRb 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. TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the PWM output mode.

2. b = 1 to 3

(1) Operation flow in PWM output mode

Figure 7-33. Software Processing Flow in PWM Output Mode (1/2)

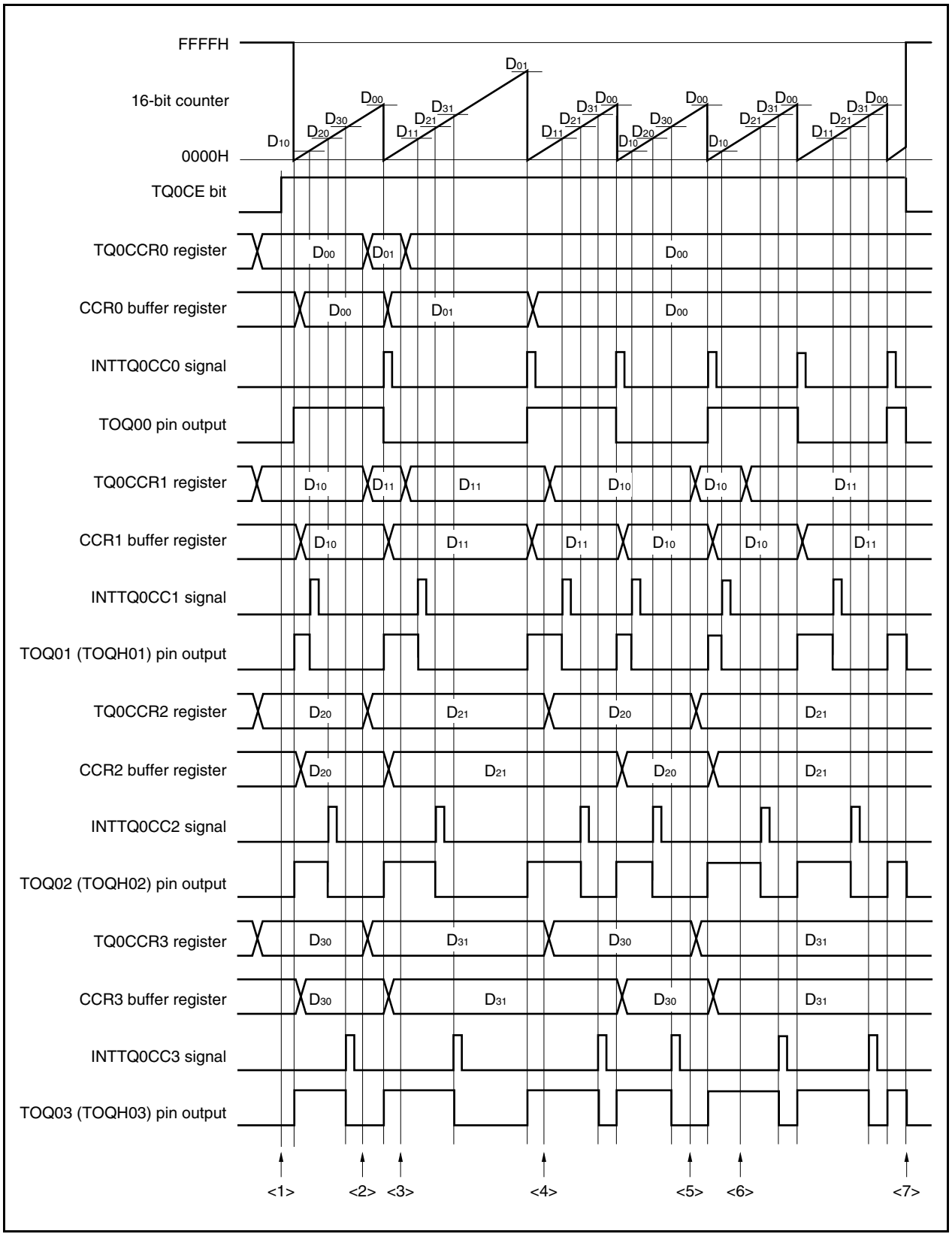
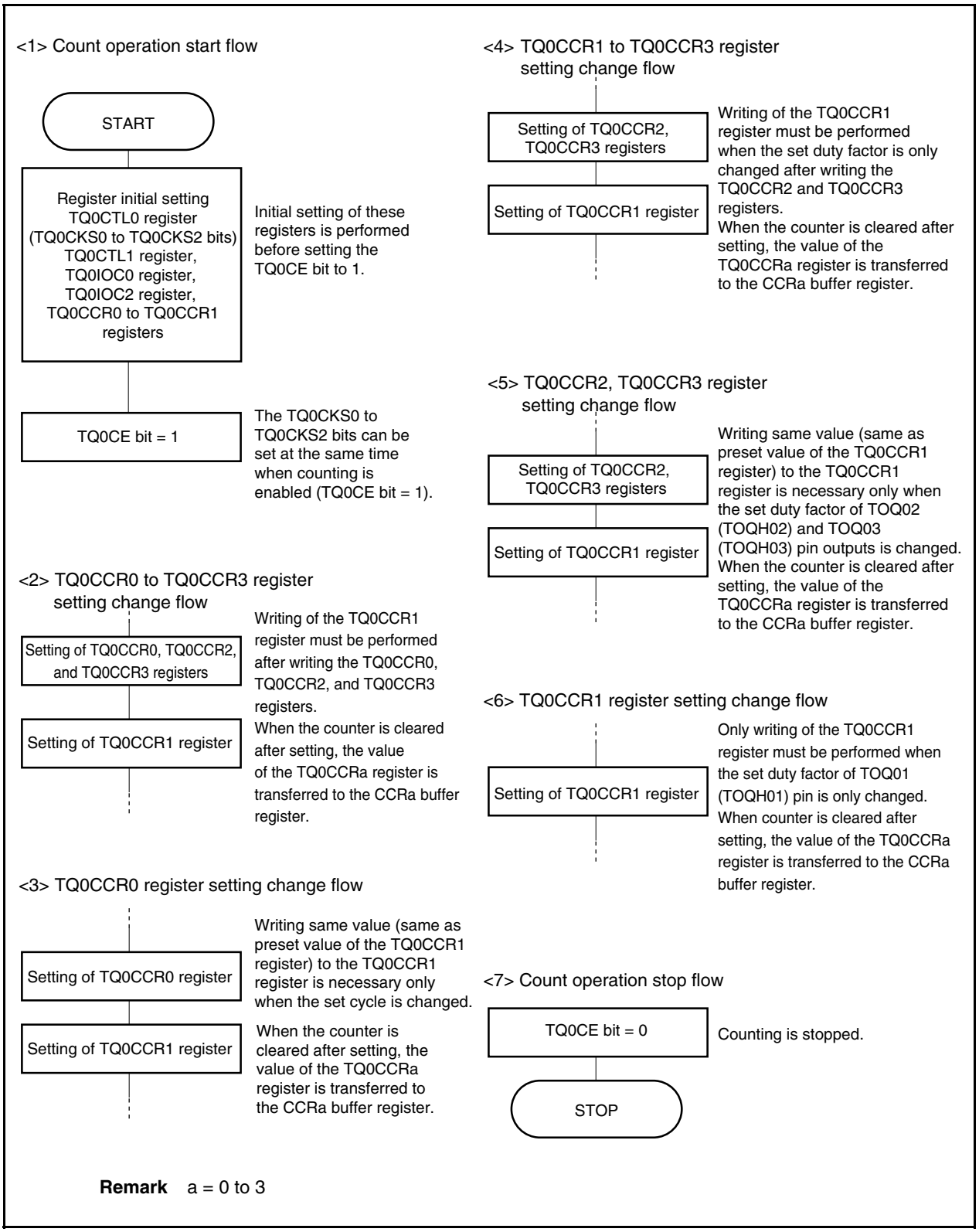


Figure 7-33. Software Processing Flow in PWM Output Mode (2/2)



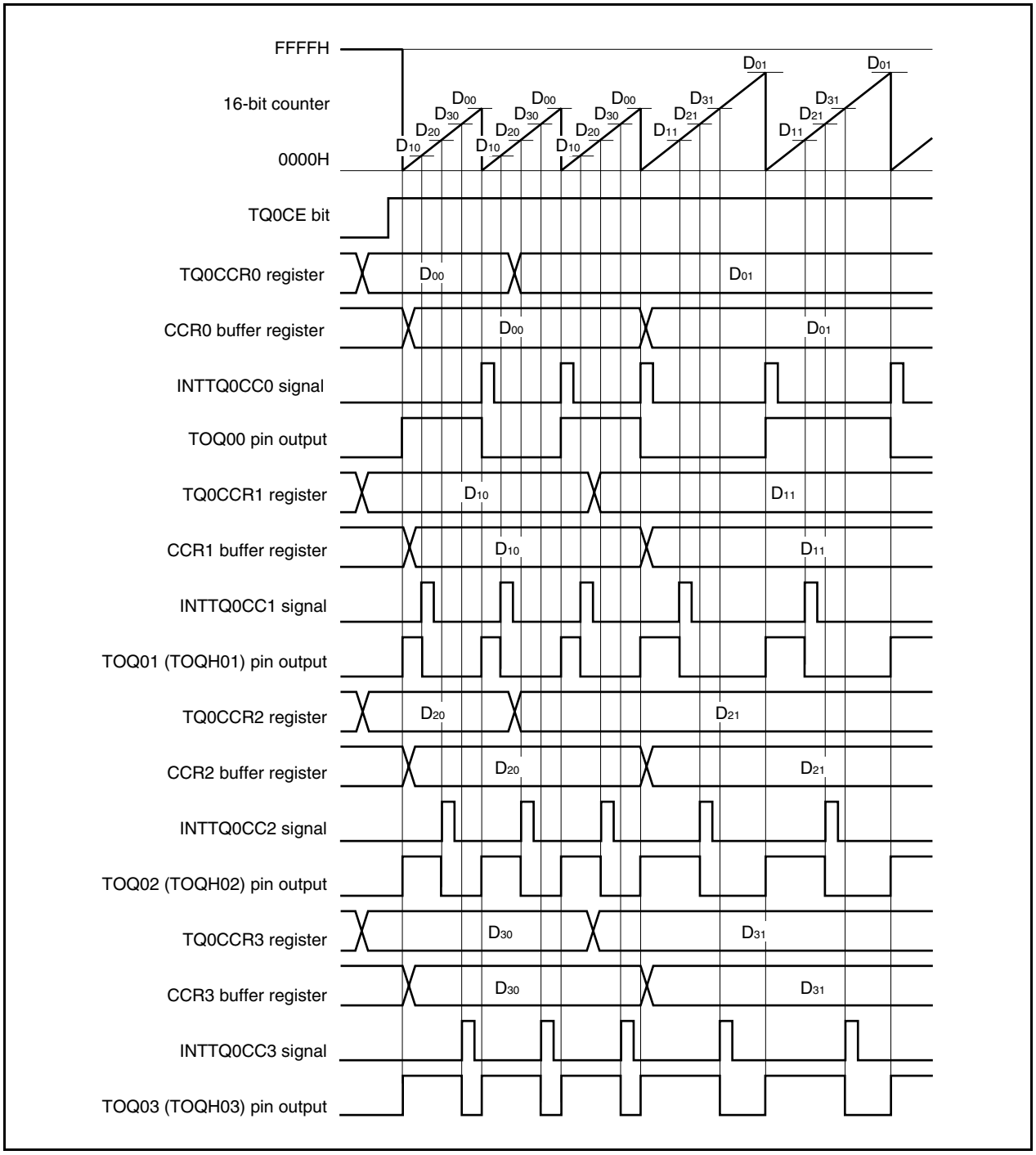
Remark a = 0 to 3

(2) PWM output mode operation timing

(a) Changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TQ0CCR1 register last.

Rewrite the TQ0CCRa register after writing the TQ0CCR1 register after the INTTQ0CC1 signal is detected.



To transfer data from the TQ0CCRa register to the CCRa buffer register, the TQ0CCR1 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 TQ0CCR0 register, set the active level width to the TQ0CCR2 and TQ0CCR3 registers, and then set an active level width to the TQ0CCR1 register.

To change only the cycle of the PWM waveform, first set a cycle to the TQ0CCR0 register, and then write the same value (same as preset value of the TQ0CCR1 register) to the TQ0CCR1 register.

To change only the active level width (duty factor) of PWM wave, first set the active level to the TQ0CCR2 and TQ0CCR3 registers, and then set an active level to the TQ0CCR1 register.

To change only the active level width (duty factor) of the PWM waveform output by the TOQ01 (TOQH01) pin, only the TQ0CCR1 register has to be set.

To change only the active level width (duty factor) of the PWM waveform output by the TOQ02 (TOQH02) and TOQ03 (TOQH03) pins, first set an active level width to the TQ0CCR2 and TQ0CCR3 registers, and then write the same value (same as preset value of the TQ0CCR1 register) to the TQ0CCR1 register.

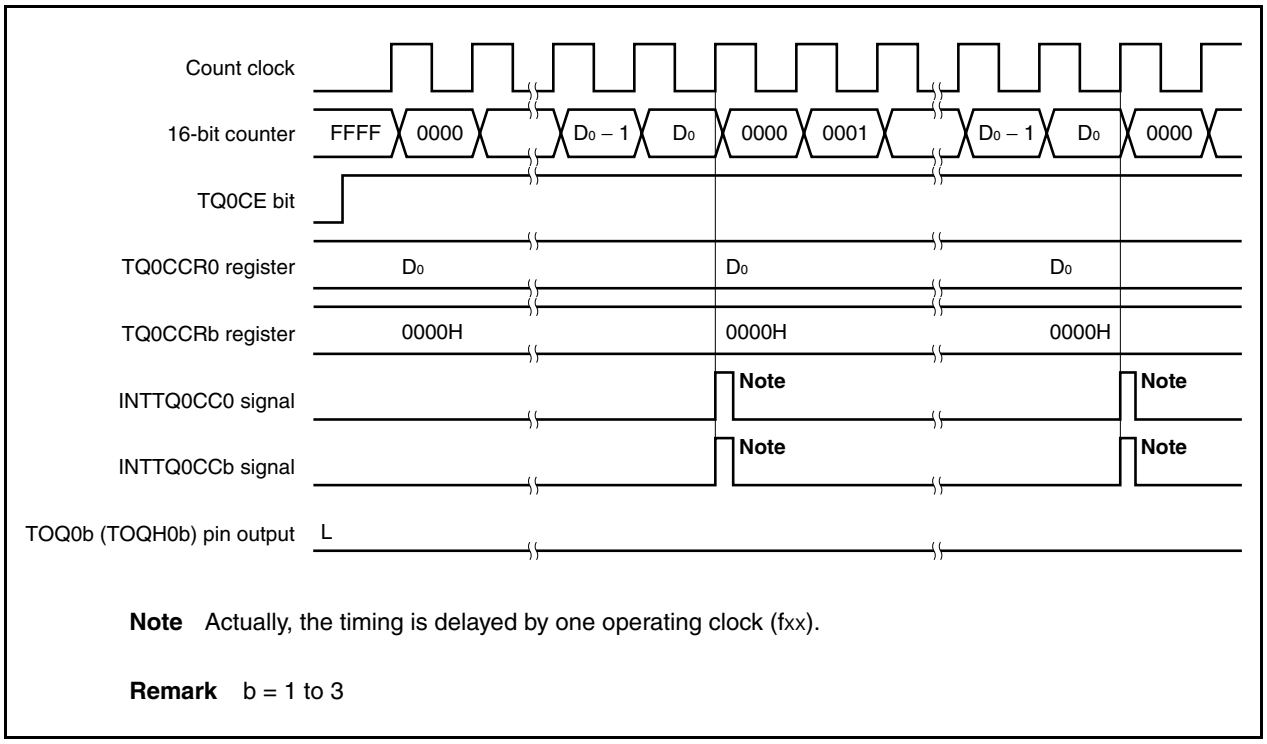
After the TQ0CCR1 register is written, the value written to the TQ0CCRa register is transferred to the CCRa buffer register in synchronization with the timing of clearing the 16-bit counter, and is used as a value to be compared with the value of the 16-bit counter.

To write the TQ0CCR0 to TQ0CCR3 registers again after writing the TQ0CCR1 register once, do so after the INTTQ0CC0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TQ0CCRa register to the CCRa buffer register conflicts with writing the TQ0CCRa register.

Remark a = 0 to 3

(b) 0%/100% output of PWM waveform

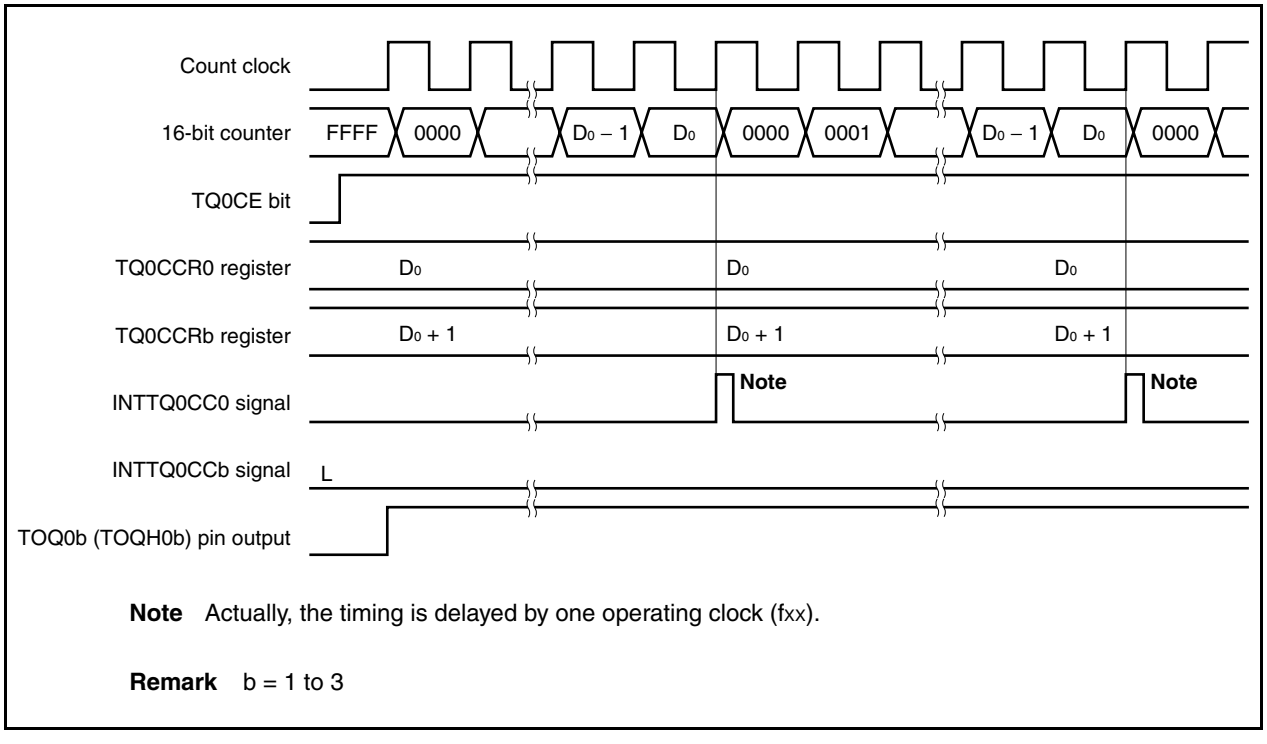
To output a 0% waveform, set the TQ0CCRb register to 0000H. The 16-bit counter is cleared to 0000H and the INTTQ0CC0 and INTTQ0CCb signals are generated at the next timing after a match between the count value of the 16-bit counter and the value of the CCR0 buffer register.



Note Actually, the timing is delayed by one operating clock (fxx).

Remark b = 1 to 3

To output a 100% waveform, set a value of (set value of TQ0CCR0 register + 1) to the TQ0CCRb register. If the set value of the TQ0CCR0 register is FFFFH, 100% output cannot be produced.

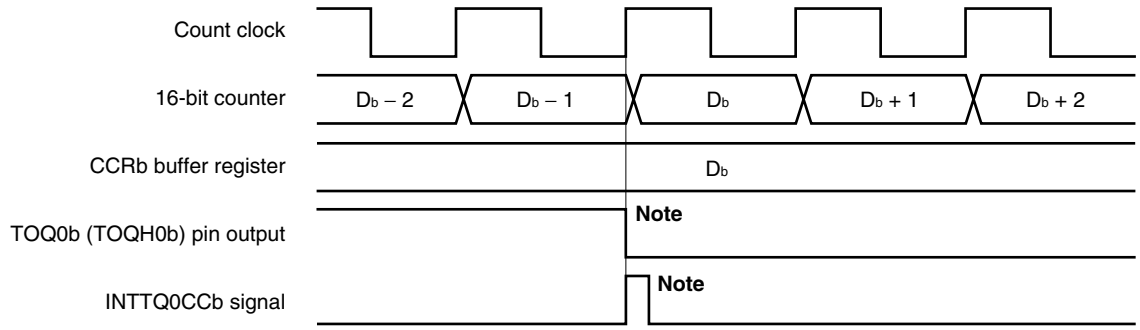


Note Actually, the timing is delayed by one operating clock (fxx).

Remark b = 1 to 3

(c) Generation timing of compare match interrupt request signal (INTTQ0CCb)

The timing of generation of the INTTQ0CCb signal in the PWM output mode differs from the timing of INTTQ0CCb signals in other mode; the INTTQ0CCb signal is generated when the count value of the 16-bit counter matches the value of the TQ0CCRb register.



Note Actually, the timing is delayed by one operating clock (fxx).

Remark b = 1 to 3

Usually, the INTTQ0CCb signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TQ0CCRb register.

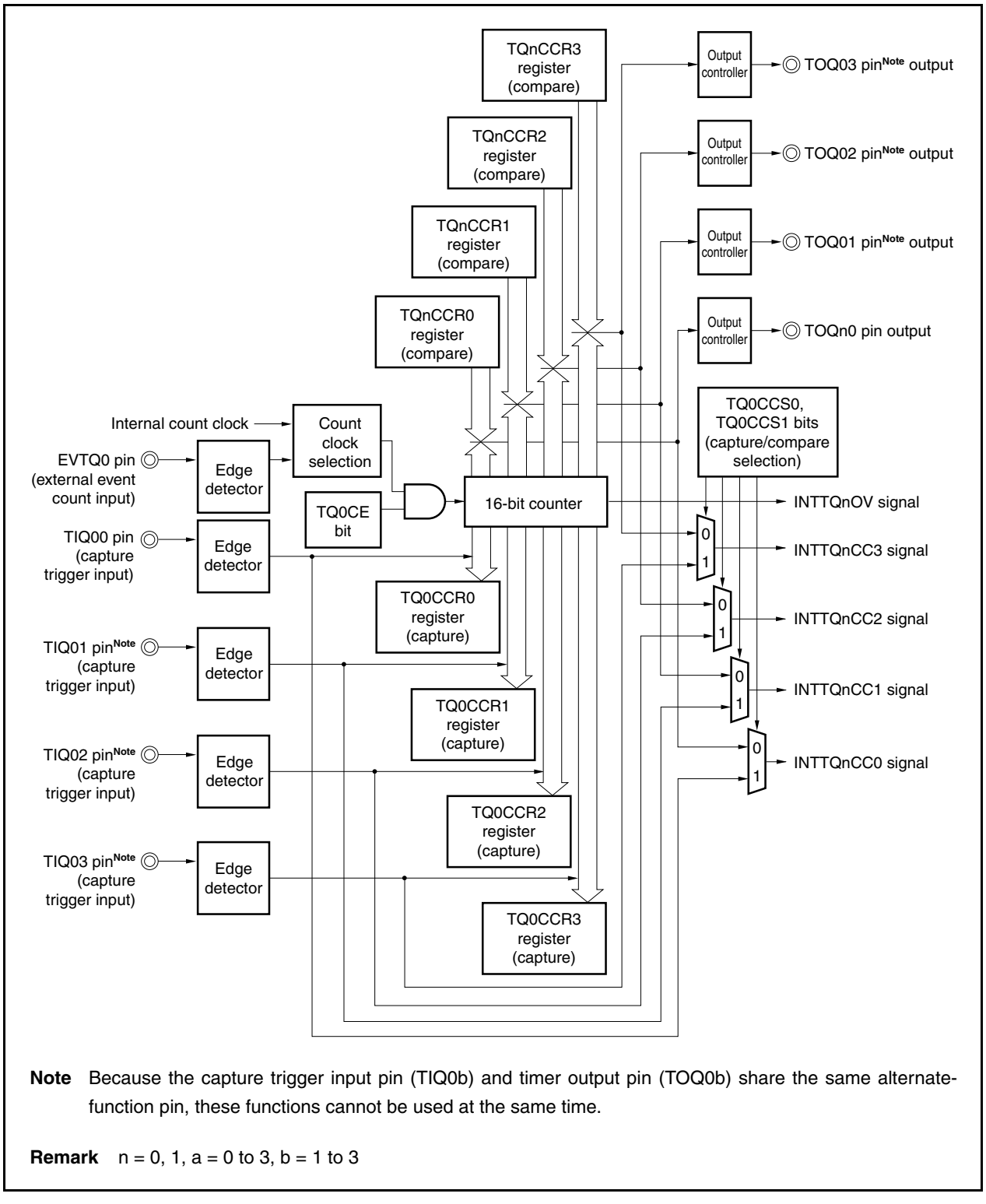
In the PWM output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the output signal of the TOQ0b (TOQH0b) pin.

7.6.6 Free-running timer mode (TQnMD2 to TQnMD0 bits = 101)

The compare function is valid in both TMQ0 and TMQ1. The capture function is valid in TMQ0 only.

In the free-running timer mode, 16-bit timer/event counter Q starts counting when the TQnCTL0.TQnCE bit is set to 1. At this time, the TQ0CCRa register can be used as a compare register or a capture register, depending on the setting of the TQ0OPT0.TQ0CCSa bit.

Figure 7-34. Configuration in Free-Running Timer Mode



Note Because the capture trigger input pin (TIQ0b) and timer output pin (TOQ0b) share 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

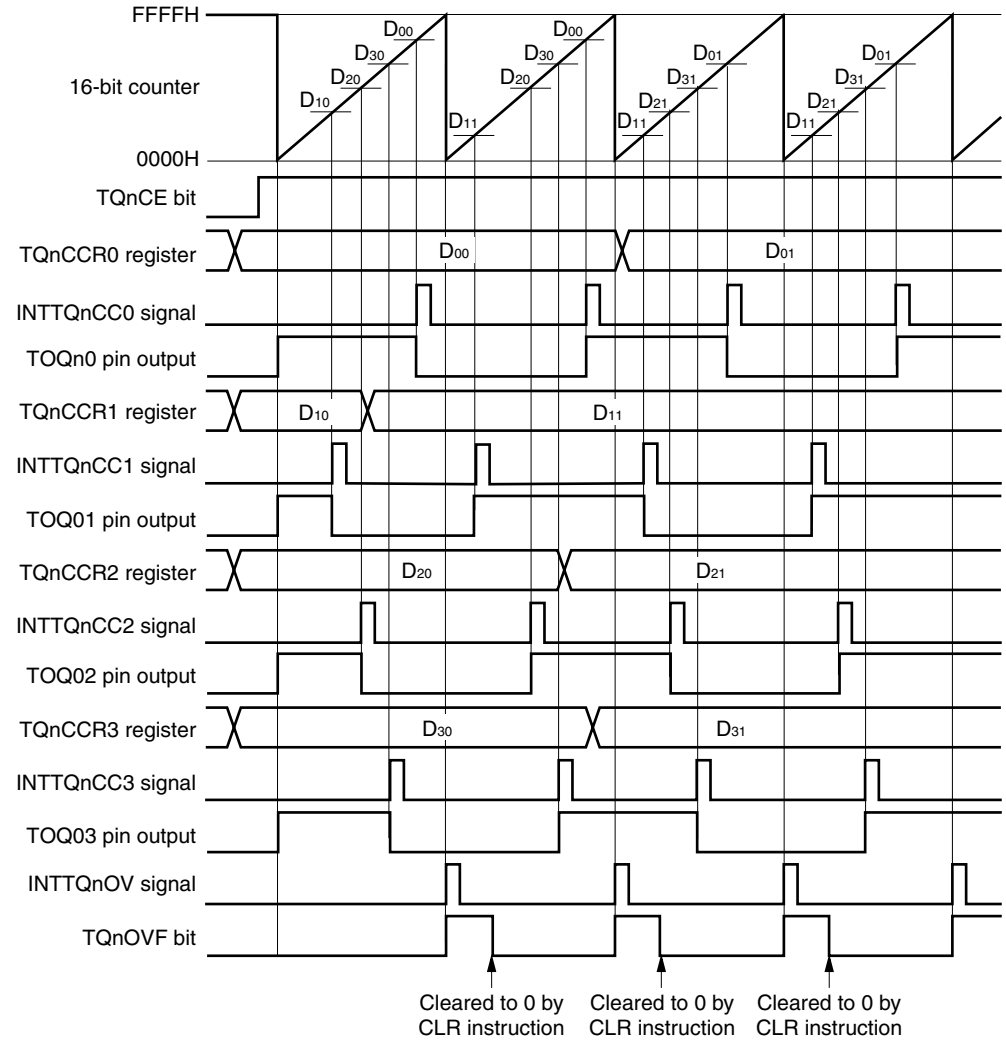
- Compare operation

  When the TQnCE bit is set to 1, 16-bit timer/event counter Q starts counting, and the output signals of the TOQ00 to TOQ03 and TOQ10 pins are inverted. When the count value of the 16-bit counter later matches the set value of the TQnCCRa register, a compare match interrupt request signal (INTTQnCCa) is generated, and the output signals of the TOQ00 to TOQ03 and TOQ10 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 (INTTQnOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TQnOPT0.TQnOVF 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 TQnCCRa 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 7-35. Basic Timing in Free-Running Timer Mode (Compare Function)



Remark n = 0, 1

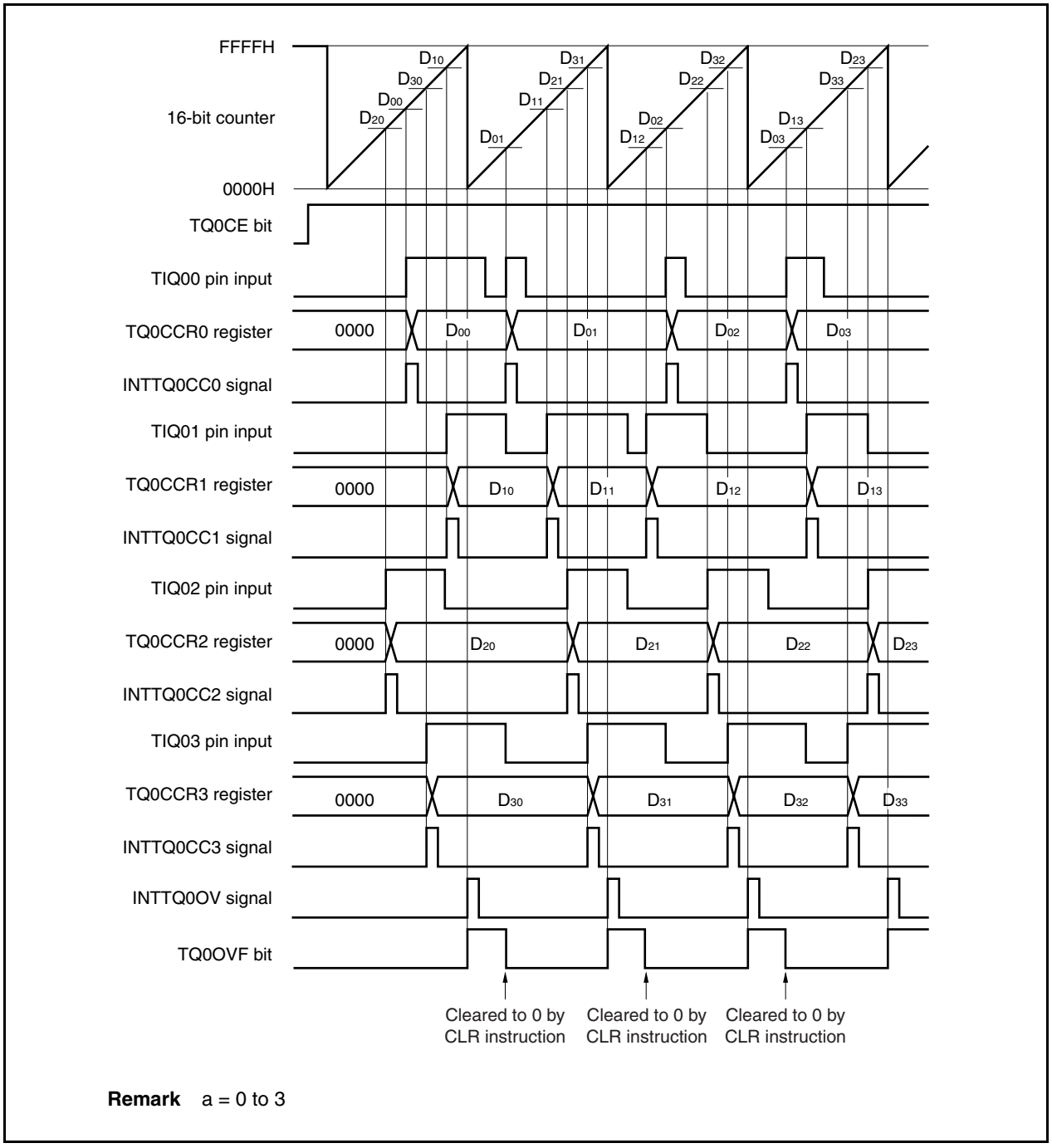
a = 0 to 3

- Capture operation

  When the TQ0CE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIQ0a pin is detected, the count value of the 16-bit counter is stored in the TQ0CCRa register, and a capture interrupt request signal (INTTQ0CCa) 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 (INTTQ0OV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TQ0OVF 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.

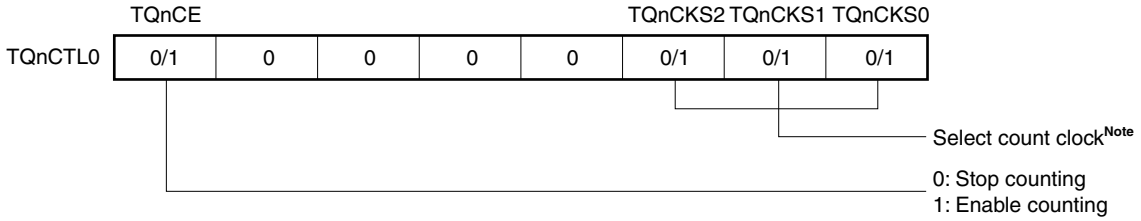
Figure 7-36. Basic Timing in Free-Running Timer Mode (Capture Function)



Remark a = 0 to 3

Figure 7-37. Register Setting in Free-Running Timer Mode (1/3)

(a) TMQn control register 0 (TQnCTL0)



Note The setting is invalid when the TQ0CTL1.TQ0EEE bit = 1.

(b) TMQn control register 1 (TQnCTL1)

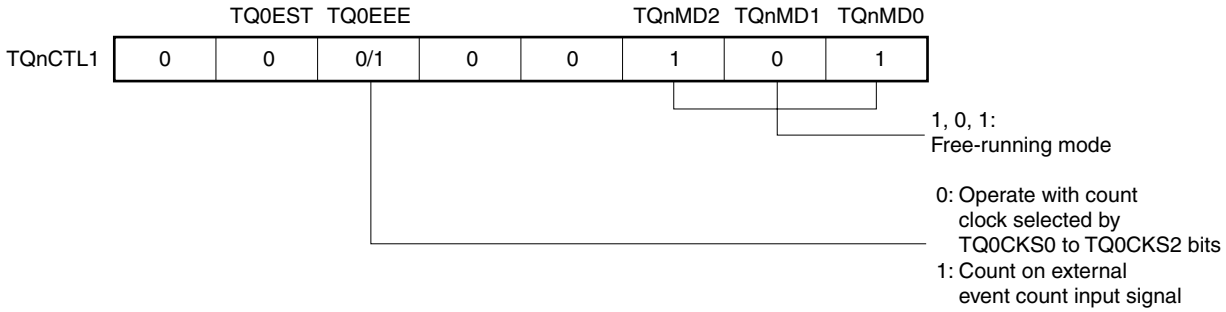
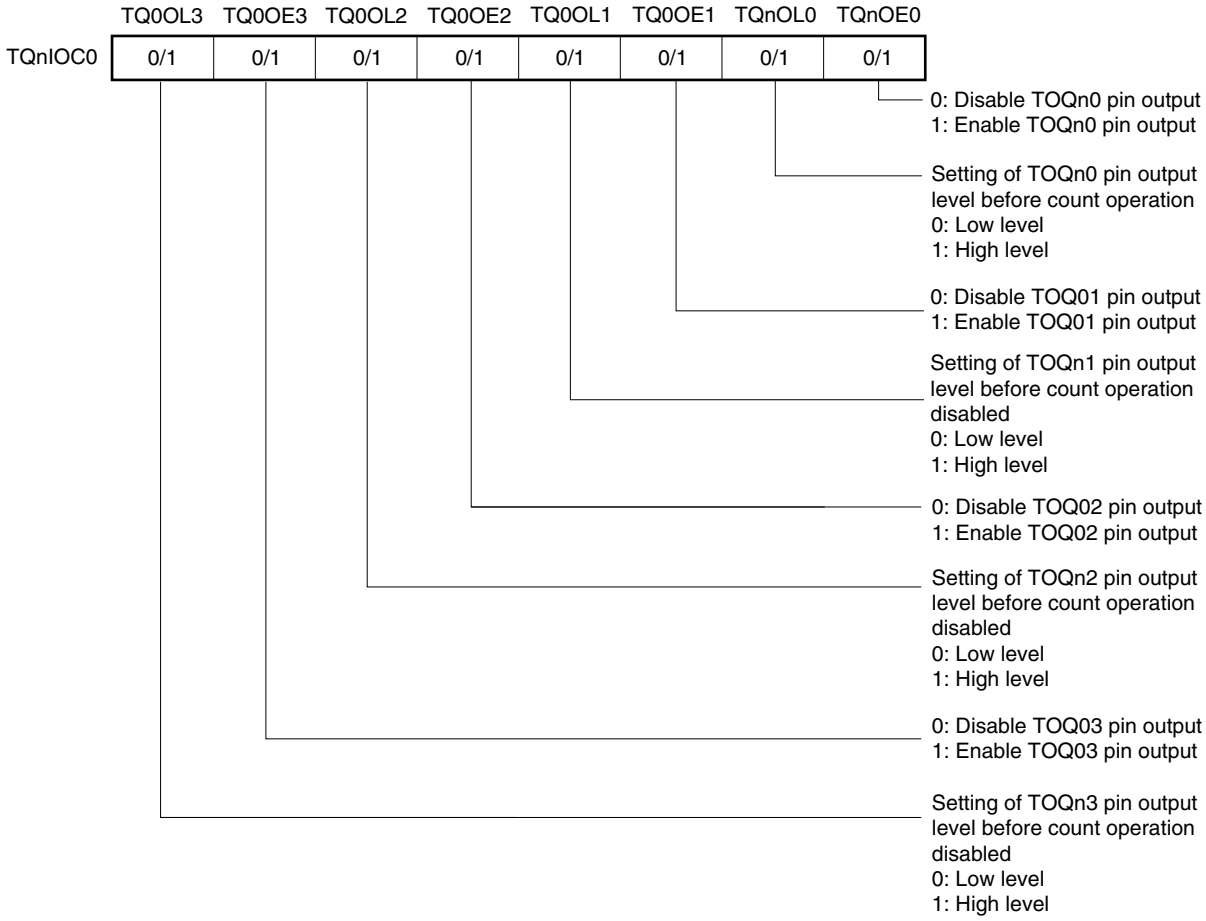


Figure 7-37. Register Setting in Free-Running Timer Mode (2/3)

(c) TMQn I/O control register 0 (TQnIOC0)



(d) TMQ0 I/O control register 1 (TQ0IOC1)

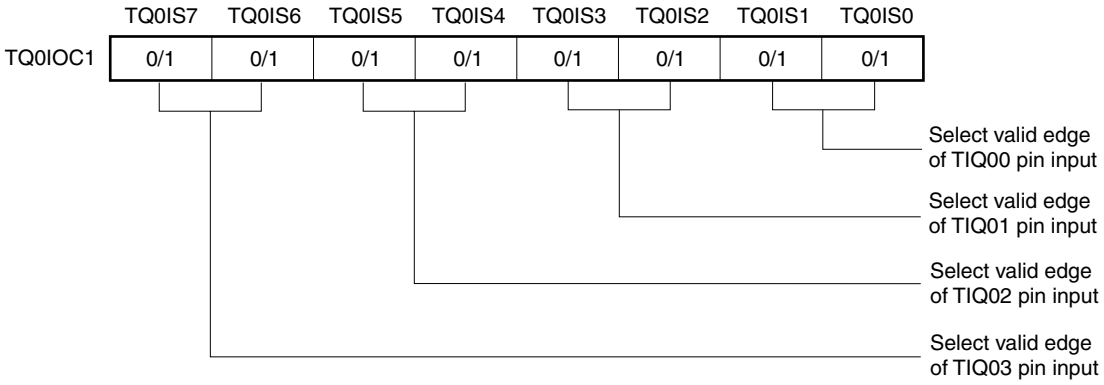
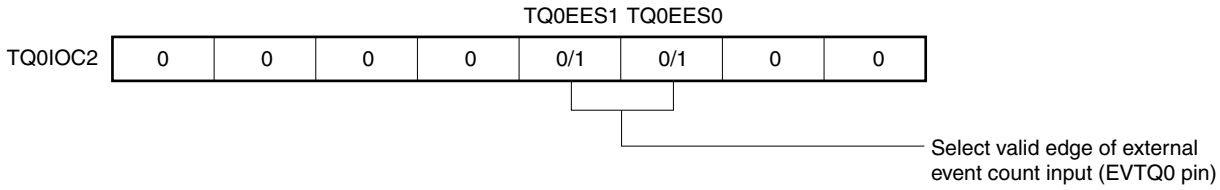
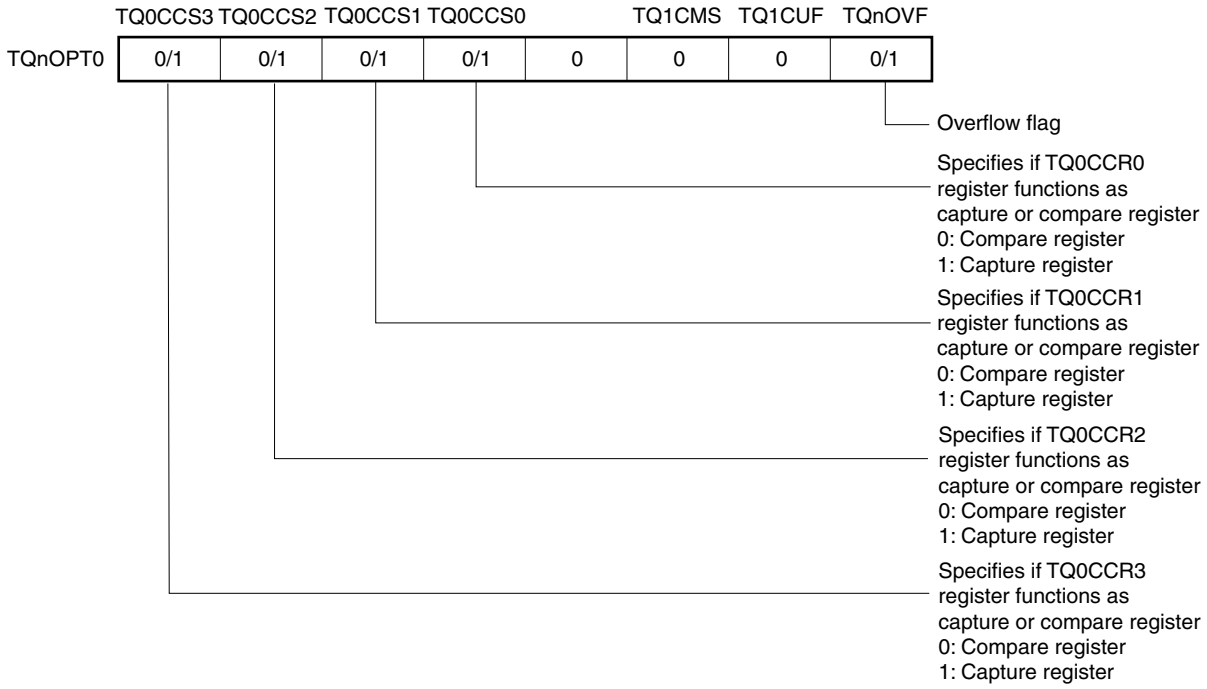


Figure 7-37. Register Setting in Free-Running Timer Mode (3/3)

(e) TMQ0 I/O control register 2 (TQ0IOC2)



(f) TMQn option register 0 (TQnOPT0)



(g) TMQn counter read buffer register (TQnCNT)

The value of the 16-bit counter can be read by reading the TQnCNT register.

(h) TMQn capture/compare registers 0 to 3 (TQnCCR0 to TQnCCR3)

These registers function as capture registers or compare registers depending on the setting of the TQ0OPT0.TQ0CCSa 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 TIQ0a pin is detected.

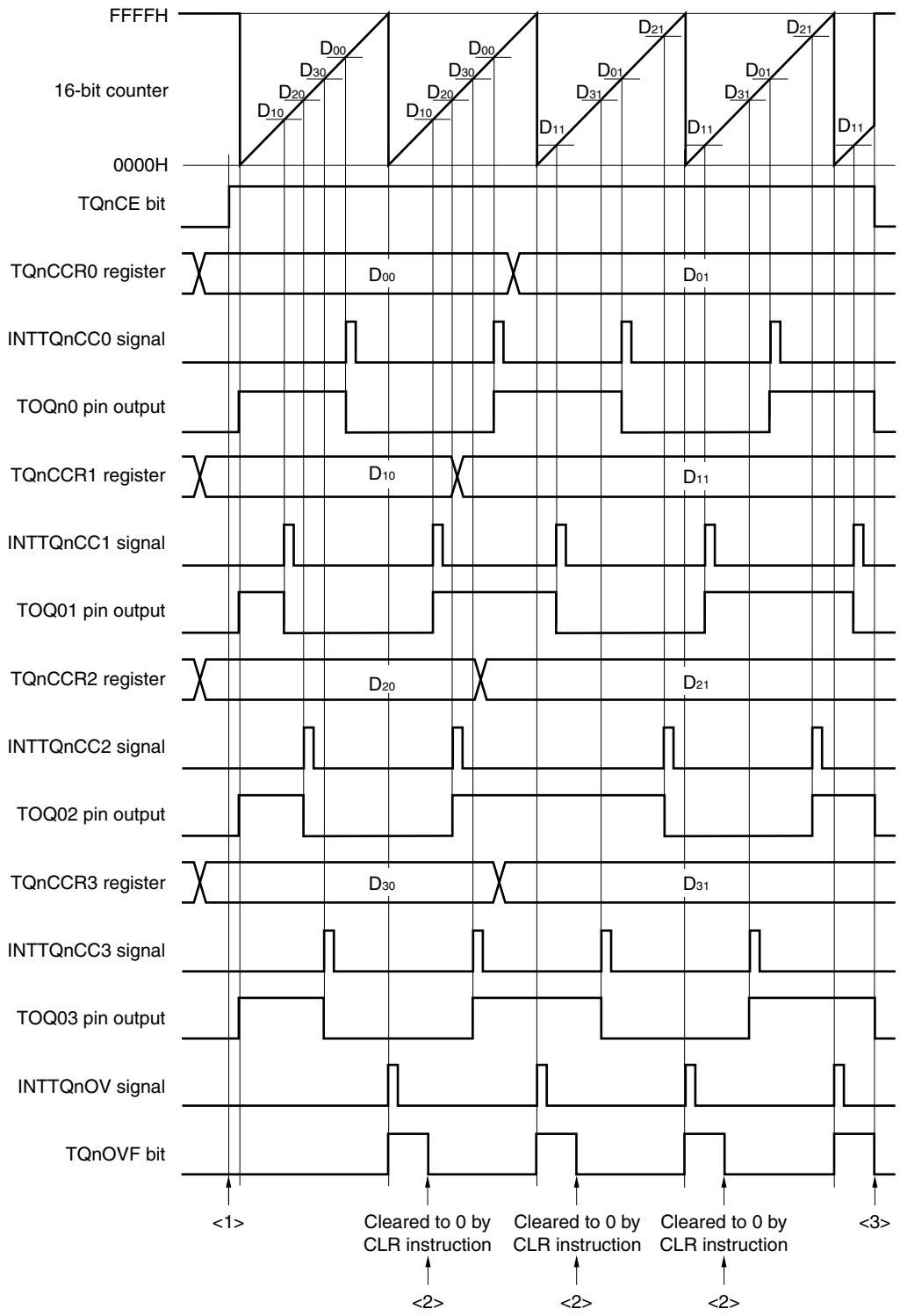
When the registers function as compare registers and when D_a is set to the TQnCCRa register, the INTTQnCCa signal is generated when the counter reaches (D_a + 1), and the output signals of the TOQ00 to TOQ03 and TOQ10 pins are inverted.

Remark a = 0 to 3

(1) Operation flow in free-running timer mode

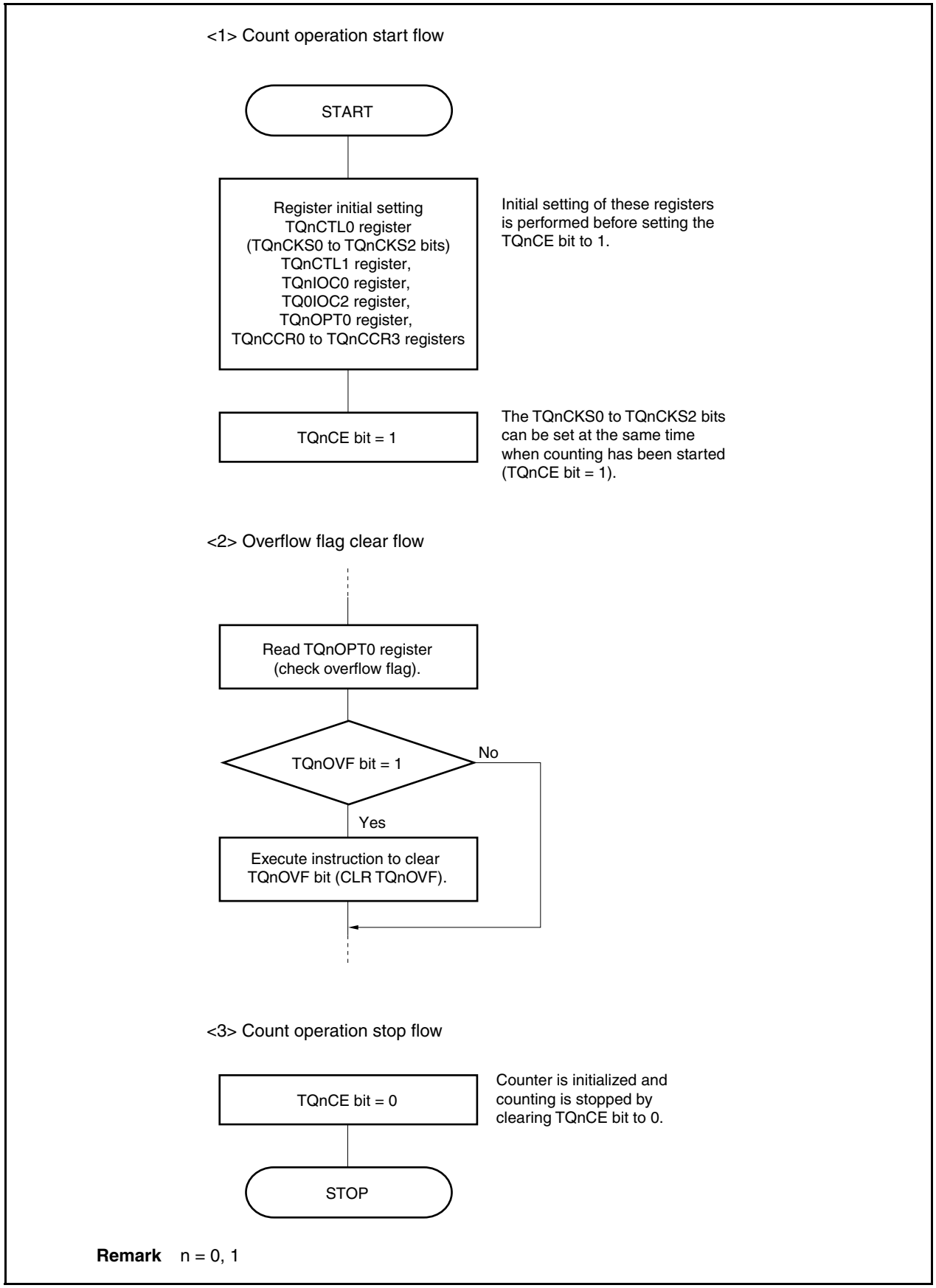
(a) When using capture/compare register as compare register

Figure 7-38. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)



Remark n = 0, 1

Figure 7-38. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)



Remark n = 0, 1

(b) When using capture/compare register as capture register

Figure 7-39. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)

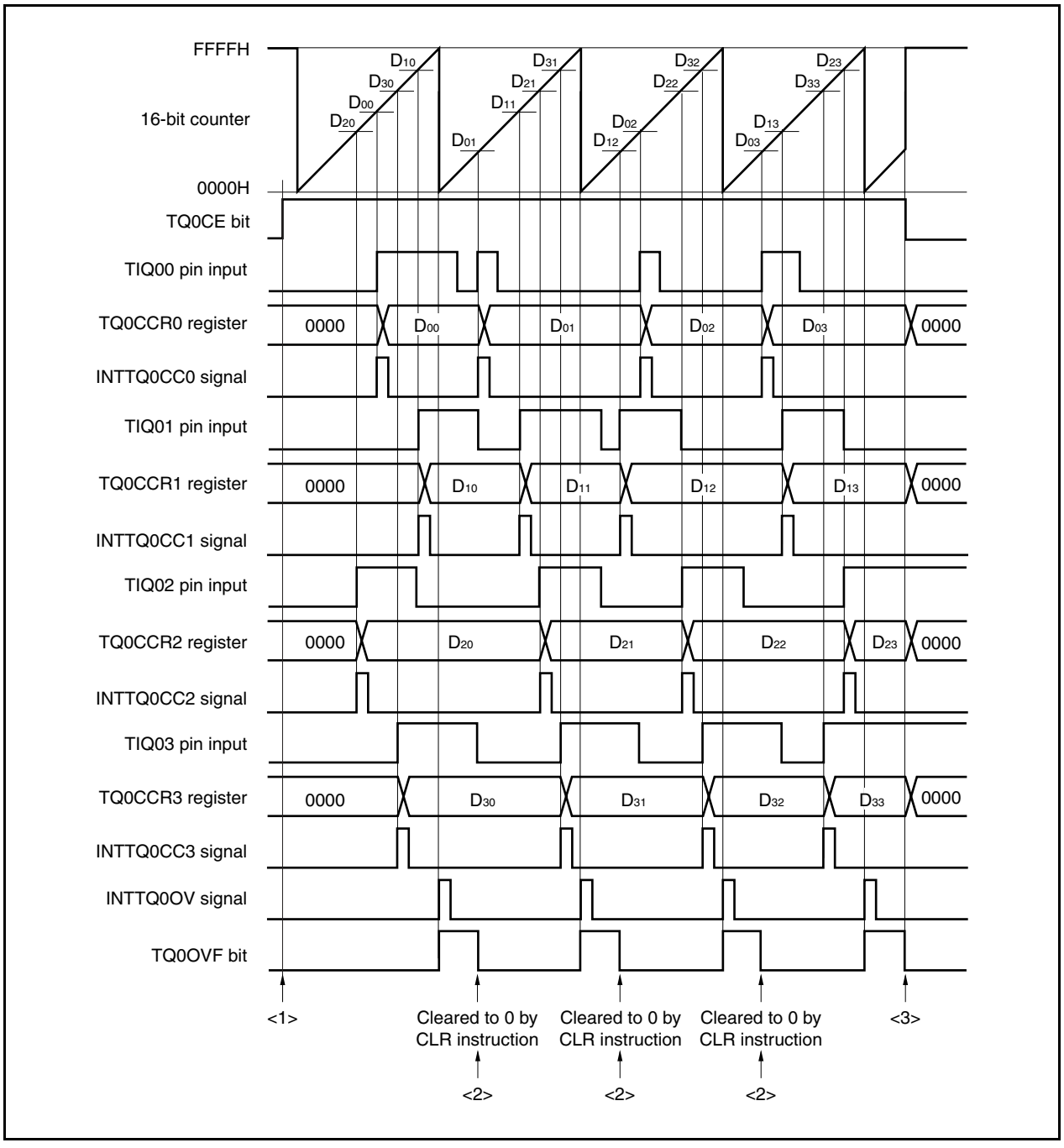
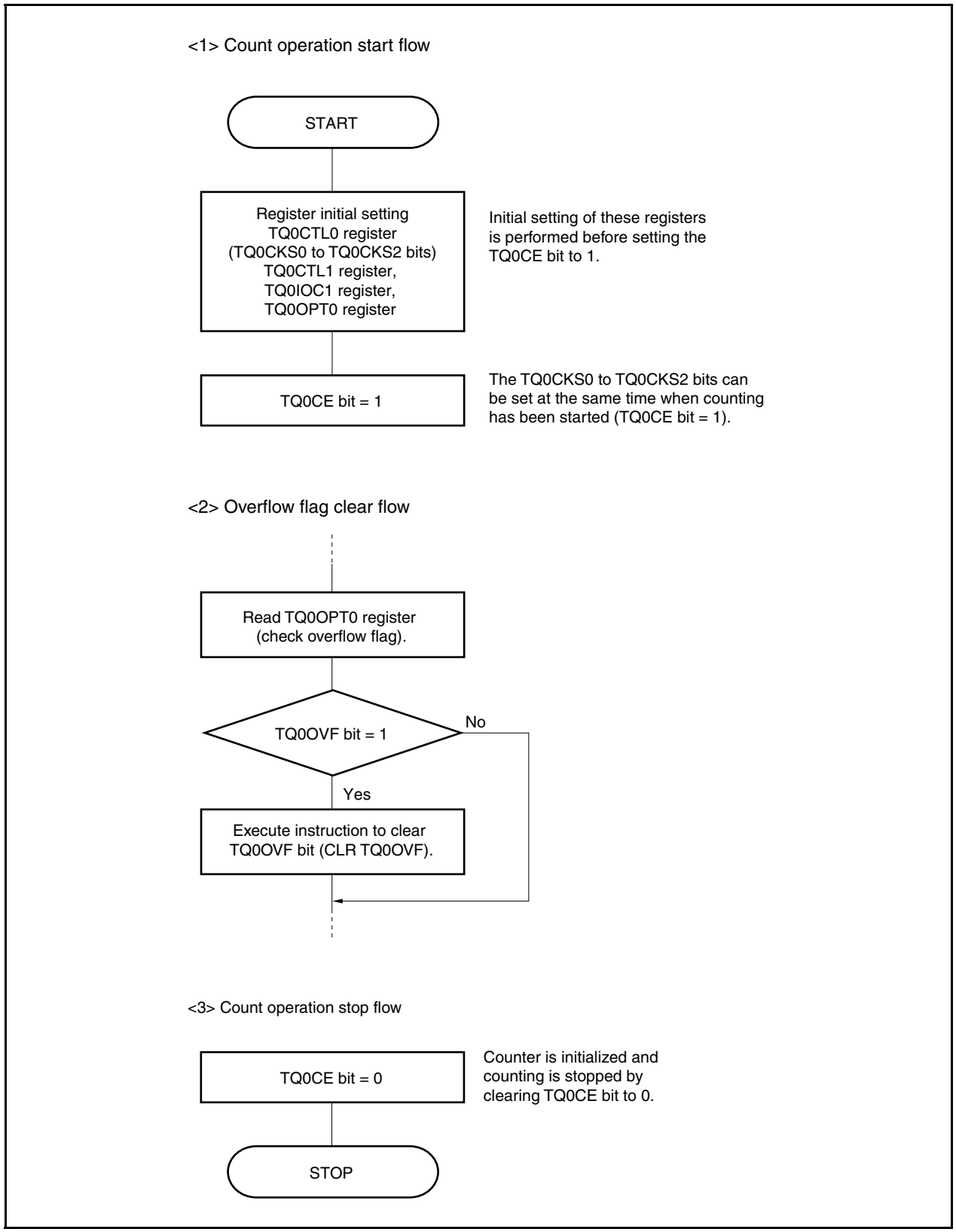


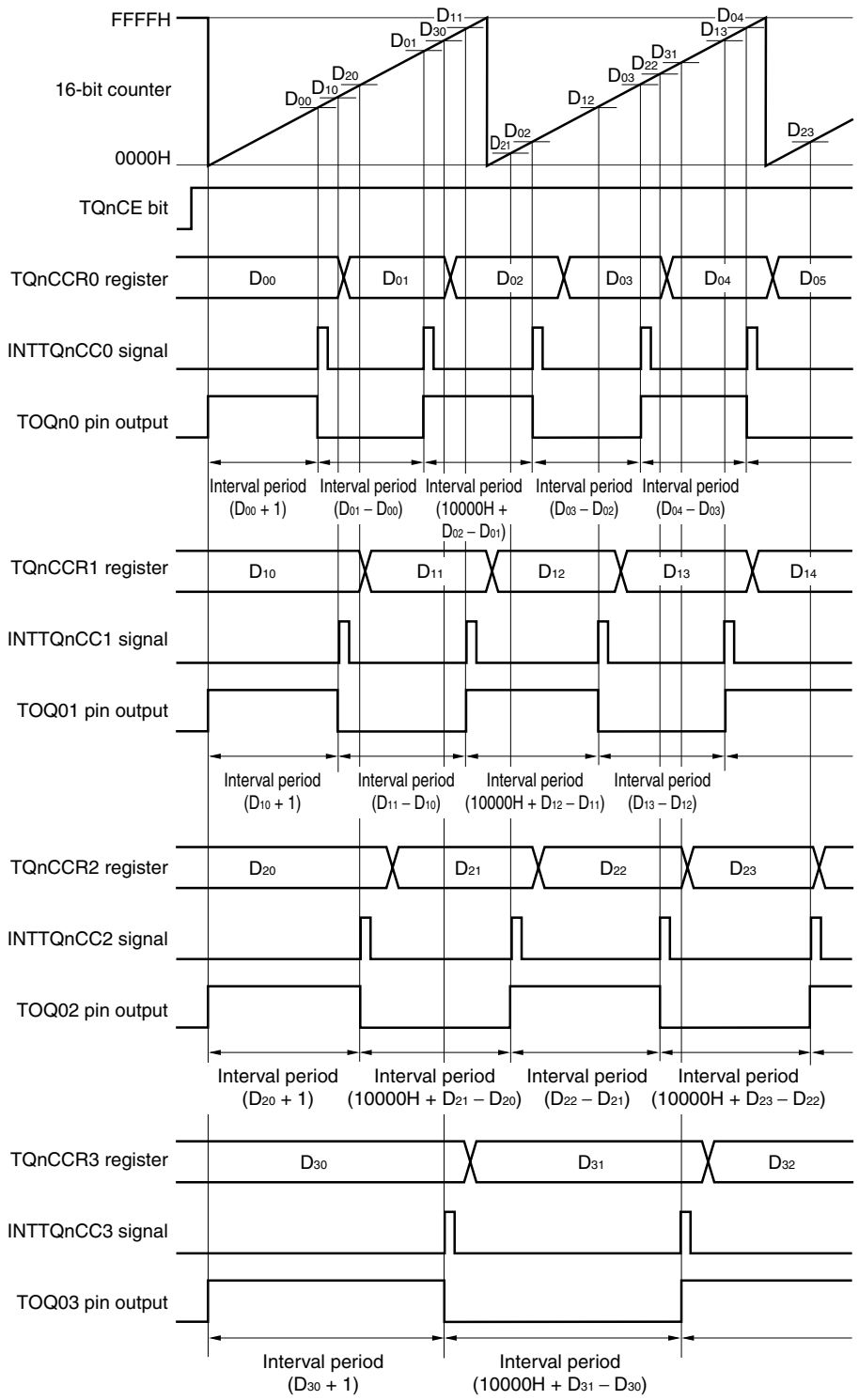
Figure 7-39. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)



(2) Operation timing in free-running timer mode

(a) Interval operation with compare register

When 16-bit timer/event counter Q is used as an interval timer with the TQnCCRa 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 INTTQnCCa 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 TQnCCRa register must be re-set in the interrupt servicing that is executed when the INTTQnCCa signal is detected.

The set value for re-setting the TQnCCRa 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

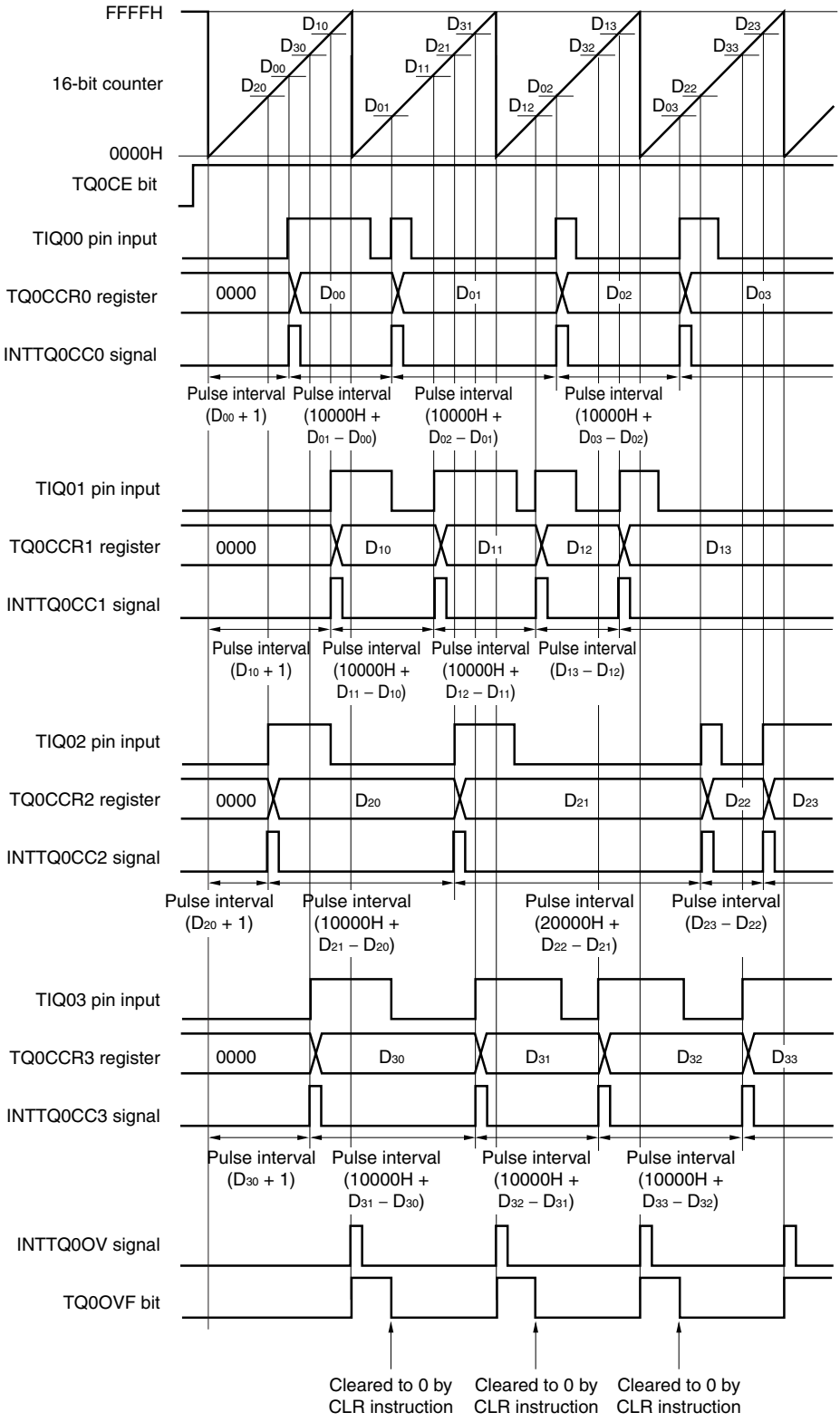
(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

Remark n = 0, 1

a = 0 to 3

(b) Pulse width measurement with capture register

When pulse width measurement is performed with the TQ0CCRa register used as a capture register, software processing is necessary for reading the capture register each time the INTTQ0CCa 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.

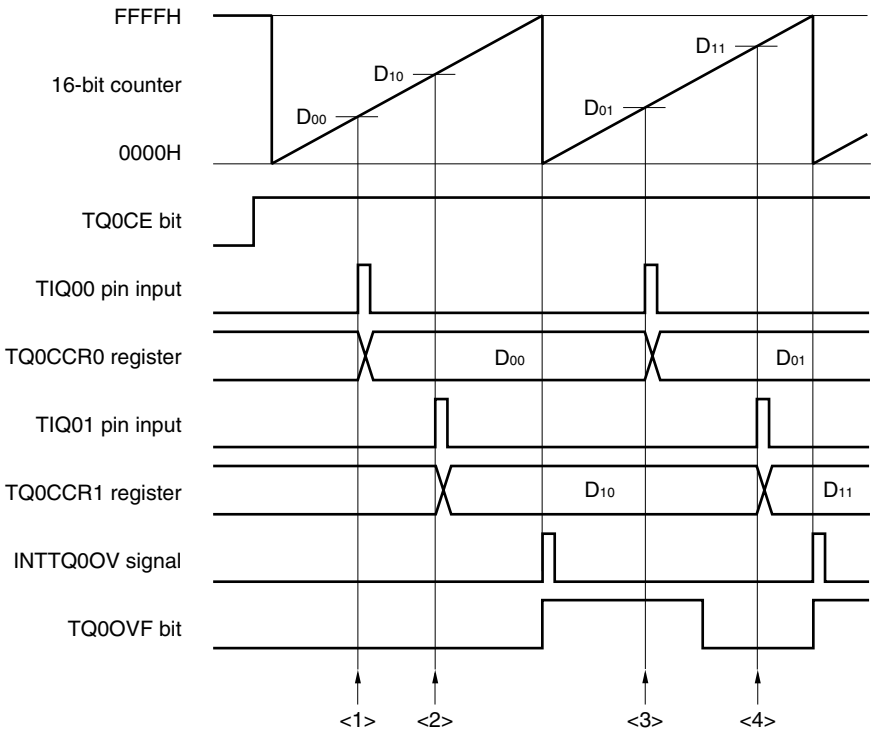
To measure a pulse width, the pulse width can be calculated by reading the value of the TQ0CCRa register in synchronization with the INTTQ0CCa signal, and calculating the difference between the read value and the previously read value.

Remark 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



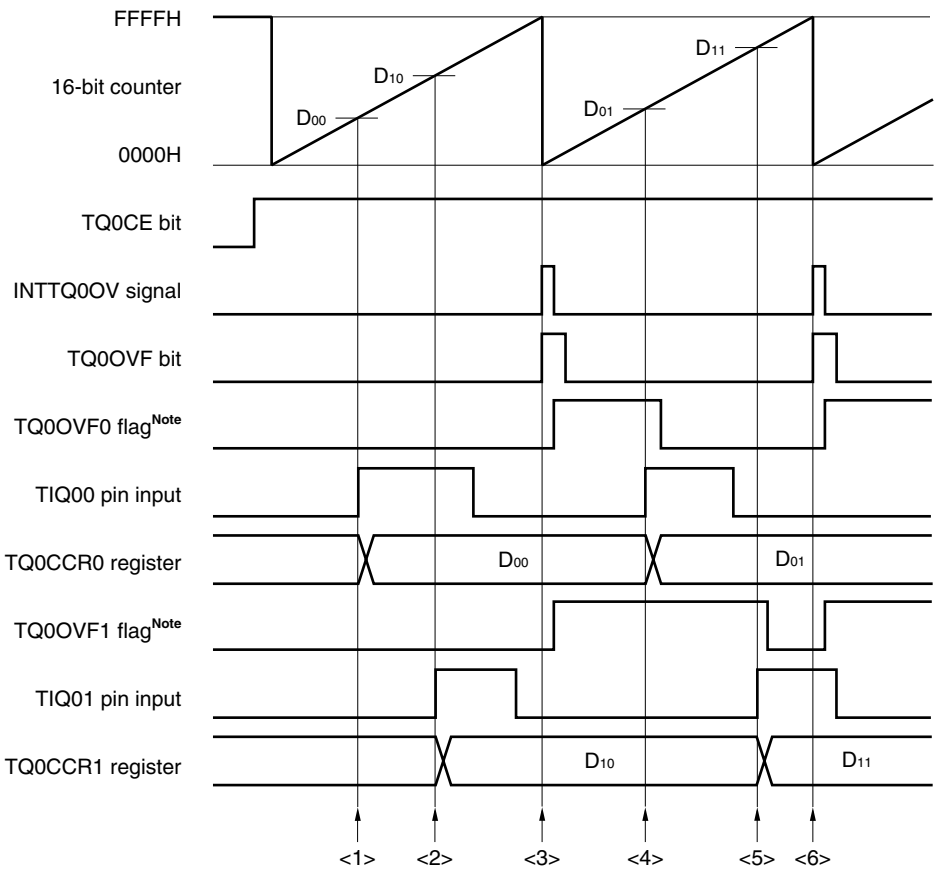
The following problem may occur when two pulse widths are measured in the free-running timer mode.

<1> Read the TQ0CCR0 register (setting of the default value of the TIQ00 pin input).
<2> Read the TQ0CCR1 register (setting of the default value of the TIQ01 pin input).
<3> Read the TQ0CCR0 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 TQ0CCR1 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).

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 TQ0OVF0 and TQ0OVF1 flags are set on the internal RAM by software.

<1> Read the TQ0CCR0 register (setting of the default value of the TIQ00 pin input).

<2> Read the TQ0CCR1 register (setting of the default value of the TIQ01 pin input).

<3> An overflow occurs. Set the TQ0OVF0 and TQ0OVF1 flags to 1 in the overflow interrupt servicing, and clear the overflow flag to 0.

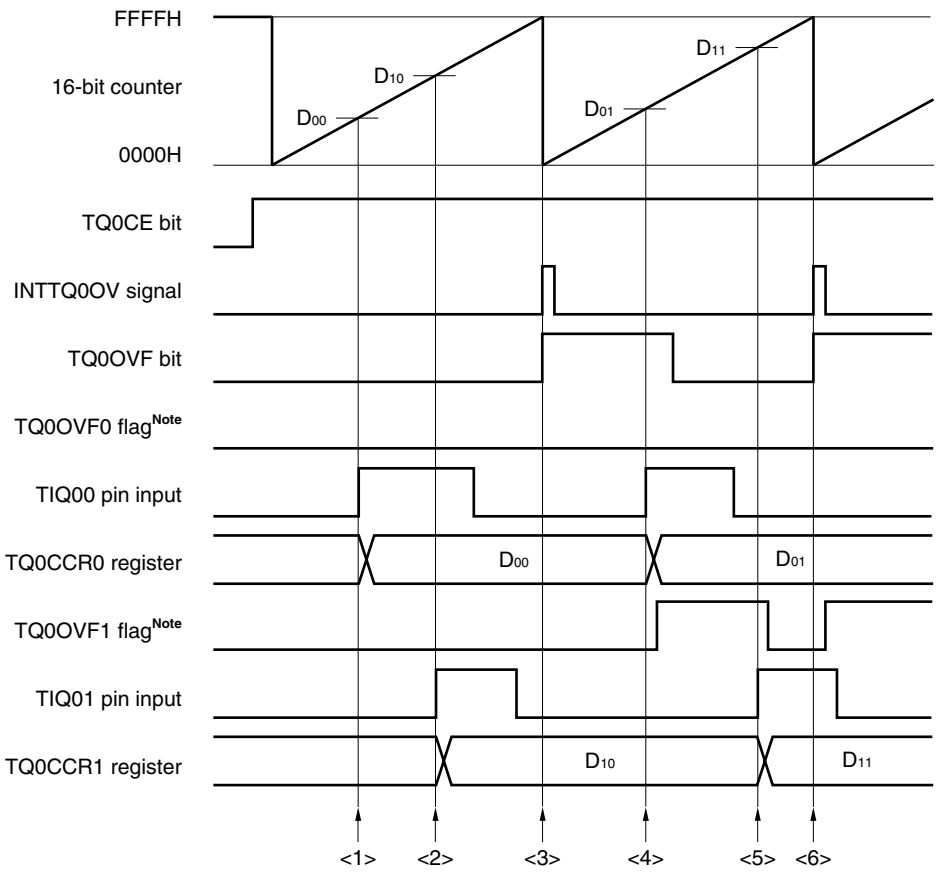
<4> Read the TQ0CCR0 register.
Read the TQ0OVF0 flag. If the TQ0OVF0 flag is 1, clear it to 0.
Because the TQ0OVF0 flag is 1, the pulse width can be calculated by (10000H + $D_{01} - D_{00}$).

<5> Read the TQ0CCR1 register.
Read the TQ0OVF1 flag. If the TQ0OVF1 flag is 1, clear it to 0 (the TQ0OVF0 flag is cleared in <4>, and the TQ0OVF1 flag remains 1).
Because the TQ0OVF1 flag is 1, the pulse width can be calculated by (10000H + $D_{11} - D_{10}$) (correct).

<6> Same as <3>

(2/2)

Example when two capture registers are used (without using overflow interrupt)



Note The TQ0OVF0 and TQ0OVF1 flags are set on the internal RAM by software.

<1> Read the TQ0CCR0 register (setting of the default value of the TIQ00 pin input).

<2> Read the TQ0CCR1 register (setting of the default value of the TIQ01 pin input).

<3> An overflow occurs. Nothing is done by software.

<4> Read the TQ0CCR0 register.

Read the overflow flag. If the overflow flag is 1, set only the TQ0OVF1 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 TQ0CCR1 register.

Read the overflow flag. Because the overflow flag is cleared in <4>, 0 is read.

Read the TQ0OVF1 flag. If the TQ0OVF1 flag is 1, clear it to 0.

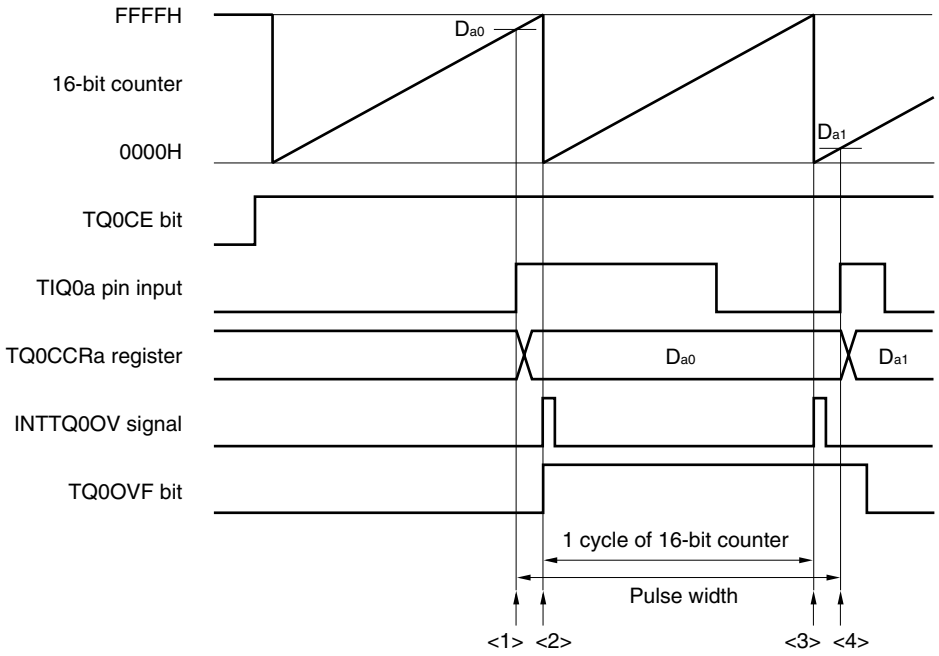
Because the TQ0OVF1 flag is 1, the pulse width can be calculated by (10000H + $D_{11} - D_{10}$) (correct).

<6> Same as <3>

(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



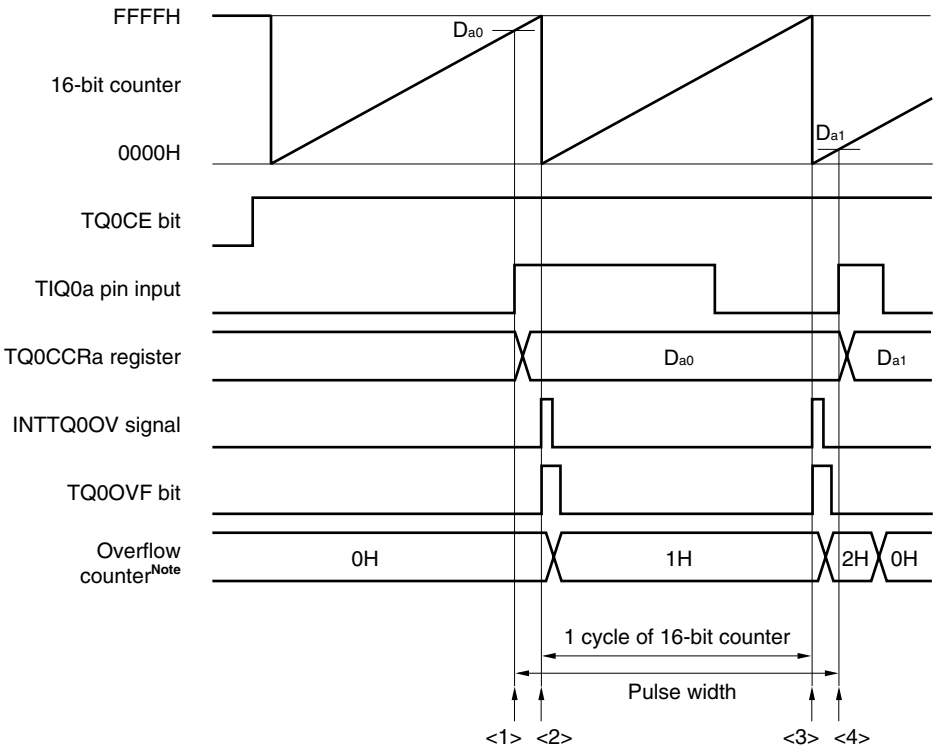
The following problem may occur when a long pulse width is measured in the free-running timer mode.

<1> Read the TQ0CCRa register (setting of the default value of the TIQ0a 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 TQ0CCRa 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 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 below.

Example when capture trigger interval is long



Note The overflow counter is set arbitrarily by software on the internal RAM.

<1> Read the TQ0CCRa register (setting of the default value of the TIQ0a 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 TQ0CCRa 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).

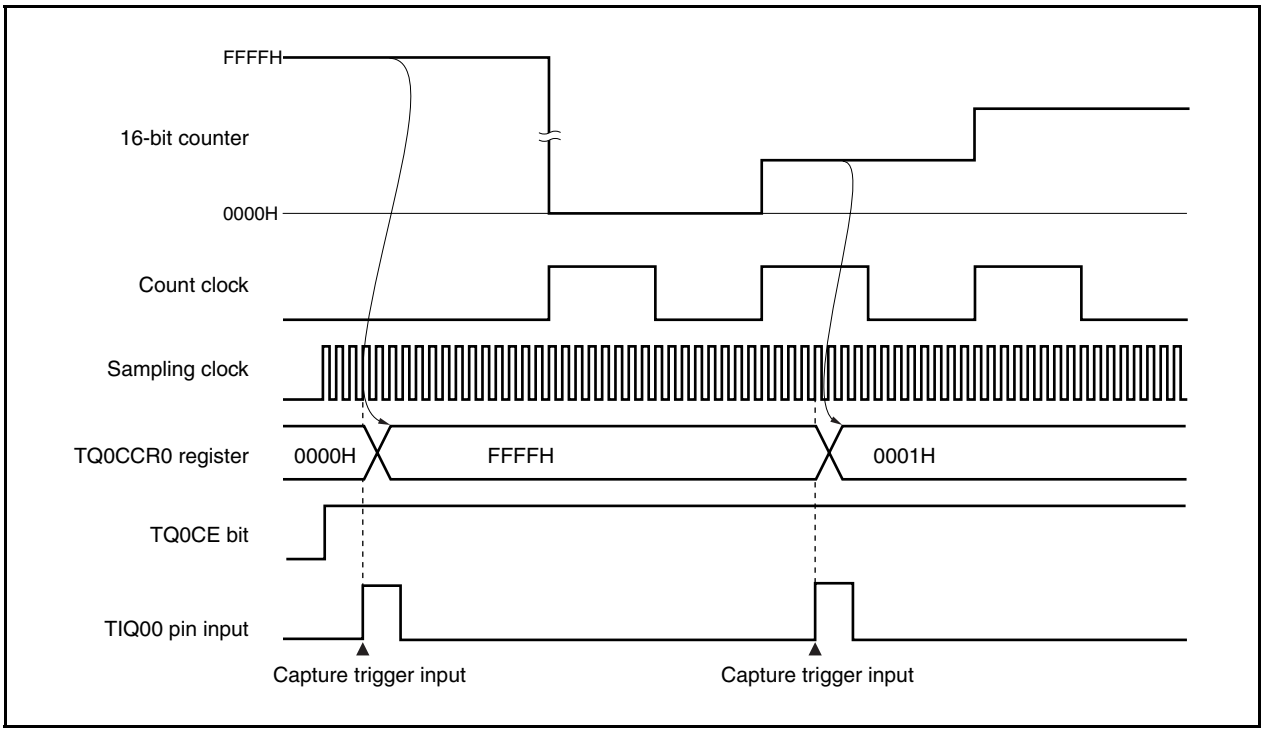
Remark a = 0 to 3

(e) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TQnOVF bit to 0 with the CLR instruction after reading the TQnOVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TQnOPT0 register after reading the TQnOVF bit when it is 1.

(3) Note on capture operation

If the capture operation is used and if a slow clock is selected as the count clock, FFFFH, not 0000H, may be captured to the TQ0CCRa register if the capture trigger is input immediately after the TQ0CTL0.TQ0CE bit is set to 1 (a = 0 to 3).



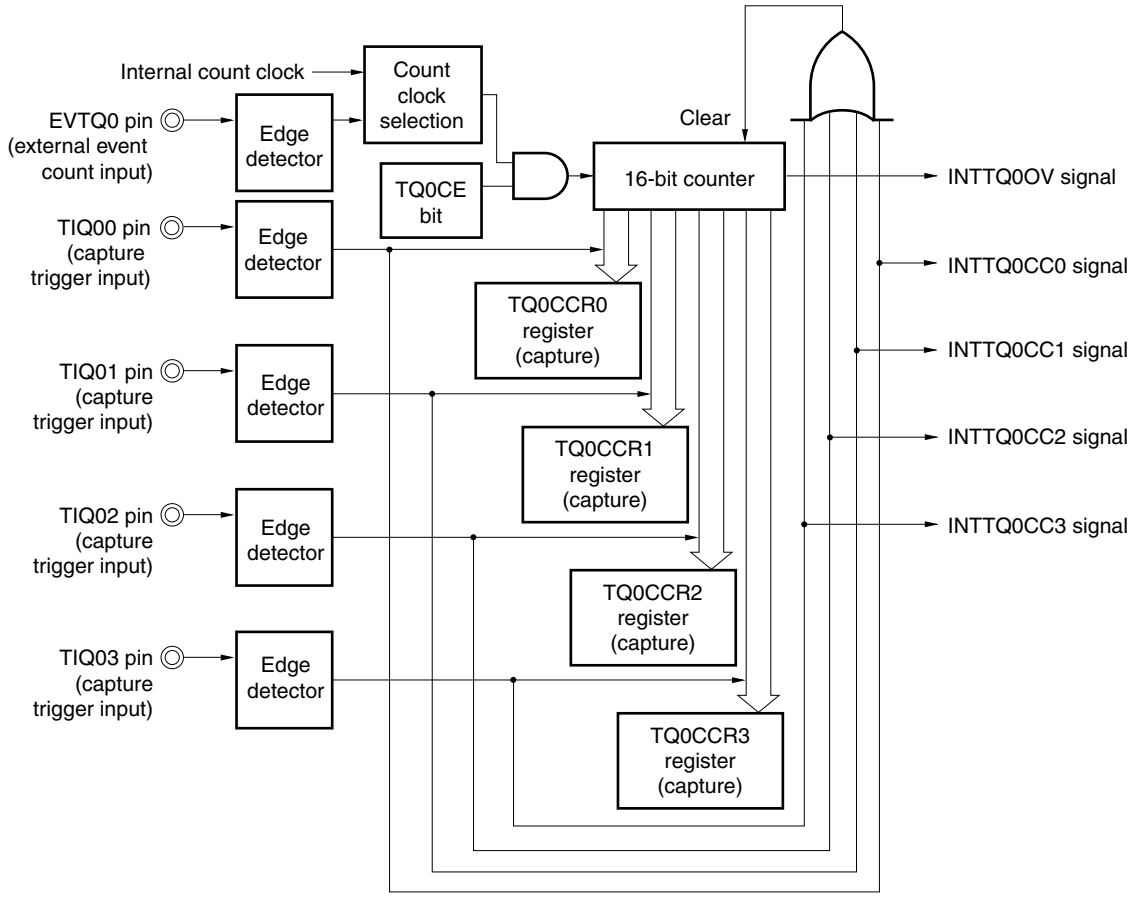
7.6.7 Pulse width measurement mode (TQ0MD2 to TQ0MD0 bits = 110)

In the pulse width measurement mode, 16-bit timer/event counter Q starts counting when the TQ0CTL0.TQ0CE bit is set to 1. Each time the valid edge input to the TIQ0a pin has been detected, the count value of the 16-bit counter is stored in the TQ0CCRa register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TQ0CCRa register after a capture interrupt request signal (INTTQ0CCa) occurs.

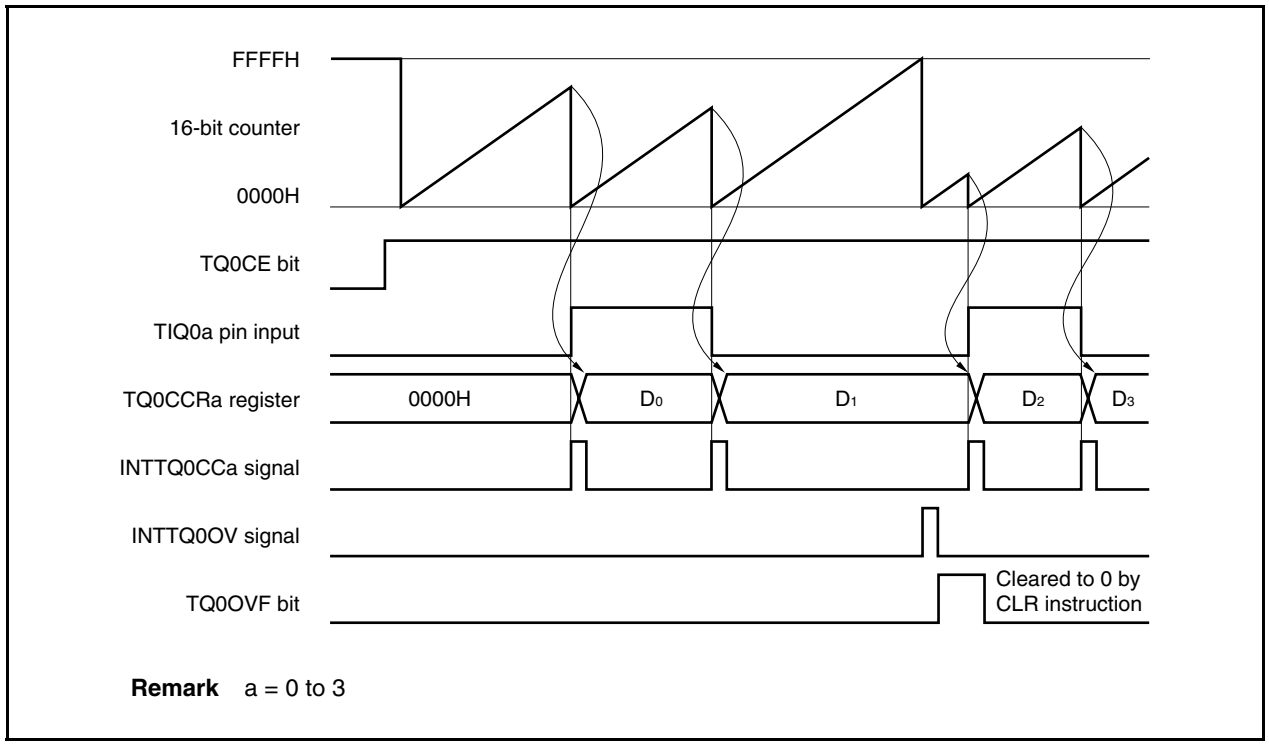
As shown in Figure 7-41, select either of the TIQ00 to TIQ03 pins as the capture trigger input pin. Specify “No edge detection” by using the TQ0IOC1 register for the unused pins.

Figure 7-40. Configuration in Pulse Width Measurement Mode



Remark a = 0 to 3

Figure 7-41. Basic Timing in Pulse Width Measurement Mode



Remark a = 0 to 3

When the TQ0CE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIQ0a pin is later detected, the count value of the 16-bit counter is stored in the TQ0CCRa register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTTQ0CCa) is generated.

The pulse width is calculated as follows.

<R> Pulse width = Captured value × Count clock cycle

If the valid edge is not input to the TIQ0m pin even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTQ0OV) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TQ0OPT0.TQ0OVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction via software.

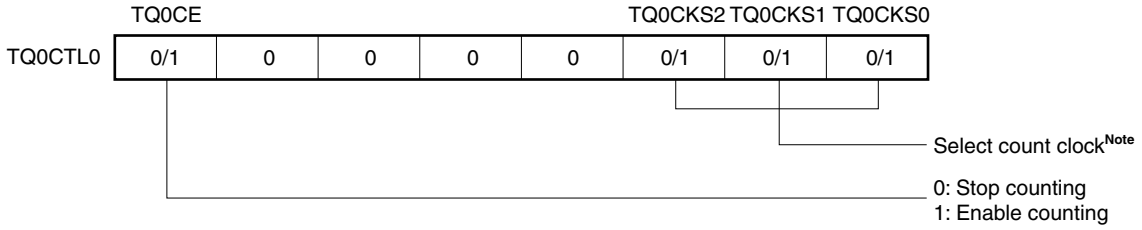
If the overflow flag is set to 1, the pulse width can be calculated as follows.

<R> Pulse width = (10000H × Number of times for which TQ0OVF bit is set to 1 + Captured value) × Count clock cycle

Remark a = 0 to 3

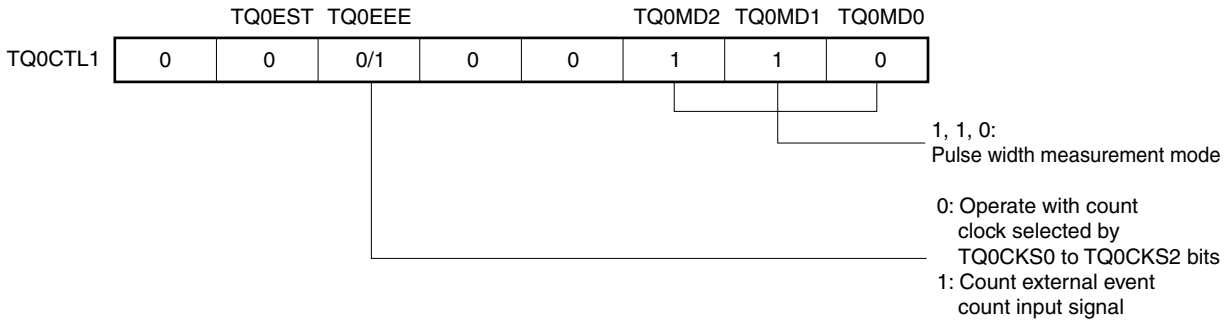
Figure 7-42. Register Setting in Pulse Width Measurement Mode (1/2)

(a) TMQ0 control register 0 (TQ0CTL0)

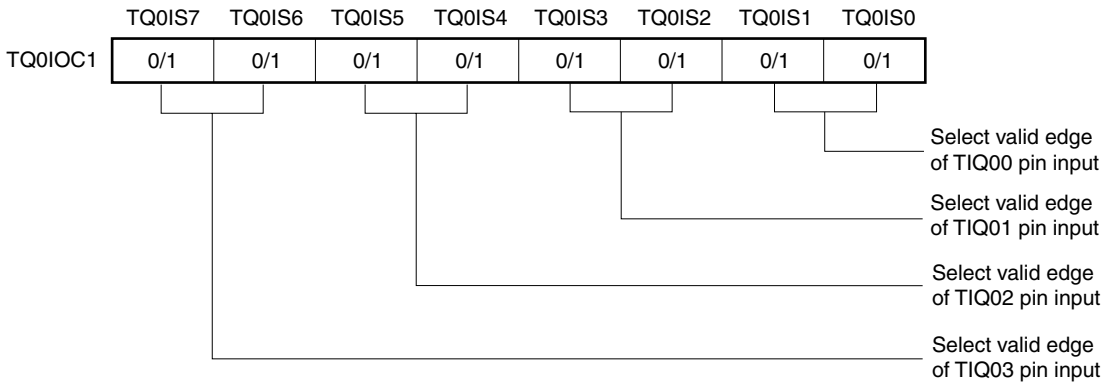


Note Setting is invalid when the TQ0CTL1.TQ0EEE bit = 1.

(b) TMQ0 control register 1 (TQ0CTL1)



(c) TMQ0 I/O control register 1 (TQ0IOC1)



(d) TMQ0 I/O control register 2 (TQ0IOC2)

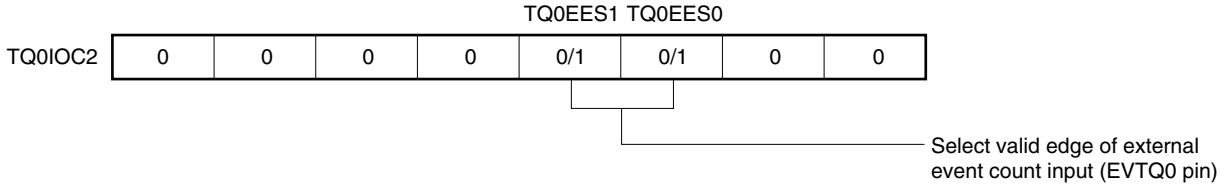


Figure 7-42. Register Setting in Pulse Width Measurement Mode (2/2)

(e) TMQ0 option register 0 (TQ0OPT0)

	TQ0CCS3	TQ0CCS2	TQ0CCS1	TQ0CCS0				TQ0OVF
TQ0OPT0	0	0	0	0	0	0	0	0/1

Overflow flag

(f) TMQ0 counter read buffer register (TQ0CNT)

The value of the 16-bit counter can be read by reading the TQ0CNT register.

(g) TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3)

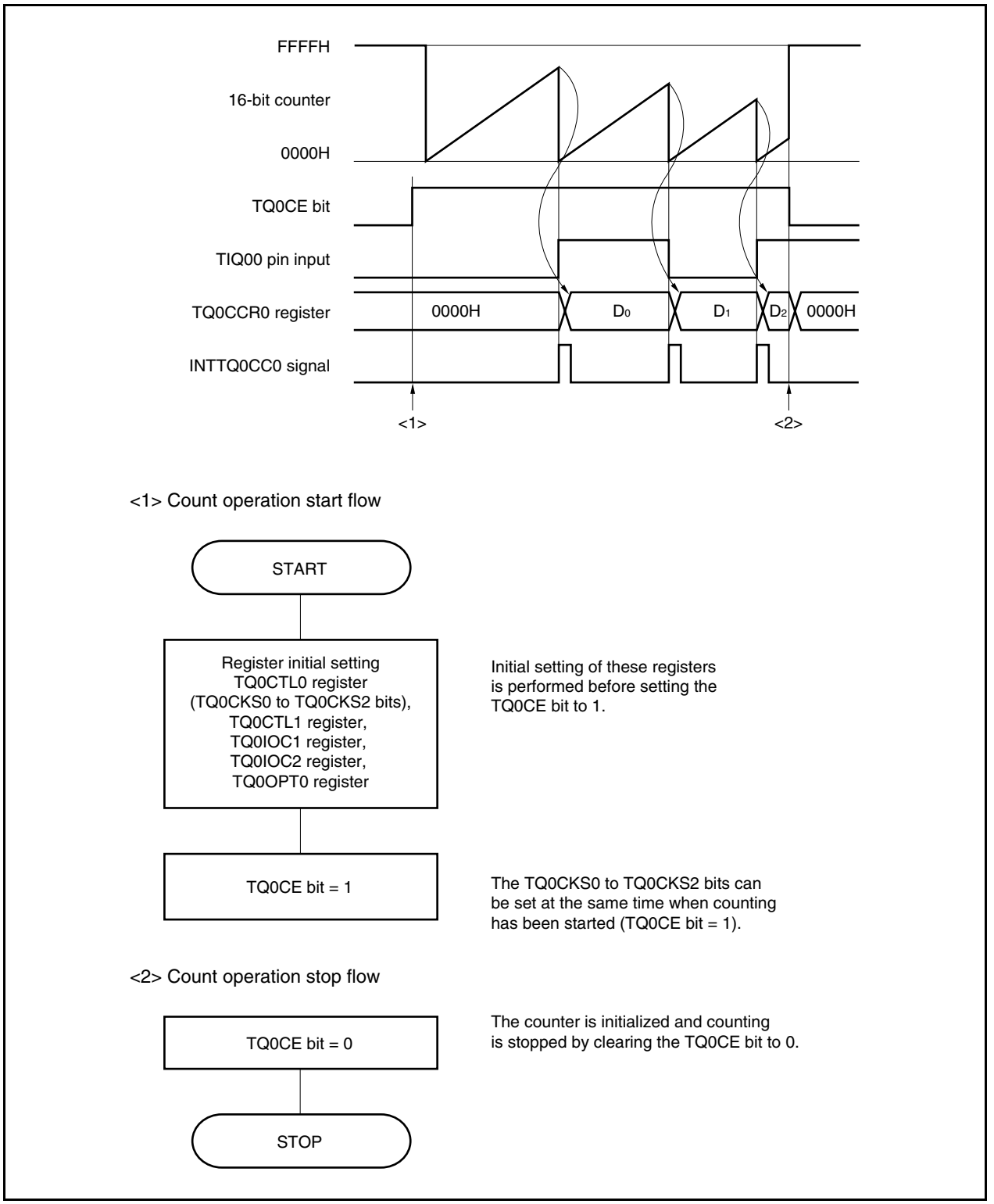
These registers store the count value of the 16-bit counter when the valid edge input to the TIQ0a pin is detected.

Remarks 1. TMQ0 I/O control register 0 (TQ0IOC0) is not used in the pulse width measurement mode.

2. a = 0 to 3

(1) Operation flow in pulse width measurement mode

Figure 7-43. Software Processing Flow in Pulse Width Measurement Mode



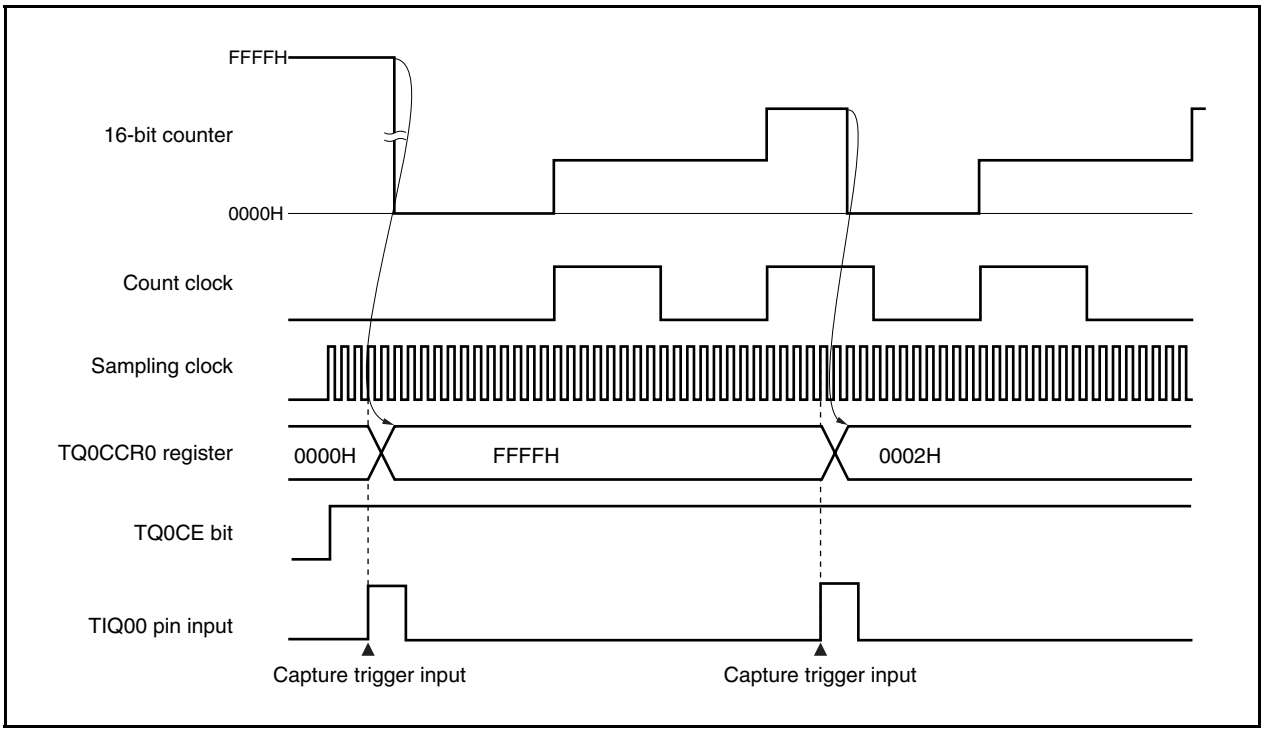
(2) Operation timing in pulse width measurement mode

(a) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TQ0OVF bit to 0 with the CLR instruction after reading the TQ0OVF bit when it is 1 and by writing 8-bit data (bit 0 is 0) to the TQ0OPT0 register after reading the TQ0OVF bit when it is 1.

(3) Note

If a slow clock is selected as the count clock, FFFFH, not 0000H, may be captured to the TQ0CCRa register if the capture trigger is input immediately after the TQ0CTL0.TQ0CE bit is set to 1 (a = 0 to 3).



CHAPTER 8 16-BIT INTERVAL TIMER M (TMM)

8.1 Overview

- 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 counter operation.)
- Compare match interrupt × 1

Timer M supports only the clear & start mode. The free-running timer mode is not supported.

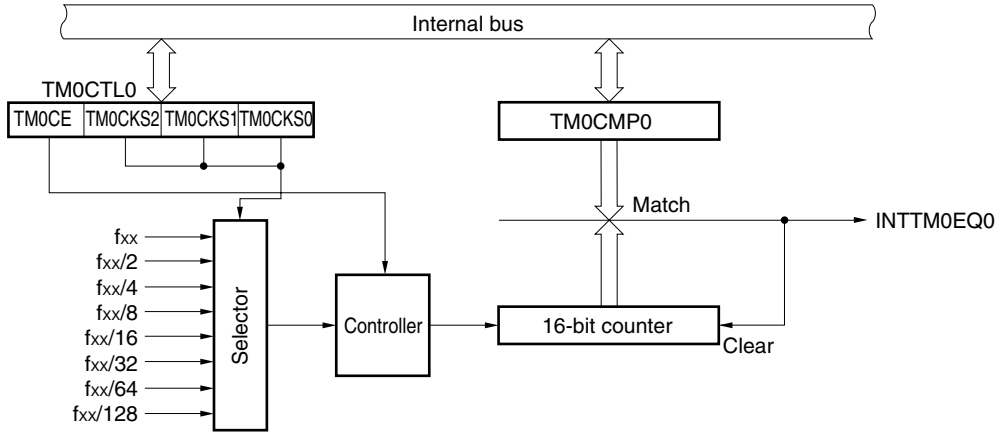
8.2 Configuration

TMM0 includes the following hardware.

Table 8-1. Configuration of TMM0

Item	Configuration
Timer register	16-bit counter
Register	TMM0 compare register 0 (TM0CMP0)
Control register	TMM0 control register 0 (TM0CTL0)

Figure 8-1. Block Diagram of TMM0



Remark fxx: Peripheral clock frequency

(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) TMM0 compare register 0 (TM0CMP0)

The TM0CMP0 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 TM0CMP0 register by software.

Rewriting the TM0CMP0 register is prohibited during TMM0 operation (TM0CTL0.TM0CE bit = 1).

After reset: 0000H R/W Address: FFFFF544H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TM0CMP0																

8.3 Control Register

(1) TMM0 control register 0 (TM0CTL0)

The TM0CTL0 register is an 8-bit register that controls the TMM0 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 TM0CTL0 register by software.

After reset: 00H R/W Address: FFFFF540H

	<7>	6	5	4	3	2	1	0
TM0CTL0	TM0CE	0	0	0	0	TM0CKS2	TM0CKS1	TM0CKS0

TM0CE	Internal clock operation enable/disable specification
0	TMM0 operation disabled (16-bit counter reset asynchronously)
1	TMM0 operation enabled. Start operation clock supply. Start TMM0 operation.
The internal clock control and internal circuit reset for TMM0 are performed asynchronously with the TM0CE bit. When the TM0CE bit is cleared to 0, the internal clock of TMM0 is stopped (fixed to low level) and 16-bit counter is reset asynchronously.	

TM0CKS2	TM0CKS1	TM0CKS0	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	$f_{XX}/128$

Cautions 1. Set the TM0CKS2 to TM0CKS0 bits when the TM0CE bit = 0. However, when changing the value of TM0CE from 0 to 1, it is impossible to set the value of the TM0CKS2 to TM0CKS0 bits simultaneously.

2. Be sure to clear bits 3 to 6 to "0".

Remark f_{XX}: Peripheral clock frequency

8.4 Operation

8.4.1 Interval timer mode

In the interval timer mode, an interrupt request signal (INTTM0EQ0) is generated at the interval set by the TM0CMP0 register if the TM0CTL0.TM0CE bit is set to 1.

Figure 8-2. Configuration of Interval Timer

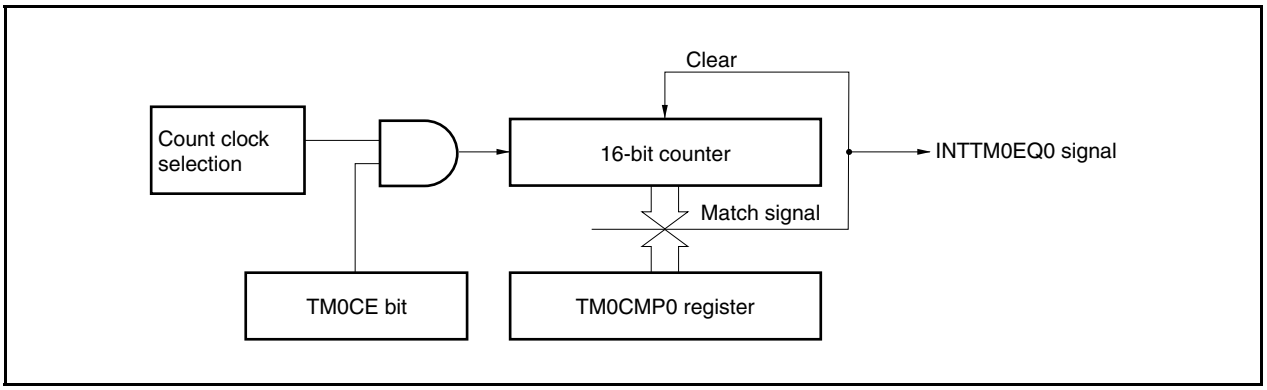
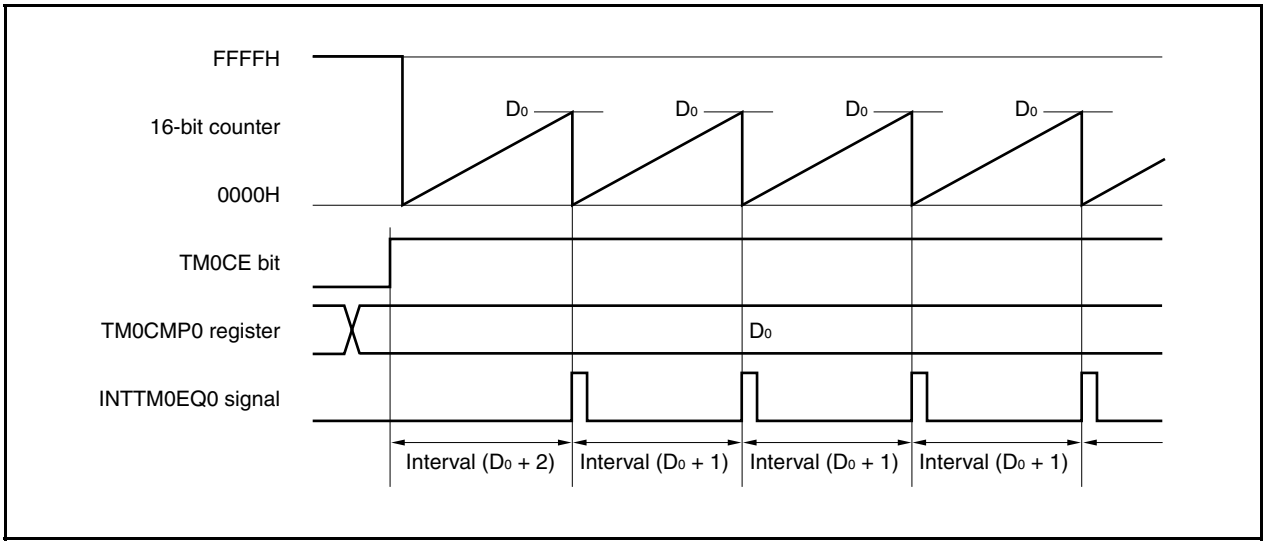


Figure 8-3. Basic Timing of Operation in Interval Timer Mode



When the TM0CE 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.

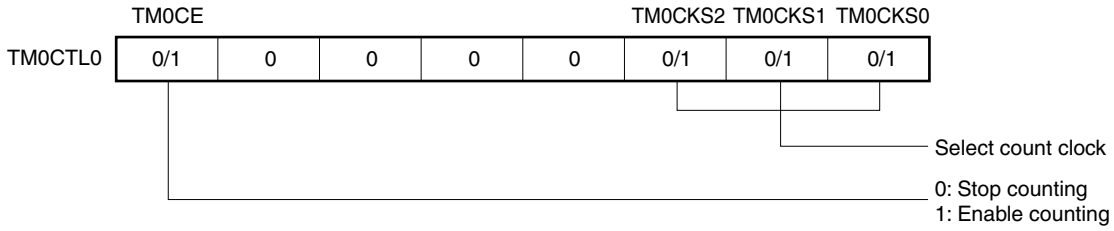
When the count value of the 16-bit counter matches the value of the TM0CMP0 register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTM0EQ0) is generated.

The interval can be calculated by the following expression.

Interval = (Set value of TM0CMP0 register + 1) × Count clock cycle

Figure 8-4. Register Setting for Interval Timer Mode Operation

(a) TMM0 control register 0 (TM0CTL0)



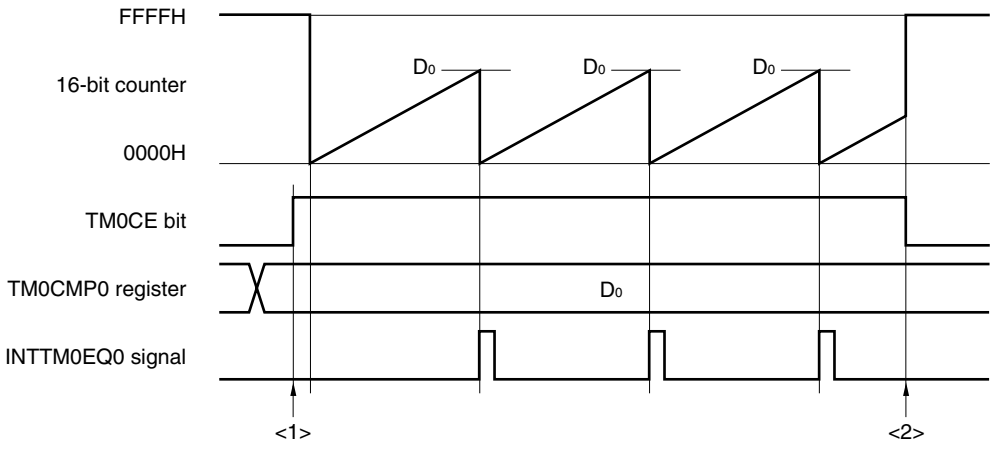
(b) TMM0 compare register 0 (TM0CMP0)

If the TM0CMP0 register is set to D_0, the interval is as follows.

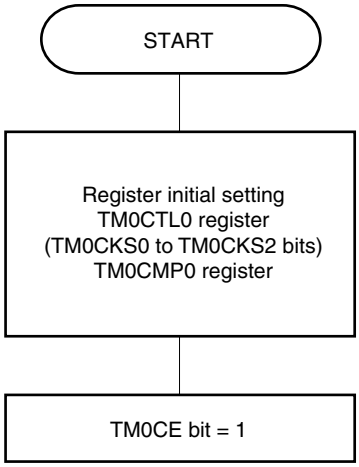
Interval = (D_0 + 1) × Count clock cycle

(1) Interval timer mode operation flow

Figure 8-5. Software Processing Flow in Interval Timer Mode



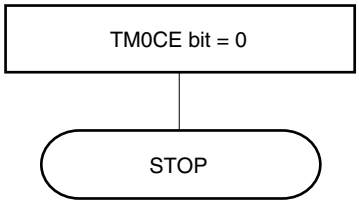
<1> Count operation start flow



Initial setting of these registers is performed before setting the TM0CE bit to 1.

The TM0CKS0 to TM0CKS2 bits cannot be set at the same time when counting has been started (TM0CE bit = 1).

<2> Count operation stop flow

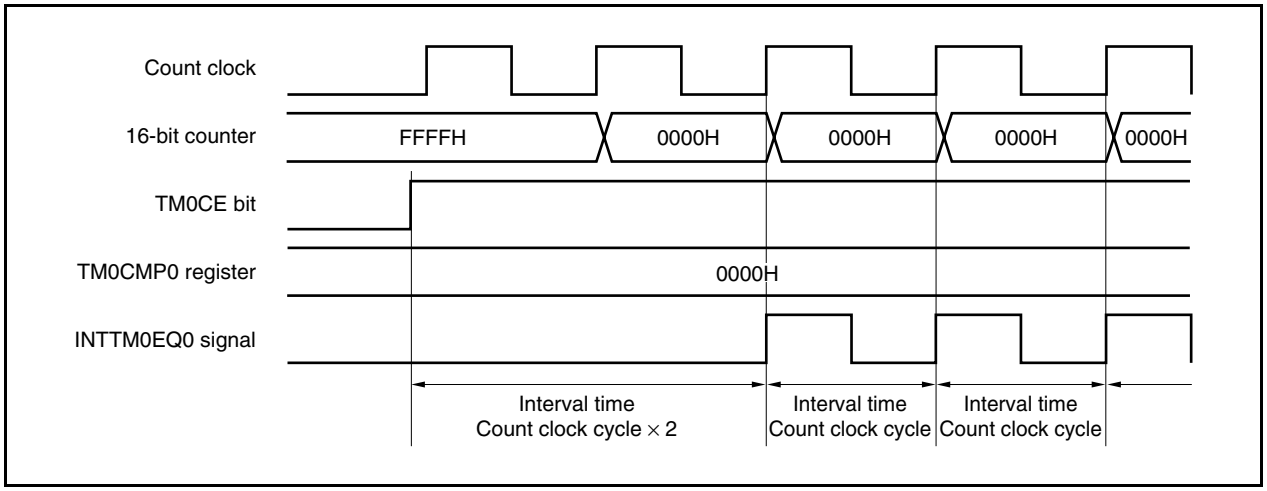


The counter is initialized and counting is stopped by clearing the TM0CE bit to 0.

(2) Interval timer mode operation timing

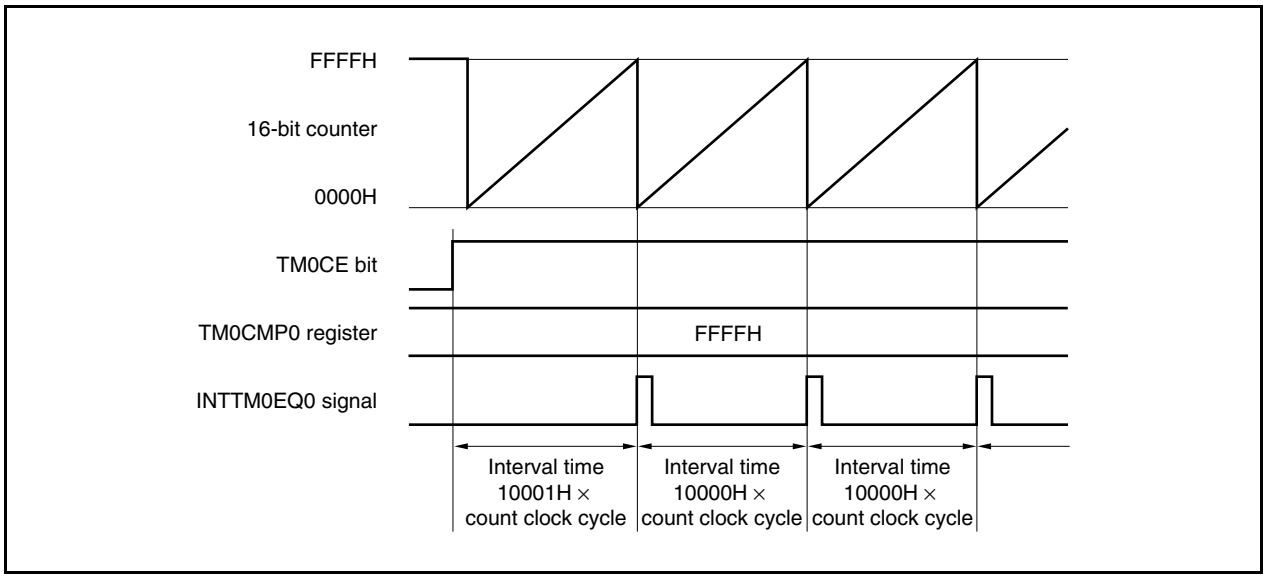
(a) Operation if TM0CMP0 register is set to 0000H

If the TM0CMP0 register is set to 0000H, the INTTM0EQ0 signal is generated at each count clock. The value of the 16-bit counter is always 0000H.



(b) Operation if TM0CMP0 register is set to FFFFH

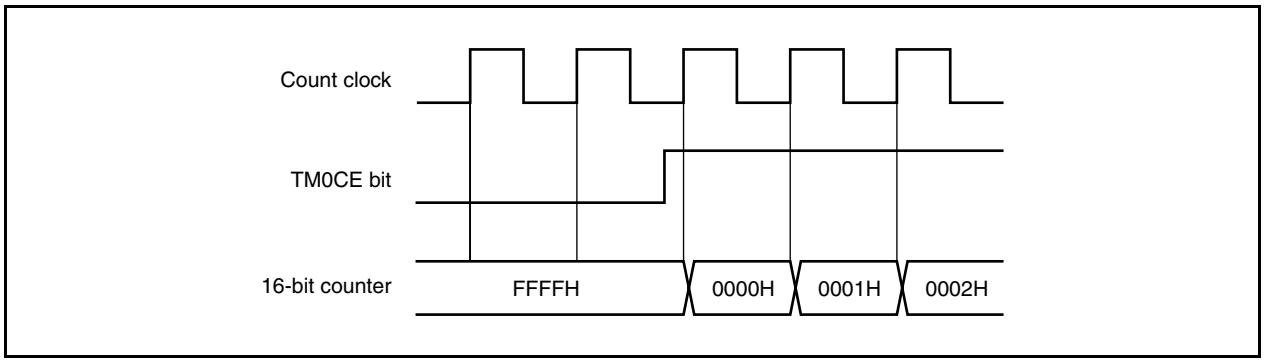
If the TM0CMP0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTM0EQ0 signal is generated.



8.5 Cautions

(1) Error on starting timer

It takes one clock to generate the first compare match interrupt request signal (INTTM0EQ0) after the TM0CTL0.TM0CE bit is set to 1 and TMM0 is started. This is because the value of the 16-bit counter is FFFFH when the TM0CE bit = 0 and TMM0 is started asynchronously to the count clock.



(2) Rewriting the TM0CMP0 and TM0CTL0 registers is prohibited while TMM0 is operating.

If these registers are rewritten while the TM0CTL0.TM0CE bit is 1, the operation cannot be guaranteed.

If they are rewritten by mistake, clear the TM0CE bit to 0, and re-set the registers.

CHAPTER 9 MOTOR CONTROL FUNCTION

9.1 Functional Overview

Timer Q1 (TMQ1) and the TMQ1 option (TMQOP1) can be used as an inverter function that controls a motor. It performs a tuning operation with timer P1 (TMP1) and A/D conversion of A/D converters 0 and 1 can be started when the value of TMQ1 matches the value of TMP1. 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 TMP1)
- Cycle setting function (cycle can be changed during operation of crest or valley interrupt)
- Compare register rewriting: Anytime rewrite, batch rewrite, or intermittent rewrite (selectable during TMQ1 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 (TOQ1OFF, TOP2OFF, TOP3OFF, TOQH0OFF)
  - At main clock oscillation stop detection by clock monitor function

9.2 Configuration

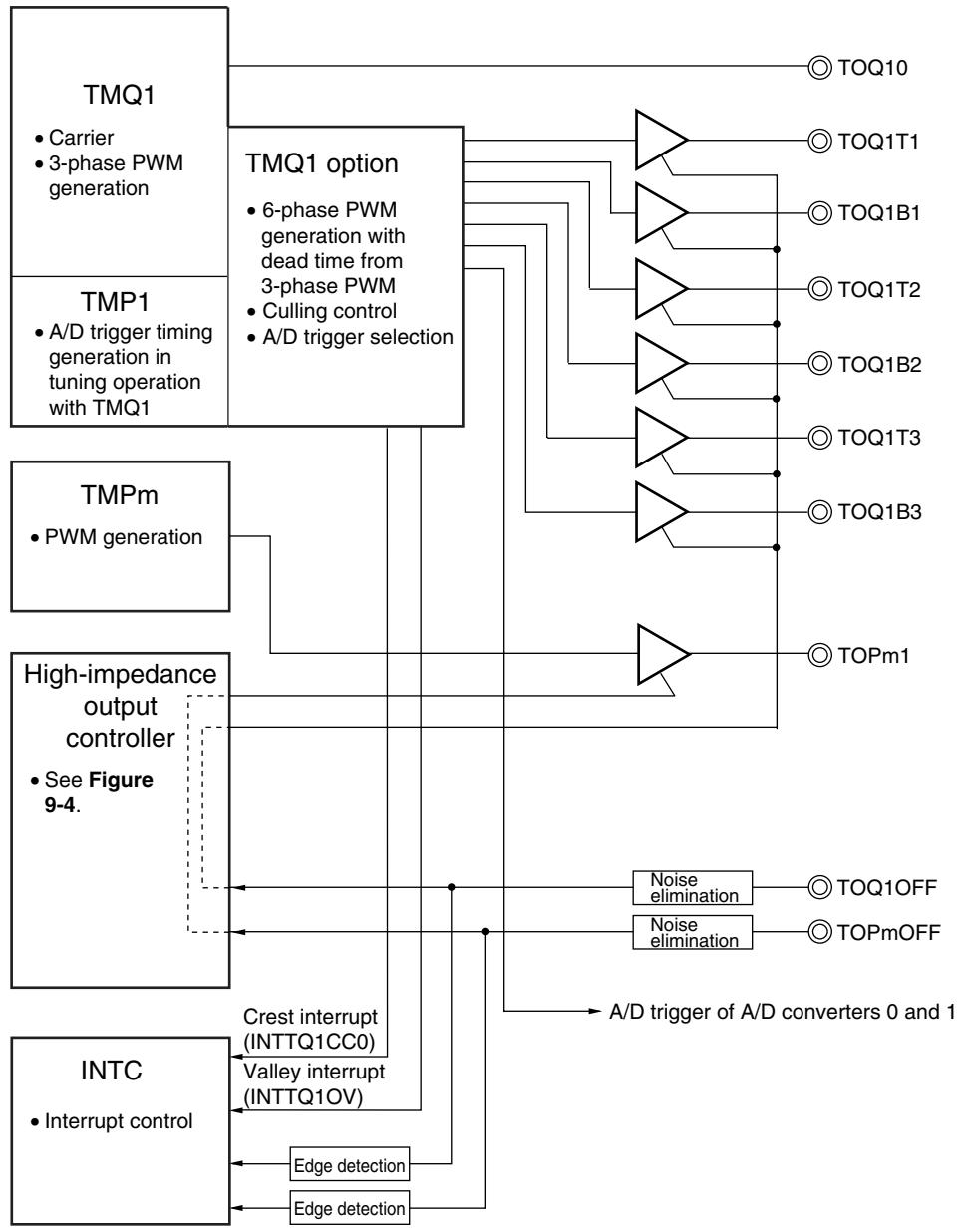
The motor control function includes the following hardware.

Item	Configuration
Timer register	Dead-time counter m
Compare register	TMQ1 dead-time compare register (TQ1DTC register)
Control registers	TMQ1 option register 0 (TQ1OPT0) TMQ1 option register 1 (TQ1OPT1) TMQ1 option register 2 (TQ1OPT2) TMQ1 option register 3 (TQ1OPT3) TMQ1 I/O control register 3 (TQ1IOC3) High-impedance output control registers 0, 1 (HZA1CTL0, HZAaCTL1)

Remark m = 0 to 3
a = 0, 1

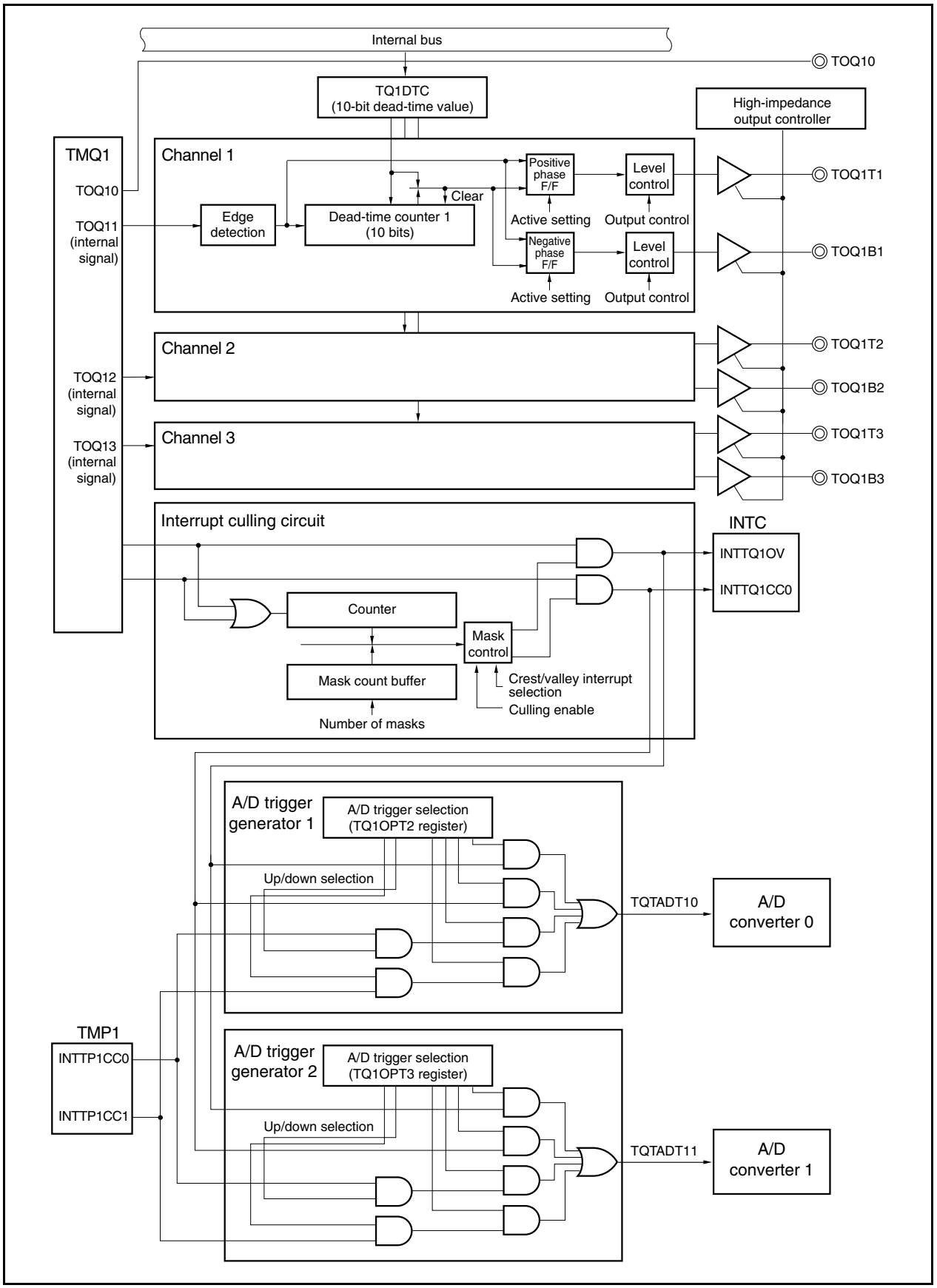
- 6-phase PWM output can be produced with dead time by using the output of TMQ1 (TOQ11, TOQ12, TOQ13)
- The output level of the 6-phase PWM output can be set individually.
- The 16-bit timer/counter of TMQ1 counts up/down triangular waves. When the timer/counter underflows and when a cycle match occurs, an interrupt is generated. Interrupt generation, however, can be suppressed up to 31 times.
- TMP1 can execute counting at the same time as TMQ1 (timer tuning operation function). TMP1 can be set in four ways as it can generate two types of A/D trigger sources (INTTP1CC0 and INTTP1CC1), and two types of interrupts: on underflow interrupt (INTTQ1OV) and cycle match interrupt (INTTQ1CC0).

Figure 9-1. Block Diagram of Motor Control



Remark m = 2, 3

Figure 9-2. TMQ1 Option



(1) TMQ1 dead-time compare register (TQ1DTC)

The TQ1DTC register is a 10-bit compare register that specifies a dead-time value.

Rewriting this register is prohibited when the TQ1CTL0.TQ1CE bit = 1.

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

Reset sets this register to 0000H.

<R> **Caution To generate the dead-time period, set the TQ1DTC register to 1 or more.**
When the operation is stopped (TQ1CTL0.TQ1CE bit = 0), the dead-time period is not generated and the output level of the TOQ1T1 to TOQ1T3 pins and TOQ1B1 to TOQ1B3 pins will be in the initial status. For the system protection, therefore, before operation is being stopped, set the TOQ1T1 to TOQ1T3 and TOQ1B1 to TOQ1B3 pins to the high impedance state, or set the output level of pins and switch them to the port mode.
If a dead time period is not needed, set the TQ1DTC register to 0.

After reset: 0000H R/W Address: FFFFF624H

	15	10	9	0
TQ1DTC	000000		TQ1DTC9 to TQ1DTC0	

(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 TOQ1m output signal by TMQ1, and are cleared or stopped when their count value matches the value of the TQ1DTC register. The count clock of these counters is the same as that set by the TQ1CTL0.TQ1CKS2 to TQ1CTL0.TQ1CKS0 bits of TMQ1.

Remarks 1. The operation differs when the TQ1OPT2.TQ1DTM bit = 1. For details, see **9.4.2 (4) Automatic dead-time width narrowing function (TQ1OPT2.TQ1DTM bit = 1)**.

2. m = 1 to 3

9.3 Control Registers

(1) TMQ1 option register 0 (TQ1OPT0)

The TQ1OPT0 register is an 8-bit register that controls the timer Q1 option function.

This register can be read or written in 8-bit or 1-bit units. However, the TQ1CUF bit is read-only.

Reset sets this register to 00H.

Caution The TQ1CMS and TQ1CUF bits can be set only in the 6-phase PWM output mode. Be sure to clear these bits to 0 when TMQ1 is used alone.

After reset: 00H R/W Address: FFFFF605H

	7	6	5	4	3	<2>	<1>	<0>
TQ1OPT0	0	0	0	0	0	TQ1CMS	TQ1CUF	TQ1OVF[Note]

TQ1CMS	Compare register rewrite mode selection
0	Batch rewrite mode (transfer operation)
1	Anytime rewrite mode

- The TQ1CMS bit is valid only when the 6-phase PWM output mode is set (when the TQ1CTL1.TQ1MD2 to TQ1CTL1.TQ1MD0 bits = 111). Clear the TQ1CMS bit to 0 in any other mode.
- The TQ1CMS bit can be rewritten while the timer is operating (when the TQ1CTL0.TQ1CE bit = 1).
- The following compare registers are rewritten in the batch write mode. TQ1CCR0 to TQ1CCR3, TP1CCR0, TP1CCR1, and TQ1OPT1 registers

TQ1CUF	Count-up/count-down flag of timer Q1
0	Timer Q1 is counting up.
1	Timer Q1 is counting down.

The TQ1CUF bit is valid only when the 6-phase PWM output mode is set (when the TQ1CTL1.TQ1MD2 to TQ1CTL1.TQ1MD0 bits = 111).

Note For details of the TQ1OVF bit, see **CHAPTER 7 16-BIT TIMER/EVENT COUNTER Q (TMQ)**.

Caution Be sure to clear bits 7 to 3 to "0".

(2) TMQ1 option register 1 (TQ1OPT1)

The TQ1OPT1 register is an 8-bit register that controls the interrupt request signal generated by the timer Q1 option function.

<R> The TQ1OPT1 register generates the signals output to the interrupt culling circuit, A/D trigger generator 1, and A/D trigger generator 2 shown in Figure 9-2.

This register can be rewritten when the TQ1CTL0.TQ1CE bit is 1.

Two rewrite modes (batch write mode and anytime write mode) can be selected, depending on the setting of the TQ1OPT0.TQ1CMS 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: FFFFF620H

	<7>	<6>	5	4	3	2	1	0
TQ1OPT1	TQ1ICE	TQ1IOE	0	TQ1ID4	TQ1ID3	TQ1ID2	TQ1ID1	TQ1ID0

<R>

TQ1ICE	Crest interrupt (INTTQ1CC0 signal) enable[Note]
0	Do not use INTTQ1CC0 signal (do not use it as count signal for interrupt culling).
1	Use INTTQ1CC0 signal (use it as count signal for interrupt culling).

<R>

TQ1IOE	Valley interrupt (INTTQ1OV signal) enable[Note]
0	Do not use INTTQ1OV signal (do not use it as count signal for interrupt culling).
1	Use INTTQ1OV signal (use it as count signal for interrupt culling).

TQ1ID4	TQ1ID3	TQ1ID2	TQ1ID1	TQ1ID0	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)

<R>

Note When using the crest interrupt (INTTQ1CC0 signal) and the valley interrupt (INTTQ1OV signal) as the count signal for interrupt culling or as the A/D trigger signal, set the signal to be used to 1.

A/D trigger is generated at the culled interrupt timing.

(3) TMQ1 option register 2 (TQ1OPT2)

The TQ1OPT2 register is an 8-bit register that controls the timer Q1 option function.

This register can be rewritten when the TQ1CTL0.TQ1CE bit is 1. However, rewriting the TQ1DTM bit is prohibited when the TQ1CE 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: FFFFF621H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TQ1OPT2	TQ1RDE	TQ1DTM	TQ1ATM03	TQ1ATM02	TQ1AT03	TQ1AT02	TQ1AT01	TQ1AT00

TQ1RDE	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 TQ1OPT1 register.

TQ1DTM	Dead-time counter operation mode selection
0	Dead-time counter counts up normally and, if TOQ1m output of TMQ1 is at a narrow interval (TOQ1m output width < dead-time width), the dead-time counter is cleared and counts up again.
1	Dead-time counter counts up normally and, if TOQ1m output of TMQ1 is at a narrow interval (TOQ1m output width < dead-time width), the dead-time counter counts down and the dead-time control width is automatically narrowed.
Rewriting the TQ1DTM bit is disabled during timer operation. If it is rewritten by mistake, stop the timer operation by clearing the TQ1CE bit to 0, and re-set the TQ1DTM bit.	

Cautions 1. When using interrupt culling (the TQ1OPT1.TQ1ID4 to TQ1OPT1.TQ1ID0 bits are set to other than 00000), be sure to set the TQ1RDE 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 (TQ1RDE bit = 0), transfer is not performed normally.**

<R> **2. To generate the dead-time period, set the TQ1DTC register to 1 or more.**

When the operation is stopped (TQ1CTL0.TQ1CE bit = 0), the dead-time period is not generated and the output level of the TOQ1T1 to TOQ1T3 pins and TOQ1B1 to TOQ1B3 pins will be in the initial status. For the system protection, therefore, before operation is being stopped, set the TOQ1T1 to TOQ1T3 and TOQ1B1 to TOQ1B3 pins to the high impedance state, or set the output level of pins and switch them to the port mode.

If a dead time period is not needed, set the TQ1DTC register to 0.

Remark m = 1 to 3

(2/2)

TQ1ATM03	TQ1ATM03 mode selection
0	Output A/D trigger signal (TQTADT10) for INTTP1CC1 interrupt while dead-time counter is counting up.
1	Output A/D trigger signal (TQTADT10) for INTTP1CC1 interrupt while dead-time counter is counting down.

TQ1ATM02	TQ1ATM02 mode selection
0	Output A/D trigger signal (TQTADT10) for INTTP1CC0 interrupt while dead-time counter is counting up.
1	Output A/D trigger signal (TQTADT10) for INTTP1CC0 interrupt while dead-time counter is counting down.

TQ1AT03Note	A/D trigger output control 3
0	Disable output of A/D trigger signal (TQTADT10) for INTTP1CC1 interrupt.
1	Enable output of A/D trigger signal (TQTADT10) for INTTP1CC1 interrupt.

TQ1AT02Note	A/D trigger output control 2
0	Disable output of A/D trigger signal (TQTADT10) for INTTP1CC0 interrupt.
1	Enable output of A/D trigger signal (TQTADT10) for INTTP1CC0 interrupt.

TQ1AT01Note	A/D trigger output control 1
0	Disable output of A/D trigger signal (TQTADT10) for INTTQ1CC0 (crest interrupt).
1	Enable output of A/D trigger signal (TQTADT10) for INTTQ1CC0 (crest interrupt).

TQ1AT00Note	A/D trigger output control 0
0	Disable output of A/D trigger signal (TQTADT10) for INTTQ1OV (valley interrupt).
1	Enable output of A/D trigger signal (TQTADT10) for INTTQ1OV (valley interrupt).

Note For the setting of the TQ1AT03 to TQ1AT00 bits, see **CHAPTER 11 A/D CONVERTERS 0 AND 1**.

(4) TMQ1 option register 3 (TQ1OPT3)

The TQ1OPT3 register is an 8-bit register that controls the timer Q1 option function.

This register can be rewritten when the TQ1CTL0.TQ1CE 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: FFFFF623H

	7	6	<5>	<4>	<3>	<2>	<1>	<0>
TQ1OPT3	0	0	TQ1ATM13	TQ1ATM12	TQ1AT13	TQ1AT12	TQ1AT11	TQ1AT10

TQ1ATM13	TQ1ATM3 mode selection
0	Output A/D trigger signal (TQTADT11) of INTTP1CC1 interrupt while dead-time counter is counting up.
1	Output A/D trigger signal (TQTADT11) of INTTP1CC1 interrupt while dead-time counter is counting down.

TQ1ATM12	TQ1ATM2 mode selection
0	Output A/D trigger signal (TQTADT11) of INTTP1CC0 interrupt while dead-time counter is counting up.
1	Output A/D trigger signal (TQTADT11) of INTTP1CC0 interrupt while dead-time counter is counting down.

TQ1AT13Note	A/D trigger output control 3
0	Disable output of A/D trigger signal (TQTADT11) for INTTP1CC1 interrupt.
1	Enable output of A/D trigger signal (TQTADT11) for INTTP1CC1 interrupt.

TQ1AT12Note	A/D trigger output control 2
0	Disable output of A/D trigger signal (TQTADT11) for INTTP1CC0 interrupt.
1	Enable output of A/D trigger signal (TQTADT11) for INTTP1CC0 interrupt.

TQ1AT11Note	A/D trigger output control 1
0	Disable output of A/D trigger signal (TQTADT11) for INTTQ1CC0 interrupt (crest interrupt).
1	Enable output of A/D trigger signal (TQTADT11) for INTTQ1CC0 interrupt (crest interrupt).

TQ1AT10Note	A/D trigger output control 0
0	Disable output of A/D trigger signal (TQTADT11) for INTTQ1OV interrupt (valley interrupt).
1	Enable output of A/D trigger signal (TQTADT11) for INTTQ1OV interrupt (valley interrupt).

Note For the setting of the TQ1AT13 to TQ1AT10 bits, see **CHAPTER 11 A/D CONVERTERS 0 AND 1**.

(5) TMQ1 I/O control register 3 (TQ1IOC3)

The TQ1IOC3 register is an 8-bit register that controls the output of the timer Q1 option function.

To output from the TOQ1Tm pin, set the TQ1IOC0.TQ1OEm bit to 1 and then set the TQ1IOC3 register.

The TQ1IOC3 register can be rewritten only when the TQ1CTL0.TQ1CE bit is 0.

Rewriting each bit of the TQ1IOC3 register is prohibited when the TQ1CTL0.TQ1CE bit is 1; however the same value can be rewritten to each bit of the TQ1IOC3 register when the TQ1CTL0.TQ1CE 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 TQ1IOC3 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 TOQ1Tm pin by the TQ1IOC0 register.

After reset: A8H R/W Address: FFFFF622H

	<7>	<6>	<5>	<4>	<3>	<2>	1	0
TQ1IOC3	TQ1OLB3	TQ1OEB3	TQ1OLB2	TQ1OEB2	TQ1OLB1	TQ1OEB1	0	0

TQ1OLBm	Setting of TOQ1Bm pin output level (m = 1 to 3)
0	Disable inversion of output of TOQ1Bm pin
1	Enable inversion of output of TOQ1Bm pin

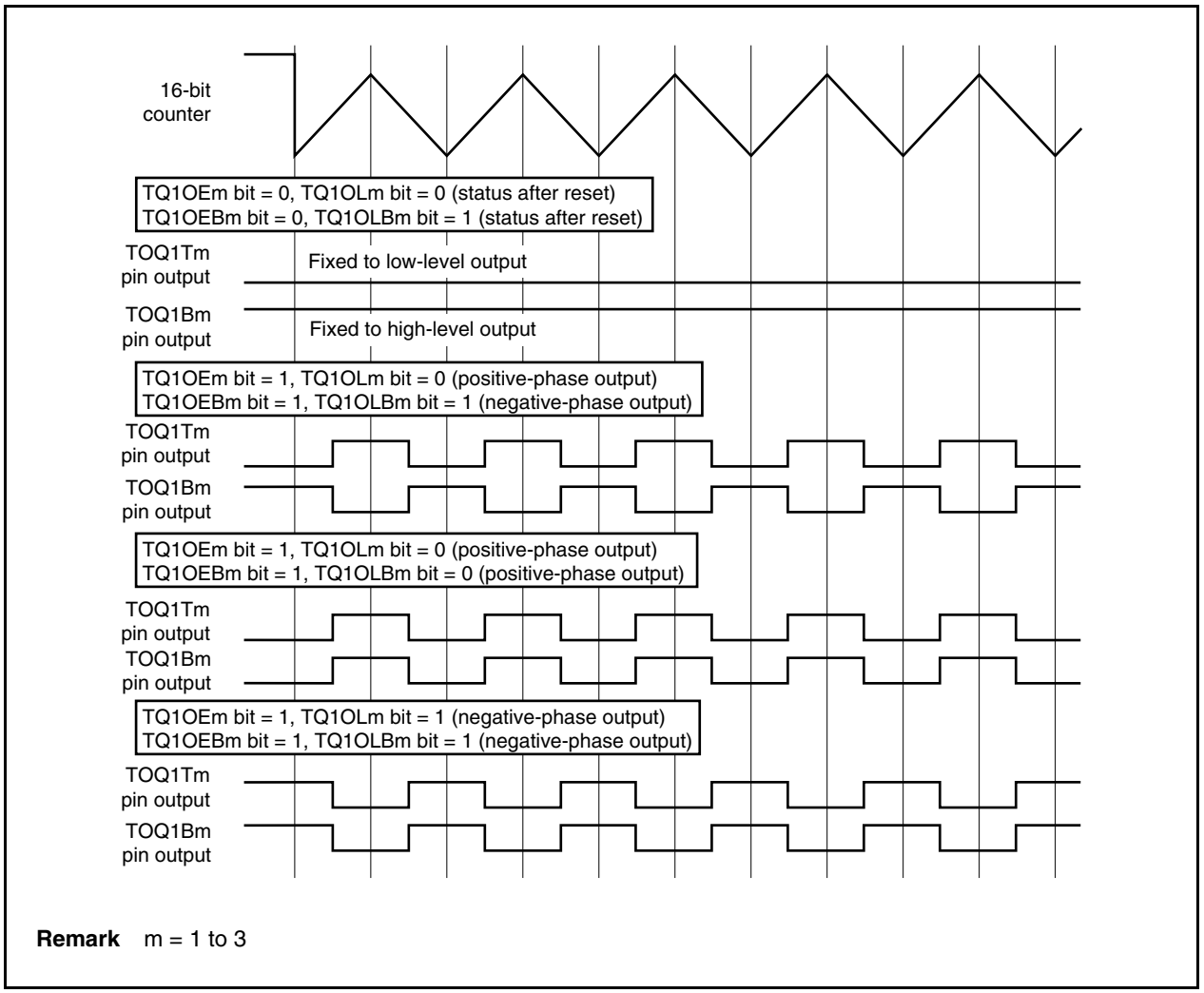
TQ1OEBm	TOQ1Bm pin output (m = 1 to 3)
0	Disable TOQ1Bm pin output. • When TQ1OLBm bit = 0, low level is output from TOQ1Bm pin. • When TQ1OLBm bit = 1, high level is output from TOQ1Bm pin.
1	Enable TOQ1Bm pin output.

(a) Output from TOQ1Tm and TOQ1Bm pins

The TOQ1Tm pin output is controlled by the TQ1IOC0.TQ1OLm and TQ1IOC0.TQ1OEm bits. The TOQ1Bm pin output is controlled by the TQ1IOC3.TQ1OLBm and TQ1IOC3.TQ1OEBm bits.

The timer output with each setting in the 6-phase PWM output mode is shown below.

<R> **Figure 9-3. Output Control of TOQ1Tm and TOQ1Bm Pins (Without Dead Time)**



Remark m = 1 to 3

Table 9-1. TOQ1Tm Pin Output

TQ1OLm Bit	TQ1OEm Bit	TQ1CE Bit	TOQ1Tm Pin Output
0	0	x	Low-level output
	1	0	Low-level output
		1	TOQ1Tm positive-phase output
1	0	x	High-level output
	1	0	High-level output
		1	TOQ1Tm negative-phase output

Remark m = 1 to 3

Table 9-2. TOQ1Bm Pin Output

TQ1OLBm Bit	TQ1OEBm Bit	TQ1CE Bit	TOQ1Bm Pin Output
0	0	x	Low-level output
	1	0	Low-level output
		1	TOQ1Bm positive-phase output
1	0	x	High-level output
	1	0	High-level output
		1	TOQ1Bm negative-phase output

Remark m = 1 to 3

(6) High-impedance output control registers 00, 01, 10, 11 (HZAyCTLn)

The HZAyCTLn 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 HZAyDCF1 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 (External Pin)	Control Register
When TOQH01 to TOQH03 are output	TOQH0OFF	HZA0CTL0
When TOP21 is output	TOP2OFF	HZA0CTL1
When TOQ1T1 to TOQ1T3 are output When TOQ1B1 to TOQ1B3 are output	TOQ1OFF	HZA1CTL0
When TOP31 is output	TOP3OFF	HZA1CTL1

Caution High-impedance control is performed only when a port pin is set to function as indicated in the above table.

(1/2)

After reset: 00H R/W Address: HZA0CTL0 FFFFF5F0H, HZA0CTL1 FFFFF5F1H, HZA1CTL0 FFFFF630H, HZA1CTL1 FFFFF631H

HZAyCTLn (n = 0, 1; y = 0, 1)

<7>	<6>	5	4	<3>	<2>	1	<0>
HZAyDCEn	HZAyDCMn	HZAyDCNn	HZAyDCPn	HZAyDCTn	HZAyDCCn	0	HZAyDCFn

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] input.
1	Setting of the HZAyDCCn bit is invalid while the external pin[Note] input holds a level detected as abnormal (active level).
Rewrite the HZAyDCMn bit when the HZAyDCEn bit = 0.	

HZAyDCNn	HZAyDCPn	External pin[Note] input edge specification
0	0	No valid edge (setting the HZAyDCFn bit by external pin[Note] input is prohibited).
0	1	Rising edge of the external pin[Note] input is valid (abnormality is detected by rising edge input).
1	0	Falling edge of the external pin[Note] input is valid (abnormality is detected by falling edge input).
1	1	Setting prohibited

- Rewrite the HZAyDCNn and HZAyDCPn bits when the HZAyDCEn bit is 0.
- For the edge specification of the INTP0 to INTP3 pins, see **14.4.2 (1) External interrupt rising edge specification register 0 (INTR0), external interrupt falling edge specification register 0 (INTF0)**.
- 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] is at the active level when the operation is enabled, therefore, high-impedance output control is not performed.

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] (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.

Note HZA0CTL0: TOQH0OFF pin, HZA0CTL1: TOP2OFF pin, HZA1CTL0: TOQ1OFF pin, HZA1CTL1: TOP3OFF pin

(2/2)

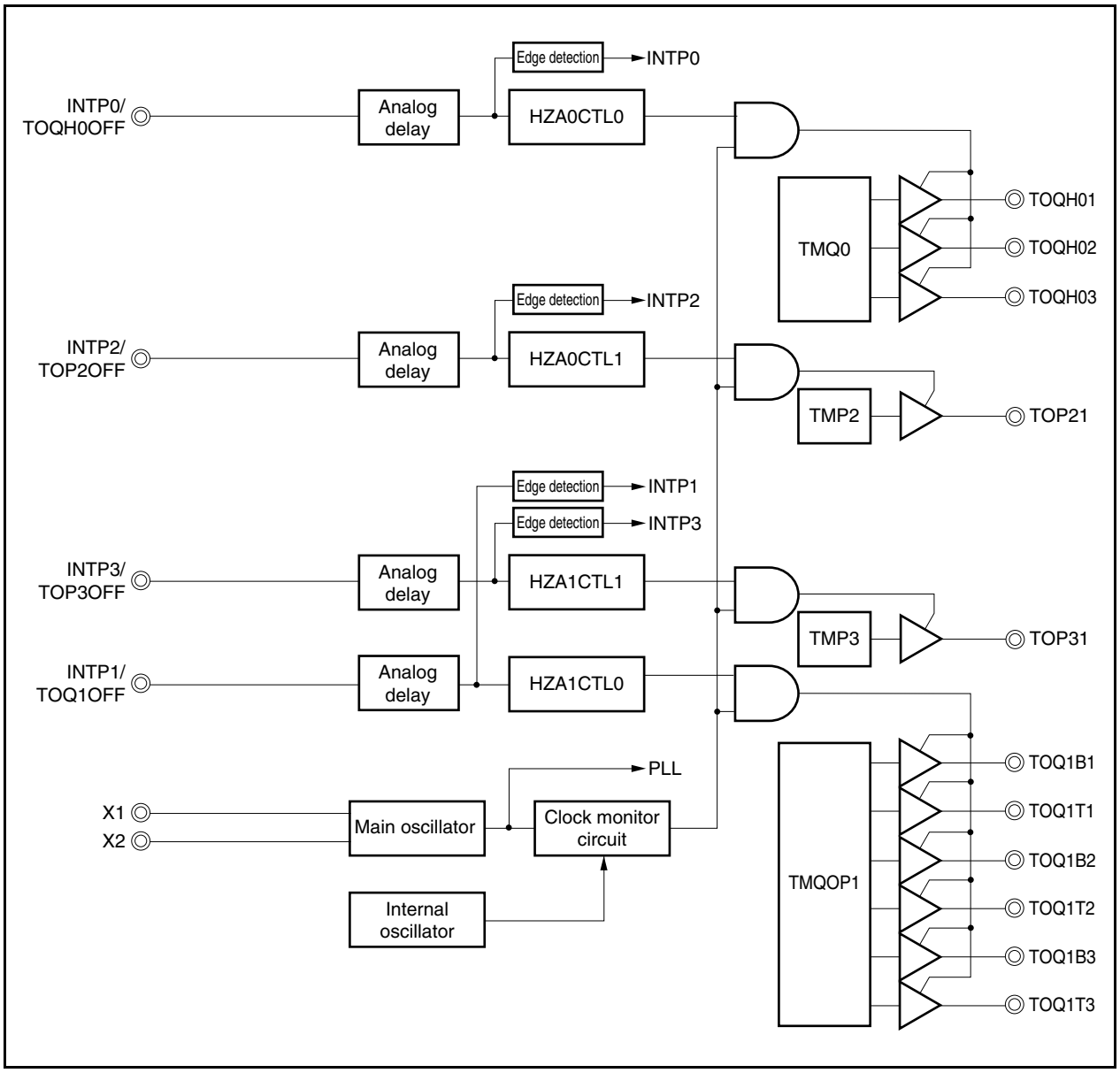
HZAyDCCn	High-impedance output control clear bit
0	No operation
1	Pins that have gone into a high-impedance state are output-enabled by software and the HZAyDCFn bit is cleared to 0.

- 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 HZA0CTL0: TOQH0OFF pin, HZA0CTL1: TOP2OFF pin, HZA1CTL0: TOQ1OFF pin, HZA1CTL1: TOP3OFF pin

Figure 9-4. High-Impedance Output Controller Configuration



(a) Setting procedure

(i) Setting of high-impedance control operation

<1> Set the HZAyDCMn, HZAyDCNn, and HZAyDCP1 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 HZAyDCP1 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 HZAyDCP1 bit and HZAyDCNn bit cleared to 0, the pin goes into a high-impedance state when the HZAyDCTn bit is set to 1.

Note HZA0CTL0: TOQH0OFF pin, HZA0CTL1: TOP2OFF pin, HZA1CTL0: TOQ1OFF pin, HZA1CTL1: TOP3OFF pin

9.4 Operation

9.4.1 System outline

(1) Outline of 6-phase PWM output

The 6-phase PWM output mode is used to generate a 6-phase PWM output wave, by using TMQ1 and the TMQ1 option in combination.

The 6-phase PWM output mode is enabled by setting the TQ1CTL1.TQ1MD2 to TQ1CTL1.TQ1MD0 bits of TMQ1 to "111".

One 16-bit counter and four 16-bit compare registers of TMQ1 are used to generate a basic 3-phase wave.

The functions of the compare registers are as follows.

TMP1 can perform a tuning operation with TMQ1 to start a conversion trigger source for A/D converters 0 and 1.

Compare Register	Function	Settable Range
TQ1CCR0 register	Setting of cycle	0002H ≤ m ≤ FFFEH
TQ1CCR1 register	Specifying output width of phase U	0000H ≤ i ≤ m + 1
TQ1CCR2 register	Specifying output width of phase V	0000H ≤ j ≤ m + 1
TQ1CCR3 register	Specifying output width of phase W	0000H ≤ k ≤ m + 1

Remark m = Set value of TQ1CCR0 register
i = Set value of TQ1CCR1 register
j = Set value of TQ1CCR2 register
k = Set value of TQ1CCR3 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 wave (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 TQ1CCR0 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 TQ1CCR0 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 TMQ1 dead-time compare register (TQ1DTC) 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 (TQ1DTC), the same dead-time value is used in the three phases.

<R>

Figure 9-5. Outline of 6-Phase PWM Output Mode

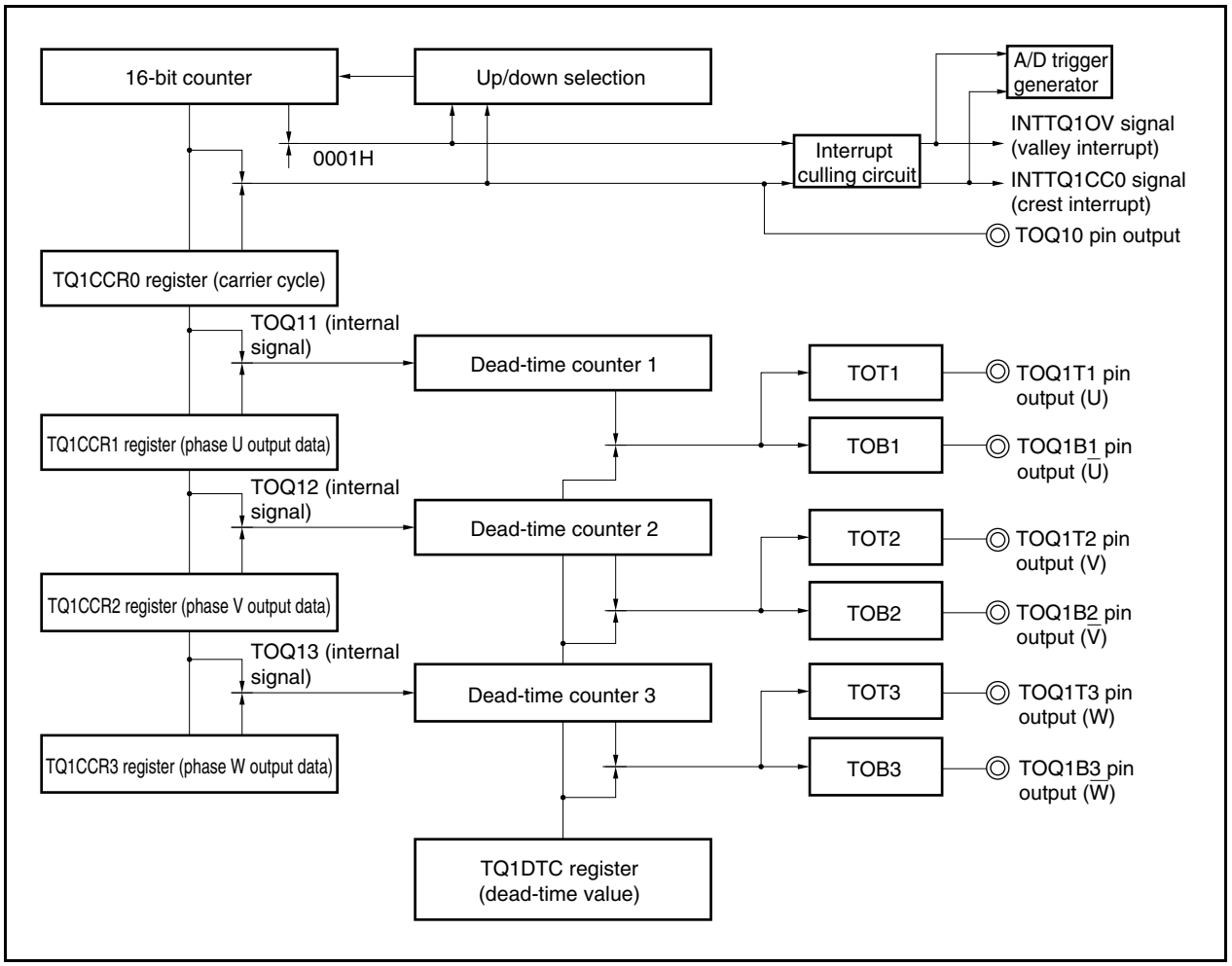
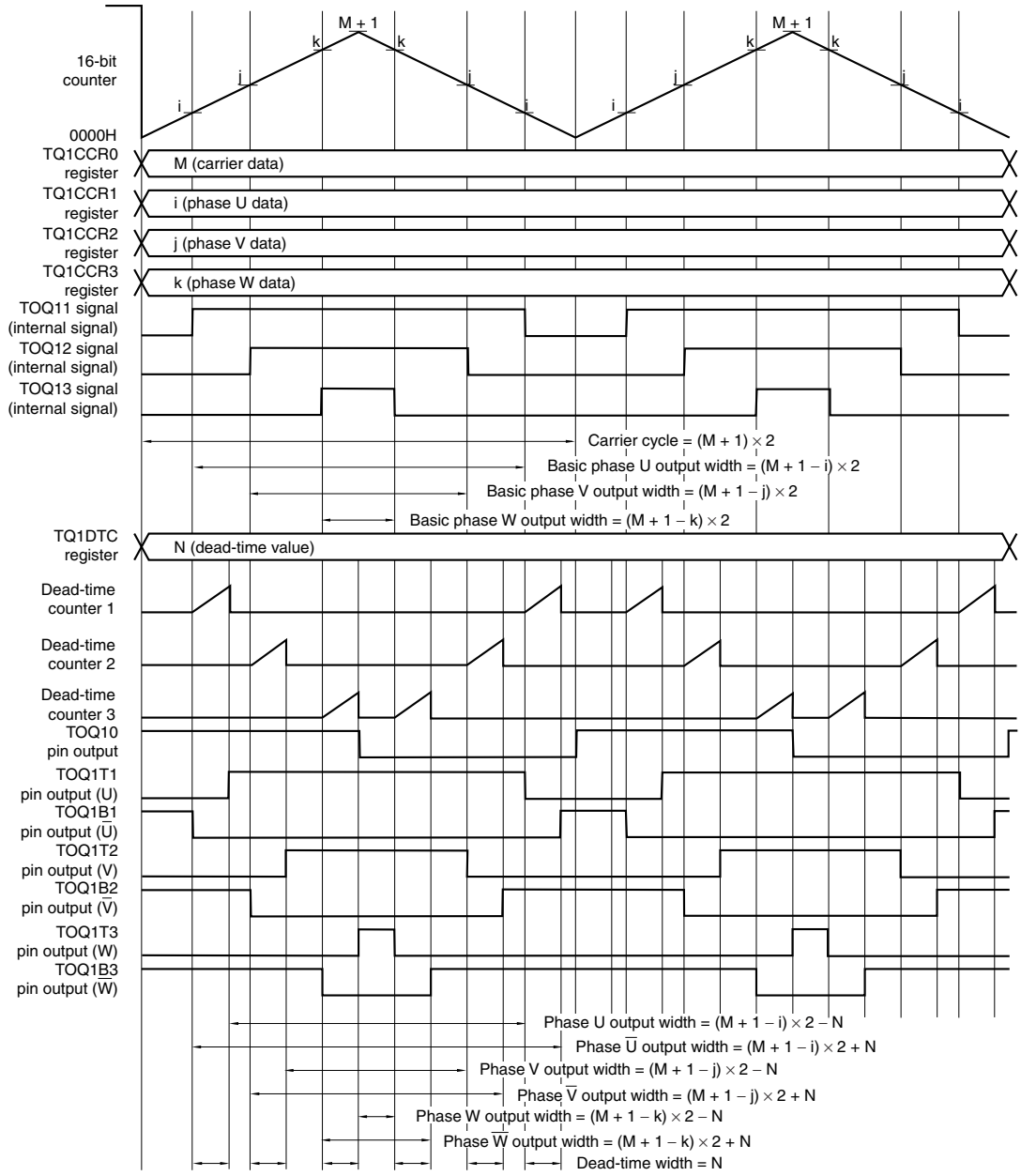


Figure 9-6. Timing Chart of 6-Phase PWM Output Mode



Cautions 1. Set the value "M" of the TQ1CCR0 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 TQ1CCR1, TQ1CCR2, and TQ1CCR3 registers.

3. The output is 100% if "0000H" is set to the TQ1CCR1, TQ1CCR2, and TQ1CCR3 registers. The output is 0% if "M + 1" is set to the TQ1CCR1, TQ1CCR2, and TQ1CCR3 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 TQ1CCR1, TQ1CCR2, and TQ1CCR3 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).

(2) Interrupt requests

Two types of interrupt requests are available: the INTTQ1CC0 (crest interrupt) signal and INTTQ1OV (valley interrupt) signal.

The INTTQ1CC0 and INTTQ1OV signals can be culled by using the TQ1OPT1 register.

For details of culling interrupts, see **9.4.3 Interrupt culling function**.

- INTTQ1CC0 (crest interrupt) signal: Interrupt signal indicating match between the value of the 16-bit counter that counts up and the value of the TQ1CCR0 register
- INTTQ1OV (valley interrupt) signal: Interrupt signal indicating match 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 rewrite mode, batch rewrite mode, or intermittent batch rewrite mode.

Related Unit	Register
Timer P1	TMP1 capture/compare register 0 (TP1CCR0) TMP1 capture/compare register 1 (TP1CCR1)
Timer Q1	TMQ1 capture/compare register 0 (TQ1CCR0) TMQ1 capture/compare register 1 (TQ1CCR1) TMQ1 capture/compare register 2 (TQ1CCR2) TMQ1 capture/compare register 3 (TQ1CCR3)
Timer Q1 option	TMQ1 option register 1 (TQ1OPT1)

For details of the transfer function of the compare register, see **9.4.4 Operation to rewrite register with transfer function**.

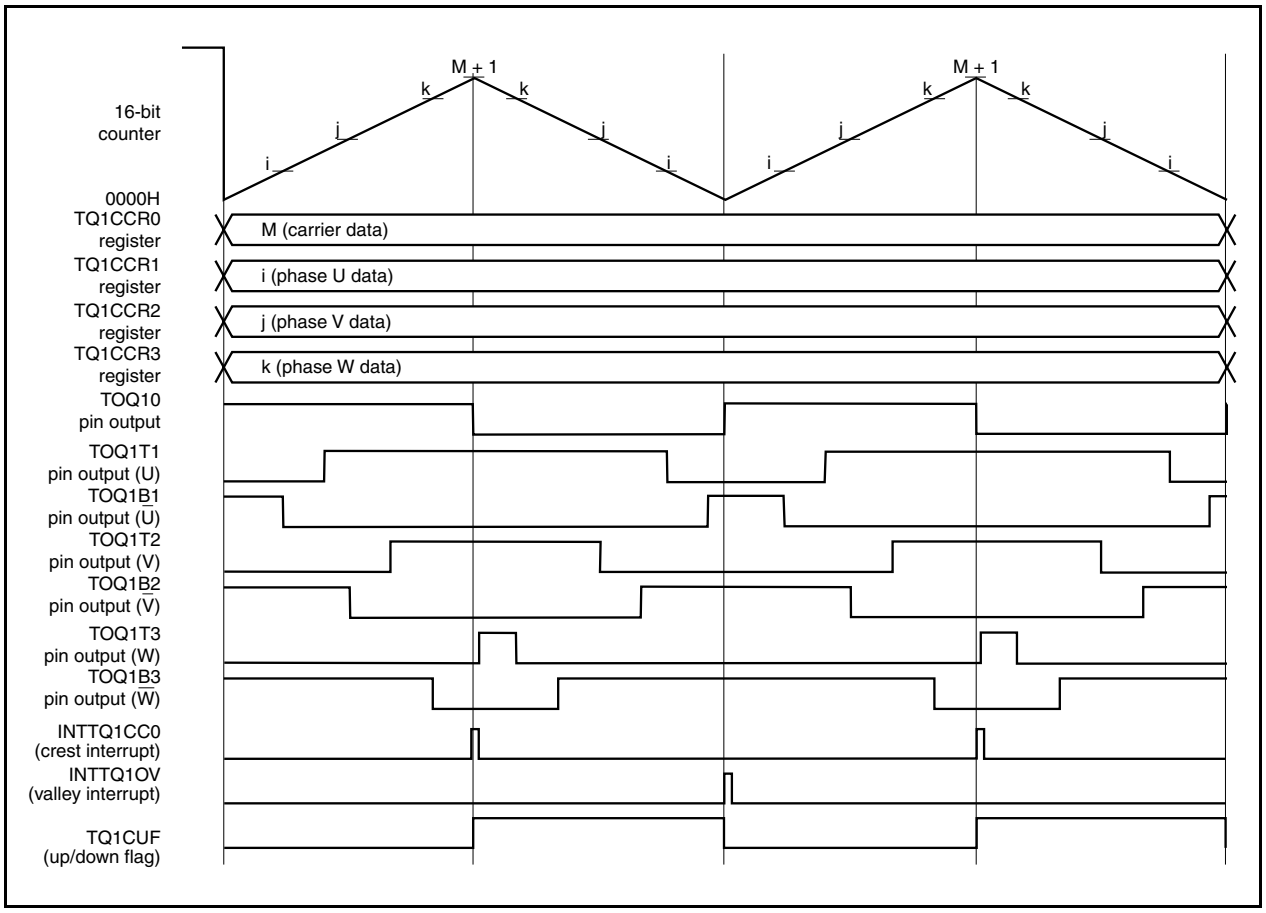
(4) Counting-up/down operation of 16-bit counter

The operation status of the 16-bit counter can be checked by using the TQ1CUF bit of TMQ1 option register 0 (TQ1OPT0).

Status of TQ1CUF Bit	Status of 16-Bit Counter	Range of 16-Bit Counter Value
TQ1CUF bit = 0	Counting up	0000H – m
TQ1CUF bit = 1	Counting down	(m + 1) – 0001H

Remark m = Set value of TQ1CCR0 register

Figure 9-7. Interrupt and Up/Down Flag



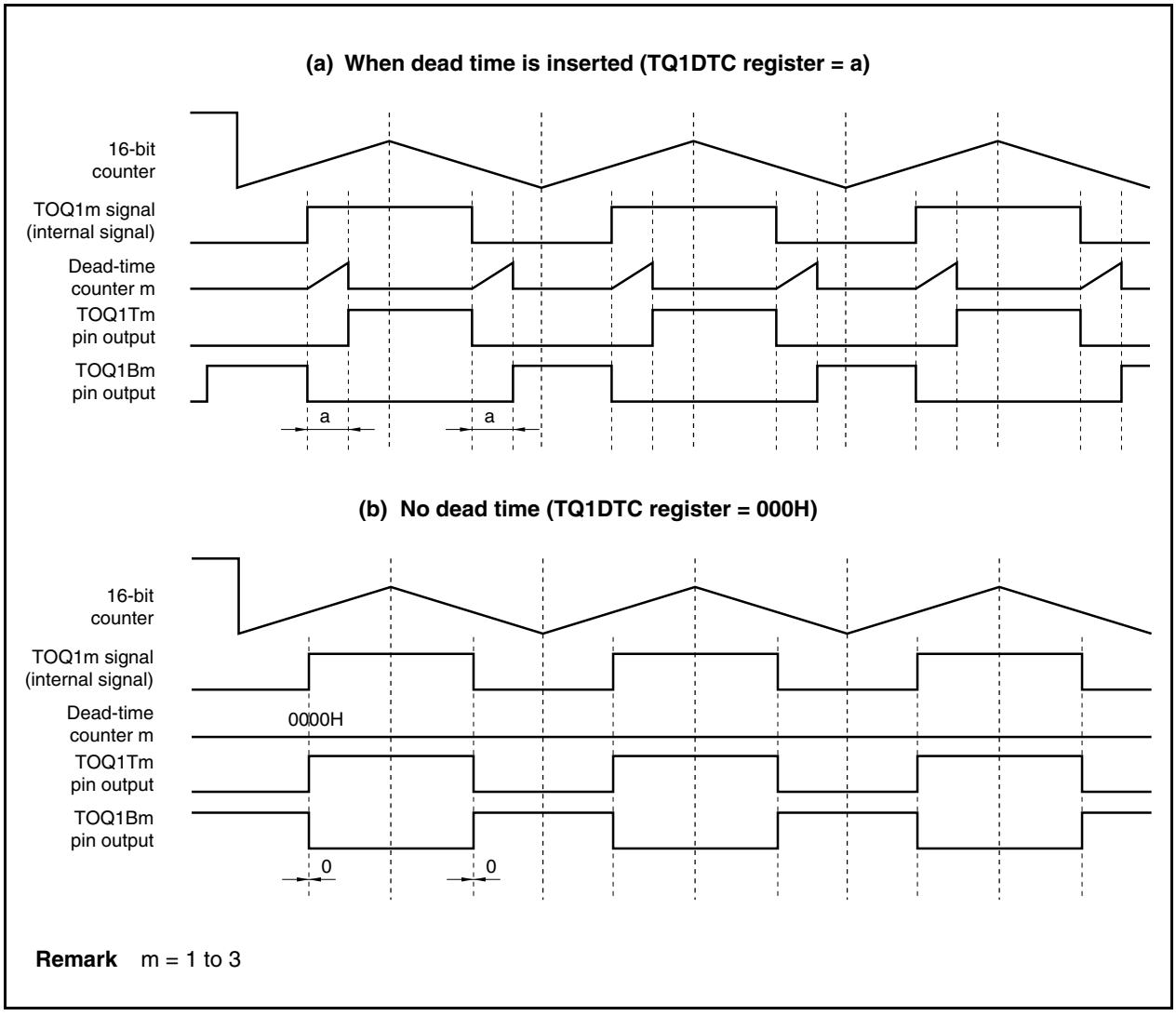
9.4.2 Dead-time control (generation of negative-phase wave signal)

(1) Dead-time control mechanism

In the 6-phase PWM output mode, compare registers 1 to 3 (TQ1CCR1, TQ1CCR2, and TQ1CCR3) are used to set the duty factor, and compare register 0 (TQ1CCR0) is used to set the cycle. By setting these four registers and by starting the operation of TMQ, 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 (TMQOP1) and their inverted signal with dead-time is created to generate three sets of (six) PWM waves.

The TMQOP1 unit consists of three 10-bit counters (dead-time counters 1 to 3) that operate in synchronization with the count clock of TMQ1, and a TMQ1 dead-time compare register (TQ1DTC) that specifies dead time. If “a” is set to the TQ1DTC register, the dead-time value is “a”, and interval “a” is created between a positive-phase wave and a negative-phase wave.

Figure 9-8. PWM Output Wave with Dead Time (1)



Remark m = 1 to 3

(2) PWM output of 0%/100%

The V850ES/IE2 is capable of 0% wave output and 100% wave output for PWM output.

A low level is continuously output from TOQ1Tm pin as the 0% wave output. A high level is continuously output from TOQ1Tm pin as the 100% wave output.

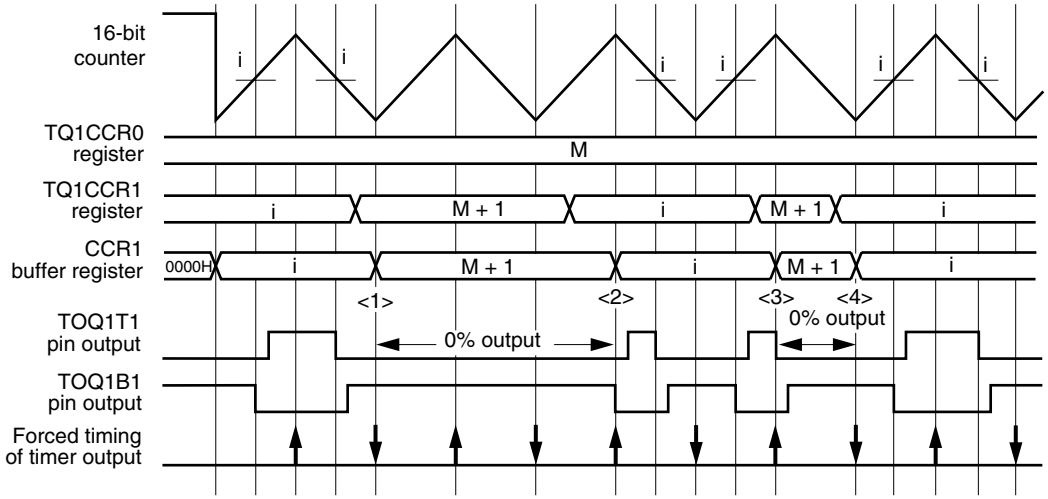
The 0% wave is output by setting the TQ1CCRm register to “M + 1” when the TQ1CCR0 register = M.

The 100% wave is output by setting the TQ1CCRm register to “0000H”.

Rewriting the TQ1CCRm register is enabled while the timer is operating, and 0% wave output or 100% wave output can be selected at the point of the crest interrupt (INTTQ1CC0) and valley interrupt (INTTQ1OV).

Remark m = 1 to 3

<R> **Figure 9-9. 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.

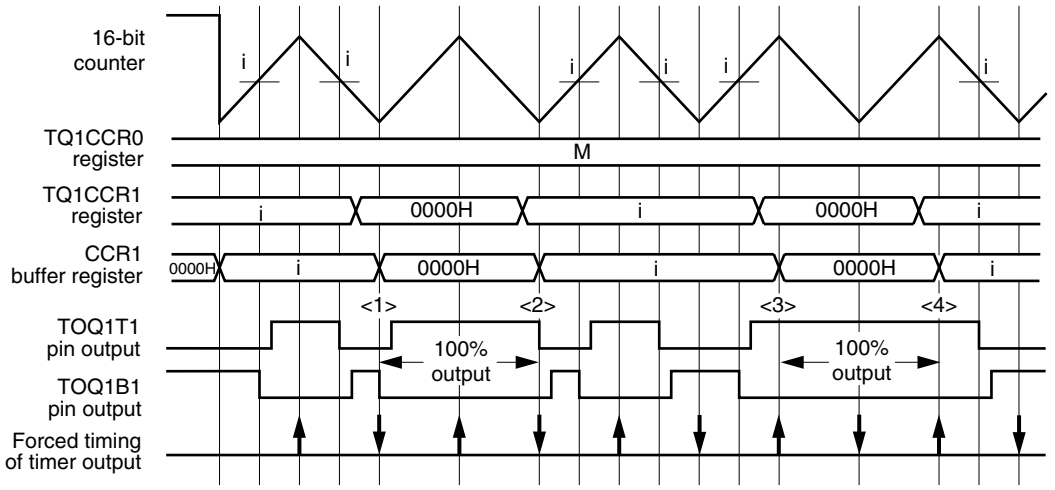
<2> 0% output is canceled by the crest interrupt (without a match with the 16-bit counter).
The crest interrupt forcibly raises the timer output. This cancels the 0% output.

<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 TQ1CCRm 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.

<R> **Figure 9-10. 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 TQ1CCRm register matches the value of the 16-bit counter. As a result, the 100% output is produced.

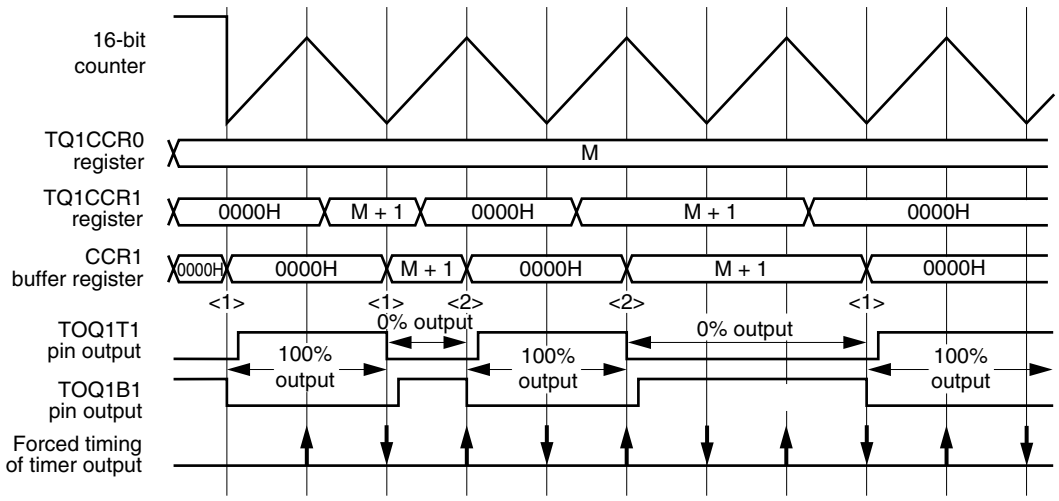
<2> 100% output is canceled by the valley interrupt (without a match with the 16-bit counter).
The valley interrupt forcibly lowers the timer output. This cancels the 100% output.

<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.

<R> **Figure 9-11. 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 TQ1CCR0 register value.

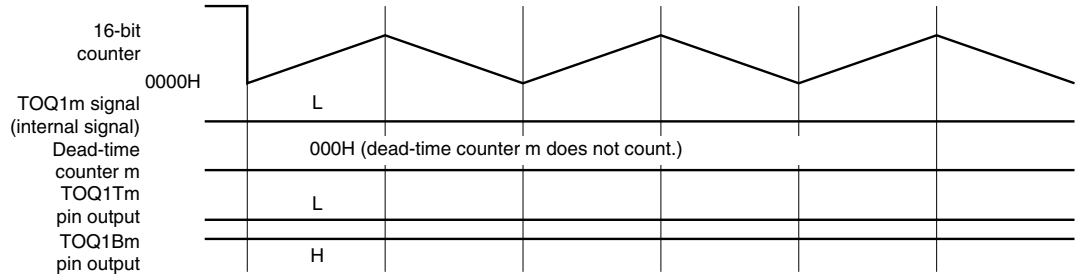
(3) Output waveform in vicinity of 0% and 100% output

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.

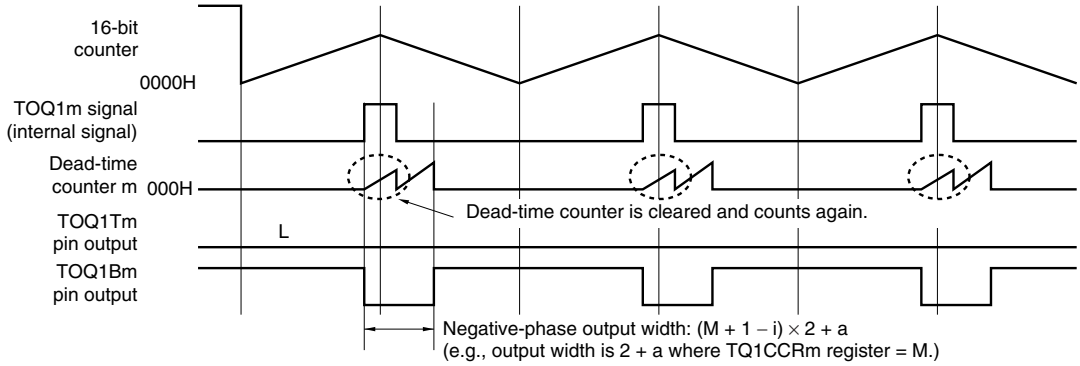
The output waveform of dead-time control in the vicinity of 0% and 100% output is shown below.

Figure 9-12. PWM Output Waveform with Dead Time (2)

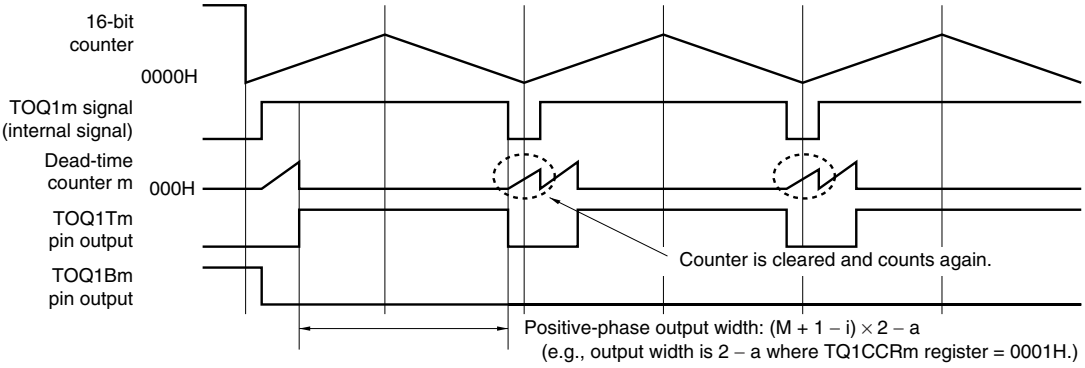
(a) 0% output (TQ1CCRm register = M + 1, TQ1CCR0 register = M, TQ1DTC register = a)



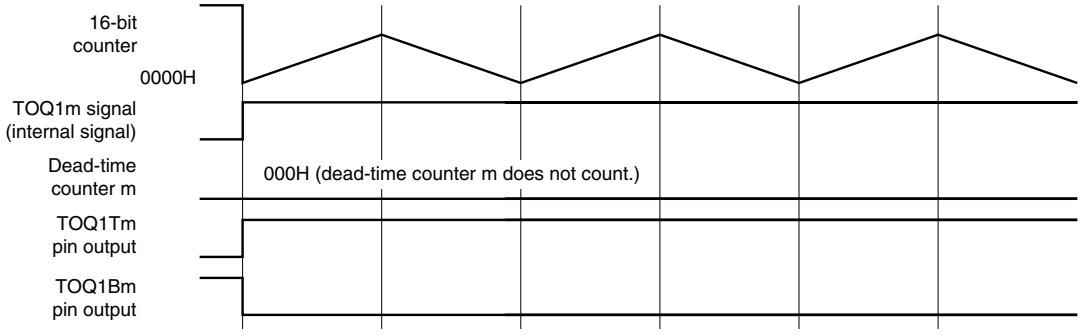
(b) In vicinity of 0% output (TQ1CCRm register = i ≥ M + 1 − a/2, TQ1CCR0 register = M, TQ1DTC register = a)



(c) In vicinity of 100% output (TQ1CCRm register = i ≤ a/2, TQ1CCR0 register = M, TQ1DTC register = a)



(d) 100% output (TQ1CCRm register = 0000H, TQ1CCR0 register = M, TQ1DTC register = a)



Remark m = 1 to 3

(4) Automatic dead-time width narrowing function (TQ1OPT2.TQ1DTM bit = 1)

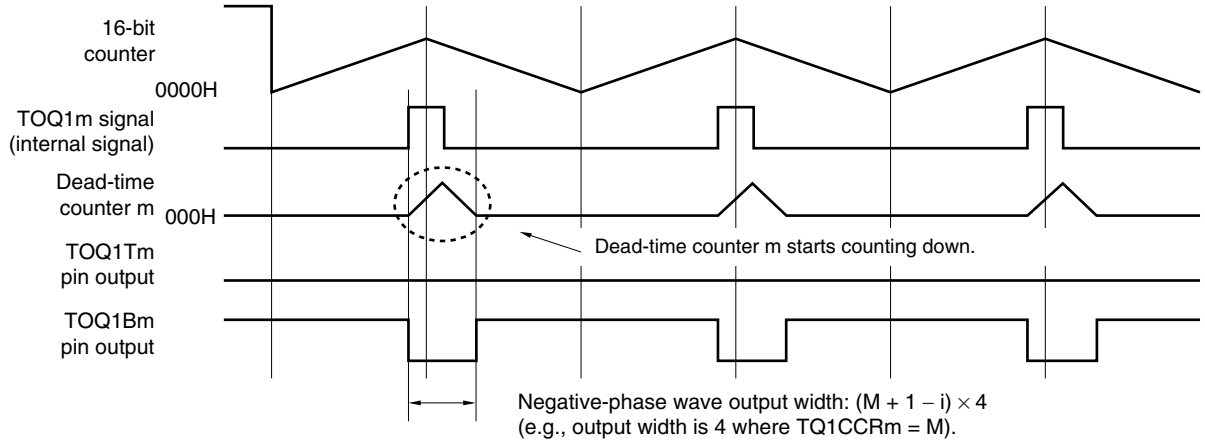
The dead-time width can be automatically narrowed in the vicinity of 0% output or 100% output by setting the TQ1OPT2.TQ1DTM bit to 1.

By setting the TQ1DTM bit to 1, the dead-time counter is not cleared, but starts counting down if the TOQ1m (internal signal) output of timer Q changes during dead-time counting.

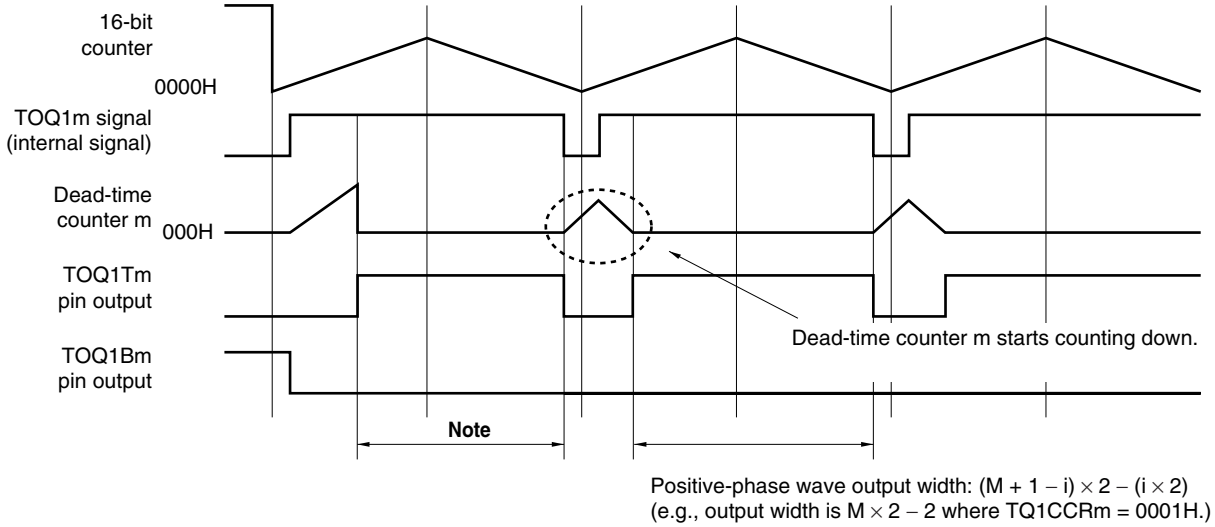
The following timing chart shows the operation of the dead-time counter when the TQ1DTM bit is set to 1.

Figure 9-13. Operation of Dead-Time Counter m (1)

(a) In vicinity of 0% output (TQ1CCRm register = i ≥ M + 1 − a/2, TQ1CCR0 register = M, TQ1DTC register = a)



(b) In vicinity of 100% output (TQ1CCRm register = i ≤ a/2, TQ1CCR0 register = M, TQ1DTC register = a)



Note The output width of the first wave differs from that of the second and subsequent waves immediately after the TQ1CTL0.TQ1CE bit has been set. The first wave is shorter than the second wave because the dead time is fully counted.

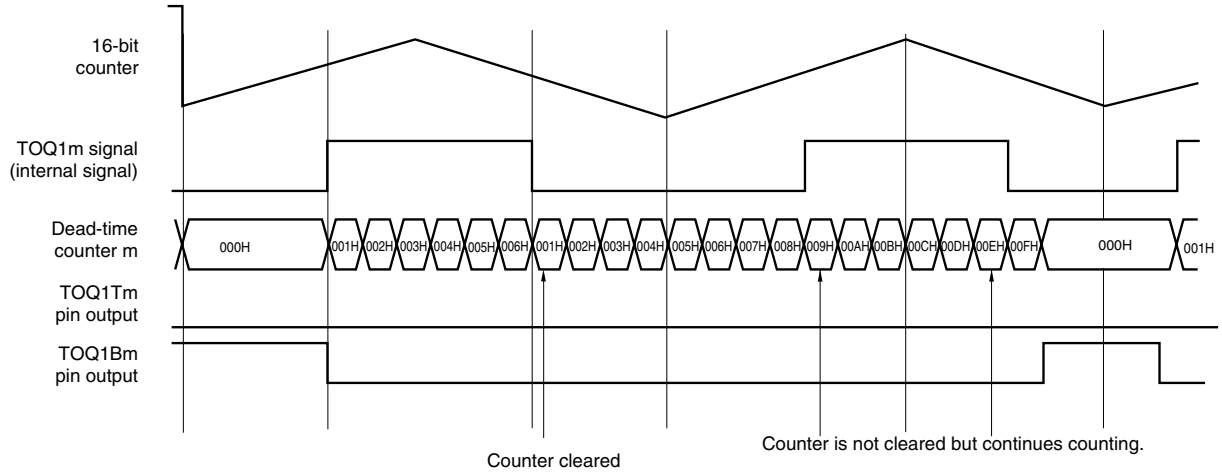
Remark m = 1 to 3

(5) Dead-time control in case of incorrect setting

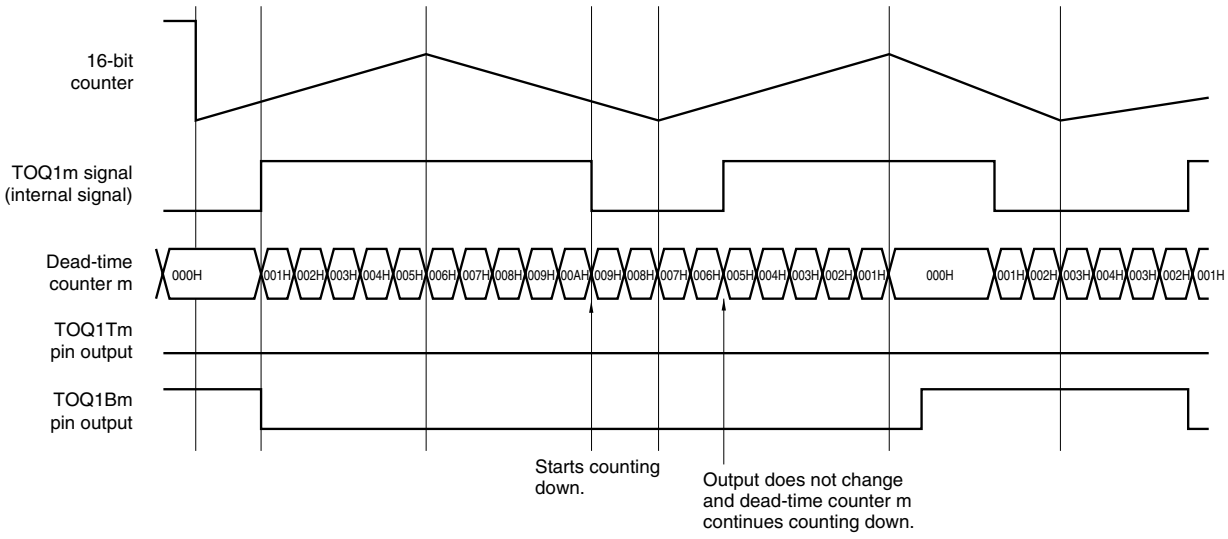
Usually, the TOQ1m (internal signal) output of TMQ1 changes only once during dead-time counting, only in the vicinity of 0% and 100% output. This section shows an example where the TQ1CCR0 register (carrier cycle) and TQ1DTC register (dead-time value) are incorrectly set. If these registers are incorrectly set, the TOQ1m (internal signal) output of TMQ1 changes more than once during dead-time counting. The following flowchart shows the 6-phase PWM output wave in this case.

Figure 9-14. Operation of Dead-Time Counter m (2)

(a) When TQ1OPT2.TQ1DTM bit = 0, TQ1CCR0 register = 0006H, TQ1DTC register = 000FH, TQ1CCRm register = 0004H



(b) When TQ1OPT2.TQ1DTM bit = 1, TQ1CCR0 register = 0006H, TQ1DTC register = 000FH, TQ1CCRm register = 0002H



Remark m = 1 to 3

9.4.3 Interrupt culling function

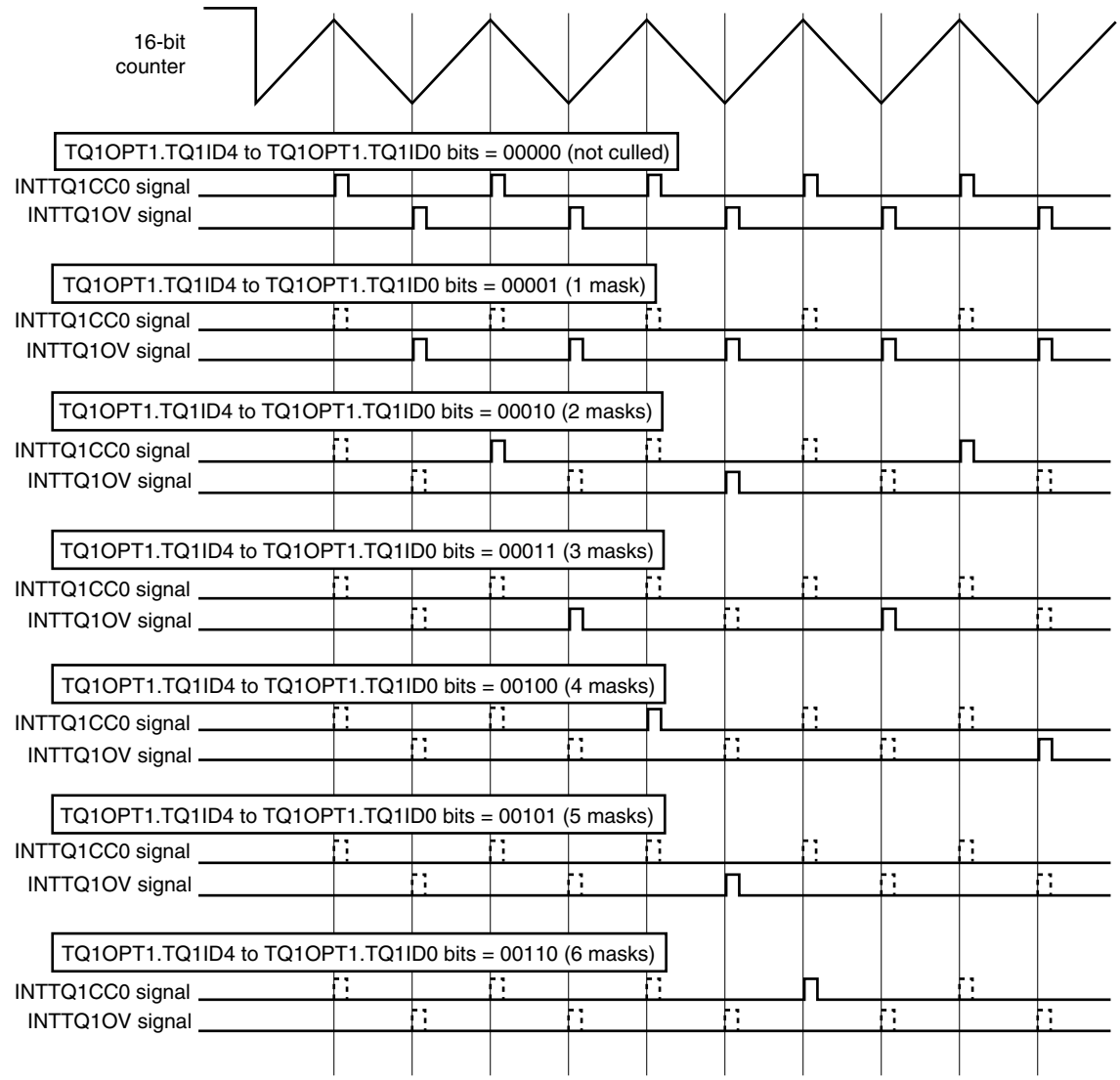
- The interrupts to be culled are INTTQ1CC0 (crest interrupt) and INTTQ1OV (valley interrupt).
- The TQ1OPT1.TQ1ICE bit is used to enable output of the INTTQ1CC0 interrupt and the number of times the interrupt is to be culled.
- The TQ1OPT1.TQ1IOE bit is used to enable output of the INTTQ1OV interrupt and the number of times the interrupt is to be culled.
- <R> The TQ1OPT1.TQ1ID4 to TQ1OPT1.TQ1ID0 bits are used to specify the number of times for which an interrupt, subject to counting of culling, is counted.
  The interrupt is masked for the specified culling count and the masked interrupt occurs at the next occurrence timing.
- The TQ1RDE bit of TQ1OPT2 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 TQ1CCR1 register has been written.
- The TQ1OPT0.TQ1CMS 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 TQ1CMS bit is 0. When the TQ1CMS 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 TQ1CCRm 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.

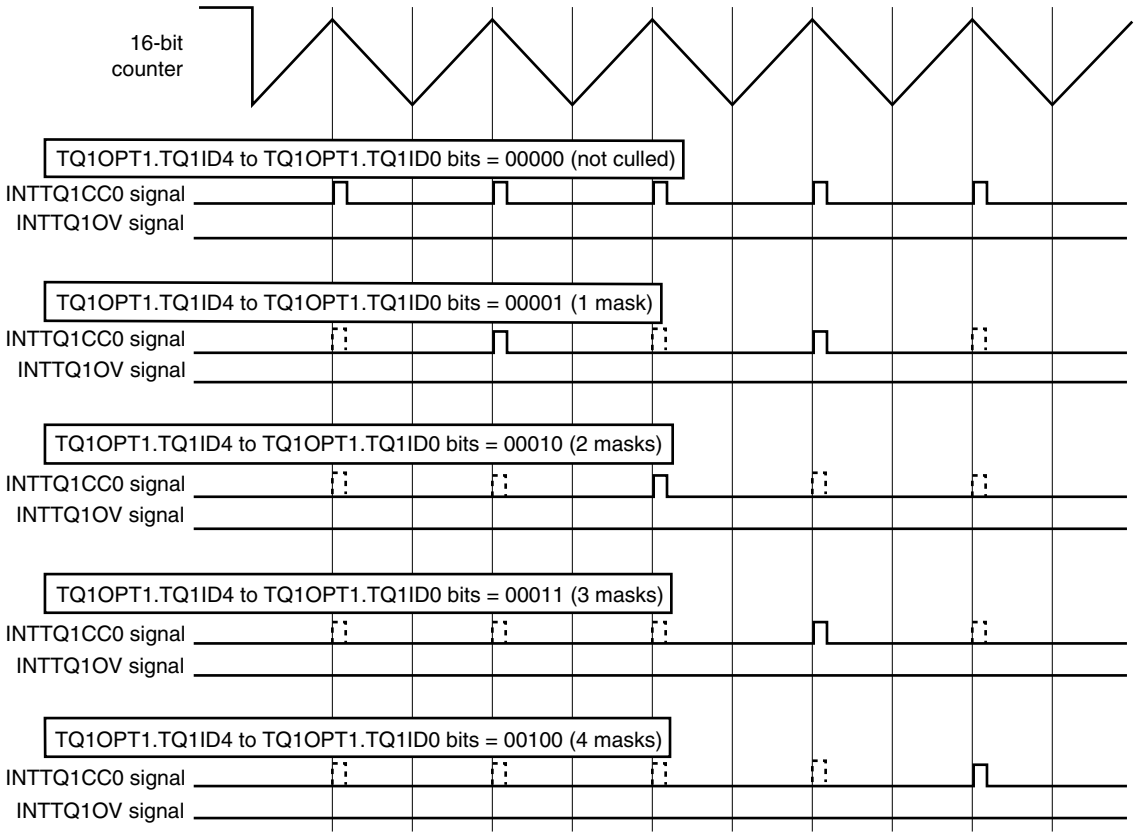
(1) Interrupt culling operation

Figure 9-15. Interrupt Culling Operation When TQ1OPT1.TQ1ICE Bit = 1, TQ1OPT1.TQ1IOE Bit = 1, TQ1OPT2.TQ1RDE Bit = 1 (Crest/Valley Interrupt Output)



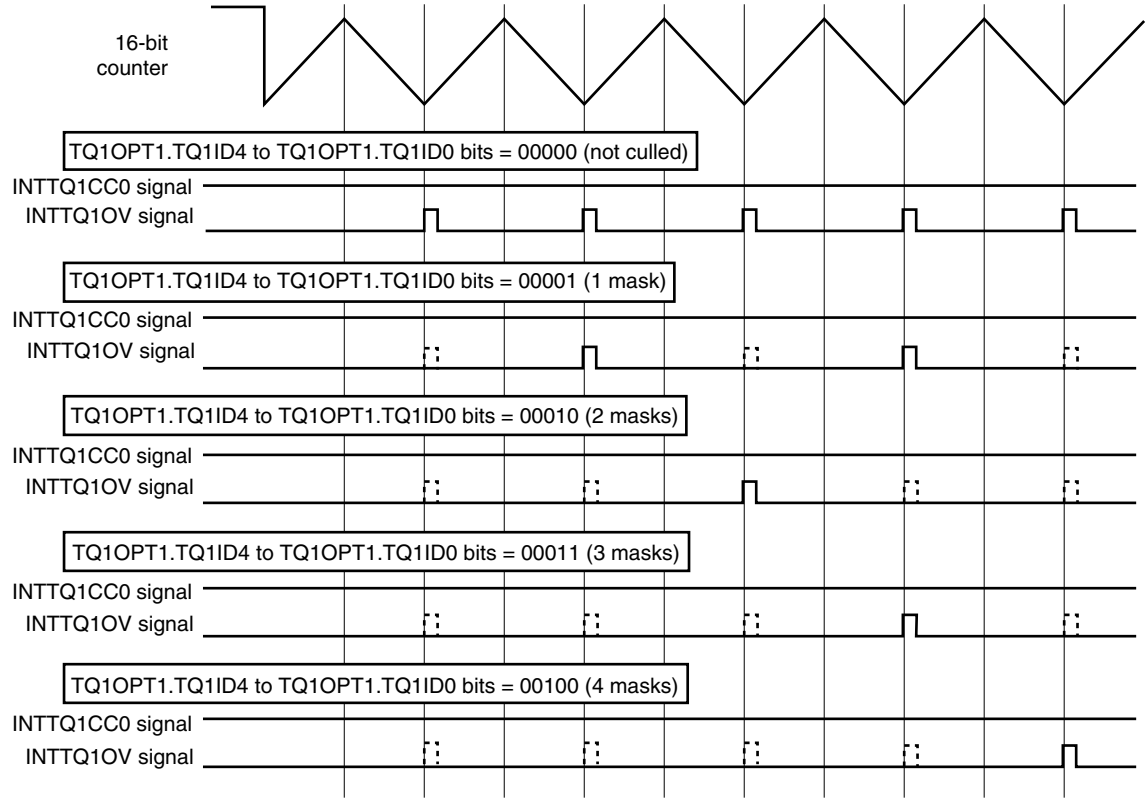
Remark : Culled interrupt

Figure 9-16. Interrupt Culling Operation When TQ1OPT1.TQ1ICE Bit = 1, TQ1OPT1.TQ1IOE Bit = 0, TQ1OPT2.TQ1RDE Bit = 1 (Crest Interrupt Output)



Remark : Culled interrupt

Figure 9-17. Interrupt Culling Operation When TQ1OPT1.TQ1ICE Bit = 0, TQ1OPT1.TQ1IOE Bit = 1, TQ1OPT2.TQ1RDE Bit = 1 (Valley Interrupt Output)



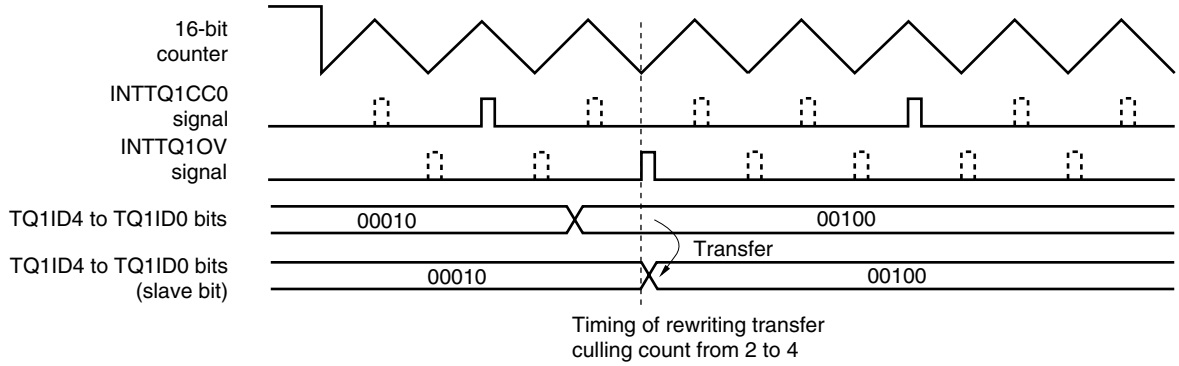
Remark : Culled interrupt

(2) To alternately output crest interrupt (INTTQ1CC0) and valley interrupt (INTTQ1OV)

To alternately output the crest and valley interrupts, set both the TQ1OPT1.TQ1ICE and TQ1OPT1.TQ1IOE bits to 1.

Figure 9-18. Crest/Valley Interrupt Output

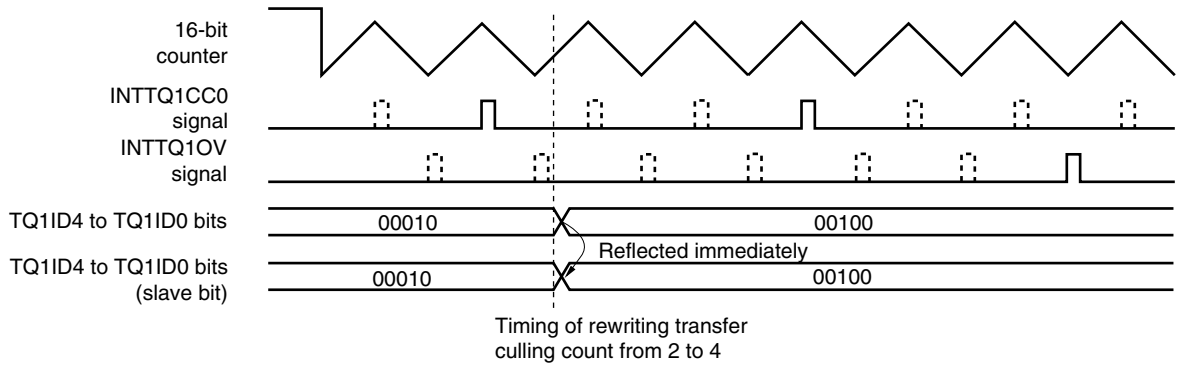
(a) TQ1OPT0.TQ1CMS bit = 0, TQ1OPT2.TQ1RDE bit = 1 (with transfer culling control)



Remarks 1. Transfer is performed at the culled interrupt output timing. The other transfer timing is ignored.

2. : Culled interrupt

(b) TQ1CMS bit = 1, TQ1RDE bit = 0 or 1 (without transfer control)



Remarks 1. Rewriting is reflected immediately. The transfer timing is ignored.

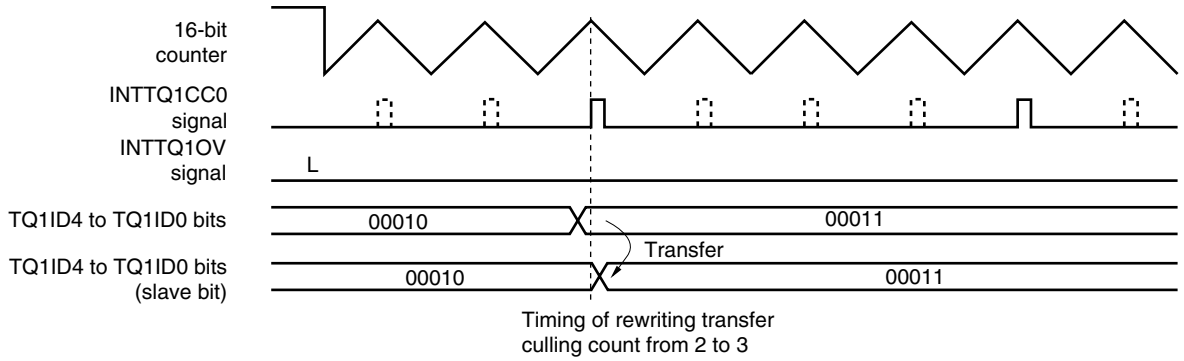
2. : Culled interrupt

(3) To output only crest interrupt (INTTQ1CC0)

Set the TQ1OPT1.TQ1ICE bit to 1 and clear the TQ1OPT1.TQ1IOE bit to 0.

Figure 9-19. Crest Interrupt Output

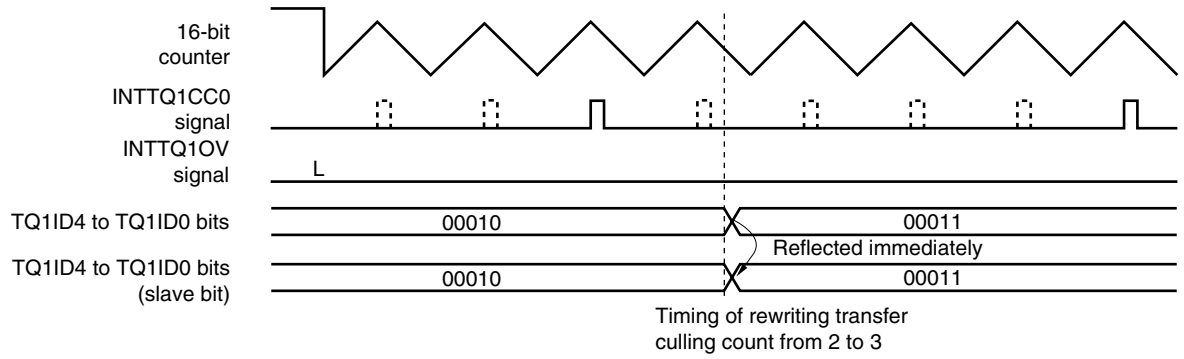
(a) TQ1OPT0.TQ1CMS bit = 0, TQ1OPT2.TQ1RDE bit = 1 (with transfer culling control)



Remarks 1. Transfer is performed at the culled interrupt output timing. The other transfer timing is ignored.

2. : Culled interrupt

(b) TQ1CMS bit = 1, TQ1RDE bit = 0 or 1 (without transfer control)



Remarks 1. Rewriting is reflected immediately. The transfer timing is ignored.

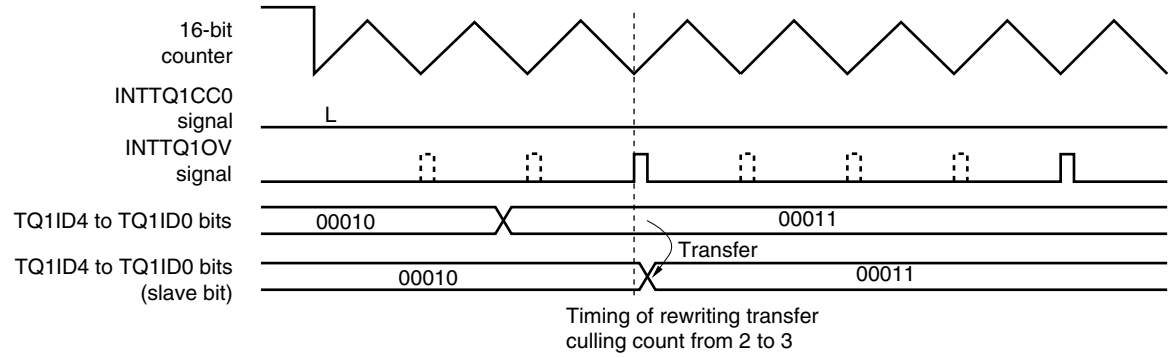
2. : Culled interrupt

(4) To output only valley interrupt (INTTQ1OV)

Clear the TQ1OPT1.TQ1ICE bit to 0 and set the TQ1IOE bit to 1.

Figure 9-20. Valley Interrupt Output

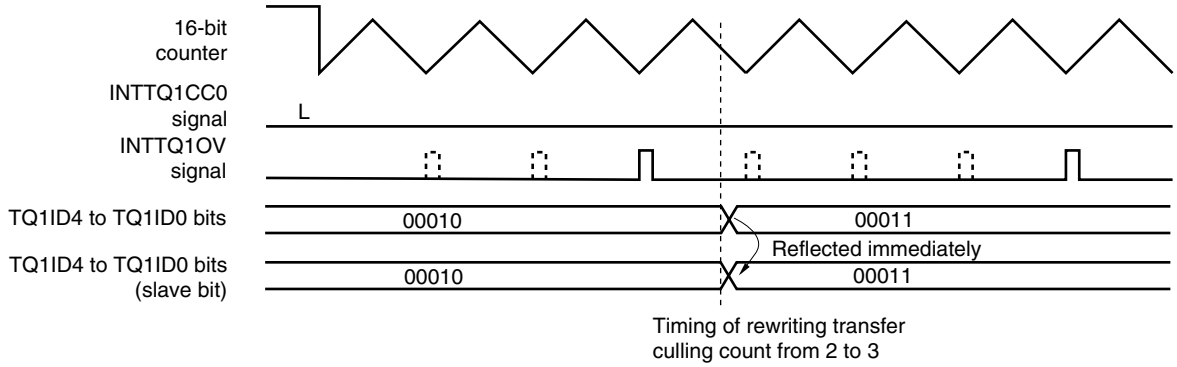
(a) TQ1OPT0.TQ1CMS bit = 0, TQ1OPT2.TQ1RDE bit = 1 (with transfer culling control)



Remarks 1. Transfer is performed at the culled interrupt output timing. The other transfer timing is ignored.

2. : Culled interrupt

(b) TQ1CMS bit = 1, TQ1RDE bit = 0 or 1 (without transfer control)



Remarks 1. Rewriting is reflected immediately. The transfer timing is ignored.

2. : Culled interrupt

9.4.4 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.

- TQ1CCR0: Register that specifies the cycle of the 16-bit counter (TMQ)
- TQ1CCR1: Register that specifies the duty factor of TOQ1T1 (U) and TOQ1B1 ($\overline{U}$)
- TQ1CCR2: Register that specifies the duty factor of TOQ1T2 (V) and TOQ1B2 ($\overline{V}$)
- TQ1CCR3: Register that specifies the duty factor of TOQ1T3 (W) and TOQ1B3 ($\overline{W}$)
- TQ1OPT1: Register that specifies the culling of interrupts
- TP1CCR0: Register that specifies the A/D conversion start trigger generation timing (TMP1 during tuning operation)
- TP1CCR1: Register that specifies the A/D conversion start trigger generation timing (TMP1 during tuning operation)

The following three rewrite modes are provided in the registers with a transfer function.

- Anytime rewrite mode
  This mode is set by setting the TQ1OPT0.TQ1CMS bit to 1. The setting of the TQ1OPT2.TQ1RDE 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 set by clearing the TQ1CMS bit of TQ1OPT0 to 0, the TQ1OPT1.TQ1ID4 to TQ1OPT1.TQ1ID0 bits to 00000, and the TQ1OPT2.TQ1RDE bit to 0. When data is written to the TQ1CCR1 register, the seven registers are transferred to the buffer register all at once at the next transfer timing. Unless the TQ1CCR1 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 TQ1CCR0 register value) and valley (match between the 16-bit counter value and 0001H) regardless of the interrupt.

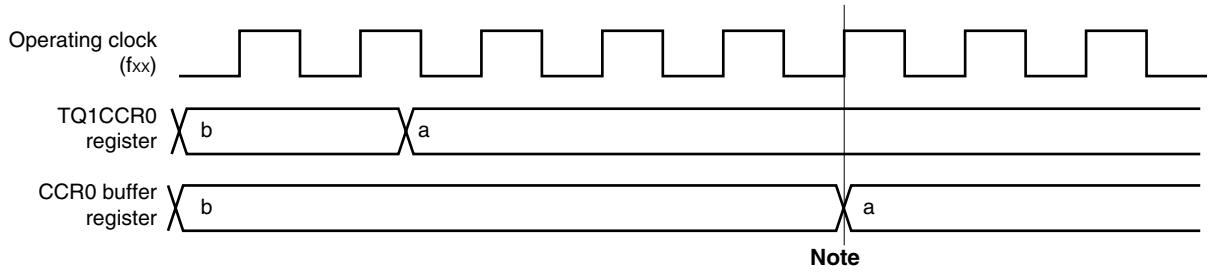
- Intermittent batch rewrite mode (transfer culling mode)
  This mode is set by clearing the TQ1OPT0.TQ1CMS bit to 0 and setting the TQ1OPT2.TQ1RDE bit to 1.
  When data is written to the TQ1CCR1 register, the seven registers are transferred to the buffer register all at once at the next transfer timing. Unless the TQ1CCR1 register is rewritten, the transfer operation is not performed even if the other six registers are rewritten.
  If interrupt culling is specified by the TQ1OPT1 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 TQ1CCR0 register value) or valley interrupt (match between the 16-bit counter value and 0001H).
  For details of the interrupt culling function, see **9.4.3 Interrupt culling function**.

(1) Anytime rewrite mode

This mode is set by setting the TQ1OPT0.TQ1CMS bit to 1. The setting of the TQ1OPT2.TQ1RDE 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 TQ1CCRm 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 counting up, the new register value becomes valid after the counter has started counting down.

Figure 9-21. Timing of Reflecting Rewritten Value

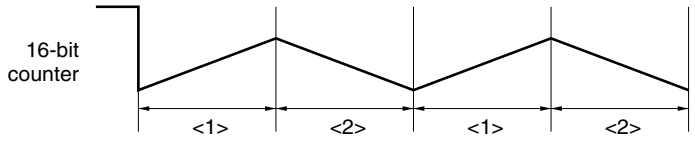


Note After the register (TQ1CCR0, TQ1CCR2, TQ1CCR3, TQ1OPT1, TP1CCR0, or TP1CCR1) has been written, the written value is transferred to an internal buffer register after four clocks of the operating clock. However, the value of only the TQ1CCR1 register is transferred after 5 more clocks.

(a) Rewriting TQ1CCR0 register

Even if the TQ1CCR0 register is rewritten in the anytime rewrite mode, the new value may not be reflected in some cases.

Figure 9-22. Example of Rewriting TQ1CCR0 Register



Rewriting during period <1> (rewriting during counting up)

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 counting down)

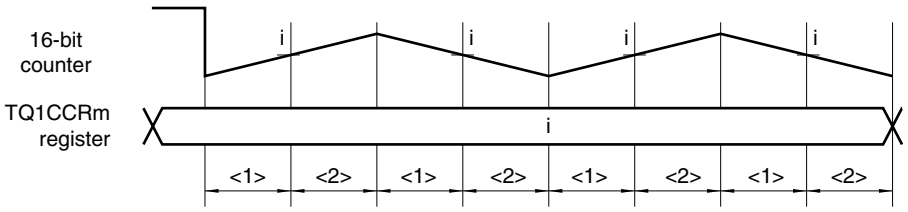
A match with the value of the 16-bit counter is ignored during counting down. Therefore, the rewritten period value is reflected starting from counting up in the next cycle as a match point.

(b) Rewriting TQ1CCRm register

Figure 9-24 shows the timing of rewriting before the value of the 16-bit counter matches the value of the TQ1CCRm register (<1> in Figure 9-23), and Figure 9-25 shows the timing of rewriting after the value of the 16-bit counter matches the value of the TQ1CCRm register (<2> in Figure 9-23).

Figure 9-23. Basic Operation of 16-Bit Counter and TQ1CCRm Register

(a) Basic figure



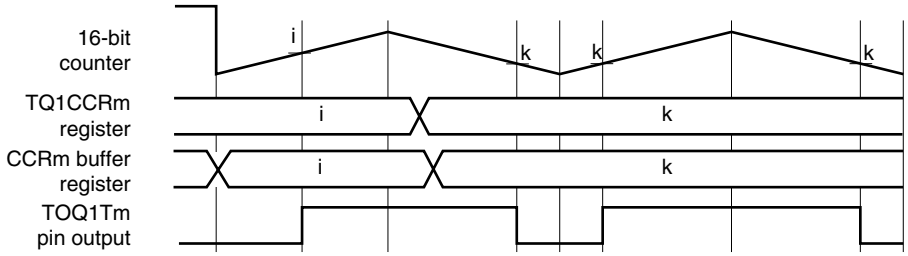
Remarks 1. i = Set value of TQ1CCRm register

2. m = 1 to 3

Figure 9-24. Example of Rewriting TQ1CCR1 to TQ1CCR3 Registers (Rewriting Before Match Occurs)

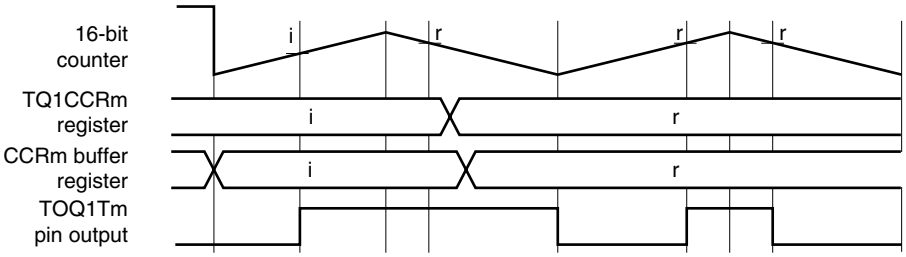
(a)

If the TQ1CCRm 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 TQ1CCRm register, the output waveform is as follows because the register value does not match the counter value.

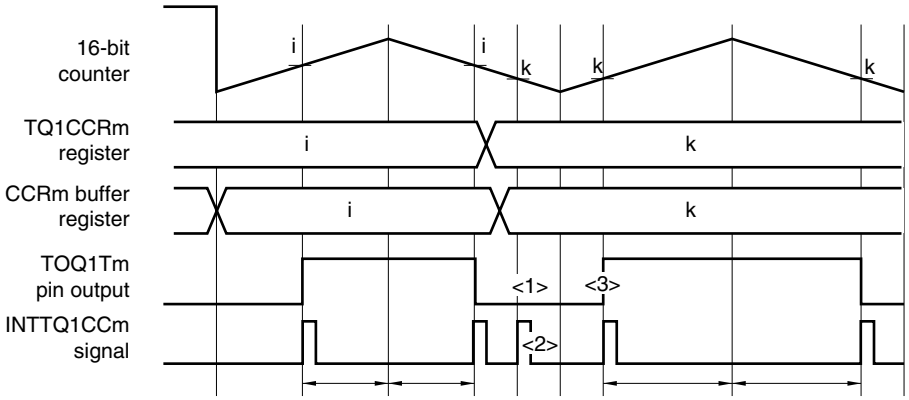


If the register value does not match the counter value, the TOQ1Tm pin output does not change. Even if the value of the 16-bit counter does not match the value of the TQ1CCRm register, the TOQ1Tm pin output always changes to the high level if the crest interrupt occurs and to the low level if the valley interrupt occurs.
This is a function provided for 0% output and 100% output.
For details, see **9.4.2 (2) PWM output of 0%/100%**.

Remarks 1. i, r, k = Set values of TQ1CCRm register

2. m = 1 to 3

Figure 9-25. Example of Rewriting TQ1CCR1 to TQ1CCR3 Registers (Rewriting After Match Occurs)



<1> Matching of the count value of the 16-bit counter and the value of the TQ1CCRm 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 TQ1CCRm register value (INTTQ1CCm) is output.

<3> The next match between the 16-bit counter and TQ1CCRm register is valid after the counter has changed its counting direction to up or down, and the PWM output changes.

If the TQ1CCRm 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 TOQ1Tm 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 TQ1CCRm register

2. m = 1 to 3

(c) Rewriting TQ1OPT1 register

The interrupt culling counter is cleared when the TQ1OPT1 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 TQ1OPT1 register in the intermittent batch rewrite mode (transfer culling mode).

For details of rewriting the TQ1OPT1 register, see **9.4.3 Interrupt culling function**.

(2) Batch rewrite mode (transfer mode)

This mode is set by clearing the TQ1OPT0.TQ1CMS bit to 0, the TQ1OPT1.TQ1ID4 to TQ1OPT1.TQ1ID0 bits to 00000, and the TQ1OPT2.TQ1RDE 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 TQ1CCR1 register, the values set to the TQ1CCR0 to TQ1CCR3, TQ1OPT1, TP1CCR0, and TP1CCR1 registers are transferred all at once to the internal buffer register at the next transfer timing. Therefore, write to the TQ1CCR1 register last. Writing to the register is prohibited after the TQ1CCR1 register has been written and before the transfer timing is generated (until the crest (match between the 16-bit counter value and TQ1CCR0 register value) or the valley (match between the 16-bit counter value and 0001H)). The operation procedure is as follows.

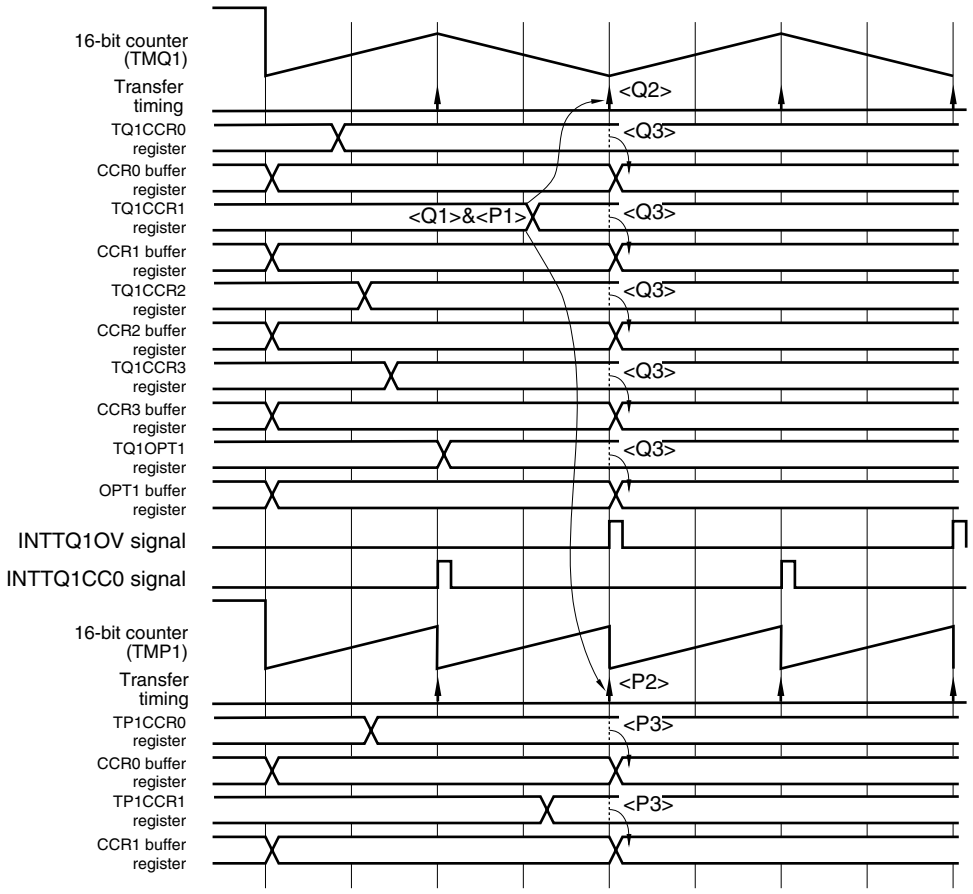
<1> Rewriting the TQ1CCR0, TQ1CCR2, TQ1CCR3, TQ1OPT1, TP1CCR0, and TP1CCR1 registers
Do not rewrite registers that do not have to be rewritten.

<2> Rewriting the TQ1CCR1 register
Rewrite the same value to the register even when it is not necessary to rewrite the TQ1CCR1 register.

<3> Holding the next rewriting pending until the transfer timing is generated
Rewrite the register next time after the INTTQ1OV or INTTQ1CC0 interrupt has occurred.

<4> Return to <1>.

Figure 9-26. Basic Operation in Batch Mode



[Operation of TMQ1]

<Q1> Write the TQ1CCR1 register

<Q2> The target timing is the first transfer timing after a write to the TQ1CCR1 register.

<Q3> The values are transferred all at once at the transfer timing.

[Operation of TMP1]

<P1> Write the TQ1CCR1 register

<P2> The target timing is the first transfer timing after a write to the TQ1CCR1 register.

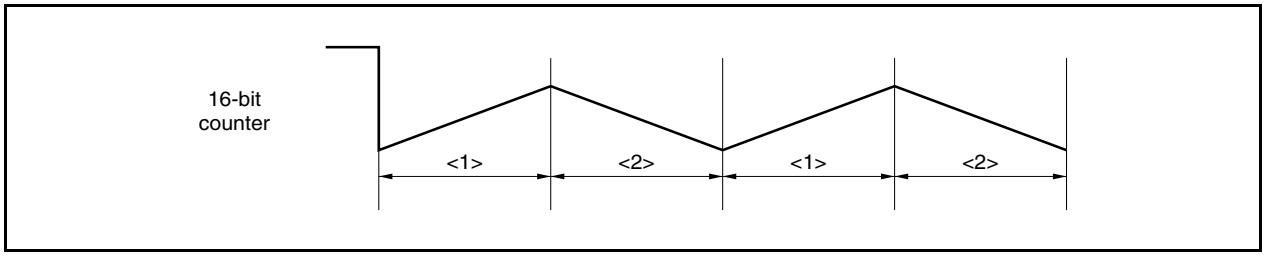
<P3> The values are transferred all at once at the transfer timing.

(b) Rewriting TQ1CCR0 register

When rewriting the TQ1CCR0 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 TQ1CCR0 register value) or at the valley (match between the 16-bit counter value and 0001H). Usually, it is recommended to rewrite the TQ1CCR0 register while the 16-bit counter is counting down, and transfer the register value at the transfer timing of the valley timing.

Figure 9-28 shows an example of rewriting the TQ1CCR0 register while the 16-bit counter is counting up (during period <1> in Figure 9-27). Figure 9-29 shows an example of rewriting the TQ1CCR0 register while the counter is counting down (during period <2> in Figure 9-27).

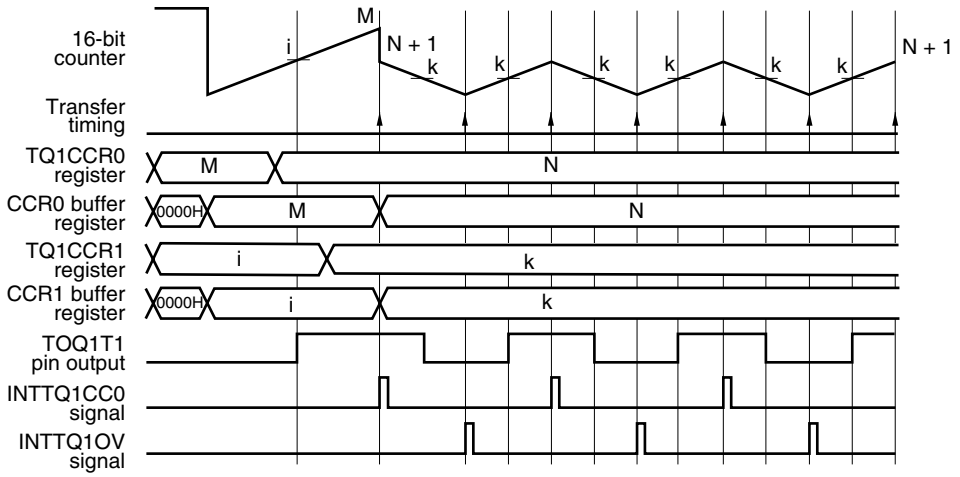
Figure 9-27. Basic Operation of 16-Bit Counter



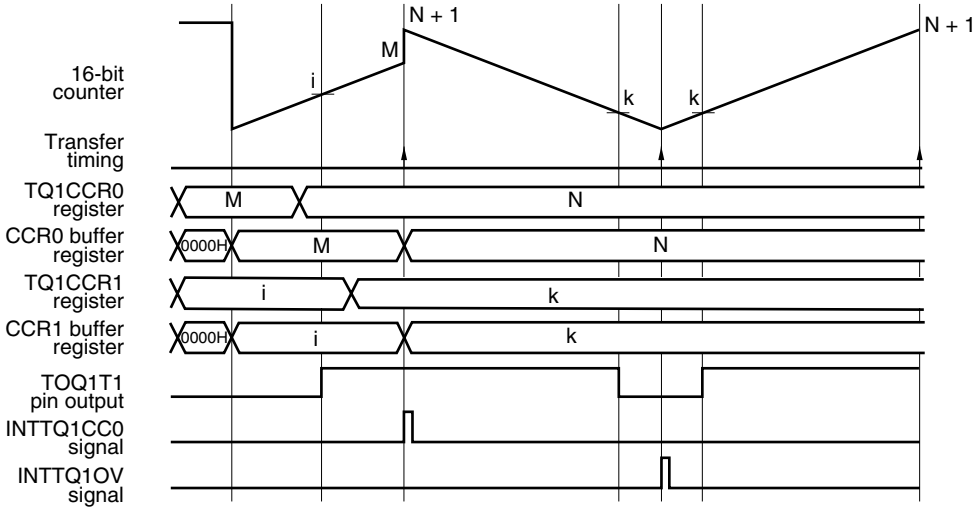
The transfer timing in Figure 9-28 is at the point where the crest timing occurs. While the 16-bit counter is counting down, the cycle changes and an asymmetrical triangular wave is output. Because the cycle changes, rewrite the duty factor (voltage data value).

Figure 9-28. Example of Rewriting TQ1CCR0 Register (During Counting Up)

(a) M > N



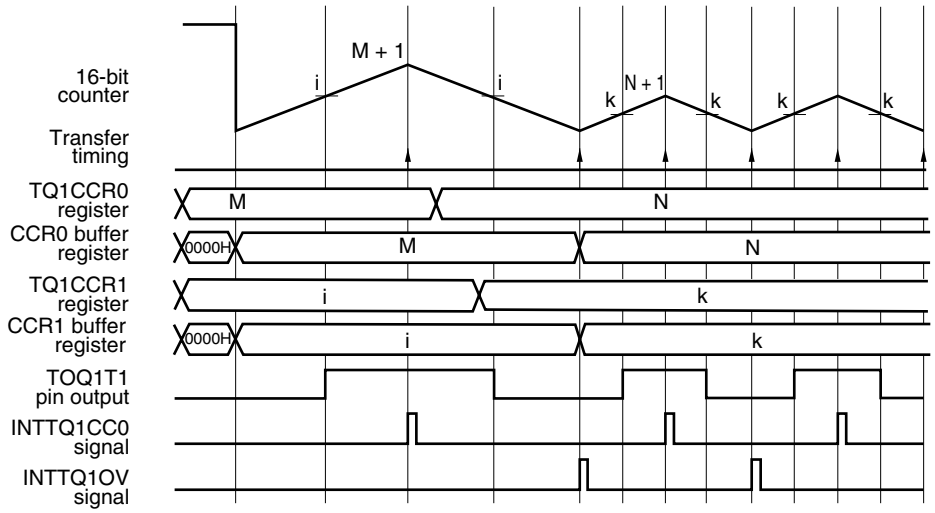
(b) M < N



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 TQ1CCR0 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 TQ1CCR0 register value) timing.

2. M: Value of CCR0 buffer register before rewriting
N: Value of CCR0 buffer register after rewriting

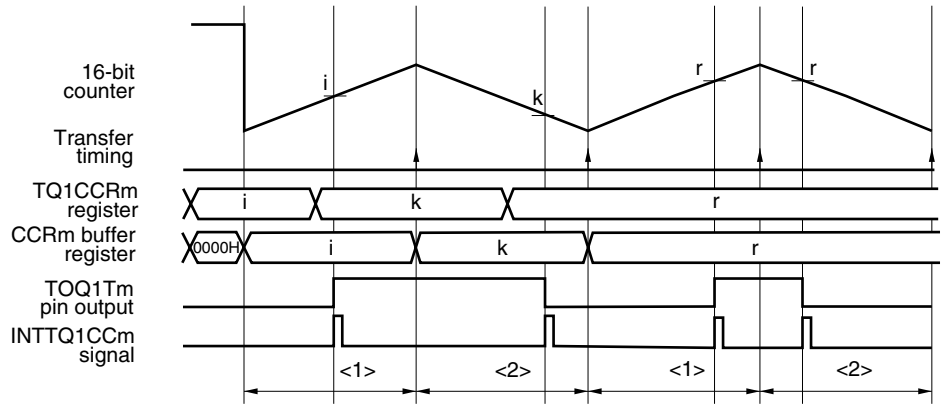
Figure 9-29. Example of Rewriting TQ1CCR0 Register (During Counting Down)



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 TQ1CCRm register

Figure 9-30. Example of Rewriting TQ1CCRm Register



Rewriting during period <1> (rewriting during counting up)

Because the TQ1CCRm register value is transferred at the transfer timing of the crest (match between the 16-bit counter value and TQ1CCR0 register value), an asymmetrical triangular wave is output.

Rewriting during period <2> (rewriting during counting down)

Because the TQ1CCRm 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 TQ1OPT1 register value

Do not set the TQ1OPT1.TQ1ID4 to TQ1OPT1.TQ1ID0 bits to other than 00000B. When using the interrupt culling function, rewrite the TQ1OPT1 register in the intermittent batch rewrite mode (transfer culling mode).

For details of rewriting the TQ1OPT1 register, see **9.4.3 Interrupt culling function**.

(3) Intermittent batch rewrite mode (transfer culling mode)

This mode is set by clearing the TQ1OPT0.TQ1CMS bit to 0 and setting the TQ1OPT2.TQ1RDE bit to 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 (INTTQ1CC0, INTTQ1OV) by interrupt culling.

For details of the interrupt culling function, see **9.4.3 Interrupt culling function**.

(a) Rewriting procedure

If data is written to the TQ1CCR1 register, the TQ1CCR0 to TQ1CCR3, TQ1OPT1, TP1CCR0, and TP1CCR1 registers are transferred all at once to the internal buffer register at the next transfer timing. Therefore, write to the TQ1CCR1 register last. Writing to the register is prohibited after the TQ1CCR1 register has been written until the transfer timing is generated (until the INTTQ1OV or INTTQ1CC0 interrupt occurs). The operation procedure is as follows.

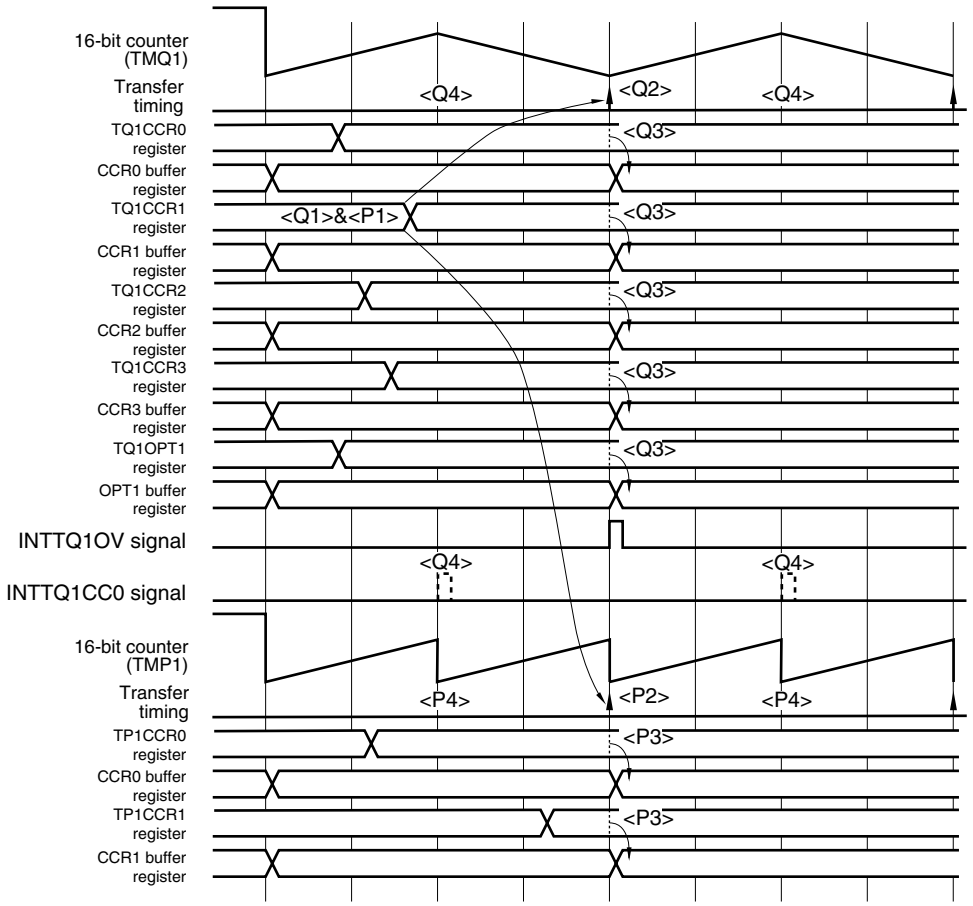
<1> Rewrite the TQ1CCR0, TQ1CCR2, TQ1CCR3, TQ1OPT1, TP1CCR0, and TP1CCR1 registers.
Do not rewrite registers that do not have to be rewritten.

<2> Rewrite the TQ1CCR1 register.
Rewrite the same value to the register even when it is not necessary to rewrite the TQ1CCR1 register.

<3> Hold the next rewriting pending until the transfer timing is generated.
Perform the next rewrite after the INTTQ1OV or INTTQ1CC0 interrupt has occurred.

<4> Return to <1>.

Figure 9-31. Basic Operation in Intermittent Batch Rewrite Mode



[TMQ1 operation]

<Q1> Write the TQ1CCR1 register.

<Q2> Rewrite the register at the transfer timing that is generated after the TQ1CCR1 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.

[TMP1 operation]

<P1> Write the TQ1CCR1 register.

<P2> Rewrite the register at the transfer timing that is generated after the TQ1CCR1 register has been rewritten.

<P3> The registers are transferred all at once at the transfer timing.

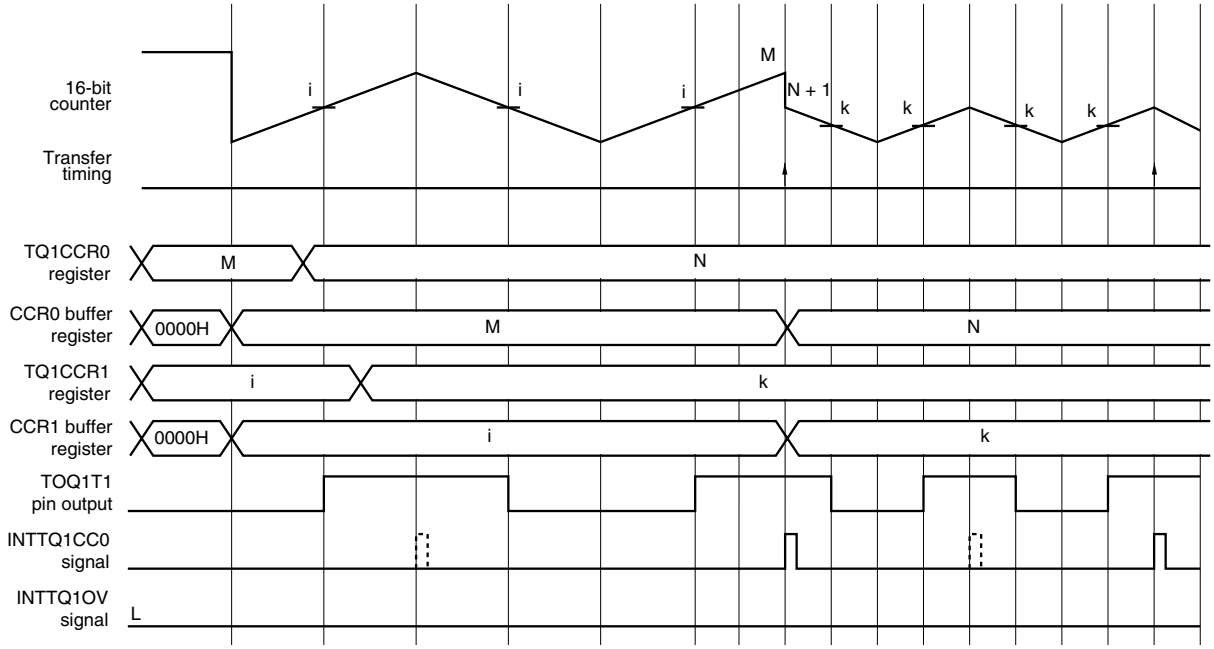
<P4> The transfer timing is also culled as the interrupts are culled.

Remark This is an example of the operation when the TQ1OPT1.TQ1ICE bit = 1, TQ1OPT1.TQ1IOE bit = 1, TQ1OPT1.TQ1ID4 to TQ1OPT1.TQ1ID0 bits = 00001.

(b) Rewriting TQ1CCR0 register

When rewriting the TQ1CCR0 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 9-32. Rewriting TQ1CCR0 Register (When Crest Interrupt Is Set)

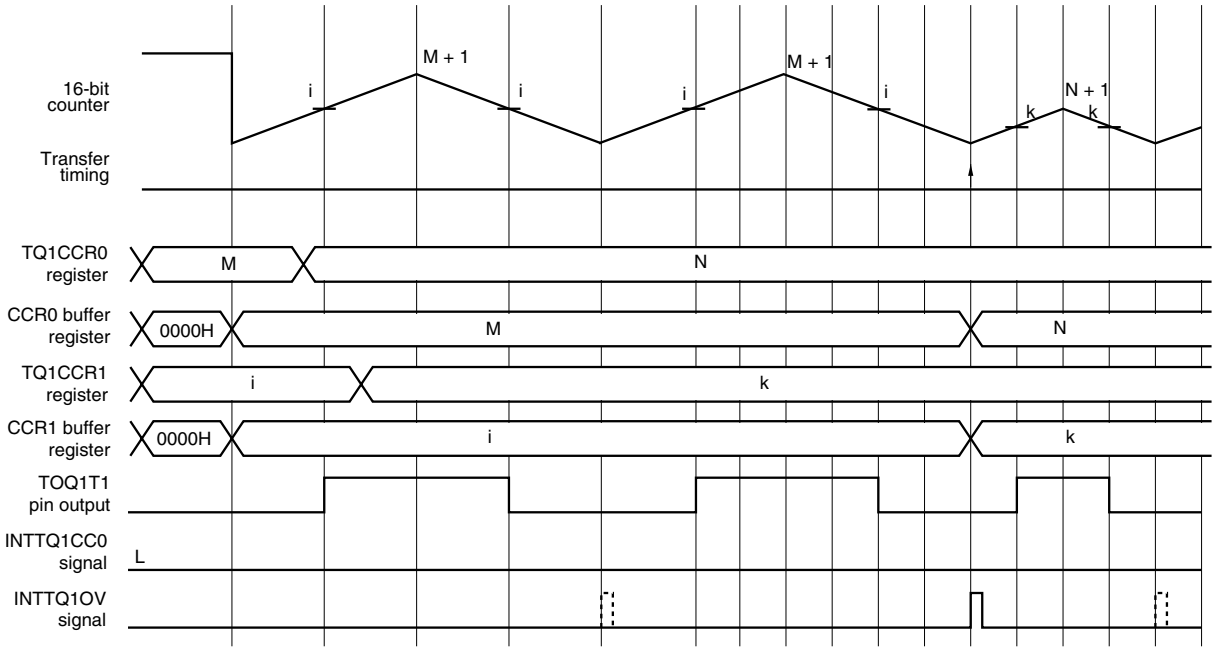


The transfer timing is generated when the crest interrupt occurs, the cycle of counting up and counting down changes, and an asymmetrical triangular wave is output.

Remarks 1. This is an example of the operation when the TQ1OPT1.TQ1ICE bit = 1, TQ1OPT1.TQ1IOE bit = 0, TQ1OPT1.TQ1ID4 to TQ1OPT1.TQ1ID0 bits = 00001.

2. : Culled interrupt

Figure 9-33. Rewriting TQ1CCR0 Register (When Valley Interrupt Is Set)



The transfer timing is generated when the valley interrupt occurs, the cycle of counting up becomes same as cycle of counting down, and a symmetrical triangular wave is output.

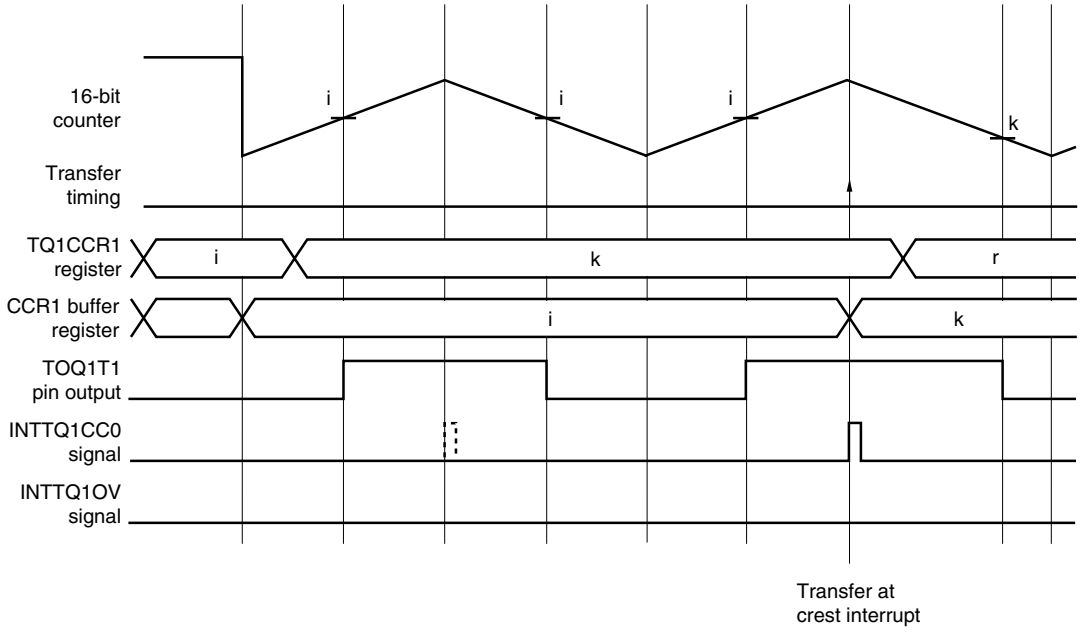
Remarks 1. This is an example of the operation when the TQ1OPT1.TQ1ICE bit = 0, TQ1OPT1.TQ1IOE bit = 1, TQ1OPT1.TQ1ID4 to TQ1OPT1.TQ1ID0 bits = 00001.

2. : Culled interrupt

(c) Rewriting TQ1CCR1 to TQ1CCR3 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 9-34. Rewriting TQ1CCR1 Register
(TQ1OPT1.TQ1ICE bit = 1, TQ1OPT1.TQ1IOE bit = 0, TQ1OPT1.TQ1ID4 to TQ1OPT1.TQ1ID0 = 00001)

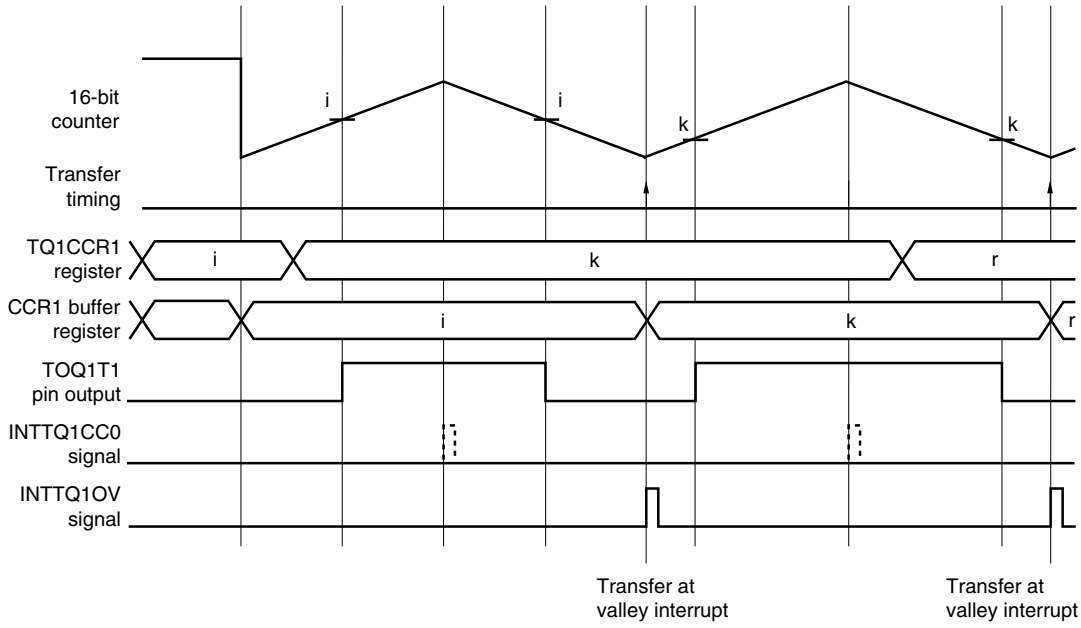


Remark : Culled interrupt

- 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 9-35. Rewriting TQ1CCR1 Register
(TQ1OPT1.TQ1ICE bit = 1, TQ1OPT1.TQ1IOE bit = 1, TQ1OPT1.TQ1ID4 to TQ1OPT1.TQ1ID0 = 00001)



Remark : Culled interrupt

(d) Rewriting TQ1OPT1 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 TQ1OPT1 register, see **9.4.3 Interrupt culling function**.

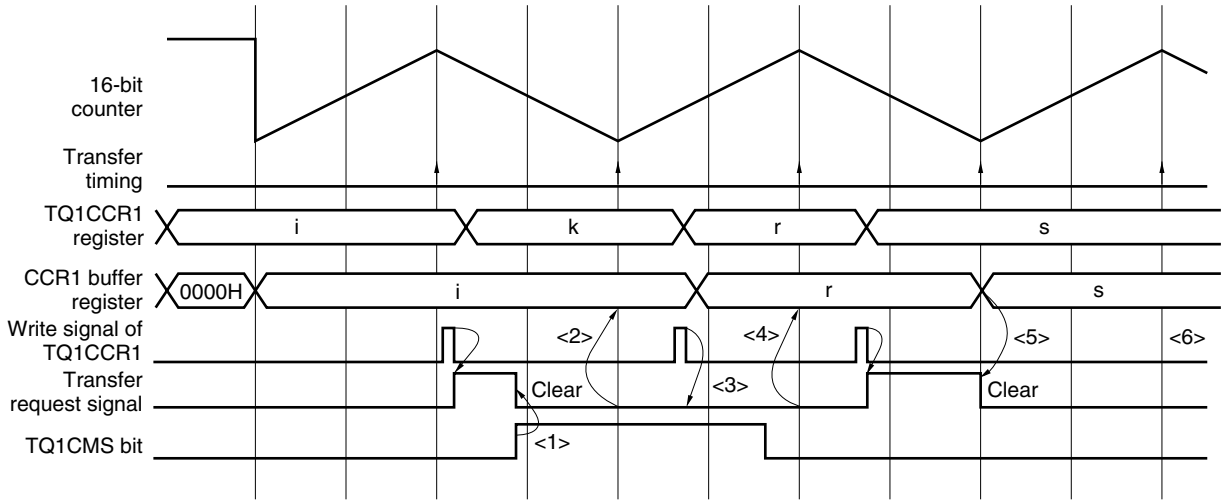
(4) Rewriting TQ1OPT0.TQ1CMS bit

The TQ1CMS bit can select the anytime rewrite mode and batch rewrite mode. This bit can be rewritten during timer operation (when TQ1CTL0.TQ1CE bit = 1). However, the operation and caution illustrated in Figure 9-31 are necessary.

If the TQ1CCR1 register is written when the TQ1CMS bit is cleared 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 TQ1CMS bit is set to 1.

Figure 9-36. Rewriting TQ1CMS Bit



<1> If the TQ1CCR1 register is rewritten when the TQ1CMS bit is 0, the transfer request signal is set.
If the TQ1CMS bit is set to 1 in this status, the transfer request signal is cleared.

<2> The register is not transferred because the TQ1CMS bit is set to 1 and the transfer request signal is cleared.

<3> The transfer request signal is not set even if the TQ1CCR1 register is written when the TQ1CMS bit is 1.

<4> The transfer request signal is not set even if the TQ1CCR1 register is written when the TQ1CMS bit is 1, so even if the TQ1CMS bit is cleared to 0, transfer does not occur at the subsequent transfer timing.

<5> The transfer request signal is set if the TQ1CCR1 register is written when the TQ1CMS 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.

9.4.5 TMP1 tuning operation for A/D conversion start trigger signal output

This section explains the tuning operation of TMP1 and TMQ1 in the 6-phase PWM output mode.

In the 6-phase PWM output mode, the tuning operation is performed with TMQ1 serving as the master and TMP1 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 INTTP1CC0 and INTTP1CC1 signals of TMP1 and the INTTQ1OV and INTTQ1CC0 signals of TMQ1.

(1) Tuning operation starting procedure

The TMP1 and TMQ1 registers should be set using the following procedure to perform the tuning operation.

(a) Setting of TMP1 register (stop the operations of TMQ1 and TMP1 (by clearing the TQ1CTL0.TQ1CE bit and TP1CTL0.TP1CE bit to 0))

- Set the TP1CTL1 register to 85H (set the tuning operation slave mode and free-running timer mode).
- Clear the TP1OPT0 register to 00H (select the compare register).
- Set an appropriate value to the TP1CCR0 and TP1CCR1 registers (set the default value for comparison for starting the operation).

(b) Setting of TMQ1 register

- Set the TQ1CTL1 register to 07H (set the master mode and 6-phase PWM output mode).
- Set an appropriate value to the TQ1IOC0 register (set the output mode of TOQ1T1 to TOQ1T3).
  However, clear the TQ1OL0 bit to 0 and set the TQ1OE0 bit to 1 (enable positive phase output). Unless this setting is made, the crest interrupt (INTTQ1CC0) and valley interrupt (INTT010V) do not occur. Consequently, the conversion start trigger signal of A/D converters 0 and 1 is not correctly generated.
- Clear the TQ1OPT0 register to 00H (select the compare register).
- Set an appropriate value to the TQ1CCR0 to TQ1CCR3 registers (set the default value for comparison for starting the operation).
- Set the TQ1CTL0 register to 0xH (clear the TQ1CE bit to 0 and set the operating clock of TMQ1).
  The operating clock of TMQ1 set by the TQ1CTL0 register is also supplied to TMP1, and the count operation is performed at the same timing. The operating clock of TMP1 set by the TP1CTL0 register is ignored.

(c) Setting of TMQOP1 (TMQ1 option) register

- Set an appropriate value to the TQ1OPT1 and TQ1OPT2 registers.
- Set an appropriate value to the TQ1IOC3 register (set TOQ1B1 to TOQ1B3 in the output mode).
- Set an appropriate value to the TQ1DTC register (set the default value for comparison for starting the operation).

(d) Setting of alternate function

- Set the alternate function to the port by setting the port control mode.

(e) Set the TP1CE bit to 1 and set the TQ1CE bit to 1 immediately after that to start the 6-phase PWM output operation

Rewriting the TQ1CTL0, TQ1CTL1, TP1CTL0, and TP1CTL1 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 TQ1CTL0.TQ1CE bit to clear it is permitted. Manipulating (reading/writing) the other TMQ1, TMP1, and TMQ1 option registers is prohibited until the TP1CTL0.TP1CE bit is set to 1 and then the TQ1CE 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 TMP1 and TMQ1 registers using the following procedure.

<1> Clear the TQ1CTL0.TQ1CE bit to 0 and stop the timer operation.

<2> Clear the TP1CTL0.TP1CE bit to 0 so that TMP1 can be separated.

<3> Stop the timer output by using the TQ1IOC0 register.

<4> Clear the TP1CTL1.TP1SYE bit to 0 to clear the tuning operation.

Caution Manipulating (reading/writing) the other TMQ1, TMP1, and TMQ1 option registers is prohibited until the TQ1CE bit is set to 0 and then the TP1CE bit is set to 0.

(3) When not tuning TMP1

When the match interrupt signal of TMP1 is not necessary as the conversion trigger source that starts A/D converters 0 and 1, TMP1 can be used independently as a separate timer without being tuned. In this case, the match interrupt signal of TMP1 cannot be used as a trigger source to start A/D conversion in the 6-phase PWM output mode. Therefore, fix the TQ1OPT2.TQ1AT00 to TQ1OPT2.TQ1AT03 bits and the TQ1OPT3.TQ1AT10 to TQ1OPT3.TQ1AT13 bits to 0.

The other control bits can be used in the same manner as when TMP1 is tuned.

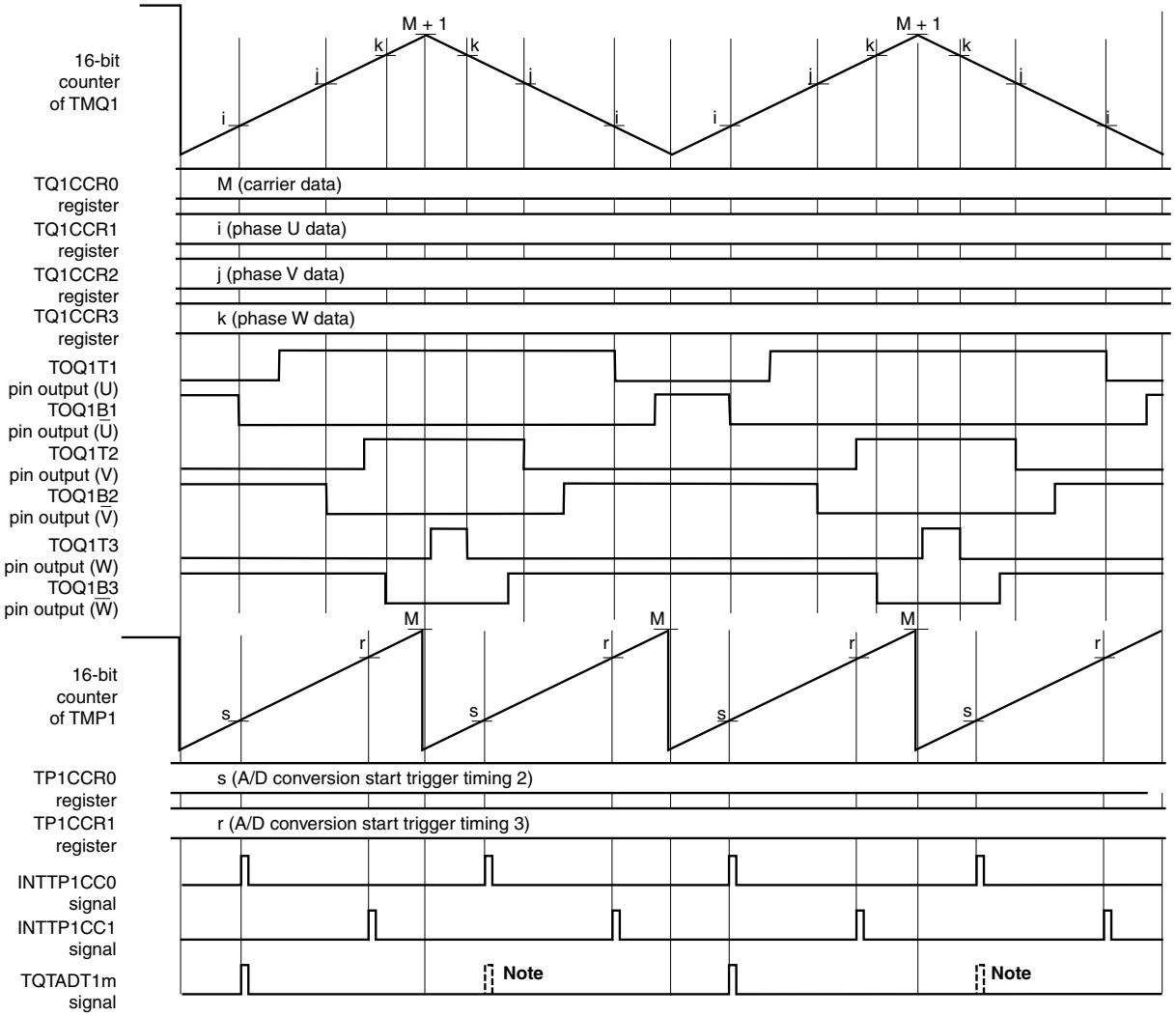
If TMP1 is not tuned, the compare registers (TP1CCR0 and TP1CCR1) of TMP1 are not affected by the setting of the TQ1OPT0.TQ1CMS and TQ1OPT2.TQ1RDE bit. For the initialization procedure when TMP1 is not tuned, see (b) to (e) in **9.4.5 (1) Tuning operation starting procedure**. (a) is not necessary because it is a step used to set TMP1 for the tuning operation.

(4) Basic operation of TMP1 during tuning operation

The 16-bit counter of TMP1 only counts up. The 16-bit counter is cleared by the set cycle value of the TQ1CCR0 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 TMQ1 when it counts up. However, it is not the same when the 16-bit counter of TMP1 counts down.

- When TMQ1 counts up (same value)
  16-bit counter of TMQ1: 0000H → M (counting up)
  16-bit counter of TMP1: 0000H → M (counting up)
- When TMQ1 counts down (not same value)
  16-bit counter of TMQ1: M + 1 → 0001H (counting down)
  16-bit counter of TMP1: 0000H → M (counting up)

Figure 9-37. TMP1 During Tuning Operation



Note The TQTADT10 signal is masked by the TQ1OPT2.TQ1ATM02 and TQ1OPT2.TQ1ATM03 bits. The TQTADT11 signal is masked by the TQ1OPT3.TQ1ATM12 and TQ1OPT3.TQ1ATM13 bits.

Remark m = 0, 1

9.4.6 A/D conversion start trigger output function

The V850ES/IE2 has a function to select four trigger sources (INTTQ1OV, INTTQ1CC0, INTTP1CC0, INTTP1CC1) to generate the A/D conversion start trigger signal (TQTADT00, TQTADT01) of A/D converters 0 and 1.

The trigger sources are specified by the TQ1OPT2.TQ1AT00 to TQ1OPT2.TQ1AT03 and TQ1OPT3.TQ1AT10 to TQ1OPT3.TQ1AT13 bits.

- TQ1AT00, TQ1AT10 bits = 1:
  A/D conversion start trigger signal generated when INTTQ1OV (counter underflow) occurs.
- TQ1AT01, TQ1AT11 bits = 1:
  A/D conversion start trigger signal generated when INTTQ1CC0 (cycle match) occurs.
- TQ1AT02, TQ1AT12 bits = 1:
  A/D conversion start trigger signal generated when INTTP1CC0 (match of TP1CCR0 register of TMP1 during tuning operation) occurs.
- TQ1AT03, TQ1AT13 bits = 1:
  A/D conversion start trigger signal generated when INTTP1CC1 (match of TP1CCR1 register of TMP1 during tuning operation) occurs.

The A/D conversion start trigger signals selected by the TQ1AT00 to TQ1AT03 and TQ1AT10 to TQ1AT13 bits are ORed and output. Therefore, two or more trigger sources can be specified at the same time.

The INTTQ1OV and INTTQ1CC0 signals selected by the TQ1AT00, TQ1AT01, TQ1AT10, and TQ1AT11 bits are culled interrupt signals.

Therefore, these signals are output after the interrupts have been culled and, unless interrupt output is enabled (TQ1OPT1.TQ1ICE, TQ1OPT1.TQ1IOE bits), the A/D conversion start trigger is not output.

The trigger sources (INTTP1CC0 and INTTP1CC1) from TMP1 have a function to mask the A/D conversion start trigger signal depending on the status of the count-up/count-down of the 16-bit counter, if so set by the TQ1AT02, TQ1AT03, TQ1AT12, and TQ1AT13 bits.

- TQ1ATM02, TQ1ATM12 bits:
  Correspond to the TQ1AT02 and TQ1AT12 bits and control INTTP1CC0 (match interrupt signal) of TMP1.
  - TQ1ATM02, TQ1ATM12 bits = 0
    The A/D conversion start trigger signal is output when the 16-bit counter counts up (TQ1OPT0.TQ1CUF bit = 0), and the A/D conversion start trigger signal is not output when the 16-bit counter counts down (TQ1OPT0.TQ1CUF bit = 1).
  - TQ1ATM02, TQ1ATM12 bits = 1
    The A/D conversion start trigger signal is output when the 16-bit counter counts down (TQ1OPT0.TQ1CUF bit = 1), and the A/D conversion start trigger signal is not output when the 16-bit counter counts up (TQ1OPT0.TQ1CUF bit = 0).

- TQ1ATM03, TQ1ATM13 bits:
  Correspond to the TQ1AT03 and TQ1AT13 bits and control INTTP1CC1 (match interrupt signal) of TMP1.
  - TQ1ATM03, TQ1ATM13 bits = 0
    The A/D conversion start trigger signal is output when the 16-bit counter counts up (TQ1OPT0.TQ1CUF bit = 0), and the A/D conversion start trigger signal is not output when the 16-bit counter counts down (TQ1OPT0.TQ1CUF bit = 1).
  - TQ1ATM03, TQ1ATM13 bits = 1
    The A/D conversion start trigger signal is output when the 16-bit counter counts down (TQ1OPT0.TQ1CUF bit = 1), and the A/D conversion start trigger signal is not output when the 16-bit counter counts up (TQ1OPT0.TQ1CUF bit = 0).

The TQ1ATM03, TQ1ATM02, TQ1AT03 to TQ1AT00, TQ1ATM13, TQ1ATM12, and TQ1AT13 to TQ1AT10 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 rewrite mode.

Cautions 1. The A/D conversion start trigger signal output that is set by the TQ1AT02, TQ1AT03, TQ1AT12, and TQ1AT13 bits can be used only when TMP1 is performing a tuning operation as the slave timer of TMQ1. If TMQ1 and TMP1 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 TOQ10 signal output is internally used to identify whether the 16-bit counter is counting up or down. Therefore, enable TOQ10 pin output by clearing the TQ1IOC0.TQ1OL0 bit to 0 and setting the TQ1IOC0.TQ1OE0 bit to 1.

Figure 9-38. Example of A/D Conversion Start Trigger (TQTADT10) Signal Output (TQ1OPT1.TQ1ICE Bit = 1, TQ1OPT1.TQ1IOE Bit = 1, TQ1OPT1.TQ1ID4 to TQ1OPT1.TQ1ID0 Bits = 00000: Without Interrupt Culling)

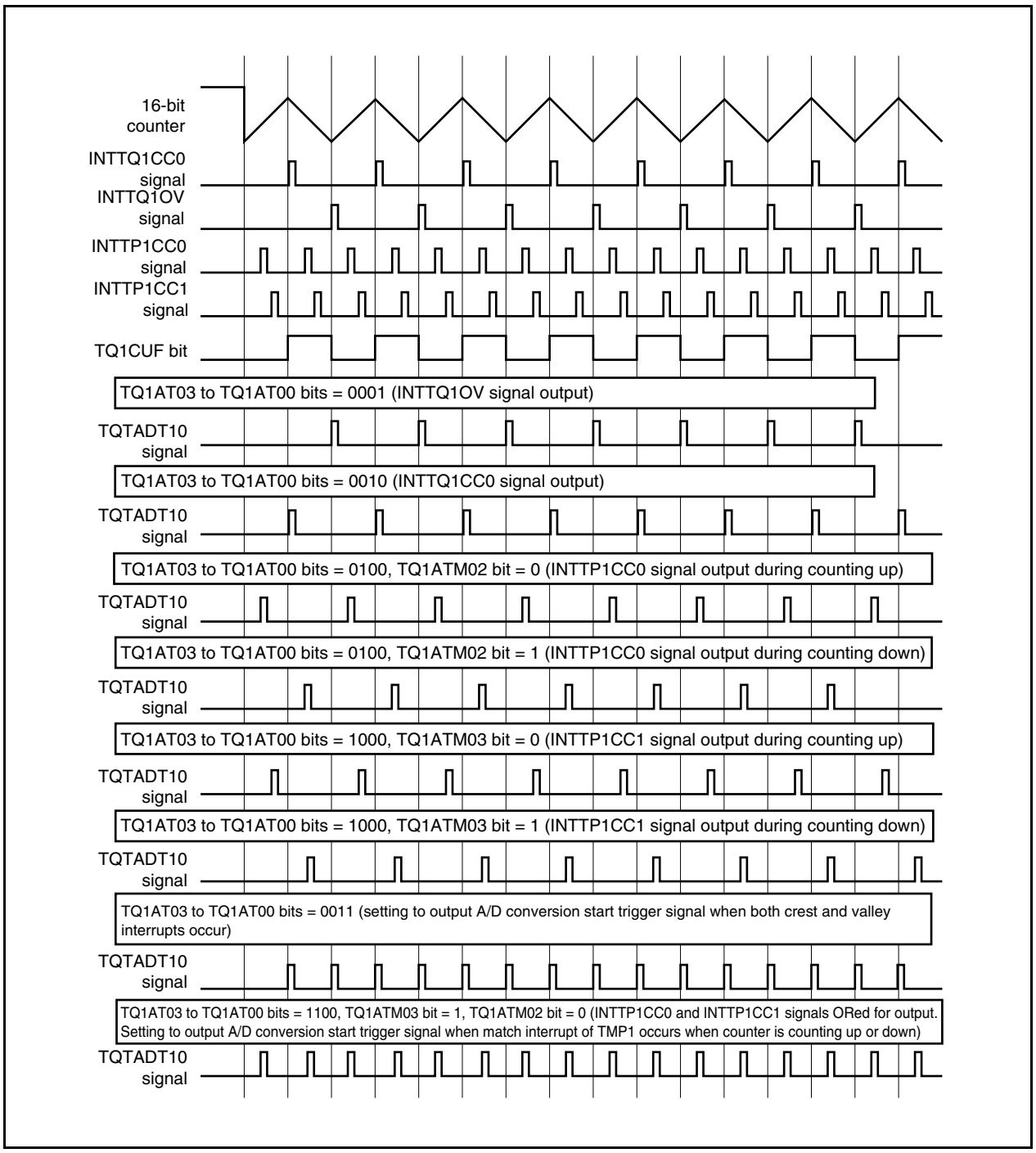
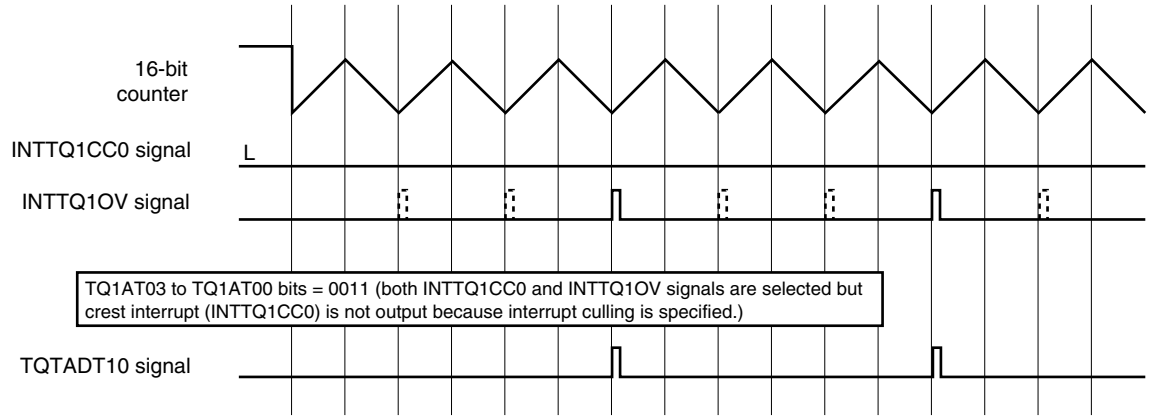
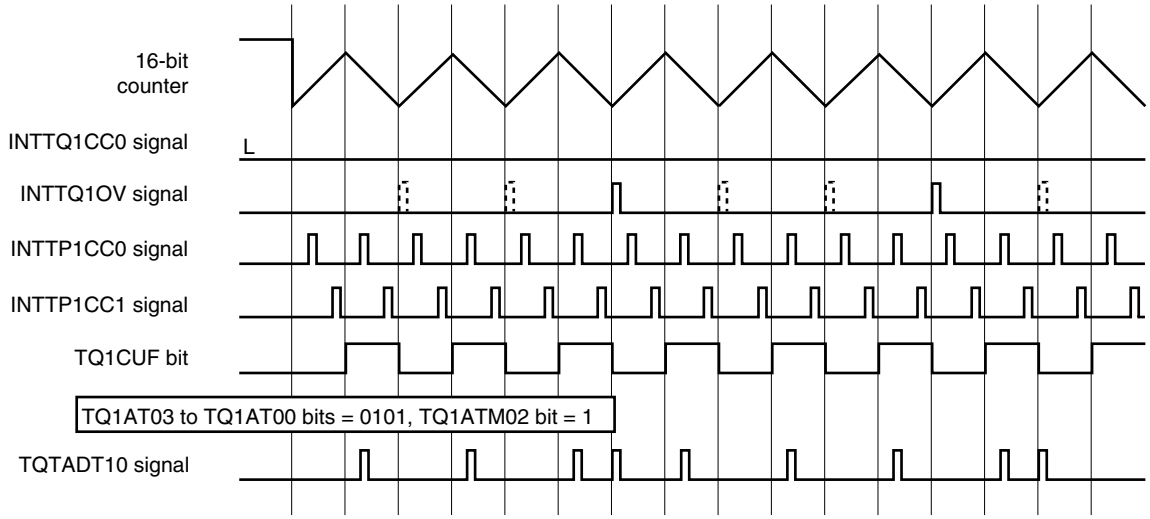


Figure 9-39. Example of A/D Conversion Start Trigger (TQTADT10) Signal Output (TQ1OPT1.TQ1ICE Bit = 0, TQ1OPT1.TQ1IOE Bit = 1, TQ1OPT1.TQ1ID4 to TQ1OPT1.Q1ID0 Bits = 00010: With Interrupt Culling) (1)



Remark : Culled interrupt

Figure 9-40. Example of A/D Conversion Start Trigger (TQTADT10) Signal Output (TQ1OPT1.TQ1ICE Bit = 0, TQ1OPT1.TQ1IOE Bit = 1, TQ1OPT1.TQ1ID4 to TQ1OPT1.TQ1ID0 Bits = 00010: With Interrupt Culling) (2)



Caution The INTTQ1CC0 signal is culled but the INTTP1CC0 signal is not.

Remark : Culled interrupt

(1) Operation under boundary condition (operation when 16-bit counter matches INTTP1CC0 signal)

Table 9-3. Operation When TQ1CCR0 Register = M, TQ1ATm2 Bit = 1, TQ1ATMm2 Bit = 0 (Counting Up Period Selected)

Value of TP1CCR0 Register	Value of 16-bit Counter of TMQ1	Value of 16-bit Counter of TMP1	Status of 16-bit Counter of TMQ1	TQTADT1m Signal Output by INTTP1CC0 Signal
0000H	0000H	0000H	–	Output
0000H	M + 1	0000H	–	Not output
0001H	0001H	0001H	Count up	Output
0001H	M	0001H	Count down	Not output
M	M	M	Count up	Output
M	0001H	M	Count down	Not output

Table 9-4. Operation When TQ1CCR0 Register = M, TQ1ATm2 Bit = 1, TQ1ATMm2 Bit = 1 (Counting Down Period Selected)

Value of TP1CCR0 Register	Value of 16-bit Counter of TMQ1	Value of 16-bit Counter of TMP1	Status of 16-bit Counter of TMQ1	TQTADT1m Signal Output by INTTP1CC0 Signal
0000H	0000H	0000H	–	Not output
0000H	M + 1	0000H	–	Output
0001H	0001H	0001H	Count up	Not output
0001H	M	0001H	Count down	Output
M	M	M	Count up	Not output
M	0001H	M	Count down	Output

Caution The TP1CCRm register enables setting of "0" to "M" when the TQ1CCR0 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 TMP1 is cleared by "M". Therefore, the TQTADT1m signal is not output.

Remark m = 0, 1

CHAPTER 10 WATCHDOG TIMER FUNCTIONS

10.1 Functions

The watchdog timer has the following functions.

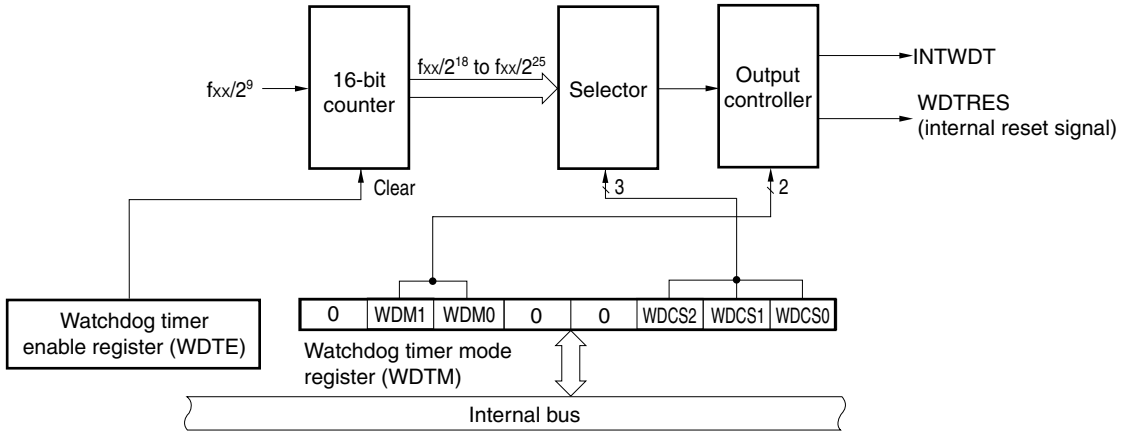
- Reset mode: Reset operation upon overflow of the watchdog timer (generation of WDTRES signal)
- Non-maskable interrupt request mode:
  Non-maskable interrupt operation upon overflow of the watchdog timer (generation of INTWDT signal)

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^{25}/fxx) do not need to be changed.

10.2 Configuration

The block diagram of the watchdog timer is shown below.

Figure 10-1. Block Diagram of Watchdog Timer



Remark $f_{XX}/2^9$: 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 includes the following hardware.

Table 10-1. Configuration of Watchdog Timer

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

10.3 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 10-2 Overflow Time.

2. Be sure to clear bits 3, 4, and 7 to "0".

Table 10-2. Overflow Time

WDCS2	WDCS1	WDCS0	Overflow Time	f_{XX} = 20 MHz
0	0	0	$2^{18}/f_{XX}$	13.1 ms
0	0	1	$2^{19}/f_{XX}$	26.2 ms
0	1	0	$2^{20}/f_{XX}$	52.4 ms
0	1	1	$2^{21}/f_{XX}$	104.9 ms
1	0	0	$2^{22}/f_{XX}$	209.7 ms
1	0	1	$2^{23}/f_{XX}$	419.4 ms
1	1	0	$2^{24}/f_{XX}$	838.9 ms
1	1	1	$2^{25}/f_{XX}$	1677.7 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.

The WDTE 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).

10.4 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.

10.5 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 11 A/D CONVERTERS 0 AND 1

11.1 Features

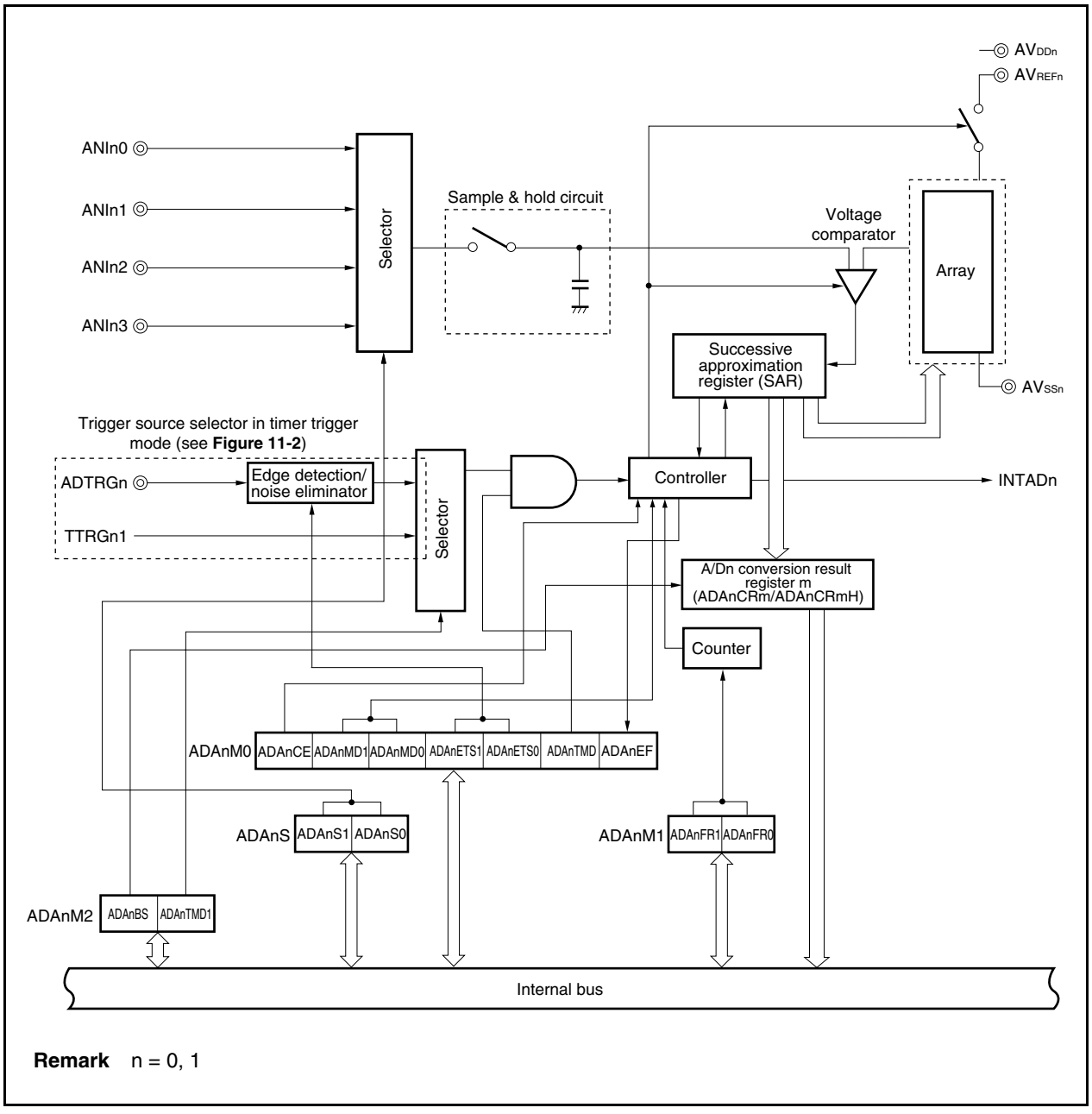
- Two 10-bit resolution A/D converter circuits (A/D converters 0 and 1)
  Simultaneous sampling of two circuits possible
- Analog input
  Two circuits, total of eight channels
  A/D converter 0: ANI00 to ANI03 (4 channels)
  A/D converter 1: ANI10 to ANI13 (4 channels)
- A/D conversion result registers 0m and 1m (ADA0CRm and ADA1CRm)
  10 bits × 4 × 2
- A/D conversion trigger mode
  - Software trigger mode
  - Hardware trigger mode
    External trigger mode
    Timer trigger mode
- A/D conversion operation mode
  Continuous select mode
  Continuous scan mode
  One-shot select mode
  One-shot scan mode
- Buffer mode
  1-buffer mode
  4-buffer mode
- Successive approximation method
- Operating voltage range
  $V_{DD} = EV_{DD} = AV_{DDn} = AV_{REFn}$ = 4.5 to 5.5 V

Remark m = 0 to 3
n = 0, 1

11.2 Configuration

The block diagram is shown below.

Figure 11-1. Block Diagram of A/D Converters 0 and 1



Remark n = 0, 1

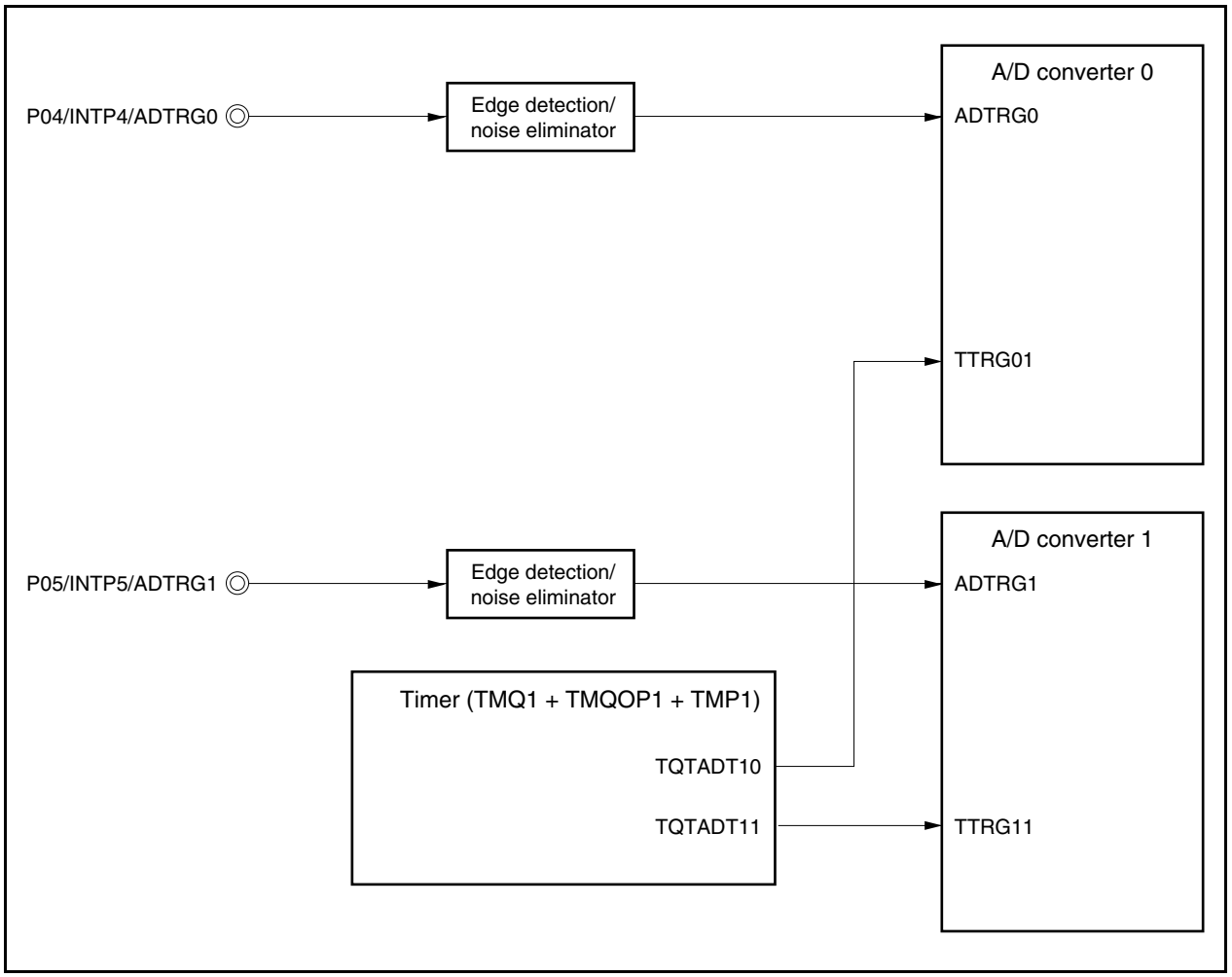
Cautions 1. If there is noise at the analog input pins (ANIn0 to ANIn3) or at the A/D converter reference voltage input pin (AV_{REFn}), that noise may generate an illegal conversion result (n = 0, 1). Software processing will be needed to avoid a negative effect on the system from this illegal conversion 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_{REFn} range to the pins that are used as input pins of A/D converters 0 and 1.

Figure 11-2. Block Diagram of Trigger Source Selector in Timer Trigger Mode



A/D converters 0 and 1 consist of the following hardware.

Table 11-1. Configuration of A/D Converters 0 and 1

Item	Configuration
Analog input	ANI00 to ANI03, ANI10 to ANI13 (two circuits, total of eight channels)
Registers	Successive approximation register (SAR) A/Dn conversion result registers 0 to 3 (ADAnCR0 to ADAnCR3) A/Dn conversion result registers 0H to 3H (ADAnCR0H to ADAnCR3H)
Control registers	A/D converter n mode register 0 (ADAnM0) A/D converter n mode register 1 (ADAnM1) A/D converter n mode register 2 (ADAnM2) A/D converter n channel specification register 0 (ADAnS)

Remark n = 0, 1

(1) Selector

The input circuit selects the analog input pin (ANIn0 to ANIn3) according to the mode set by the ADAnM0, ADAnM1, ADAnM2, and ADAnS registers and sends the input to the sample & hold circuit (n = 0, 1).

(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. 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 ($1/2AV_{REFn}$) 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 ($1/2AV_{REFn}$), the MSB of the SAR is reset.

After that, bit 8 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 9, to which the result has been already set.

Bit 9 = 0: ($1/4AV_{REFn}$)
Bit 9 = 1: ($3/4AV_{REFn}$)

The voltage tap of the array and the analog input voltage are compared and bit 8 of the SAR is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of array: Bit 8 = 1
Analog input voltage ≤ Voltage tap of array: Bit 8 = 0

Comparison is continued like this to bit 0 of the SAR.

(4) Array

The array generates comparison voltage by the voltage input from an analog input pin (ANIn0 to ANIn3) (n = 0, 1).

(5) Successive approximation register (SAR)

The SAR is a 10-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 3 (ADAnCR0 to ADAnCR3) (n = 0, 1). 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 3 (ADAnCR0 to ADAnCR3), A/Dn conversion result registers 0H to 3H (ADAnCR0H to ADAnCR3H) (n = 0, 1)

The ADAnCR0 to ADAnCR3 and ADAnCR0H to ADAnCR3H 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 ADAnCR0 to ADAnCR3 registers. 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 ADAnCR0H to ADAnCR3H registers. To read the result of A/D conversion in 16-bit units, specify the ADAnCR0 to ADAnCR3 registers. To read the higher 8 bits, specify the ADAnCR0H to ADAnCR3H registers.

(7) A/D converter n mode register 0 (ADAnM0) (n = 0, 1)

This register is used to specify the operation mode and controls the conversion operation.

(8) A/D converter n mode register 1 (ADAnM1) (n = 0, 1)

This register is used to set the number of conversion clocks of the analog input to be A/D converted.

(9) A/D converter n channel specification register (ADAnS) (n = 0, 1)

This register is used to specify the analog input pin to be A/D converted.

(10) A/D converter n mode register 2 (ADA2M2) (n = 0, 1)

This register is used to specify the buffer mode and specify the mode in the hardware trigger mode.

(11) ANIn0 to ANIn3 pins (n = 0, 1)

The ANIn0 to ANIn3 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 ANIn0 to ANIn3 do not exceed the rated values. If a voltage higher than or equal to AV_{REFn} 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.

(12) AV_{REF0} and AV_{REF1} pins

This is the pin for inputting the reference voltage of A/D converters 0 and 1. It converts signals input to the ANIn0 to ANIn3 pins to digital signals based on the voltage applied between AV_{REFn} and AV_{SSn} (n= 0, 1). Always make the potential at the AV_{REFn} pin the same as that at the EV_{DD} pin even when A/D converters 0 and 1 are not used.

The operating voltage range of the AV_{REFn} pin is $V_{DD} = EV_{DD} = AV_{DDn} = AV_{REFn}$ = 4.5 to 5.5 V.

(13) AV_{SS0} and AV_{SS1} pins

These pins are the ground pins of A/D converters 0 and 1. Always make the potential at the AV_{SSn} pin the same as that at the EV_{SS} pin even when A/D converters 0 and 1 are not used.

(14) AV_{DD0} and AV_{DD1} pins

These pins are the analog power supply pins of A/D converters 0 and 1.

Supply the same potential to the AV_{DD0} and AV_{DD1} pins.

Always make the potential at the AV_{DDn} pin the same as that at the EV_{DD} pin even when A/D converters 0 and 1 are not used.

The operating voltage range of the AV_{DDn} pin is $V_{DD} = EV_{DD} = AV_{REFn} = AV_{DDn} = 4.5$ to 5.5 V.

(15) Controller

This circuit executes control, such as enabling/disabling A/D converters 0 and 1 and selecting the operation mode and trigger mode.

11.3 Control Registers

A/D converters 0 and 1 are controlled by the following registers.

- A/D converter n mode registers 0 to 2 (ADAnM0 to ADAnM2)
- A/D converter n channel specification register (ADAnS)

The following registers are also used.

- A/Dn conversion result registers 0 to 3 (ADAnCR0 to ADAnCR3)
- A/Dn conversion result registers 0H to 3H (ADAnCR0H to ADAnCR3H)

(1) A/D converter n mode register 0 (ADAnM0)

The ADAnM0 register is an 8-bit 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. Writing executed to bit 0 is ignored.

Reset sets this register to 00H.

After reset: 00H R/W Address: ADA0M0 FFFFF200H, ADA1M0 FFFFF220H

ADAnM0 (n = 0, 1)

<7>	6	5	4	3	2	1	0
ADAnCE	0	ADAnMD1	ADAnMD0	ADAnETS1	ADAnETS0	ADAnTMD	ADAnEF

ADAnCE	A/D conversion operation control
0	Stop conversion operation
1	Start conversion operation

ADAnMD1	ADAnMD0	Operation mode specification
0	0	Continuous select mode
0	1	Continuous scan mode
1	0	One-shot select mode
1	1	One-shot scan mode

ADAnETS1	ADAnETS0	Specification of external trigger (ADTRGn) valid edge
0	0	No edge detection (external trigger invalid)
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

ADAnTMD	Trigger mode specification
0	Software trigger mode
1	Hardware trigger mode[Note]

ADAnEF	Status of A/D converter n
0	A/D conversion stopped
1	A/D conversion operating

Note When the hardware trigger mode is selected, select a trigger mode by using the ADAnM2.ADAnTMD1 bit.

Cautions 1. In the software trigger mode, conversion is triggered when 1 is written to the ADAnCE bit. In the hardware trigger mode (external trigger mode or timer trigger mode), the trigger signal stands by when 1 is written to the ADAnCE bit.
The ADAnCE 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 ADAnCE bit.

2. If the ADAnM0, ADAnM2, and ADAnS registers are written during A/D conversion (ADAnEF bit = 1), the operation is performed as follows in each mode.

- **In software trigger mode**
  A/D conversion is stopped and executed again from the beginning.
- **In hardware trigger mode**
  A/D conversion is stopped and the trigger standby state is restored again.

(2) A/D converter n mode register 1 (ADAnM1)

The ADAnM1 register is an 8-bit register that specifies the number of conversion clocks.

The number of conversion clocks includes the number of sampling clocks.

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: ADA0M1 FFFFF201H, ADA1M1 FFFFF221H

	7	6	5	4	3	2	1	0
ADAnM1 (n = 0, 1)	0	0	0	0	0	0	ADAnFR1	ADAnFR0

Cautions 1. See Table 11-2 Number of Conversion Clocks for the ADAnFR1 and ADAnFR0 bits.

2. When ADAnM0.ADAnCE bit = 1 (conversion enabled), changing the ADAnFR1 and ADAnFR0 bits is prohibited.

3. Be sure to clear bits 2 to 7 to "0".

Table 11-2. Number of Conversion Clocks

ADAnFR1	ADAnFR0	Number of Conversion Clocks (f_{XX} = 20 MHz)	Number of Stabilized Clocks (f_{XX} = 20 MHz)	Number of Trigger Acknowledge Clocks (f_{XX} = 20 MHz)
0	0	Setting prohibited	–	–
0	1	62 (3.10 μs)	33	6
1	0	93 (4.65 μs)	50	7
1	1	124 (6.20 μs)	54	8

- Total number of A/D conversion clocks

Trigger Mode		Operation Mode		Total Number of A/D Conversion Clocks by First Trigger After Setting ADAnCE Bit = 1	Total Number of A/D Conversion Clocks by Second or Subsequent Trigger After Setting ADAnCE Bit = 1
Software trigger		Continuous select	1 buffer	Number of stabilized clocks + number of trigger acknowledgment clocks + number of conversion clocks	Number of conversion clocks
			4 buffers		
		Continuous scan	1 buffer		
		One-shot select	1 buffer		– (Conversion end with one conversion)
			4 buffers		Number of conversion clocks
		One-shot scan	1 buffer		Number of conversion clocks[Note 1]
Hardware trigger	Timer trigger	Continuous select/ one-shot select	1 buffer	Number of trigger acknowledgment clocks + number of conversion clocks[Note 2]	– (Conversion end with one conversion)
			4 buffers		Number of conversion clocks
		Continuous scan/ one-shot scan	1 buffer		Number of conversion clocks[Note 1]
	External trigger	Continuous select/ one-shot select	1 buffer	Noise elimination time + number of trigger acknowledgment clocks + number of conversion clocks[Note 2]	– (Conversion end with one conversion)
			4 buffers		Number of conversion clocks
		Continuous scan/ one-shot scan	1 buffer		Number of conversion clocks[Note 1]

Notes 1. When two or more channels are scanned (ADAnS register ≠ 00H)

2. The stabilization time of the number of stabilized clocks elapses after the ADAnCE bit is set from 0 to 1. If the trigger is input during this time, the trigger is acknowledged after the lapse of the stabilization time. As a result, the maximum total number of A/D conversion clocks is the above stabilization time plus the number of stabilized clocks.

(3) A/D converter n channel specification register (ADAnS)

The ADAnS register is an 8-bit register that specifies the analog input 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: ADA0S FFFFF202H, ADA1S FFFFF222H

	7	6	5	4	3	2	1	0
ADAnS (n = 0, 1)	0	0	0	0	0	0	ADAnS1	ADAnS0

ADAnS1	ADAnS0	Select mode	Scan mode
0	0	ANIn0	ANIn0
0	1	ANIn1	ANIn0, ANIn1
1	0	ANIn2	ANIn0 to ANIn2
1	1	ANIn3	ANIn0 to ANIn3

Cautions 1. If the ADAnS register is written during A/D conversion (ADAnM0.ADAnEF bit = 1), the operation is performed as follows in each mode.

- **In software trigger mode**
  A/D conversion is stopped and executed again from the beginning.
- **In hardware trigger mode**
  A/D conversion is stopped and the trigger standby state is restored again.

2. Be sure to clear bits 2 to 7 to “0”.

(4) A/D converter n mode register 2 (ADAnM2)

The ADAnM2 register is an 8-bit register that specifies the buffer mode and hardware trigger 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: ADA0M2 FFFFF203H, ADA1M2 FFFFF223H

	7	6	5	4	3	2	1	0
ADAnM2 (n = 0, 1)	ADAnBS	0	0	0	0	0	ADAnTMD1	0

ADAnBS	Buffer mode specification
0	1-buffer mode
1	4-buffer mode[Note]

ADAnTMD1	Hardware trigger mode specification
0	External trigger mode
1	Timer trigger mode

Note In 4-buffer mode, the A/D conversion results are stored in the order of ADAnCR0 → ADAnCR1 → ADAnCR2 → ADAnCR3 regardless of the selected analog input pin.
Setting of the 4-buffer mode is prohibited in the continuous scan mode and one-shot scan mode.

Cautions 1. The external triggers of A/D converters 0 and 1 are respectively input from the P04/INTP4/ADTRG0 and P05/INTP5/ADTRG1 pins. To use the external trigger mode, therefore, be sure to set the PMC0.PMC04 and PMC0.PMC05 bits to 1.

2. The timer trigger of A/D converter n is the A/D conversion start trigger signal (TQTADT1n) of the timer (motor control function). The TQTADT1n signal is connected to the TTRGn1 signal of A/D converter n (see Figure 11-2).

- **Timer trigger of A/D converter 0**
  In timer trigger mode: TQTADT10
- **Timer trigger of A/D converter 1**
  In timer trigger mode: TQTADT11

The TQTADT1n signal is set by using the TQ1AT00 to TQ1AT03 bits of TMQ1 option register 2 (TQ1OPT2) and the TQ1AT10 to TQ1AT13 bits of TMQ1 option register 3 (TQ1OPT3). The trigger sources of the motor control function that can be selected as an A/D conversion start trigger (timer trigger) are the INTTP1CC0, INTTP1CC1, INTTQ1CC0, and INTTQ1OV signals (two or more signals can be selected).

3. If the ADAnM2 register is written during A/D conversion (ADAnM0.ADAnEF bit = 1), the operation is performed as follows in each mode.

- **In software trigger mode**
  A/D conversion is stopped and executed again from the beginning.
- **In hardware trigger mode**
  A/D conversion is stopped and the trigger standby state is restored again.

(5) A/Dn conversion result registers 0 to 3, 0H to 3H (ADAnCR0 to ADAnCR3, ADAnCR0H to ADAnCR3H)

The ADAnCRm and ADAnCRH registers are registers that hold the A/D conversion results. Four 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 10 bits of the ADAnCRm 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 ADAnCRmH 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 ADAnCRm register is specified, and when the higher 8 bits are read, the ADAnCRmH register is specified.

Reset makes these registers undefined.

After reset: Undefined R Address: ADA0CR0 FFFFF210H, ADA0CR1 FFFFF212H, ADA0CR2 FFFFF214H, ADA0CR3 FFFFF216H, ADA1CR0 FFFFF230H, ADA1CR1 FFFFF232H, ADA1CR2 FFFFF234H, ADA1CR3 FFFFF236H

ADAnCRm (n = 0, 1) (m = 0 to 3)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADnm9	ADnm8	ADnm7	ADnm6	ADnm5	ADnm4	ADnm3	ADnm2	ADnm1	ADnm0	0	0	0	0	0	0

After reset: Undefined R Address: ADA0CR0H FFFFF211H, ADA0CR1H FFFFF213H, ADA0CR2H FFFFF215H, ADA0CR3H FFFFF217H, ADA1CR0H FFFFF231H, ADA1CR1H FFFFF233H, ADA1CR2H FFFFF235H, ADA1CR3H FFFFF237H

ADAnCRmH (n = 0, 1) (m = 0 to 3)

7	6	5	4	3	2	1	0
ADnm9	ADnm8	ADnm7	ADnm6	ADnm5	ADnm4	ADnm3	ADnm2

The correspondence between the analog input pins and the ADAnCRm and ADAnCRmH registers is shown below.

Table 11-3. Correspondence Between Analog Input Pins and ADAnCRm and ADAnCRmH Registers

A/D Converter	Analog Input Pin	A/D Conversion Result Register
A/D converter 0	ANI00	ADA0CR0, ADA0CR0H
	ANI01	ADA0CR1, ADA0CR1H
	ANI02	ADA0CR2, ADA0CR2H
	ANI03	ADA0CR3, ADA0CR3H
A/D converter 1	ANI10	ADA1CR0, ADA1CR0H
	ANI11	ADA1CR1, ADA1CR1H
	ANI12	ADA1CR2, ADA1CR2H
	ANI13	ADA1CR3, ADA1CR3H

The relationship between the analog voltage input to the analog input pin (ANInm) and the A/D conversion result (of A/Dn conversion result register m (ADAnCRm)) is as follows:

$$SAR = INT\left(\frac{V_{IN}}{AV_{REF}} \times 1{,}024 + 0.5\right)$$

$$ADCR^{Note} = SAR \times 64$$

or,

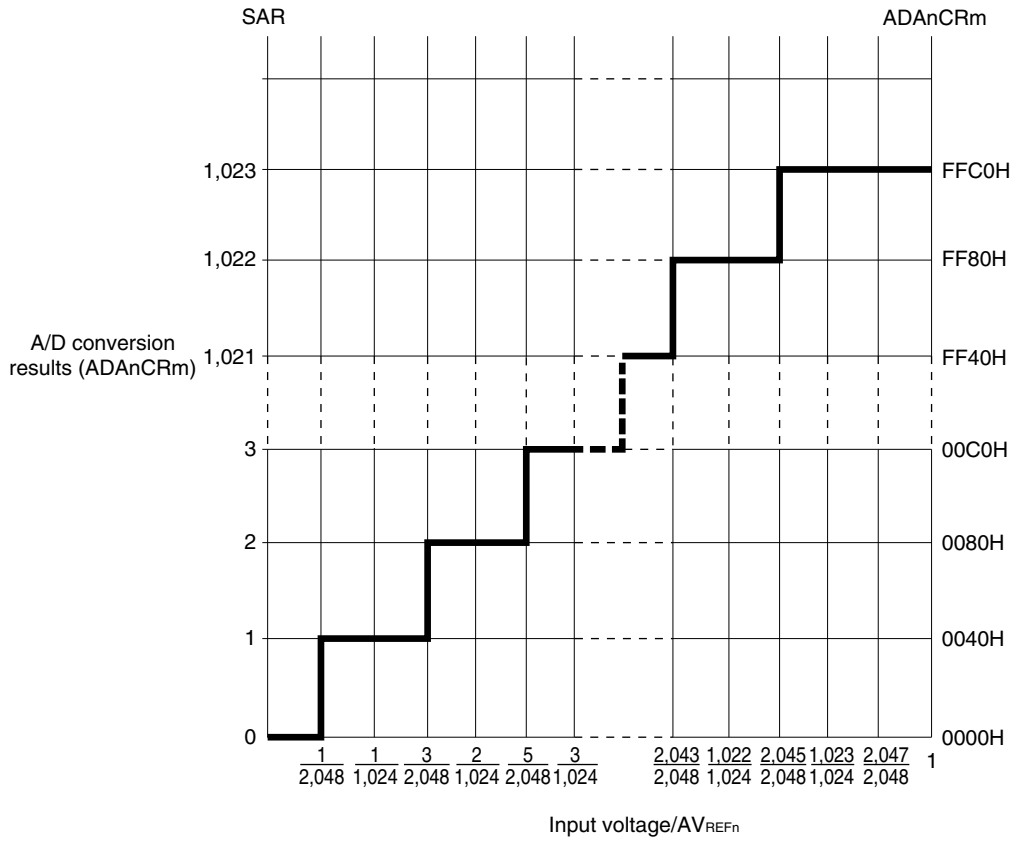
$$(SAR - 0.5) \times \frac{AV_{REF}}{1{,}024} \leq V_{IN} < (SAR + 0.5) \times \frac{AV_{REF}}{1{,}024}$$

INT( ): Function that returns the integer of the value in ( )
V_{IN}: Analog input voltage
AV_{REF}: AV_{REFn} pin voltage
ADCR: Value of A/Dn conversion result register m (ADAnCRm)

Note The lower 6 bits of the ADAnCRm register are fixed to 0.

The relationship between the analog input voltage and the A/D conversion results is shown below.

Figure 11-3. Relationship Between Analog Input Voltage and A/D Conversion Results



Remark n = 0, 1
m = 0 to 3

11.4 Operation

Caution A/D converters 0 and 1 are capable of simultaneous sampling of two circuits.

11.4.1 Basic operation

A/D conversion is executed by the following procedure.

(1) Select an analog input pin, operation mode, and trigger mode, by using the ADAnM0, ADAnM1, ADAnM2, and ADAnS registers[Note] (n = 0, 1). The setting of the number of stabilized clocks immediately after A/D conversion is enabled is determined by the specification of the ADAnM1.ADAnFR0 and ADAnM1.ADAnFR1 bits.

<R> **Note** If the ADAnM0, ADAnM2 and ADAnS registers are written during A/D conversion, or if a valid trigger is input, the conversion result is not correctly stored in the ADAnCRm register (m = 0 to 3) and the conversion operation before the change is initialized and performed from the beginning again.

(2) In the software trigger mode, setting the ADAnM0.ADAnCE bit to 1 starts A/D conversion after the lapse of the number of stabilized clocks (n = 0, 1). If the ADAnCE bit is set to 1 in the hardware trigger mode (external trigger mode, timer trigger mode), the A/D converter enters the trigger standby status. For details, see **11.3 (2) A/D converter n mode register 1 (ADAnM1)**.

(3) When A/D conversion is started, the voltage input to the selected analog input channel is sampled by the sample & hold circuit.

(4) 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.

(5) Set bit 9 of the successive approximation register (SAR). The tap selector changes the level of the voltage tap of the array to the reference voltage ($1/2AV_{REFn}$).

(6) 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 ($1/2AV_{REFn}$) 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 ($1/2AV_{REFn}$), the MSB of the SAR is reset.

(7) Next, bit 8 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 9, to which the result has been already set.

Bit 9 = 0: ($1/4AV_{REFn}$)
Bit 9 = 1: ($3/4AV_{REFn}$)

The voltage tap of the array and the analog input voltage are compared and bit 8 of the SAR is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of array: Bit 8 = 1
Analog input voltage ≤ Voltage tap of array: Bit 8 = 0

Comparison is continued like this to bit 0 of the SAR.

(8) When comparison of 10 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 (ADAnCRm) and the conversion result is stored in this register (n = 0, 1, m = 0 to 3). When A/D conversion has ended the specified number of times, an A/Dn conversion end interrupt request signal (INTADn) is generated.

11.4.2 Operation mode and trigger mode

Various conversion operations can be specified for A/D converters 0 and 1 by specifying the operation mode and trigger mode. The operation mode and trigger mode are set by the ADAnM0, ADAnM1, ADAnM2, and ADAnS registers.

The following shows the relationship between the operation mode and trigger mode.

Remark n = 0, 1

Trigger Mode		Operation Mode		Setting Value			
				ADAnM0	ADAnM1	ADAnM2	ADAnS
Software trigger		Continuous select	1 buffer	X000XX0XB	000000XXB	00000000B	00000XXXB
			4 buffers	X000XX0XB	000000XXB	10000000B	00000XXXB
		Continuous scan		X001XX0XB	000000XXB	00000000B	00000XXXB
		One-shot select	1 buffer	X010XX0XB	000000XXB	00000000B	00000XXXB
			4 buffers	X010XX0XB	000000XXB	10000000B	00000XXXB
		One-shot scan		X011XX0XB	000000XXB	00000000B	00000XXXB
Hardware trigger	External trigger	Continuous select	1 buffer[Note 1]	X000XX1XB	000000XXB	00000000B	00000XXXB
			4 buffers[Note 2]	X000XX1XB	000000XXB	10000000B	00000XXXB
		Continuous scan[Note 3]		X001XX1XB	000000XXB	00000000B	00000XXXB
		One-shot select	1 buffer[Note 1]	X010XX1XB	000000XXB	00000000B	00000XXXB
			4 buffers[Note 2]	X010XX1XB	000000XXB	10000000B	00000XXXB
		One-shot scan[Note 3]		X011XX1XB	000000XXB	00000000B	00000XXXB
	Timer trigger	Continuous select	1 buffer[Note 1]	X000XX1XB	000000XXB	00000010B	00000XXXB
			4 buffers[Note 2]	X000XX1XB	000000XXB	10000010B	00000XXXB
		Continuous scan[Note 3]		X001XX1XB	000000XXB	00000010B	00000XXXB
		One-shot select	1 buffer[Note 1]	X010XX1XB	000000XXB	00000010B	00000XXXB
			4 buffers[Note 2]	X010XX1XB	000000XXB	10000010B	00000XXXB
		One-shot scan[Note 3]		X011XX1XB	000000XXB	00000010B	00000XXXB

Notes 1. The same operation is performed regardless of the trigger type.

2. The same operation is performed regardless of the trigger type.

3. The same operation is performed regardless of the trigger type.

(1) Trigger mode

There are two types of trigger modes that serve as the start timing of an A/D conversion operation: software trigger mode and hardware trigger mode. There are two types of hardware trigger modes: external trigger mode and timer trigger mode.

These trigger modes are set by the ADAnM0 and ADAnM2 registers.

Remark n = 0, 1

(a) Software trigger mode

Of the ANIn0 to ANIn3 pins, the analog input pin specified by the ADAnS.ADAnS1 and ADAnS.ADAnS0 bits is used for the A/D conversion start timing when the ADAnM0.ADAnCE bit is set to 1 in this mode.

After end of A/D conversion, the conversion result is stored in A/Dn conversion result register m (ADAnCRm) (m = 0 to 3). At the same time, an A/Dn conversion end interrupt request signal (INTADn) is generated.

If the operation mode set by the ADAnM0.ADAnMD1 and ADAnM0.ADAnMD0 bits is the continuous select mode or continuous scan mode, the conversion operation is repeated unless the ADAnM0.ADAnCE bit is cleared to 0. In the one-shot select mode or one-shot scan mode, the conversion operation is stopped after A/D conversion ends.

When conversion is started, the ADAnM0.ADAnEF bit is set to 1 (conversion in progress).

If the ADAnM0, ADAnM2, and ADAnS registers are written during A/D conversion, the conversion is stopped and executed again from the beginning.

Remark n = 0, 1
m = 0 to 3

(b) Timer trigger mode

Of the ANIn0 to ANIn3 pins, the analog input pin specified by the ADAnS.ADAnS1 and ADAnS.ADAnS0 bits is used for A/D conversion in this mode. The timer (motor control function) is used for the A/D conversion start timing.

The timer trigger signal of A/D converter n is the timer interrupt request signal (TQTADT10, TQTADT11) of the timer (motor control function). The TQTADT10 and TQTADT11 signals are connected to the TTRG01 and TTRG11 signals of A/D converter n (see **Figure 11-2**).

- Timer trigger of A/D converter 0: TQTADT10
- Timer trigger of A/D converter 1: TQTADT11

The TQTADT10 and TQTADT11 signals are set by using the TQ1AT00 to TQ1AT03 bits of TMQ1 option register 2 (TQ1OPT2) and the TQ1AT10 to TQ1AT13 bits of TMQ1 option register 3 (TQ1OPT3). The interrupt request signals of the motor control function that can be selected as a timer trigger signal are the INTTP1CC0, INTTP1CC1, INTTQ1CC0, and INTTQ1OV signals (two or more signals can be selected).

When the ADAnM2.ADAnTMD1 bit is set to 1, A/D conversion is started at the rising edge of the timer interrupt request signal (TQTADT10 or TQTADT11) set for the motor control function.

When the ADAnM0.ADAnCE bit is set to 1, the A/D converter waits for a trigger and, when the timer interrupt request signal is input, starts A/D conversion.

After the end of A/D conversion, the conversion result is stored in A/Dn conversion result register m (ADAnCRm) and, at the same time, an A/Dn conversion end interrupt request signal (INTADn) is generated.

If the operation mode set by the ADAnM0.ADAnMD1 and ADAnM0.ADAnMD0 bits is the continuous select mode or continuous scan mode, the conversion operation is repeated, with the next timer interrupt request signal as the trigger, unless the ADAnM0.ADAnCE bit is cleared to 0. In the one-shot select mode or one-shot scan mode, the A/D converter waits for a trigger.

When conversion is started, the ADAnM0.ADAnEF bit is set to 1 (conversion in progress). While the converter waits for a trigger, however, the ADAnM0.ADAnEF bit = 0 (conversion stopped).

If the valid trigger is input during A/D conversion, the conversion is stopped and is executed again from the beginning. If the ADAnM0, ADAnM2, and ADAnS registers are written during A/D conversion, the conversion is stopped and the converter waits for a trigger again.

Caution In timer trigger mode, make sure that the timer interrupt request signal (A/D conversions start timing) is not generated at an interval shorter than the minimum number of conversion clocks that can be specified by the ADAnM1.ADAnFR1 and ADAnM1.ADAnFR0 bits. If the interrupt request signal is generated at an interval shorter than the minimum number of conversion clocks, the last trigger is valid.

Remark n = 0, 1
m = 0 to 3

(c) External trigger mode

Of the ANIn0 to ANIn3 pins, the analog input pin specified by the ADAnS.ADAnS1 and ADAnS.ADAnS0 bits is used for A/D conversion in this mode. The ADTRGn pin is used for the A/D conversion start timing.

The ADTRG0 pin alternates as the P04/INTP4 pin, and the ADTRG1 pin as the P05/INTP5 pin. To set the external trigger mode, set the PMC04 and PMC05 bits of port mode control register 0 (PMC0) to 1, and the ADAnM2.ADAnTMD1 bit to 0.

The valid edge of the external input signal in the external trigger mode can be selected from the rising edge, falling edge, or both the rising and falling edges, according to the setting of the ADAnM0.ADAnETS1 and ADAnM0.ADAnETS0 bits.

When the ADAnM0.ADAnCE bit is set to 1, the A/D converter waits for a trigger and starts conversion when the trigger is input from the ADTRGn pin.

After the end of conversion, the conversion result is stored in A/Dn conversion result register m (ADAnCRm) and, at the same time, an A/Dn conversion end interrupt request signal (INTADn) is generated.

If the operation mode set by the ADAnM0.ADAnMD1 and ADAnM0.ADAnMD0 bits is the continuous select mode or continuous scan mode, the conversion operation is repeated, with the next ADTRGn signal as the trigger, unless the ADAnM0.ADAnCE bit is cleared to 0. In the one-shot select mode or one-shot scan mode, the A/D converter waits for a trigger.

When conversion is started, the ADAnM0.ADAnEF bit is set to 1 (conversion in progress). While the converter waits for a trigger, however, the ADAnEF bit = 0 (conversion stopped).

If the valid trigger is input during A/D conversion, the conversion is stopped and is executed again from the beginning. If the ADAnM0, ADAnM2, and ADAnS registers are written during A/D conversion, the conversion is stopped and the converter waits for a trigger again.

Caution In the external trigger mode, make sure that the ADTRGn signal (A/D conversion start timing) is not generated at an interval shorter than the minimum number of conversion clocks that can be specified by the ADAnM1.ADAnFR1 and ADAnM1.ADAnFR0 bits. If the ADTRGn signal is generated at an interval shorter than the minimum number of conversion clocks, the last trigger is valid.

Remark n = 0, 1
m = 0 to 3

(2) Operation mode

There are four operation modes in which the ANIn0 to ANIn3 pins are set: continuous select mode, continuous scan mode, one-shot select mode, and one-shot scan mode. The continuous select mode and one-shot select mode have sub-modes that consist of 1-buffer mode and 4-buffer mode. These modes are set by the ADAnM0 and ADAnM2 registers.

Remark n = 0, 1

(a) Continuous select mode

In this mode, the analog input pin (ANInm) specified by the ADAnS register is A/D converted continuously. The conversion results are stored in the A/Dn conversion result register (ADAnCRm) corresponding to the ANInm pin. In this mode, the 1-buffer mode and 4-buffer mode are provided for storing the A/D conversion results.

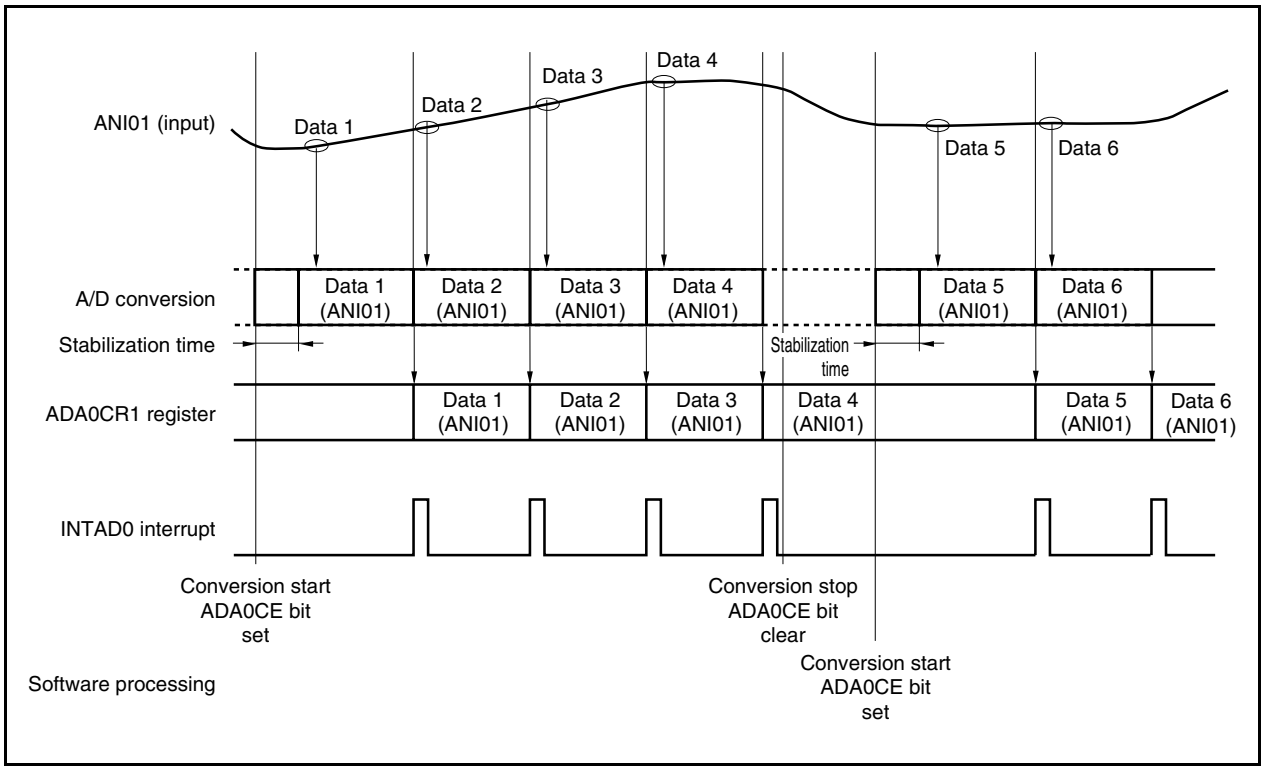
- **1-buffer mode**

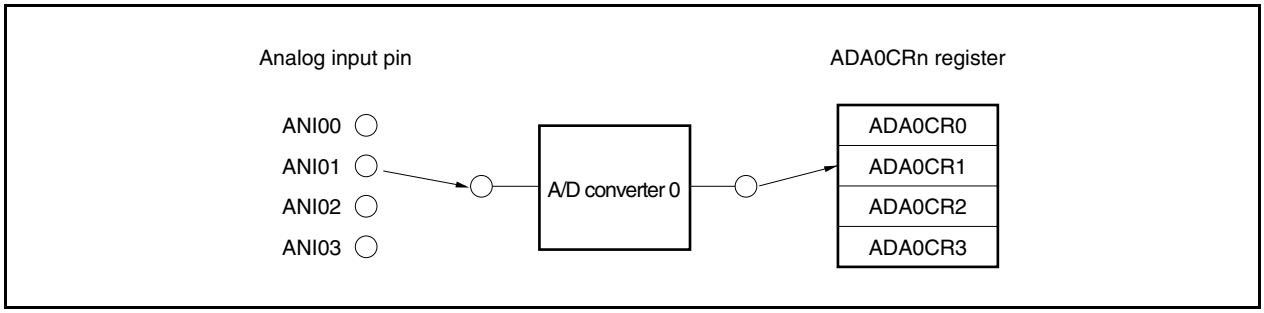
  In this mode, the voltage of the analog input pin (ANInm) specified by the ADAnS register is A/D converted. The conversion results are stored in the ADAnCRm register corresponding to the ANInm pin. The ANInm pin and the ADAnCRm register correspond one to one, and an A/Dn conversion end interrupt request signal (INTADn) is generated each time one A/D conversion ends.

  After the end of A/D conversion, the conversion is repeated again unless the ADAnM0.ADAnCE bit is cleared to 0.

  Remark n = 0, 1
  m= 0 to 3

Figure 11-4. Continuous Select 1-Buffer Mode Operation Timing (When ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 Bits = 00, ADA0M2.ADA0BS Bit = 0, ADA0S.ADA0S1 and ADA0S.ADA0S0 Bits = 01)



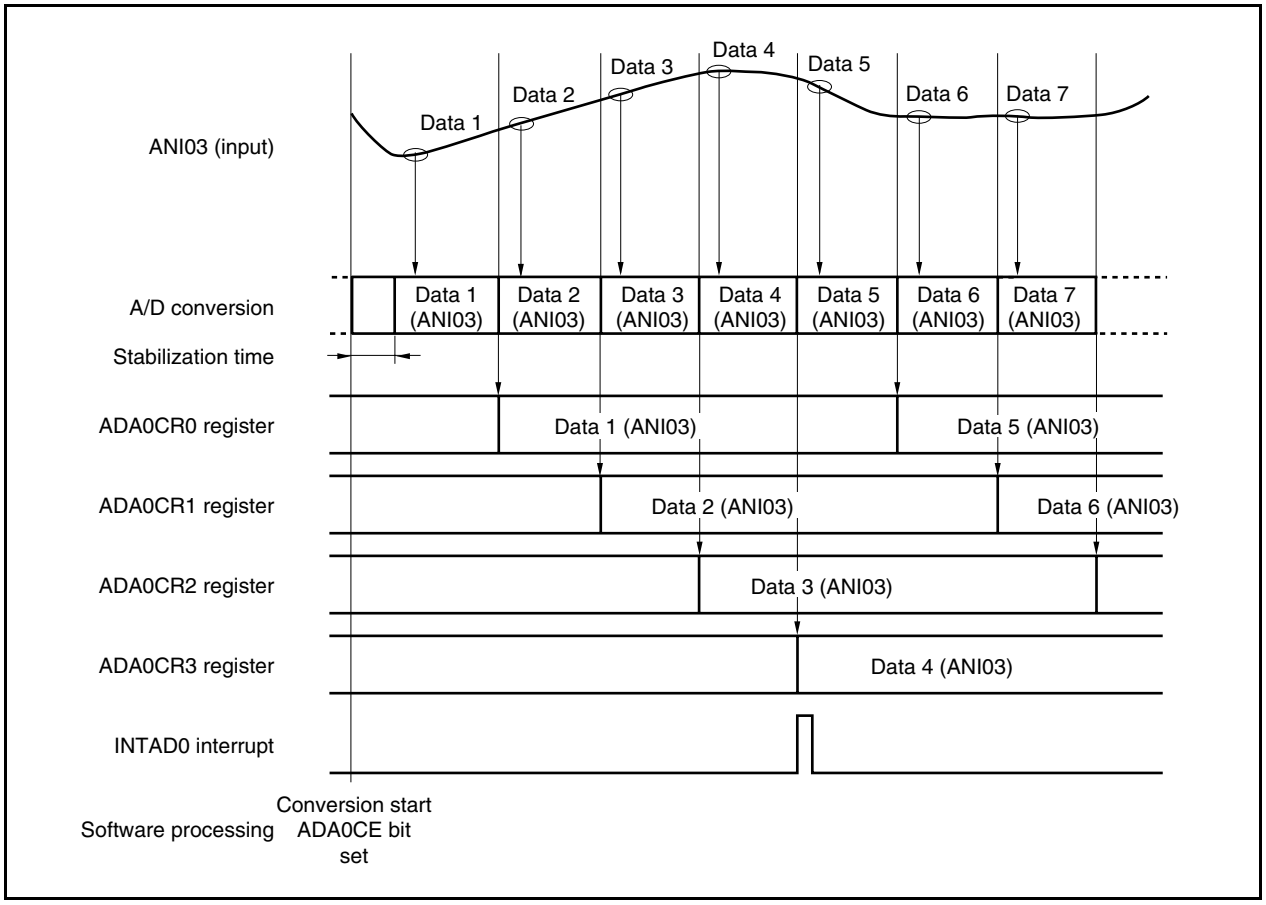


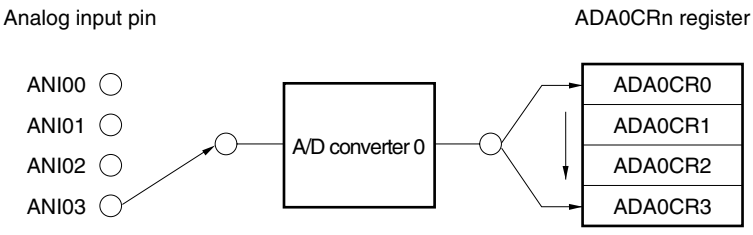
- **4-buffer mode**

  In this mode, the voltage of one analog input pin (ANInm) is A/D converted four times and the results are stored in the ADAnCRm register. The A/Dn conversion end interrupt request signal (INTADn) is generated when the four A/D conversions end. After end of A/D conversion, the conversion is started again from the beginning, unless the ADAnM0.ADAnCE bit is cleared to 0.

Remark n = 0, 1, m = 0 to 3

Figure 11-5. Continuous Select 4-Buffer Mode Operation Timing (When ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 Bits = 00, ADA0M2.ADA0BS Bit = 1, ADA0S.ADA0S1 and ADA0S.ADA0S0 Bits = 11)





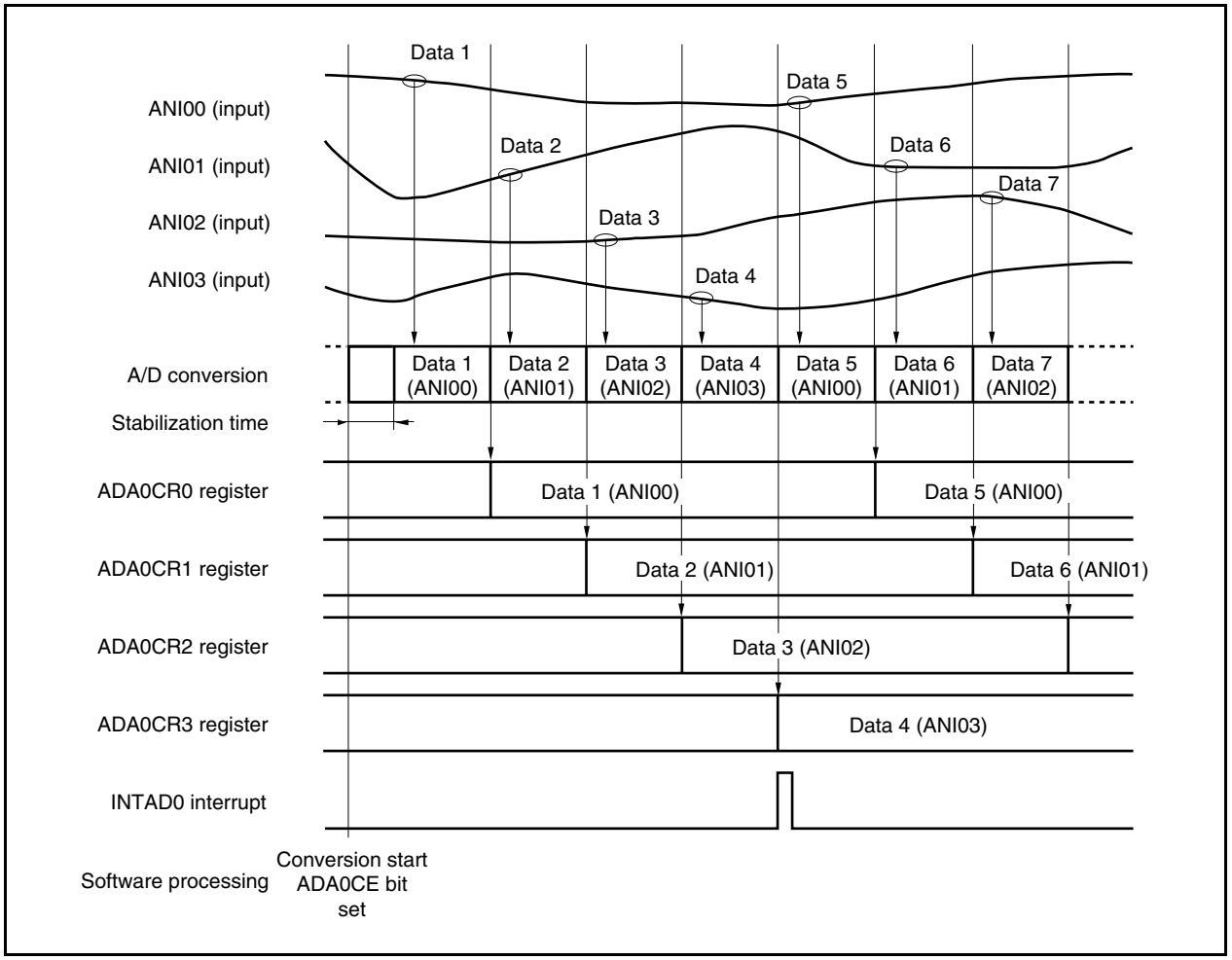
Caution The A/D conversion results are stored in the order of ADA0CR0 → ADA0CR1 → ADA0CR2 → ADA0CR3 regardless of the selected analog input pin.

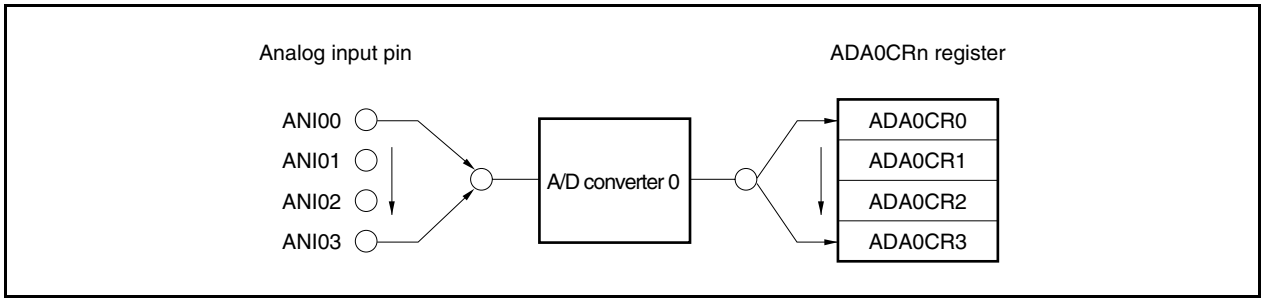
(b) Continuous scan mode

In this mode, the analog input pins (ANInm) specified by the ADAnS register are selected sequentially from the ANIn0 pin, and A/D conversion is executed continuously. The A/D conversion results are stored in the ADAnCRm register corresponding to the analog input pin. When conversion of all the specified analog input pin ends, the A/Dn conversion end interrupt request signal (INTADn) is generated. After the end of A/D conversion, the conversion is started again from the ANIn0 pin, unless the ADAnM0.ADAnCE bit is cleared to 0.

Remark n = 0, 1, m = 0 to 3

Figure 11-6. Continuous Scan Mode Operation Timing
(When ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 Bits = 01, ADA0S.ADA0S1 and ADA0S.ADA0S0 Bits = 11)





(c) One-shot select mode

In this mode, the analog input pin (ANInm) specified by the ADAnS register is A/D converted once. The conversion results are stored in the A/Dn conversion result register (ADAnCRm) corresponding to the ANInm pin. In this mode, the 1-buffer mode and 4-buffer mode are provided for storing the A/D conversion results.

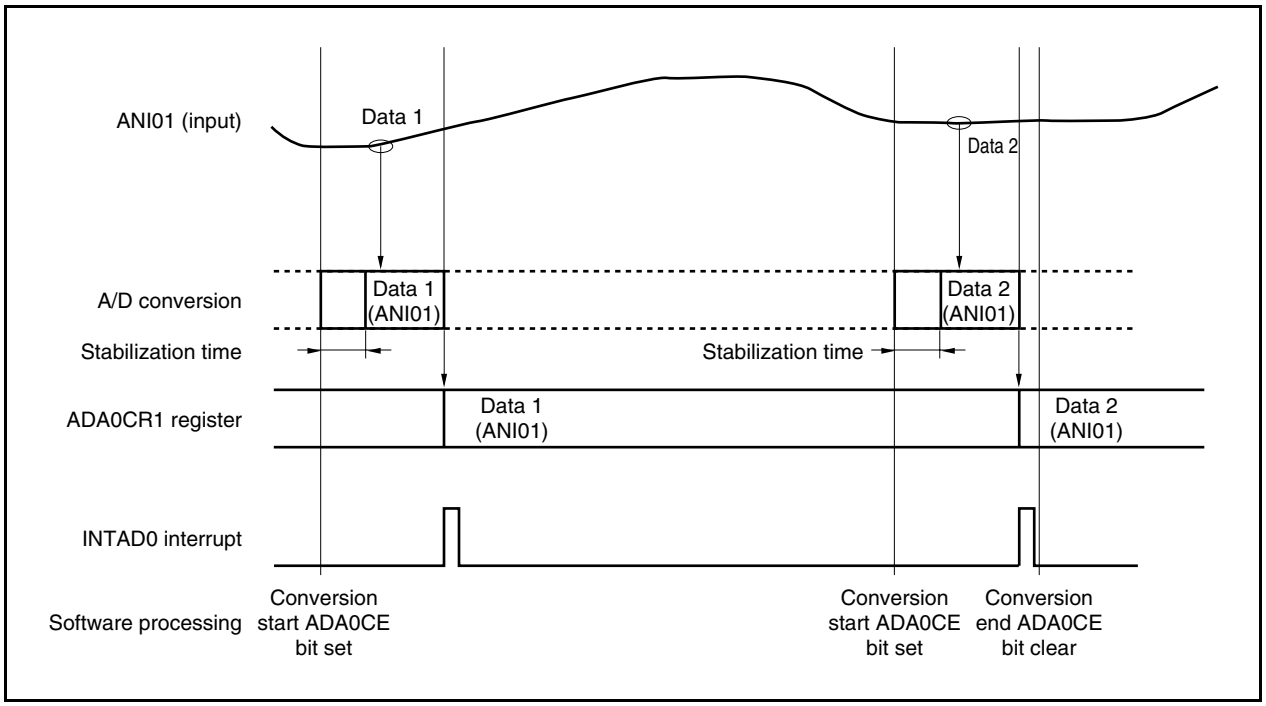
- **1-buffer mode**

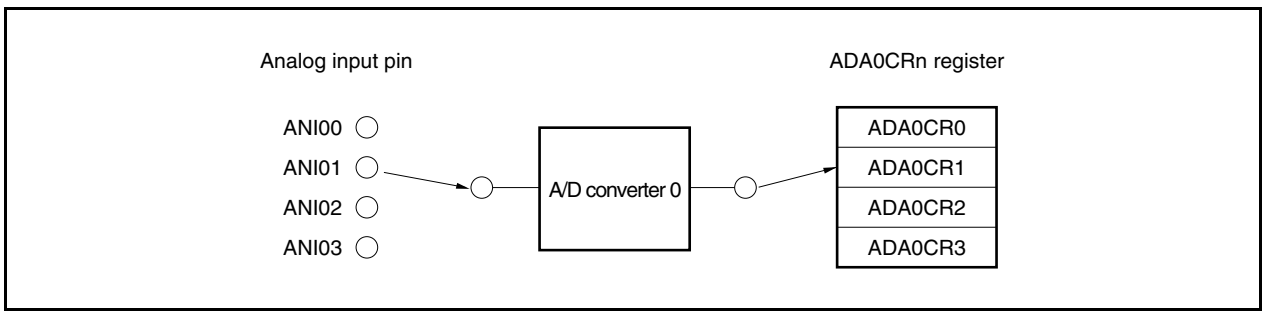
  In this mode, the voltage of the analog input pin (ANInm) specified by the ADAnS register is A/D converted. The conversion results are stored in the ADAnCRm register corresponding to the ANInm pin. The ANInm pin and the ADAnCRm register correspond one to one, and an A/Dn conversion end interrupt request signal (INTADn) is generated each time one A/D conversion ends.

  After the end of A/D conversion, the conversion operation is stopped.

Remark n = 0, 1, m = 0 to 3

Figure 11-7. One-Shot Select 1-Buffer Mode Operation Timing (When ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 Bits = 10, ADA0M2.ADA0BS Bit = 0, ADA0S.ADA0S1 and ADA0S.ADA0S0 Bits = 01)



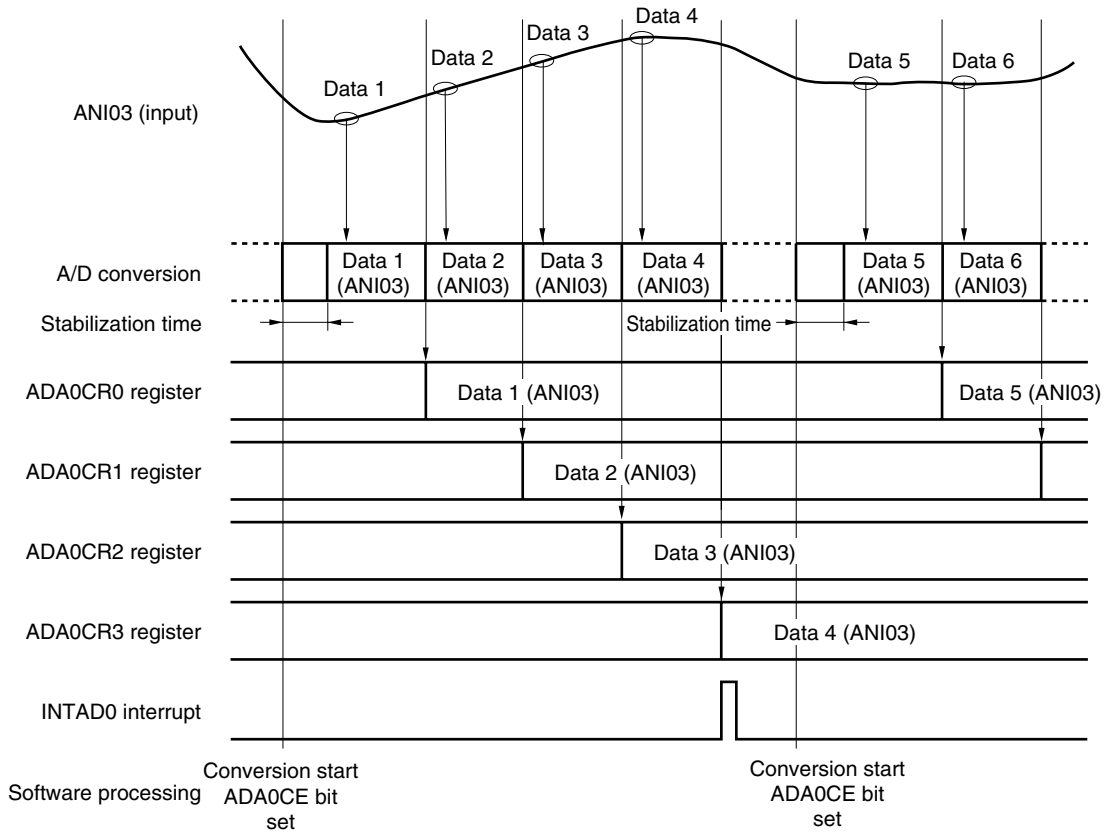


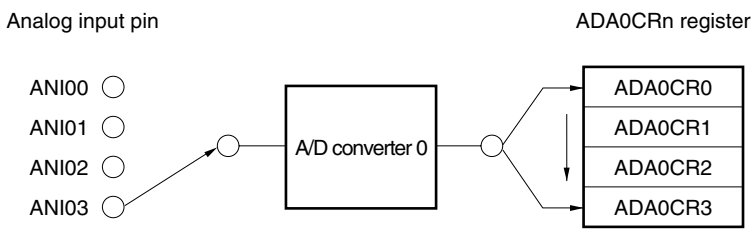
- **4-buffer mode**

  In this mode the voltage of one analog input pin (ANInm) is A/D converted four times and the results are stored in the ADAnCRm register. The A/Dn conversion end interrupt request signal (INTADn) is generated when the four A/D conversions end. After end of A/D conversion, the conversion operation is stopped.

Remark n = 0, 1, m = 0 to 3

Figure 11-8. One-Shot Select 4-Buffer Mode Operation Timing (When ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 Bits = 10, ADA0M2.ADA0BS Bit = 1, ADA0S.ADA0S1 and ADA0S.ADA0S0 Bits = 11)





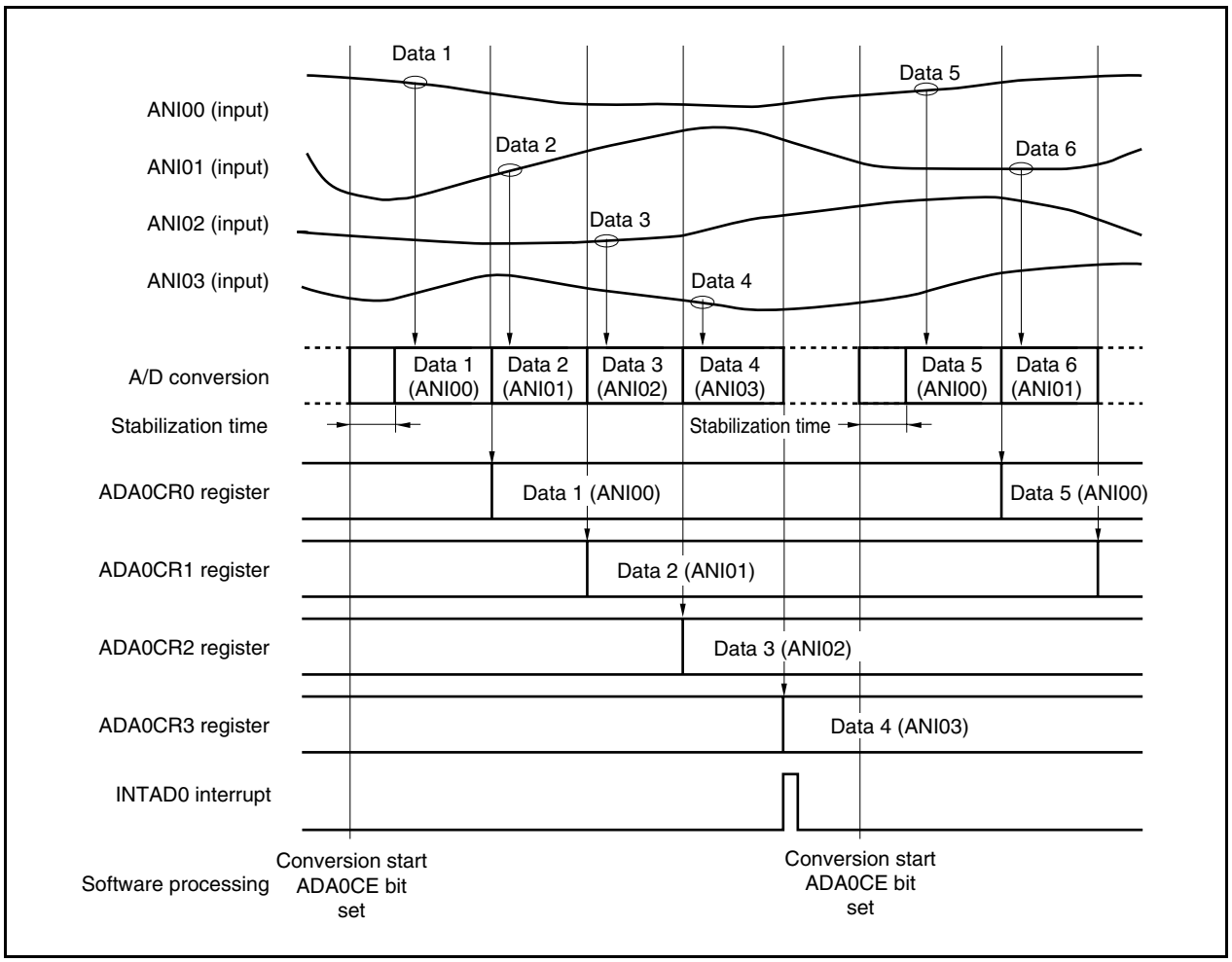
Caution The A/D conversion results are stored in the order of ADA0CR0 → ADA0CR1 → ADA0CR2 → ADA0CR3 regardless of the selected analog input pin.

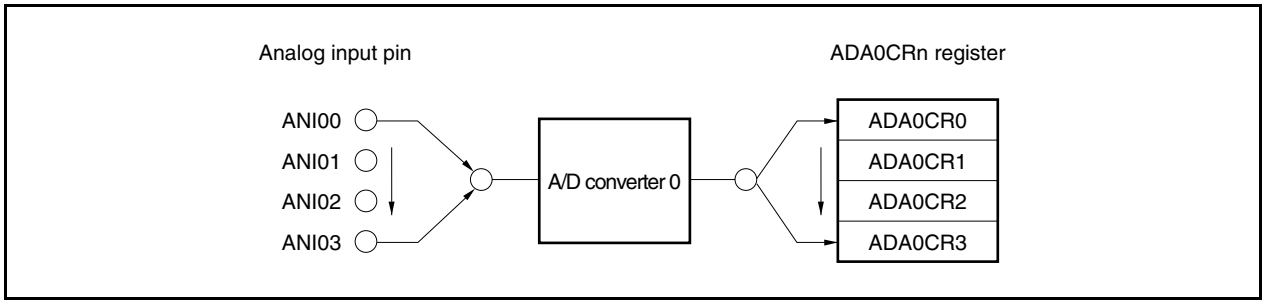
(d) One-shot scan mode

In this mode, the analog input pins (ANInm) specified by the ADAnS register are selected sequentially from the ANIn0 pin, and A/D conversion is executed. The A/D conversion results are stored in the ADAnCRm register corresponding to the analog input pin. When conversion of all the specified analog input pin ends, the A/Dn conversion end interrupt request signal (INTADn) is generated. After end of A/D conversion, the conversion operation is stopped.

Remark n = 0, 1
m = 0 to 3

Figure 11-9. One-Shot Scan Mode Operation Timing
(When ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 Bits = 11, ADA0S.ADA0S1 and ADA0S.ADA0S0 Bits = 11)





11.5 Operation in Software Trigger Mode

When the ADAnM0.ADAnCE bit is set (1), A/D conversion is started.

When A/D conversion is started, the ADAnM0.ADAnEF bit = 1 (conversion in progress).

If the ADAnM0, ADAnM2, and ADAnS registers are written during A/D conversion, the conversion is stopped and executed again from the beginning.

Remark n = 0, 1

11.5.1 Continuous select mode operations

In this mode, the analog input pin (ANInm) specified by the ADAnS register is A/D converted continuously. The conversion results are stored in the ADAnCRm register. In the continuous select mode, the 1-buffer mode and 4-buffer mode are supported according to the method of storing the A/D conversion results.

Remark n = 0, 1
m = 0 to 3

(1) 1-buffer mode (software trigger continuous select: 1 buffer)

In this mode, the voltage of one analog input pin (ANInm) is A/D converted once. The conversion results are stored in one ADAnCRm register. The ANInm pin and ADAnCRm register correspond one to one.

Each time an A/D conversion is executed, an A/Dn conversion end interrupt request signal (INTADn) is generated and A/D conversion ends. After the end of A/D conversion, the conversion is repeated again unless the ADAnM0.ADAnCE bit is cleared to 0.

It is not necessary to set (1) the ADAnM0.ADAnCE bit to restart A/D conversion[Note].

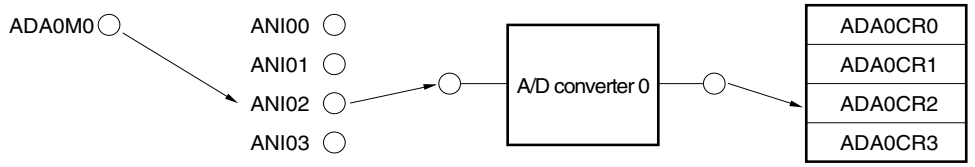
Note In the software trigger continuous select 1-buffer mode, the A/D conversion operation is not stopped unless the ADAnM0.ADAnCE bit is cleared to 0. If the ADAnCRm register is not read before the next A/D conversion ends, it is overwritten.

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
ANInm	ADAnCRm

Remark n = 0, 1
m = 0 to 3

Figure 11-10. Example of 1-Buffer Mode Operation (Software Trigger Continuous Select: 1 Buffer)



(1) The ADA0CE bit = 1 (enable)
(2) The ANI02 pin is A/D converted
(3) The conversion result is stored in the ADA0CR2 register
(4) The ADA0M0.ADA0EF bit = 0
(5) The INTAD0 interrupt request signal is generated
(6) Return to (2)
(7) To end the conversion, the ADA0CE bit = 0 (stop)

Remark This is an operation example with the following setting.
ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits = 00, ADA0M0.ADA0TMD bit = 0,
ADA0M2.ADA0BS bit = 0, ADA0S.ADA0S1 and ADA0S.ADA0S0 bits = 10

(2) 4-buffer mode (software trigger continuous select: 4 buffers)

In this mode, the voltage of one analog input pin (ANInm) is A/D converted four times and the results are stored in the ADAnCRm register.

When the 4th A/D conversion ends, an A/Dn conversion end interrupt request signal (INTADn) is generated. After the end of A/D conversion, the conversion is started again from the beginning, unless the ADAnM0.ADAnCE bit is cleared to 0. It is not necessary to set (1) the ADAnM0.ADAnCE bit to restart A/D conversion[Note].

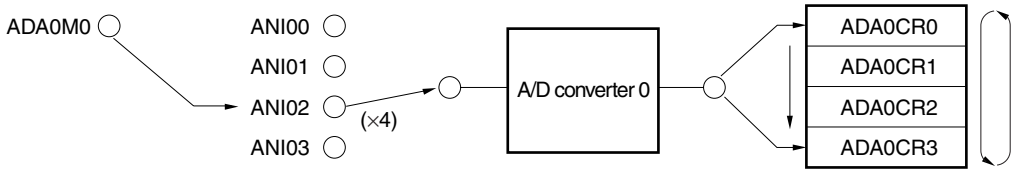
Note In the software trigger continuous select 4-buffer mode, the A/D conversion operation is not stopped unless the ADAnM0.ADAnCE bit is cleared to 0. If the ADAnCRm register is not read before the next A/D conversion ends, it is overwritten.

This mode is suitable for applications in which the average of the A/D conversion results of one analog input pin is calculated.

Analog Input Pin	A/D Conversion Result Register
ANInm	ADAnCR0
ANInm	ADAnCR1
ANInm	ADAnCR2
ANInm	ADAnCR3

Remark n = 0, 1
m = 0 to 3

Figure 11-11. Example of 4-Buffer Mode Operation (Software Trigger Continuous Select: 4 Buffers)



(1) The ADA0CE bit = 1 (enable)
(2) The ANI02 pin is A/D converted
(3) The conversion result is stored in the ADA0CR0 register
(4) The ANI02 pin is A/D converted
(5) The conversion result is stored in the ADA0CR1 register
(6) The ANI02 pin is A/D converted
(7) The conversion result is stored in the ADA0CR2 register
(8) The ANI02 pin is A/D converted
(9) The conversion result is stored in the ADA0CR3 register
(10) The INTAD0 interrupt request signal is generated
(11) Return to (2)
(12) To end the conversion, the ADA0CE bit = 0 (stop)

Remark This is an operation example with the following setting.
ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits = 00, ADA0M0.ADA0TMD bit = 0,
ADA0M2.ADA0BS bit = 1, ADA0S.ADA0S1 and ADA0S.ADA0S0 bits = 10

11.5.2 Continuous scan mode operations

In this mode, the analog input pins (ANInm) specified by the ADAnS register are selected sequentially from the ANIn0 pin, and A/D conversion is executed continuously. The A/D conversion results are stored in the ADAnCRm register corresponding to the analog input pin.

When conversion of all the specified analog input pins ends, the A/Dn conversion end interrupt request signal (INTADn) is generated. After the end of A/D conversion, the conversion is started again from the ANIn0 pin, unless the ADAnM0.ADAnCE bit is cleared to 0. It is not necessary to set (1) the ADAnM0.ADAnCE bit to restart A/D conversion[Note].

In the continuous scan mode, only the 1-buffer mode is supported.

Note In the software trigger continuous scan mode, the A/D conversion operation is not stopped unless the ADAnM0.ADAnCE bit is cleared to 0. If the ADAnCRm register is not read before the next A/D conversion ends, it is overwritten.

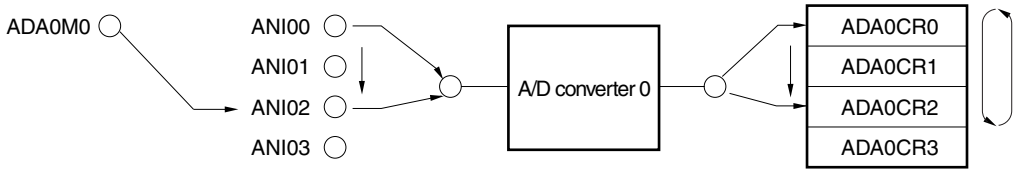
This mode is suitable for applications in which multiple analog inputs are constantly monitored.

Analog Input Pin	A/D Conversion Result Register
ANIn0	ADAnCR0
⋮	⋮
ANInm[Note]	ADAnCRm

Note Set by the ADAnS.ADAnS0 and ADAnS.ADAnS1 bits.

Remark n = 0, 1
m = 0 to 3

Figure 11-12. Example of Continuous Scan Mode Operation (Software Trigger Continuous Scan)



(1) The ADA0CE bit = 1 (enable)
(2) The ANI00 pin is A/D converted
(3) The conversion result is stored in the ADA0CR0 register
(4) The ANI01 pin is A/D converted
(5) The conversion result is stored in the ADA0CR1 register
(6) The ANI02 pin is A/D converted
(7) The conversion result is stored in the ADA0CR2 register
(8) The INTAD0 interrupt request signal is generated
(9) Return to (2)
(10) To end the conversion, the ADA0CE bit = 0 (stop)

Remark This is an operation example with the following setting.
ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits = 01, ADA0M0.ADA0TMD bit = 0,
ADA0M2.ADA0BS bit = 0, ADA0S.ADA0S1 and ADA0S.ADA0S0 bits = 10

11.5.3 One-shot select mode operations

In this mode, the analog input pin (ANInm) specified by the ADAnS register is A/D converted continuously. The conversion results are stored in the ADAnCRm register. In the one-shot select mode, the 1-buffer mode and 4-buffer mode are supported according to the method of storing the A/D conversion results.

Remark n = 0, 1
m = 0 to 3

(1) 1-buffer mode (software trigger one-shot select: 1 buffer)

In this mode, the voltage of one analog input pin (ANInm) is A/D converted once. The conversion results are stored in one ADAnCRm register. The ANInm pin and the ADAnCRm register correspond one to one.

Each time an A/D conversion is executed, an A/Dn conversion end interrupt request signal (INTADn) is generated and A/D conversion ends. After the end of A/D conversion, the conversion operation is stopped.

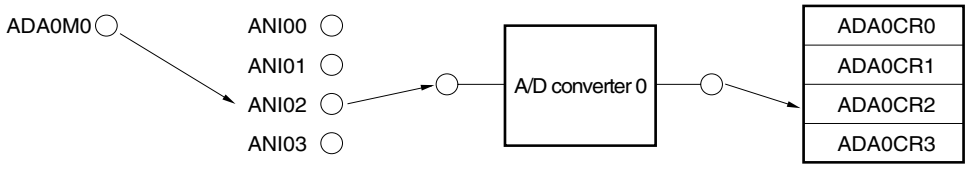
If the ADAnM0.ADAnCE bit is set (1), A/D conversion can be restarted.

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
ANInm	ADAnCRm

Remark n = 0, 1
m = 0 to 3

Figure 11-13. Example of 1-Buffer Mode Operation (Software Trigger One-Shot Select: 1 Buffer)



(1) The ADA0CE bit = 1 (enable)
(2) The ANI02 pin is A/D converted
(3) The conversion result is stored in the ADA0CR2 register
(4) The ADA0M0.ADA0EF bit = 0
(5) The INTAD0 interrupt request signal is generated

Remark This is an operation example with the following setting.
ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits = 10, ADA0M0.ADA0TMD bit = 0,
ADA0M2.ADA0BS bit = 0, ADA0S.ADA0S1 and ADA0S.ADA0S0 bits = 10

(2) 4-buffer mode (software trigger one-shot select: 4 buffers)

In this mode, the voltage of one analog input pin (ANInm) is A/D converted four times and the results are stored in the ADAnCRm register.

When the 4th A/D conversion ends, an A/Dn conversion end interrupt request signal (INTADn) is generated. After the end of A/D conversion, the conversion operation is stopped.

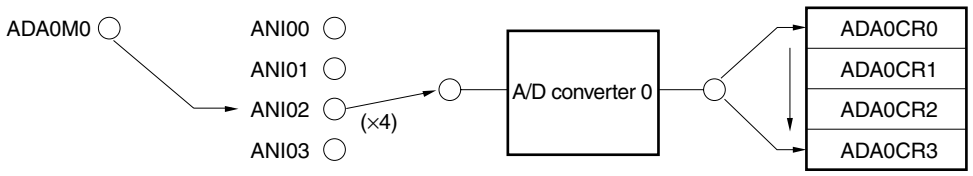
If the ADAnM0.ADAnCE bit is set (1), A/D conversion can be restarted.

This mode is suitable for applications in which the average of the A/D conversion results is calculated.

Analog Input Pin	A/D Conversion Result Register
ANInm	ADAnCR0
ANInm	ADAnCR1
ANInm	ADAnCR2
ANInm	ADAnCR3

Remark n = 0, 1
m = 0 to 3

Figure 11-14. Example of 4-Buffer Mode Operation (Software Trigger One-Shot Select: 4 Buffers)



(1) The ADA0CE bit = 1 (enable)
(2) The ANI02 pin is A/D converted
(3) The conversion result is stored in the ADA0CR0 register
(4) The ANI02 pin is A/D converted
(5) The conversion result is stored in the ADA0CR1 register
(6) The ANI02 pin is A/D converted
(7) The conversion result is stored in the ADA0CR2 register
(8) The ANI02 pin is A/D converted
(9) The conversion result is stored in the ADA0CR3 register
(10) The ADA0M0.ADA0EF bit = 0
(11) The INTAD0 interrupt request signal is generated

Remark This is an operation example with the following setting.
ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits = 10, ADA0M0.ADA0TMD bit = 0,
ADA0M2.ADA0BS bit = 1, ADA0S.ADA0S1 and ADA0S.ADA0S0 bits = 10

11.5.4 One-shot scan mode operations

In this mode, the analog input pins (ANInm) specified by the ADAnS register are selected sequentially from the ANIn0 pin, and A/D conversion is executed continuously. The A/D conversion results are stored in the ADAnCRm register corresponding to the analog input pin.

When conversion of all the specified analog input pin ends, the A/Dn conversion end interrupt request signal (INTADn) is generated. After the end of A/D conversion, the conversion operation is stopped.

If the ADAnM0.ADAnCE bit is set (1), A/D conversion can be restarted.

In the one-shot scan mode, only the 1-buffer mode is supported.

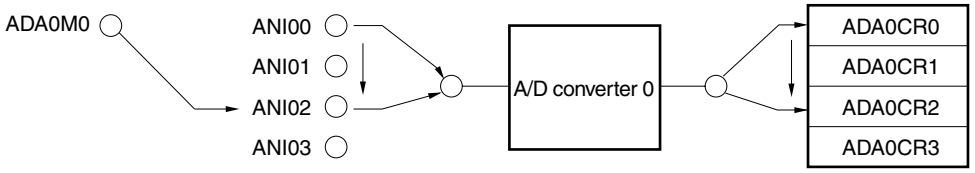
This mode is suitable for applications in which multiple analog inputs are constantly monitored.

Analog Input Pin	A/D Conversion Result Register
ANIn0	ADAnCR0
⋮	⋮
ANInm[Note]	ADAnCRm

Note Set by the ADAnS.ADAnS0 and ADAnS.ADAnS1 bits.

Remark n = 0, 1
m = 0 to 3

Figure 11-15. Example of One-Shot Scan Mode Operation (Software Trigger One-Shot Scan)



(1) The ADA0CE bit = 1 (enable)
(2) The ANI00 pin is A/D converted
(3) The conversion result is stored in the ADA0CR0 register
(4) The ANI01 pin is A/D converted
(5) The conversion result is stored in the ADA0CR1 register
(6) The ANI02 pin is A/D converted
(7) The conversion result is stored in the ADA0CR2 register
(8) The ADA0M0.ADA0EF bit = 0
(9) The INTAD0 interrupt request signal is generated

Remark This is an operation example with the following setting.
ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits = 11, ADA0M0.ADA0TMD bit = 0,
ADA0M2.ADA0BS bit = 0, ADA0S.ADA0S1 and ADA0S.ADA0S0 bits = 10

11.6 Operation in Timer Trigger Mode

With A/D converter n, the conversion timing is specified by using the A/D conversion start trigger signal (TQTADT1n) from the timer (motor control function) (see **Figure 11-2**).

- Timer trigger of A/D converter 0: TQTADT10
- Timer trigger of A/D converter 1: TQTADT11

The TQTADT1n signal is set by using the TQ1AT00 to TQ1AT03 bits of TMQ1 option register 2 (TQ1OPT2) and the TQ1AT10 to TQ1AT13 bits of TMQ1 option register 3 (TQ1OPT3). The trigger sources of the motor control function that can be selected as the A/D conversion start trigger, which is a timer trigger, are the INTTP1CC0, INTTP1CC1, INTTQ1CC0, and INTTQ1OV signals (two or more signals can be selected).

When the ADAnM2.ADAnTMD1 bit is set 1, A/D conversion is started at the rising edge of the A/D conversion start trigger signal (TQTADT1n) selected for the motor control function.

When the ADAnM0.ADAnCE bit is set to 1, the A/D converter waits for a trigger and, when the A/D conversion start trigger signal is input, it starts A/D conversion.

After the end of A/D conversion, the conversion result is stored in A/Dn conversion result register m (ADAnCRm) and, at the same time, the A/Dn conversion end interrupt request signal (INTADn) is generated.

After the end of A/D conversion, the A/D converter waits for a trigger regardless of the operation mode set by the ADAnM0.ADAnMD1 and ADAnM0.ADAnMD0 bits.

When conversion is started, the ADAnM0.ADAnEF bit is set to 1 (conversion in progress). However, while the A/D converter is waiting for a trigger, the ADAnM0.ADAnEF bit = 0 (conversion stopped).

If a valid trigger is input during A/D conversion, the conversion operation is stopped and started again from the beginning. If the ADAnM0, ADAnM2, and ADAnS registers are written during A/D conversion, the conversion is stopped and the A/D converter waits for a trigger again.

Caution In timer trigger mode, make sure that the A/D conversion start trigger signal (A/D conversions start timing) is not generated at an interval shorter than the minimum number of conversion clocks that can be specified by the ADAnM1.ADAnFR1 and ADAnM1.ADAnFR0 bits. If the A/D conversion start trigger signal is generated at an interval shorter than the minimum number of conversion clocks, the last trigger is valid.

Remark n = 0, 1
m = 0 to 3

11.6.1 Continuous select mode/one-shot select mode operations

In this mode, the voltage of the analog input pin (ANInm) specified by the ADAnS register is A/D converted. The conversion results are stored in the ADAnCRm register. In the continuous select mode or one-shot select mode, the 1-buffer mode and 4-buffer mode are supported according to the method of storing the A/D conversion results.

Remark n = 0, 1
m = 0 to 3

(1) 1-buffer mode operation (1 buffer of continuous select/one-shot select by timer trigger)

In this mode, the voltage of one analog input pin (ANInm) is A/D converted once using the A/D conversion start trigger signal from the timer (motor control function) as a trigger, and the results are stored in one ADAnCRm register. The ANInm pin and the ADAnCRm register correspond one to one.

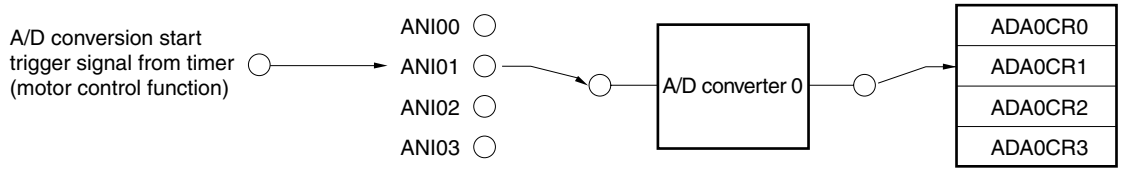
An A/Dn conversion end interrupt request signal (INTADn) is generated for each A/D conversion. After the end of A/D conversion, the A/D converter waits for a trigger.

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
ANInm	ADAnCRm

Remark n = 0, 1
m = 0 to 3

Figure 11-16. Example of 1-Buffer Mode Operation (1 Buffer of Continuous Select/One-Shot Select by Timer Trigger)



(1) The ADA0CE bit = 1 (enable)
(2) The A/D conversion start trigger signal from the timer (motor control function) is generated
(3) The ANI01 pin is A/D converted
(4) The conversion result is stored in the ADA0CR1 register
(5) The ADA0M0.ADA0EF bit = 0
(6) The INTAD0 interrupt request signal is generated
(7) If next trigger is input, return to (3)
(8) To end the conversion, the ADA0CE bit = 0 (stop)

Remark This is an operation example with the following setting.
ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits = 00 or 10, ADA0M0.ADA0TMD bit = 1,
ADA0M2.ADA0BS bit = 0, ADA0M2.ADA0TMD1 bit = 1,
ADA0S.ADA0S1 and ADA0S.ADA0S0 bits = 01

(2) 4-buffer mode operation (4 buffers of continuous select/one-shot select by timer trigger)

In this mode, the voltage of one analog input pin (ANInm) is A/D converted four times using the A/D conversion start trigger signal from the timer (motor control function) as a trigger, and the results are stored in the ADAnCRm register.

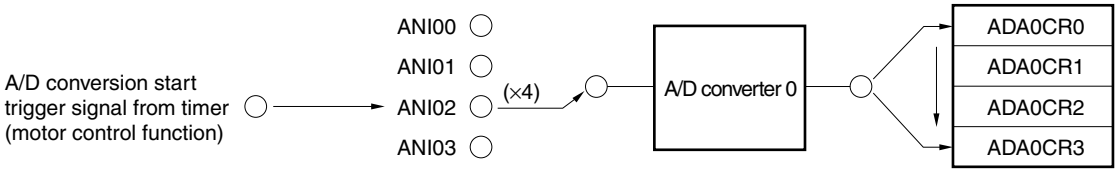
The A/Dn conversion end interrupt request signal (INTADn) is generated when the four A/D conversions end. After the end of A/D conversion, the A/D converter waits for a trigger.

This mode is suitable for applications in which the average of the A/D conversion results is calculated.

Analog Input Pin	A/D Conversion Result Register
ANInm	ADAnCR0
ANInm	ADAnCR1
ANInm	ADAnCR2
ANInm	ADAnCR3

Remark n = 0, 1
m = 0 to 3

Figure 11-17. Example of 4-Buffer Mode Operation
(4 Buffers of Continuous Select/One-Shot Select by Timer Trigger)



(1) The ADA0CE bit = 1 (enable)
(2) The A/D conversion start trigger signal from the timer (motor control function) is generated
(3) The ANI02 pin is A/D converted
(4) The conversion result is stored in the ADA0CR0 register
(5) The ANI02 pin is A/D converted
(6) The conversion result is stored in the ADA0CR1 register
(7) The ANI02 pin is A/D converted
(8) The conversion result is stored in the ADA0CR2 register
(9) The ANI02 pin is A/D converted
(10) The conversion result is stored in the ADA0CR3 register
(11) The ADA0M0.ADA0EF bit = 0
(12) The INTAD0 interrupt request signal is generated
(13) If next trigger is input, return to (3)
(14) To end the conversion, the ADA0CE bit = 0 (stop)

Remark This is an operation example with the following setting.
ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits = 00 or 10, ADA0M0.ADA0TMD bit = 1,
ADA0M2.ADA0BS bit = 1, ADA0M2.ADA0TMD1 bit = 1,
ADA0S.ADA0S1 and ADA0S.ADA0S0 bits = 10

11.6.2 Continuous scan mode/one-shot scan mode operations

In this mode, the analog input pins (ANInm) specified by the ADAnS register are selected sequentially from the ANIn0 pin using the A/D conversion start trigger signal from the timer (motor control function) as a trigger and A/D conversion is performed continuously. The A/D conversion results are stored in the ADAnCRm register corresponding to the analog input pin.

When conversion of all the specified analog input pins ends, the A/Dn conversion end interrupt request signal (INTADn) is generated. After the end of A/D conversion, the A/D converter waits for a trigger.

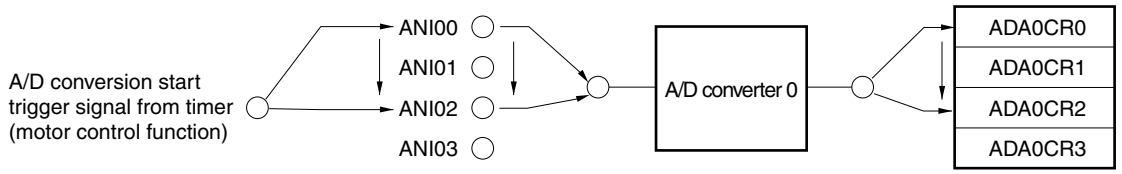
This mode is suitable for applications in which multiple analog input pins are constantly monitored.

In the continuous scan mode or one-shot scan mode, only the 1-buffer mode is supported.

Analog Input Pin	A/D Conversion Result Register
ANIn0	ADAnCR0
ANIn1	ADAnCR1
ANIn2	ADAnCR2
ANIn3	ADAnCR3

Remark n = 0, 1
m = 0 to 3

Figure 11-18. Example of Scan Mode Operation (Continuous Scan/One-Shot Scan by Timer Trigger)



(1) The ADA0CE bit = 1 (enable)
(2) The A/D conversion start trigger signal from the timer (motor control function)
(3) The ANI00 pin is A/D converted
(4) The conversion result is stored in the ADA0CR0 register
(5) The ANI01 pin is A/D converted
(6) The conversion result is stored in the ADA0CR1 register
(7) The ANI02 pin is A/D converted
(8) The conversion result is stored in the ADA0CR2 register
(9) The ADA0M0.ADA0EF bit = 0
(10) The INTAD0 interrupt request signal is generated
(11) If next trigger is input, return to (3)
(12) To end the conversion, the ADA0CE bit = 0 (stop)

Remark This is an operation example with the following setting.
ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits = 01 or 11, ADA0M0.ADA0TMD bit = 1,
ADA0M2.ADA0BS bit = 0, ADA0M2.ADA0TMD1 bit = 1,
ADA0S.ADA0S1 and ADA0S.ADA0S0 bits = 10

11.7 Operation in External Trigger Mode

In the external trigger mode, the analog input pins (ANIn0 to ANIn3) are A/D converted at the ADTRGn pin input timing.

The ADTRG0 pin has an alternate function as the P04/INTP4 pin and the ADTRG1 pin has an alternate function as the P05/INTP5 pin. To set the external trigger mode, set the PMC04 bit of port mode control register 0 (PMC0) to 1 and the ADA0M2.ADA0TMD1 bit to 0 with A/D converter 0. With A/D converter 1, set the PMC05 bit of port mode control register 0 (PMC0) to 1 and the ADA1M2.ADA1TMD1 bit to 0.

For the valid edge of the external input signal in the external trigger mode, the rising edge, falling edge, or both rising and falling edges can be specified by setting the ADAnM0.ADAnETS1 and ADAnM0.ADAnETS0 bits.

When the ADAnM0.ADAnCE bit is set (1), the A/D converter waits for a trigger and, when the trigger is input from the ADTRGn pin, starts A/D conversion.

After the end of A/D conversion, the conversion result is stored in A/Dn conversion result register m (ADAnCRm) and, at the same time, the A/Dn conversion end interrupt request signal (INTADn) is generated.

After the end of A/D conversion, the A/D converter waits for a trigger regardless of the operation mode set by the ADAnM0.ADAnMD1 and ADAnM0.ADAnMD0 bits.

When conversion is started, the ADAnM0.ADAnEF bit is set to 1 (conversion in progress). However, while the A/D converter is waiting for a trigger, the ADAnEF bit = 0 (conversion stopped).

If a valid trigger is input during A/D conversion, the conversion operation is stopped and started again from the beginning. If the ADAnM0, ADAnM2, and ADAnS registers are written during A/D conversion, the conversion is stopped and the A/D converter waits for a trigger again.

Remark n = 0, 1
m = 0 to 3

11.7.1 Continuous select mode/one-shot select mode operations

In this mode, the analog input pin (ANInm) specified by the ADAnS register is A/D converted. The conversion results are stored in the ADAnCRm register. In the continuous select mode or one-shot select mode, there are two select modes: 1-buffer mode and 4-buffer mode, according to the method of storing the A/D conversion results.

Remark n = 0, 1
m = 0 to 3

(1) 1-buffer mode (1 buffer of continuous select/one-shot select by external trigger)

In this mode, the voltage of one analog input pin (ANInm) is A/D converted once using the ADTRGn signal as a trigger. The conversion results are stored in one ADAnCRm register. The ANInm pin and the ADAnCRm register correspond one to one.

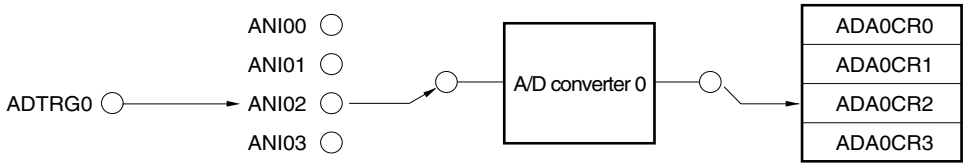
The A/Dn conversion end interrupt request signal (INTADn) is generated for each A/D conversion. After the end of A/D conversion, the A/D converter waits for a trigger.

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
ANInm	ADAnCRm

Remark n = 0, 1
m = 0 to 3

Figure 11-19. Example of 1-Buffer Mode Operation (1 Buffer of Continuous Select/One-Shot Select by External Trigger)



(1) The ADA0CE bit = 1 (enable)
(2) The external trigger signal (ADTRG0) is generated
(3) The ANI02 pin is A/D converted
(4) The conversion result is stored in the ADA0CR2 register
(5) The ADA0M0.ADA0EF bit = 0
(6) The INTAD0 interrupt request signal is generated
(7) If next trigger is input, return to (3)
(8) To end the conversion, the ADA0CE bit = 0 (stop)

Remark This is an operation example with the following setting.
ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits = 00 or 10, ADA0M0.ADA0TMD bit = 1,
ADA0M2.ADA0BS bit = 0, ADA0M2.ADA0TMD1 bit = 0,
ADA0S.ADA0S1 and ADA0S.ADA0S0 bits = 10

(2) 4-buffer mode (4 buffers of continuous select/one-shot select by external trigger)

In this mode, the voltage of one analog input pin (ANInm) is A/D converted four times using the ADTRGn signal as a trigger and the results are stored in the ADAnCRm register (n = 0, 1, m = 0 to 3).

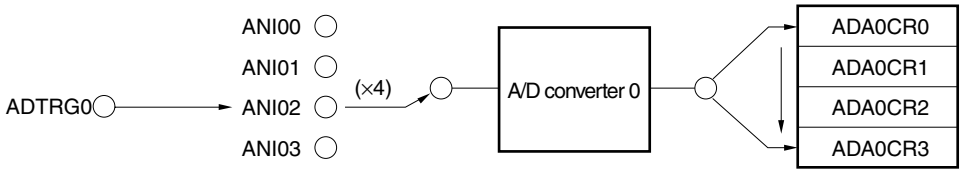
The A/Dn conversion end interrupt request signal (INTADn) is generated when the four A/D conversions end. After the end of A/D conversion, the A/D converter waits for a trigger.

This mode is suitable for applications in which the average of the A/D conversion results is calculated.

Analog Input Pin	A/D Conversion Result Register
ANInm	ADAnCR0
ANInm	ADAnCR1
ANInm	ADAnCR2
ANInm	ADAnCR3

Remark n = 0, 1
m = 0 to 3

Figure 11-20. Example of 4-Buffer Mode Operation (4 Buffers of Continuous Select/One-Shot Select by External Trigger)



(1) The ADA0CE bit = 1 (enable)
(2) The external trigger signal (ADTRG0) is generated
(3) The ANI02 pin is A/D converted
(4) The conversion result is stored in the ADA0CR0 register
(5) The ANI02 pin is A/D converted
(6) The conversion result is stored in the ADA0CR1 register
(7) The ANI02 pin is A/D converted
(8) The conversion result is stored in the ADA0CR2 register
(9) The ANI02 pin is A/D converted
(10) The conversion result is stored in the ADA0CR3 register
(11) The ADA0M0.ADA0EF bit = 0
(12) The INTAD0 interrupt request signal is generated
(13) If next trigger is input, return to (3)
(14) To end the conversion, the ADA0CE bit = 0 (stop)

Remark This is an operation example with the following setting.
ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits = 00 or 10, ADA0M0.ADA0TMD bit = 1,
ADA0M2.ADA0BS bit = 1, ADA0M2.ADA0TMD1 bit = 0,
ADA0S.ADA0S1 and ADA0S.ADA0S0 bit = 10

11.7.2 Continuous scan mode/one-shot scan mode operations

In this mode, the analog input pins (ANInm) specified by the ADAnS register are selected sequentially from the ANIn0 pin using the ADTRGn signal as a trigger, and A/D converted continuously. The A/D conversion results are stored in the ADAnCRm register corresponding to the analog input pin.

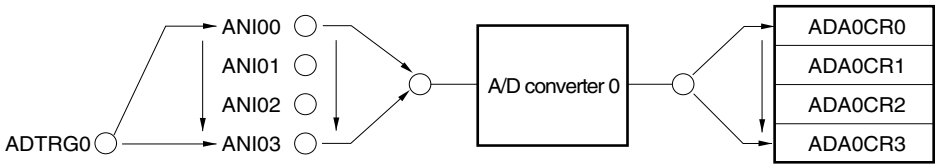
When conversion of all the specified analog input pins ends, the A/Dn conversion end interrupt request signal (INTADn) is generated. After the end of A/D conversion, the A/D converter waits for a trigger.

This is most suitable for applications in which multiple analog inputs are constantly monitored.

Analog Input Pin	A/D Conversion Result Register
ANIn0	ADAnCR0
ANIn1	ADAnCR1
ANIn2	ADAnCR2
ANIn3	ADAnCR3

Remark n = 0, 1
m = 0 to 3

Figure 11-21. Example of Scan Mode Operation (Continuous Scan/One-Shot Scan by External Trigger)



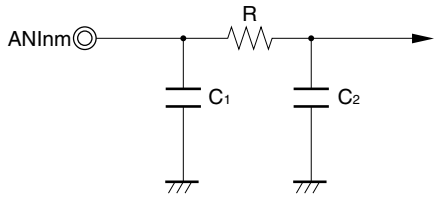
(1) The ADA0CE bit = 1 (enable)
(2) The external trigger signal (ADTRG0) is generated
(3) The ANI00 pin is A/D converted
(4) The conversion result is stored in the ADA0CR0 register
(5) The ANI01 pin is A/D converted
(6) The conversion result is stored in the ADA0CR1 register
(7) The ANI02 pin is A/D converted
(8) The conversion result is stored in the ADA0CR2 register
(9) The ANI03 pin is A/D converted
(10) The conversion result is stored in the ADA0CR3 register
(11) The ADA0M0.ADA0EF bit = 0
(12) The INTAD0 interrupt request signal is generated
(13) If next trigger is input, return to (3)
(14) To end the conversion, the ADA0CE bit = 0 (stop)

Remark This is an operation example with the following setting.
ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits = 01 or 11, ADA0M0.ADA0TMD bit = 1,
ADA0M2.ADA0BS bit = 0, ADA0M2.ADA0TMD1 bit = 0,
ADA0S.ADA0S1 and ADA0S.ADA0S0 bits = 11

11.8 Internal Equivalent Circuit

The following figure shows the equivalent circuit of the analog input block.

Figure 11-22. ANInm Pin Internal Equivalent Circuit

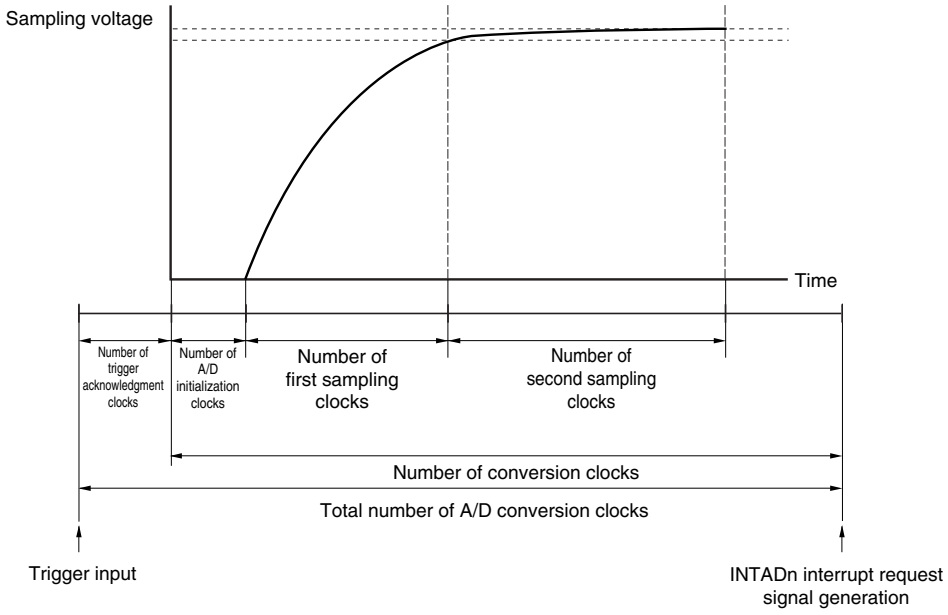


R	C_1	C_2
4.0 kΩ	15 pF	4.2 pF

Remarks 1. The maximum values are shown (reference values).

2. n = 0, 1
m = 0 to 3

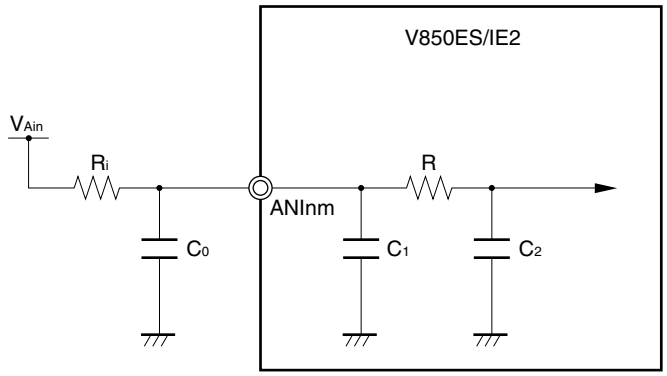
Caution A/D converters 0 and 1 perform the first sampling when A/D conversion starts after A/D initialization. Sampling power is almost fully charged during this period. Therefore, the sampling error can be calculated under this condition. After that, the converters perform the second sampling to correct the error.
The sampling period of A/D converters 0 and 1 is from the start of conversion to the end of the second sampling period. Avoid noise during this period.



ADAnM1 Register		Number of A/D Initialization Clocks	Number of First Sampling Clocks	Number of Second Sampling Clocks
ADAnFR1 Bit	ADAnFR0 Bit			
0	1	2	16	18
1	0	3	24	27
1	1	4	32	36

An example of calculating an overall error of A/D converters 0 and 1 is shown below.

Figure 11-23. Example of Calculating Overall Error of A/D Converters 0 and 1



fXX (MHz)	Conversion Time (μs)	A/D Initialization Clock (μs)	First Sampling (μs)	Second Sampling (μs)	R (kΩ)	C1 (pF)	C2 (pF)	C0 (pF)	Ri (kΩ)	Sampling Error (LSB)[Note]
20	3.10 (62 clocks)	0.1 (2 clocks)	0.8 (16 clocks)	0.9 (18 clocks)	4.0	15	4.2	100	1.0	18.4
								100	0.5	0.2
								100	0.25	0.1 or lower
								100	0.1	0.1 or lower
								50	1.0	1.0
								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. n = 0, 1
m = 0 to 3

3. fXX: System clock frequency

11.9 Notes on Operation

11.9.1 Stopping conversion operation

When the ADAnM0.ADAnCE bit is cleared to 0 during a conversion operation, the conversion operation stops and the conversion results are not stored in A/Dn conversion result register m (ADAnCRm).

The ADAnCE bit is not cleared to 0 even after the A/Dn conversion end interrupt request signal (INTADn) has been generated in all modes.

Remark n = 0, 1, m = 0 to 3

11.9.2 Timer/external trigger interval

Make sure that the occurrence interval of the trigger in timer trigger mode or external trigger mode is longer than the total number of conversion clocks specified by the ADAnM1.ADAnFR1 and ADAnM1.ADAnFR0 bits (see **Table 11-2 Number of Conversion Clocks**).

(1) When 0 < trigger occurrence interval < total number of A/D conversion clocks

When the timer/external trigger is input during a conversion operation, the conversion operation is aborted and the conversion starts according to the last timer/external trigger input.

When conversion operations are aborted, the conversion results from the conversion operation immediately before are not stored in the ADAnCRm register. Note, therefore, that the generation of the INTADn signal and storing of the result in the ADAnCRm register are not guaranteed.

Remark n = 0, 1, m = 0 to 3

(2) When trigger occurrence interval ≥ total number of A/D conversion clocks

The INTADn signal is generated, and the value at the end of conversion is correctly stored in the ADAnCRm register. Design so that the trigger occurrence interval is equal or greater than the total number of A/D conversion clocks.

Remark n = 0, 1, m = 0 to 3

11.9.3 Operation in standby mode

(1) HALT mode

In this mode, A/D conversion continues.

(2) IDLE mode, STOP mode

As clock supply to A/D converters 0 and 1 is stopped, no conversion operations are performed.

When these modes are released by the maskable interrupt request signal input pin[Note], the ADAnM0, ADAnM1, ADAnM2, and ADAnS registers and A/Dn conversion result register m (ADAnCRm) hold their values. However, when the IDLE or STOP mode is set during a conversion operation, the conversion operation is suspended. At this time, if the mode released by the maskable interrupt request signal input pin[Note], the conversion operation resumes. At this time, the A/Dn conversion end interrupt request signal (INTADn) may be generated, but the conversion result written to the ADAnCRm register will be undefined.

Note INTP0 to INTP5

Remark n = 0, 1, m = 0 to 3

11.9.4 Timer interrupt request signal in timer trigger mode

The timer interrupt request signal (TQTADT1n) becomes an A/D conversion start trigger and starts the conversion operation. When this happens, the timer interrupt request signal also functions as an interrupt for the CPU. In order to prevent the generation of interrupts for the CPU, disable interrupts using the mask bits of the interrupt control register.

Remark n = 0, 1

11.9.5 Re-conversion start trigger input during stabilization time

If the stabilization time end timing and the register writing conflict or the stabilization time end timing and the trigger input conflict, the stabilization time is inserted again.

11.9.6 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.

11.9.7 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 if the output impedance from the analog input source is too high.

- 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, lower the output impedance from the analog input source or execute A/D conversion twice consecutively on the same channel, and discard the first conversion result.

11.9.8 Restrictions on setting one-shot mode and software trigger mode

If the A/D converters 0 and 1 are set in the one-shot select mode and software trigger mode (ADAnM0 register = 1010XX0XB) or one-shot scan mode and software trigger mode (ADAnM0 register = 1011XX0XB), a re-conversion operation should be performed in a new condition when data is written to any of the ADAnM0, ADAnM2, and ADAnS registers upon completion of an A/D conversion operation. However, the re-conversion operation is not performed but the conversion operation is enabled (ADAnM0.ADAnCE bit = 1) and stopped (ADAnM0.ADAnEF bit = 0). The A/Dn conversion end interrupt request signal (INTADn) is not generated, nor is the last A/D conversion result stored. However, the data is correctly written to any of the ADAnM0, ADAnM2, and ADAnS registers.

If this happens, normal operation can be restored by setting the ADAnM0.ADAnCE bit to 1.

For example, if the ANIn0 and ANIn1 pins are set in the scan mode (ADAnS register = 00000001B) and data is written to the ADAnM0 register upon completion of an A/D conversion operation in the one-shot scan mode and software trigger mode (ADAnM0 register = 1011XX0XB), the signal of the ANIn0 pin is correctly converted and the conversion result is correctly stored in the ADAnCR0 register. However, the result of converting the signal of the ANIn1 pin which has been performed immediately before the completion of the A/D conversion is not stored in the ADAnCR1 register, nor is the INTADn interrupt request signal generated.

[Countermeasure]

The above restriction can be avoided by performing any of steps <1> to <3>, below.

<1> Before writing to any of the ADAnM0, ADAnM2, and ADAnS registers, confirm that A/D conversion is stopped (ADAnM0.ADAnEF bit = 0).

<2> After disabling the interrupt (PSW.ID bit = 1), execute an instruction that writes data to any of the ADAnM0, ADAnM2, and ADAnS registers and an instruction that sets the ADAnM0.ADAnCE bit to 1 consecutively, and then enable the interrupt (PSW.ID bit = 0).
This action is to avoid coincidence between the completion of the A/D conversion operation and writing to the ADAnM0, ADAnM2, or ADAnS register. If, for example, executing a write instruction and the completion of the A/D conversion operation coincide and thus the A/D conversion is stopped, the A/D conversion can be started by setting of the ADAnCE bit to 1. If the ADAnM0.ADAnCE bit = 1, the ADAnCE bit is set to 1 again consecutively.

<3> Disable the A/D conversion operation by clearing the ADAnCE bit to 0, write data to any of the ADAnM0, ADAnM2, and ADAnS registers, enable the A/D conversion operation by setting the ADAnCE bit to 1, and start the A/D conversion.

11.10 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_{REFn} – 0)/100
= AV_{REFn}/100

1 LSB is as follows when the resolution is 10 bits.

1 LSB = $1/2^{10}$ = 1/1,024
= 0.098%FSR

Accuracy has no relation to resolution, but is determined by overall error.

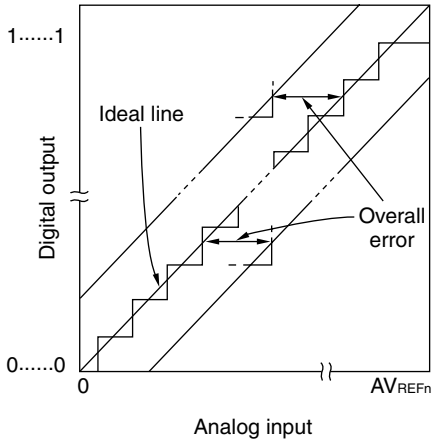
(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, linearity error and errors that are combinations of these express the overall error.

Note that the quantization error is not included in the overall error in the characteristics table.

Figure 11-24. Overall Error

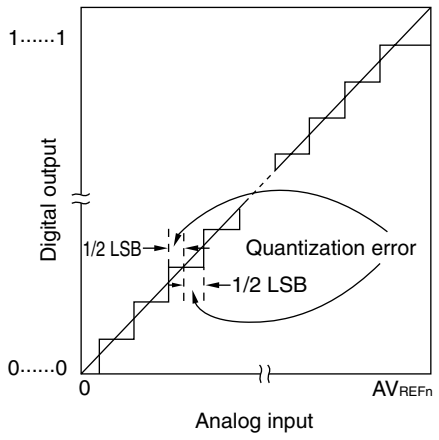


(3) Quantization error

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.

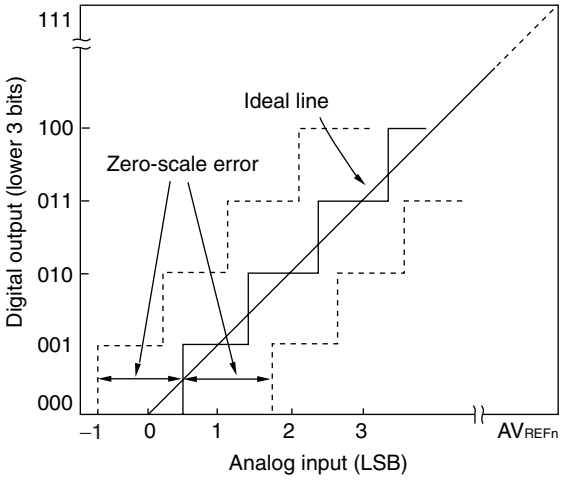
Figure 11-25. Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2 LSB) when the digital output changes from 0……000 to 0……001.

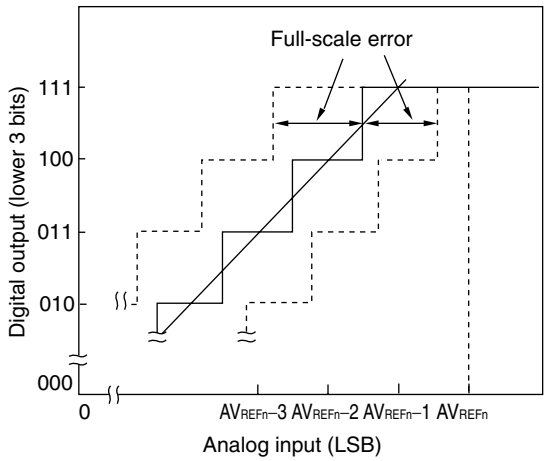
Figure 11-26. Zero-Scale Error



(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 11-27. Full-Scale Error

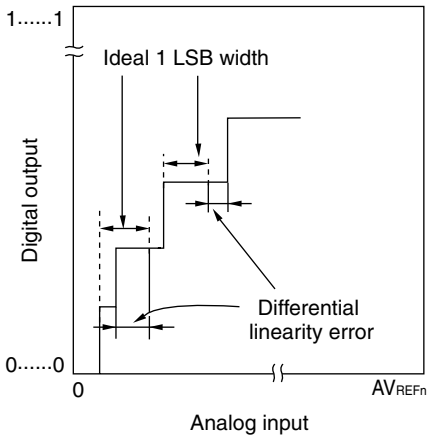


(6) Differential linearity error

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_{REFn}. See **11.10 (2) Overall error** for when the input voltage is increased or decreased, or when two or more channels are used.

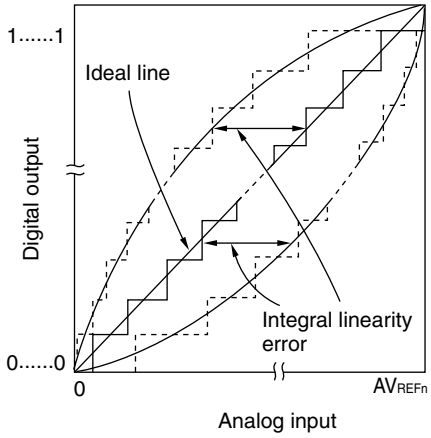
Figure 11-28. Differential Linearity Error



(7) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

Figure 11-29. Integral Linearity Error



(8) Conversion time

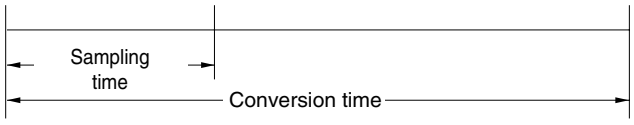
This expresses the time from when the trigger is generated to when the digital output is obtained.

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 11-30. Sampling Time



CHAPTER 12 ASYNCHRONOUS SERIAL INTERFACE A (UARTA)

The V850ES/IE2 incorporates UARTA0 and UARTA1.

12.1 Features

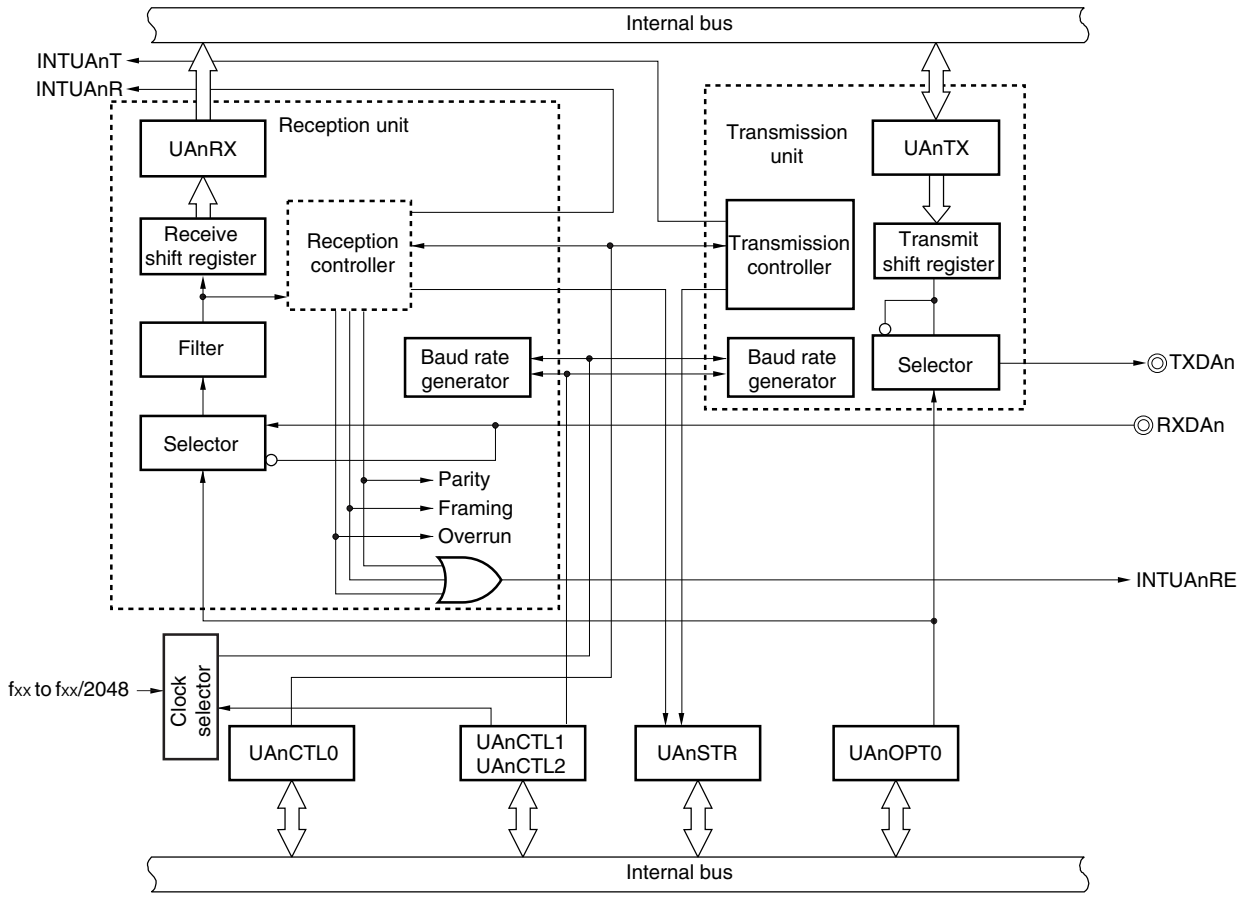
- Transfer rate: 300 bps to 1.25 Mbps (using peripheral clock (f_{XX}) of 20 MHz and dedicated baud rate generator)
- Full-duplex communication: Internal UARTA receive data register n (UAnRX)
  Internal UARTA transmit data register n (UAnTX)
- 2-pin configuration: TXDAn: Transmit data output pin
  RXDAn: 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, 1

12.2 Configuration

The block diagram of the UARTAn is shown below.

<R> **Figure 12-1. Block Diagram of UARTAn**



Remarks 1. n = 0, 1

2. For the configuration of the baud rate generator, see **Figure 12-9**.

UARTAn includes the following hardware units.

Table 12-1. 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.

<R> **(5) UARTAn status register (UAnSTR)**

The UAnSTRn 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.

12.3 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

	<7>	<6>	<5>	<4>	3	2	1	0
UAnCTL0 (n = 0, 1)	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

<R>

- 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 **12.6 (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

<R>

- 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 **12.6 (1) (a) Base clock**).

<R>

- 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. When a start bit is received before data reception is enabled, the start bit is ignored.
- 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 **12.5.6 Parity types and operations**.

(2) UARTAn control register 1 (UAnCTL1)

For details, see **12.6 (2) UARTAn control register 1 (UAnCTL1)**.

(3) UARTAn control register 2 (UAnCTL2)

For details, see **12.6 (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

UAnOPT0 (n = 0, 1)

7	6	5	4	3	2	1	0
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 (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 clear 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

	<7>	6	5	4	3	<2>	<1>	<0>
UAnSTR	UAnTSF	0	0	0	0	UAnPE	UAnFE	UAnOVE

(n = 0, 1)

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 UAnTXB bit
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 UAnRXB 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

	7	6	5	4	3	2	1	0
UAnRX								

(n = 0, 1)

(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). When the data of the UAnTX register has been transferred 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

	7	6	5	4	3	2	1	0
UAnTX								

(n = 0, 1)

12.4 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 12-2. 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.

12.5 Operation

12.5.1 Data format

Full-duplex serial data reception and transmission is performed.

As shown in Figure 12-2, one data frame of transmit/receive data consists of a start bit, character bits, parity bit, and stop bit(s).

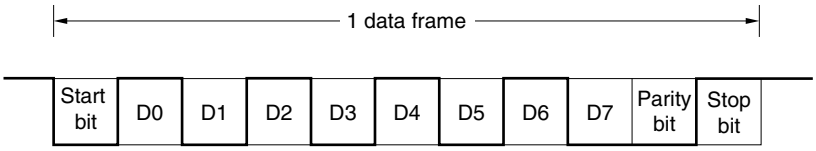
Specification of the character bit length within 1 data frame, parity selection, specification of the stop bit length, and specification of MSB/LSB-first transfer are performed using the UAnCTL0 register.

Moreover, control of UARTAn output/inverted output for the TXDAn bit is performed using the UAnOPT0.UAnTDL bit.

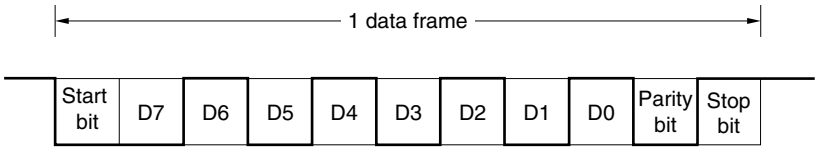
- Start bit ..... 1 bit
- Character bits..... 7 bits/8 bits
- Parity bit ..... Even parity/odd parity/0 parity/no parity
- Stop bit..... 1 bit/2 bits

Figure 12-2. UARTA Transmit/Receive Data Format

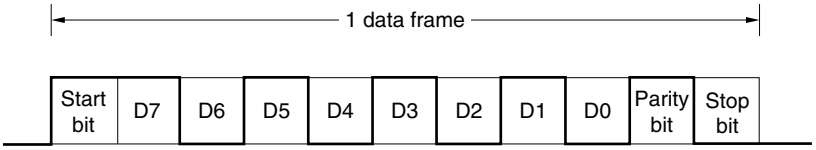
(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H



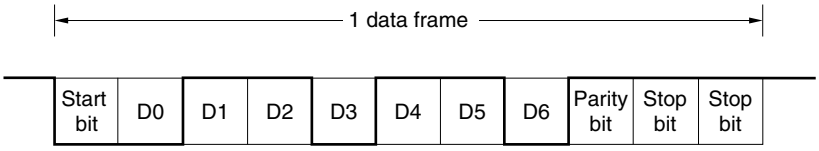
(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H



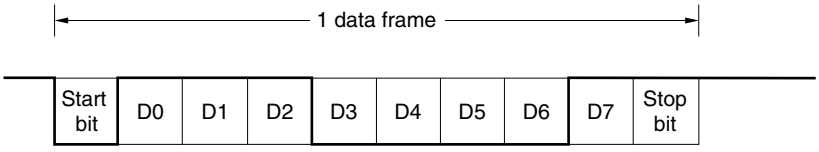
(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H, TXDAn inversion



(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



12.5.2 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.

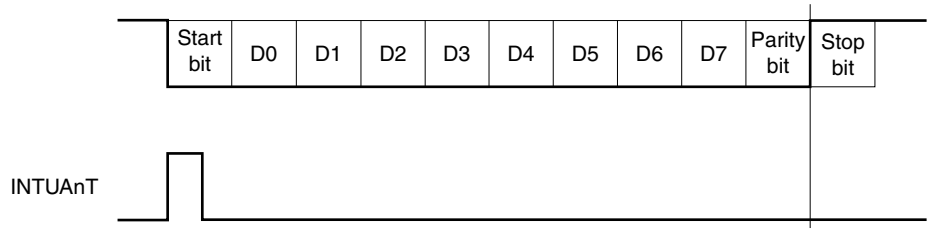
Since the CTS (transmit enable signal) input pin is not provided in UARTAn, use a port to check that reception is enabled at the transmit destination.

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 12-3. UART Transmission



Remarks 1. LSB first

2. n = 0, 1

12.5.3 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, 1

Figure 12-4. Continuous Transmission Processing Flow

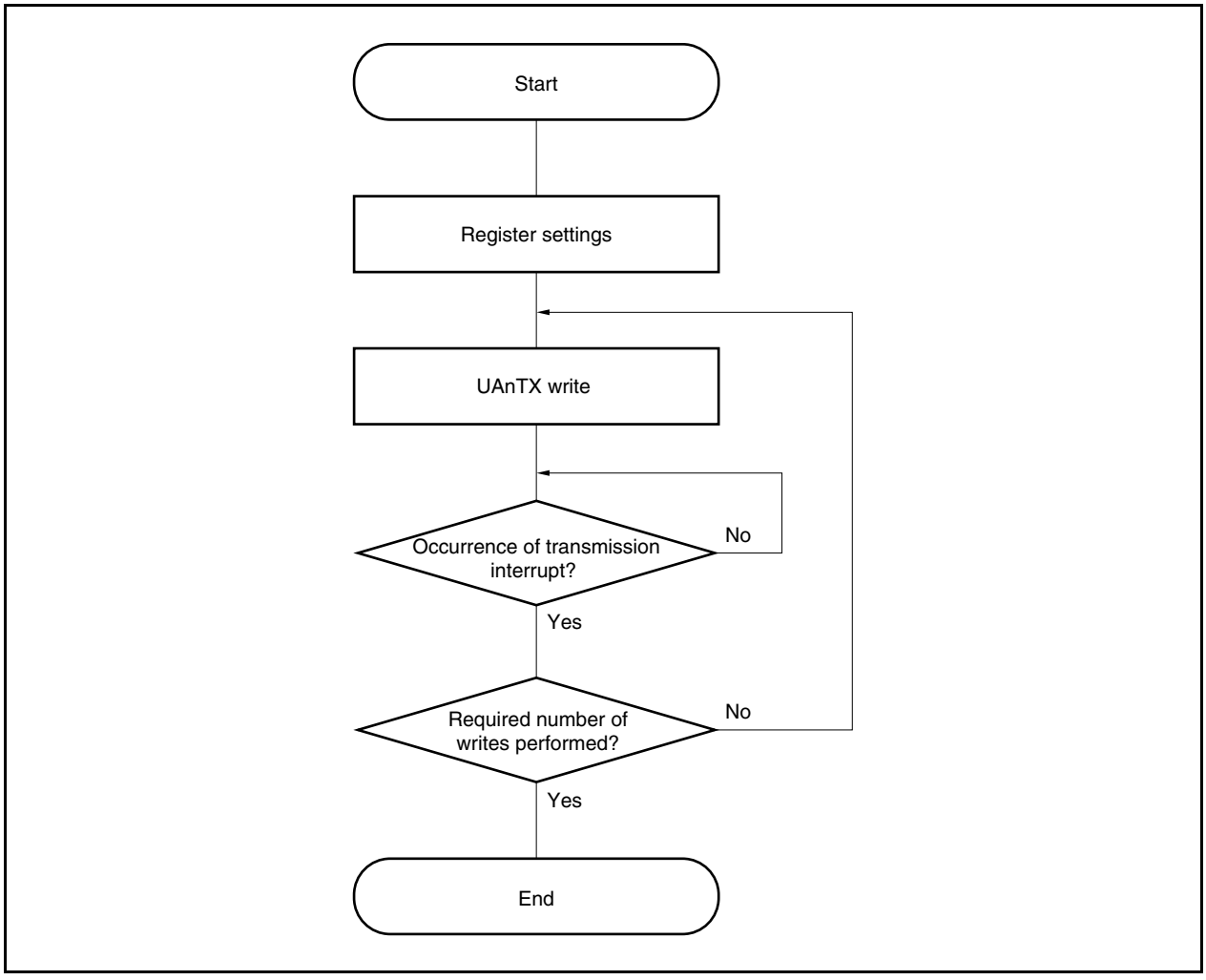
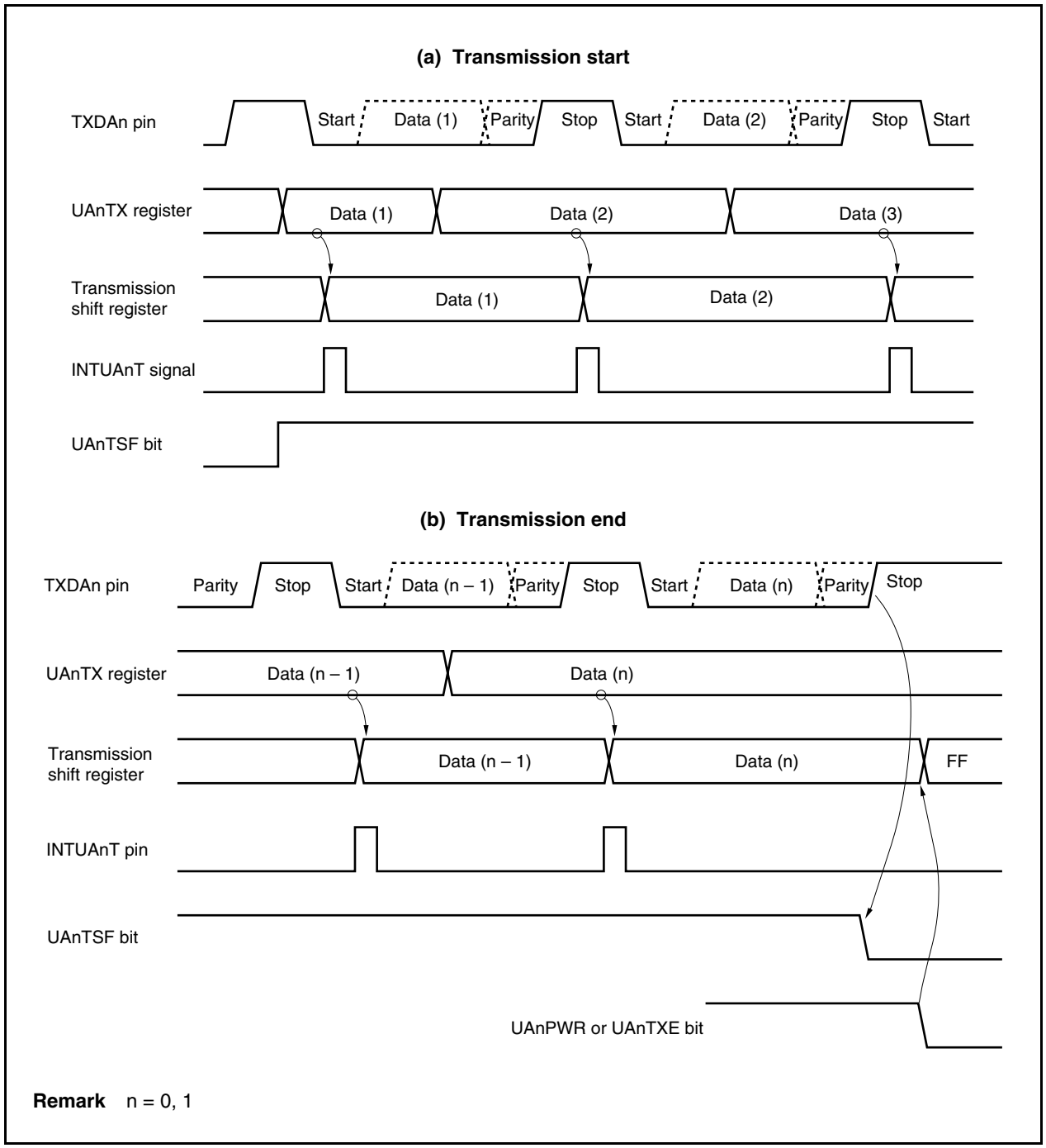


Figure 12-5. Continuous Transmission Operation Timing



Remark n = 0, 1

12.5.4 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.

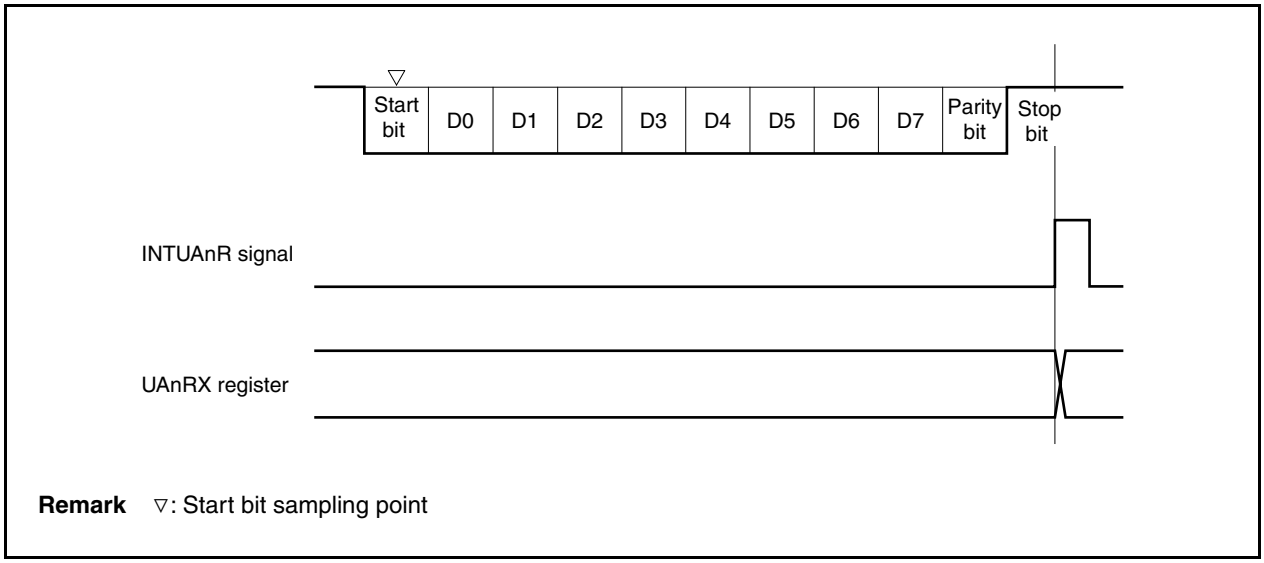
First the falling edge of the RXDAn pin is detected and sampling is started at the falling edge. The start bit is recognized if the RXDAn pin is low level at the start bit sampling point. After a start bit has been recognized, the receive operation starts, and serial data is saved to the UARTAn receive 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, 1

Figure 12-6. 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.

12.5.5 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, 1

- 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

12.5.6 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.

12.5.7 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 12-8**). See **12.6 (1) (a) Base clock** regarding the base clock.

Moreover, since the circuit is as shown in Figure 12-7, the processing that goes on within the receive operation is delayed by 3 clocks in relation to the external signal status.

Remark n = 0, 1

Figure 12-7. Noise Filter Circuit

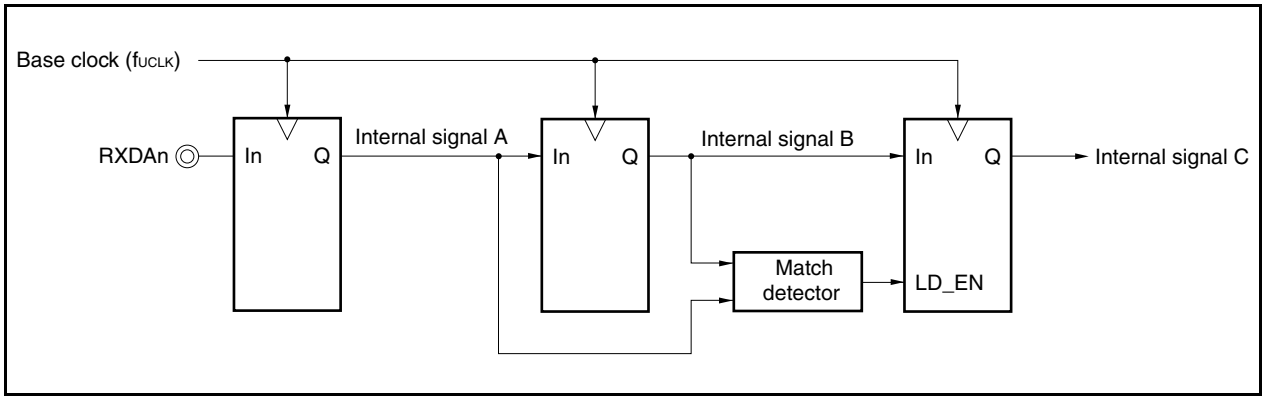
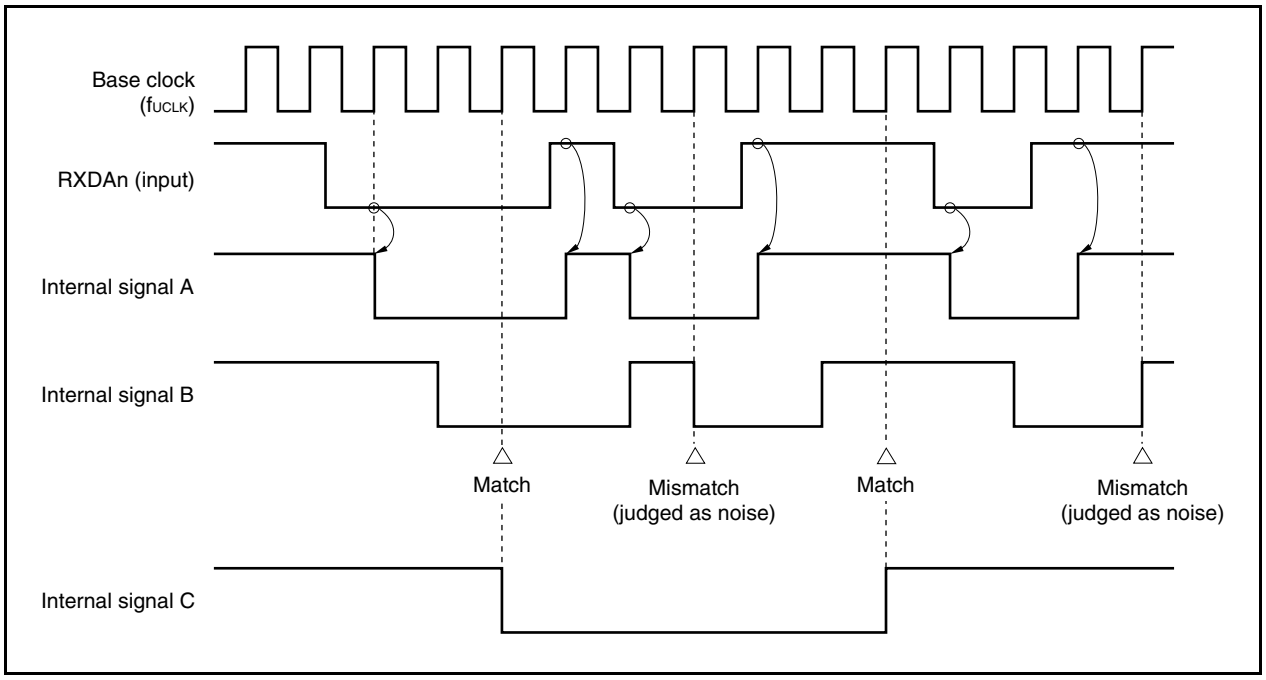


Figure 12-8. Timing of RXDAn Signal Judged as Noise



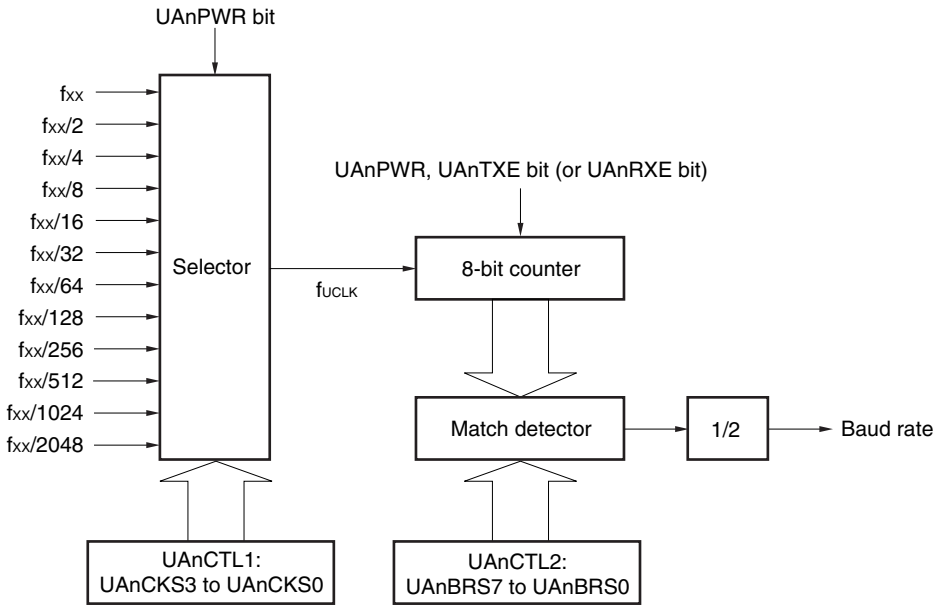
12.6 Dedicated Baud Rate Generator

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 12-9. 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, 1

2. f_{XX}: 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 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

	7	6	5	4	3	2	1	0
UAnCTL1 (n = 0, 1)	0	0	0	0	UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0

UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0	Base clock (f_{UCLK}) selection
0	0	0	0	fxx
0	0	0	1	fxx/2
0	0	1	0	fxx/4
0	0	1	1	fxx/8
0	1	0	0	fxx/16
0	1	0	1	fxx/32
0	1	1	0	fxx/64
0	1	1	1	fxx/128
1	0	0	0	fxx/256
1	0	0	1	fxx/512
1	0	1	0	fxx/1,024
1	0	1	1	fxx/2,048
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

	7	6	5	4	3	2	1	0
UAnCTL2 (n = 0, 1)	UAnBRS7	UAnBRS6	UAnBRS5	UAnBRS4	UAnBRS3	UAnBRS2	UAnBRS1	UAnBRS0

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 = 20 MHz = 20,000,000 Hz

Setting value of UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits = 0000B (f_{UCLK} = 20,000,000 Hz)

Setting value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits = 01000001B (k = 65)

Target baud rate = 153,600

Baud rate = 20,000,000/ (2 × 65) = 153,846 [bps]

Error = (153,846/153,600 − 1) × 100

= 0.160 [%]

(6) Baud rate setting example

Table 12-3. Baud Rate Generator Setting Data

Baud Rate (bps)	f_{XX} = 20 MHz		
	UAnCTL1	UAnCTL2	ERR (%)
300	08H	82H	0.16
600	07H	82H	0.16
1,200	06H	82H	0.16
2,400	05H	82H	0.16
4,800	04H	82H	0.16
9,600	03H	82H	0.16
19,200	02H	82H	0.16
31,250	01H	A0H	0
38,400	01H	82H	0.16
76,800	00H	82H	0.16
153,600	00H	41H	0.16
312,500	00H	20H	0
625,000	00H	10H	0
1,250,000	00H	8H	0

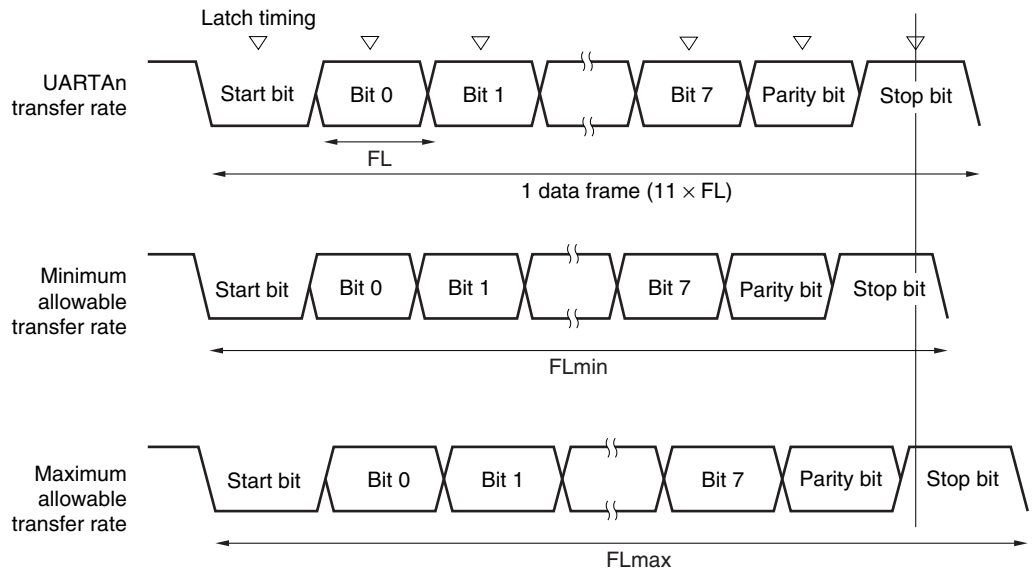
Remark f_{XX}: Peripheral clock frequency
ERR: Baud rate error (%)

(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 12-10. Allowable Baud Rate Range During Reception



Remark n = 0, 1

As shown in Figure 12-10, 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, 1)

k: Setting value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (n = 0, 1)

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 12-4. 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%

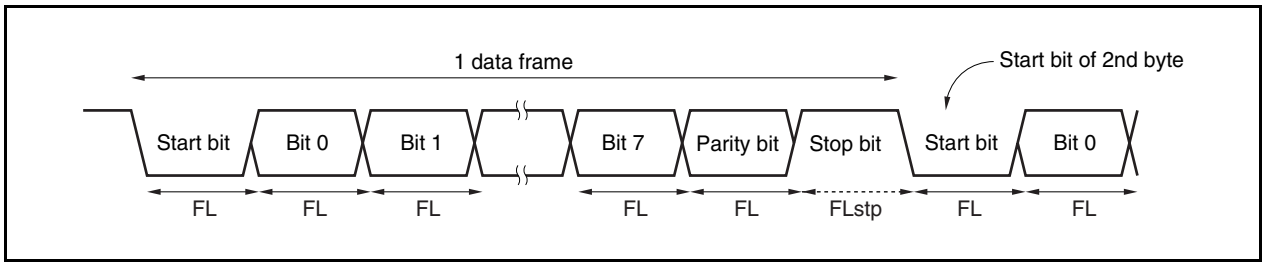
Remarks 1. The reception accuracy depends on the bit count in 1 frame, the input clock frequency, and the division ratio (k). The higher the input clock frequency and the larger the division ratio (k), the higher the accuracy.

2. k: Setting value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (n = 0, 1)

(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 12-11. 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.

$$\text{Transfer rate} = 11 \times FL + (2/f_{UCLK})$$

12.7 Cautions

When the clock supply to UARTAn is stopped (for example, in IDLE1 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, 1

CHAPTER 13 3-WIRE VARIABLE-LENGTH SERIAL I/O (CSIB)

The V850ES/IE2 incorporates CSIB0.

13.1 Features

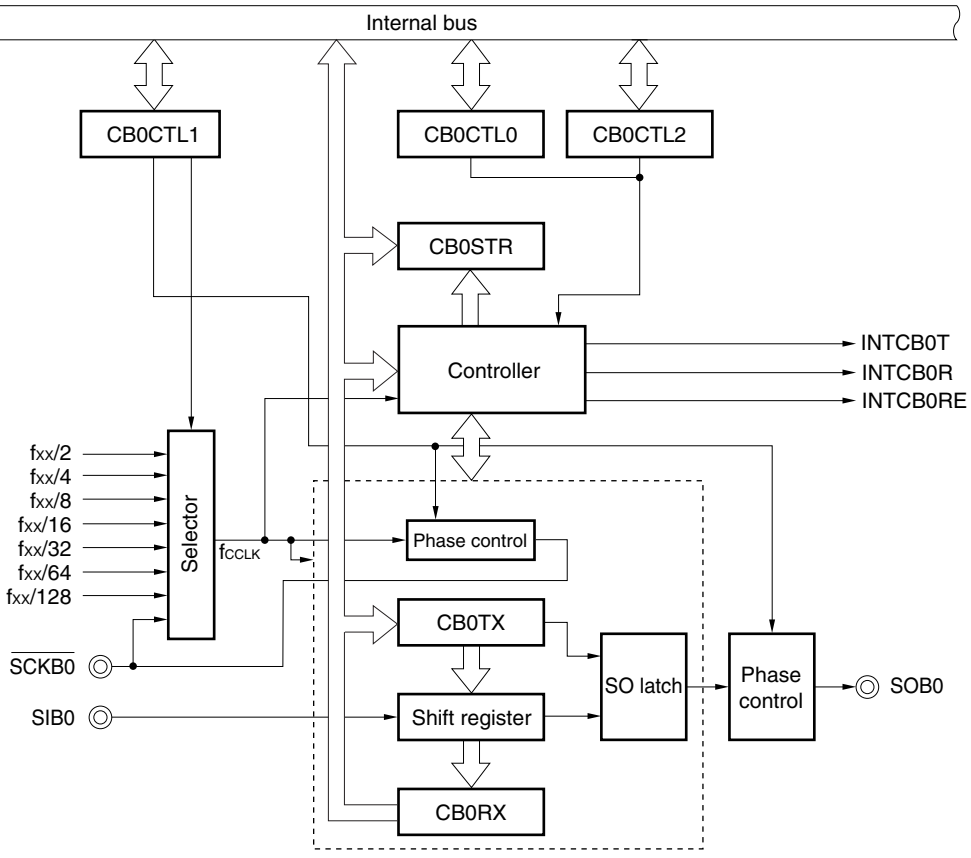
- Transfer rate: 8 Mbps (using internal clock)
- Master mode and slave mode selectable
- 8-bit to 16-bit transfer, 3-wire serial interface
- Interrupt request signals (INTCB0RE, INTCB0T, INTCB0R)
- Serial clock and data phase switchable
- Transfer data length selectable in 1-bit units between 8 and 16 bits
- Transfer data MSB-first/LSB-first switchable
- 3-wire transfer SOB0: Serial data output
  SIB0: Serial data input
  $\overline{\text{SCKB0}}$: Serial clock I/O

  Transmission mode, reception mode, and transmission/reception mode specifiable

13.2 Configuration

The following shows the block diagram of CSIB0.

Figure 13-1. Block Diagram of CSIB0



Remark f_{CCLK}: Communication clock (8 MHz (max.))

CSIB0 includes the following hardware.

Table 13-1. Configuration of CSIB0

Item	Configuration
Registers	CSIB0 receive data register (CB0RX) CSIB0 transmit data register (CB0TX)
Control registers	CSIB0 control register 0 (CB0CTL0) CSIB0 control register 1 (CB0CTL1) CSIB0 control register 2 (CB0CTL2) CSIB0 status register (CB0STR)

(1) CSIB0 receive data register (CB0RX)

The CB0RX register is a 16-bit buffer register that holds receive data.

This register is read-only, in 16-bit units.

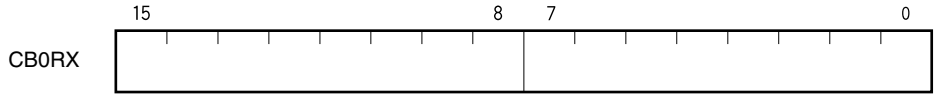
The receive operation is started by reading the CB0RX register in the reception enabled status.

If the transfer data length is 8 bits, the lower 8 bits of this register are read-only in 8-bit units as the CB0RXL register.

Reset sets this register to 0000H.

In addition to reset, the CB0RX register can be initialized by clearing (to 0) the CB0CTL0.CB0PWR bit.

After reset: 0000H R Address: CB0RX FFFFFD04H, CB0RXL FFFFFD04H



(2) CSIB0 transmit data register (CB0TX)

The CB0TX register is a 16-bit buffer register used to write the CSIB0 transfer data.

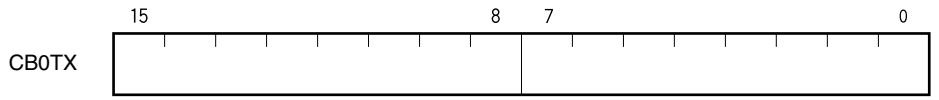
This register can be read or written in 16-bit units.

The transmit operation is started by writing data to the CB0TX 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 CB0TXL register.

Reset sets this register to 0000H.

After reset: 0000H R/W Address: CB0TX FFFFFD06H, CB0TXL FFFFFD06H



Remark The communication start conditions are shown below.

Transmission mode (CB0TXE bit = 1, CB0RXE bit = 0):	Write to CB0TX register
Transmission/reception mode (CB0TXE bit = 1, CB0RXE bit = 1):	Write to CB0TX register
Reception mode (CB0TXE bit = 0, CB0RXE bit = 1):	Read from CB0RX register

13.3 Control Registers

The following registers are used to control CSIB0.

- CSIB0 control register 0 (CB0CTL0)
- CSIB0 control register 1 (CB0CTL1)
- CSIB0 control register 2 (CB0CTL2)
- CSIB0 status register (CB0STR)

(1) CSIB0 control register 0 (CB0CTL0)

CB0CTL0 is a register that controls the CSIB0 serial transfer operation.

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

Reset sets this register to 01H.

(1/2)

After reset: 01H R/W Address: FFFFFD00H

	<7>	<6>	<5>	<4>	3	2	1	<0>
CB0CTL0	CB0PWR	CB0TXE Note	CB0RXE Note	CB0DIR Note	0	0	CB0TMS Note	CB0SCE

CB0PWR	Specification of CSIB0 operation disable/enable
0	Disable CSIB0 operation and reset the CB0STR register
1	Enable CSIB0 operation
• The CB0PWR bit controls the CSIB0 operation and resets the internal circuit.	

CB0TXE Note	Specification of transmit operation disable/enable
0	Disable transmit operation
1	Enable transmit operation
• The SOB0 output is low level when the CB0TXE bit is 0.	

CB0RXE Note	Specification of receive operation disable/enable
0	Disable receive operation
1	Enable receive operation
• When the CB0RXE 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 (CB0RX register) is not updated.	

Note These bits can only be rewritten when the CB0PWR bit = 0. However, CB0PWR bit = 1 can also be set at the same time as rewriting these bits.

Caution Be sure to clear bits 3 and 2 to "0".

(2/2)

CB0DIR[Note 1]	Specification of transfer direction mode (MSB/LSB)
0	MSB first
1	LSB first

CB0TMS[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 (CB0CTL1.CB0DAP bit = 1), write the transfer data to the CB0TX register after checking that the CB0STR.CB0TSF bit is 0.

CB0SCE	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
  Clear the CB0SCE bit to 0 before reading the receive data (CB0RX register)[Note 2].
  (b) In continuous reception mode
  Clear the CB0SCE 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 CB0SCE bit to 1[Note 4].
- In single transmission or transmission/reception mode, or continuous transmission or transmission/reception mode
  The function of the CB0SCE bit is invalid. It is recommended to set this bit to 1.

Notes 1. These bits can only be rewritten when the CB0PWR bit = 0. However, the CB0PWR can be set to 1 at the same time as these bits are rewritten.

2. If the CB0SCE bit is read while it is 1, the next communication operation is started.

3. The CB0SCE bit is not cleared 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 CB0SCE bit to 1 and perform a dummy read of the CB0RX register.

4. To start the reception, a dummy read is necessary.

(a) How to use CB0SCE bit

(i) In single reception mode

<1> When the reception of the last data is completed with INTCB0R interrupt servicing, clear the CB0SCE bit to 0, and then read the CB0RX register.

<2> When the reception is disabled after the reception of the last data has been completed, check that the CB0STR.CB0TSF bit is 0, and then clear the CB0PWR and CB0RXE bits to 0. To continue reception, set the CB0SCE bit to 1 and start the next receive operation by performing a dummy read of the CB0RX register.

(ii) In continuous reception mode

<1> Clear the CB0SCE bit to 0 during reception of the last data with INTCB0R interrupt servicing by the reception before the last reception, and then read the CB0RX register.

<2> After receiving the INTCB0R signal of the last reception, read the last data from the CB0RX register.

<3> When the reception is disabled after the reception of the last data has been completed, check that the CB0STR.CB0TSF bit is 0, and then clear the CB0PWR and CB0RXE bits to 0. To continue reception, set the CB0SCE bit to 1 and start the next receive operation by performing a dummy read of the CB0RX register.

Caution In continuous reception mode, the serial clock is not stopped until the reception executed when the CB0SCE bit is cleared to 0 is completed after the reception is started by a dummy read.

(2) CSIB0 control register 1 (CB0CTL1)

CB0CTL1 is an 8-bit register that controls the CSIB0 serial transfer operation.

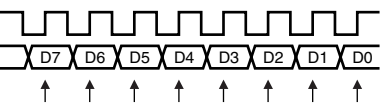
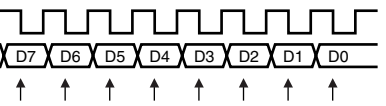
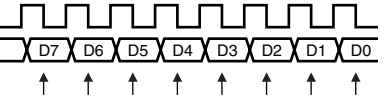
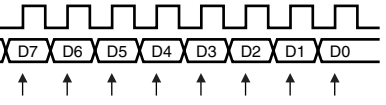
This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

Caution The CB0CTL1 register can be rewritten only when the CB0CTL0.CB0PWR bit = 0.

After reset: 00H R/W Address: FFFFFD01H

	7	6	5	4	3	2	1	0
CB0CTL1	0	0	0	CB0CKP	CB0DAP	CB0CKS2	CB0CKS1	CB0CKS0

	CB0CKP	CB0DAP	Specification of data transmission/ reception timing in relation to $\overline{SCKB0}$
Communication type 1	0	0	
Communication type 2	0	1	
Communication type 3	1	0	
Communication type 4	1	1	

CB0CKS2	CB0CKS1	CB0CKS0	Communication clock	Mode
0	0	0	$f_{XX}/2$	Master mode
0	0	1	$f_{XX}/4$	Master mode
0	1	0	$f_{XX}/8$	Master mode
0	1	1	$f_{XX}/16$	Master mode
1	0	0	$f_{XX}/32$	Master mode
1	0	1	$f_{XX}/64$	Master mode
1	1	0	$f_{XX}/128$	Master mode
1	1	1	External clock ($\overline{SCKB0}$)	Slave mode

Caution Set f_{CCLK} to 8 MHz or lower.

(3) CSIB0 control register 2 (CB0CTL2)

CB0CTL2 is an 8-bit register that controls the number of CSIB0 serial transfer bits.

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

Reset sets this register to 00H.

Caution The CB0CTL2 register can be rewritten only when the CB0CTL0.CB0PWR bit = 0 or when both the CB0TXE and CB0RXE bits = 0.

After reset: 00H R/W Address: FFFFFD02H

	7	6	5	4	3	2	1	0
CB0CTL2	0	0	0	0	CB0CL3	CB0CL2	CB0CL1	CB0CL0

CB0CL3	CB0CL2	CB0CL1	CB0CL0	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

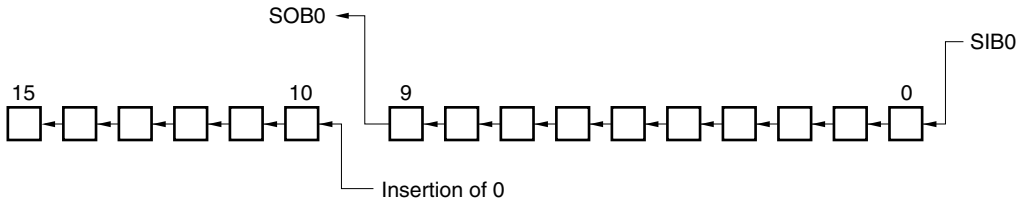
Remark If the number of transfer bits is other than 8 or 16, prepare and use data stuffed from the LSB of the CB0TX and CB0RX registers.

(a) Transfer data length change function

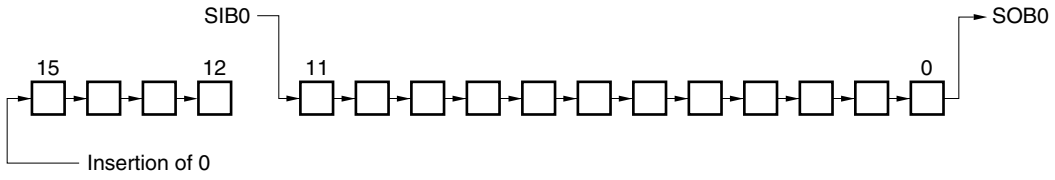
The CSIB0 transfer data length can be set in 1-bit units between 8 and 16 bits using the CB0CTL2.CB0CL3 to CB0CTL2.CB0CL0 bits.

When the transfer bit length is set to a value other than 16 bits, set the data to the CB0TX or CB0RX 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.

(i) Transfer bit length = 10 bits, MSB first



(ii) Transfer bit length = 12 bits, LSB first



(4) CSIB0 status register (CB0STR)

CB0STR is an 8-bit register that displays the CSIB0 status.

This register can be read or written in 8-bit or 1-bit units, but the CB0TSF flag is read-only.

Reset sets this register to 00H.

In addition to reset, the CB0STR register can be initialized by clearing (0) the CB0CTL0.CB0PWR bit.

After reset: 00H R/W Address: FFFFFD03H

	<7>	6	5	4	3	2	1	<0>
CB0STR	CB0TSF	0	0	0	0	0	0	CB0OVE

CB0TSF	Communication status flag
0	Communication stopped
1	Communicating

- During transmission, this register is set when data is prepared in the CB0TX register, and during reception, it is set when a dummy read of the CB0RX register is performed.
  When transfer ends, this flag is cleared to 0 at the last edge of the clock.

CB0OVE	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 CB0RX register, upon end of the receive operation. The CB0OVE flag displays the overrun error occurrence status in this case.
- The CB0OVE flag is cleared by writing 0 to it. It cannot be set even by writing 1 to it.

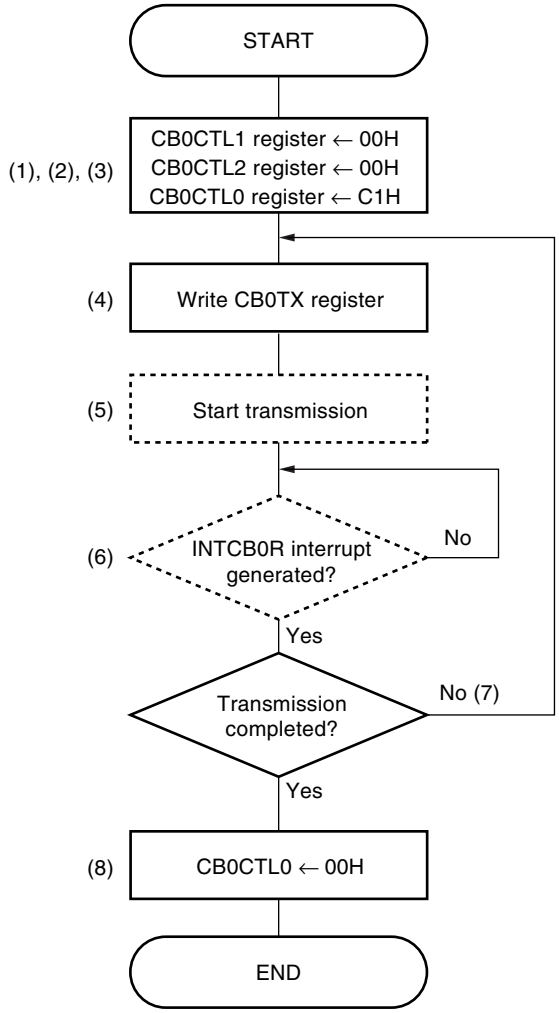
Caution In single transfer mode, writing to the CB0TX register with the CB0TSF bit set to 1 is ignored. This has no influence on the operation during transfer.
Use the continuous transfer mode, not the single transfer mode.

13.4 Operation

13.4.1 Single transfer mode (master mode, transmission mode)

MSB first (CB0CTL0.CB0DIR bit = 0), communication type 1 (CB0CTL1.CB0CKP and CB0CTL1.CB0DAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/2 (CB0CTL1.CB0CKS2 to CB0CTL1.CB0CKS0 bits = 000), transfer data length = 8 bits (CB0CTL2.CB0CL3 to CB0CTL2.CB0CL0 bits = 0000)

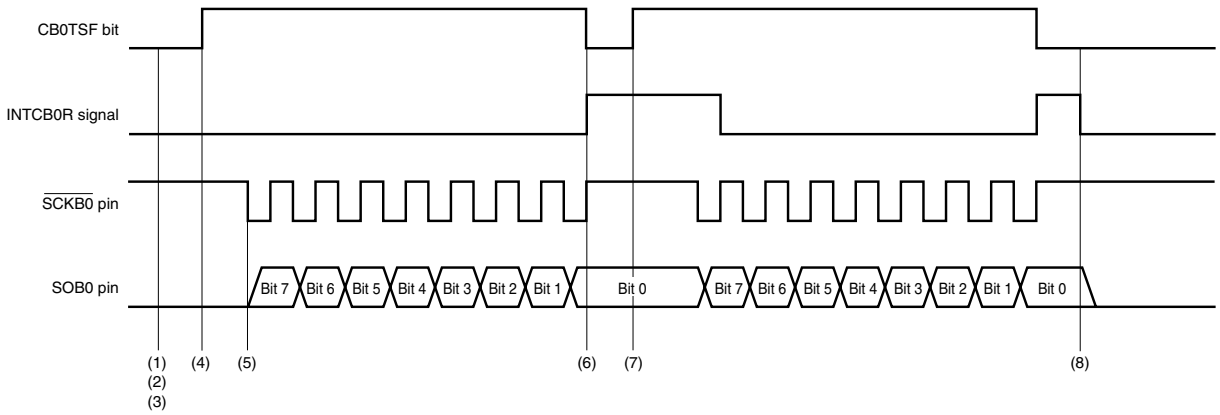
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 00H to the CB0CTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/2, and master mode.

(2) Write 00H to the CB0CTL2 register, and set the transfer data length to 8 bits.

(3) Write C1H to the CB0CTL0 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 CB0STR.CB0TSF bit is set to 1 by writing the transmit data to the CB0TX register, and transmission is started.

(5) When transmission is started, output the serial clock to the $\overline{\text{SCKB0}}$ pin, and output the transmit data from the SOB0 pin in synchronization with the serial clock.

(6) When transmission of the transfer data length set with the CB0CTL2 register is completed, stop the serial clock output and transmit data output, generate the reception end interrupt request signal (INTCB0R) at the last edge of the serial clock, and clear the CB0TSF bit to 0.

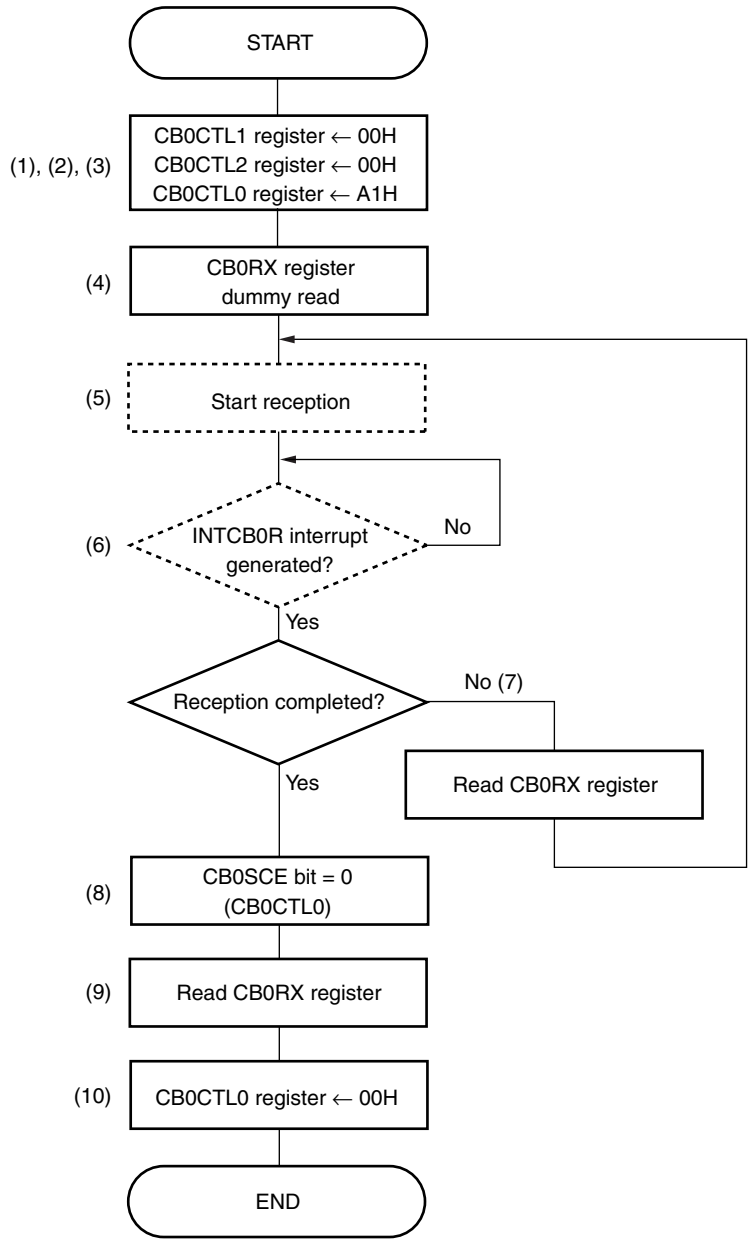
(7) To continue transmission, start the next transmission by writing the transmit data to the CB0TX register again after the INTCB0R signal is generated.

(8) To end transmission, write the CB0CTL0.CB0PWR bit = 0 and the CB0CTL0.CB0TXE bit = 0.

13.4.2 Single transfer mode (master mode, reception mode)

MSB first (CB0CTL0.CB0DIR bit = 0), communication type 1 (CB0CTL1.CB0CKP and CB0CTL1.CB0DAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/2 (CB0CTL1.CB0CKS2 to CB0CTL1.CB0CKS0 bits = 000), transfer data length = 8 bits (CB0CTL2.CB0CL3 to CB0CTL2.CB0CL0 bits = 0000)

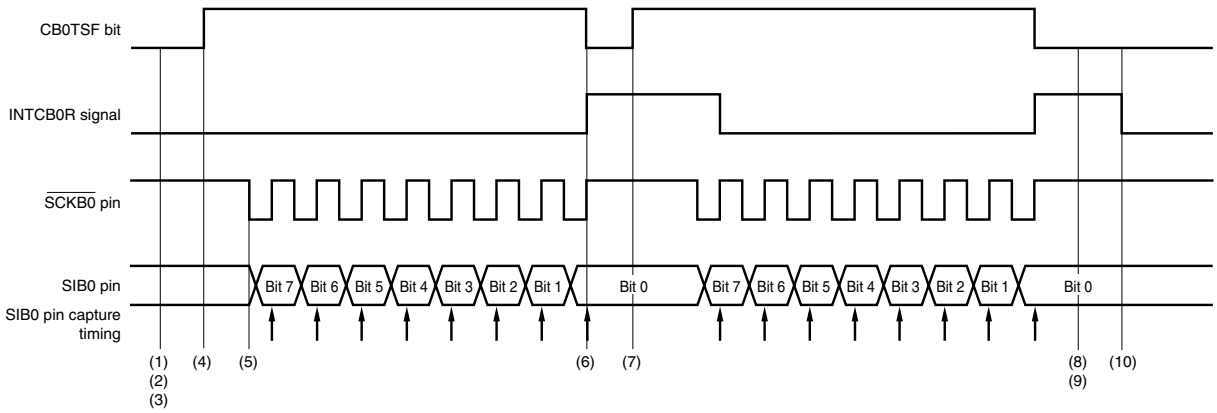
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 00H to the CB0CTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/2, and master mode.

(2) Write 00H to the CB0CTL2 register, and set the transfer data length to 8 bits.

(3) Write A1H to the CB0CTL0 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 CB0STR.CB0TSF bit is set to 1 by performing a dummy read of the CB0RX register, and reception is started.

(5) When reception is started, output the serial clock to the $\overline{SCKB0}$ pin, and capture the receive data of the SIB0 pin in synchronization with the serial clock.

(6) When reception of the transfer data length set with the CB0CTL2 register is completed, stop the serial clock output and data capturing, generate the reception end interrupt request signal (INTCB0R) at the last edge of the serial clock, and clear the CB0TSF bit to 0.

(7) To continue reception, read the CB0RX register with the CB0CTL0.CB0SCE bit = 1 remained after the INTCB0R signal is generated.

(8) To read the CB0RX register without starting the next reception, write the CB0SCE bit = 0.

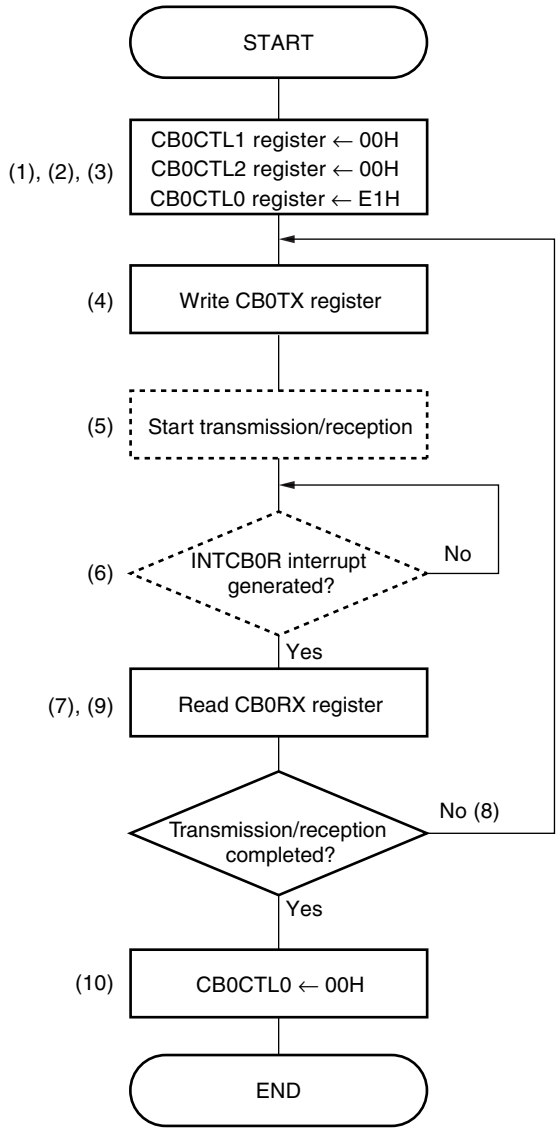
(9) Read the CB0RX register.

(10) To end reception, write the CB0CTL0.CB0PWR bit = 0 and the CB0CTL0.CB0RXE bit = 0.

13.4.3 Single transfer mode (master mode, transmission/reception mode)

MSB first (CB0CTL0.CB0DIR bit = 0), communication type 1 (CB0CTL1.CB0CKP and CB0CTL1.CB0DAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/2 (CB0CTL1.CB0CKS2 to CB0CTL1.CB0CKS0 bits = 000), transfer data length = 8 bits (CB0CTL2.CB0CL3 to CB0CTL2.CB0CL0 bits = 0000)

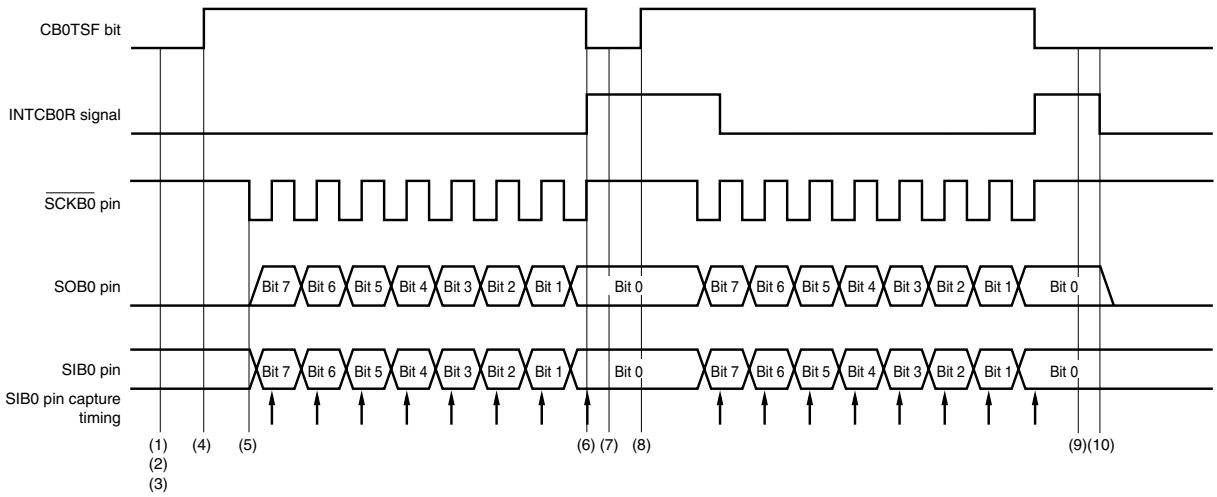
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 00H to the CB0CTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/2, and master mode.

(2) Write 00H to the CB0CTL2 register, and set the transfer data length to 8 bits.

(3) Write E1H to the CB0CTL0 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 CB0STR.CB0TSF bit is set to 1 by writing the transmit data to the CB0TX register, and transmission/reception is started.

(5) When transmission/reception is started, output the serial clock to the $\overline{SCKB0}$ pin, output the transmit data to the SOB0 pin in synchronization with the serial clock, and capture the receive data of the SIB0 pin.

(6) When transmission/reception of the transfer data length set with the CB0CTL2 register is completed, stop the serial clock output, transmit data output, and data capturing, generate the reception end interrupt request signal (INTCB0R) at the last edge of the serial clock, and clear the CB0TSF bit to 0.

(7) Read the CB0RX register.

(8) To continue transmission/reception, write the transmit data to the CB0TX register again.

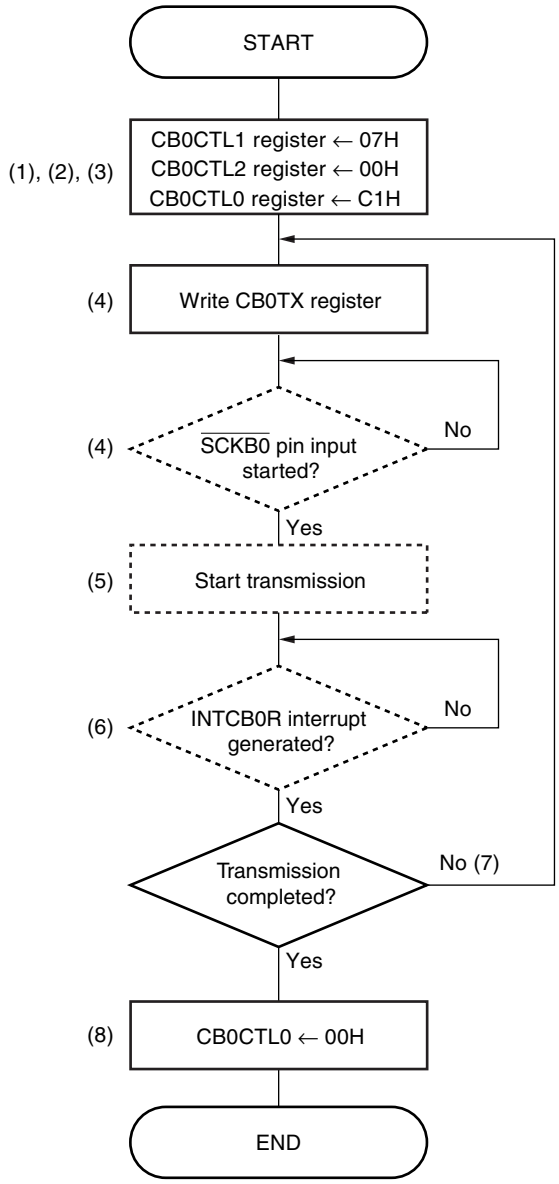
(9) Read the CB0RX register.

(10) To end transmission/reception, write the CB0CTL0.CB0PWR bit = 0, the CB0CTL0.CB0TXE bit = 0, and the CB0CTL0.CB0RXE bit = 0.

13.4.4 Single transfer mode (slave mode, transmission mode)

MSB first (CB0CTL0.CB0DIR bit = 0), communication type 1 (CB0CTL1.CB0CKP and CB0CTL1.CB0DAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{SCKB0}$) (CB0CTL1.CB0CKS2 to CB0CTL1.CB0CKS0 bits = 111), transfer data length = 8 bits (CB0CTL2.CB0CL3 to CB0CTL2.CB0CL0 bits = 0000)

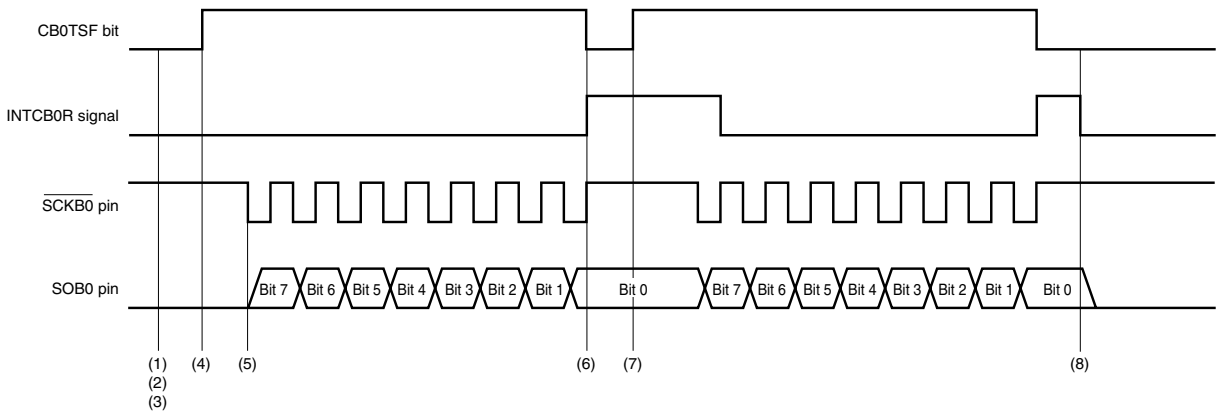
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 07H to the CB0CTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKB0}$), and slave mode.

(2) Write 00H to the CB0CTL2 register, and set the transfer data length to 8 bits.

(3) Write C1H to the CB0CTL0 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 CB0STR.CB0TSF bit is set to 1 by writing the transmit data to the CB0TX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data from the SOB0 pin in synchronization with the serial clock.

(6) When transmission of the transfer data length set with the CB0CTL2 register is completed, stop the serial clock output and transmit data output, generate the reception end interrupt request signal (INTCB0R) at the last edge of the serial clock, and clear the CB0TSF bit to 0.

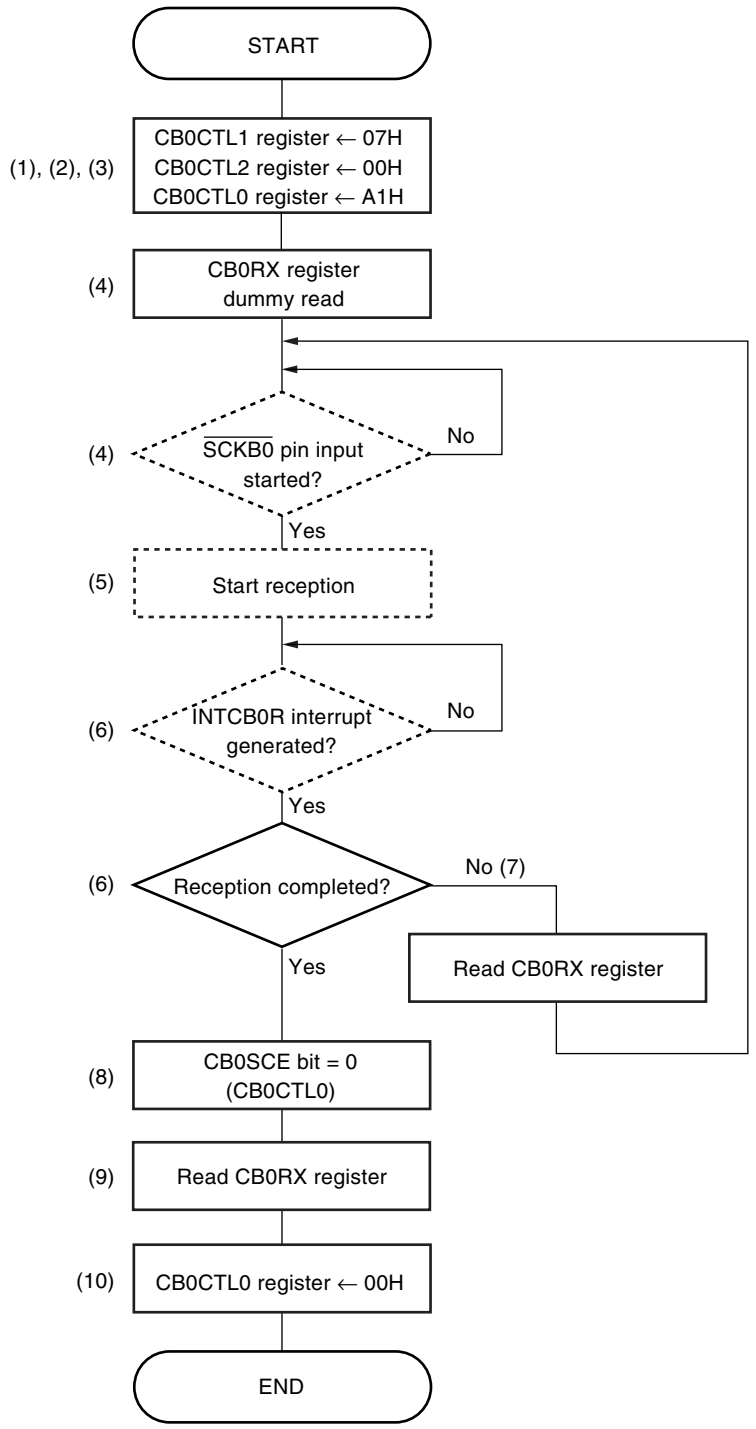
(7) To continue transmission, write the transmit data to the CB0TX register again after the INTCB0R signal is generated, and wait for a serial clock input.

(8) To end transmission, write the CB0CTL0.CB0PWR bit = 0 and the CB0CTL0.CB0TXE bit = 0.

13.4.5 Single transfer mode (slave mode, reception mode)

MSB first (CB0CTL0.CB0DIR bit = 0), communication type 1 (CB0CTL1.CB0CKP and CB0CTL1.CB0DAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{SCKB0}$) (CB0CTL1.CB0CKS2 to CB0CTL1.CB0CKS0 bits = 111), transfer data length = 8 bits (CB0CTL2.CB0CL3 to CB0CTL2.CB0CL0 bits = 0000)

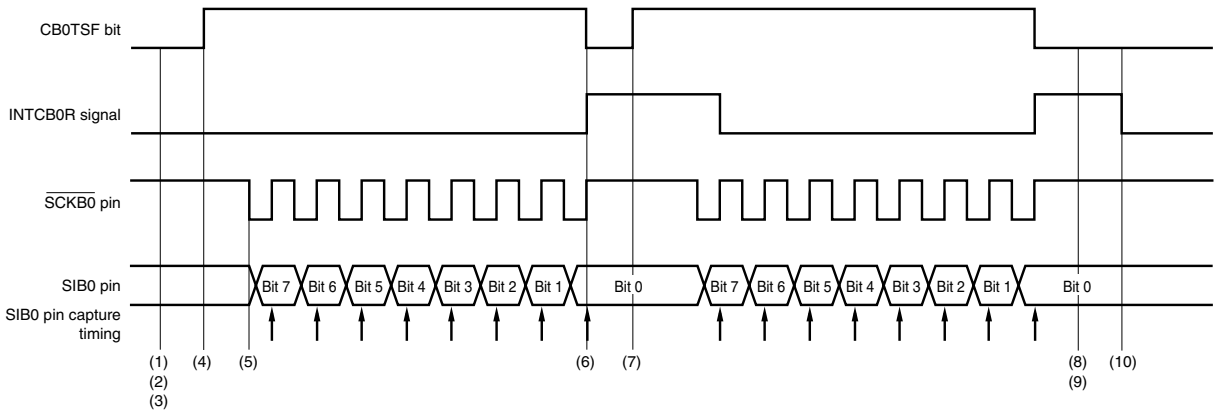
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 07H to the CB0CTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKB0}$), and slave mode.

(2) Write 00H to the CB0CTL2 register, and set the transfer data length to 8 bits.

(3) Write A1H to the CB0CTL0 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 CB0STR.CB0TSF bit is set to 1 by performing a dummy read of the CB0RX register, and the device waits for a serial clock input.

(5) When a serial clock is input, capture the receive data of the SIB0 pin in synchronization with the serial clock.

(6) When reception of the transfer data length set with the CB0CTL2 register is completed, stop the serial clock output and data capturing, generate the reception end interrupt request signal (INTCB0R) at the last edge of the serial clock, and clear the CB0TSF bit to 0.

(7) To continue reception, read the CB0RX register with the CB0CTL0.CB0SCE bit = 1 remained after the INTCB0R signal is generated, and wait for a serial clock input.

(8) To end reception, write the CB0SCE bit = 0.

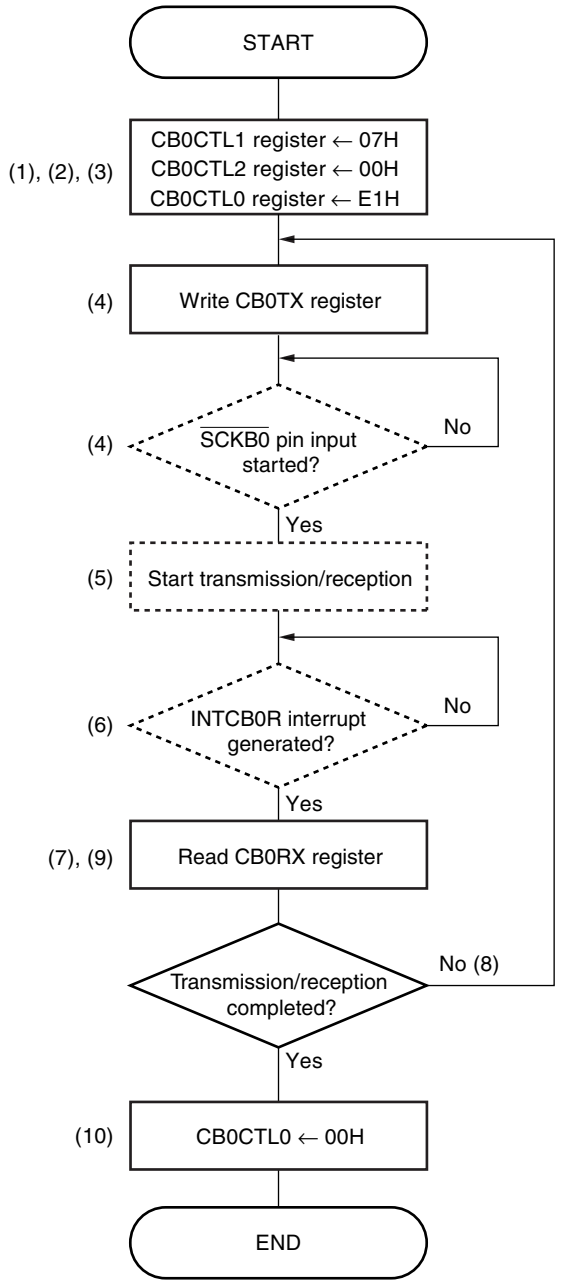
(9) Read the CB0RX register.

(10) To end reception, write the CB0CTL0.CB0PWR bit = 0 and the CB0CTL0.CB0RXE bit = 0.

13.4.6 Single transfer mode (slave mode, transmission/reception mode)

MSB first (CB0CTL0.CB0DIR bit = 0), communication type 1 (CB0CTL1.CB0CKP and CB0CTL1.CB0DAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{SCKB0}$) (CB0CTL1.CB0CKS2 to CB0CTL1.CB0CKS0 bits = 111), transfer data length = 8 bits (CB0CTL2.CB0CL3 to CB0CTL2.CB0CL0 bits = 0000)

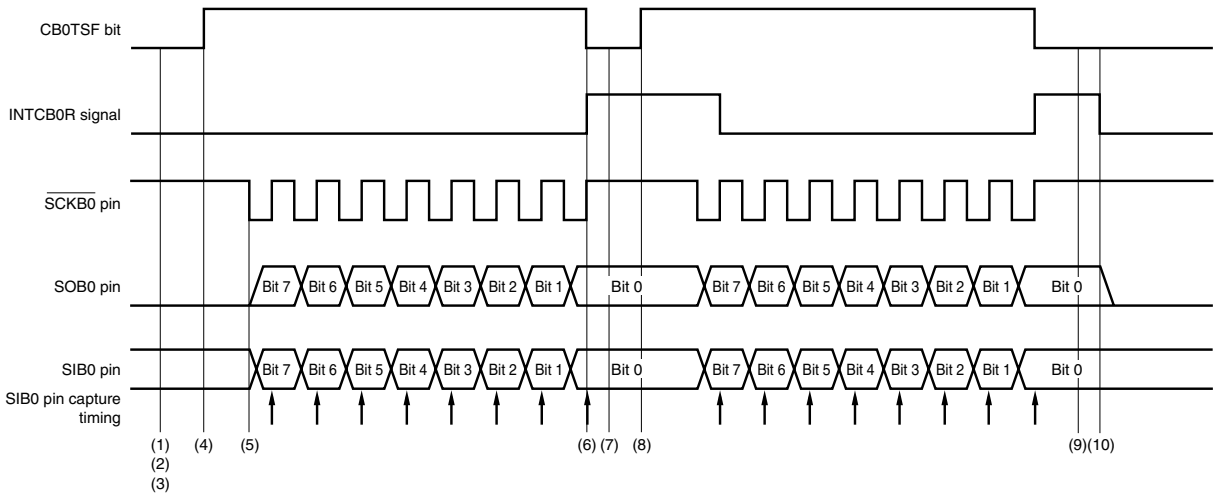
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 07H to the CB0CTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKB0}$), and slave mode.

(2) Write 00H to the CB0CTL2 register, and set the transfer data length to 8 bits.

(3) Write E1H to the CB0CTL0 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 CB0STR.CB0TSF bit is set to 1 by writing the transmit data to the CB0TX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data to the SOB0 pin in synchronization with the serial clock, and capture the receive data of the SIB0 pin.

(6) When transmission/reception of the transfer data length set with the CB0CTL2 register is completed, stop the serial clock output, transmit data output, and data capturing, generate the reception end interrupt request signal (INTCB0R) at the last edge of the serial clock, and clear the CB0TSF bit to 0.

(7) Read the CB0RX register.

(8) To continue transmission/reception, write the transmit data to the CB0TX register again, and wait for a serial clock input.

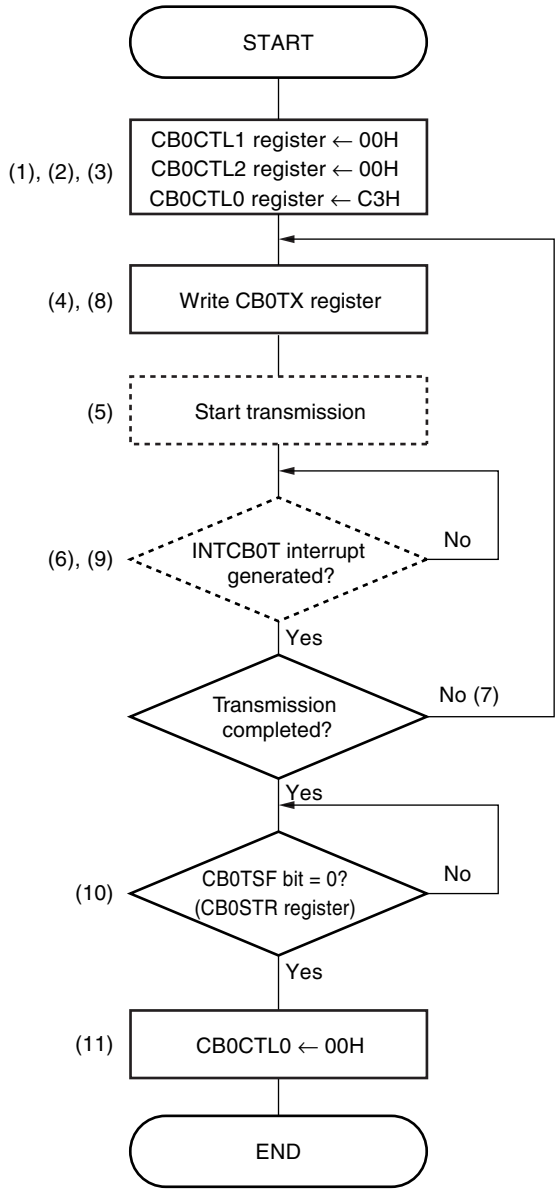
(9) Read the CB0RX register.

(10) To end transmission/reception, write the CB0CTL0.CB0PWR bit = 0, the CB0CTL0.CB0TXE bit = 0, and the CB0CTL0.CB0RXE bit = 0.

13.4.7 Continuous transfer mode (master mode, transmission mode)

MSB first (CB0CTL0.CB0DIR bit = 0), communication type 1 (CB0CTL1.CB0CKP and CB0CTL1.CB0DAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/2 (CB0CTL1.CB0CKS2 to CB0CTL1.CB0CKS0 bits = 000), transfer data length = 8 bits (CB0CTL2.CB0CL3 to CB0CTL2.CB0CL0 bits = 0000)

(1) Operation flow

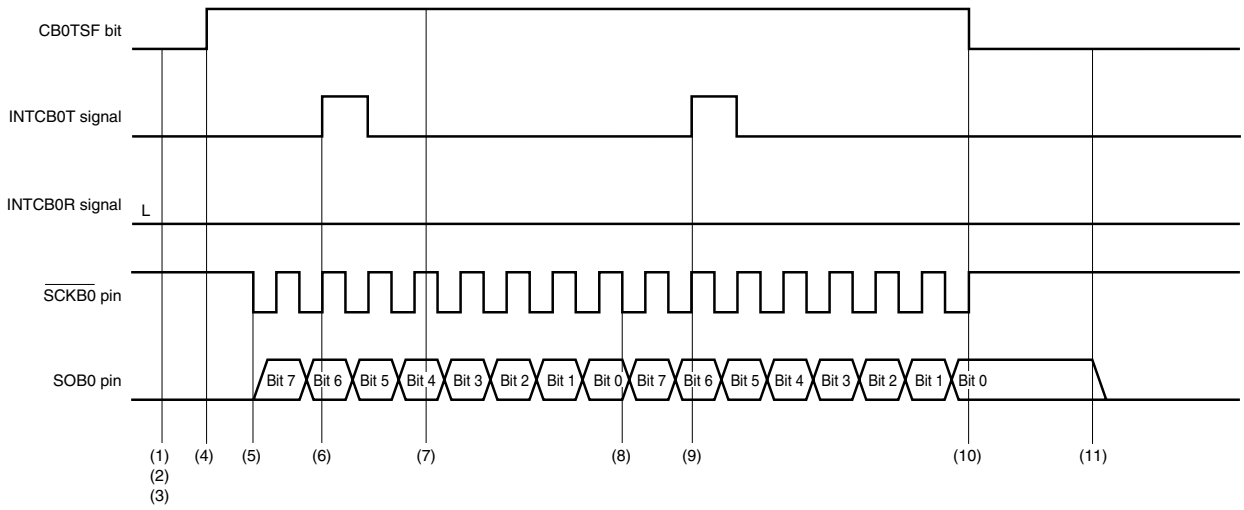


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing

<R>



(1) Write 00H to the CB0CTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/2, and master mode.

(2) Write 00H to the CB0CTL2 register, and set the transfer data length to 8 bits.

(3) Write C3H to the CB0CTL0 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 CB0STR.CB0TSF bit is set to 1 by writing the transmit data to the CB0TX register, and transmission is started.

(5) When transmission is started, output the serial clock to the $\overline{SCKB0}$ pin, and output the transmit data from the SOB0 pin in synchronization with the serial clock.

(6) When transfer of the transmit data from the CB0TX register to the shift register is completed and writing to the CB0TX register is enabled, the transmission enable interrupt request signal (INTCB0T) is generated.

(7) To continue transmission, write the transmit data to the CB0TX register again after the INTCB0T signal is generated.

(8) When a new transmit data is written to the CB0TX register before communication completion, the next communication is started following communication completion.

(9) The transfer of the transmit data from the CB0TX register to the shift register is completed and the INTCB0T signal is generated. To end continuous transmission with the current transmission, do not write to the CB0TX register.

(10) When the next transmit data is not written to the CB0TX register before transfer completion, stop the serial clock output to the $\overline{SCKB0}$ pin after transfer completion, and clear the CB0TSF bit to 0.

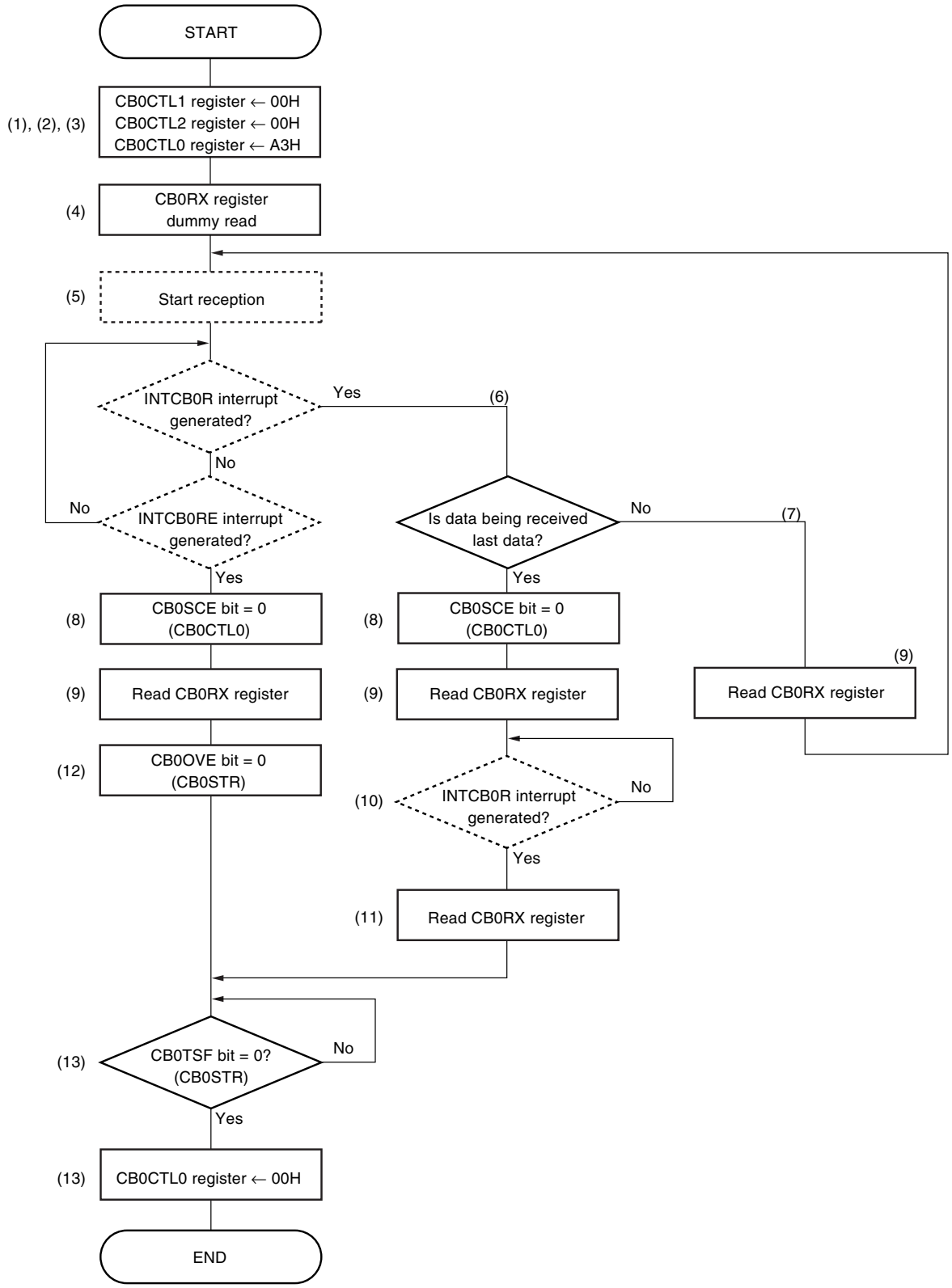
(11) To release the transmission enable status, write the CB0CTL0.CB0PWR bit = 0 and the CB0CTL0.CB0TXE bit = 0 after checking that the CB0TSF bit = 0.

Caution In continuous transmission mode, the reception end interrupt request signal (INTCB0R) is not generated.

13.4.8 Continuous transfer mode (master mode, reception mode)

MSB first (CB0CTL0.CB0DIR bit = 0), communication type 1 (CB0CTL1.CB0CKP and CB0CTL1.CB0DAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/2 (CB0CTL1.CB0CKS2 to CB0CTL1.CB0CKS0 bits = 000), transfer data length = 8 bits (CB0CTL2.CB0CL3 to CB0CTL2.CB0CL0 bits = 0000)

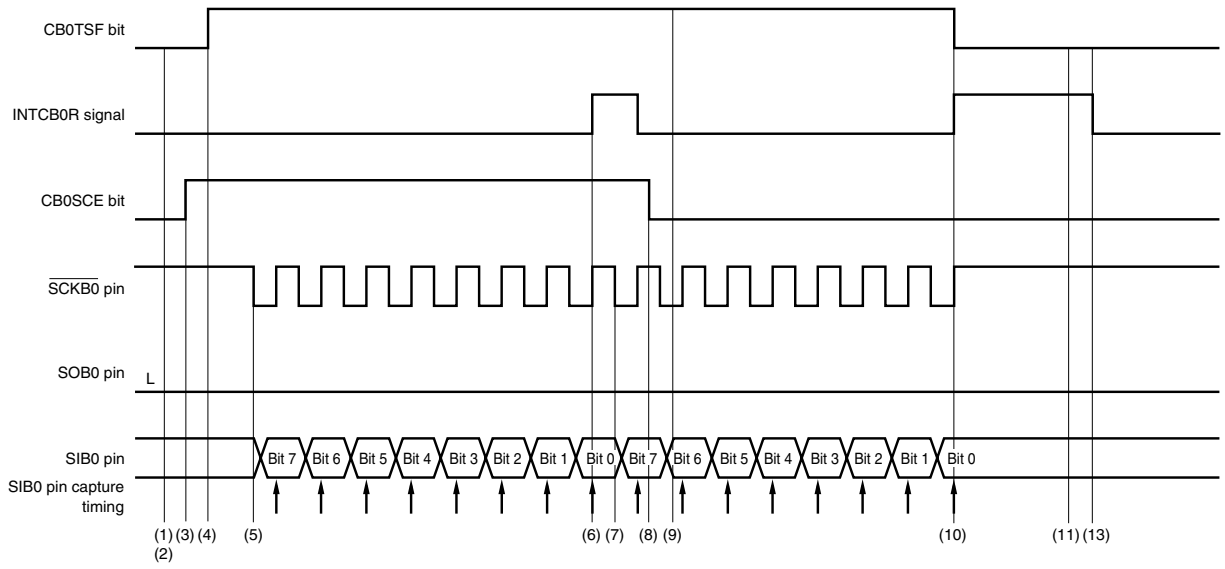
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 00H to the CB0CTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/2, and master mode.

(2) Write 00H to the CB0CTL2 register, and set the transfer data length to 8 bits.

(3) Write A3H to the CB0CTL0 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 CB0STR.CB0TSF bit is set to 1 by performing a dummy read of the CB0RX register, and reception is started.

(5) When reception is started, output the serial clock to the $\overline{\text{SCKB0}}$ pin, and capture the receive data of the SIB0 pin in synchronization with the serial clock.

(6) When reception is completed, the reception end interrupt request signal (INTCB0R) is generated, and reading of the CB0RX register is enabled.

(7) When the CB0CTL0.CB0SCE bit = 1 upon communication completion, the next communication is started following communication completion.

(8) To end continuous reception with the current reception, write the CB0SCE bit = 0.

(9) Read the CB0RX register.

(10) When reception is completed, the INTCB0R signal is generated, and reading of the CB0RX register is enabled. When the CB0SCE bit = 0 is set before communication completion, stop the serial clock output to the $\overline{\text{SCKB0}}$ pin, and clear the CB0TSF bit to 0, to end the receive operation.

(11) Read the CB0RX register.

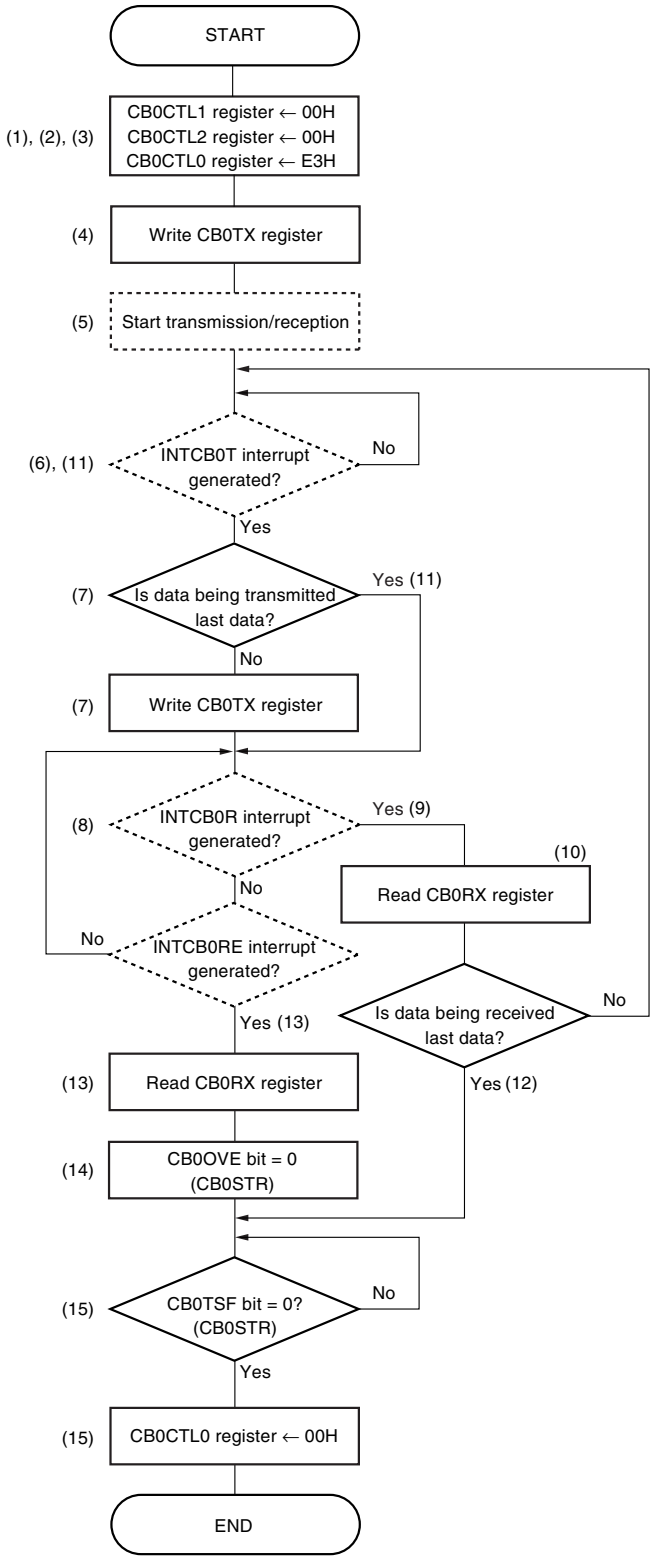
(12) If an overrun error occurs, write the CB0STR.CB0OVE bit = 0, and clear the error flag.

(13) To release the reception enable status, write the CB0CTL0.CB0PWR bit = 0 and the CB0CTL0.CB0RXE bit = 0 after checking that the CB0TSF bit = 0.

13.4.9 Continuous transfer mode (master mode, transmission/reception mode)

MSB first (CB0CTL0.CB0DIR bit = 0), communication type 1 (CB0CTL1.CB0CKP and CB0CTL1.CB0DAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/2 (CB0CTL1.CB0CKS2 to CB0CTL1.CB0CKS0 bits = 000), transfer data length = 8 bits (CB0CTL2.CB0CL3 to CB0CTL2.CB0CL0 bits = 0000)

(1) Operation flow

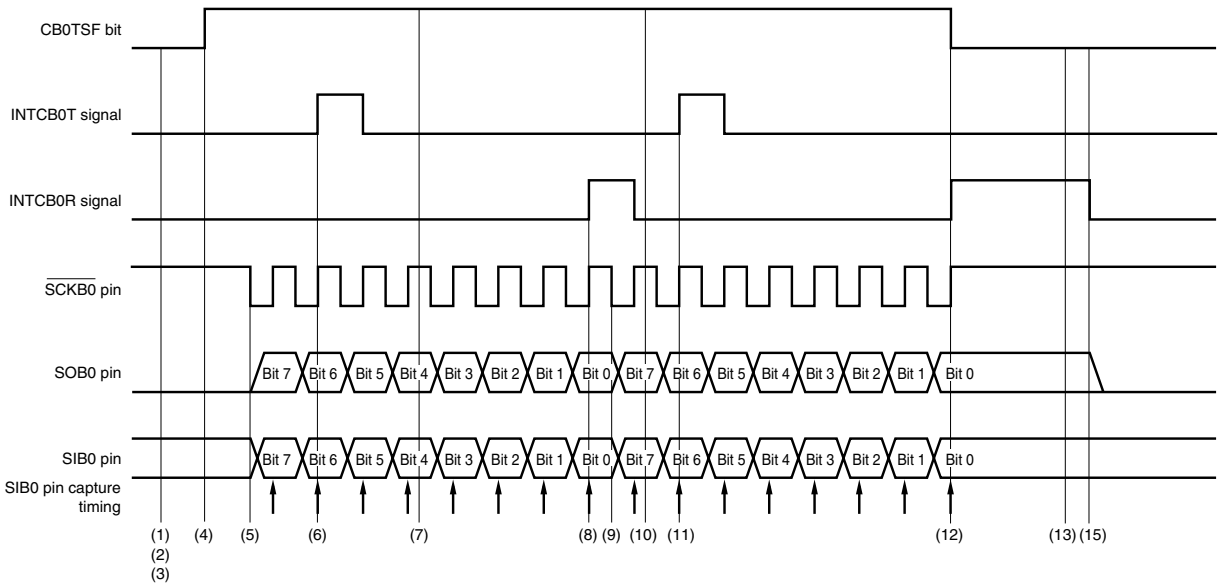


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing

(1/2)



(1) Write 00H to the CB0CTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/2, and master mode.

(2) Write 00H to the CB0CTL2 register, and set the transfer data length to 8 bits.

(3) Write E3H to the CB0CTL0 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 CB0STR.CB0TSF bit is set to 1 by writing the transmit data to the CB0TX register, and transmission/reception is started.

(5) When transmission/reception is started, output the serial clock to the $\overline{\text{SCKB0}}$ pin, output the transmit data to the SOB0 pin in synchronization with the serial clock, and capture the receive data of the SIB0 pin.

(6) When transfer of the transmit data from the CB0TX register to the shift register is completed and writing to the CB0TX register is enabled, the transmission enable interrupt request signal (INTCB0T) is generated.

(7) To continue transmission/reception, write the transmit data to the CB0TX register again after the INTCB0T signal is generated.

(8) When one transmission/reception is completed, the reception end interrupt request signal (INTCB0R) is generated, and reading of the CB0RX register is enabled.

(9) When a new transmit data is written to the CB0TX register before communication completion, the next communication is started following communication completion.

(10) Read the CB0RX register.

(2/2)

(11) The transfer of the transmit data from the CB0TX register to the shift register is completed and the INTCB0T signal is generated. To end continuous transmission/reception with the current transmission/reception, do not write to the CB0TX register.

(12) When the next transmit data is not written to the CB0TX register before transfer completion, stop the serial clock output to the $\overline{\text{SCKB0}}$ pin after transfer completion, and clear the CB0TSF bit to 0.

(13) When the reception error interrupt request signal (INTCB0RE) is generated, read the CB0RX register.

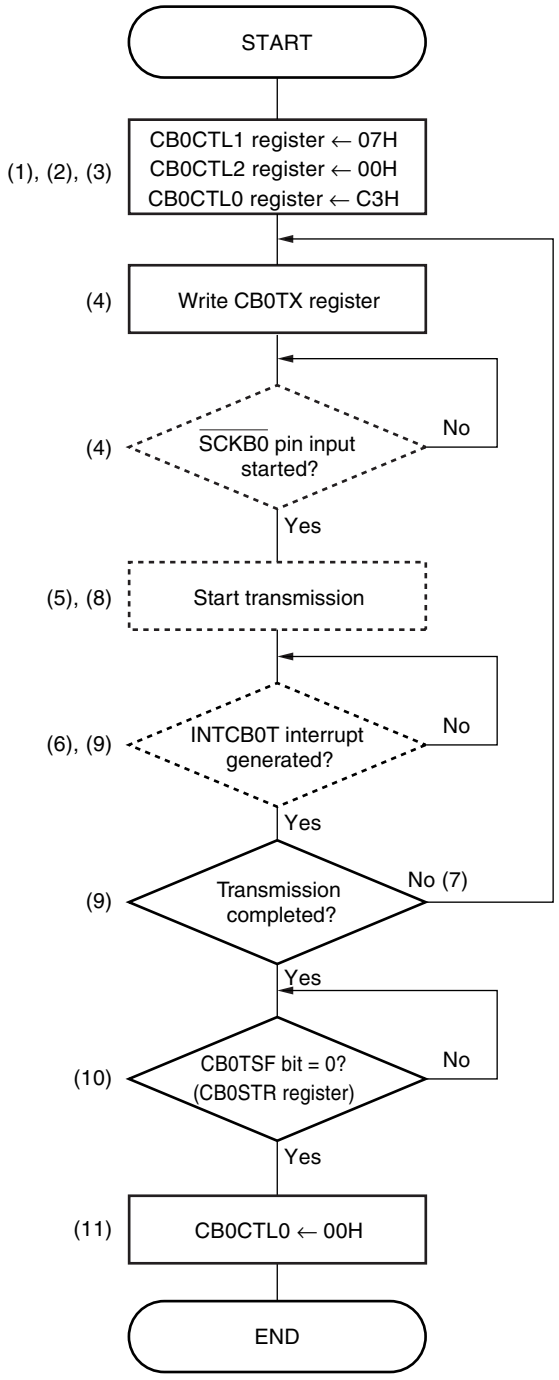
(14) If an overrun error occurs, write the CB0STR.CB0OVE bit = 0, and clear the error flag.

(15) To release the transmission/reception enable status, write the CB0CTL0.CB0PWR bit = 0, the CB0CTL0.CB0TXE bit = 0, and the CB0CTL0.CB0RXE bit = 0 after checking that the CB0TSF bit = 0.

13.4.10 Continuous transfer mode (slave mode, transmission mode)

MSB first (CB0CTL0.CB0DIR bit = 0), communication type 1 (CB0CTL1.CB0CKP and CB0CTL1.CB0DAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{SCKB0}$) (CB0CTL1.CB0CKS2 to CB0CTL1.CB0CKS0 bits = 111), transfer data length = 8 bits (CB0CTL2.CB0CL3 to CB0CTL2.CB0CL0 bits = 0000)

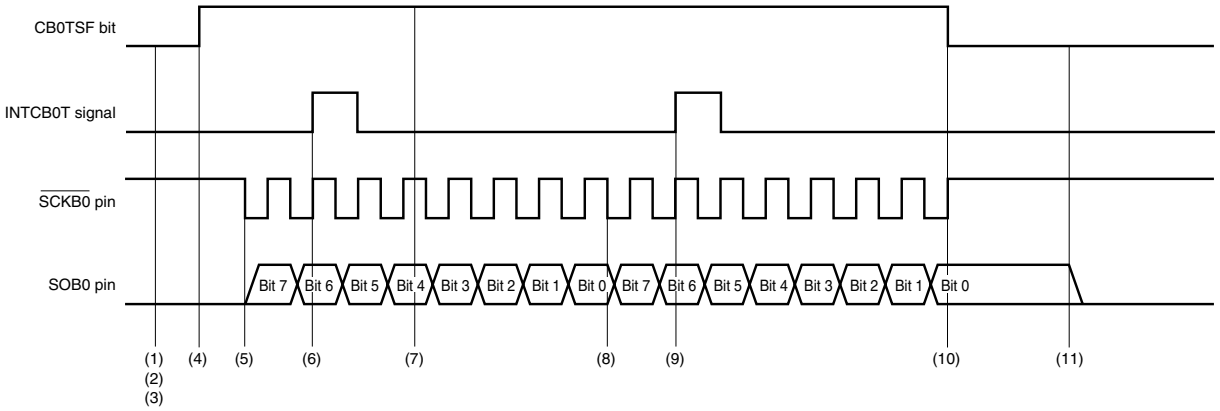
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 07H to the CB0CTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKB0}$), and slave mode.

(2) Write 00H to the CB0CTL2 register, and set the transfer data length to 8 bits.

(3) Write C3H to the CB0CTL0 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 CB0STR.CB0TSF bit is set to 1 by writing the transmit data to the CB0TX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data from the SOB0 pin in synchronization with the serial clock.

(6) When transfer of the transmit data from the CB0TX register to the shift register is completed and writing to the CB0TX register is enabled, the transmission enable interrupt request signal (INTCB0T) is generated.

(7) To continue transmission, write the transmit data to the CB0TX register again after the INTCB0T signal is generated.

(8) When a serial clock is input following completion of the transmission of the transfer data length set with the CB0CTL2 register, continuous transmission is started.

(9) When transfer of the transmit data from the CB0TX register to the shift register is completed and writing to the CB0TX register is enabled, the INTCB0T signal is generated. To end continuous transmission with the current transmission, do not write to the CB0TX register.

(10) When the clock of the transfer data length set with the CB0CTL2 register is input without writing to the CB0TX register, clear the CB0TSF bit to 0 to end transmission.

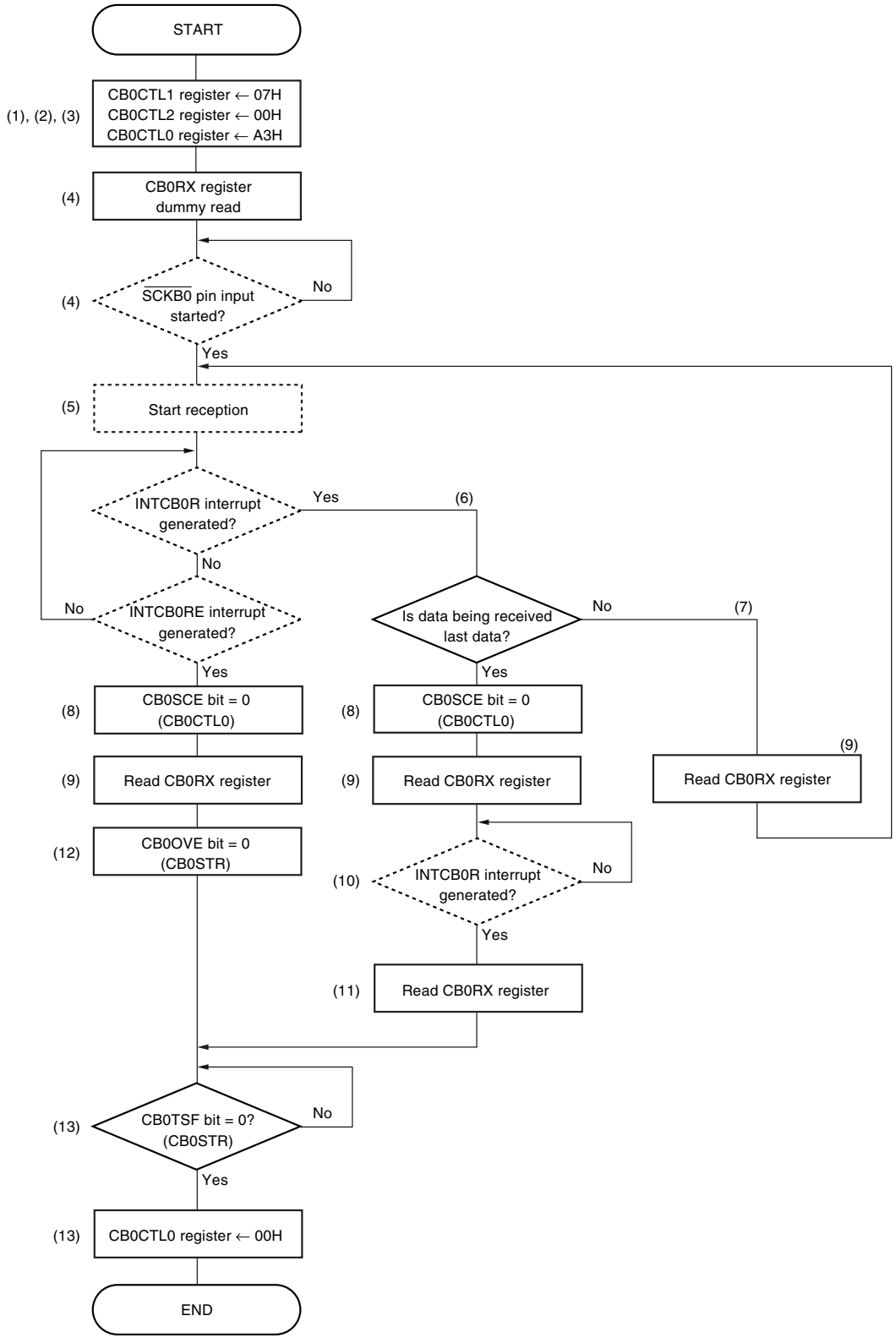
(11) To release the transmission enable status, write the CB0CTL0.CB0PWR bit = 0 and the CB0CTL0.CB0TXE bit = 0 after checking that the CB0TSF bit = 0.

Caution In continuous transmission mode, the reception end interrupt request signal (INTCB0R) is not generated.

13.4.11 Continuous transfer mode (slave mode, reception mode)

MSB first (CB0CTL0.CB0DIR bit = 0), communication type 1 (CB0CTL1.CB0CKP and CB0CTL1.CB0DAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{SCKB0}$) (CB0CTL1.CB0CKS2 to CB0CTL1.CB0CKS0 bits = 111), transfer data length = 8 bits (CB0CTL2.CB0CL3 to CB0CTL2.CB0CL0 bits = 0000)

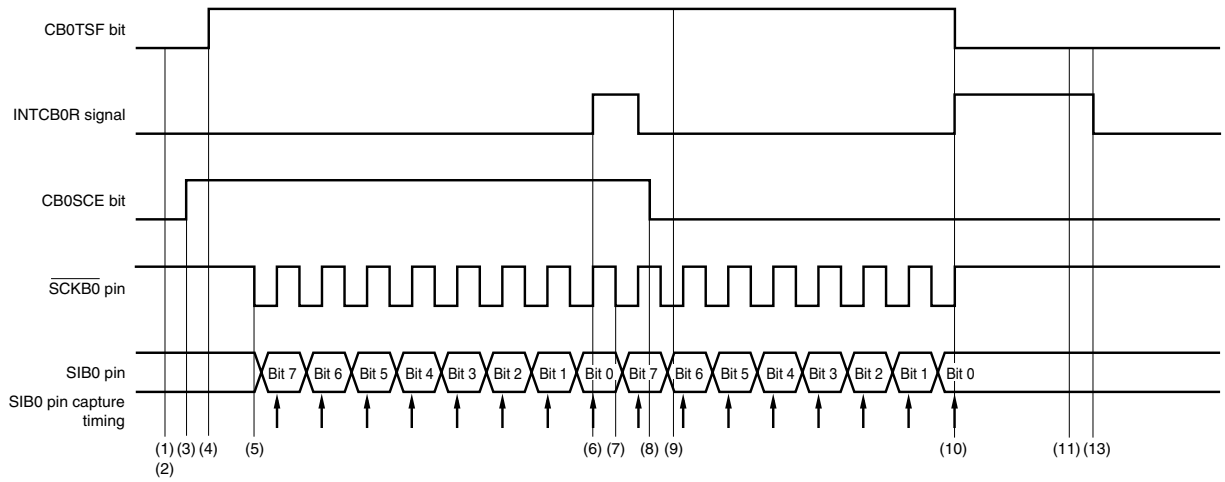
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 07H to the CB0CTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKB0}$), and slave mode.

(2) Write 00H to the CB0CTL2 register, and set the transfer data length to 8 bits.

(3) Write A3H to the CB0CTL0 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 CB0STR.CB0TSF bit is set to 1 by performing a dummy read of the CB0RX register, and the device waits for a serial clock input.

(5) When a serial clock is input, capture the receive data of the SIB0 pin in synchronization with the serial clock.

(6) When reception is completed, the reception end interrupt request signal (INTCB0R) is generated, and reading of the CB0RX register is enabled.

(7) When a serial clock is input in the CB0CTL0.CB0SCE bit = 1 status, continuous reception is started.

(8) To end continuous reception with the current reception, write the CB0SCE bit = 0.

(9) Read the CB0RX register.

(10) When reception is completed, the INTCB0R signal is generated, and reading of the CB0RX register is enabled. When the CB0SCE bit = 0 is set before communication completion, clear the CB0TSF bit to 0 to end the receive operation.

(11) Read the CB0RX register.

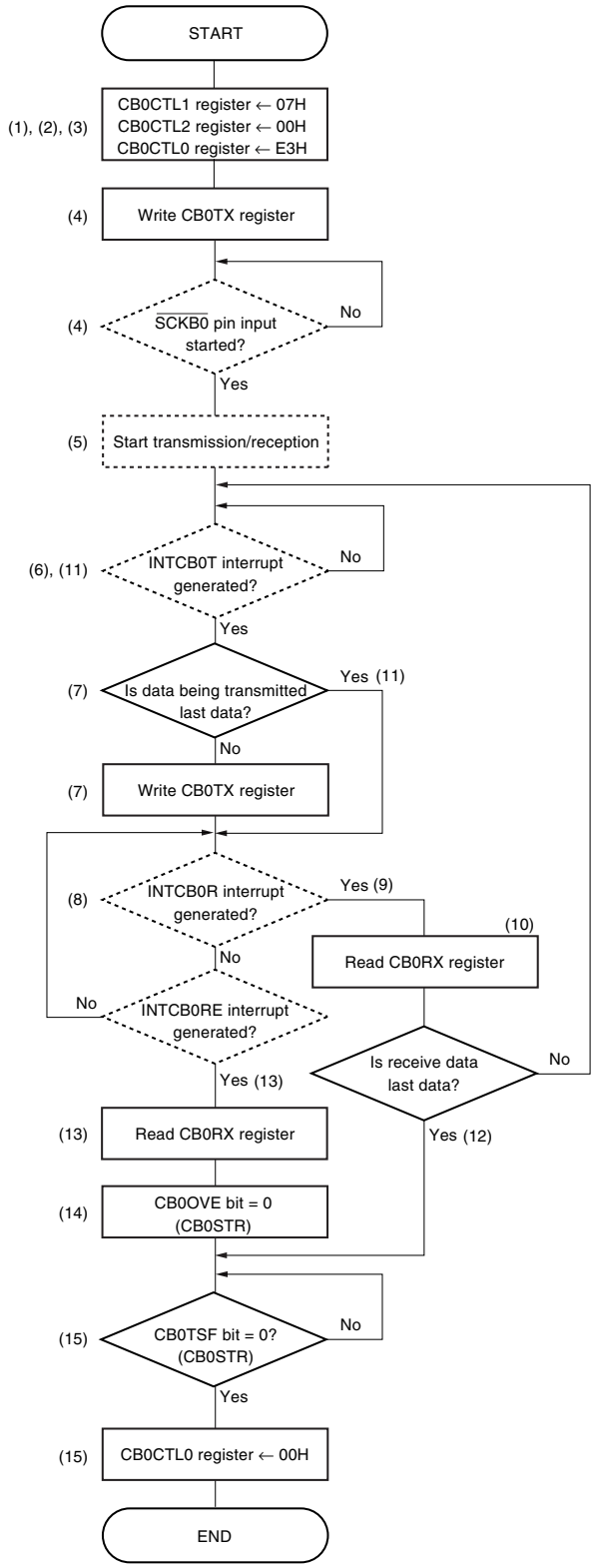
(12) If an overrun error occurs, write the CB0STR.CB0OVE bit = 0, and clear the error flag.

(13) To release the reception enable status, write the CB0CTL0.CB0PWR bit = 0 and the CB0CTL0.CB0RXE bit = 0 after checking that the CB0TSF bit = 0.

13.4.12 Continuous transfer mode (slave mode, transmission/reception mode)

MSB first (CB0CTL0.CB0DIR bit = 0), communication type 1 (CB0CTL1.CB0CKP and CB0CTL1.CB0DAP bits = 00), communication clock (f_{CCLK}) = external clock ($\overline{SCKB0}$) (CB0CTL1.CB0CKS2 to CB0CTL1.CB0CKS0 bits = 111), transfer data length = 8 bits (CB0CTL2.CB0CL3 to CB0CTL2.CB0CL0 bits = 0000)

(1) Operation flow

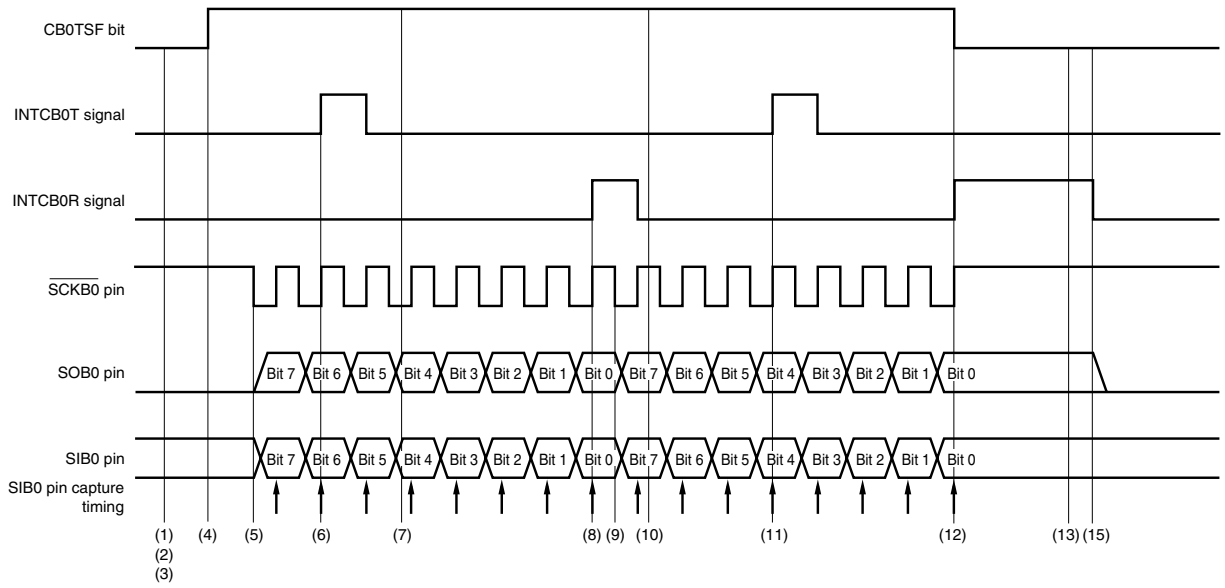


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing

(1/2)



(1) Write 07H to the CB0CTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKB0}$), and slave mode.

(2) Write 00H to the CB0CTL2 register, and set the transfer data length to 8 bits.

(3) Write E3H to the CB0CTL0 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 CB0STR.CB0TSF bit is set to 1 by writing the transmit data to the CB0TX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data to the SOB0 pin in synchronization with the serial clock, and capture the receive data of the SIB0 pin.

(6) When transfer of the transmit data from the CB0TX register to the shift register is completed and writing to the CB0TX register is enabled, the transmission enable interrupt request signal (INTCB0T) is generated.

(7) To continue transmission, write the transmit data to the CB0TX register again after the INTCB0T signal is generated.

(8) When reception of the transfer data length set with the CB0CTL2 register is completed, the reception end interrupt request signal (INTCB0R) is generated, and reading of the CB0RX register is enabled.

(9) When a serial clock is input continuously, continuous transmission/reception is started.

(10) Read the CB0RX register.

(11) When transfer of the transmit data from the CB0TX register to the shift register is completed and writing to the CB0TX register is enabled, the INTCB0T signal is generated. To end continuous transmission/reception with the current transmission/reception, do not write to the CB0TX register.

(2/2)

(12) When the clock of the transfer data length set with the CB0CTL2 register is input without writing to the CB0TX register, the INTCB0R signal is generated. Clear the CB0TSF bit to 0 to end transmission/reception.
(13) When the reception error interrupt request signal (INTCB0RE) is generated, read the CB0RX register.
(14) If an overrun error occurs, write the CB0STR.CB0OVE bit = 0, and clear the error flag.
(15) To release the transmission/reception enable status, write the CB0CTL0.CB0PWR bit = 0, the CB0CTL0.CB0TXE bit = 0, and the CB0CTL0.CB0RXE bit = 0 after checking that the CB0TSF bit = 0.

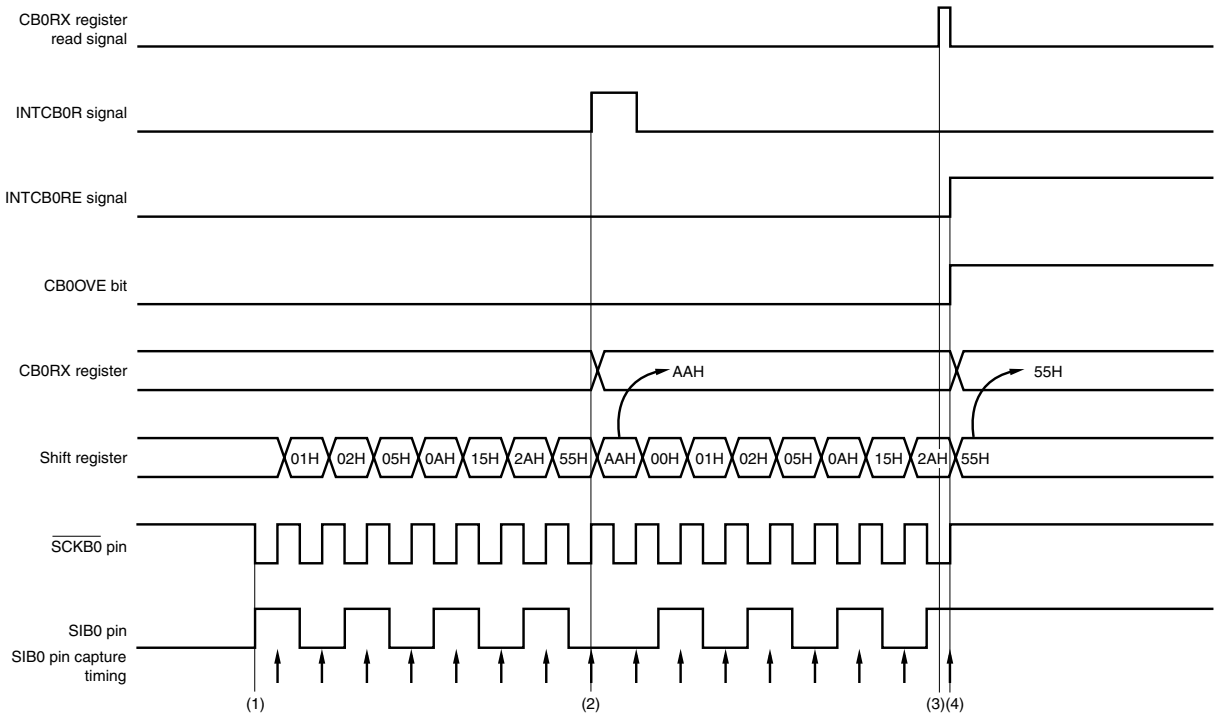
13.4.13 Reception error

When transfer is performed with reception enabled (CB0CTL0.CB0RXE bit = 1) in the continuous transfer mode, the reception error interrupt request signal (INTCB0RE) is generated when the next receive operation is completed before the CB0RX register is read after the reception end interrupt request signal (INTCB0R) is generated, and the overrun error flag (CB0STR.CB0OVE) is set to 1.

Even if an overrun error has occurred, the previous receive data is lost since the CB0RX register is updated. Even if a reception error has occurred, the INTCB0RE signal is generated again upon the next reception completion if the CB0RX register is not read.

To avoid an overrun error, complete reading the CB0RX register until one half clock before sampling the last bit of the next receive data from the INTCB0R signal generation.

(1) Operation timing

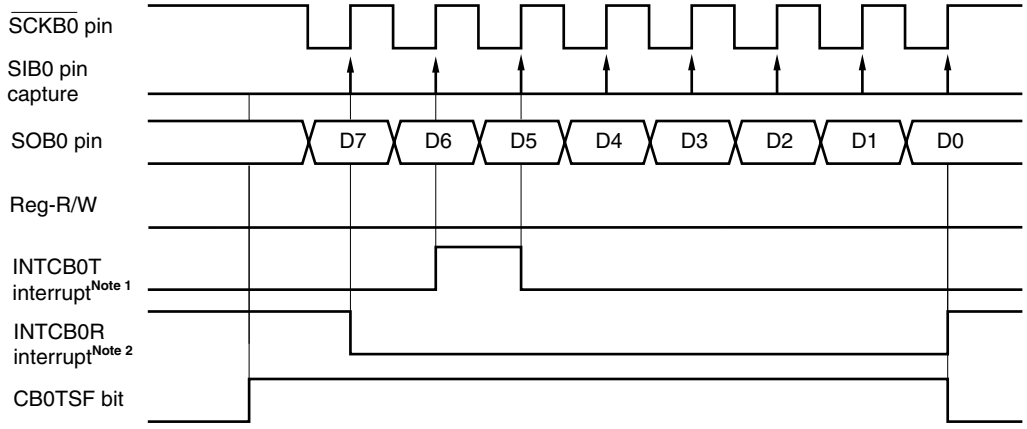


(1) Start continuous transfer.
(2) Completion of the first transfer
(3) The CB0RX register cannot be read until one half clock before the completion of the second transfer.
(4) An overrun error occurs, and the reception error interrupt request signal (INTCB0RE) is generated. The receive data is overwritten.

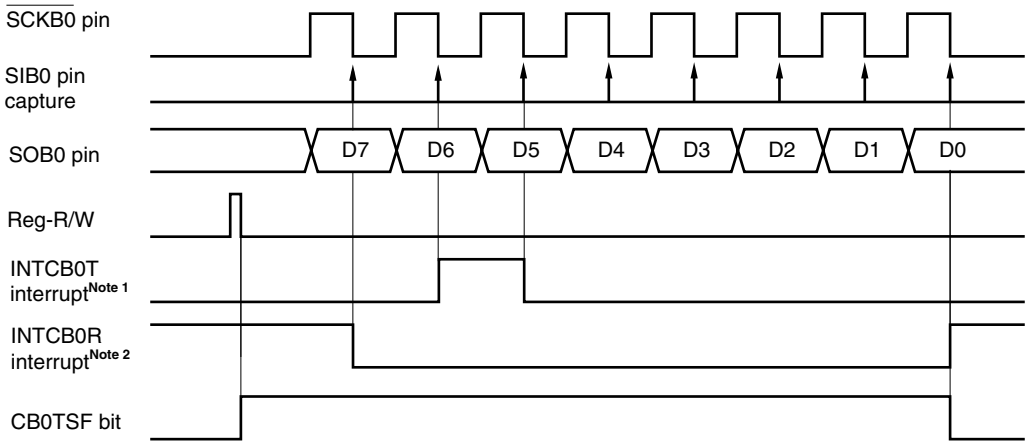
13.4.14 Clock timing

(1/2)

(i) Communication type 1 (CB0CKP and CB0DAP bits = 00)



(ii) Communication type 3 (CB0CKP and CB0DAP bits = 10)



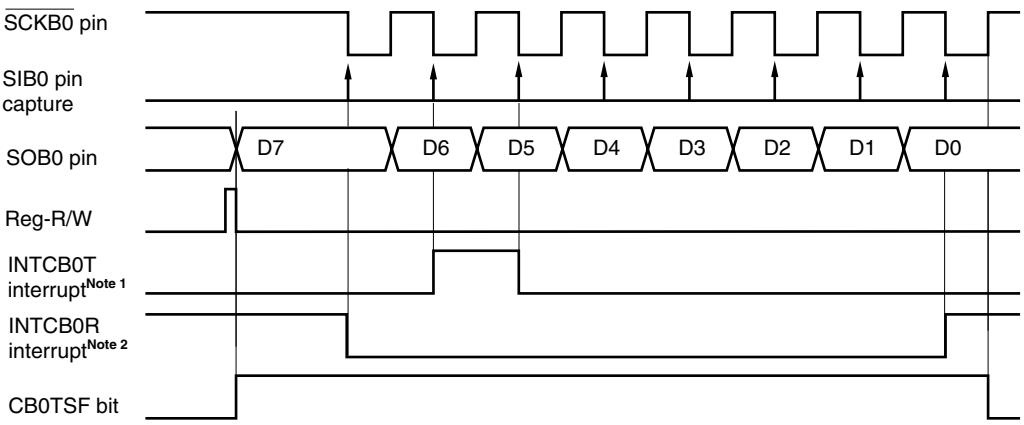
Notes 1. The INTCB0T interrupt is set when the data written to the CB0TX register is transferred to the data shift register in the continuous transmission or continuous transmission/reception mode. In the single transmission or single transmission/reception mode, the INTCB0T interrupt request signal is not generated, but the INTCB0R interrupt request signal is generated upon end of communication.

2. The INTCB0R interrupt occurs if reception is correctly ended and receive data is ready in the CB0RX register while reception is enabled. In the single mode, the INTCB0R interrupt request signal is generated even in the transmission mode, upon end of communication.

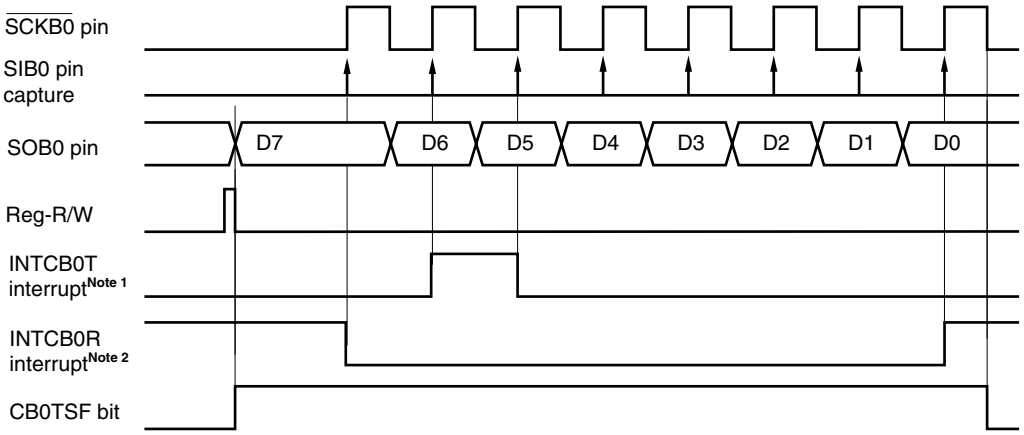
Caution In single transfer mode, writing to the CB0TX register with the CB0TSF bit set to 1 is ignored. This has no influence on the operation during transfer.
Use the continuous transfer mode, not the single transfer mode.

(2/2)

(iii) Communication type 2 (CB0CKP and CB0DAP bits = 01)



(iv) Communication type 4 (CB0CKP and CB0DAP bits = 11)



Notes 1. The INTCB0T interrupt is set when the data written to the CB0TX 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 INTCB0T interrupt request signal is not generated, but the INTCB0R interrupt request signal is generated upon end of communication.

2. The INTCB0R interrupt occurs if reception is correctly ended and receive data is ready in the CB0RX register while reception is enabled. In the single mode, the INTCB0R interrupt request signal is generated even in the transmission mode, upon end of communication.

Caution In single transfer mode, writing to the CB0TX register with the CB0TSF bit set to 1 is ignored. This has no influence on the operation during transfer.
Use the continuous transfer mode, not the single transfer mode.

13.5 Output Pins

(1) $\overline{\text{SCKB0}}$ pin

When CSIB0 operation is disabled (CB0CTL0.CB0PWR bit = 0), the $\overline{\text{SCKB0}}$ pin output status is as follows.

CB0CKP	CB0CKS2	CB0CKS1	CB0CKS0	$\overline{\text{SCKB0}}$ 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 $\overline{\text{SCKB0}}$ pin changes if any of the CB0CTL1.CB0CKP and CB0CKS2 to CB0CKS0 bits is rewritten.

(2) SOB0 pin

When CSIB0 operation is disabled (CB0PWR bit = 0), the SOB0 pin output status is as follows.

CB0TXE	CB0DAP	CB0DIR	SOB0 Pin Output
0	×	×	Fixed to low level
1	0	×	SOB0 latch value (low level)
	1	0	CB0TX0 value (MSB)
		1	CB0TX0 value (LSB)

Remarks 1. The SOB0 pin output changes if any of the CB0CTL0.CB0TXE, CB0CTL0.CB0DIR, or CB0CTL1.CB0DAP bit is rewritten.

2. ×: Don't care

CHAPTER 14 INTERRUPT/EXCEPTION PROCESSING FUNCTION

The V850ES/IE2 is provided with a dedicated interrupt controller (INTC) for interrupt servicing and can process a total of 43 interrupt requests.

An interrupt is an event that occurs independently of program execution, and an exception is an event whose occurrence is dependent on program execution.

The V850ES/IE2 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).

14.1 Features

○ Interrupts
- Non-maskable interrupts: 1 source (external: none, internal: 1 source)
- Maskable interrupts
  External: 7 sources, internal: 35 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 below.

Table 14-1. Interrupt Source List (1/3)

Type	Classification	Default Priority	Name	Generating Source	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Reset	Interrupt	–	RESET	$\overline{\text{RESET}}$ pin input	Pin	0000H	00000000H	Undefined	–
				WDT overflow (WDTRES)	WDT				
				Low-voltage detection (LVIRES)	POC/LVI				
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	Invalid instruction code/ DBTRAP instruction	–	0060H	00000060H	nextPC	–
Maskable	Interrupt	0	INTP0	INTP0 pin valid edge input	Pin	0080H	00000080H	nextPC	PIC0
	Interrupt	1	INTP1	INTP1 pin valid edge input	Pin	0090H	00000090H	nextPC	PIC1
	Interrupt	2	INTP2	INTP2 pin valid edge input	Pin	00A0H	000000A0H	nextPC	PIC2
	Interrupt	3	INTP3	INTP3 pin valid edge input	Pin	00B0H	000000B0H	nextPC	PIC3
	Interrupt	4	INTP4	INTP4 pin valid edge input	Pin	00C0H	000000C0H	nextPC	PIC4
	Interrupt	5	INTP5	INTP5 pin valid edge input	Pin	00D0H	000000D0H	nextPC	PIC5
	Interrupt	6	INTP6	INTP6 pin valid edge input	Pin	00E0H	000000E0H	nextPC	PIC6
	Interrupt	7	INTLVI	Low-voltage detection	POC/LVI	00F0H	000000F0H	nextPC	LVIIC
	Interrupt	–	–	Not used	–	–	00000100H	–	–
	Interrupt	–	–	Not used	–	–	00000110H	–	–
	Interrupt	8	INTTQ0OV	TMQ0 overflow	TMQ0	0120H	00000120H	nextPC	TQ0OVIC
	Interrupt	9	INTTQ0CC0	TQ0CCR0 capture input/ compare match	TMQ0	0130H	00000130H	nextPC	TQ0CCIC0
	Interrupt	10	INTTQ0CC1	TQ0CCR1 capture input/ compare match	TMQ0	0140H	00000140H	nextPC	TQ0CCIC1
	Interrupt	11	INTTQ0CC2	TQ0CCR2 capture input/ compare match	TMQ0	0150H	00000150H	nextPC	TQ0CCIC2
	Interrupt	12	INTTQ0CC3	TQ0CCR3 capture input/ compare match	TMQ0	0160H	00000160H	nextPC	TQ0CCIC3
	Interrupt	13	INTTQ1OV	TMQ1 overflow[Note 2]	TMQ1	0170H	00000170H	nextPC	TQ1OVIC
	Interrupt	14	INTTQ1CC0	TQ1CCR0 compare match[Note 3]	TMQ1	0180H	00000180H	nextPC	TQ1CCIC0
	Interrupt	15	INTTQ1CC1	TQ1CCR1 compare match	TMQ1	0190H	00000190H	nextPC	TQ1CCIC1
	Interrupt	16	INTTQ1CC2	TQ1CCR2 compare match	TMQ1	01A0H	000001A0H	nextPC	TQ1CCIC2
	Interrupt	17	INTTQ1CC3	TQ1CCR3 compare match	TMQ1	01B0H	000001B0H	nextPC	TQ1CCIC3
	Interrupt	–	–	Not used	–	–	000001C0H	–	–
	Interrupt	–	–	Not used	–	–	000001D0H	–	–

Notes 1. n is the value between 0 to FH.

2. When TMQ1 is used in the 6-phase PWM output mode, it functions as INTTQ1OV (valley interrupt) from the TMQ1 option (TMQOP1).

3. When TMQ1 is used in the 6-phase PWM output mode, it functions as INTTQ1CC0 (crest interrupt) from the TMQ1 option (TMQOP1).

Table 14-1. Interrupt Source List (2/3)

Type	Classification	Default Priority	Name	Generating Source	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	–	–	Not used	–	–	000001E0H	–	–
	Interrupt	–	–	Not used	–	–	000001F0H	–	–
	Interrupt	–	–	Not used	–	–	00000200H	–	–
	Interrupt	–	–	Not used	–	–	00000210H	–	–
	Interrupt	–	–	Not used	–	–	00000220H	–	–
	Interrupt	–	–	Not used	–	–	00000230H	–	–
	Interrupt	18	INTTP0OV	TMP0 overflow	TMP0	0240H	00000240H	nextPC	TP0OVIC
	Interrupt	19	INTTP0CC0	TP0CCR0 capture input/ compare match	TMP0	0250H	00000250H	nextPC	TP0CCIC0
	Interrupt	20	INTTP0CC1	TP0CCR1 capture input/ compare match	TMP0	0260H	00000260H	nextPC	TP0CCIC1
	Interrupt	21	INTTP1OV	TMP1 overflow	TMP1	0270H	00000270H	nextPC	TP1OVIC
	Interrupt	22	INTTP1CC0	TP1CCR0 compare match	TMP1	0280H	00000280H	nextPC	TP1CCIC0
	Interrupt	23	INTTP1CC1	TP1CCR1 compare match	TMP1	0290H	00000290H	nextPC	TP1CCIC1
	Interrupt	24	INTTP2OV	TMP2 overflow	TMP2	02A0H	000002A0H	nextPC	TP2OVIC
	Interrupt	25	INTTP2CC0	TP2CCR0 capture input/ compare match	TMP2	02B0H	000002B0H	nextPC	TP2CCIC0
	Interrupt	26	INTTP2CC1	TP2CCR1 capture input/ compare match	TMP2	02C0H	000002C0H	nextPC	TP2CCIC1
	Interrupt	27	INTTP3OV	TMP3 overflow	TMP3	02D0H	000002D0H	nextPC	TP3OVIC
	Interrupt	28	INTTP3CC0	TP3CCR0 compare match	TMP3	02E0H	000002E0H	nextPC	TP3CCIC0
	Interrupt	29	INTTP3CC1	TP3CCR1 compare match	TMP3	02F0H	000002F0H	nextPC	TP3CCIC1
	Interrupt	–	–	Not used	–	–	00000300H	–	–
	Interrupt	–	–	Not used	–	–	00000310H	–	–
	Interrupt	–	–	Not used	–	–	00000320H	–	–
	Interrupt	–	–	Not used	–	–	00000330H	–	–
	Interrupt	30	INTUA0RE	UARTA0 receive error	UARTA0	0340H	00000340H	nextPC	UA0REIC
	Interrupt	31	INTUA0R	UARTA0 reception end	UARTA0	0350H	00000350H	nextPC	UA0RIC
	Interrupt	32	INTUA0T	UARTA0 reception enable	UARTA0	0360H	00000360H	nextPC	UA0TIC
	Interrupt	33	INTCB0RE	CSIB0 receive error	CSIB0	0370H	00000370H	nextPC	CB0REIC
	Interrupt	34	INTCB0R	CSIB0 transmission/reception completion	CSIB0	0380H	00000380H	nextPC	CB0RIC
	Interrupt	35	INTCB0T	CSIB0 continuous transmission enable	CSIB0	0390H	00000390H	nextPC	CB0TIC
	Interrupt	36	INTUA1RE	UARTA1 receive error	UARTA1	03A0H	000003A0H	nextPC	UA1REIC
	Interrupt	37	INTUA1R	UARTA1 reception end	UARTA1	03B0H	000003B0H	nextPC	UA1RIC
	Interrupt	38	INTUA1T	UARTA1 transmission enable	UARTA1	03C0H	000003C0H	nextPC	UA1TIC

Table 14-1. Interrupt Source List (3/3)

Type	Classification	Default Priority	Name	Generating Source	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	–	–	Not used	–	–	000003D0H	–	–
	Interrupt	–	–	Not used	–	–	000003E0H	–	–
	Interrupt	–	–	Not used	–	–	000003F0H	–	–
	Interrupt	39	INTAD0	ADC0 conversion completion	ADC0	0400H	00000400H	nextPC	AD0IC
	Interrupt	40	INTAD1	ADC1 conversion completion	ADC1	0410H	00000410H	nextPC	AD1IC
	Interrupt	–	–	Not used	–	–	00000420H	–	–
	Interrupt	41	INTTM0EQ0	TM0CMP0 compare match	TMM0	0430H	00000430H	nextPC	TM0EQIC0

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

Restored PC: The value of the program counter (PC) saved to EIPC, FEPC, or DBPC 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 from which the processing starts following interrupt/exception processing.

2. The execution address of the illegal instruction when an illegal opcode exception occurs is calculated by (Restored PC – 4).

14.2 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 signal of the V850ES/IE2 is the non-maskable interrupt request signal generated by the overflow of the watchdog timer (INTWDT).

INTWDT functions when the WDTM.WDM1 and WDTM.WDM0 bits are set to “01”.

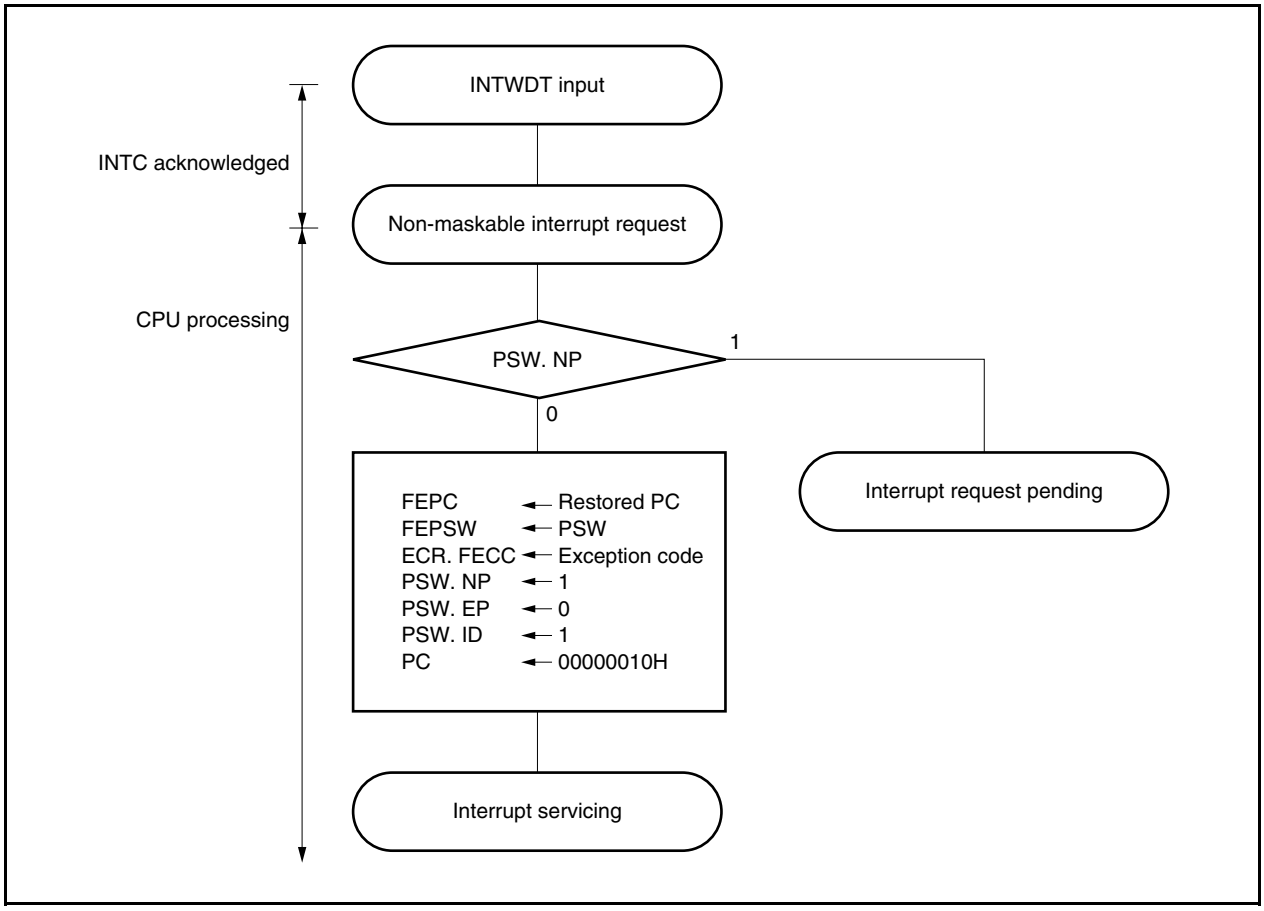
14.2.1 Operation

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 14-1. Non-Maskable Interrupt Servicing



14.2.2 Restore

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

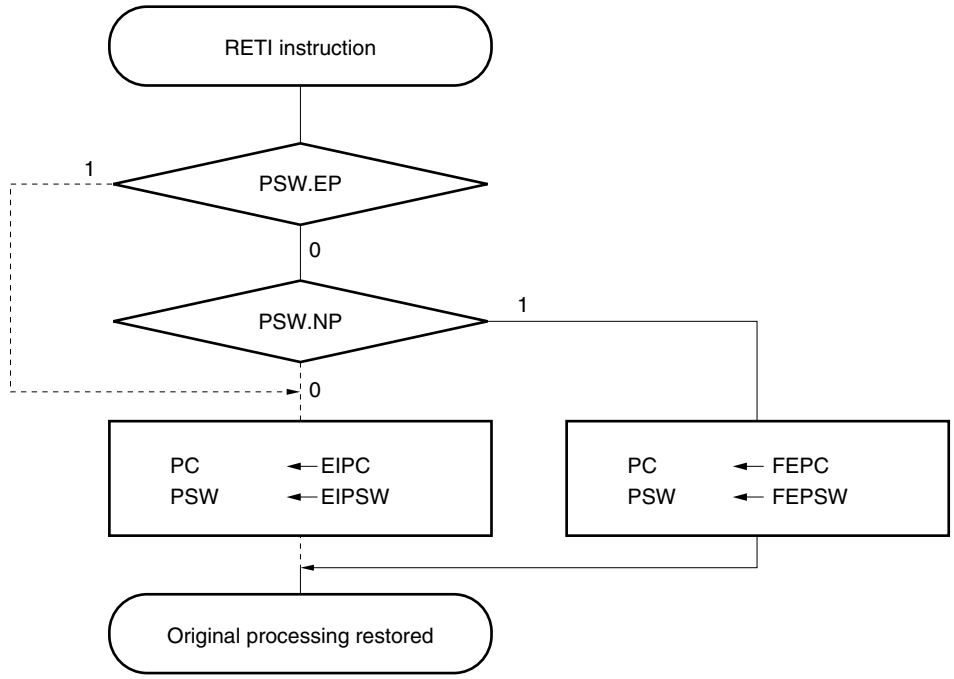
When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

<1> Restores the values of the PC and the PSW from FEPC and FEPSW, respectively, because the PSW.EP bit is 0 and the PSW.NP bit is 1.

<2> Transfers control back to the address of the restored PC and PSW.

The following illustrates how the RETI instruction is processed.

Figure 14-2. 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 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.

14.2.3 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 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

14.3 Maskable Interrupts

Maskable interrupt request signals can be masked by interrupt control registers. The V850ES/IE2 has 42 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.

14.3.1 Operation

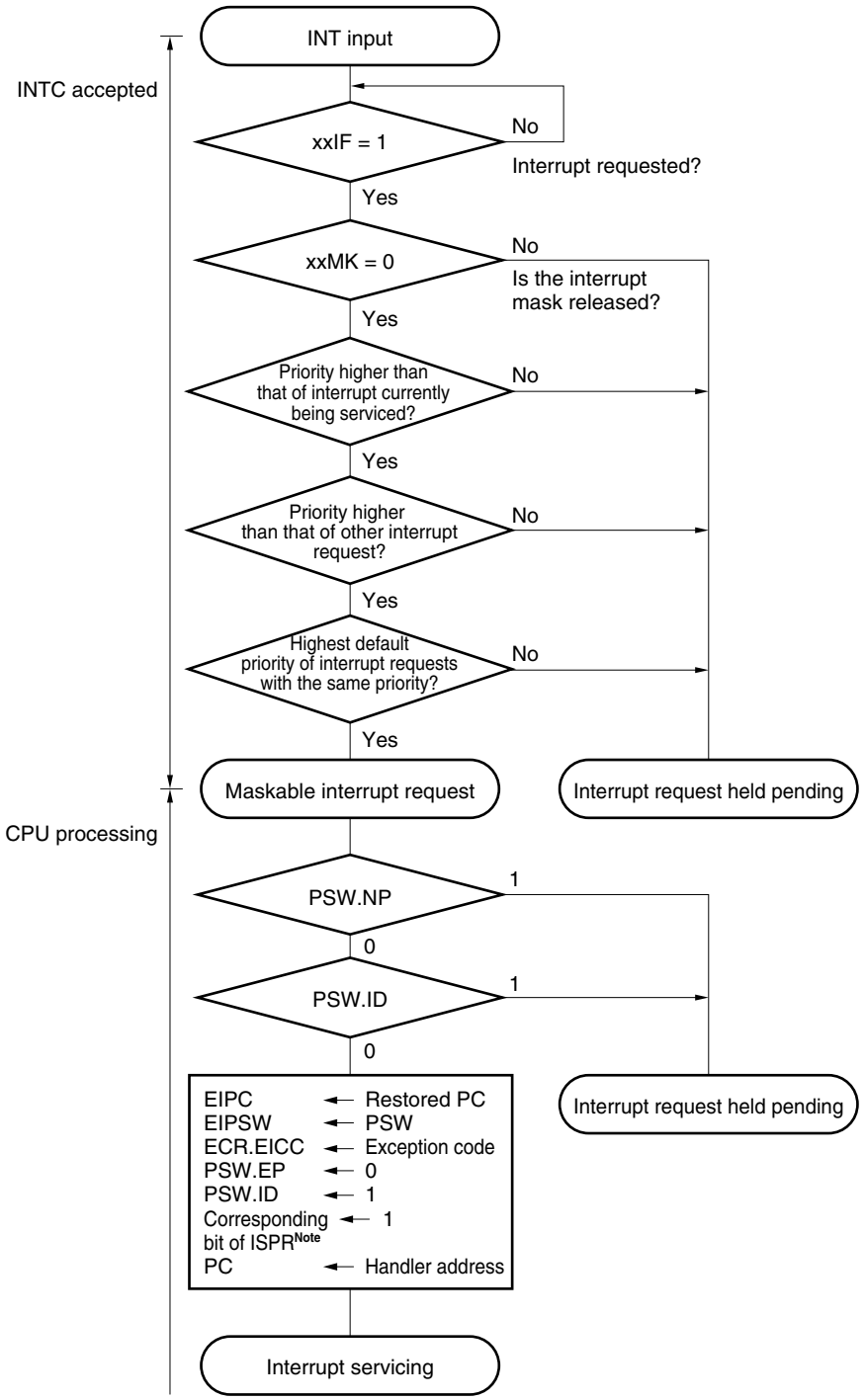
If a maskable interrupt occurs, the CPU performs the following processing, and transfers control to the handler routine.

<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 14-3. Maskable Interrupt Servicing



Note For details of the ISPR register, see **14.3.6 In-service priority register (ISPR)**.

14.3.2 Restore

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

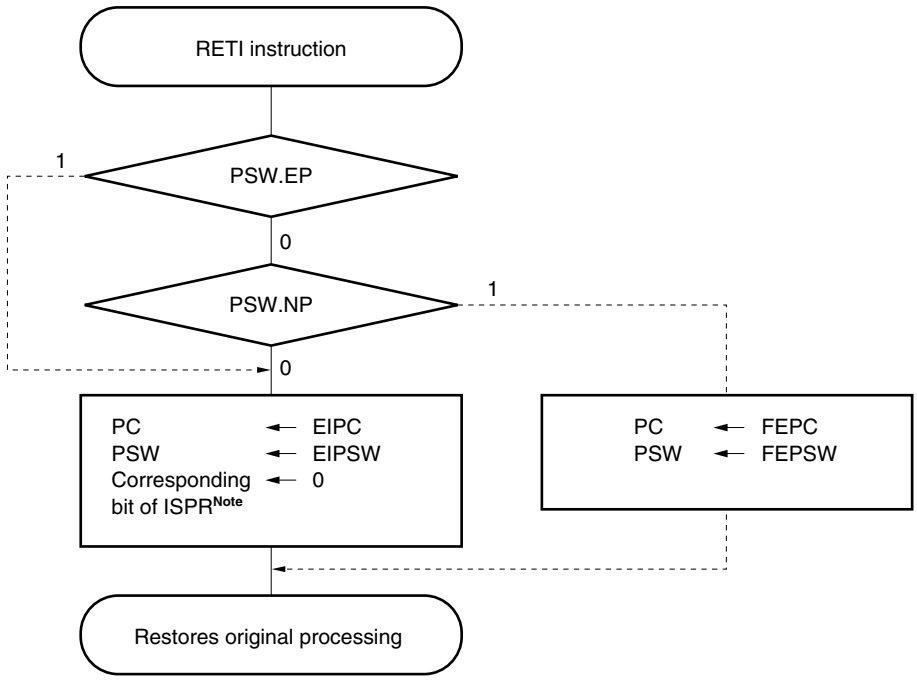
When the RETI instruction is executed, the CPU performs the following steps, and transfers control to the address of the restored PC.

<1> Restores the values of the PC and the PSW from EIPC and EIPSW because the PSW.EP bit is 0 and the PSW.NP bit is 0.

<2> Transfers control back to the address of the restored PC and PSW.

The processing of the RETI instruction is shown below.

Figure 14-4. RETI Instruction Processing



Note For the ISPR register, see **14.3.6 In-service priority register (ISPR)**.

Caution When the EP and NP bits are changed by the LDSR instruction during maskable interrupt servicing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set EP back to 0 and NP back to 0 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

14.3.3 Priorities of maskable interrupts

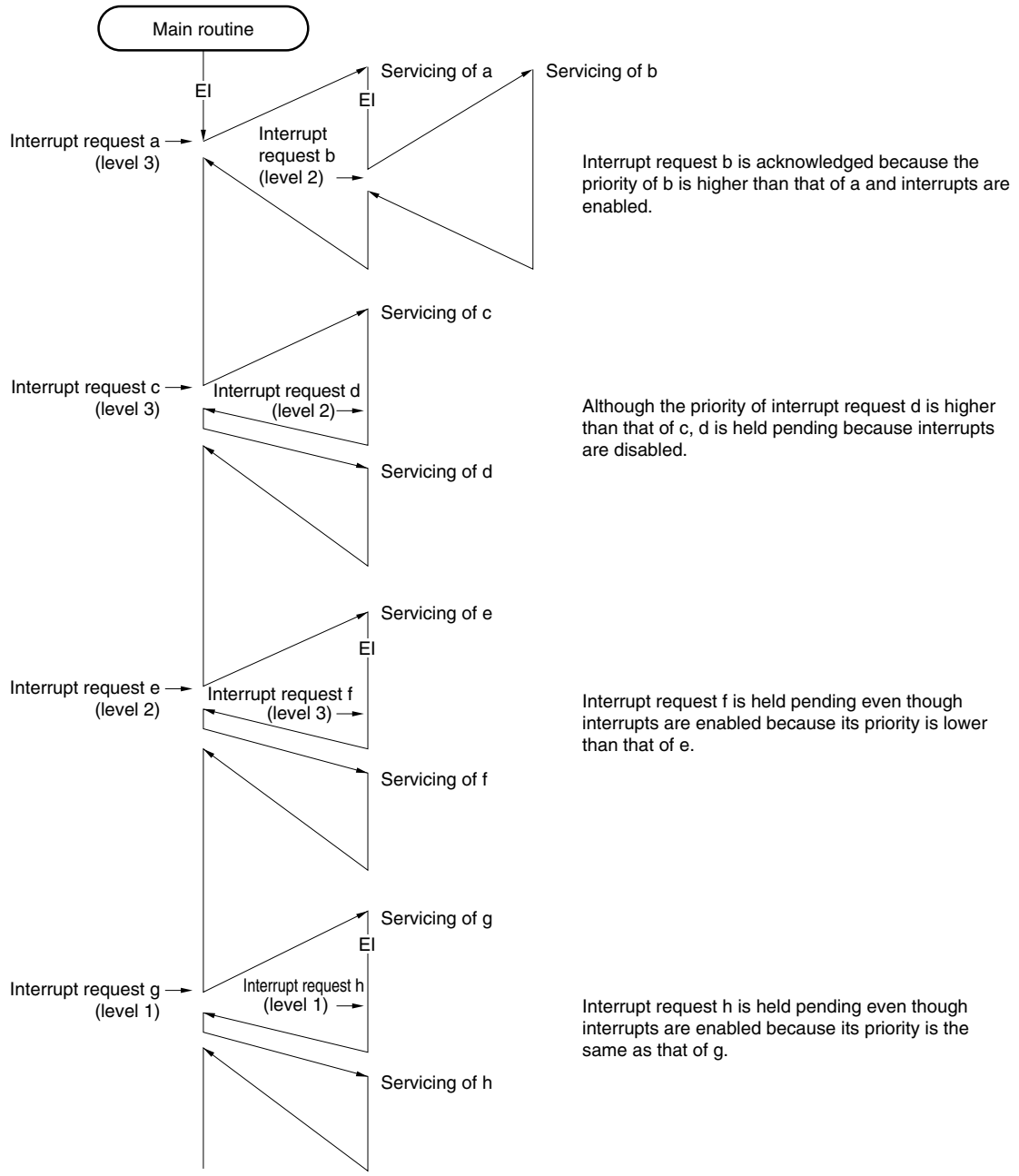
The INTC provides multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced. Multiple interrupts can be controlled by priority levels.

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 14-1 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 14-2**)
n: Peripheral unit number (see **Table 14-2**)

Figure 14-5. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (1/2)

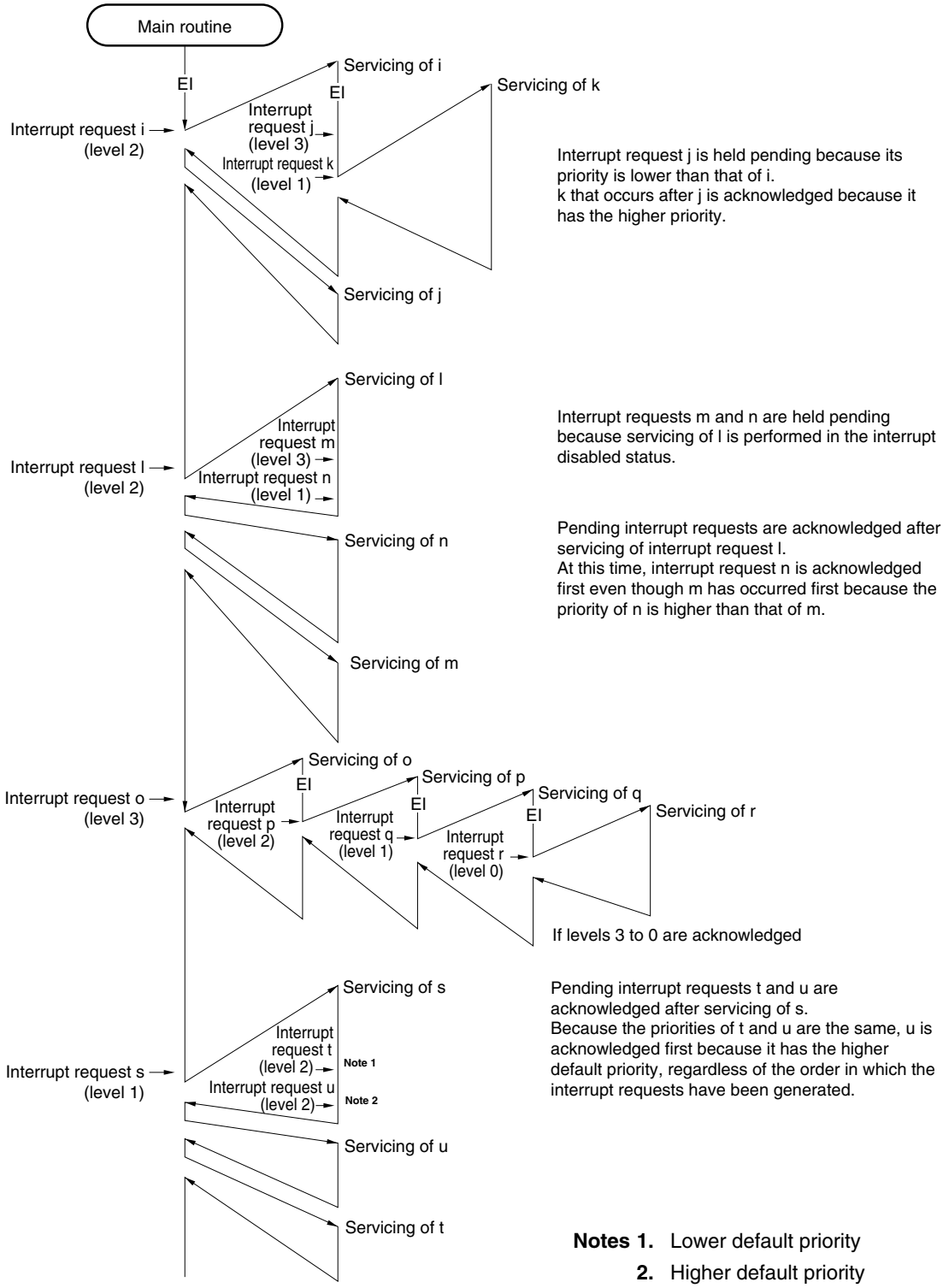


Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Remarks 1. a to u in the figure are the temporary names of interrupt request signals shown for the sake of explanation.

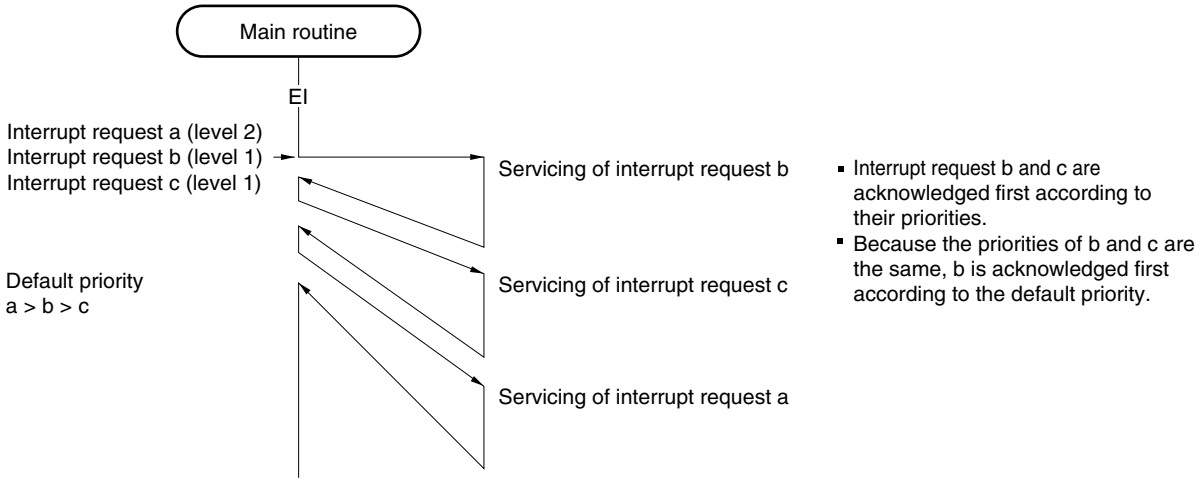
2. The default priority in the figure indicates the relative priority between two interrupt request signals.

Figure 14-5. 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 14-6. 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.

14.3.4 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.

<R> **2. When manipulating the xxICn.xxMKn bit while interrupt requests may occur (including the state in which interrupts are disabled (DI)), be sure to use a bit manipulation instruction or use the xxMKn bit of the IMRm register (m = 0 to 3).**

After reset: 47H R/W Address: FFFFF110H to FFFFF186H

	<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 14-2**)
n: Peripheral unit number (see **Table 14-2**)

The addresses and bits of the interrupt control registers are as follows.

Table 14-2. Addresses and Bits of Interrupt Control Registers (1/2)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF110H	PIC0	PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00
FFFFF112H	PIC1	PIF1	PMK1	0	0	0	PPR12	PPR11	PPR10
FFFFF114H	PIC2	PIF2	PMK2	0	0	0	PPR22	PPR21	PPR20
FFFFF116H	PIC3	PIF3	PMK3	0	0	0	PPR32	PPR31	PPR30
FFFFF118H	PIC4	PIF4	PMK4	0	0	0	PPR42	PPR41	PPR40
FFFFF11AH	PIC5	PIF5	PMK5	0	0	0	PPR52	PPR51	PPR50
FFFFF11CH	PIC6	PIF6	PMK6	0	0	0	PPR62	PPR61	PPR60
FFFFF11EH	LVIIC	LVIIF	LVIMK	0	0	0	LVIPR2	LVIPR1	LVIPR2
FFFFF120H	Not used	–	–	–	–	–	–	–	–
FFFFF122H	Not used	–	–	–	–	–	–	–	–
FFFFF124H	TQ0OVIC	TQ0OVIF	TQ0OVMK	0	0	0	TQ0OVPR2	TQ0OVPR1	TQ0OVPR0
FFFFF126H	TQ0CCIC0	TQ0CCIF0	TQ0CCMK0	0	0	0	TQ0CCPR02	TQ0CCPR01	TQ0CCPR00
FFFFF128H	TQ0CCIC1	TQ0CCIF1	TQ0CCMK1	0	0	0	TQ0CCPR12	TQ0CCPR11	TQ0CCPR10
FFFFF12AH	TQ0CCIC2	TQ0CCIF2	TQ0CCMK2	0	0	0	TQ0CCPR22	TQ0CCPR21	TQ0CCPR20
FFFFF12CH	TQ0CCIC3	TQ0CCIF3	TQ0CCMK3	0	0	0	TQ0CCPR32	TQ0CCPR31	TQ0CCPR30
FFFFF12EH	TQ1OVIC	TQ1OVIF	TQ1OVMK	0	0	0	TQ1OVPR2	TQ1OVPR1	TQ1OVPR0
FFFFF130H	TQ1CCIC0	TQ1CCIF0	TQ1CCMK0	0	0	0	TQ1CCPR02	TQ1CCPR01	TQ1CCPR00
FFFFF132H	TQ1CCIC1	TQ1CCIF1	TQ1CCMK1	0	0	0	TQ1CCPR12	TQ1CCPR11	TQ1CCPR10
FFFFF134H	TQ1CCIC2	TQ1CCIF2	TQ1CCMK2	0	0	0	TQ1CCPR22	TQ1CCPR21	TQ1CCPR20
FFFFF136H	TQ1CCIC3	TQ1CCIF3	TQ1CCMK3	0	0	0	TQ1CCPR32	TQ1CCPR31	TQ1CCPR30
FFFFF138H	Not used	–	–	–	–	–	–	–	–
FFFFF13AH	Not used	–	–	–	–	–	–	–	–
FFFFF13CH	Not used	–	–	–	–	–	–	–	–
FFFFF13EH	Not used	–	–	–	–	–	–	–	–
FFFFF140H	Not used	–	–	–	–	–	–	–	–
FFFFF142H	Not used	–	–	–	–	–	–	–	–
FFFFF144H	Not used	–	–	–	–	–	–	–	–
FFFFF146H	Not used	–	–	–	–	–	–	–	–
FFFFF148H	TP0OVIC	TP0OVIF	TP0OVMK	0	0	0	TP0OVPR2	TP0OVPR1	TP0OVPR0
FFFFF14AH	TP0CCIC0	TP0CCIF0	TP0CCMK0	0	0	0	TP0CCPR02	TP0CCPR01	TP0CCPR00
FFFFF14CH	TP0CCIC1	TP0CCIF1	TP0CCMK1	0	0	0	TP0CCPR12	TP0CCPR11	TP0CCPR10
FFFFF14EH	TP1OVIC	TP1OVIF	TP1OVMK	0	0	0	TP1OVPR2	TP1OVPR1	TP1OVPR0
FFFFF150H	TP1CCIC0	TP1CCIF0	TP1CCMK0	0	0	0	TP1CCPR02	TP1CCPR01	TP1CCPR00
FFFFF152H	TP1CCIC1	TP1CCIF1	TP1CCMK1	0	0	0	TP1CCPR12	TP1CCPR11	TP1CCPR10
FFFFF154H	TP2OVIC	TP2OVIF	TP2OVMK	0	0	0	TP2OVPR2	TP2OVPR1	TP2OVPR0
FFFFF156H	TP2CCIC0	TP2CCIF0	TP2CCMK0	0	0	0	TP2CCPR02	TP2CCPR01	TP2CCPR00
FFFFF158H	TP2CCIC1	TP2CCIF1	TP2CCMK1	0	0	0	TP2CCPR12	TP2CCPR11	TP2CCPR10
FFFFF15AH	TP3OVIC	TP3OVIF	TP3OVMK	0	0	0	TP3OVPR2	TP3OVPR1	TP3OVPR0
FFFFF15CH	TP3CCIC0	TP3CCIF0	TP3CCMK0	0	0	0	TP3CCPR02	TP3CCPR01	TP3CCPR00
FFFFF15EH	TP3CCIC1	TP3CCIF1	TP3CCMK1	0	0	0	TP3CCPR12	TP3CCPR11	TP3CCPR10
FFFFF160H	Not used	–	–	–	–	–	–	–	–

Table 14-2. Addresses and Bits of Interrupt Control Registers (2/2)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF162H	Not used	–	–	–	–	–	–	–	–
FFFFF164H	Not used	–	–	–	–	–	–	–	–
FFFFF166H	Not used	–	–	–	–	–	–	–	–
FFFFF168H	UA0REIC	UA0REIF	UA0REMK	0	0	0	UA0REPR2	UA0REPR1	UA0REPR0
FFFFF16AH	UA0RIC	UA0RIF	UA0RMK	0	0	0	UA0RPR2	UA0RPR1	UA0RPR0
FFFFF16CH	UA0TIC	UA0TIF	UA0TMK	0	0	0	UA0TPR2	UA0TPR1	UA0TPR0
FFFFF16EH	CB0REIC	CB0REIF	CB0REMK	0	0	0	CB0REPR2	CB0REPR1	CB0REPR0
FFFFF170H	CB0RIC	CB0RIF	CB0RMK	0	0	0	CB0RPR2	CB0RPR1	CB0RPR0
FFFFF172H	CB0TIC	CB0TIF	CB0TMK	0	0	0	CB0TPR2	CB0TPR1	CB0TPR0
FFFFF174H	UA1REIC	UA1REIF	UA1REMK	0	0	0	UA1REPR2	UA1REPR1	UA1REPR0
FFFFF176H	UA1RIC	UA1RIF	UA1RMK	0	0	0	UA1RPR2	UA1RPR1	UA1RPR0
FFFFF178H	UA1TIC	UA1TIF	UA1TMK	0	0	0	UA1TPR2	UA1TPR1	UA1TPR0
FFFFF17AH	Not used	–	–	–	–	–	–	–	–
FFFFF17CH	Not used	–	–	–	–	–	–	–	–
FFFFF17EH	Not used	–	–	–	–	–	–	–	–
FFFFF180H	AD0IC	AD0IF	AD0MK	0	0	0	AD0PR2	AD0PR1	AD0PR0
FFFFF182H	AD1IC	AD1IF	AD1MK	0	0	0	AD1PR2	AD1PR1	AD1PR0
FFFFF184H	Not used	–	–	–	–	–	–	–	–
FFFFF186H	TM0EQIC0	TM0EQIF0	TM0EQMK0	0	0	0	TM0EQPR02	TM0EQPR01	TM0EQPR00

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

The IMR0 to IMR3 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 (m = 0 to 3) can be read or written in 16-bit units.

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).

After reset: FFFFH R/W Address: IMR3 FFFFF106H
IMR3L FFFFF106H, IMR3H FFFFF107H

	15	14	13	12	11	10	9	8
IMR3 (IMR3H[Note])	1	1	1	1	TM0EQMK0	1	AD1MK	AD0MK
	7	6	5	4	3	2	1	0
(IMR3L)	1	1	1	UA1TMK	UA1RMK	UA1REMK	CB0TMK	CB0RMK

After reset: FFFFH R/W Address: IMR2 FFFFF104H
IMR2L FFFFF104H, IMR2H FFFFF105H

	15	14	13	12	11	10	9	8
IMR2 (IMR2H[Note])	CB0REMK	UA0TMK	UA0RMK	UA0REMK	1	1	1	1
	7	6	5	4	3	2	1	0
(IMR2L)	TP3CCMK1	TP3CCMK0	TP3OVMK	TP2CCMK1	TP2CCMK0	TP2OVMK	TP1CCMK1	TP1CCMK0

After reset: FFFFH R/W Address: IMR1 FFFFF102H
IMR1L FFFFF102H, IMR1H FFFFF103H

	15	14	13	12	11	10	9	8
IMR1 (IMR1H[Note])	TP1OVMK	TP0CCMK1	TP0CCMK0	TP0OVMK	1	1	1	1
	7	6	5	4	3	2	1	0
(IMR1L)	1	1	1	1	TQ1CCMK3	TQ1CCMK2	TQ1CCMK1	TQ1CCMK0

After reset: FFFFH R/W Address: IMR0 FFFFF100H
IMR0L FFFFF100H, IMR0H FFFFF101H

	15	14	13	12	11	10	9	8
IMR0 (IMR0H[Note])	TQ1OVMK	TQ0CCMK3	TQ0CCMK2	TQ0CCMK1	TQ0CCMK0	TQ0OVMK	1	1
	7	6	5	4	3	2	1	0
(IMR0L)	LVIMK	PMK6	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0

xxMKn	Interrupt mask flag setting
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Note When reading/writing bits 15 to 8 of the IMR0 to IMR3 registers in 8-bit or 1-bit units, specify these bits as bits 7 to 0 of the IMR0H to IMR3H registers.

Caution Set bits 9 and 8 of the IMR0 register (bits 1 and 0 of the IMR0H register), bits 11 to 4 of the IMR1 register (bits 3 to 0 of the IMR1H register and bits 7 to 4 of the IMR1L register), bits 11 to 8 of the IMR2 register (bits 3 to 0 of the IMR2H register), bits 15 to 12, 10, and 7 to 5 of the IMR3 register (bits 7 to 4 and 2 of the IMR3H register and bits 7 to 5 of the IMR3L register) to 1. The operation when these settings are changed is not guaranteed.

Remark xx: Identification name of each peripheral unit (see **Table 14-2**)
n: Peripheral unit number (see **Table 14-2**)

14.3.6 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)

14.3.7 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.

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

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 signals and exceptions are acknowledged regardless of this flag. When a maskable interrupt 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).

14.4 External Interrupt Request Input Pins (INTP0 to INTP6)

14.4.1 Noise elimination

(1) Noise elimination of INTP0 to INTP5 pins

The INTP0 to INTP5 pins incorporate a noise eliminator that uses analog delay. 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 INTP6 pin

The INTP6 pin incorporates a digital noise eliminator.

The sampling clock that performs digital sampling can be selected from fxx/2, fxx/4, fxx/8, or fxx/16.

The system clock stops in the IDLE and STOP modes, so the INTP6 pin cannot be used to cancel the IDLE and STOP modes.

(a) External interrupt noise elimination control register (INTPNRC)

The INTPNRC register is used to select the sampling clock that is used to eliminate digital noise on the INTP6 pin. If the same level is not detected five times in a row, the signal is eliminated as noise.

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

Reset sets this register to 00H.

Cautions 1. If the input pulse lasts for the duration of 4 to 5 clocks, it is undefined whether the pulse is detected as a valid edge or eliminated as noise. So that the pulse is actually detected as a valid edge, the same pulse level must be input for the duration of 5 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: FFFFF310H

	7	6	5	4	3	2	1	0
INTPNRC	0	0	0	0	0	0	INTPNRC1	INTPNRC0

INTPNRC1	INTPNRC0	Selection of sampling clock
0	0	fxx/16
0	1	fxx/8
1	0	fxx/4
1	1	fxx/2

14.4.2 Edge detection

The valid edges of the INTP0 to INTP6 pins can be selected by program. 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 INTP0 to INTP6 signals become interrupt sources.
The valid edge is specified by the INTR0 and INTF0 registers.

(1) External interrupt rising edge specification register 0 (INTR0), external interrupt falling edge specification register 0 (INTF0)

The INTR0 and INTF0 registers are 8-bit registers that specify the trigger mode of the INTP0 to INTP6 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 INTF0n and INTR0n bits to 00, and then set the port mode.

After reset: 00H R/W Address: FFFFFC20H

	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
INTR0	0	INTR06	INTR05	INTR04	INTR03	INTR02	INTR01	INTR00

After reset: 00H R/W Address: FFFFFC00H

	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
INTF0	0	INTF06	INTF05	INTF04	INTF03	INTF02	INTF01	INTF00

Remark For the valid edge specification, see **Table 14-3**.

Table 14-3. Valid Edge Specification

INTF0n	INTR0n	Valid Edge Specification (n = 0 to 6)
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 INTP0 to INTP6 pins, be sure to clear the INTF0n and INTR0n bits to 00.

14.5 Software Exception

A software exception is generated when the CPU executes the TRAP instruction, and can always be acknowledged.

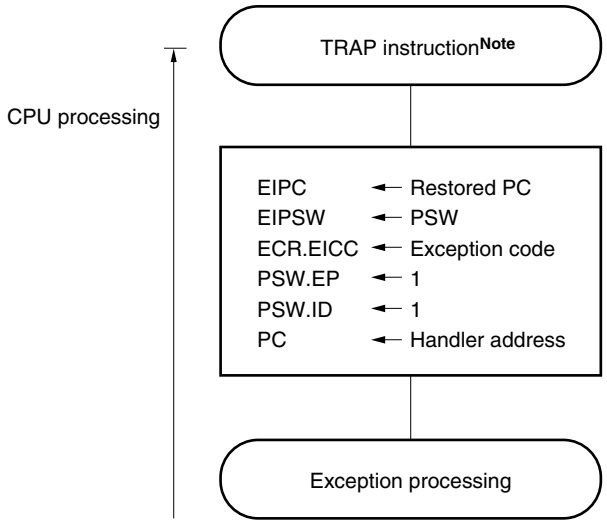
14.5.1 Operation

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 14-7. 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.

14.5.2 Restore

Recovery from software exception processing is carried out by the RETI instruction.

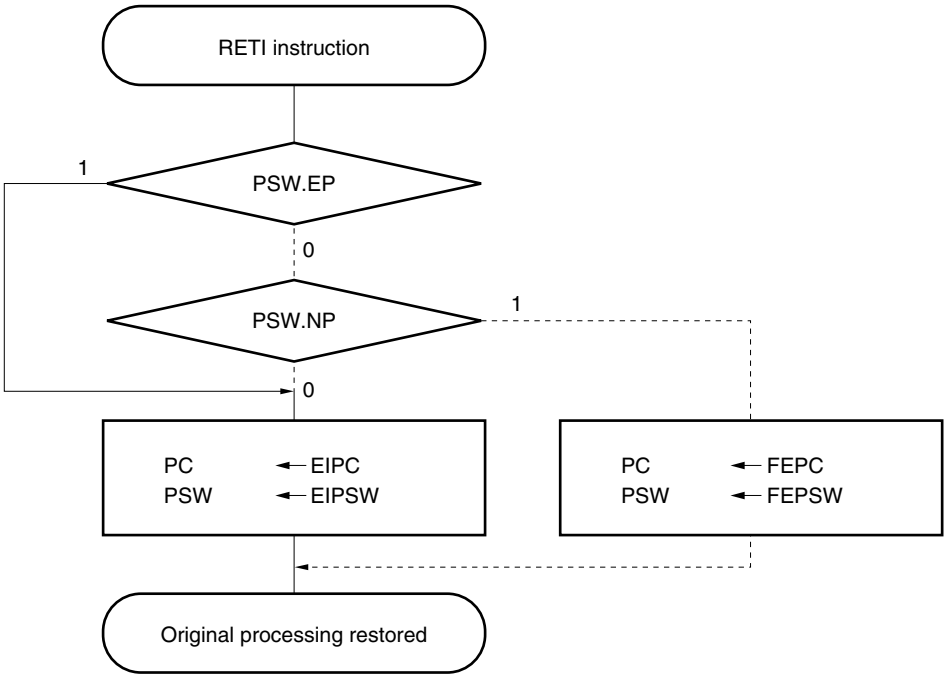
By executing the RETI instruction, the CPU carries out the following processing and shifts control to the restored PC's address.

<1> Loads the restored PC and PSW from EIPC and EIPSW because the PSW.EP bit is 1.

<2> Transfers control to the address of the restored PC and PSW.

The processing of the RETI instruction is shown below.

Figure 14-8. RETI Instruction Processing



Caution When the EP and NP bits are changed by the LDSR instruction during software exception processing, in order to restore the PC and PSW correctly during recovery 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.

14.5.3 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

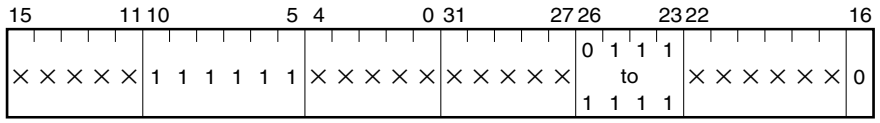
EP	Exception processing status
0	Exception processing not in progress
1	Exception processing in progress

14.6 Exception Trap

An exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. In the V850ES/IE2, an illegal opcode trap (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

14.6.1 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.

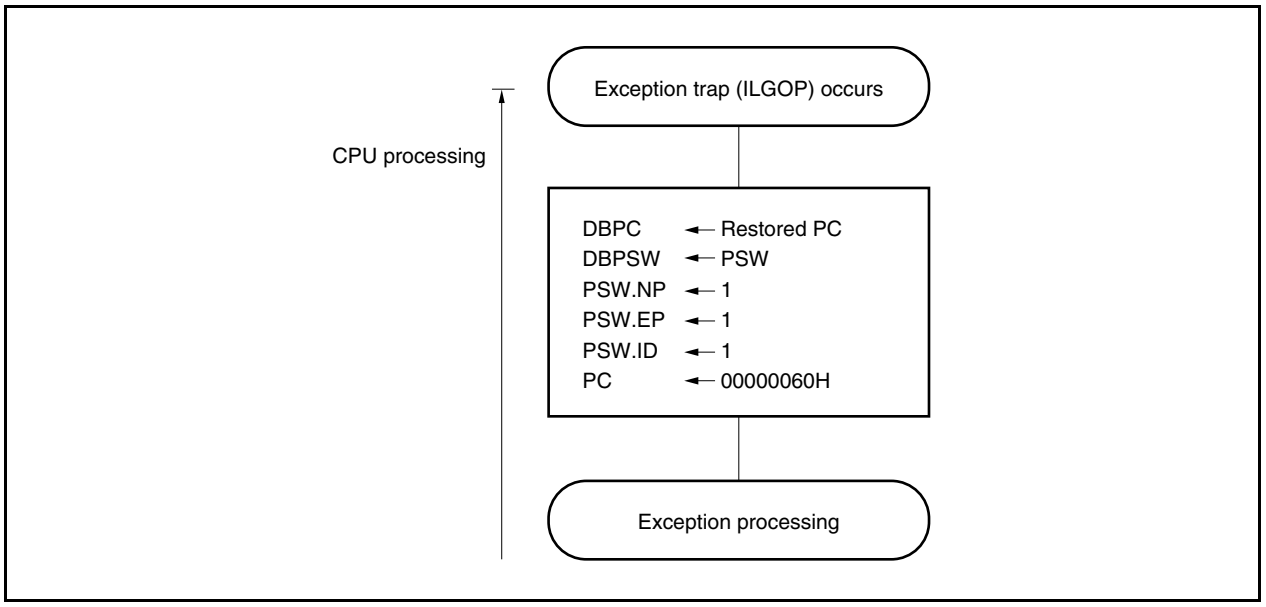
(1) Operation

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 14-9. Exception Trap Processing



(2) Restore

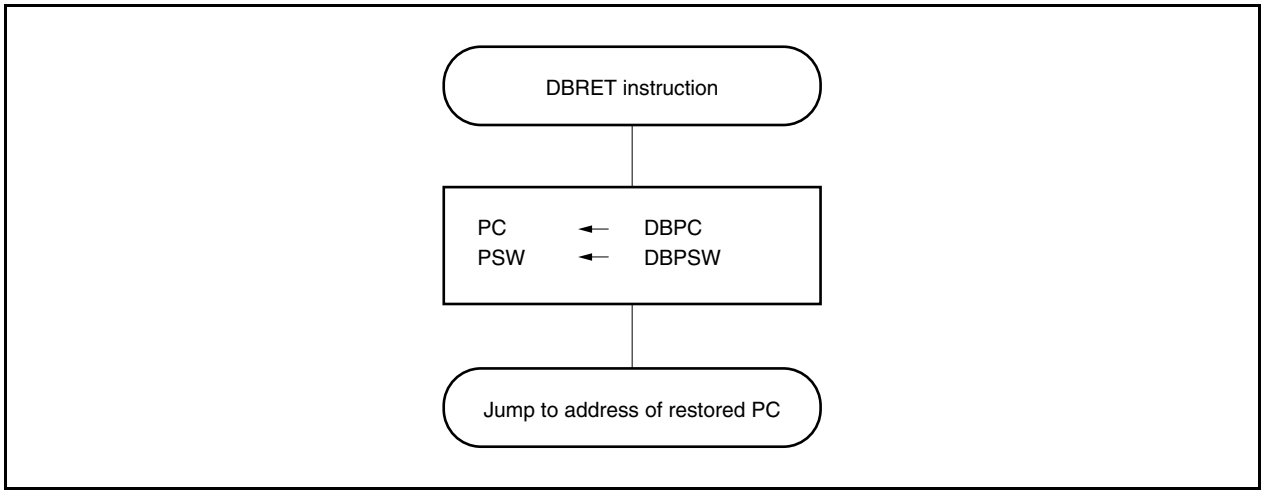
Recovery from an exception trap is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

<1> Loads the restored PC and PSW from DBPC and DBPSW.
<2> Transfers control to the address indicated by the restored PC and PSW.

<R> **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 14-10. Restore Processing from Exception Trap



14.6.2 Debug trap

The debug trap is an exception that can be acknowledged anytime and is generated by execution of the DBTRAP instruction.

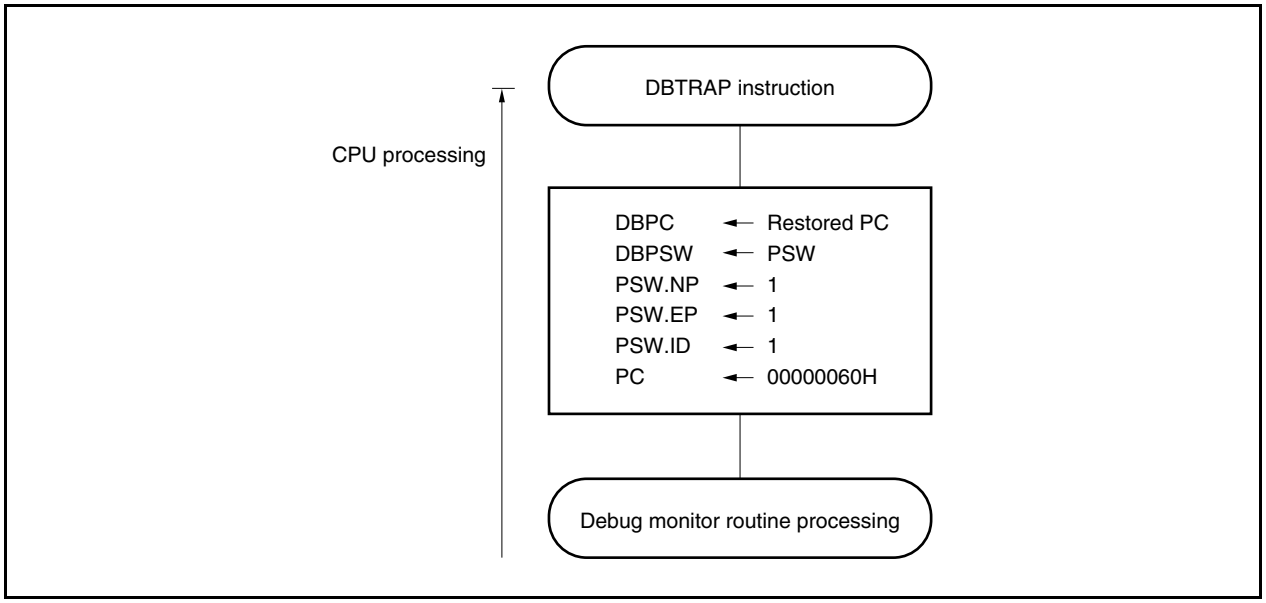
When the debug trap is generated, the CPU performs the following processing.

(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 14-11. Debug Trap Processing



(2) Restore

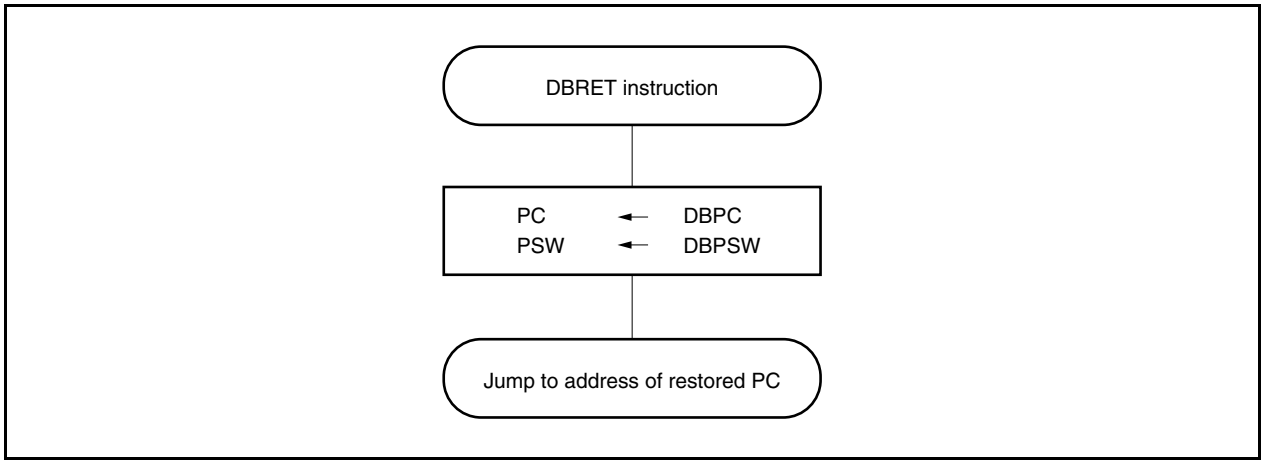
Recovery from a debug trap is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

<1> Loads the restored PC and PSW from DBPC and DBPSW.
<2> Transfers control to the address indicated by the restored PC and PSW.

Caution DBPC and DBPSW can be accessed 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 14-12. Restore Processing from Debug Trap



14.7 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)
        ...
        ...                                  ← Maskable interrupt acknowledgment
        ...
        ...
• DI instruction (interrupt acknowledgment disabled)
• Saved value restored to EIPSW
• Saved value restored to EIPC
• RETI instruction
```

(2) Generation of exception in 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 14-2**)
n: Peripheral unit number (see **Table 14-2**)

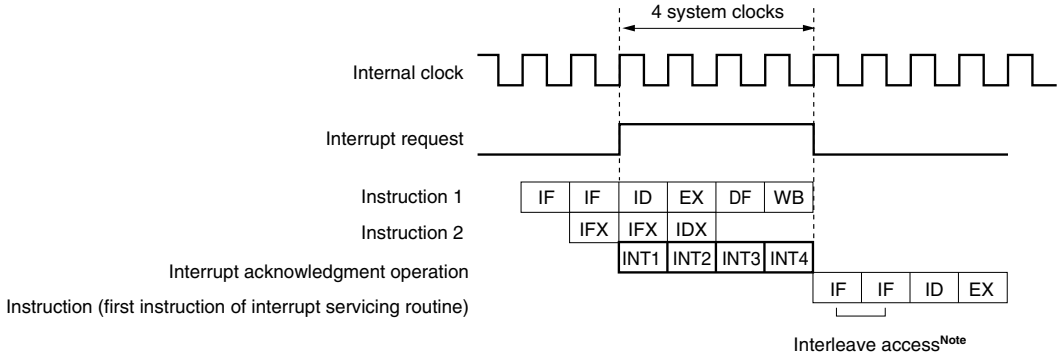
14.8 Interrupt Response Time of CPU

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 **14.9 Periods in Which CPU Does Not Acknowledge Interrupts**.)
- When an on-chip peripheral I/O register is accessed

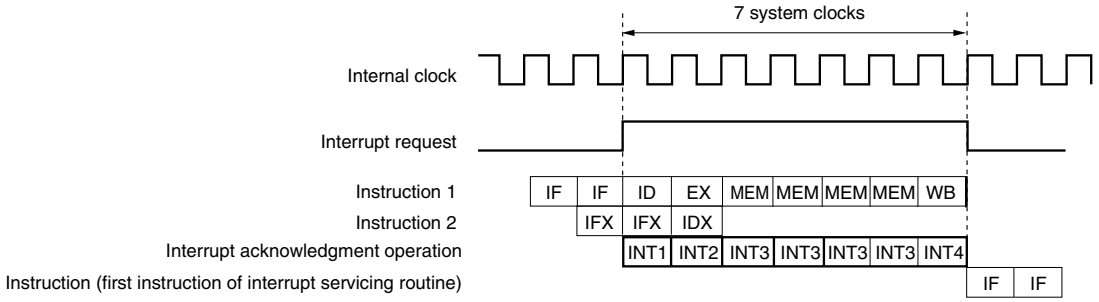
Figure 14-13. Pipeline Operation at Interrupt Request Signal Acknowledgment (Outline)

(1) Minimum interrupt response time



Note For interleave access, see **8.1.2 2-clock branch** in the **V850ES Architecture User's Manual (U15943E)**.

(2) Maximum interrupt response time



Remark INT1 to INT4: Interrupt acknowledgment processing
IFX: Invalid instruction fetch
IDX: Invalid instruction decode

Interrupt response time (internal system clock)				Conditions
	Internal interrupt	External interrupt		
		INTP0 to INTP5	INTP6	
Minimum	4	4 + Analog delay time	4 + **Note** + Digital noise filter	The following cases are exceptions. • In IDLE/STOP mode • Two or more interrupt request non-sample instructions are executed in succession • Access to on-chip peripheral I/O register
Maximum	7	7 + Analog delay time	7 + **Note** + Digital noise filter	

Note For details, see **14.4.1 (2) (a) External interrupt noise elimination control register (INTPNRC)**.

14.9 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) and command register 2 (PRCMD2).
- 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 3 (IMR0 to IMR3)
  - Power save control register (PSC)
  - Internal memory size setting register (IMS)

Remark xx: Identification name of each peripheral unit (see **Table 14-2**)
n: Peripheral unit number (see **Table 14-2**)

14.10 Caution

Note that if a port is set to external interrupt request input (INTPn), the timer/counter interrupt and A/D converter interrupt, which are alternate functions, do not occur (n = 0 to 7).

CHAPTER 15 STANDBY FUNCTION

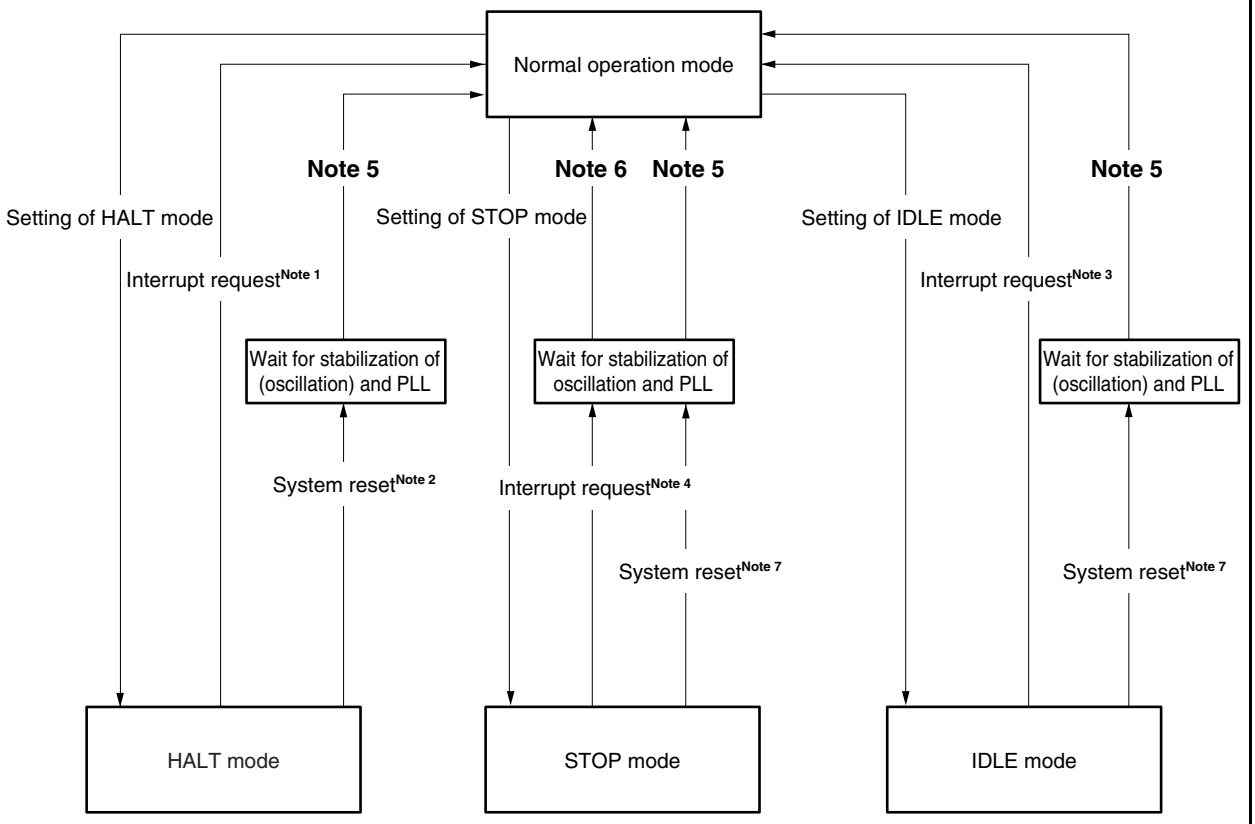
15.1 Overview

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 15-1.

Table 15-1. 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 circuits except the low-voltage detector (LVI), power-on-clear circuit (POC), oscillator, PLL, and CSIB in the slave mode
STOP mode	Mode to stop all the operations of the internal circuits except the low-voltage detector (LVI), power-on-clear circuit (POC), and CSIB in the slave mode

Figure 15-1. Status Transition



Notes 1. Non-maskable interrupt request signal (INTWDT) or unmasked maskable interrupt request signal

2. $\overline{\text{RESET}}$ pin input, reset signal generation by watchdog timer overflow (WDTRES), reset signal generation by low-voltage detection (LVIRES), or reset signal generation by power-on clear (POCRES).

3. Unmasked external interrupt request signal (INTP0 to INTP5), unmasked internal interrupt request signal (INTLVI), or unmasked internal interrupt request signal from peripheral functions operable in IDLE mode (interrupt request signal related to CSIB in slave mode)

4. Unmasked external interrupt request signal (INTP0 to INTP5), unmasked internal interrupt request signal (INTLVI), or unmasked internal interrupt request signal from peripheral functions operable in STOP mode (interrupt request signal related to CSIB in slave mode)

5. 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.

6. Oscillation stabilization time count by oscillation stabilization time wait control (OST)
The stabilization time is determined by the setting of the OSTS register.

7. $\overline{\text{RESET}}$ pin input, reset signal generation by low-voltage detection (LVIRES), or reset signal generation by power-on clear (POCRES).

15.2 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 **3.4.7 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 by maskable interrupt request (INTxx[Note])
0	Standby mode release by INTxx request enabled
1	Standby mode release by INTxx request disabled

STB	Operation mode setting
0	Normal mode
1	Standby mode

Note For details, see **Table 14-1 Interrupt Source List**.

Cautions 1. Be sure to clear 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.

<R> **4. When writing to the PSC register, use command register PRCMD.**

(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 clear bits 1 to 7 to "0".

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

15.3 HALT Mode

15.3.1 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 15-3 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 unmasked interrupt request signal is being held pending, the HALT mode is set but is released immediately by the pending interrupt request signal.

15.3.2 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 generation by watchdog timer overflow (WDTRES), reset signal generation by low-voltage detection (LVIRES), and reset signal generation by power-on clear (POCRES)).

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.

Caution When the PSC.INTM bit is set to 1, the HALT mode cannot be released by an unmasked maskable interrupt request signal.

(a) If an interrupt request signal with a priority lower than or same as the interrupt request signal 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 request signal 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, the execution branches to the handler address.

Table 15-2. 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, WDTRES signal generation, LVIRES signal generation, or POCRES signal generation

The same operation as the normal reset operation is performed.

Table 15-3. Operation Status in HALT Mode

Setting of HALT Mode / Item		Operation Status
Clock generator, PLL		Operates
System clock (f_{XX})		Supply
CPU		Stops operation
Interrupt controller		Operable
Timer	TMM0	Operable
	TMP0 to TMP3	Operable
	TMQ0, TMQ1	Operable
Watchdog timer		Operable
Serial interface	CSIB0	Operable
	UARTA0, UARTA1	Operable
A/D converters 0 and 1		Operable
Low-voltage detector (LVI)		Operable when LVI is used
Power-on-clear circuit (POC)		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.

15.4 IDLE Mode

15.4.1 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 15-5 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.

15.4.2 Releasing IDLE mode

The IDLE mode is released by an unmasked external interrupt request signal (INTP0 to INTP5 pin input), unmasked internal interrupt request signal (INTLVI), unmasked internal interrupt request signal from the peripheral functions operable in the IDLE mode (interrupt request signal related to CSIB in the slave mode), or reset signal ($\overline{\text{RESET}}$ pin input, reset signal generation by low-voltage detection (LVIRES), and reset signal generation by power-on clear (POCRES)).

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 the PSC.INTM bit is set to 1, the IDLE mode cannot be released by an unmasked maskable interrupt request signal.

(a) If an interrupt request signal with a priority lower than or same as the interrupt currently being serviced is generated, the IDLE 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 IDLE instruction.

(b) If an interrupt request signal with a priority higher than that of the interrupt currently being serviced is issued, the IDLE mode is released and that interrupt request signal is acknowledged. Therefore, the execution branches to the handler address.

Table 15-4. 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, LVIRES signal generation, and POCRES signal generation

The same operation as the normal reset operation is performed.

Table 15-5. Operation Status in IDLE Mode

Setting of IDLE Mode / Item		Operation Status
Clock generator, PLL		Operates
System clock (f_{XX})		Stops supply
CPU		Stops operation
Interrupt controller		Stops operation
Timer	TMM0	Stops operation
	TMP0 to TMP3	Stops operation
	TMQ0, TMQ1	Stops operation
Watchdog timer		Stops operation
Serial interface	CSIB0	Operable when $\overline{\text{SCKB0}}$ input clock is selected as count clock (in slave mode)
	UARTA0, UARTA1	Stops operation
A/D converters 0 and 1		Stops operation
Low-voltage detector (LVI)		Operable when LVI is used
Power-on-clear circuit (POC)		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.

15.5 STOP Mode

15.5.1 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 <R> are retained. The CPU and other on-chip peripheral functions stop operating. The on-chip peripheral functions that can operate with an external clock continue operating.

Table 15-7 shows the operation status in the STOP mode.

Because the STOP mode stops operation of the clock generator, it reduces the power consumption to a level lower than the IDLE mode. The power consumption is therefore minimized with only leakage current flowing if the external clock is not used.

Caution Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the STOP mode.

15.5.2 Releasing STOP mode

The STOP mode is released by an unmasked external interrupt request signal (INTP0 to INTP5 pin input), unmasked internal interrupt request signal (INTLVI), unmasked internal interrupt request signal from the peripheral functions operable in the STOP mode (interrupt request signal related to CSIB in the slave mode), or reset signal ($\overline{\text{RESET}}$ pin input, reset signal generation by low-voltage detection (LVIRES), and reset signal generation by power-on clear (POCRES)).

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 the PSC.INTM bit is set to 1, the STOP mode cannot be released by an unmasked maskable interrupt request signal.

(a) If an interrupt request signal with a priority lower than or same as the interrupt request signal currently being serviced is generated, the STOP 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 STOP instruction.

(b) If an interrupt request signal with a priority higher than that of the interrupt request signal currently being serviced is issued, the STOP mode is released and that interrupt request signal is acknowledged. Therefore, the execution branches to the handler address.

Table 15-6. Operation After Releasing STOP Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Unmasked maskable interrupt request signal	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, LVIRES signal generation, and POCRES signal generation

The same operation as the normal reset operation is performed.

Table 15-7. Operation Status in STOP Mode

Item \ Setting of STOP Mode		Operation Status
Clock generator, PLL		Stops operation
System clock (f_{XX})		Stops supply
CPU		Stops operation
Interrupt controller		Stops operation
Timer	TMM0	Stops operation
	TMP0 to TMP3	Stops operation
	TMQ0, TMQ1	Stops operation
Watchdog timer		Stops operation
Serial interface	CSIB0	Operable when $\overline{\text{SCKB0}}$ input clock is selected as count clock (in slave mode)
	UARTA0, UARTA1	Stops operation
A/D converters 0 and 1		Stops operation
Low-voltage detector (LVI)		Operable when LVI is used
Power-on-clear circuit (POC)		Operable
Clock monitor		Stops operation (however, internal oscillator continues operating)
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.

15.6 Securing Oscillation Stabilization Time

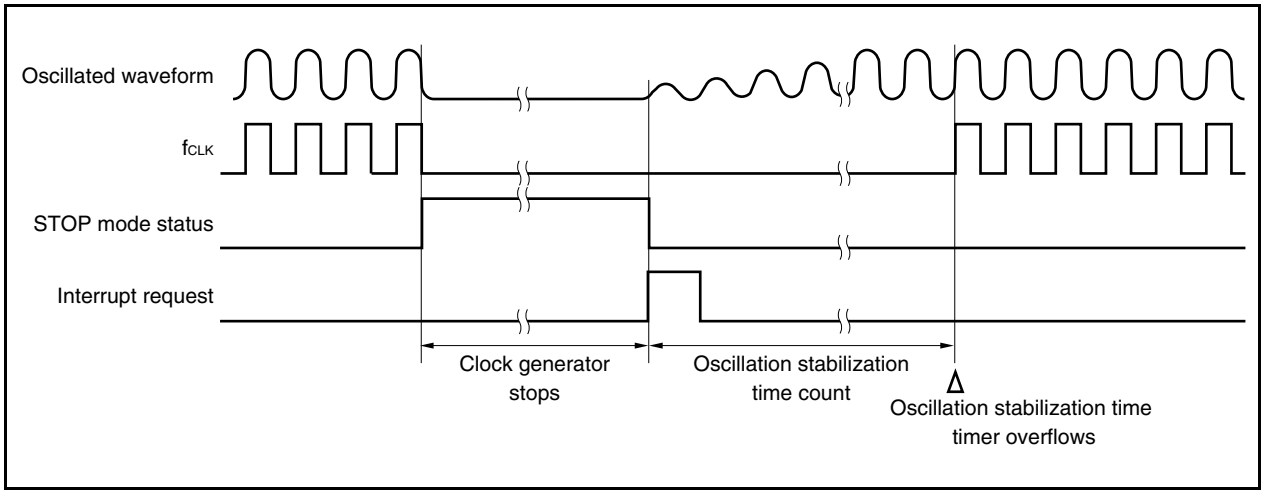
When the STOP mode is released, the oscillation stabilization time set by the OSTS register elapses. The oscillation stabilization time is the reset value of the OSTS register, $2^{16}/f_X$ (26.2 ms at f_X = 2.5 MHz), if the STOP mode is released by reset signal ($\overline{\text{RESET}}$ pin input, reset signal generation by low-voltage detection (LVIRES), and reset signal generation by power-on clear (POCRES)).

However, the actual oscillation stabilization time is half this value (after reset: $2^{15}/f_X$ (13.1 ms at f_X = 2.5 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.

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 15-2. Oscillation Stabilization Time



Caution For details of the OSTS register, see 5.3 (6) Oscillation stabilization time select register (OSTS).

CHAPTER 16 RESET FUNCTIONS

16.1 Overview

The following reset functions are available.

- Reset by $\overline{\text{RESET}}$ pin input
- Reset by watchdog timer overflow (WDTRES)
- System reset by low-voltage detector (LVI) (LVIRES)
- System reset by power-on-clear circuit (POC) (POCRES)

16.2 Registers to Check Reset Source

(1) Reset source flag register (RESF)

The RESF register is a special register that can be written only by a combination of specific sequences (see **3.4.7 Special registers**).

The RESF register indicates that a reset signal is generated by the watchdog timer (WDT).

The LVIRF and WDT2RF bits are cleared by reset via the $\overline{\text{RESET}}$ pin or by a bit manipulation instruction or store instruction (writing 0 to the LVIRF and WDT2RF bits).

This register is read or written in 8-bit units. However, bit 0 is write-only.

This register is cleared 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 these.

After reset: 00H[Note] R/W Address: FFFFF888H

	7	6	5	4	3	2	1	0
RESF	0	0	0	WDT2RF	0	0	0	LVIRF

WDT2RF	Reset signal from WDT
0	Not generated/cleared
1	Generated

LVIRF	Clear of RESF2.LVIRFS bit
0	Cleared
1	Write disabled

Note The value of this register is cleared to 00H after a reset by $\overline{\text{RESET}}$ pin input or the power-on-clear circuit (POC). When a reset is executed by watchdog timer overflow, this register is set to 10H or 11H. When reset by the low-voltage detector (LVI), bit 4 retains the value before reset and bit 0 is undefined.

Cautions 1. Only "0" can be written to each bit of this register. If writing "0" conflicts with setting the flag (occurrence of reset), setting the flag takes precedence.

<R> **2. When writing to the RESF register, use command register PRCMD.**

(2) Reset source flag register 2 (RESF2)

The RESF2 register indicates that a reset signal is generated by the low-voltage detector (LVI).

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

This register is cleared to 00H or set to 10H by $\overline{\text{RESET}}$ pin input and the power-on-clear circuit (POC). The default values are different when a reset is executed by other than these.

After reset: **Note 1** R Address: FFFFF8A8H

	7	6	5	4	3	2	1	0
RESF2	0	0	0	Undefined	0	0	0	LVIRFS Note 2

LVIRFS Note 2	Reset signal from LVI
0	Not generated
1	Generated

Notes 1. After a reset by $\overline{\text{RESET}}$ pin input or the power-on-clear circuit (POC), this register is set to 00H or 10H. After a reset by the low-voltage detector (LVI), this register is set to 01H or 11H. After a reset by watchdog timer overflow, the register value before reset is held.

2. Writing 0 to the RESF.LVIRF bit clears (0) the LVIRFS flag.

Table 16-1. Values of Bits 0 and 4 of RESF and RESF2 Registers When Reset Source Is Generated

Register / Reset Source	RESF Register		RESF2 Register	
	Bit 4	Bit 0	Bit 4	Bit 0
Reset function by $\overline{\text{RESET}}$ pin input	0	0	Undefined	0
System reset (POCRES) by power-on-clear circuit (POC)	0	0	Undefined	0
Reset function (WDTRES) by WDT overflow	1	Undefined	Held	Held
System reset (LVIRES) by low-voltage detector (LVI)	Held	Undefined	Undefined	1

16.3 Operation

16.3.1 Reset operation via $\overline{\text{RESET}}$ pin

When a low level is input to the $\overline{\text{RESET}}$ pin, the system is reset, and each hardware unit is initialized.

When the level of the $\overline{\text{RESET}}$ pin is changed from low to high, the reset status is released.

If the reset status is released by $\overline{\text{RESET}}$ pin input, the oscillation stabilization time of the oscillator elapses (reset value of OSTS register: $2^{16}/f_X$) and then the CPU starts program execution.

Table 16-2. Hardware Status on $\overline{\text{RESET}}$ Pin Input

Item	During Reset Period	After Reset Release
Oscillator (f_X)	Oscillation stops	Oscillation starts
Internal oscillator	Oscillation stops	Oscillation starts
Peripheral clock (f_X to $f_X/2{,}048$)	Operation stops	Operation starts after securing oscillation stabilization time
Internal system clock (f_{CLK}), CPU clock (f_{CPU})	Operation stops	Operation starts after securing oscillation stabilization time (initialized to $f_{XX}/8$)
CPU	Initialized	Program execution starts after securing oscillation stabilization time[Note 1]
Internal RAM	Undefined in case of power-on reset or if writing data to RAM (by CPU) and reset conflict (data is damaged). Otherwise value immediately after reset is retained[Note 2].	
I/O lines (ports/alternate-function pins)	High impedance	
On-chip peripheral I/O registers	Initialized to specified status	
Other on-chip peripheral functions	Operation stops	Operation can be started after securing oscillation stabilization time

Notes 1. With the μPD70F3714, program execution is delayed by insertion of internal processing for boot switching.

2. The firmware of the μPD70F3714 uses part of the internal RAM (used RAM area: 3FFE000H to 3FFE095H, 3FFEFBAH to 3FFEFFFH) after the internal system reset operation has been released because it supports a boot switching function. Therefore, the contents of some RAM areas are not retained on power-on reset.

Figure 16-1. Timing of Reset Operation by $\overline{\text{RESET}}$ Pin Input

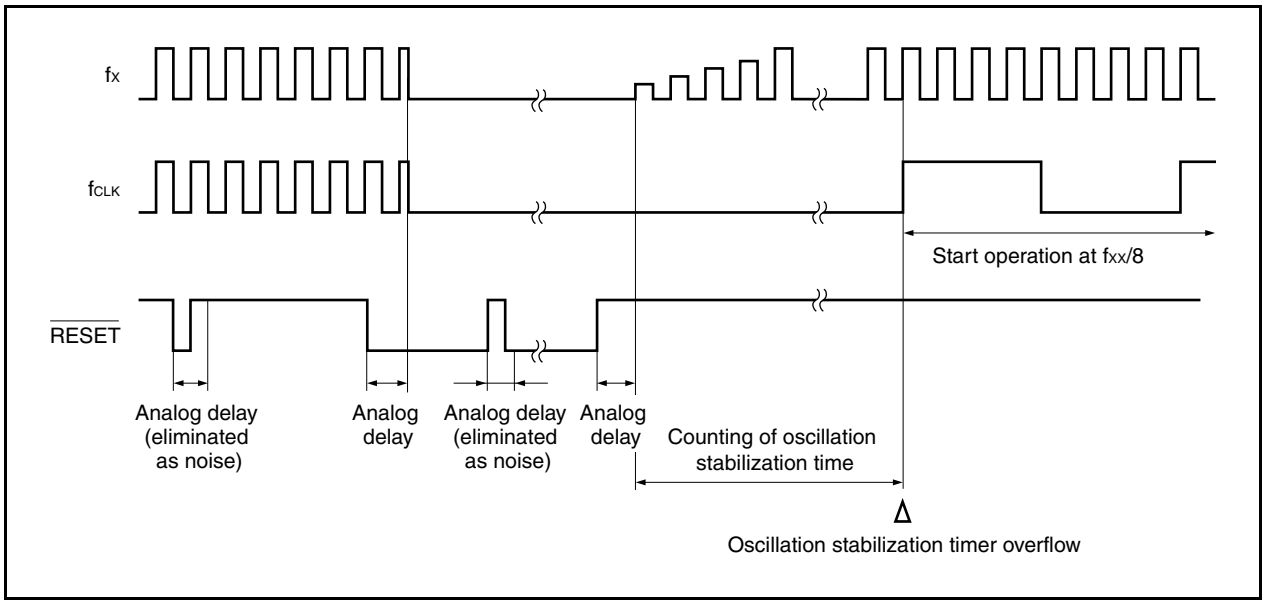
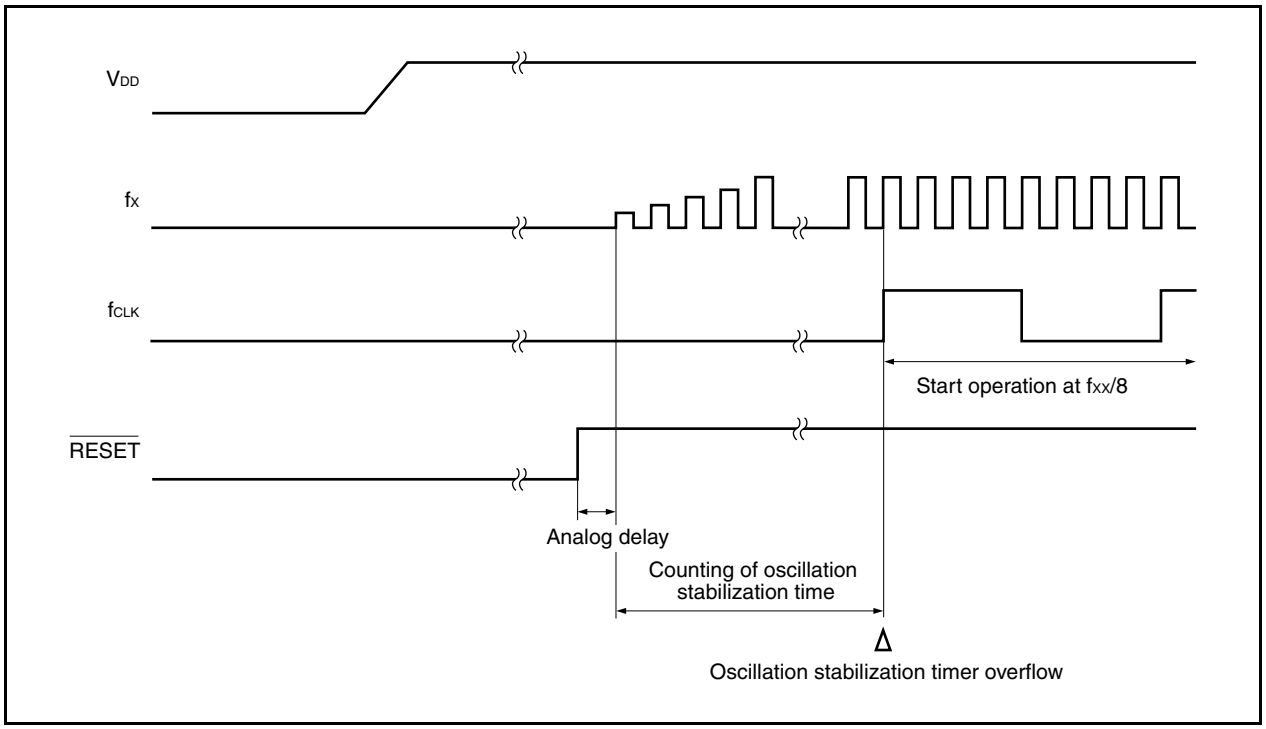


Figure 16-2. Timing of Power-on (Power Application) Reset Operation



16.3.2 Reset operation by watchdog timer (WDT) overflow (WDTRES)

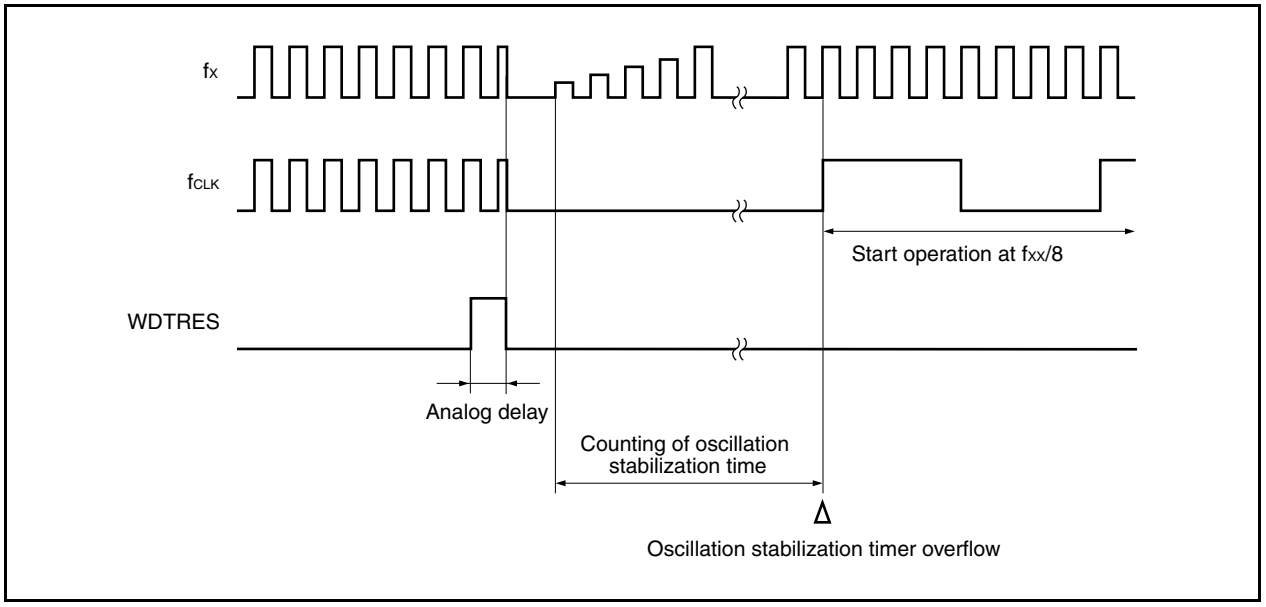
When the watchdog timer is set to the reset operation mode due to overflow, upon watchdog timer overflow (WDTRES signal generation), a system reset is executed and the hardware is initialized to the initial status.

Following watchdog timer overflow, the reset status is entered and lasts the predetermined time (analog delay), and the reset status is then automatically released. Following reset release, the CPU starts program execution after securing the oscillation stabilization time (initial value of OSTS register: $2^{16}/f_X$) of the oscillator.

The oscillator is stopped during the reset period, 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 **16.3.1 Reset operation via $\overline{\text{RESET}}$ pin**).

Figure 16-3. Timing of Reset Operation by WDTRES Signal Generation



16.3.3 Low-voltage detector (LVI)

(1) Functions

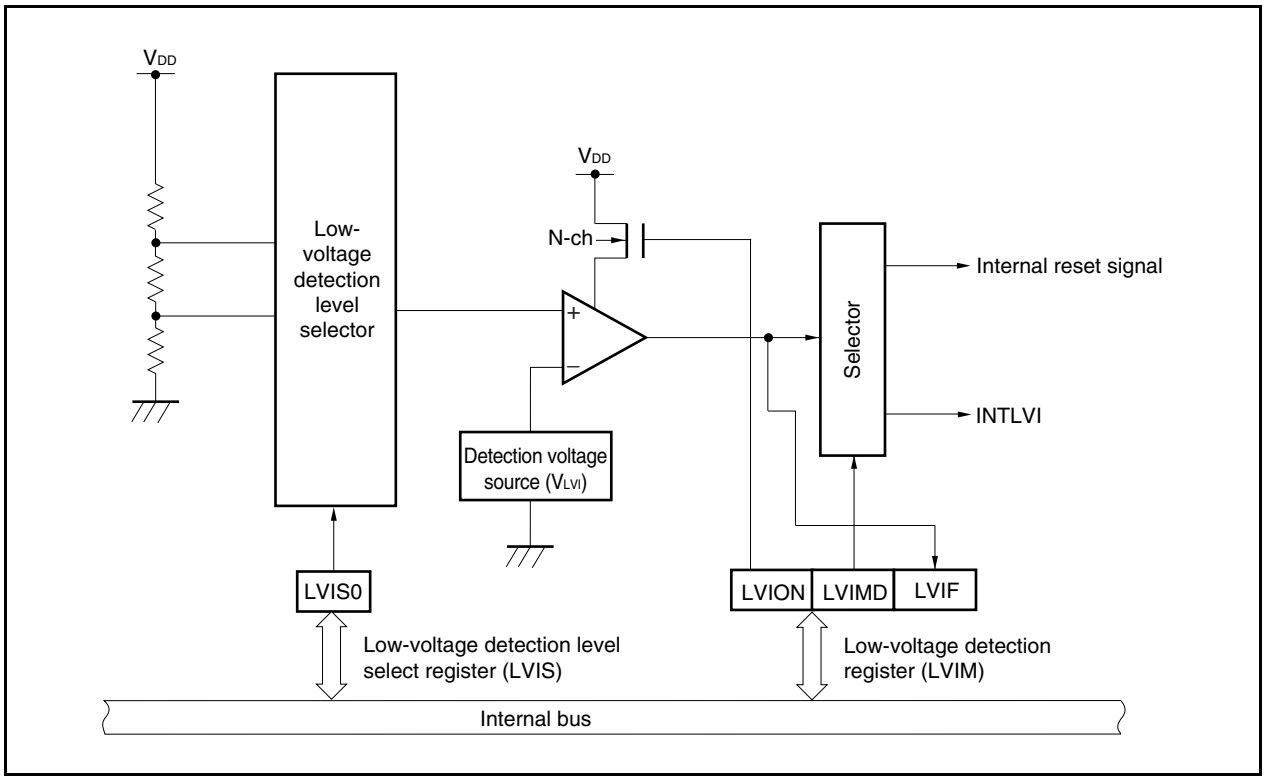
The low-voltage detector (LVI) has the following functions.

- Compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}) and generates an interrupt request signal or internal reset signal when $V_{DD} < V_{LVI}$.
- The level of the supply voltage to be detected can be changed in two steps.
- An interrupt request signal or internal reset signal can be selected.
- Can operate in HALT/IDLE/STOP mode.
- Operation can be stopped by software.

(2) Configuration

The block diagram is shown below.

Figure 16-4. Block Diagram of Low-Voltage Detector



(3) Control registers

(a) 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 **3.4.7 Special registers**).

This register can be read or written in 8-bit or 1-bit units. However, bit 0 is read-only.

This register is cleared to 00H by a reset via $\overline{\text{RESET}}$ pin input, power-on-clear circuit (POC), and watchdog timer overflow. This register is set to 82H after a reset by the low-voltage detector (LVI).

After reset: 00H[Note] 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 INTLVI 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

Note This register is cleared to 00H after a reset by $\overline{\text{RESET}}$ pin input, power-on-clear circuit (POC), and watchdog timer overflow. This register is set to 82H after a reset by the low-voltage detector (LVI).

Cautions 1. After setting the LVION bit to 1, wait for 0.1 ms (TYP) (target value) before checking the voltage using the LVIF bit.

2. The value of the LVIF flag is output as the output signal INTLVI when the LVION bit = 1 and LVIMD bit = 0.

3. Be sure to clear bits 2 to 6 to "0".

<R> **4. When writing to the LVIM register, use command register PRCMD2.**

(b) 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 or 1-bit units.

This register is cleared to 00H by a reset via $\overline{\text{RESET}}$ pin input, power-on-clear circuit (POC), and watchdog timer overflow. The value of this register is retained after a reset by the low-voltage detector (LVI).

After reset: 00H[Note] 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

Note This register is cleared to 00H by a reset via $\overline{\text{RESET}}$ pin input, power-on-clear circuit (POC), and watchdog timer overflow. The value of this register is retained after a reset by the low-voltage detector (LVI).

Caution Be sure to clear bits 1 to 7 to "0".

(c) Internal RAM data status register (RAMS)

The RAMS register is a flag register that indicates whether the internal RAM is valid or not. The RAMS register is a special register. It can be written only by a combination of specific sequences (see **3.4.7 Special registers**).

For the RAMS register, see **(5) RAM retention voltage detection operation**.

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

This register is set to 01H by $\overline{\text{RESET}}$ pin input (during RAM access only) and watchdog timer overflow. The value is retained when reset by other factors.

<R> **Cautions 1. When writing to the RAMS register, use command register PRCMD2.**

2. The following shows the specific sequence after reset.

- **Setting conditions: Detection of voltage lower than specified level (2.0 V ± 0.1 V)**
  Writing 1 via a specific sequence
  Generation of reset signal by watchdog timer overflow
  $\overline{\text{RESET}}$ pin input while RAM is being accessed
- **Clearing condition: Writing of 0 in specific sequence**

After reset: 01H[Note] R/W Address: FFFFF892H

	7	6	5	4	3	2	1	<0>
RAMS	0	0	0	0	0	0	0	RAMF

RAMF	Internal RAM data valid/invalid
0	Valid
1	Invalid

Note This register is set to 01H by $\overline{\text{RESET}}$ pin input (during RAM access only) and watchdog timer overflow. The value is retained when reset by other factors.

(4) Operation

Depending on the setting of the LVIM.LVIMD bit, an interrupt request signal (INTLVI) or an internal reset signal is generated.

(a) To use for internal reset signal

If the supply voltage falls below the voltage detected by the low-voltage detector when LVI operation is enabled, a system reset is executed (when the LVIM.LVIMD bit is set to 1), and the hardware is initialized to the initial status.

The reset status lasts from when a supply voltage drop has been detected until the supply voltage rises above the LVI detection voltage. Following reset release, the CPU starts program execution after securing the oscillation stabilization time (initial value of OSTS register: 2^{16}/fx) of the oscillator.

The main clock oscillator is stopped during the reset period, 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 **16.3.1 Reset operation via $\overline{\text{RESET}}$ pin**). The following shows the operation setting method and timing chart.

<To start operation>

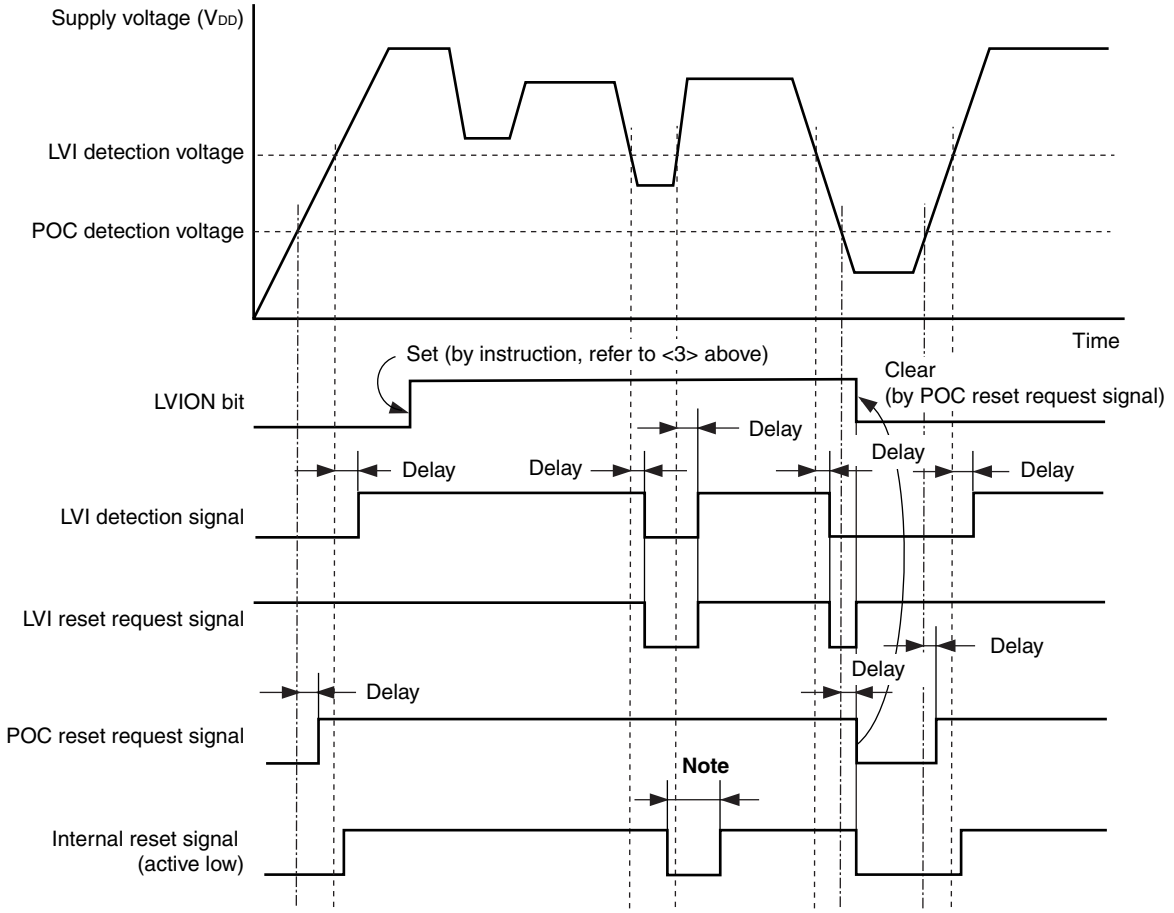
<1> Mask the interrupt of LVI.
<2> Select the voltage to be detected by using the LVIS.LVIS0 bit.
<3> Set the LVIM. LVION bit to 1 (to enable operation).
<4> Insert a wait cycle of 0.1 ms (TYP) (target value) or more by software.
<5> By using the LVIM.LVIF bit, check if the supply voltage > detection 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 LVI is generated.

<To stop operation>

LVI operation cannot be stopped until a reset request is generated from other than LVI.

Figure 16-5. Operation Timing of Low-Voltage Detector (LVIMD Bit = 1)



Note During the period in which the supply voltage is the set voltage or lower, the internal reset signal is retained (internal reset state).

(b) To use for interrupt

When the operation of LVI is enabled, the supply voltage and detection voltage are compared. If the supply voltage is lower than the detection voltage, an interrupt request signal (INTLVI) is generated (when the LVIM.LVIMD bit is cleared to 0).

The following shows the operation setting method and timing chart.

<To start operation>

<1> Mask the interrupt of 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 0.1 ms (TYP) (target value) or more by software.

<5> By using the LVIM.LVIF bit, check if the supply voltage > detection voltage.

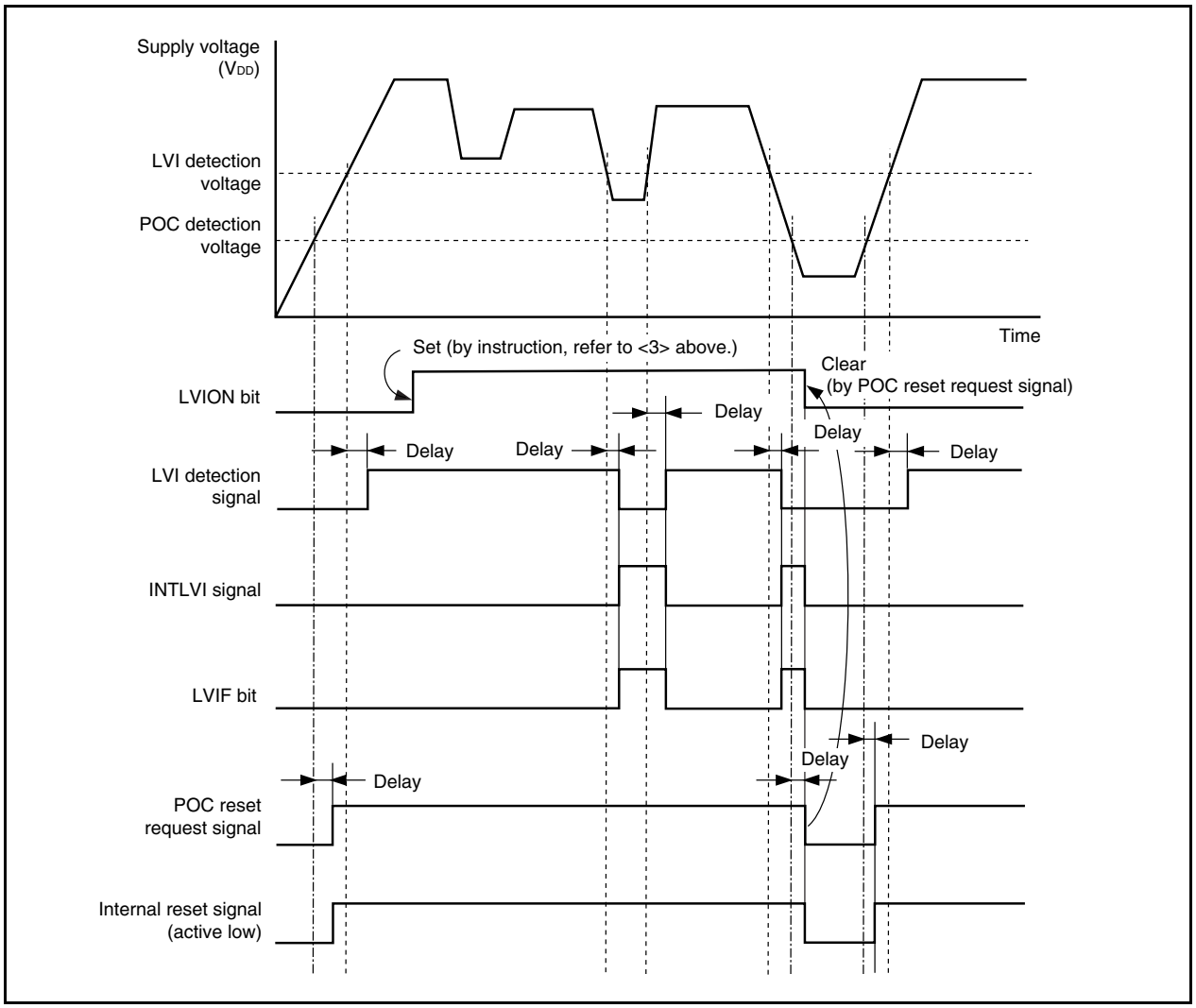
<6> Clear the interrupt request flag of LVI.

<7> Unmask the interrupt of LVI.

<To stop operation>

Clear the LVION bit to 0.

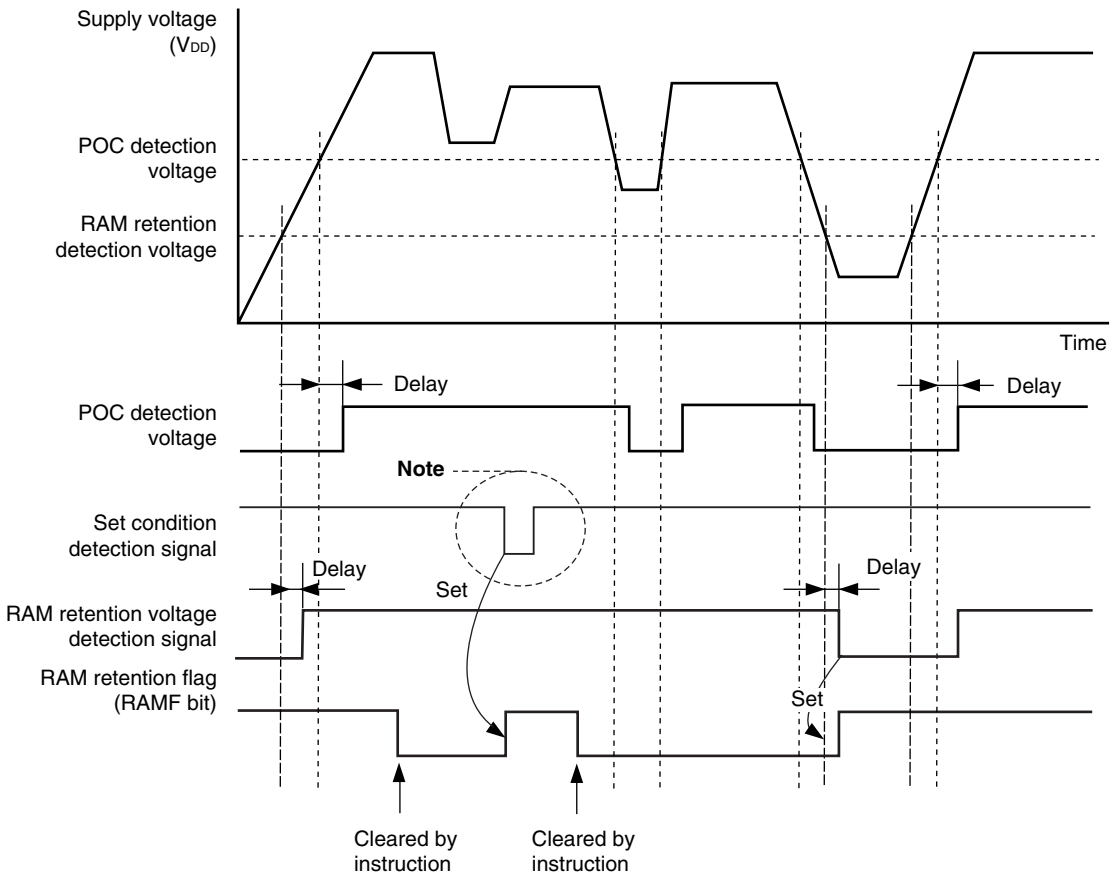
Figure 16-6. Operation Timing of Low-Voltage Detector (LVIMD Bit = 0)



(5) RAM retention voltage detection operation

The supply voltage and detection voltage (V_{RAMH} = 2.0 V ±0.1 V) are compared. When the supply voltage drops below the detection voltage (including on power application), the RAMS.RAMF bit is set (1).

Figure 16-7. Operation Timing of RAM Retention Voltage Detection Function



Note A reset signal (WDTRES) is generated due to an overflow of the watchdog timer or $\overline{\text{RESET}}$ pin input during RAM access.

16.3.4 Power-on-clear circuit (POC)

(1) Overview

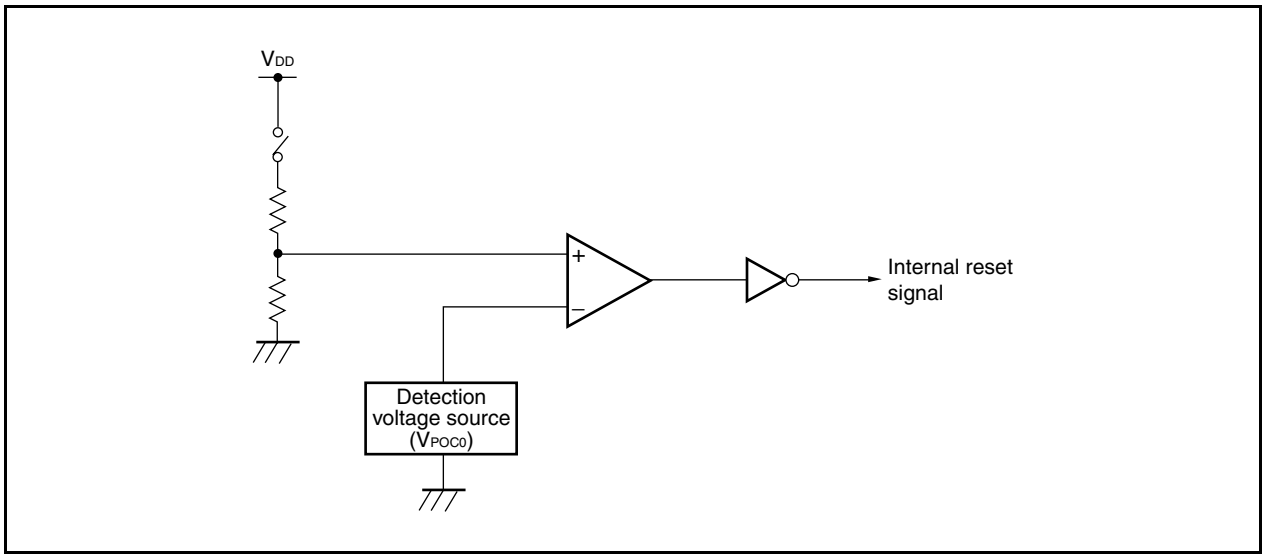
An overview of the power-on-clear (POC) circuit is shown below.

- Generates a reset signal upon power application.
- Compares the supply voltage (V_{DD}) and detection voltage (V_{POC0}), and generates a reset signal when $V_{DD} < V_{POC0}$.

(2) Configuration

The block diagram is shown below.

Figure 16-8. Block Diagram of Power-on-Clear Circuit



(3) Operation

When the supply voltage and detection voltage are compared and if the supply voltage is lower than the detection voltage (including at power application), the system is reset and each hardware is returned to the specific status.

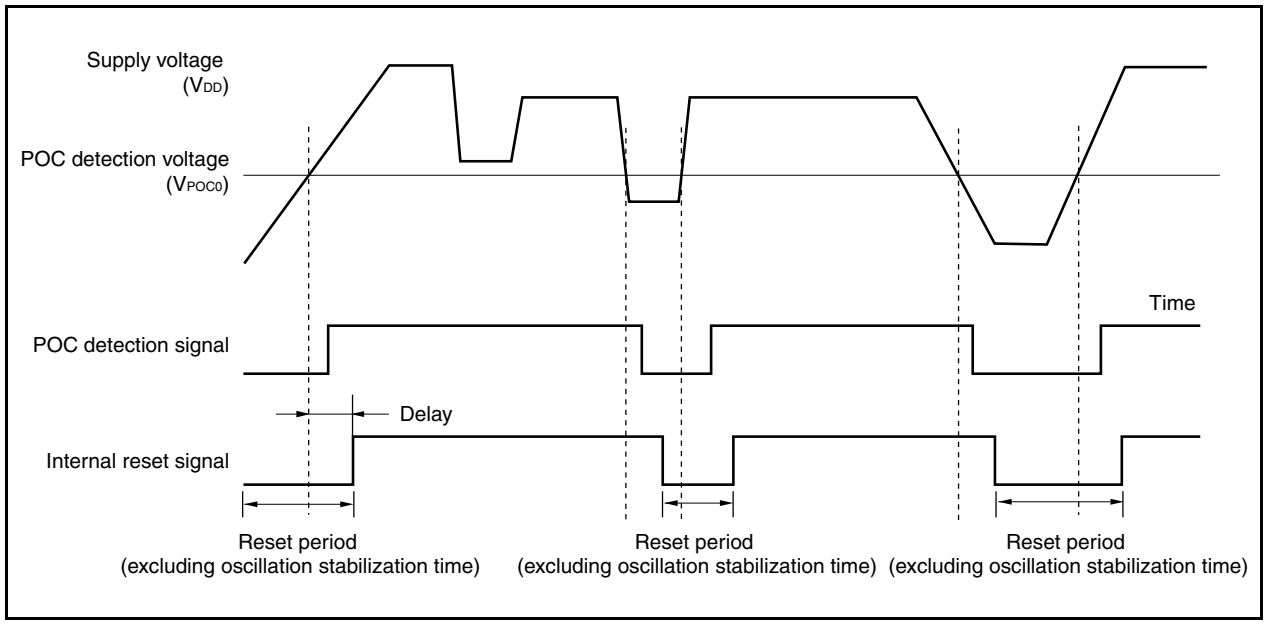
The system is reset from when low voltage is detected until the supply voltage becomes higher than the detection voltage. After a reset is released, when the oscillation stabilization time (default value of the OSTS register: $2^{16}/f_X$) of the oscillator has elapsed, the CPU starts executing the program.

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 **16.3.1 Reset operation via $\overline{\text{RESET}}$ pin**).

The following shows the timing chart.

Figure 16-9. Timing of Reset Signal Generation by Power-on-Clear Circuit



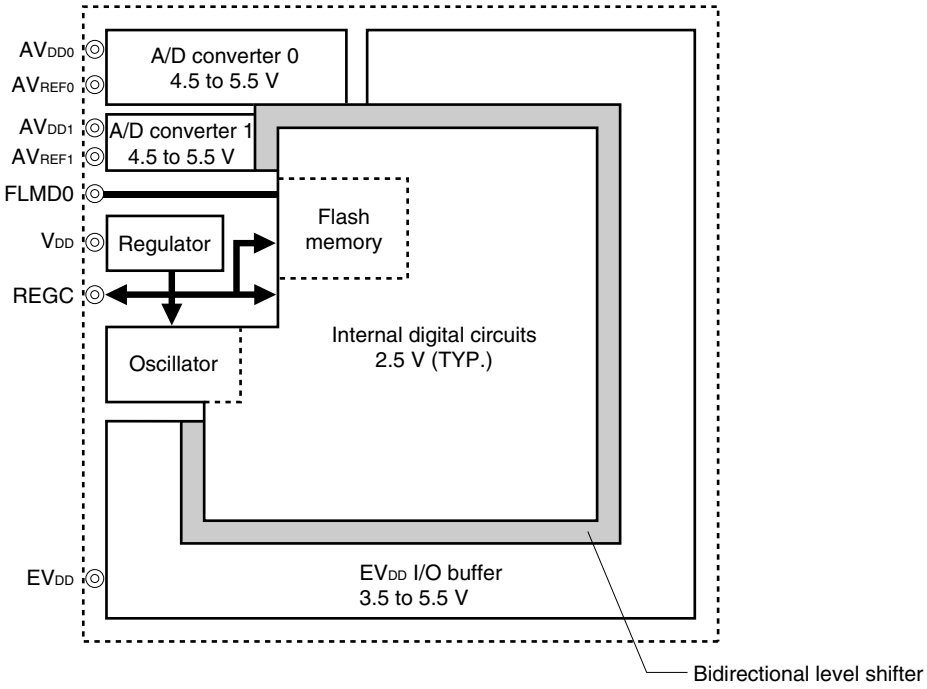
CHAPTER 17 REGULATOR

17.1 Overview

The V850ES/IE2 includes a regulator to reduce power consumption and noise.

This regulator supplies a stepped-down V_{DD} power supply voltage to the oscillation block and internal logic circuits (except the A/D converter and I/O buffers). The regulator output voltage (REGC pin) is set to 2.5 V (TYP.).

Figure 17-1. Regulator



Caution Use the regulator with a setting of $V_{DD} = EV_{DD} = AV_{DDn} = AV_{REFn}$ (n = 0, 1).

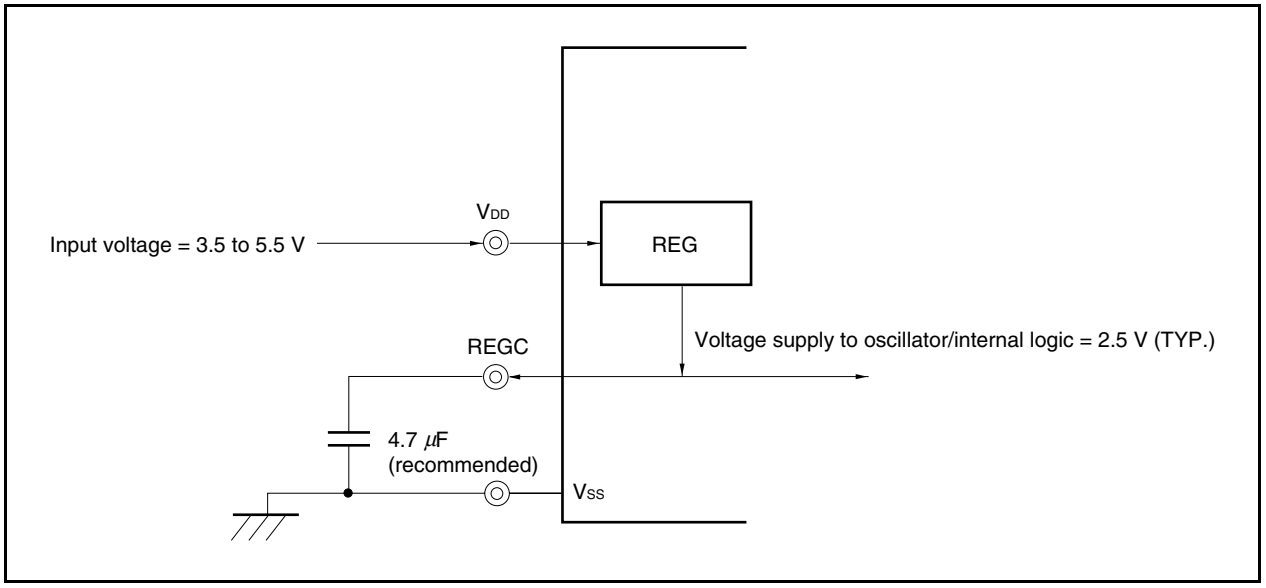
17.2 Operation

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)) to the REGC pin to stabilize the regulator output.

A diagram of the regulator pin connection method is shown below.

Figure 17-2. REGC Pin Connection



CHAPTER 18 FLASH MEMORY

The μPD70F3713 and 70F3714 incorporate 64 KB and 128 KB flash memories as the internal ROM.

Caution For the electrical specifications related to the flash memory rewriting, refer to CHAPTER 19 ELECTRICAL SPECIFICATIONS.

The V850ES/IE2 is commonly used in the following development environments and mass production applications.

- For altering software after the V850ES/IE2 is soldered onto the target system.
- For data adjustment when starting mass production.
- For differentiating software according to the specification in small scale production of various models.
- For facilitating inventory management.
- For updating software after shipment.

Instruction fetch to the flash memory can access 4 bytes in 1 clock.

Writing to the flash memory can be performed with it is mounted on the target system (on-board). Use the dedicated flash memory programmer connecting to the target system.

18.1 Features

- 4-byte/1-clock access (when instruction is fetched)
- Capacity: 128 KB/64 KB
- Write voltage: Erase/write with a single power supply
- Rewriting method
  - Rewriting by communication with dedicated flash memory programmer via serial interface (on-board/off-board programming)
  - Rewriting flash memory by user program (self programming (μPD70F3714 only))
- Flash memory write prohibit function supported (security function)
- Safe rewriting[Note] of entire flash memory area by self programming using boot swap function (μPD70F3714 only)

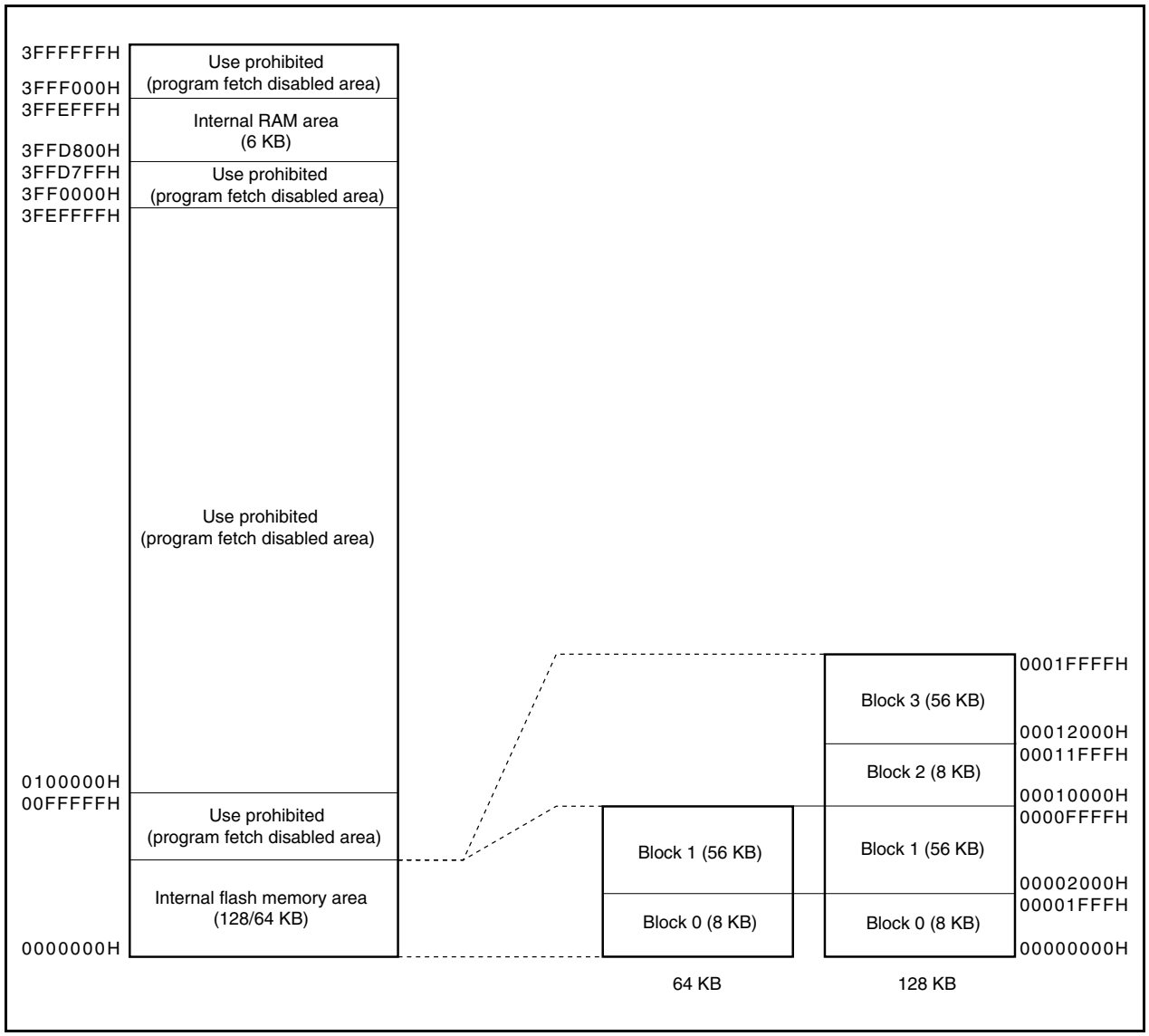
Note In the μPD70F3713, boot swap function cannot be used, but safe rewriting of the entire flash memory area by self programming is possible.

- Interrupts can be acknowledged during self programming.

18.2 Memory Configuration

The 128 KB/64 KB internal flash memory area is divided into 4/2 blocks and can be programmed/erased in block units. All the blocks can also be erased at once.

Figure 18-1. Flash Memory Mapping



18.3 Functional Overview

The internal flash memory of the V850ES/IE2 can be rewritten by using the rewrite function of the dedicated flash memory programmer, regardless of whether the V850ES/IE2 has already been mounted on the target system or not (on-board/off-board programming).

In addition, a security function that prohibits rewriting the user program written to the internal flash memory is also supported, so that the program cannot be changed by an unauthorized person.

The rewrite function using the user program (self programming (μPD70F3714 only)) is ideal for an application where it is assumed that the program is changed after production/shipment of the target system. A boot swap function (μPD70F3714 only) that rewrites the entire flash memory area safely is also supported. In addition, interrupt servicing is supported during self programming, so that the flash memory can be rewritten under various conditions, such as while communicating with an external device.

Table 18-1. Rewrite Method

Rewrite Method	Functional Outline	Operation Mode
On-board programming	Flash memory can be rewritten after the device is mounted on the target system, by using a dedicated flash memory 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 memory programmer and a dedicated program adapter board (FA series).	
Self programming (μPD70F3714 only)	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 internal flash memory area. Therefore, the rewrite program must be transferred to the internal RAM or external memory in advance).	Normal operation mode

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 18-2. Basic Functions

Function	Functional Outline	Support (√: Supported, ×: Not supported)	
		On-Board/Off-Board Programming	Self Programming
Block erasure	The contents of specified memory blocks are erased.	√	√
Chip erasure	The contents of the entire memory area are erased all at once.	√	×
Write	Writing to specified addresses, and a verify check to see if write level is secured are performed.	√	√
Verify/checksum	Data read from the flash memory is compared with data transferred from the flash memory programmer.	√	× (Can be read by user program)
Blank check	The erasure status of the entire memory is checked.	√	√
<R> Security setting	Use of the block erase command, chip erase command, program command, and read command can be prohibited, and rewriting boot area can be prohibited.	√	× (Supported only when setting is changed from enable to prohibit)

The following table lists the security functions. The block erase command prohibit, chip erase command prohibit, and program command prohibit functions are enabled by default after shipment, and security can be set by rewriting via on-board/off-board programming. Each security function can be used in combination with the others at the same time.

<R> **Table 18-3. Security Functions**

Function	Function Outline
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.
Chip erase command prohibit	Execution of block erase and chip erase commands on all the blocks is prohibited. Once prohibition is set, setting of prohibition cannot be initialized because the chip erase command cannot be executed.
Program command prohibit	Execution of program and block erase commands on all the blocks are prohibited. Setting of prohibition can be initialized by execution of the chip erase command.
Read command prohibit	Execution of read command on all the blocks is prohibited. Setting of prohibition can be initialized by execution of the chip erase command.
Boot area rewrite prohibit	Boot areas from block 0 to the specified last block can be protected. The protected boot area cannot be rewritten (erased and written). Setting of prohibition cannot be initialized by execution of the chip erase command.

<R>

Table 18-4. Security Setting

Function	Erase, Write, Read Operations When Each Security Is Set (√: Executable, ×: Not Executable, –: Not Supported)		Notes on Security Setting	
	On-Board/ Off-Board Programming	Self Programming	On-Board/ Off-Board Programming	Self Programming
Block erase command prohibit	Block erase command: × Chip erase command: √ Program command: √ Read command: √	Block erasure (FlashBlockErase): √ Chip erasure: – Write (FlashWordWrite): √ Read (FlashWordRead): √	Setting of prohibition can be initialized by chip erase command.	Supported only when setting is changed from enable to prohibit
Chip erase command prohibit	Block erase command: × Chip erase command: × Program command: √Note 1 Read command: √	Block erasure (FlashBlockErase): √ Chip erasure: – Write (FlashWordWrite): √ Read (FlashWordRead): √	Setting of prohibition cannot be initialized.	
Program command prohibit	Block erase command: × Chip erase command: √ Program command: × Read command: √	Block erasure (FlashBlockErase): √ Chip erasure: – Write (FlashWordWrite): √ Read (FlashWordRead): √	Setting of prohibition can be initialized by chip erase command.	
Read command prohibit	Block erase command: √ Chip erase command: √ Program command: √ Read command: ×	Block erasure (FlashBlockErase): √ Chip erasure: – Write (FlashWordWrite): √ Read (FlashWordRead): √	Setting of prohibition can be initialized by chip erase command.	
Boot area rewrite prohibit	Block erase command: √Note 2 Chip erase command: × Program command: √Note 2 Read command: √	Block erasure (FlashBlockErase): √ Chip erasure: – Write (FlashWordWrite): √ Read (FlashWordRead): √	Setting of prohibition cannot be initialized.	

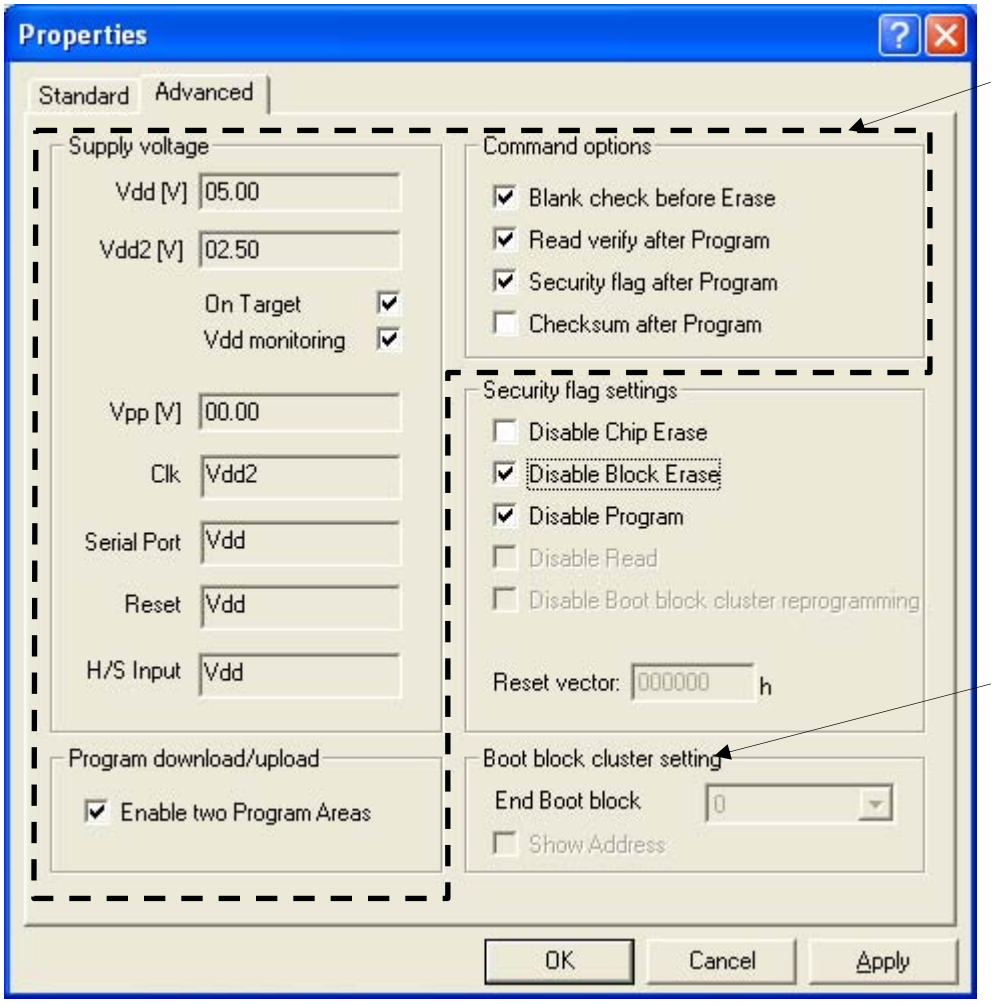
Notes 1. In this case, since the erase command is invalid, data different from the data already written in the flash memory cannot be written.

2. The boot area for which rewriting is prohibited is invalid.

<R> **(1) Security setting by PG-FP4 and PG-FP5 (Security flag settings)**

When disabling the read command (Disable Read), to raise the security level, it is recommended to also disable the block erase command (Disable Block Erase) and program command (Disable Program).

Furthermore, when rewriting program is not necessary similarly to the mask ROM versions, additionally disable the chip erase command (Disable Chip Erase).



Notes 1. Set “Supply voltage”, “Program download/upload”, and “Command options” in broken lines in accordance with the use conditions.

2. To disable rewriting the boot area (Boot block cluster setting), select “Disable Boot block cluster reprogramming” in “Security flag settings” and select the last block of the boot area for which rewriting is to be disabled.

18.4 Rewriting by Dedicated Flash Memory Programmer

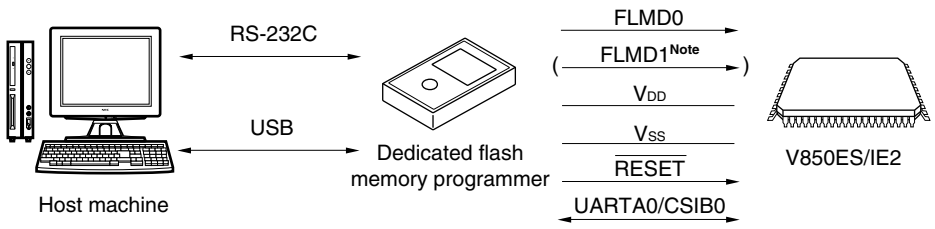
The flash memory can be rewritten by using a dedicated flash memory programmer after the V850ES/IE2 is mounted on the target system (on-board programming). The flash memory can also be rewritten before the device is mounted on the target system (off-board programming) by using a dedicated program adapter (FA series).

18.4.1 Programming environment

The following shows the environment required for writing programs to the flash memory of the V850ES/IE2.

Figure 18-2. Environment Required for Writing Programs to Flash Memory

<R>



Note Connect the FLMD1 pin to the flash memory programmer or connect to a GND via a pull-down resistor on the board.

A host machine is required for controlling the dedicated flash memory programmer.

UARTA0 or CSIB0 is used for the interface between the dedicated flash memory programmer and the V850ES/IE2 to perform writing, erasing, etc. A dedicated program adapter (FA series) required for off-board writing.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

18.4.2 Communication mode

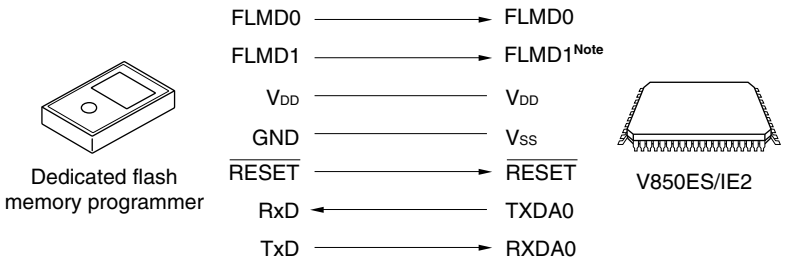
Communication between the dedicated flash memory programmer and the V850ES/IE2 is performed by serial communication using the UARTA0 or CSIB0 interfaces of the V850ES/IE2.

(1) UARTA0

Transfer rate: 9,600 to 153,600 bps

Figure 18-3. Communication with Dedicated Flash Memory Programmer (UARTA0)

<R>



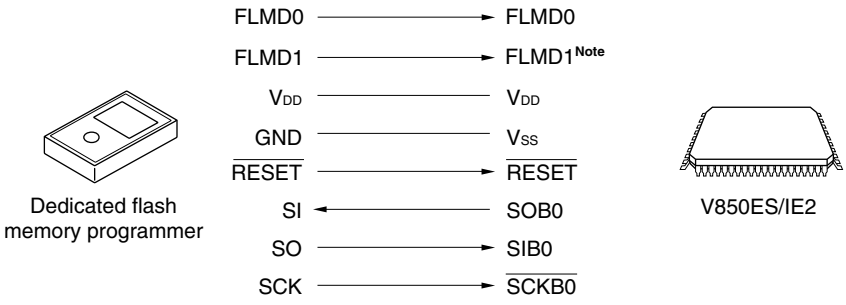
Note Connect the FLMD1 pin to the flash memory programmer or connect to a GND via a pull-down resistor on the board.

(2) CSIB0

Serial clock: 2.4 kHz to 2.5 MHz (MSB first)

Figure 18-4. Communication with Dedicated Flash Memory Programmer (CSIB0)

<R>



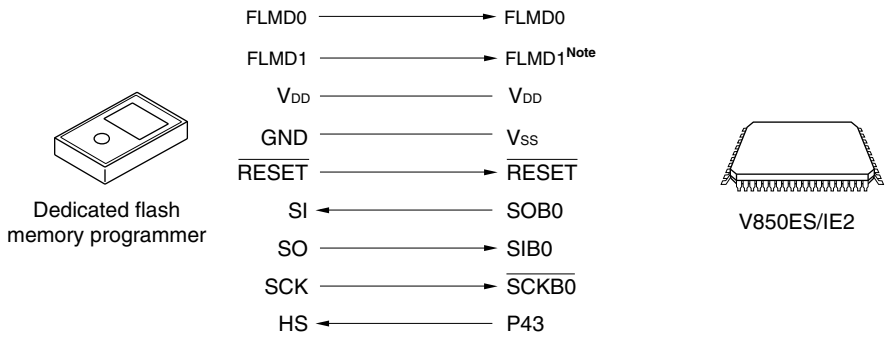
Note Connect the FLMD1 pin to the flash memory programmer or connect to a GND via a pull-down resistor on the board.

(3) CSIB0 + HS

Serial clock: 2.4 kHz to 2.5 MHz (MSB first)

Figure 18-5. Communication with Dedicated Flash Memory Programmer (CSIB0 + HS)

<R>



Note Connect the FLMD1 pin to the flash memory programmer or connect to a GND via a pull-down resistor on the board.

The dedicated flash memory programmer outputs the transfer clock, and the V850ES/IE2 operates as a slave.

<R> When the PG-FP4 or PG-FP5 is used as the dedicated flash memory programmer, it sends the following signals to

<R> the V850ES/IE2. For details, refer to the **PG-FP4 User's Manual (U15260E)** and **PG-FP5 User's Manual (U18865E)**.

<R> **Table 18-5. Signal Connections of Dedicated Flash Memory Programmer (PG-FP4 and PG-FP5)**

<R>

PG-FP4 and PG-FP5			V850ES/IE2	Processing for Connection		
Signal Name	I/O	Pin Function	Pin Name	UARTA0	CSIB0	CSIB0 + HS
FLMD0	Output	Write enable/disable	FLMD0	◎	◎	◎
FLMD1	Output	Write enable/disable	FLMD1	◎Note 1	◎Note 1	◎Note 1
VDD	–	V_{DD} voltage generation/voltage monitor	V_{DD}, EV_{DD}	◎Note 2	◎Note 2	◎Note 2
GND	–	Ground	V_{SS}	◎	◎	◎
CLKNote 3	Output	–	X1Note 3, X2Note 3	–	–	–
$\overline{RESET}$	Output	Reset signal	$\overline{RESET}$	◎	◎	◎
SI/RxD	Input	Receive signal	SOB0/TXDA0	◎	◎	◎
SO/TxD	Output	Transmit signal	SIB0/RXDA0	◎	◎	◎
SCK	Output	Transfer clock	$\overline{SCKB0}$	×	◎	◎
HS	Input	Handshake signal for CSIB0 + HS communication	P43	×	×	◎

Notes 1. Wire these pins as shown in Figure 18-6, or connect them to GND via a pull-down resistor on board.

<R> **2.** Connect this pin when power is supplied from PG-FP4 and PG-FP5.
It does not have to be connected if an on-board power supply is used.

3. In the V850ES/IE2, external clock input is prohibited. Create an oscillator on board and supply the clock via that oscillator.

Remark ◎: Must be connected.
×: Does not have to be connected.

Table 18-6. Wiring of V850ES/IE2 Flash Writing Adapters (FA-64GC-8BS-A)

<R>

Flash Memory Programmer (PG-FP4 and PG-FP5) Connection Pin			Name of FA Board Pin	CSIB0 + HS Used		CSIB0 Used		UARTA0 Used	
Signal Name	I/O	Pin Function		Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	SI	P41/SOB0	39	P41/SOB0	39	P31/TXDA0	43
SO/TxD	Output	Transmit signal	SO	P40/SIB0	40	P40/SIB0	40	P30/RXDA0	44
SCK	Output	Transfer clock	SCK	P42/$\overline{\text{SCKB0}}$	38	P42/$\overline{\text{SCKB0}}$	38	Not needed	–
CLK[Note]	Output	Clock to V850ES/IE2	X1[Note]	X1[Note]	6	X1[Note]	6	X1[Note]	6
			X2[Note]	X2[Note]	7	X2[Note]	7	X2[Note]	7
/RESET	Output	Reset signal	/RESET	$\overline{\text{RESET}}$	5	$\overline{\text{RESET}}$	5	$\overline{\text{RESET}}$	5
FLMD0	Input	Write voltage	FLMD0	FLMD0	25	FLMD0	25	FLMD0	25
FLMD1	Input	Write voltage	FLMD1	PDL5/FLMD1	30	PDL5/FLMD1	30	PDL5/FLMD1	30
HS	Input	Handshake signal for CSI0 + HS	RESERVE/ HS	P43/TOP00/ TIP00	37	Not needed	–	Not needed	–
VDD	–	VDD voltage generation/ voltage monitor	VDD	V_{DD}	9	V_{DD}	9	V_{DD}	9
				EV_{DD}	26, 47	EV_{DD}	26, 47	EV_{DD}	26, 47
				AV_{REF0}	64	AV_{REF0}	64	AV_{REF0}	64
				AV_{REF1}	59	AV_{REF1}	59	AV_{REF1}	59
				AV_{DD0}	63	AV_{DD0}	63	AV_{DD0}	63
				AV_{DD1}	60	AV_{DD1}	60	AV_{DD1}	60
GND	–	Ground	GND	V_{SS}	8	V_{SS}	8	V_{SS}	8
				EV_{SS}	27, 48	EV_{SS}	27, 48	EV_{SS}	27, 48
				AV_{SS0}	62	AV_{SS0}	62	AV_{SS0}	62
				AV_{SS1}	61	AV_{SS1}	61	AV_{SS1}	61

Note The clock cannot be supplied from the CLK pin of the flash memory programmer. Create an oscillator on the board and supply the clock via that oscillator.

Caution Be sure to connect the REGC pin in either of the following ways.

- **Connect to GND via 4.7 μF capacitor**
- **Connect directly to V_{DD}**

Figure 18-6. Wiring Example of V850ES/IE2 Flash Writing Adapter (FA-64GC-8BS-A) (1/2)

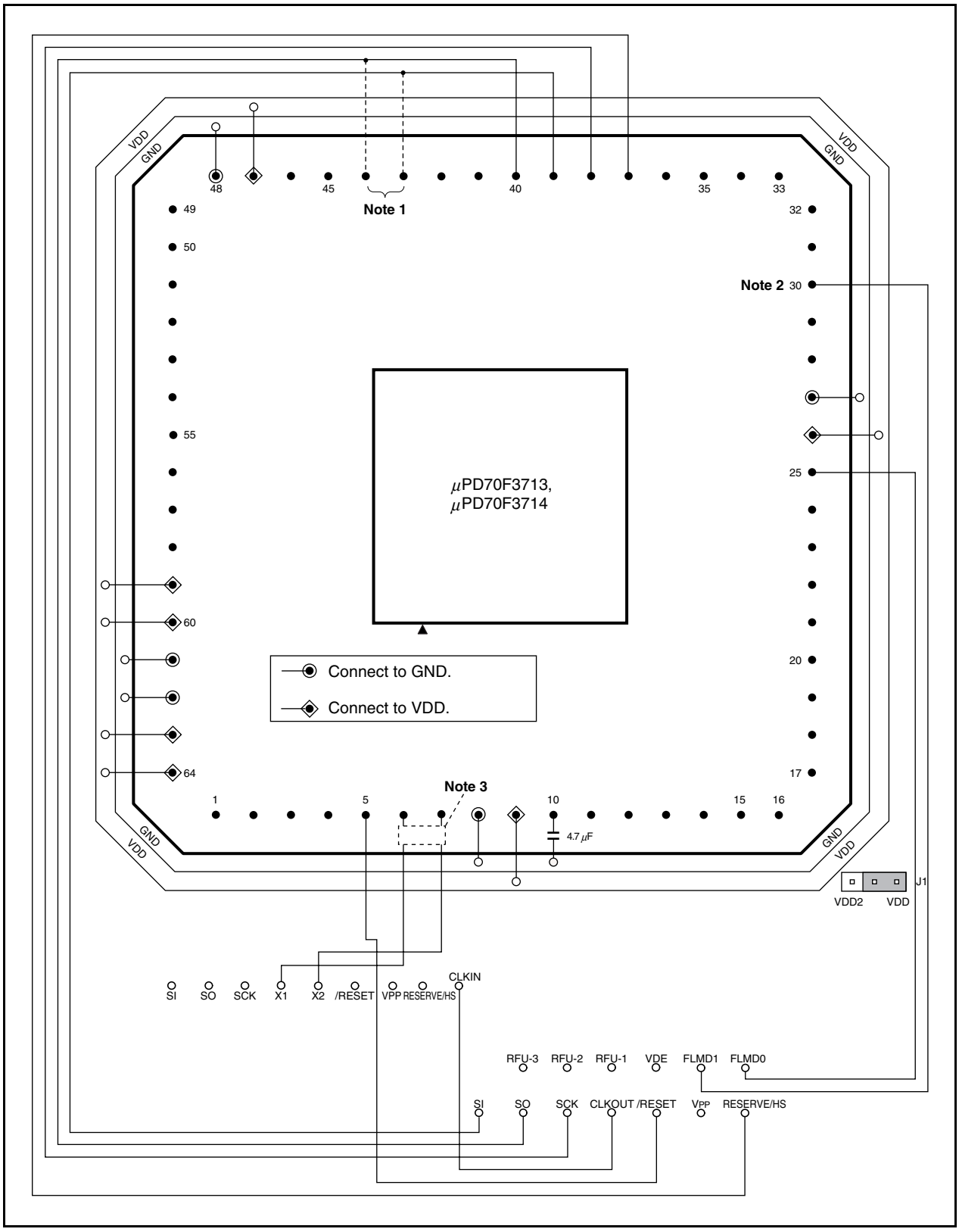


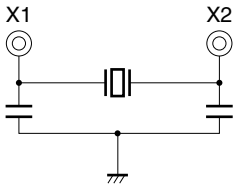
Figure 18-6. Wiring Example of V850ES/IE2 Flash Writing Adapter (FA-64GC-8BS-A) (2/2)

Notes 1. Corresponding pins when UARTA0 is used.

2. Wire the FLMD1 pin as shown below, or connect it to GND on board via a pull-down resistor.

3. Create an oscillator on the flash writing adapter (shown in broken lines) and supply a clock. Here is an example of the oscillator.

Example:



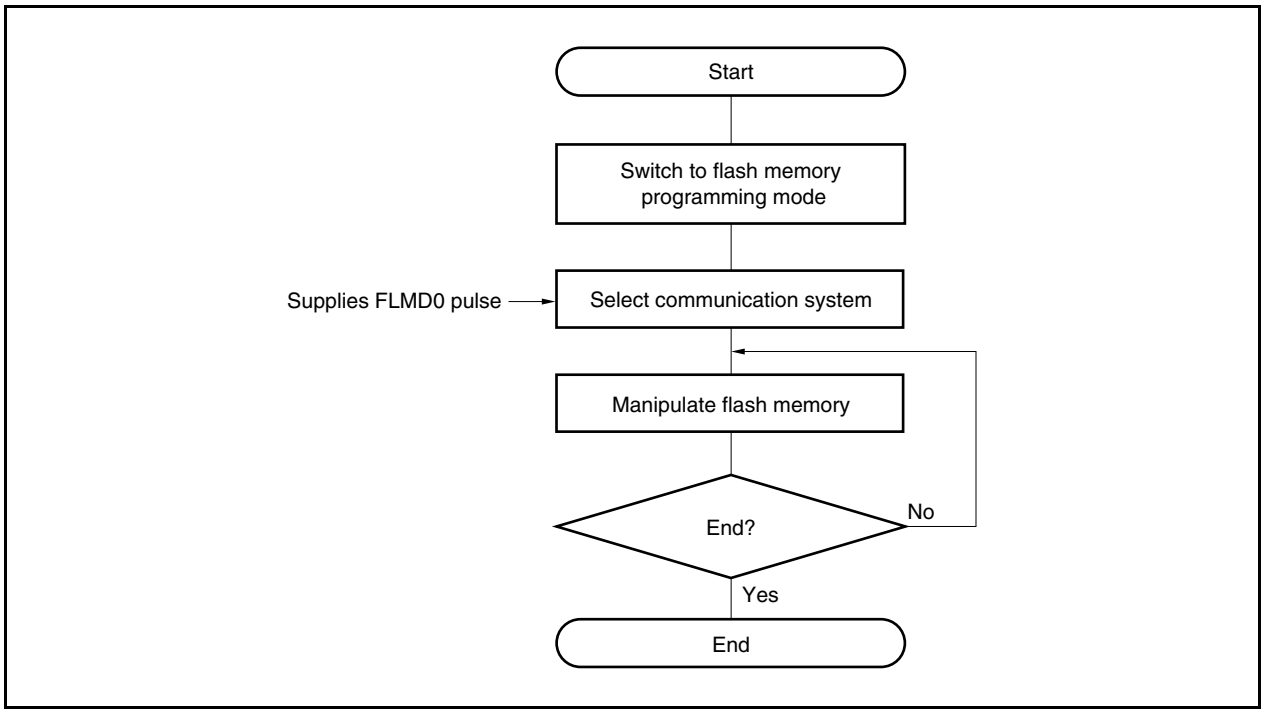
Remarks 1. Process the pins not shown in accordance with the handling of unused pins (see **2.2 Pin I/O Circuit Types and Connection of Unused Pins**). It is recommended to use a resistor of 1 kΩ to 10 kΩ.

2. This figure shows the wiring when CSIB0 supporting handshaking is used.

18.4.3 Flash memory control

The following shows the procedure for manipulating the flash memory.

<R> **Figure 18-7. Procedure for Manipulating Flash Memory**

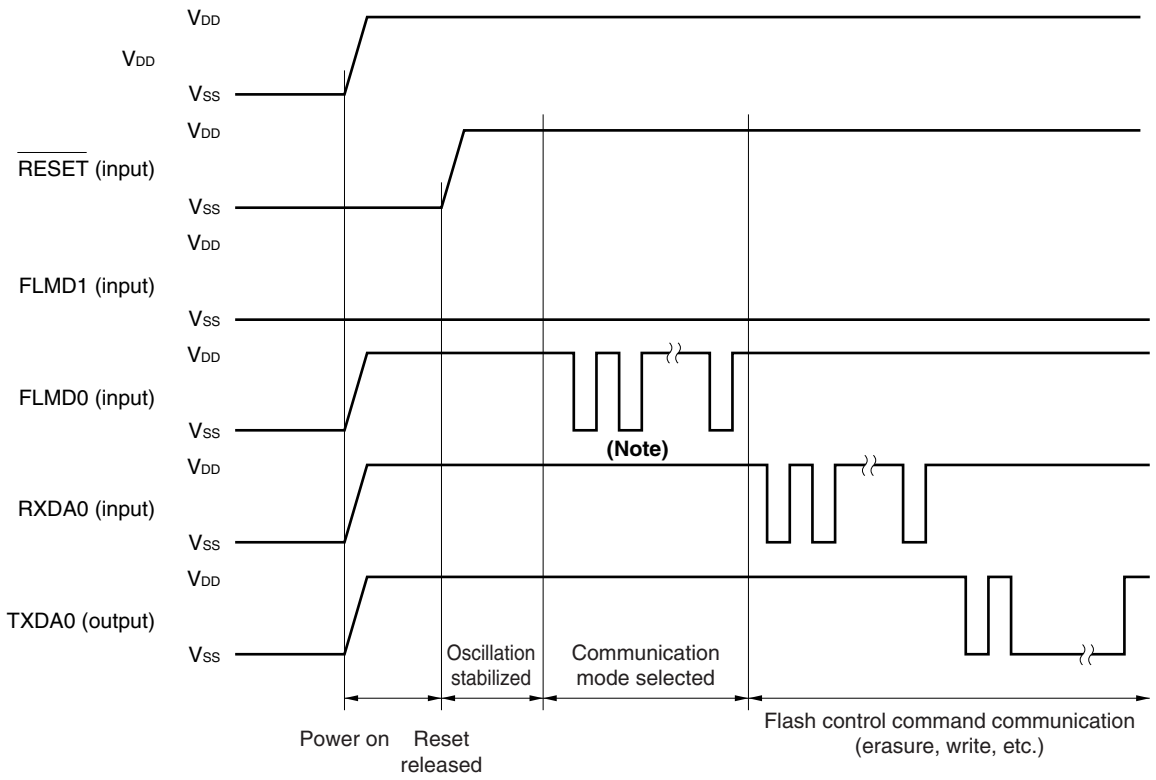


18.4.4 Selection of communication mode

In the V850ES/IE2, 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 dedicated flash memory programmer.

The following shows the relationship between the number of pulses and the communication mode.

Figure18-8. Selection of 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	V850ES/IE2 performs slave operation, MSB first
11	CSIB0 + HS	V850ES/IE2 performs slave operation, MSB first
Other	RFU	Setting prohibited

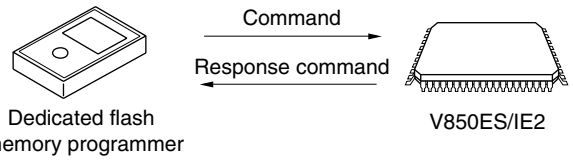
Caution When UARTA0 is selected, the receive clock is calculated based on the reset command sent from the dedicated flash memory programmer after receiving the FLMD0 pulse.

18.4.5 Communication commands

The V850ES/IE2 communicates with the dedicated flash memory programmer by means of commands. The signals sent from the dedicated flash memory programmer to the V850ES/IE2 are called “commands”. The response signals sent from the V850ES/IE2 to the dedicated flash memory programmer are called “response commands”.

Figure 18-9. Communication Commands

<R>



The following shows the commands for flash memory control in the V850ES/IE2. All of these commands are issued from the dedicated flash memory programmer, and the V850ES/IE2 performs the processing corresponding to the commands.

Table 18-7. Flash Memory Control Commands

Classification	Command Name	Support			Function
		CSIB0	CSIB0 + HS	UARTA0	
Blank check	Block blank check command	√	√	√	Checks if the contents of the memory in the specified block have been correctly erased.
Erase	Chip erase command	√	√	√	Erases the contents of the entire memory.
	Block erase command	√	√	√	Erases the contents of the memory of the specified block.
<R> Write	Program command	√	√	√	Writes the specified address range, and executes a contents verify check.
Verify	Verify command	√	√	√	Compares the contents of memory in the specified address range with data transferred from the flash memory programmer.
	Checksum command	√	√	√	Reads the checksum in the specified address range.
System setting, control	Silicon signature command	√	√	√	Reads silicon signature information.
<R>	Security setting command	√	√	√	Disables the chip erase command, block erase command, program command and read command, and disables rewriting to the boot area.

18.4.6 Pin connection

When performing on-board writing, mount a connector on the target system to connect to the dedicated flash memory programmer. Also, incorporate a function on-board to switch from the normal operation mode to the flash memory programming mode.

In the flash memory programming mode, all the pins not used for flash memory programming become the same status as that immediately after reset. Therefore, pin handling is required when the external device does not acknowledge the status immediately after a reset.

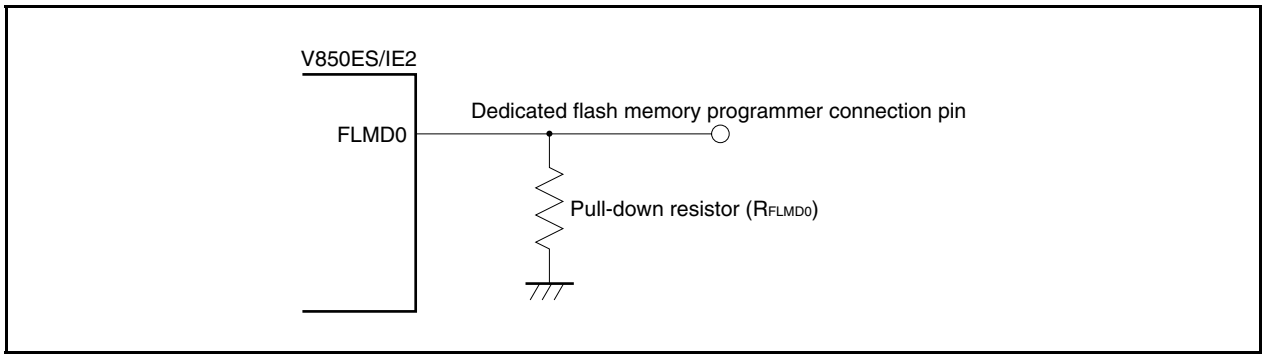
(1) FLMD0 pin

In the normal operation mode, input a voltage of V_{SS} level to the FLMD0 pin. In the flash memory programming mode, supply a write voltage of V_{DD} level to the FLMD0 pin.

Because the FLMD0 pin serves as a write protection pin in the self programming mode (μPD70F3714 only), a voltage of V_{DD} level must be supplied to the FLMD0 pin via port control, etc., before writing to the flash memory.

For details, see **18.5.5 (1) FLMD0 pin**.

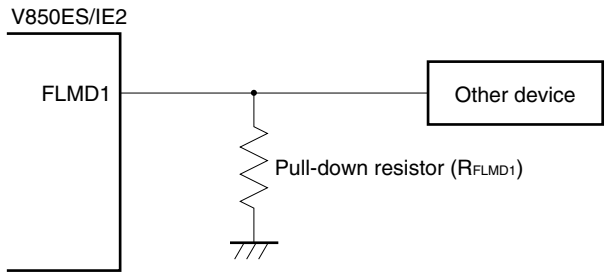
Figure 18-10. FLMD0 Pin Connection Example



(2) FLMD1 pin

When 0 V is input to the FLMD0 pin, the FLMD1 pin does not function. When V_{DD} 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.

Figure 18-11. FLMD1 Pin Connection Example



Caution If the V_{DD} signal is input to the FLMD1 pin from another device during on-board writing and immediately after reset, isolate this signal.

Table 18-8. Relationship Between FLMD0 and FLMD1 Pins and Operation Mode When Reset Is Released

FLMD0	FLMD1	Operation Mode
0	Don't care	Normal operation mode
V_{DD}	0	Flash memory programming mode
V_{DD}	V_{DD}	Setting prohibited

(3) Serial interface pin

The following shows the pins used by each serial interface.

Table 18-9. Pins Used by Serial Interfaces

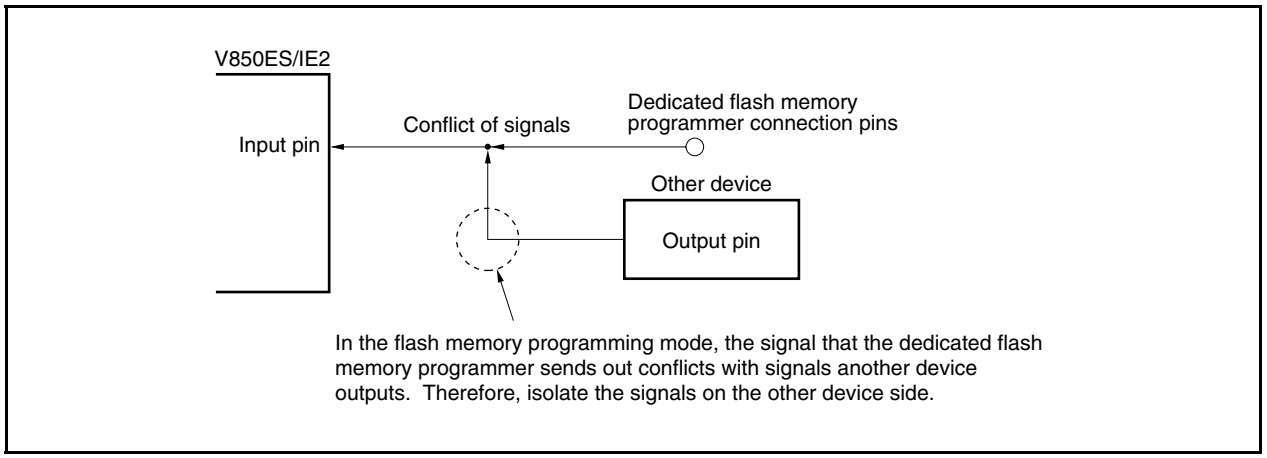
Serial Interface	Pins Used
UARTA0	TXDA0, RXDA0
CSIB0	SOB0, SIB0, $\overline{\text{SCKB0}}$
CSIB0 + HS	SOB0, SIB0, $\overline{\text{SCKB0}}$, P43

When connecting a dedicated flash memory programmer to a serial interface pin that is connected to another device on-board, care should be taken to avoid conflict of signals and malfunction of the other device.

(a) Conflict of signals

When the dedicated flash memory programmer (output) is connected to a serial 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.

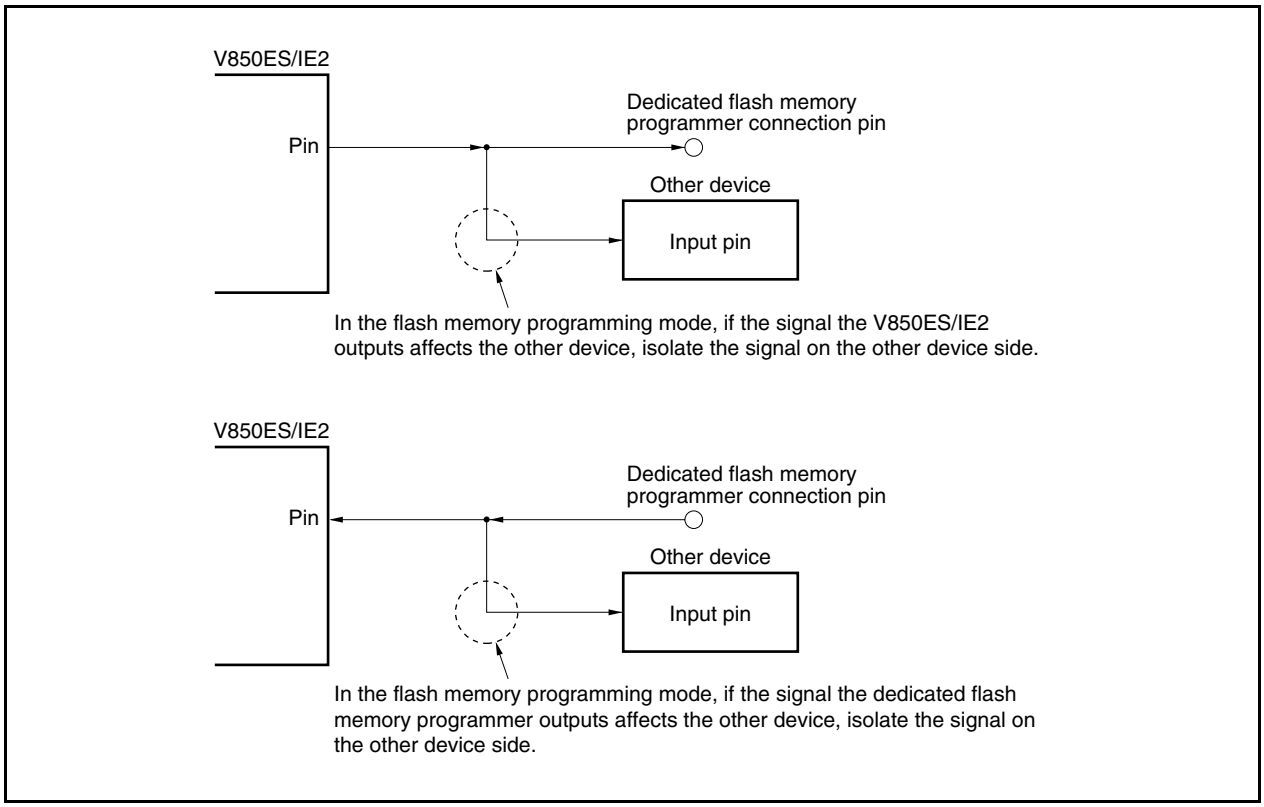
Figure 18-12. Conflict of Signals (Serial Interface Input Pin)



(b) Malfunction of other device

When the dedicated flash memory programmer (output or input) is connected to a serial 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.

Figure 18-13. Malfunction of Other Device

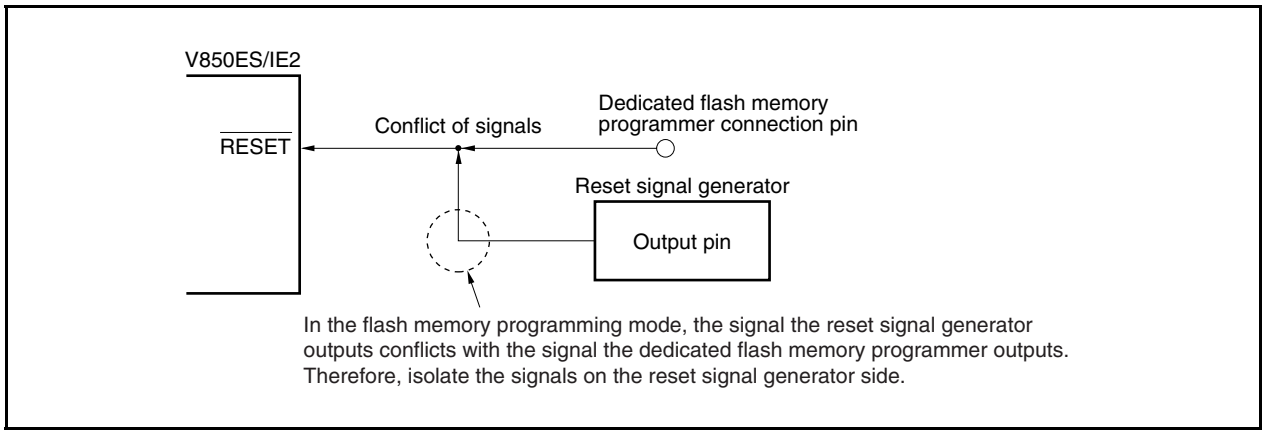


(4) $\overline{\text{RESET}}$ pin

When the reset signals of the dedicated flash memory programmer are 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 dedicated flash memory programmer.

Figure 18-14. Conflict of Signals ($\overline{\text{RESET}}$ Pin)



(5) Port pins

When the system shifts to the flash memory programming mode, all the pins that are not used for flash memory programming are in the same status as that immediately after reset. If the external device connected to each port does not recognize the status of the port immediately after reset, pins require appropriate processing, such as connecting to V_{DD} via a resistor or connecting to V_{SS} via a resistor.

(6) Other signal pins

Connect X1, X2, and REGC in the same status as that in the normal operation mode.

(7) Power supply

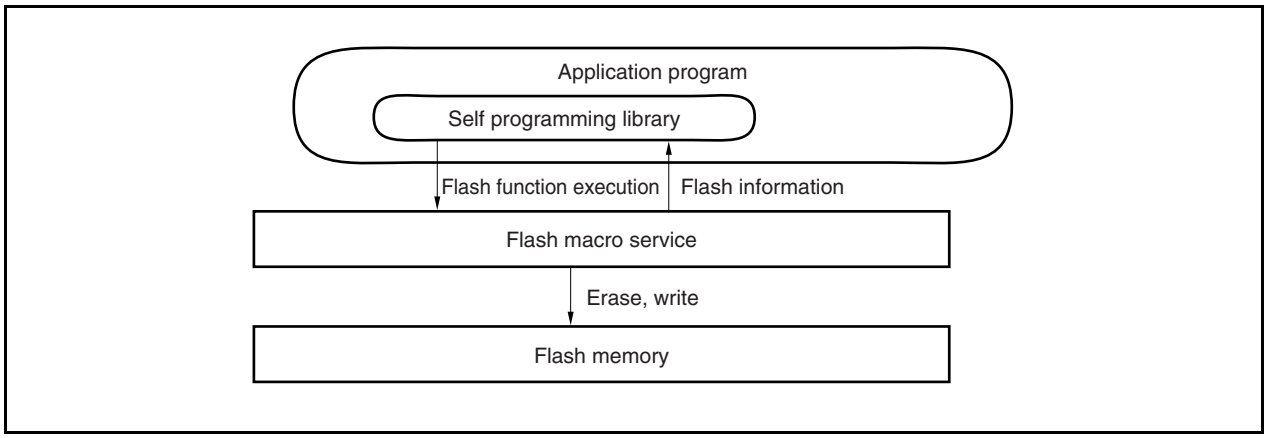
Supply the same power (V_{DD}, V_{SS}, EV_{DD}, EV_{SS}, AV_{SS0}, AV_{SS1}, AV_{DD0}, AV_{DD1}, AV_{REF0}, AV_{REF1}) as in normal operation mode.

18.5 Rewriting by Self Programming (μPD70F3714 only)

18.5.1 Overview

The V850ES/IE2 supports a flash macro service that allows the user program to rewrite the internal flash memory by itself. By using this interface and a self programming library that is used to rewrite the flash memory with a user application program, the flash memory can be rewritten by a user application transferred in advance to the internal RAM or external memory. Consequently, the user program can be upgraded and constant data can be rewritten in the field.

Figure 18-15. Concept of Self Programming

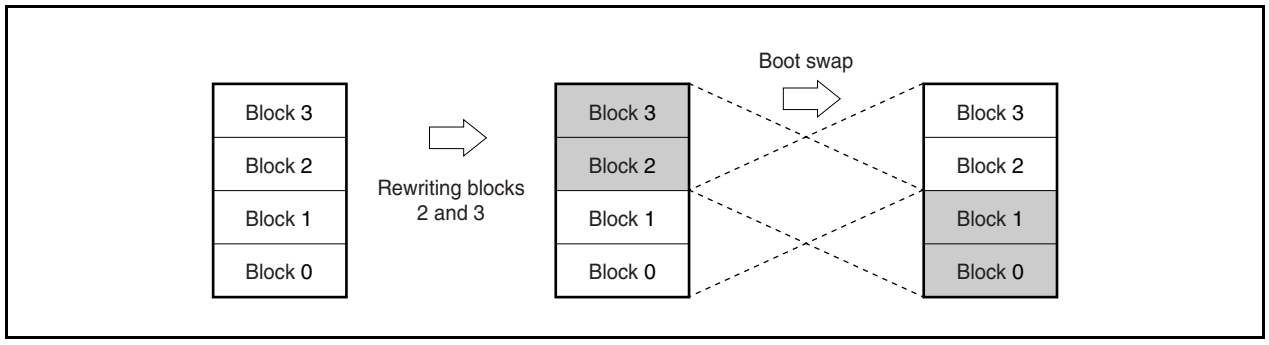


18.5.2 Features

(1) Secure self programming (boot swap function) (μPD70F3714 only)

The μPD70F3714 supports a boot swap function that can exchange the physical memory of blocks 0 and 1 with the physical memory of blocks 2 and 3. By writing the start program to be rewritten to blocks 2 and 3 in advance and then swapping the physical memory, the entire area can be safely rewritten even if a power failure occurs during rewriting because the correct user program always exists in blocks 0 and 1.

Figure 18-16. Rewriting Entire Memory Area (Boot Swap)



(2) Interrupt support

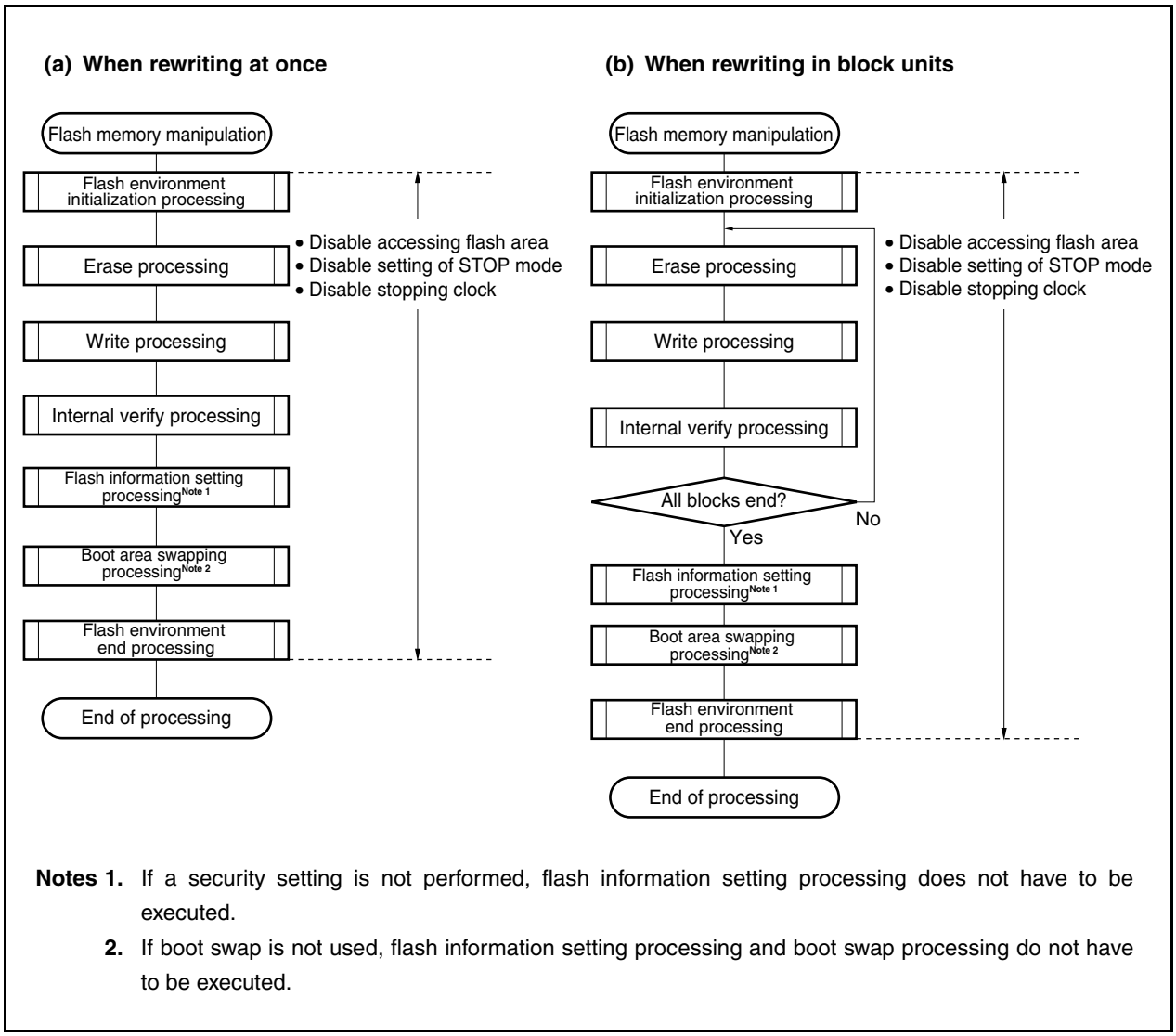
Instructions cannot be fetched from the flash memory during self programming. Conventionally, therefore, a user handler written to the flash memory could not be used even if an interrupt occurred. With the V850ES/IE2, a user handler can be registered to an entry RAM area by using a library function, so that interrupt servicing can be performed by internal RAM or external memory execution.

18.5.3 Standard self programming flow

The entire processing to rewrite the flash memory by flash self programming is illustrated below.

<R>

Figure 18-17. Standard Self Programming Flow



Notes 1. If a security setting is not performed, flash information setting processing does not have to be executed.

2. If boot swap is not used, flash information setting processing and boot swap processing do not have to be executed.

18.5.4 Flash functions

<R>

Table 18-10. Flash Function List

Function Name	Outline	Support
FlashEnv	Initialization of flash control macro	√
FlashBlockErase	Erasure of only specified one block	√
FlashWordWrite	Writing from specified address	√
FlashBlockIVerify	Internal verification of specified block	√
FlashBlockBlankCheck	Blank check of specified block	√
FlashFLMDCheck	Check of FLMD pin	√
FlashGetInfo	Reading of flash information	√
FlashSetInfo	Setting of flash information	√
FlashBootSwap[Note]	Swapping of boot area	√
FlashWordRead	Reading data from specified address	√

Note μPD70F3714 only

Remark For details, refer to the **V850 Series Flash Memory Self Programming (Single Power Supply Flash Memory) User's Manual**.
Contact an NEC Electronics sales representative for the above manual.

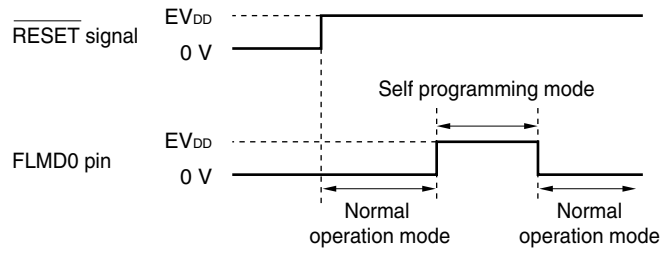
18.5.5 Pin processing

(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_{DD} 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 18-18. Mode Change Timing



Caution Make sure that the FLMD0 pin is at 0 V when reset is released.

18.5.6 Internal resources used

The following table lists the internal resources used for self programming. These internal resources can also be used freely for purposes other than self programming.

Table 18-11. Internal Resources Used

Resource Name	Description
Entry RAM area (internal RAM/external RAM size[Note])	Routines and parameters used for the flash macro service are located in this area. The entry program and default parameters are copied by calling a library initialization function.
Stack area (stack size[Note])	An extension of the stack used by the user is used by the library (can be used in both the internal RAM and external RAM).
Library code (code size[Note])	Program entity of library (can be used anywhere other than the flash memory block to be manipulated).
Application program	Executed as user application. Calls flash functions.
Maskable interrupt	Can be used in user application execution status or self programming status. To use this interrupt in the self programming status, the interrupt servicing start address must be registered in advance by a registration function.

Note For details, refer to the **V850 Series Flash Memory Self Programming (Single Power Supply Flash Memory) User's Manual**.

Contact an NEC Electronics sales representative for the above manual.

Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}	$V_{DD} = EV_{DD}$		−0.5 to +6.5	V
	V_{SS}	$V_{SS} = EV_{SS} = AV_{SS0} = AV_{SS1}$		−0.5 to +0.5	V
	EV_{DD}	$V_{DD} = EV_{DD}$		−0.5 to +6.5	V
	EV_{SS}	$V_{SS} = EV_{SS} = AV_{SS0} = AV_{SS1}$		−0.5 to +0.5	V
	AV_{DD}	$EV_{DD} = AV_{DD0} = AV_{DD1}$		−0.5 to +6.5	V
	AV_{SS}	$V_{SS} = EV_{SS} = AV_{SS0} = AV_{SS1}$		−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 VRO Note 3 + 0.5	V
Output current, low	I_{OL}	P20 to P25	Per pin	18	mA
		Pins other than P20 to P25	Per pin	4	mA
		P00 to P06, P10 to P14, P16, P17, P30 to P33, P40 to P44, PDL0 to PDL7	Total of all pins	50	mA
		P20 to P27	Total of all pins	50	mA
Output current, high	I_{OH}	All pins	Per pin	−4.0	mA
		P00 to P06, P10 to P14, P16, P17, P30 to P33, P40 to P44, PDL0 to PDL7	Total of all pins	−40	mA
		P20 to P27	Total of all pins	−15	mA
Analog input voltage	V_{IAN}	ANI00 to ANI03, ANI10 to ANI13		−0.5 to AV_{DD} + 0.5 Note 2	V
Analog reference input voltage	V_{IREF}	AV_{REF0}, AV_{REF1}		−0.5 to AV_{DD} + 0.5 Note 2	V
Operating ambient temperature	T_A	At normal operation		−40 to +85	°C
		At flash memory programming		−40 to +85	°C
Storage temperature	T_{stg}			−40 to +125	°C

Notes 1. P00 to P06, P10 to P14, P16, P17, P20 to P27, P30 to P33, P40 to P44, PDL0 to PDL7, $\overline{\text{RESET}}$, FLMD0

2. Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

3. VRO: Regulator output voltage (2.5 V (TYP.))

Cautions 1. Do not directly connect the output (or I/O) pins of IC products to each other, or to V_{DD}, EV_{DD}, and GND. Open-drain pins or open-collector pins, however, can be directly connected to each other. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

The ratings and conditions indicated for DC characteristics and AC characteristics represent the quality assurance range during normal operation.

Capacitance (T_A = 25°C, V_{DD} = V_{SS} = EV_{DD} = EV_{SS} = AV_{DD0} = AV_{SS0} = AV_{DD1} = AV_{SS1} = 0 V)

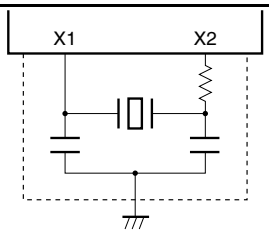
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	f_c = 1 MHz Unmeasured pins returned to 0 V.			15	pF
I/O capacitance	C_{IO}				15	pF
Output capacitance	C_O				15	pF

Operating Conditions (T_A = –40 to +85°C, V_{SS} = EV_{SS} = AV_{SS0} = AV_{SS1} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
System clock frequency	f_{XX}	PLL mode	20		20	MHz
		Clock through mode	2.5		2.5	MHz
CPU clock frequency	f_{CPU}	PLL mode	2.5		20	MHz
		Clock through mode	0.3125		2.5	MHz
V_{DD}, EV_{DD} voltage	V_{DD}, EV_{DD}		3.5		5.5	V
AV_{DD0}, AV_{DD1} voltage	AV_{DD}	Operation is not guaranteed when EV_{DD} is 4.5 V or less.	4.5		EV_{DD}	V

Clock Oscillator Characteristics

(T_A = –40 to +85°C, V_{DD} = 3.5 to 5.5 V (Internal V_{DD} = 2.5 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator/ crystal resonator		Oscillation frequency (f_X)		2.5		2.5	MHz
		Oscillation stabilization time	After reset release		$2^{16}/f_X$		ms
			After STOP mode release		**Note**		ms

Note The value differs depending on the settings of the oscillation stabilization time select register (OSTS).

Cautions 1. Keep the oscillator as close to the X1 and X2 pins as possible.

2. Do not route signal lines through the area enclosed by broken lines.

3. The duty factor of the oscillation waveform must be within 45% to 55%.

4. Inputting an external clock to the V850ES/IE2 is prohibited.

<R> **(i) Murata Mfg. Co., Ltd.: Ceramic resonator (T_A = –40 to +85°C)**

Type	Part Number	Oscillation Frequency f_X (MHz)	Recommended Circuit Constant			Oscillation Voltage Range	
			C1	C2	Rd (kΩ)	MIN. (V)	MAX. (V)
Surface mounting	CSTCC2M50G56-R0	2.5	On chip (47 pF)	On chip (47 pF)	1	V_{POC0}	5.5

Caution These oscillator constants are reference values based on evaluation under a specific environment by the resonator manufacturer. When optimization of the oscillator characteristics on the actual application is necessary, request evaluation on the mounting circuit from the resonator manufacturer.

The oscillation voltage and oscillation frequency indicate only oscillator characteristics, therefore use the V850E/IE2 within the DC characteristics and AC characteristics for internal operation conditions.

DC Characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, AV_{DD0} = AV_{DD1} = 4.5 to 5.5 V, V_{SS} = EV_{SS} = AV_{SS0} = AV_{SS1} = 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
Input voltage, low	V_{IL1}	**Note 1**			EV_{SS}		0.3EV_{DD}	V
	V_{IL2}	**Note 2**			EV_{SS}		0.2EV_{DD}	V
Input leakage current, high	I_{LIH1}	V_I = V_{DD} = EV_{DD}		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 = V_{DD} = EV_{DD}					5	μA
Output leakage current, low	I_{LOL}	V_O = 0 V					–5	μA
Output voltage, high	V_{OH}	**Note 3**	Per pin I_{OH} = –1.0 mA	Total of pins –31 mA	EV_{DD} – 1.0			V
		Note 4	Per pin I_{OH} = –1.0 mA	Total of pins –8 mA	EV_{DD} – 1.0			V
Output voltage, low	V_{OL1}	**Note 5**, **Note 6**	Per pin I_{OL} = 15 mA	Total of pins 45 mA			2.0	V
	V_{OL2}	**Note 3**	Per pin I_{OL} = 1.0 mA	Total of pins 31 mA			0.4	V
		Note 4	Per pin I_{OL} = 1.0 mA	Total of pins 8 mA			0.4	V
Pull-up resistor	R_1				10	30	100	kΩ

Notes 1. P20 to P27, P31, P33, P41, and PDL0 to PDL7 pins

2. P00 to P06, P10 to P14, P16, P17, P30, P32, P40, P42 to P44, $\overline{\text{RESET}}$, and FLMD0 pins

3. P00 to P06, P10 to P14, P16, P17, P30 to P33, P40 to P44, and PDL0 to PDL7 pins

4. P20 to P27 pins

5. P20 to P25 pins

6. Up to three pins can output a low level at the same time (the other three must output a high level or high impedance).

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

2. When the I_{OH} and I_{OL} conditions are not satisfied for a pin but the total value of all pins is satisfied, only that pin does not satisfy the DC characteristics.

DC Characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, AV_{DD0} = AV_{DD1} = 4.5 to 5.5 V, V_{SS} = EV_{SS} = AV_{SS0} = AV_{SS1} = 0 V) (2/2)

	Parameter	Symbol	Conditions	MIN.	TYP.[Note 2]	MAX.	Unit
<R>	Supply current[Note 1]	I_{DD1}	Normal operation, f_{XX} = 20 MHz		38	60	mA
<R>		I_{DD2}	HALT mode, f_{XX} = 20 MHz		22	35	mA
<R>		I_{DD3}	IDLE mode, f_{XX} = 20 MHz		6	10	mA
		I_{DD4}	STOP mode		30	200	μA

Notes 1. Current flowing through the output buffer and pull-up resistor is not included.

2. The TYP. value is a reference value at V_{DD} = 5.0 V.

Data Retention Characteristics

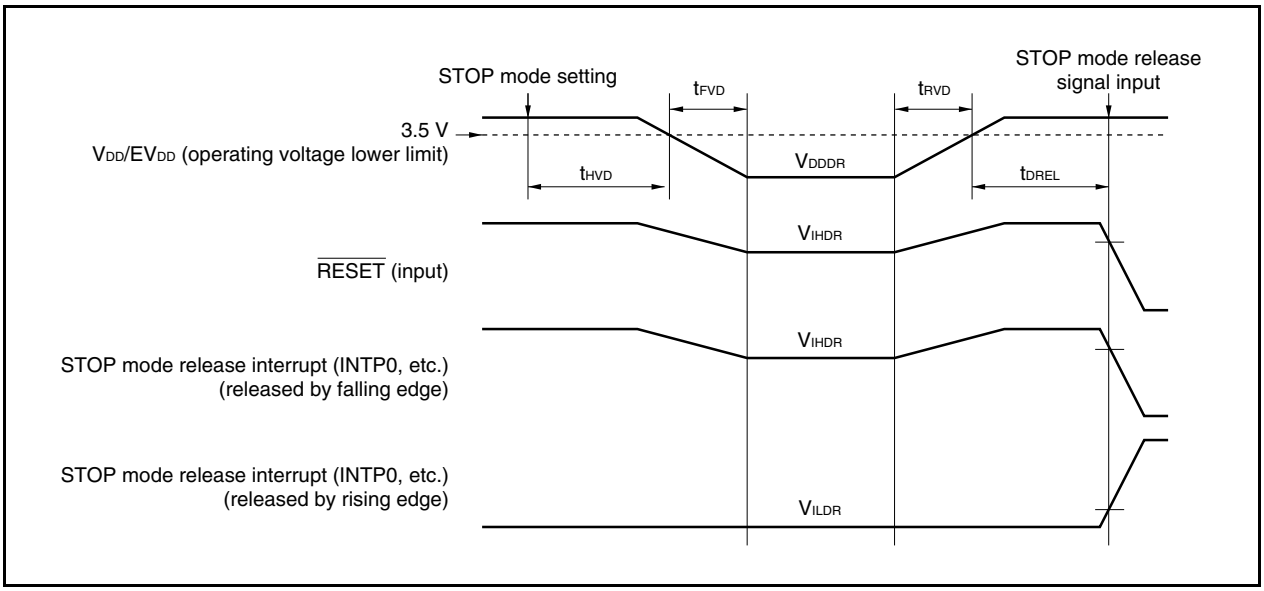
STOP Mode (T_A = –40 to +85°C, V_{SS} = EV_{SS} = AV_{SS0} = AV_{SS1} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Data retention voltage	V_{DDDR}	In STOP mode	V_{DD}, EV_{DD}	**Note**		5.5	V
		After reset other than power supply	V_{DD}, EV_{DD}	1.9		**Note**	V
Data retention current	I_{DDDR}	V_{DD} = V_{DDDR}			30	200	μA
Supply voltage rise time	t_{RVD}	V_{DD}		1			μs
Supply voltage fall time	t_{FVD}	V_{DD}		1			μs
Supply voltage retention time (for STOP mode setting)	t_{HVD}			0			ms
STOP mode release signal input time	t_{DREL}			0			ms
Data retention input voltage, high	V_{IHDR}	All input pins		0.9V_{DDDR}		V_{DDDR}	V
Data retention input voltage, low	V_{ILDR}	All input pins		0		0.1V_{DDDR}	V

Note When reset mode (LVIM.LVIMD bit = 0) of low-voltage detector (LVI) is not used:
POC detection voltage (V_{POC0})
When reset mode (LVIM.LVIMD bit = 1) of low-voltage detector (LVI) is used:
LVI detection voltage (V_{LVI0}/V_{LVI1})

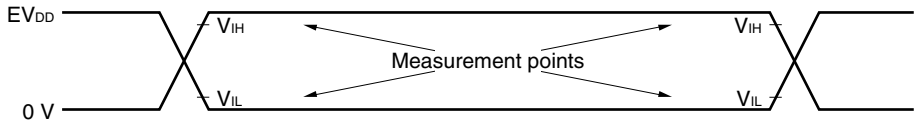
Caution Shifting to STOP mode and restoring from STOP mode must be performed within the rated operating range.
- **When LVI (reset mode) is not used: POC detection voltage (V_{POC0}) or higher**
- **When LVI (reset mode) is used: LVI detection voltage (V_{LVI0}/V_{LVI1}) or higher**

If the voltage level reaches the POC detection voltage (V_{POC0}) when LVI (reset mode) is not used or to LVI detection voltage (V_{LVI0}/V_{LVI1}) when LVI (reset mode) is used, the STOP mode is automatically released because internal reset (POCRES/LVIRES) is effected.

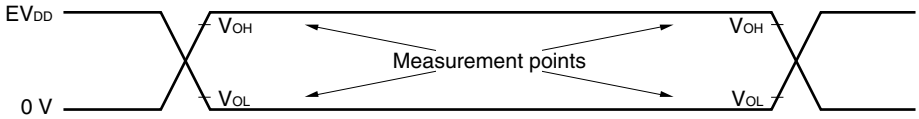


AC Characteristics

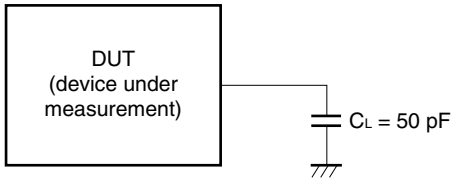
AC Test Input Measurement Points



AC Test Output Measurement Points



Load Conditions

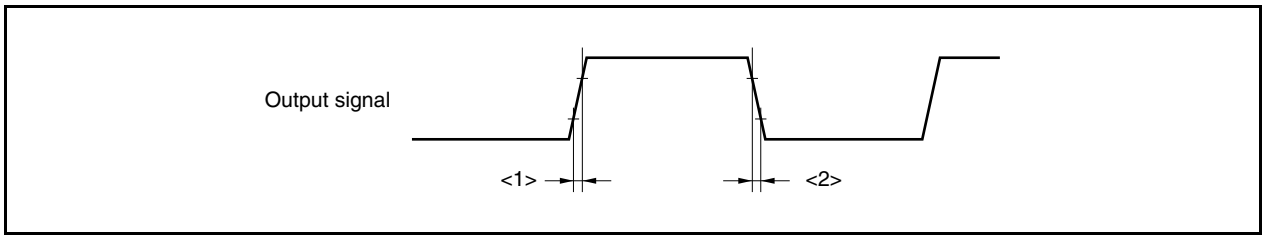


Caution If the load capacitance exceeds 50 pF due to the circuit configuration, bring the load capacitance of the device to 50 pF or less by inserting a buffer or by some other means.

Output Signal Timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, V_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output rise time	t_{OR}	<1>			15	ns
Output fall time	t_{OF}	<2>			15	ns

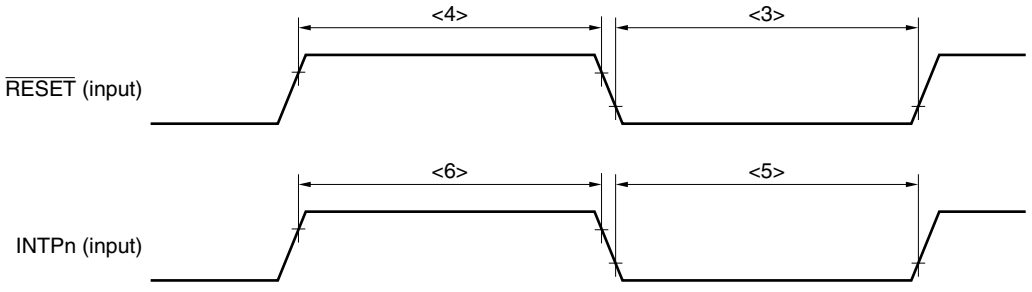


Reset, External Interrupt Timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, V_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ low-level width	t_{WRSL}	<3>		500		ns
$\overline{\text{RESET}}$ high-level width	t_{WRSH}	<4>		500		ns
INTPn low-level width	t_{WITL}	<5>	n = 0 to 5 (analog noise elimination)	500		ns
			n = 6 (digital noise elimination)	$5T_{smp}$ + 10		ns
INTPn high-level width	t_{WITH}	<6>	n = 0 to 5 (analog noise elimination)	500		ns
			n = 6 (digital noise elimination)	$5T_{smp}$ + 10		ns

Remark T_{smp}: Noise elimination sampling clock cycle (set by INTPNRC register)



Remark n = 0 to 6

Timer Timing

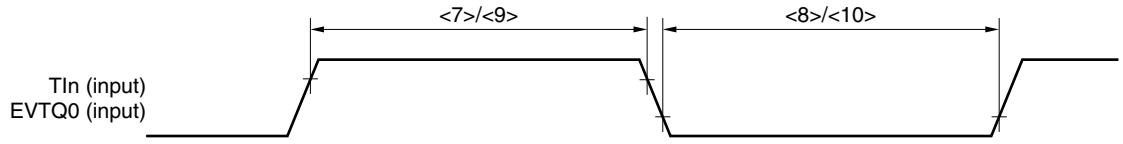
(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, V_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
TIn high-level width[Note 1]	t_{WTIHn}	<7>		10T + 10		ns
TIn low-level width[Note 1]	t_{WTILn}	<8>		10T + 10		ns
EVTQ0 high-level width[Note 1]	t_{WEVHn}	<9>		10T + 10		ns
EVTQ0 low-level width[Note 1]	t_{WEVLn}	<10>		10T + 10		ns

Note T = 1/f_{XX}

Remarks 1. n = Q00, Q01, Q02, Q03, P00, P01, P20, P21

2. 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.



Remark n = Q00, Q01, Q02, Q03, P00, P01, P20, P21

High-Impedance Control Timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, AV_{DD0} = AV_{DD1} = 4.5 to 5.5 V, V_{SS} = EV_{SS} = AV_{SS0} = AV_{SS1} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Oscillation stop → timer output high impedance	t_{CLM}	At clock monitor operation		65	μs
TOQH0OFF input → timer output high impedance	t_{HTQH0}			300	ns
TOQ1OFF input → timer output high impedance	t_{HTQ1}			300	ns
TOPnOFF input → timer output high impedance	t_{HTPn}			300	ns

Remark n = 2, 3

CSIB Timing

(1) Master mode

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, V_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

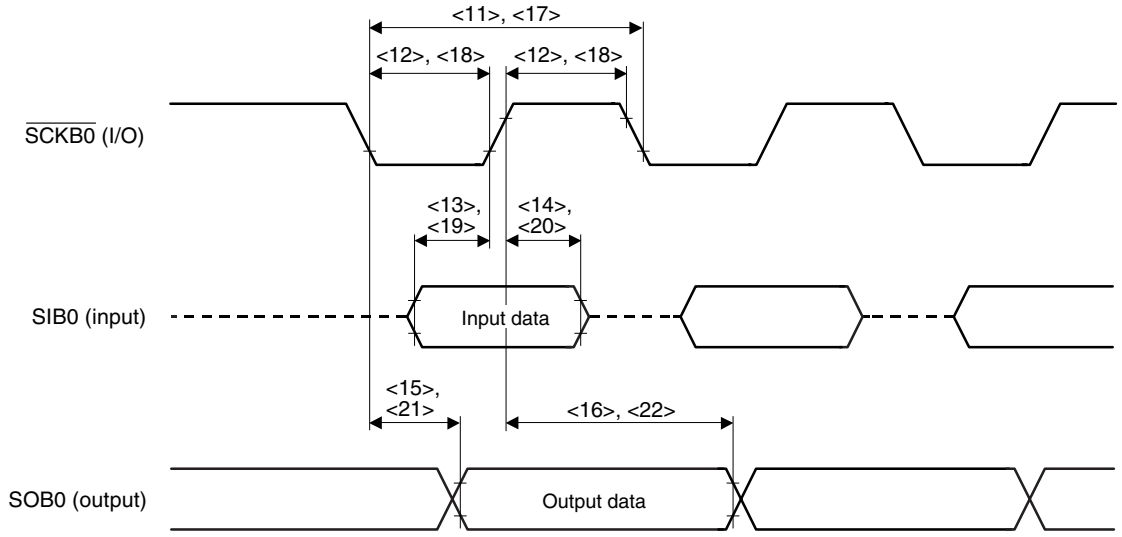
	Parameter	Symbol		Conditions	MIN.	MAX.	Unit
	$\overline{SCKB0}$ cycle	t_{KCYM}	<11>		125		ns
	$\overline{SCKB0}$ high-/low-level width	t_{KWHM}, t_{KWLM}	<12>		$t_{KCYM}/2 - 10$		ns
	SIB0 setup time (to $\overline{SCKB0}\uparrow$)	t_{SSIM}	<13>		30		ns
	SIB0 hold time (from $\overline{SCKB0}\uparrow$)	t_{HSIM}	<14>		30		ns
	SOB0 output delay time (from $\overline{SCKB0}\downarrow$)	t_{DSOM}	<15>			30	ns
	SOB0 output delay time (from $\overline{SCKB0}\uparrow$)					30	ns
<R>	SOB0 output hold time (from $\overline{SCKB0}\uparrow$)	t_{HSOM}	<16>		$t_{KCYM}/2 - 10$		ns
<R>	SOB0 output hold time (from $\overline{SCKB0}\downarrow$)				$t_{KCYM}/2 - 10$		ns

(2) Slave mode

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, V_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

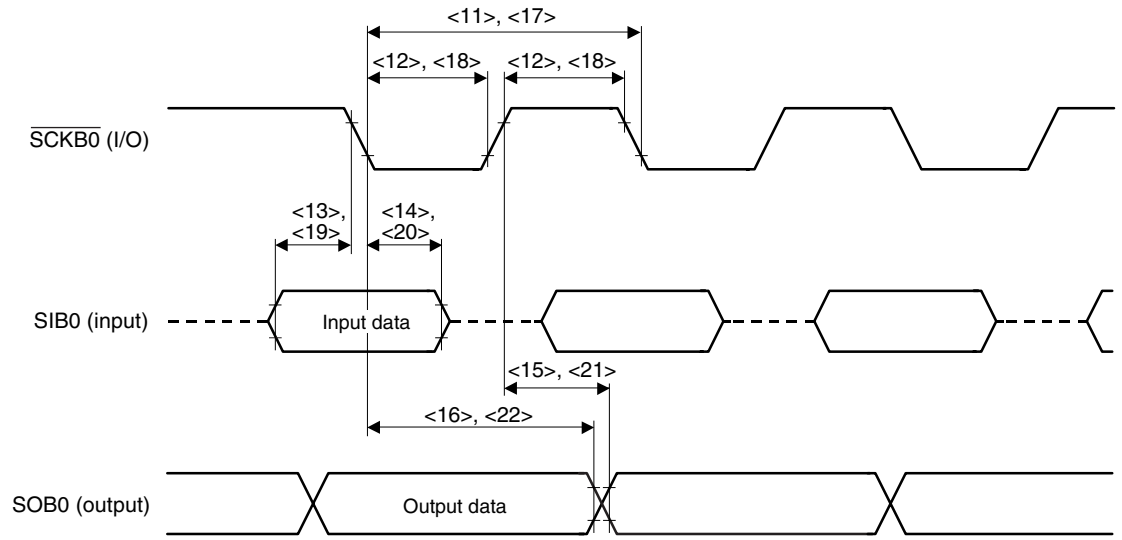
	Parameter	Symbol		Conditions	MIN.	MAX.	Unit
	$\overline{SCKB0}$ cycle	t_{KCYS}	<17>		125		ns
	$\overline{SCKB0}$ high-/low-level width	t_{KWHS}, t_{KWLS}	<18>		$t_{KCYS}/2 - 10$		ns
	SIB0 setup time (to $\overline{SCKB0}\uparrow$)	t_{SSIS}	<19>		30		ns
	SIB0 hold time (from $\overline{SCKB0}\uparrow$)	t_{HSIS}	<20>		30		ns
	SOB0 output delay time (from $\overline{SCKB0}\downarrow$)	t_{DSOS}	<21>			30	ns
	SOB0 output delay time (from $\overline{SCKB0}\uparrow$)					30	ns
<R>	SOB0 output hold time (from $\overline{SCKB0}\uparrow$)	t_{HSOS}	<22>		$t_{KCYS}/2 - 10$		ns
<R>	SOB0 output hold time (from $\overline{SCKB0}\downarrow$)				$t_{KCYS}/2 - 10$		ns

CSIB Timing: CB0CTL1.CB0CKP, CB0CTL1.CB0DAP Bits = 00



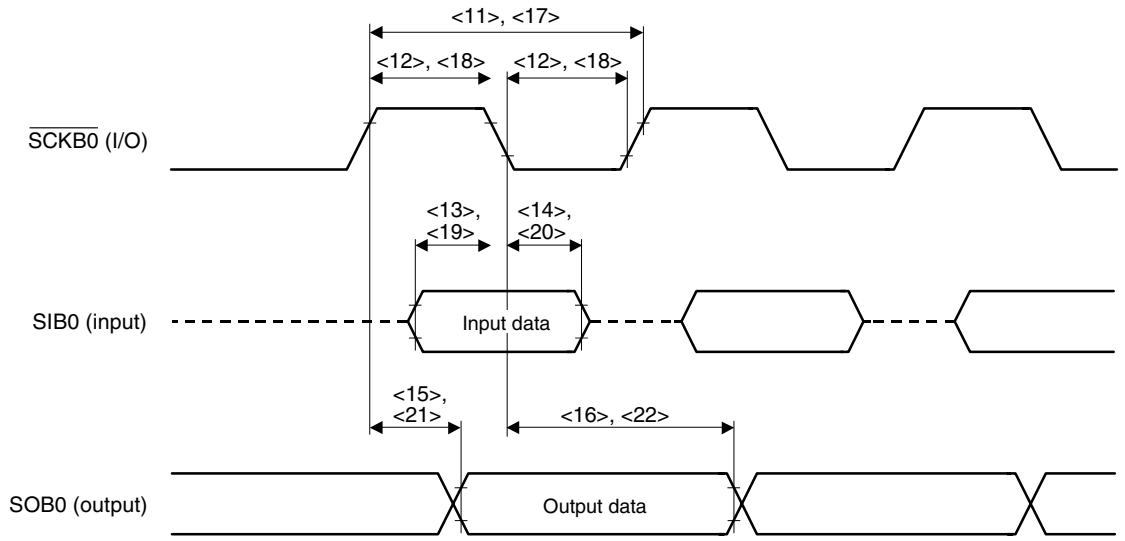
Remark The broken lines indicate high impedance.

CSIB Timing: CB0CTL1.CB0CKP, CB0CTL1.CB0DAP Bits = 01



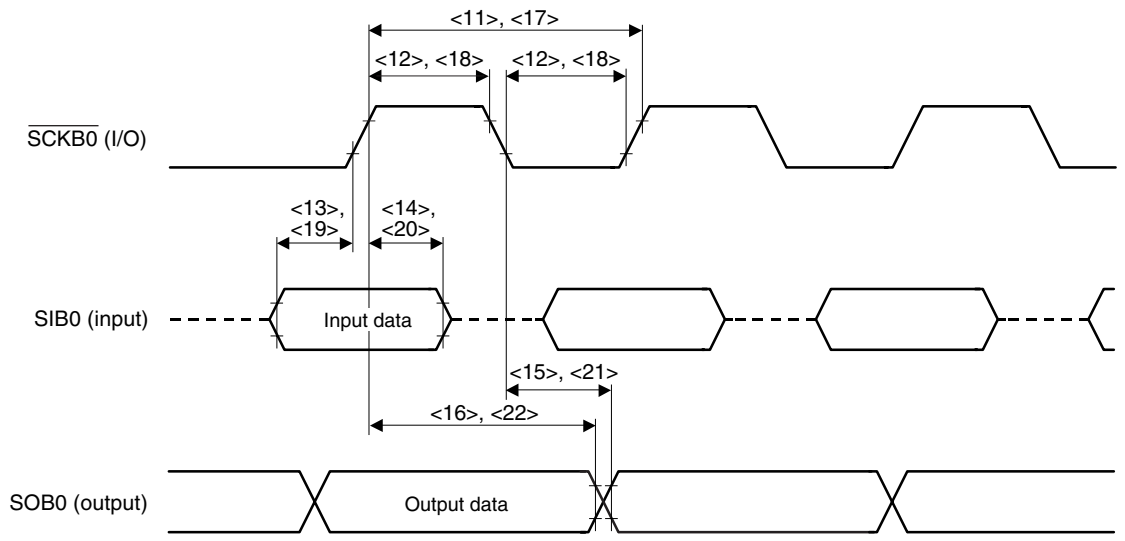
Remark The broken lines indicate high impedance.

CSIB Timing: CB0CTL1.CB0CKP, CB0CTL1.CB0DAP Bits = 10



Remark The broken lines indicate high impedance.

CSIB Timing: CB0CTL1.CB0CKP, CB0CTL1.CB0DAP Bits = 11



Remark The broken lines indicate high impedance.

Characteristics of A/D Converters 0, 1

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 4.5 to 5.5 V, AV_{DD0} = AV_{DD1} = 4.5 to 5.5 V, V_{SS} = EV_{SS} = AV_{SS0} = AV_{SS1} = 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}		2		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]					–1 to +2	LSB
Analog reference voltage	AV_{REF}	AV_{REF0} = AV_{REF1} = AV_{DD0} = AV_{DD1}	4.5		5.5	V
Analog input voltage	V_{IAN}		AV_{SS}		AV_{DD}	V
AV_{DD0}, AV_{DD1} supply current[Note 2]	AI_{DD}	During operation		6	10	mA
<R>	AI_{DDS}	In STOP mode[Note 3]		0.5	25	μA

Notes 1. Excluding quantization error (±0.5 LSB).

2. This value is one cycle of A/D converter 0 or A/D converter 1.

3. Stop the operation of A/D converters 0 and 1 (ADAnM0.ADAnCE bit = 0) before setting to the STOP mode.

Remarks 1. LSB: Least Significant Bit

2. n = 0, 1

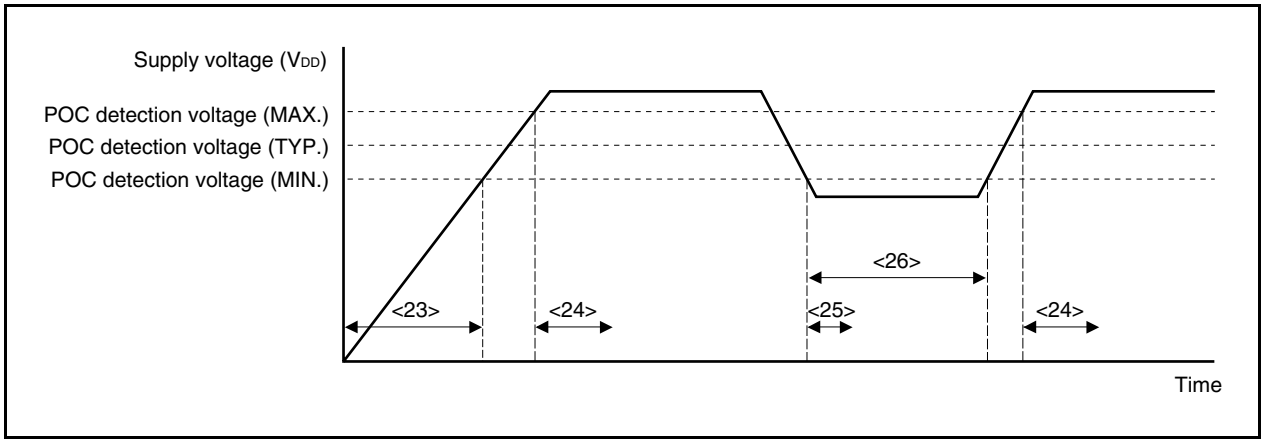
Power-on-Clear Circuit (POC)

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, V_{SS} = EV_{SS} = 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}	<23>	V_{DD} = 0 to 3.5 V	0.002			ms
Response time 1[Note 1]	t_{PTHD}	<24>	After V_{DD} reaches POC detection voltage at power-on.			3.0	ms
Response time 2[Note 2]	t_{PD}	<25>	After V_{DD} drops to POC detection voltage at power-off.			1.0	ms
Minimum V_{DD} width	t_{PW}	<26>		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.



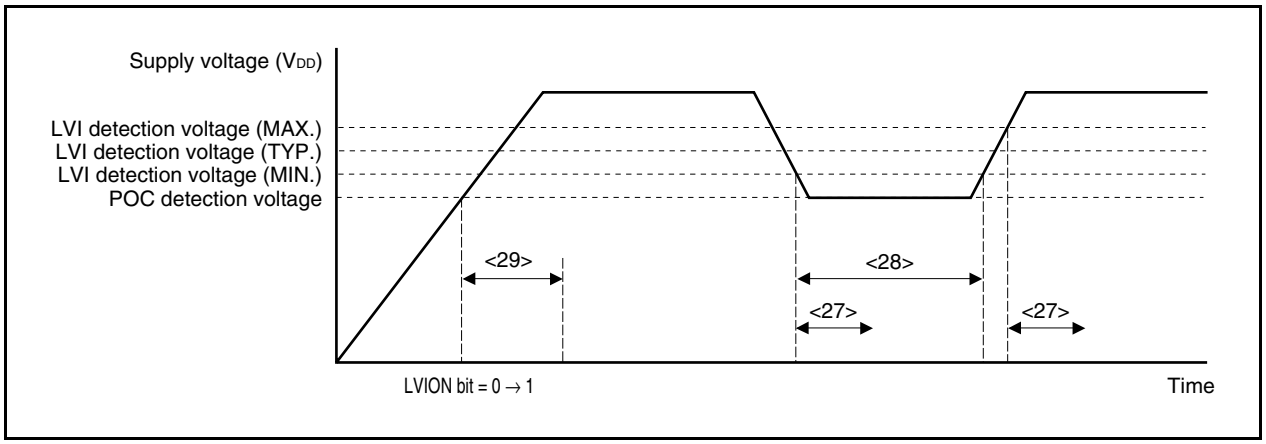
Low-Voltage Detector (LVI)

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, V_{SS} = EV_{SS} = 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}	<27>	After V_{DD} reaches V_{LVI0}/V_{LVI1} (MAX.) or drops to V_{LVI0}/V_{LVI1} (MIN.)		0.2	2.0	ms
Minimum V_{DD} width	t_{LW}	<28>		0.2			ms
<R> Reference voltage stabilization wait time	t_{LWAIT}	<29>	After V_{DD} reaches to POC detection voltage, and after LVIM.LVION is changed from 0 to 1.		0.1	0.2	ms

Note The time required to output an interrupt request signal (INTLVI) or internal reset signal (LVIRES) after the detection voltage is detected.

<R>

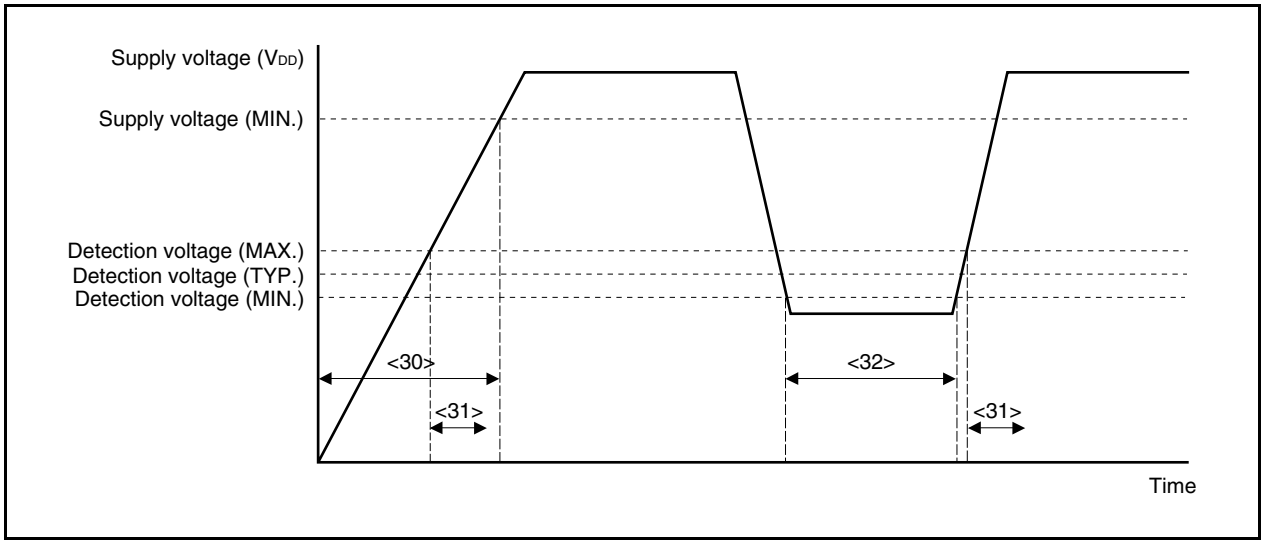


RAM Retention Voltage Detection

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, V_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{RAMH}			1.9	2.0	2.1	V
Supply voltage rise time	t_{RAMHTH}	<30>	V_{DD} = 0 to 3.5 V	0.002		1800	ms
Response time 1[Note]	t_{RAMHD}	<31>	After V_{DD} reaches 2.1 V		0.2	2.0	ms
Minimum V_{DD} width	t_{RAMHW}	<32>		0.2			ms

Note Time required to set the RAMS.RAMF bit to 1 after the detection voltage is detected.



Flash Memory Programming Characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.5 to 5.5 V, V_{SS} = EV_{SS} = 0 V, C_L = 50 pF)

(1) Basic characteristics

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operating frequency	f_{CPU}		2.5		20	MHz
Supply voltage	V_{DD}		3.5		5.5	V
Rewrite time	C_{ERWR}	**Note**			100	Times
Rewrite temperature	t_{PRG}		–40		85	°C

Note When writing initially to shipped products, it is counted as one rewrite for both “erase to write” and “write only”.

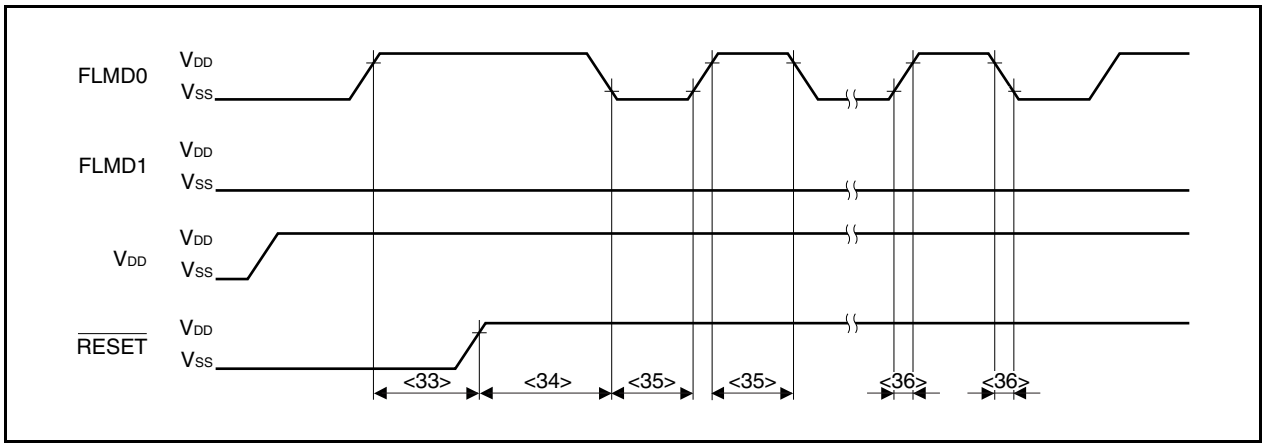
Example (P: Write, E: Erase)

Shipped product ⟶P→E→P→E→P: 3 rewrites

Shipped product → E→P→E→P→E→P: 3 rewrites

(2) Serial write operation characteristics

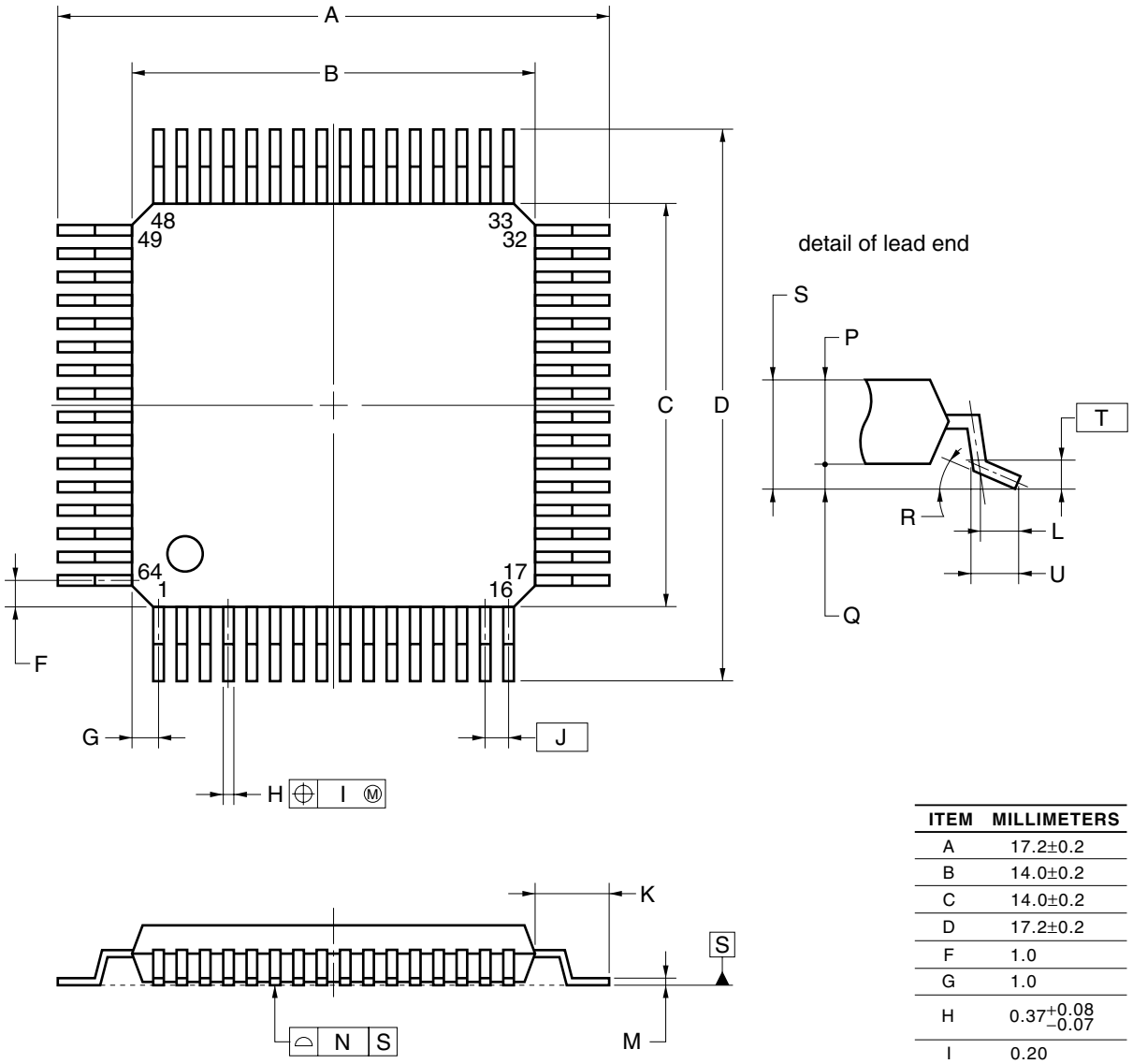
Parameter	Symbol		Conditions	MIN.	TYP.	MAX.	Unit
FLMD0, FLMD1 setup time	t_{MDSET}	<33>		2		3000	s
FLMD0 pulse input start time from $\overline{\text{RESET}}$↑	t_{RFCF}	<34>		30560/f_X + oscillation stabilization time			ns
FLMD0 pulse width	t_{CH}/t_{CL}	<35>		10	100		μs
FLMD0 pulse rise/fall time	t_R/t_F	<36>				50	ns



<R> **(3) Programming characteristics**

Parameter	Symbol	Conditions	TYP.	TYP.-Worst	MAX.	Unit
Block erase		f_{XX} = 20 MHz, 8 KB	0.78		5.78	s
		f_{XX} = 20 MHz, 56 KB	3.9		35.0	s
Write (128 bytes)		f_{XX} = 20 MHz	5.3		43.1	ms
Block internal verify		f_{XX} = 20 MHz, 8 KB	80.3		3000	ms
		f_{XX} = 20 MHz, 56 KB	562		3000	ms
Block blank check		f_{XX} = 20 MHz, 8 KB	35.5		44.4	ms
		f_{XX} = 20 MHz, 56 KB	249		311	ms
Flash information setting		f_{XX} = 20 MHz	1.32		10.8	ms

64-PIN PLASTIC LQFP (14x14)



ITEM	MILLIMETERS
A	17.2±0.2
B	14.0±0.2
C	14.0±0.2
D	17.2±0.2
F	1.0
G	1.0
H	$0.37^{+0.08}_{-0.07}$
I	0.20
J	0.8 (T.P.)
K	1.6±0.2
L	0.8
M	$0.17^{+0.03}_{-0.06}$
N	0.10
P	1.4±0.1
Q	0.127±0.075
R	$3^{\circ}{}^{+4^{\circ}}_{-3^{\circ}}$
S	1.7 MAX.
T	0.25
U	0.886±0.15

P64GC-80-8BS

NOTE

Each lead centerline is located within 0.20 mm of its true position (T.P.) at maximum material condition.

<R>

CHAPTER 21 RECOMMENDED SOLDERING CONDITIONS

These products should be soldered and mounted under the following recommended conditions.
For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 21-1. Surface Mounting Type Soldering Conditions

μPD70F3713GC-8BS-A: 64-pin plastic LQFP (14 × 14)
μPD70F3714GC-8BS-A: 64-pin plastic LQFP (14 × 14)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: 3 times or less, Exposure limit: 3 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	IR60-203-3
Wave soldering	For details, contact an NEC Electronics sales representative.	–
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Remarks 1. Products with -A at the end of the part number are lead-free products.

2. For soldering methods and conditions other than those recommended above, please contact an NEC Electronics sales representative.

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

(1/5)

Symbol	Name	Unit	Page
AD0IC	Interrupt control register	INTC	559
AD1IC	Interrupt control register	INTC	559
ADA0CR0	A/D0 conversion result register 0	ADC0	436
ADA0CR0H	A/D0 conversion result register 0H	ADC0	436
ADA0CR1	A/D0 conversion result register 1	ADC0	436
ADA0CR1H	A/D0 conversion result register 1H	ADC0	436
ADA0CR2	A/D0 conversion result register 2	ADC0	436
ADA0CR2H	A/D0 conversion result register 2H	ADC0	436
ADA0CR3	A/D0 conversion result register 3	ADC0	436
ADA0CR3H	A/D0 conversion result register 3H	ADC0	436
ADA0M0	A/D converter 0 mode register 0	ADC0	431
ADA0M1	A/D converter 0 mode register 1	ADC0	433
ADA0M2	A/D converter 0 mode register 2	ADC0	435
ADA0S	A/D converter 0 channel specification register	ADC0	434
ADA1CR0	A/D1 conversion result register 0	ADC1	436
ADA1CR0H	A/D1 conversion result register 0H	ADC1	436
ADA1CR1	A/D1 conversion result register 1	ADC1	436
ADA1CR1H	A/D1 conversion result register 1H	ADC1	436
ADA1CR2	A/D1 conversion result register 2	ADC1	436
ADA1CR2H	A/D1 conversion result register 2H	ADC1	436
ADA1CR3	A/D1 conversion result register 3	ADC1	436
ADA1CR3H	A/D1 conversion result register 3H	ADC1	436
ADA1M0	A/D converter 1 mode register 0	ADC1	431
ADA1M1	A/D converter 1 mode register 1	ADC1	433
ADA1M2	A/D converter 1 mode register 2	ADC1	435
ADA1S	A/D converter 1 channel specification register	ADC1	434
CB0CTL0	CSIB0 control register 0	CSIB0	503
CB0CTL1	CSIB0 control register 1	CSIB0	506
CB0CTL2	CSIB0 control register 2	CSIB0	507
CB0REIC	Interrupt control register	INTC	559
CB0RIC	Interrupt control register	INTC	559
CB0RX	CSIB0 receive data register	CSIB0	502
CB0RXL	CSIB0 receive data register L	CSIB0	502
CB0STR	CSIB0 status register	CSIB0	509
CB0TIC	Interrupt control register	INTC	559
CB0TX	CSIB0 transmit data register	CSIB0	502
CB0TXL	CSIB0 transmit data register L	CSIB0	502
CKC	Clock control register	CG	125
CLM	Clock monitor mode register	CG	129
HZA0CTL0	High-impedance output control register 00	Timer	372
HZA0CTL1	High-impedance output control register 01	Timer	372

(2/5)

Symbol	Name	Unit	Page
HZA1CTL0	High-impedance output control register 10	Timer	372
HZA1CTL1	High-impedance output control register 11	Timer	372
IMR0	Interrupt mask register 0	INTC	562
IMR0H	Interrupt mask register 0H	INTC	562
IMR0L	Interrupt mask register 0L	INTC	562
IMR1	Interrupt mask register 1	INTC	562
IMR1H	Interrupt mask register 1H	INTC	562
IMR1L	Interrupt mask register 1L	INTC	562
IMR2	Interrupt mask register 2	INTC	562
IMR2H	Interrupt mask register 2H	INTC	562
IMR2L	Interrupt mask register 2L	INTC	562
IMR3	Interrupt mask register 3	INTC	562
IMR3H	Interrupt mask register 3H	INTC	562
IMR3L	Interrupt mask register 3L	INTC	562
IMS	Internal memory size setting register	CPU	45
INTF0	External interrupt falling edge specification register 0	INTC	567
INTPNRC	External interrupt noise elimination control register	INTC	566
INTR0	External interrupt rising edge specification register 0	INTC	567
ISPR	In-service priority register	INTC	564
LVIIC	Interrupt control register	INTC	559
LVIM	Low-voltage detection register	POC/LVI	596
LVIS	Low-voltage detection level select register	POC/LVI	597
OSTS	Oscillation stabilization time select register	CG	128
P0	Port 0 register	Port	69
P1	Port 1 register	Port	74
P2	Port 2 register	Port	86
P3	Port 3 register	Port	91
P4	Port 4 register	Port	98
PCC	Processor clock control register	CG	125
PDLL	Port DL register L	Port	106
PFC1	Port 1 function control register	Port	76
PFC3	Port 3 function control register	Port	92
PFC4	Port 4 function control register	Port	99
PFCE1	Port 1 function control expansion register	Port	76
PIC0	Interrupt control register	INTC	559
PIC1	Interrupt control register	INTC	559
PIC2	Interrupt control register	INTC	559
PIC3	Interrupt control register	INTC	559
PIC4	Interrupt control register	INTC	559
PIC5	Interrupt control register	INTC	559
PIC6	Interrupt control register	INTC	559
PLLCTL	PLL control register	CG	124
PM0	Port 0 mode register	Port	69

(3/5)

Symbol	Name	Unit	Page
PM1	Port 1 mode register	Port	74
PM2	Port 2 mode register	Port	86
PM3	Port 3 mode register	Port	91
PM4	Port 4 mode register	Port	98
PMC0	Port 0 mode control register	Port	70
PMC1	Port 1 mode control register	Port	75
PMC2	Port 2 mode control register	Port	87
PMC3	Port 3 mode control register	Port	92
PMC4	Port 4 mode control register	Port	99
PMDLL	Port DL mode register L	Port	106
PRCMD	Command register	CPU	58
PRCMD2	Command register 2	CPU	58
PSC	Power save control register	CPU	126, 581
PSMR	Power save mode register	CPU	127, 582
PU0	Pull-up resistor option register 0	Port	70
PU1	Pull-up resistor option register 1	Port	78
PU2	Pull-up resistor option register 2	Port	88
PU3	Pull-up resistor option register 3	Port	93
PU4	Pull-up resistor option register 4	Port	100
PUDLL	Pull-up resistor option register DLL	Port	106
RAMS	Internal RAM data status register	POC/LVI	598
RESF	Reset source flag register	Reset	590
RESF2	Reset source flag register 2	Reset	591
SYS	System status register	CPU	59
SYS2	System status register 2	CPU	59
TM0CMP0	TMM0 compare register 0	Timer	352
TM0CTL0	TMM0 control register 0	Timer	353
TM0EQIC0	Interrupt control register	INTC	559
TP0CCIC0	Interrupt control register	INTC	559
TP0CCIC1	Interrupt control register	INTC	559
TP0CCR0	TMP0 capture/compare register 0	Timer	151
TP0CCR1	TMP0 capture/compare register 1	Timer	153
TP0CNT	TMP0 counter read buffer register	Timer	155
TP0CTL0	TMP0 control register 0	Timer	143
TP0CTL1	TMP0 control register 1	Timer	144
TP0IOC0	TMP0 I/O control register 0	Timer	146
TP0IOC1	TMP0 I/O control register 1	Timer	148
TP0IOC2	TMP0 I/O control register 2	Timer	149
TP0OPT0	TMP0 option register 0	Timer	150
TP0OVIC	Interrupt control register	INTC	559
TP1CCIC0	Interrupt control register	INTC	559
TP1CCIC1	Interrupt control register	INTC	559
TP1CCR0	TMP1 capture/compare register 0	Timer	151

(4/5)

Symbol	Name	Unit	Page
TP1CCR1	TMP1 capture/compare register 1	Timer	153
TP1CNT	TMP1 counter read buffer register	Timer	155
TP1CTL0	TMP1 control register 0	Timer	143
TP1CTL1	TMP1 control register 1	Timer	144
TP1OPT0	TMP1 option register 0	Timer	150
TP1OVIC	Interrupt control register	INTC	559
TP2CCIC0	Interrupt control register	INTC	559
TP2CCIC1	Interrupt control register	INTC	559
TP2CCR0	TMP2 capture/compare register 0	Timer	151
TP2CCR1	TMP2 capture/compare register 1	Timer	153
TP2CNT	TMP2 counter read buffer register	Timer	155
TP2CTL0	TMP2 control register 0	Timer	143
TP2CTL1	TMP2 control register 1	Timer	144
TP2IOC0	TMP2 I/O control register 0	Timer	146
TP2IOC1	TMP2 I/O control register 1	Timer	148
TP2IOC2	TMP2 I/O control register 2	Timer	149
TP2OPT0	TMP2 option register 0	Timer	150
TP2OVIC	Interrupt control register	INTC	559
TP3CCIC0	Interrupt control register	INTC	559
TP3CCIC1	Interrupt control register	INTC	559
TP3CCR0	TMP3 capture/compare register 0	Timer	151
TP3CCR1	TMP3 capture/compare register 1	Timer	153
TP3CNT	TMP3 counter read buffer register	Timer	155
TP3CTL0	TMP3 control register 0	Timer	143
TP3CTL1	TMP3 control register 1	Timer	144
TP3IOC0	TMP3 I/O control register 0	Timer	146
TP3OPT0	TMP3 option register 0	Timer	150
TP3OVIC	Interrupt control register	INTC	559
TQ0CCIC0	Interrupt control register	INTC	559
TQ0CCIC1	Interrupt control register	INTC	559
TQ0CCIC2	Interrupt control register	INTC	559
TQ0CCIC3	Interrupt control register	INTC	559
TQ0CCR0	TMQ0 capture/compare register 0	Timer	251
TQ0CCR1	TMQ0 capture/compare register 1	Timer	253
TQ0CCR2	TMQ0 capture/compare register 2	Timer	255
TQ0CCR3	TMQ0 capture/compare register 3	Timer	257
TQ0CNT	TMQ0 counter read buffer register	Timer	259
TQ0CTL0	TMQ0 control register 0	Timer	244
TQ0CTL1	TMQ0 control register 1	Timer	244
TQ0IOC0	TMQ0 I/O control register 0	Timer	246
TQ0IOC1	TMQ0 I/O control register 1	Timer	248
TQ0IOC2	TMQ0 I/O control register 2	Timer	249
TQ0OPT0	TMQ0 option register 0	Timer	250

(5/5)

Symbol	Name	Unit	Page
TQ0OVIC	Interrupt control register	INTC	559
TQ1CCIC0	Interrupt control register	INTC	559
TQ1CCIC1	Interrupt control register	INTC	559
TQ1CCIC2	Interrupt control register	INTC	559
TQ1CCIC3	Interrupt control register	INTC	559
TQ1CCR0	TMQ1 capture/compare register 0	Timer	251
TQ1CCR1	TMQ1 capture/compare register 1	Timer	253
TQ1CCR2	TMQ1 capture/compare register 2	Timer	255
TQ1CCR3	TMQ1 capture/compare register 3	Timer	257
TQ1CNT	TMQ1 counter read buffer register	Timer	259
TQ1CTL0	TMQ1 control register 0	Timer	244
TQ1CTL1	TMQ1 control register 1	Timer	244
TQ1DTC	TMQ1 dead-time compare register	Timer	363
TQ1IOC0	TMQ1 I/O control register 0	Timer	246
TQ1IOC3	TMQ1 I/O control register 3	Timer	369
TQ1OPT0	TMQ1 option register 0	Timer	250, 364
TQ1OPT1	TMQ1 option register 1	Timer	365
TQ1OPT2	TMQ1 option register 2	Timer	366
TQ1OPT3	TMQ1 option register 3	Timer	368
TQ1OVIC	Interrupt control register	INTC	559
UA0CTL0	UARTA0 control register 0	UARTA0	477
UA0CTL1	UARTA0 control register 1	UARTA0	478
UA0CTL2	UARTA0 control register 2	UARTA0	478
UA0OPT0	UARTA0 option control register 0	UARTA0	479
UA0REIC	Interrupt control register	INTC	559
UA0RIC	Interrupt control register	INTC	559
UA0RX	UARTA0 receive data register	UARTA0	481
UA0STR	UARTA0 status register	UARTA0	479
UA0TIC	Interrupt control register	INTC	559
UA0TX	UARTA0 transmit data register	UARTA0	481
UA1CTL0	UARTA1 control register 0	UARTA1	477
UA1CTL1	UARTA1 control register 1	UARTA1	478
UA1CTL2	UARTA1 control register 2	UARTA1	478
UA1OPT0	UARTA1 option control register 0	UARTA1	479
UA1REIC	Interrupt control register	INTC	559
UA1RIC	Interrupt control register	INTC	559
UA1RX	UARTA1 receive data register	UARTA1	481
UA1STR	UARTA1 status register	UARTA1	479
UA1TIC	Interrupt control register	INTC	559
UA1TX	UARTA1 transmit data register	UARTA1	481
VSWC	System wait control register	BCU	61
WDTE	Watchdog timer enable register	WDT	425
WDTM	Watchdog timer mode register	WDT	424

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 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 (r3)
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

(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
×	Set or cleared in accordance with the results.
R	Previously saved values are restored.

(6) Condition codes

Condition Name (cond)	Condition Code (cccc)	Condition Formula	Explanation
V	0 0 0 0	OV = 1	Overflow
NV	1 0 0 0	OV = 0	No overflow
C/L	0 0 0 1	CY = 1	Carry Lower (less than)
NC/NL	1 0 0 1	CY = 0	No carry Not lower (greater than or equal)
Z	0 0 1 0	Z = 1	Zero
NZ	1 0 1 0	Z = 0	Not zero
NH	0 0 1 1	(CY or Z) = 1	Not higher (less than or equal)
H	1 0 1 1	(CY or Z) = 0	Higher (greater than)
S/N	0 1 0 0	S = 1	Negative
NS/P	1 1 0 0	S = 0	Positive
T	0 1 0 1	–	Always (unconditional)
SA	1 1 0 1	SAT = 1	Saturated
LT	0 1 1 0	(S xor OV) = 1	Less than signed
GE	1 1 1 0	(S xor OV) = 0	Greater than or equal signed
LE	0 1 1 1	((S xor OV) or Z) = 1	Less than or equal signed
GT	1 1 1 1	((S xor OV) or Z) = 0	Greater than signed

C.2 Instruction Set (in Alphabetical Order)

(1/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010010iiiii	GR[reg2]←GR[reg2]+sign-extend(imm5)		1	1	1	×	×	×	×	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1	×	×	×	×	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]		1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND zero-extend(imm16)		1	1	1		0	×	×	
Bcond	disp9	ddddd1011dddcccc **Note 1**	if conditions are satisfied then PC←PC+sign-extend(disp9)	When conditions are satisfied	2 Note 2	2 Note 2	2 Note 2					
				When conditions are not satisfied	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 wwwww01101000010	GR[reg3]←GR[reg2] (23 : 16) ll GR[reg2] (31 : 24) ll GR[reg2] (7 : 0) ll GR[reg2] (15 : 8)		1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 wwwww01101000000	GR[reg3]←GR[reg2] (7 : 0) ll GR[reg2] (15 : 8) ll GR [reg2] (23 : 16) ll GR[reg2] (31 : 24)		1	1	1	×	0	×	×	
CALLT	imm6	0000001000iiiiii	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+zero-extend(imm6 logically shift left by 1) PC←CTBP+zero-extend(Load-memory(adr,Halfword))		4	4	4					
CLR1	bit#3,disp16[reg1]	10bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)		3 Note 3	3 Note 3	3 Note 3				×	
CMOV	cccc,imm5,reg2,reg3	rrrrr111111iiiii wwwww011000cccc0	if conditions are satisfied then GR[reg3]←sign-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	rrrrr010011iiiii	result←GR[reg2]–sign-extend(imm5)		1	1	1	×	×	×	×	
CTRET		0000011111100000 0000000101000100	PC←CTPC PSW←CTPSW		3	3	3	R	R	R	R	R
DBRET		0000011111100000 0000000101000110	PC←DBPC PSW←DBPSW		3	3	3	R	R	R	R	R

(2/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
DBTRAP		1111100001000000	DBPC←PC+2 (restored PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	3	3	3					
DI		0000011111100000 0000000101100000	PSW.ID←1	1	1	1					
DISPOSE	imm5,list12	0000001100 1iiiiiL LLLLLLLLLLL00000	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	imm5,list12,[reg1]	0000001100 1iiiiiL LLLLLLLLLLRRRRR **Note 5**	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded PC←GR[reg1]	n+3 Note 4	n+3 Note 4	n+3 Note 4					
DIV	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000000	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVH	reg1,reg2	rrrrr000010RRRRR	GR[reg2]←GR[reg2]÷GR[reg1]Note 6	35	35	35		×	×	×	
	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000000	GR[reg2]←GR[reg2]÷GR[reg1]Note 6 GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVHU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000010	GR[reg2]←GR[reg2]÷GR[reg1]Note 6 GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
DIVU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000010	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
EI		1000011111100000 0000000101100000	PSW.ID←0	1	1	1					
HALT		0000011111100000 0000000100100000	Stop	1	1	1					
HSW	reg2,reg3	rrrrr11111100000 wwwww01101000100	GR[reg3]←GR[reg2](15 : 0) ll GR[reg2] (31 : 16)	1	1	1	×	0	×	×	
JARL	disp22,reg2	rrrrr11110ddddd ddddddddddddddd0 **Note 7**	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	3	3	3					
JR	disp22	0000011110ddddd ddddddddddddddd0 **Note 7**	PC←PC+sign-extend(disp22)	2	2	2					
LD.B	disp16[reg1],reg2	rrrrr111000RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 11					
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRRR ddddddddddddd1 **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	r r r r r 0 1 0 0 0 0 i i i i i	GR[reg2]←sign-extend(imm5)		1	1	1					
	imm32,reg1	00000110001RRRRR i i i i i i i i i i i i i i i i l l l l l l l l l l l l l l l l	GR[reg1]←imm32		2	2	2					
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1					
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1]+(imm16 ll 0^{16})		1	1	1					
MUL Note 22	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000 **Note 14**	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1]		1	4	5					
	imm9,reg2,reg3	r r r r r 1 1 1 1 1 1 i i i i i w w w w w 0 1 0 0 1 l l l l 0 0 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xsign-extend(imm9)		1	4	5					
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2] Note 6 xGR[reg1] Note 6		1	1	2					
	imm5,reg2	r r r r r 0 1 0 1 1 1 i i i i i	GR[reg2]←GR[reg2] Note 6 xsign-extend(imm5)		1	1	2					
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1] Note 6 ximm16		1	1	2					
MULU Note 22	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010 **Note 14**	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1]		1	4	5					
	imm9,reg2,reg3	r r r r r 1 1 1 1 1 1 i i i i i w w w w w 0 1 0 0 1 l l l l 1 0 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xzero-extend(imm9)		1	4	5					
NOP		0000000000000000	Pass at least one clock cycle doing nothing.		1	1	1					
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])		1	1	1		0	×	×	
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR 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) sp←sp+4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend (imm5) ep←sp/imm	n+2 Note 4 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	3	3	3	R	R	R	R	R
SAR	reg1,reg2	rrrrr111111RRRRR 0000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010101iiiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend (imm5)	1	1	1	×	0	×	×	
SASF	cccc,reg2	rrrrr1111110cccc 0000001000000000	if conditions are satisfied then GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000001H else GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000000H	1	1	1					
SATADD	reg1,reg2	rrrrr000110RRRRR	GR[reg2]←saturated(GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×
	imm5,reg2	rrrrr010001iiiii	GR[reg2]←saturated(GR[reg2]+sign-extend(imm5)	1	1	1	×	×	×	×	×
SATSUB	reg1,reg2	rrrrr000101RRRRR	GR[reg2]←saturated(GR[reg2]–GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16,reg1,reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiii	GR[reg2]←saturated(GR[reg1]–sign-extend(imm16)	1	1	1	×	×	×	×	×
SATSUBR	reg1,reg2	rrrrr000100RRRRR	GR[reg2]←saturated(GR[reg1]–GR[reg2])	1	1	1	×	×	×	×	×
SETF	cccc,reg2	rrrrr1111110cccc 0000000000000000	If conditions are satisfied then GR[reg2]←00000001H else GR[reg2]←00000000H	1	1	1					

(5/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SET1	bit#3,disp16[reg1]	00bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,1)	3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 Note 3	3 Note 3	3 Note 3				×	
SHL	reg1,reg2	rrrrr111111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010110iiiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×	
SHR	reg1,reg2	rrrrr111111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010100iiiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SLD.B	disp7[ep],reg2	rrrrr0110ddddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.BU	disp4[ep],reg2	rrrrr0000110dddd **Note 18**	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.H	disp8[ep],reg2	rrrrr1000ddddddd **Note 19**	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,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	rrrrr1010dddddd0 **Note 21**	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	Note 9					
SST.B	reg2,disp7[ep]	rrrrr0111ddddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1					
SST.H	reg2,disp8[ep]	rrrrr1001ddddddd **Note 19**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],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 iiiii 0000000100000000	EIPC ←PC+4 (Restored PC) EIPSW ←PSW ECR.EICC ←Interrupt code PSW.EP ←1 PSW.ID ←1 PC ←00000040H (when vector is 00H to 0FH) 00000050H (when vector is 10H to 1FH)	3	3	3					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR 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. 3 if there is an instruction that rewrites the contents of the PSW immediately before.

3. If there is no wait state (3 + the number of read access wait states).

4. n is the total number of list12 load registers. (According to the number of wait states. Also, if there are no wait states, n is the 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. ddddddddddddddddddddddd: The higher 21 bits of disp22.

8. ddddddddddddddd: The higher 15 bits of disp16.

9. According to the number of wait states (1 if there are no wait states).

10. b: bit 0 of disp16.

11. According to the number of wait states (2 if there are no wait states).

Notes 12. In this instruction, for convenience of mnemonic description, the source register is made reg2, but the reg1 field is used in the opcode. Therefore, the meaning of register specification in the mnemonic description and in the opcode differs from other instructions.

rrrrr = regID specification

RRRRR = reg2 specification

13. iiiii: Lower 5 bits of imm9.

IIII: Higher 4 bits of imm9.

14. Do not specify the same register for general-purpose registers reg1 and reg3.

15. sp/imm: specified by bits 19 and 20 of the sub-opcode.

16. ff = 00: Load sp in ep.

01: Load sign expanded 16-bit immediate data (bits 47 to 32) in ep.

10: Load 16-bit logically left shifted 16-bit immediate data (bits 47 to 32) in ep.

11: Load 32-bit immediate data (bits 63 to 32) in ep.

17. If imm = imm32, n + 3 clocks.

18. rrrrr: Other than 00000.

19. ddddddd: Higher 7 bits of disp8.

20. dddd: Higher 4 bits of disp5.

21. dddddd: Higher 6 bits of disp8.

22. Do not make a register combination that satisfies all the following conditions when executing the “MUL reg1, reg2, reg3” and “MULU reg1, reg2, reg3” instructions. If an instruction that satisfies these conditions is executed, the operation is not guaranteed.

- reg1 = reg3
- reg1 ≠ reg2
- reg1 ≠ r0
- reg3 ≠ r0

<R>

APPENDIX D REVISION HISTORY

D.1 Major Revisions in This Edition

(1/3)

Page	Description
Throughout	• Change of under development state of all products → Development completed • Addition of PG-FP5
p. 23	Addition of description to **2.1 (1) Port pins**
p. 37	Modification of description in **3.2.2 (6) Exception/debug trap status saving registers (DBPC, DBPSW)**
p. 56	Addition of description to **3.4.7 Special registers**
p. 58	Addition of description to **3.4.7 (2) (a) Command register (PRCMD)**
p. 58	Addition of description to **3.4.7 (2) (b) Command register 2 (PRCMD2)**
p. 117	Modification of description in **Table 4-12 Noise Eliminator**
p. 118	Addition of **Figure 4-22 Example of Noise Elimination Timing**
p. 119	Addition of **4.6 (1) External interrupt noise elimination control register (INTPNRC)**
p. 125	Addition of description to **5.3 (2) Clock control register (CKC)**
p. 125	Addition of description to **5.3 (3) Processor clock control register (PCC)**
p. 126	Addition of description to **5.3 (4) Power save control register (PSC)**
p. 129	Addition of description to **5.3 (7) Clock monitor mode register (CLM)**
p. 137	Addition of description to **Table 6-1 TMPn Overview**
p. 145	Modification of description in **6.4 (2) TMPn control register 1 (TPnCTL1)**
p. 147	Modification of description in **6.4 (3) TMPm I/O control register 0 (TPmIOC0)**
p. 159	Modification of description in **6.6 (1) (a) Counter start operation**
pp. 166 to 168	Modification of description in **Figure 6-11 Register Setting for Interval Timer Mode Operation**
p. 169	Addition of description to **Figure 6-12 Software Processing Flow in Interval Timer Mode**
p. 176	Addition of **6.6.1 (3) Operation by external event count input (TIPk0)**
p. 177	Addition of description to **6.6.2 External event count mode (TPkMD2 to TPkMD0 bits = 001)**
p. 180	Addition of description to **Figure 6-18 Register Setting for Operation in External Event Count Mode**
p. 182	Addition of description to **6.6.2 (2) Operation timing in external event count mode**
p. 186	Modification of description in **Figure 6-23 Configuration in External Trigger Pulse Output Mode**
p. 187	Modification of **Figure 6-24 Basic Timing in External Trigger Pulse Output Mode**
p. 187	Addition of description to **6.6.3 External trigger pulse output mode (TPmMD2 to TPmMD0 bits = 010)**
pp. 188, 189	Modification of description in **Figure 6-25 Setting of Registers in External Trigger Pulse Output Mode**
p. 195	Modification of figure in **6.6.3 (2) (b) 0%/100% output of PWM waveform**
p. 199	Modification of description in **Figure 6-27 Configuration in One-Shot Pulse Output Mode**
pp. 201, 202	Modification of description in **Figure 6-29 Setting of Registers in One-Shot Pulse Output Mode**
p. 203	Modification of description in **Figure 6-30 Software Processing Flow in One-Shot Pulse Output Mode**
p. 204	Modification of figure in **6.6.4 (2) (a) Note on rewriting TPmCCRa register**
p. 208	Addition of description to **Figure 6-33 Register Setting in PWM Output Mode**

(2/3)

Page	Description
p. 213	Modification of description in **6.6.5 (2) (b) 0%/100% output of PWM waveform**
p. 232	Modification of description in **Figure 6-41 Configuration in Pulse Width Measurement Mode**
p. 233	Modification of description in **6.6.7 Pulse width measurement mode (TPkMD2 to TPkMD0 bits = 110)**
pp. 234, 235	Modification of description in **Figure 6-43 Register Setting in Pulse Width Measurement Mode**
p. 236	Deletion of description from **Figure 6-44 Software Processing Flow in Pulse Width Measurement Mode**
p. 238	Addition of description to **Table 7-1 TMQn Overview**
p. 246	Modification of description in **7.4 (3) TMQn I/O control register 0 (TQnIOC0)**
p. 262	Modification of description in **7.6 (1) (a) Counter start operation**
pp. 270 to 272	Modification of description in **Figure 7-10 Register Setting for Interval Timer Mode Operation**
p. 273	Addition of description to **Figure 7-11 Software Processing Flow in Interval Timer Mode**
p. 278	Modification of description in **7.6.1 (2) (d) Operation of TQnCCR1 to TQnCCR3 registers**
p. 280	Addition of **7.6.1 (3) Operation by external event count input (EVTQ0)**
p. 281	Addition of description to **7.6.2 External event count mode (TQ0MD2 to TQ0MD0 bits = 001)**
p. 284	Addition of description to **Figure 7-17 Register Setting for Operation in External Event Count Mode**
p. 286	Addition of description to **7.6.2 (2) Operation timing in external event count mode**
p. 292	Modification of figure description in **Figure 7-23 Basic Timing in External Trigger Pulse Output Mode**
p. 293	Addition of description to **7.6.3 External trigger pulse output mode (TQ0MD2 to TQ0MD0 bits = 010)**
p. 301	Modification of description in **7.6.3 (2) (b) 0%/100% output of PWM waveform**
p. 309	Modification of description in **Figure 7-28 Register Setting in One-Shot Pulse Output Mode**
pp. 310, 311	Modification of description in **Figure 7-29 Software Processing Flow in One-Shot Pulse Output Mode**
p. 323	Modification of description in **7.6.5 (2) (b) 0%/100% output of PWM waveform**
p. 346	Modification of description in **7.6.7 Pulse width measurement mode (TQ0MD2 to TQ0MD0 bits = 110)**
p. 363	Addition of description to **9.2 (1) TMQ1 dead-time compare register (TQ1DTC)**
p. 365	Addition of description to **9.3 (2) TMQ1 option register 1 (TQ1OPT1)**
p. 366	Addition of description to **9.3 (3) TMQ1 option register 2 (TQ1OPT2)**
p. 370	Addition of description to **Figure 9-3 Output Control of TOQ1Tm and TOQ1Bm Pins (Without Dead Time)**
p. 378	Modification of **Figure 9-5 Outline of 6-Phase PWM Output Mode**
p. 383	Modification of **Figure 9-9 0% PWM Output Waveform (with Dead Time)**
p. 384	Modification of **Figure 9-10 100% PWM Output Waveform (with Dead Time)**
p. 385	Modification of **Figure 9-11 PWM Output Waveform from 0% to 100% and from 100% to 0% (with Dead Time)**
p. 389	Addition of description to **9.4.3 Interrupt culling function**
p. 438	Modification of description in **11.4.1 Basic operation**
p. 475	Modification of **Figure 12-1 Block Diagram of UARTAn**
p. 476	Modification of description in **12.2 (5) UARTAn status register (UAnSTR)**
p. 477	Modification of description in **12.3 (1) UARTAn control register 0 (UAnCTL0)**
p. 487	Modification of **Figure 12-5 Continuous Transmission Operation Timing**
p. 523	Modification of figure description in **13.4.7 (2) Operation timing**
p. 559	Addition of description to **14.3.4 Interrupt control registers (xxICn)**
p. 572	Addition of description to **14.6.1 (2) Restore**

(3/3)

Page	Description
p. 581	Addition of description to **15.2 (1) Power save control register (PSC)**
p. 587	Addition of description to **15.5.1 Setting and operation status**
p. 590	Addition of description to **16.2 (1) Reset source flag register (RESF)**
p. 596	Addition of description to **16.3.3 (3) (a) Low-voltage detection register (LVIM)**
p. 598	Addition of description to **16.3.3 (3) (c) Internal RAM data status register (RAMS)**
p. 610	Modification of description in **Table 18-2 Basic Functions**
p. 610	Modification of description in **Table 18-3 Security Functions**
p. 611	Addition of **Table 18-4 Security Setting**
p. 612	Addition of **18.3 (1) Security setting by PG-FP4 and PG-FP5 (Security flag settings)**
p. 619	Modification of **Figure 18-7 Procedure for Manipulating Flash Memory**
p. 621	Modification of description in **Table 18-7 Flash Memory Control Commands**
p. 629	Modification of description in **Figure 18-17 Standard Self Programming Flow**
p. 630	Modification of description in **Table 18-10 Flash Function List**
p. 634	Addition of **Clock Oscillator Characteristics (i) Murata Mfg. Co., Ltd.: Ceramic resonator (T_A = –40 to +85°C)** in **CHAPTER 19 ELECTRICAL SPECIFICATIONS**
p. 636	Addition of description to **DC Characteristics** in **CHAPTER 19 ELECTRICAL SPECIFICATIONS**
p. 642	Modification of description of **CSIB Timing** in **CHAPTER 19 ELECTRICAL SPECIFICATIONS**
p. 645	Modification of description of **Characteristics of A/D Converters 0, 1** in **CHAPTER 19 ELECTRICAL SPECIFICATIONS**
p. 647	Addition of description to **Low-Voltage Detector (LVI)** in **CHAPTER 19 ELECTRICAL SPECIFICATIONS**
p. 650	Modification of description of **Flash Memory Programming Characteristics (3) Programming characteristics** in **CHAPTER 19 ELECTRICAL SPECIFICATIONS**
p. 652	Addition of **CHAPTER 21 RECOMMENDED SOLDERING CONDITIONS**
p. 669	Addition of **APPENDIX D REVISION HISTORY**

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