NEC

User's Manual

V850E/MA3

32-Bit Single-Chip Microcontrollers

Hardware

μPD703131A
μPD703131AY
μPD703132A
μPD703132AY
μPD703133A
μPD703133AY
μPD703134A
μPD703134AY
μPD703136A
μPD703136AY
μPD70F3134A
μPD70F3134AY

Document No. U16397EJ3V0UD00 (3rd edition)
Date Published January 2006 N CP(K)

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.

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

- **The information in this document is current as of September, 2005. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific".
  The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.
  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.
  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).
  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

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

(Note)
(1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.
(2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

M8E 02. 11-1

[MEMO]

PREFACE

Readers

This manual is intended for users who wish to understand the functions of the V850E/MA3 to design application systems using the V850E/MA3.

Purpose

This manual is intended to give users an understanding of the hardware functions of the V850E/MA3.

Organization

The **V850E/MA3 User's Manual** is divided into two parts: Hardware (this manual) and Architecture **(V850E1 Architecture User's Manual)**. The organization of each manual is as follows:

Hardware	Architecture
• Pin functions	• Data type
• CPU function	• Register set
• Internal 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 find the details of a register where the name is known
  →Refer to **APPENDIX A REGISTER INDEX**.

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

- To know the electrical specifications of the V850E/MA3
  →Refer to **CHAPTER 26 ELECTRICAL SPECIFICATIONS**.

- To understand the overall functions of the V850E/MA3
  →Read this manual according to the **CONTENTS**.

- 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.

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 V850E/MA3

Document Name	Document No.
V850E1 Architecture User's Manual	U14559E
V850E/MA3 Hardware User's Manual	This manual
V850E/MA1, V850E/MA2, V850E/MA3, V850E/ME2 PCI Host Bridge Macro Application Note	U17121E
Inverter Control by V850 Series 120° Excitation Method Control by Zero-Cross Detection Application Note	U17209E
Inverter Control by V850 Series Vector Control by Hole Sensor Application Note	U17338E

Documents related to development tools (user's manuals)

Document Name		Document No.
IE-V850E1-CD-NW (PCMCIA card type on-chip debug emulator)		U16647E
CA850 (Ver. 3.00) (C compiler package)	Operation	U17293E
	C Language	U17291E
	Assembly Language	U17292E
	Link Directive	U17294E
PM+ (Ver. 6.00) (Project manager)		U17178E
ID850 (Ver. 3.00) (Integrated debugger)	Operation	U17358E
ID850NW (Ver. 3.00, 3.10) (Integrated debugger)	Operation	U17369E
ID850NWC (Ver. 2.51) (Integrated debugger)	Operation	U16525E
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. 3.20) (Real-time OS)	Basics	U13773E
	Installation	U17421E
	Technical	U13772E
	Task Debugger	U17422E
AZ850 (Ver. 3.30) (System performance analyzer)		U17423E
PG-FP4 Flash Memory Programmer		U15260E

CONTENTS

CHAPTER 1 INTRODUCTION

The V850E/MA3 is a product of the NEC Electronics V850 Series of single-chip microcontrollers. This chapter gives a simple outline of the V850E/MA3.

1.1 Overview

The V850E/MA3 is a 32-bit single-chip microcontroller that integrates the V850E1 CPU, which is a 32-bit RISC-type CPU core for ASIC, newly developed as the CPU core central to system LSI for the current age of system-on-chip. This device incorporates ROM, RAM, and various peripheral functions such as memory controllers, a DMA controller, timers/counters, serial interfaces, an A/D converter, a D/A converter, ROM correction, and on-chip debugging for realizing high-capacity data processing and sophisticated real-time control.

(1) V850E1 CPU

The V850E1 CPU is a CPU core that enhances the external bus interface performance of the V850 CPU, which is the CPU core integrated in the V850 Series, and has added instructions supporting high-level languages, such as C-language switch statement processing, table lookup branching, stack frame creation/deletion, and data conversion. This enhances the performance of both data processing and control.

It is possible to use the software resources of the V850 CPU integrated system since the instruction codes of the V850E1 are upwardly compatible at the object code level with those of the V850 CPU.

(2) External memory interface function

The V850E/MA3 features various on-chip external memory interfaces including separately configured address (26-bit) and multiplex configured address/data (16-bit) buses, and SDRAM and ROM interfaces, as well as on-chip memory controllers that can be directly linked to page ROM, etc., thereby raising system performance and reducing the number of parts needed for application systems.

Also, through the DMA controller, CPU internal calculations and data transfers can be performed simultaneously with transfers to and from the external memory, so it is possible to process large volumes of image data or voice data, etc., and through high-speed execution of instructions using internal ROM and RAM, motor control, communications control and other real-time control tasks can be realized simultaneously.

(3) On-chip flash memory (flash memory versions only (see Table 1-1))

The on-chip flash memory versions have on-chip flash memory, which is capable of high-speed access, and since it is possible to rewrite a program with the V850E/MA3 mounted as is in the application system, system development time can be reduced and system maintainability after shipment can be markedly improved.

(4) A full range of middleware and development environment products

The V850E/MA3 can execute middleware such as JPEG, JBIG, and MH/MR/MMR at high speed. Also, middleware that enables speech recognition, voice synthesis, and other such processing is available, and by including these middleware programs, a multimedia system can be easily realized.

A development environment system that includes an optimized C compiler, debugger, in-circuit emulator, simulator, system performance analyzer, and other elements is also available.

The following shows the V850E/MA3 product list.

<R> **Table 1-1. V850E/MA3 Product List**

Function / Part Number	ROM		RAM Size	I^2C Bus	Maskable Interrupts		Non-Maskable Interrupts
	Type	Size			External	Internal	
μPD703131A	Mask ROM	256 KB	16 KB	None	25	49	2
μPD703131AY				On-chip		50	
μPD703132A			32 KB	None		49	
μPD703132AY				On-chip		50	
μPD703133A		512 KB	16 KB	None		49	
μPD703133AY				On-chip		50	
μPD703134A			32 KB	None		49	
μPD703134AY				On-chip		50	
μPD703136A		256 KB	8 KB	None		49	
μPD703136AY				On-chip		50	
μPD70F3134A	Flash memory	512 KB	32 KB	None		49	
μPD70F3134AY				On-chip		50	

Remark The part numbers of the V850E/MA3 are shown as follows in this manual.

- Mask ROM version
  μPD703131A, 703131AY, 703132A, 703132AY, 703133A, 703133AY, 703134A, 703134AY, 703136A, 703136AY
- Flash memory version
  μPD70F3134A, 70F3134AY
- I^2C bus version (Y version)
  μPD703131AY, 703132AY, 703133AY, 703134AY, 703136AY, 70F3134AY

1.2 Features

○ Minimum instruction execution time:
12.5 ns (at internal 80 MHz operation)

○ General-purpose registers: 32 bits × 32

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

○ Memory space: 256 MB of linear address space (common program/data use)
Chip select output function: 8 spaces
Memory block division function: 2, 64 MB/block

<R>
- Internal memory: RAM: 8/16/32 KB (See **Table 1-1**)
  Mask ROM: 256/512 KB (See **Table 1-1**)
  Flash memory: 512 KB (See **Table 1-1**)
- External bus interface: Separate bus/multiplexed bus output selectable
  8-/16-bit data bus sizing function
  External bus division function: Divided by 1, 2, 3, 4 (50 MHz max.)
  Wait function
    - Programmable wait function
    - External wait function
  Idle state function
  Bus hold function
  Address setup wait function
  Endian control function
- Memory access controller: DRAM controller (compatible with SDRAM)
  Page ROM controller

○ Interrupts and exceptions:
Non-maskable interrupts: 2 sources (external: 1 source, internal: 1 source)
Maskable interrupts: 74/75 sources (external: 25 sources, internal: 49/50 sources (see **Table 1-1**)
Software exceptions: 32 sources
Exception trap: 2 sources

○ DMA controller: 4 channels
Transfer unit: 8 bits/16 bits
Maximum transfer count: 65,536 (2^{16})
Transfer type: Flyby (1-cycle)/2-cycle
Transfer mode: Single/Single step/Block
Transfer target: Memory ↔ memory, memory ↔ I/O
Transfer request: External request/On-chip peripheral I/O/Software
DMA transfer terminate (terminal count) output signal
Next address setting function

- I/O lines: Total: 112

- Timer/counter function: Up/down counter/general-purpose timer (TMENC) for 16-bit 2-phase encoder input: 1 channel
  16-bit interval timer D (TMD): 4 channels
  16-bit timer/event counter Q (TMQ): 1 channel
  16-bit timer/event counter P (TMP): 3 channels
  Motor control function (Timers used: TMQ: 1 channel (TMQ0), TMP: 1 channel (TMP2)
    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)
  Clocked serial interface B (CSIB)
  I^2C bus interface (I^2C) (I^2C bus versions (Y products) only)
    CSIB/UARTA: 3 channels
    UARTA/I^2C: 1 channel

- A/D converter: 10-bit resolution A/D converter: 8 channels

- D/A converter: 8-bit resolution: 2 channels

- ROM correction: Four places can be corrected.

- On-chip debug function

- Clock generator: $\times$1.25, 2.5, 5, 10 function via a PLL clock synthesizer (input clock: 4 to 8 MHz)
  External clock input function (input clock: 5 to 25 MHz)

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

- Package: 144-pin plastic LQFP (fine pitch) (20 $\times$ 20)
  161-pin plastic FBGA (13 $\times$ 13)

- CMOS technology: Fully static circuits

1.3 Applications

Printers, DVD players, inverters, servos, NC machine tools, PPC, robot control, digital home electronics, etc.

<R> 1.4 Ordering Information

Part Number	Package	Internal ROM
μPD703131AGJ-xxx-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (256 KB)
μPD703131AGJ-xxx-UEN-A	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (256 KB)
μPD703131AYGJ-xxx-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (256 KB)
μPD703131AYGJ-xxx-UEN-A	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (256 KB)
μPD703132AGJ-xxx-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (256 KB)
μPD703132AGJ-xxx-UEN-A	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (256 KB)
μPD703132AYGJ-xxx-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (256 KB)
μPD703132AYGJ-xxx-UEN-A	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (256 KB)
μPD703133AGJ-xxx-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (512 KB)
μPD703133AGJ-xxx-UEN-A	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (512 KB)
μPD703133AYGJ-xxx-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (512 KB)
μPD703133AYGJ-xxx-UEN-A	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (512 KB)
μPD703134AGJ-xxx-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (512 KB)
μPD703134AGJ-xxx-UEN-A	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (512 KB)
μPD703134AYGJ-xxx-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (512 KB)
μPD703134AYGJ-xxx-UEN-A	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (512 KB)
μPD703136AGJ-xxx-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (256 KB)
μPD703136AGJ-xxx-UEN-A	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (256 KB)
μPD703136AYGJ-xxx-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (256 KB)
μPD703136AYGJ-xxx-UEN-A	144-pin plastic LQFP (fine pitch) (20 × 20)	Mask ROM (256 KB)
μPD703131AF1-xxx-EN4[Note]	161-pin plastic FBGA (13 × 13)	Mask ROM (256 KB)
μPD703131AYF1-xxx-EN4[Note]	161-pin plastic FBGA (13 × 13)	Mask ROM (256 KB)
μPD703132AF1-xxx-EN4[Note]	161-pin plastic FBGA (13 × 13)	Mask ROM (256 KB)
μPD703132AYF1-xxx-EN4[Note]	161-pin plastic FBGA (13 × 13)	Mask ROM (256 KB)
μPD703133AF1-xxx-EN4[Note]	161-pin plastic FBGA (13 × 13)	Mask ROM (512 KB)
μPD703133AYF1-xxx-EN4[Note]	161-pin plastic FBGA (13 × 13)	Mask ROM (512 KB)
μPD703134AF1-xxx-EN4[Note]	161-pin plastic FBGA (13 × 13)	Mask ROM (512 KB)
μPD703134AYF1-xxx-EN4[Note]	161-pin plastic FBGA (13 × 13)	Mask ROM (512 KB)
μPD703136AF1-xxx-EN4[Note]	161-pin plastic FBGA (13 × 13)	Mask ROM (256 KB)
μPD703136AYF1-xxx-EN4[Note]	161-pin plastic FBGA (13 × 13)	Mask ROM (256 KB)
μPD70F3134AGJ-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Flash memory (512 KB)
μPD70F3134AGJ-UEN-A	144-pin plastic LQFP (fine pitch) (20 × 20)	Flash memory (512 KB)
μPD70F3134AYGJ-UEN	144-pin plastic LQFP (fine pitch) (20 × 20)	Flash memory (512 KB)
μPD70F3134AYGJ-UEN-A	144-pin plastic LQFP (fine pitch) (20 × 20)	Flash memory (512 KB)
μPD70F3134AF1-EN4[Note]	161-pin plastic FBGA (13 × 13)	Flash memory (512 KB)
μPD70F3134AF1-EN4-A[Note]	161-pin plastic FBGA (13 × 13)	Flash memory (512 KB)
μPD70F3134AYF1-EN4[Note]	161-pin plastic FBGA (13 × 13)	Flash memory (512 KB)
μPD70F3134AYF1-EN4-A[Note]	161-pin plastic FBGA (13 × 13)	Flash memory (512 KB)

Note Under development

Remarks 1. xxx indicates ROM code suffix.

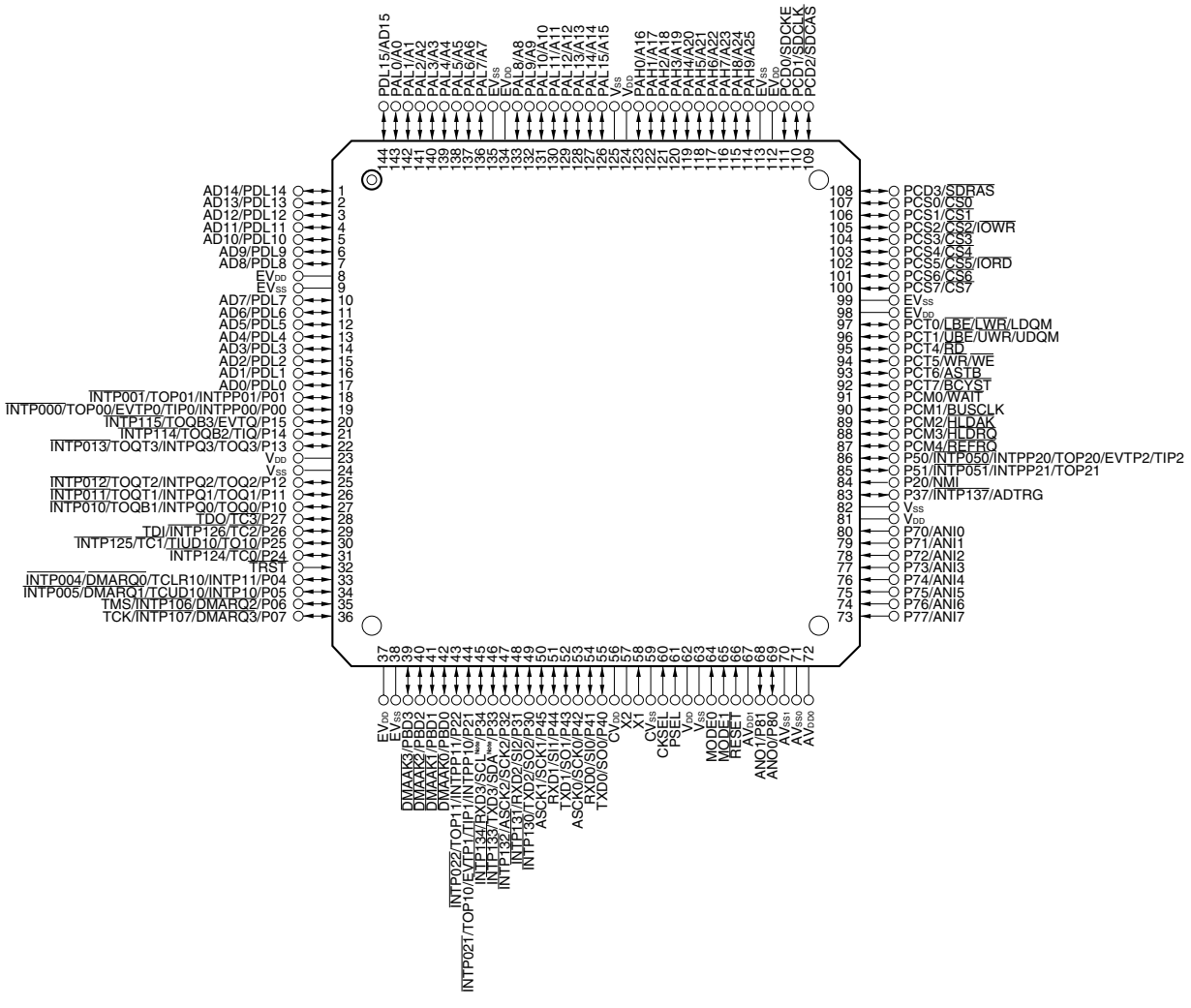
2. Products with -A at the end of the part number are lead-free products.

<R> 1.5 Pin Configuration

- 144-pin plastic LQFP (fine pitch) (20 × 20)

μPD703131AGJ-xxx-UEN	μPD703133AGJ-xxx-UEN	μPD703136AGJ-xxx-UEN
μPD703131AGJ-xxx-UEN-A	μPD703133AGJ-xxx-UEN-A	μPD703136AGJ-xxx-UEN-A
μPD703131AYGJ-xxx-UEN	μPD703133AYGJ-xxx-UEN	μPD703136AYGJ-xxx-UEN
μPD703131AYGJ-xxx-UEN-A	μPD703133AYGJ-xxx-UEN-A	μPD703136AYGJ-xxx-UEN-A
μPD703132AGJ-xxx-UEN	μPD703134AGJ-xxx-UEN	μPD70F3134AGJ-UEN
μPD703132AGJ-xxx-UEN-A	μPD703134AGJ-xxx-UEN-A	μPD70F3134AGJ-UEN-A
μPD703132AYGJ-xxx-UEN	μPD703134AYGJ-xxx-UEN	μPD70F3134AYGJ-UEN
μPD703132AYGJ-xxx-UEN-A	μPD703134AYGJ-xxx-UEN-A	μPD70F3134AYGJ-UEN-A

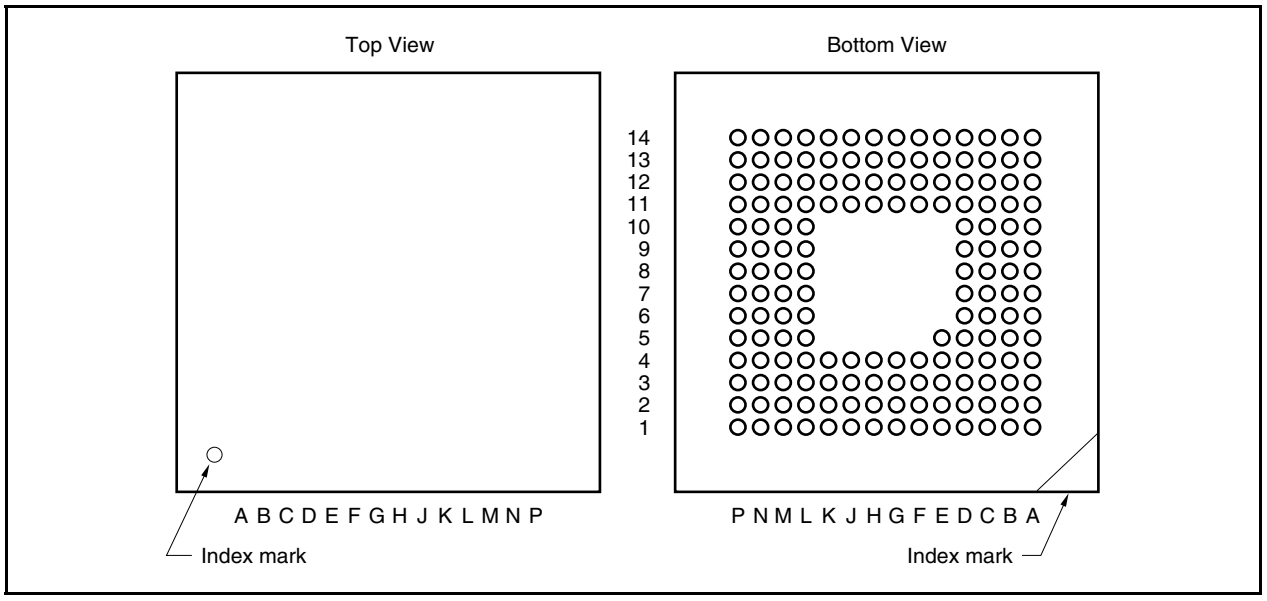
Top View



Note I^2C bus versions (Y products) only (see **Table 1-1**)

- 161-pin plastic FBGA (13 × 13)

  μPD703131AF1-xxx-EN4 μPD703133AF1-xxx-EN4 μPD703136AF1-xxx-EN4
  μPD703131AYF1-xxx-EN4 μPD703133AYF1-xxx-EN4 μPD703136AYF1-xxx-EN4
  μPD703132AF1-xxx-EN4 μPD703134AF1-xxx-EN4 μPD70F3134AF1-EN4
  μPD703132AYF1-xxx-EN4 μPD703134AYF1-xxx-EN4 μPD70F3134AF1-EN4-A
  μPD70F3134AYF1-EN4
  μPD70F3134AYF1-EN4-A



(1/2)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
A1	EV_{SS}	B10	A21/PAH5	D5	A6/PAL6
A2	AD15/PDL15	B11	A25/PAH9	D6	A10/PAL10
A3	A2/PAL2	B12	SDCLK/PCD1	D7	A14/PAL14
A4	A5/PAL5	B13	$\overline{CS1}$/PCS1	D8	A16/PAH0
A5	EV_{SS}	B14	EV_{SS}	D9	A20/PAH4
A6	A9/PAL9	C1	EV_{SS}	D10	A23/PAH7
A7	A12/PAL12	C2	AD9/PDL9	D11	SDCKE/PCD0
A8	A15/PAL15	C3	AD13/PDL13	D12	$\overline{CS0}$/PCS0
A9	A17/PAH1	C4	A1/PAL1	D13	$\overline{CS5}$/$\overline{IORD}$/PCS5
A10	–	C5	A7/PAL7	D14	EV_{SS}
A11	A24/PAH8	C6	EV_{DD}	E1	AD5/PDL5
A12	EV_{DD}	C7	A11/PAL11	E2	AD7/PDL7
A13	$\overline{SDCAS}$/PCD2	C8	V_{DD}	E3	AD8/PDL8
A14	$\overline{SDRAS}$/PCD3	C9	A19/PAH3	E4	AD11/PDL11
B1	EV_{SS}	C10	A22/PAH6	E5	–
B2	AD12/PDL12	C11	EV_{SS}	E11	$\overline{CS6}$/PCS6
B3	A0/PAL0	C12	$\overline{CS3}$/PCS3	E12	$\overline{CS4}$/PCS4
B4	A4/PAL4	C13	$\overline{CS2}$/$\overline{IOWR}$/PCS2	E13	$\overline{CS7}$/PCS7
B5	EV_{SS}	C14	EV_{SS}	E14	EV_{SS}
B6	A8/PAL8	D1	EV_{SS}	F1	AD2/PDL2
B7	A13/PAL13	D2	AD10/PDL10	F2	AD3/PDL3
B8	V_{SS}	D3	AD14/PDL14	F3	AD4/PDL4
B9	A18/PAH2	D4	A3/PAL3	F4	EV_{DD}

(2/2)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
F11	$\overline{RD}$/PCT4	L6	ASCK2/SCK2/$\overline{INTP132}$/P32	P5	EV_{SS}
F12	EV_{DD}	L7	ASCK1/SCK1/P45	P6	RXD1/SI1/P44
F13	$\overline{LBE}$/$\overline{LWR}$/LDQM/PCT0	L8	TXD0/SO0/P40	P7	RXD0/SI0/P41
F14	$\overline{UBE}$/$\overline{UWR}$/UDQM/PCT1	L9	MODE0	P8	PSEL
G1	TOP01/$\overline{INTP001}$/INTPP01/P01	L10	AV_{DD0}	P9	CV_{DD}
G2	TOP00/$\overline{INTP000}$/EVTP0/TIP0/INTPP00/P00	L11	ANI7/P77	P10	X1
G3	AD0/PDL0	L12	ANI4/P74	P11	–
G4	AD6/PDL6	L13	ANI3/P73	P12	$\overline{RESET}$
G11	$\overline{WAIT}$/PCM0	L14	ANI2/P72	P13	ANO0/P80
G12	$\overline{WR}$/$\overline{WE}$/PCT5	M1	EV_{SS}	P14	–
G13	$\overline{BCYST}$/PCT7	M2	$\overline{DMARQ1}$/TCUD10/INTP10/$\overline{INTP005}$/P05		
G14	ASTB/PCT6	M3	$\overline{DMARQ0}$/INTP11/TCLR10/$\overline{INTP004}$/P04		
H1	TOQB3/$\overline{INTP115}$/EVTQ/P15	M4	$\overline{DMAAK2}$/PBD2		
H2	TOQB2/$\overline{INTP114}$/TIQ/P14	M5	RXD3/SCL[Note]/$\overline{INTP134}$/P34		
H3	TOQT3/$\overline{INTP013}$/INTPQ3/TOQ3/P13	M6	RXD2/SI2/$\overline{INTP131}$/P31		
H4	AD1/PDL1	M7	TXD1/SO1/P43		
H11	$\overline{REFRQ}$/PCM4	M8	V_{DD}		
H12	$\overline{HLDRQ}$/PCM3	M9	CKSEL		
H13	$\overline{HLDAK}$/PCM2	M10	MODE1		
H14	BUSCLK/PCM1	M11	AV_{SS0}		
J1	V_{DD}	M12	ANI6/P76		
J2	TOQT2/$\overline{INTP012}$/INTPQ2/TOQ2/P12	M13	ANI5/P75		
J3	TOQB1/$\overline{INTP010}$/INTPQ0/TOQ0/P10	M14	–		
J4	V_{SS}	N1	EV_{SS}		
J11	ADTRG/$\overline{INTP137}$/P37	N2	$\overline{DMARQ3}$/TCK/$\overline{INTP107}$/P07		
J12	TOP21/INTPP21/$\overline{INTP051}$/P51	N3	$\overline{DMAAK3}$/PBD3		
J13	TOP20/INTPP20/EVTP2/TIP2/$\overline{INTP050}$/P50	N4	$\overline{DMAAK0}$/PBD0		
J14	NMI/P20	N5	TXD3/SDA[Note]/$\overline{INTP133}$/P33		
K1	TOQT1/$\overline{INTP011}$/INTPQ1/TOQ1/P11	N6	TXD2/SO2/$\overline{INTP130}$/P30		
K2	$\overline{TC3}$/TDO/P27	N7	ASCK0/SCK0/P42		
K3	$\overline{TC0}$/$\overline{INTP124}$/P24	N8	V_{SS}		
K4	$\overline{TC2}$/TDI/$\overline{INTP126}$/P26	N9	X2		
K11	ANI1/P71	N10	CV_{SS}		
K12	ANI0/P70	N11	ANO1/P81		
K13	V_{SS}	N12	AV_{SS1}		
K14	V_{DD}	N13	AV_{DD1}		
L1	EV_{SS}	N14	–		
L2	$\overline{TC1}$/TIUD10/TO10/$\overline{INTP125}$/P25	P1	EV_{DD}		
L3	$\overline{DMARQ2}$/TMS/$\overline{INTP106}$/P06	P2	EV_{SS}		
L4	$\overline{TRST}$	P3	$\overline{DMAAK1}$/PBD1		
L5	TOP11/INTPP11/$\overline{INTP022}$/P22	P4	TOP10/INTPP10/EVTP1/TIP1/$\overline{INTP021}$/P21		

Note I^2C bus versions (Y products) only (see **Table 1-1**)

Remark Leave the A10, E5, M14, N14, P11, and P14 pins open.

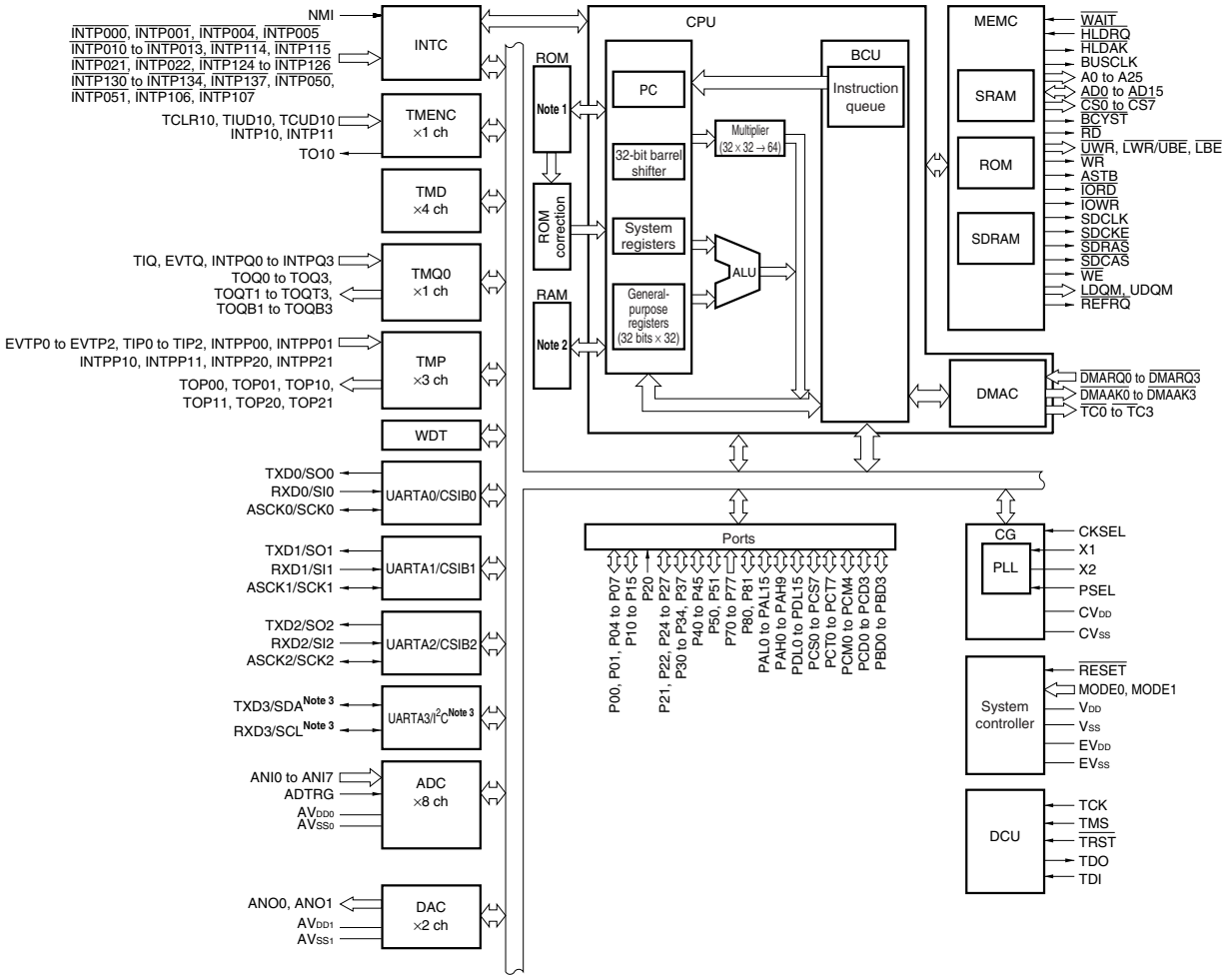
Pin Identification

A0 to A25:	Address bus	LDQM:	Lower DQ mask enable
AD0 to AD15:	Address/data bus	$\overline{\text{LWR}}$:	Lower byte write strobe
ADTRG:	A/D trigger input	MODE0, MODE1:	Mode
ANI0 to ANI7:	Analog input	NMI:	Non-maskable interrupt request
ANO0, ANO1:	Analog output	P00, P01, P04 to P07:	Port 0
ASCK0 to ASCK2:	Asynchronous serial clock	P10 to P15:	Port 1
ASTB:	Address strobe	P20 to P22, P24 to P27:	Port 2
AV_{DD0}, AV_{DD1}:	Analog power supply	P30 to P34, P37:	Port 3
AV_{SS0}, AV_{SS1}:	Analog ground	P40 to P45:	Port 4
$\overline{\text{BCYST}}$:	Bus cycle start timing	P50, P51:	Port 5
BUSCLK:	Bus clock output	P70 to P77:	Port 7
CKSEL:	Clock generator operating mode	P80, P81:	Port 8
	select	PAH0 to PAH9:	Port AH
$\overline{\text{CS0}}$ to $\overline{\text{CS7}}$:	Chip select	PAL0 to PAL15:	Port AL
CV_{DD}:	Power supply for clock generator	PBD0 to PBD3:	Port BD
CV_{SS}:	Ground for clock generator	PCD0 to PCD3:	Port CD
$\overline{\text{DMAAK0}}$ to $\overline{\text{DMAAK3}}$:	DMA acknowledge	PCM0 to PCM4:	Port CM
$\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$:	DMA request	PCS0 to PCS7:	Port CS
EV_{DD}:	Power supply for external pins	PCT0, PCT1,	
EV_{SS}:	Ground for external pins	PCT4 to PCT7:	Port CT
EVTP0 to EVTP2, EVTQ:	Timer event count input	PDL0 to PDL15:	Port DL
$\overline{\text{HLDAK}}$:	Hold acknowledge	PSEL:	PLL select
$\overline{\text{HLDRQ}}$:	Hold request	$\overline{\text{RD}}$:	Read strobe
$\overline{\text{INTP000}}$, $\overline{\text{INTP001}}$,		$\overline{\text{REFRQ}}$:	Refresh request
$\overline{\text{INTP004}}$, $\overline{\text{INTP005}}$,		$\overline{\text{RESET}}$:	Reset
$\overline{\text{INTP010}}$ to $\overline{\text{INTP013}}$,		RXD0 to RXD3:	Receive data
$\overline{\text{INTP021}}$, $\overline{\text{INTP022}}$,		SCK0 to SCK2:	Serial clock
$\overline{\text{INTP050}}$, $\overline{\text{INTP051}}$,		SCL:	Serial clock
$\overline{\text{INTP106}}$, $\overline{\text{INTP107}}$,		SDA:	Serial data
$\overline{\text{INTP114}}$ to $\overline{\text{INTP115}}$,		$\overline{\text{SDCAS}}$:	SDRAM column address strobe
$\overline{\text{INTP124}}$ to $\overline{\text{INTP126}}$,		SDCKE:	SDRAM clock enable
$\overline{\text{INTP130}}$ to $\overline{\text{INTP134}}$,		SDCLK:	SDRAM clock output
$\overline{\text{INTP137}}$:	External interrupt input	$\overline{\text{SDRAS}}$:	SDRAM row address strobe
INTP10, INTP11,		SI0 to SI2:	Serial input
INTPP00, INTPP01,		SO0 to SO2:	Serial output
INTPP10, INTPP11,		$\overline{\text{TC0}}$ to $\overline{\text{TC3}}$:	Terminal count signal
INTPP20, INTPP21,		TCK:	Debug clock
INTPQ0 to INTPQ3:	Timer input	TCLR10:	Timer clear
$\overline{\text{IORD}}$:	I/O read strobe	TCUD10:	Timer control pulse input
$\overline{\text{IOWR}}$:	I/O write strobe	TDI:	Debug data input
$\overline{\text{LBE}}$:	Lower byte enable	TDO:	Debug data output

TIP0 to TIP3, TIQ:	Timer trigger input
TIUD10:	Timer count pulse input
TMS:	Debug mode select
TO10, TOP00, TOP01, TOP10, TOP11, TOP20, TOP21, TOQ0 to TOQ3, TOQT1 to TOQT3, TOQB1 to TOQB3:	Timer output
$\overline{\text{TRST}}$:	Debug reset
TXD0 to TXD3:	Transmit data
$\overline{\text{UBE}}$:	Upper byte enable
UDQM:	Upper DQ mask enable
$\overline{\text{UWR}}$:	Upper byte write strobe
V_{DD}:	Power supply
V_{SS}:	Ground
$\overline{\text{WAIT}}$:	Wait
$\overline{\text{WE}}$:	Write enable
$\overline{\text{WR}}$:	Write strobe
X1, X2:	Crystal

1.6 Function Blocks

1.6.1 Internal block diagram



Notes 1. 256/512 KB (mask ROM) (see **Table 1-1**)
512 KB (flash memory) (see **Table 1-1**)

<R> **2.** 8/16/32 KB (see **Table 1-1**)

3. I^2C bus versions (Y products) only (see **Table 1-1**)

1.6.2 Internal units

(1) 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 or 32 bits × 32 bits → 64 bits) and a barrel shifter (32 bits), help accelerate complex processing.

(2) Bus control unit (BCU)

The BCU starts the required external bus cycle based on the physical address obtained by the CPU. When an instruction is fetched from external memory area and the CPU does not send a bus cycle start request, the BCU generates a prefetch address and prefetches the instruction code. The prefetched instruction code is stored in an instruction queue in the CPU.

The BCU controls a memory controller (MEMC) and DMA controller (DMAC) and performs external memory access and DMA transfer.

(a) Memory controller (MEMC)

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

(i) SRAM, external ROM, external I/O interface

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

Separate bus mode or multiplexed bus mode can be selected for the SRAM and external ROM interface.

(ii) SDRAM controller

The SDRAM controller generates the $\overline{\text{SDRAS}}$, $\overline{\text{SDCAS}}$, UDQM, and LDQM signals and controls access to SDRAM.

CAS latency 1, 2, and 3 are supported, and the burst length is fixed to 1.

(iii) Page ROM controller

This controller supports accessing ROM that includes a page access function.

It performs address comparisons with the immediately preceding bus cycle and controls wait for normal access (off-page)/page access (on-page).

(b) DMA controller (DMAC)

This controller controls data transfer between memory and I/O instead of the CPU.

There are two address modes: flyby (1-cycle) transfer, and 2-cycle transfer. There are three bus modes: single transfer, single-step transfer, and block transfer.

(3) ROM

This is mask ROM or flash memory of 256/512 KB mapped to addresses x0000000H to x003FFFFH/ x0000000H to x007FFFFH.

During instruction fetch, mask ROM/flash memory can be accessed from the CPU in 1 clock.

(4) RAM

<R> This is RAM of 8/16/32 KB mapped from addresses xFFFD000H to xFFFEFFFH/xFFFB000H to xFFFEFFFH/xFFF7000H to xFFFEFFFH.

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

(5) Interrupt controller (INTC)

This controller handles hardware interrupt requests (NMI, INTPn) from on-chip peripheral hardware and external hardware (n = 000, 001, 004, 005, 010 to 013, 021, 022, 050, 051, 106, 107, 114, 115, 124 to 126, 130 to 134, 137). Eight levels of interrupt priorities can be specified for these interrupt requests, and multiple-interrupt servicing control can be performed.

(6) Clock generator (CG)

The clock generator includes two basic operation modes: PLL mode 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}).

(7) Timer/counters (TMQ, TMP, TMD, TMENC)

This unit incorporates one 16-bit timer/event counter Q (TMQ) channel, three 16-bit timer/event counter P (TMP) channels, four 16-bit interval timer D (TMD) channels, and one up/down counter/general-purpose timer (TMENC) channel for 16-bit 2-phase encoder input, and can measure pulse interval widths or frequency, enable an inverter function for motor control, and output a programmable pulse.

(8) Watchdog timer (WDT)

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

It can also be used as an interval timer.

When used as a watchdog timer, it generates a non-maskable interrupt request signal (INTWDT) after an overflow occurs. When used as an interval timer, it generates a maskable interrupt request signal (INTWDTM) after an overflow occurs.

(9) Serial interface

The V850E/MA3 includes serial interface channels for asynchronous serial interface A (UARTA), clocked serial interface B (CSIB), and the I^2C bus interface (I^2C). Three of these channels are switchable between UARTA and CSIB and another is switchable between UARTA and I^2C.

For UARTA, data is transferred via the TXDn and RXDn pins (n = 0 to 3).

For CSIB, data is transferred via the SOn, SIn, and SCKn pins (n = 0 to 2).

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

(10) A/D converter (ADC)

This high-speed 10-bit A/D converter includes 8 analog input pins.

(11) D/A converter (DAC)

Two 8-bit-resolution D/A converter channels that use the R-2R ladder method are provided on chip.

(12) ROM correction

A ROM correction function that replaces part of a program in the mask ROM or flash memory with a program in the internal RAM is provided. Up to four correction addresses can be specified.

(13) On-chip debug function (DCU)

An on-chip debug function via an N-Wire type emulator is provided.

(14) Ports

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

Port	Port Function	Alternate Function
Port 0	6-bit I/O	Timer/counter I/O, external interrupt input, DMA controller input, debug input
Port 1	6-bit I/O	Timer/counter I/O, external interrupt input
Port 2	1-bit input, 6-bit I/O	NMI input, timer/counter I/O, external interrupt input, DMA controller output, debug I/O
Port 3	6-bit I/O	Serial interface I/O, external interrupt input, A/D converter external trigger input
Port 4	6-bit I/O	Serial interface I/O
Port 5	2-bit I/O	Timer/counter I/O, external interrupt input
Port 7	8-bit input	A/D converter input
Port 8	2-bit input	D/A converter output
Port AL	16-bit I/O	External address bus
Port AH	10-bit I/O	External address bus
Port DL	16-bit I/O	External address/data bus
Port CS	8-bit I/O	External bus interface control signal output
Port CT	6-bit I/O	External bus interface control signal output
Port CM	5-bit I/O	Wait insertion signal input, external bus interface control signal I/O
Port CD	4-bit I/O	External bus interface control signal output
Port BD	4-bit I/O	DMA controller output

CHAPTER 2 PIN FUNCTIONS

The names and functions of the pins in the V850E/MA3 are listed below. These pins can be divided into port pins and non-port pins according to their functions.

2.1 List of Pin Functions

(1) Port pins

(1/4)

Pin Name	Pin No.		I/O	Function	Alternate Function
	GJ	F1			
P00	19	G2	I/O	Port 0 6-bit I/O port Input data can be read/output data can be written in 1-bit units.	TOP00/$\overline{\text{INTP000}}$/EVTP0/TIP0/INTPP00
P01	18	G1			TOP01/$\overline{\text{INTP001}}$/INTPP01
P04	33	M3			$\overline{\text{DMARQ0}}$/INTP11/TCLR10/$\overline{\text{INTP004}}$
P05	34	M2			$\overline{\text{DMARQ1}}$/TCUD10/INTP10/$\overline{\text{INTP005}}$
P06	35	L3			$\overline{\text{DMARQ2}}$/TMS/$\overline{\text{INTP106}}$
P07	36	N2			$\overline{\text{DMARQ3}}$/TCK/$\overline{\text{INTP107}}$
P10	27	J3	I/O	Port 1 6-bit I/O port Input data can be read/output data can be written in 1-bit units.	TOQB1/$\overline{\text{INTP010}}$/INTPQ0/TOQ0
P11	26	K1			TOQT1/$\overline{\text{INTP011}}$/INTPQ1/TOQ1
P12	25	J2			TOQT2/$\overline{\text{INTP012}}$/INTPQ2/TOQ2
P13	22	H3			TOQT3/$\overline{\text{INTP013}}$/INTPQ3/TOQ3
P14	21	H2			TOQB2/$\overline{\text{INTP114}}$/TIQ
P15	20	H1			TOQB3/$\overline{\text{INTP115}}$/EVTQ
P20	84	J14	Input	Port 2 P20 is an input-only port. If a valid edge is input, it operates as an NMI input. Also, the status of the NMI input is shown by bit 0 of the P2 register. P21, P22, P24 to P27 are 6-bit I/O port pins. Input data can be read/output data can be written in 1-bit units.	NMI
P21	44	P4	I/O		TOP10/INTPP10/EVTP1/TIP1/$\overline{\text{INTP021}}$
P22	43	L5			TOP11/INTPP11/$\overline{\text{INTP022}}$
P24	31	K3			$\overline{\text{TC0}}$/$\overline{\text{INTP124}}$
P25	30	L2			$\overline{\text{TC1}}$/TIUD10/TO10/$\overline{\text{INTP125}}$
P26	29	K4			$\overline{\text{TC2}}$/TDI/$\overline{\text{INTP126}}$
P27	28	K2			$\overline{\text{TC3}}$/TDO
P30	49	N6	I/O	Port 3 6-bit I/O port Input data can be read/output data can be written in 1-bit units.	TXD2/SO2/$\overline{\text{INTP130}}$
P31	48	M6			RXD2/SI2/$\overline{\text{INTP131}}$
P32	47	L6			ASCK2/SCK2/$\overline{\text{INTP132}}$
P33	46	N5			TXD3/SDA[Note]/$\overline{\text{INTP133}}$
P34	45	M5			RXD3/SCL[Note]/$\overline{\text{INTP134}}$
P37	83	J11			ADTRG/$\overline{\text{INTP137}}$

Note I^2C bus versions (Y products) only (see **Table 1-1**)

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(2/4)

Pin Name	Pin No.		I/O	Function	Alternate Function
	GJ	F1			
P40	55	L8	I/O	Port 4 6-bit I/O port Input data can be read/output data can be written in 1-bit units.	TXD0/SO0
P41	54	P7			RXD0/SI0
P42	53	N7			ASCK0/SCK0
P43	52	M7			TXD1/SO1
P44	51	P6			RXD1/SI1
P45	50	L7			ASCK1/SCK1
P50	86	J13	I/O	Port 5 2-bit I/O port Input data can be read/output data can be written in 1-bit units.	TOP20/INTPP20/EVTP2/TIP2/$\overline{\text{INTP050}}$
P51	85	J12			TOP21/INTPP21/$\overline{\text{INTP051}}$
P70	80	K12	Input	Port 7 8-bit input-only port	ANI0
P71	79	K11			ANI1
P72	78	L14			ANI2
P73	77	L13			ANI3
P74	76	L12			ANI4
P75	75	M13			ANI5
P76	74	M12			ANI6
P77	73	L11			ANI7
P80	69	P13	Input	Port 8 2-bit input-only port	ANO0
P81	68	N11			ANO1
PAL0	143	B3	I/O	Port AL 8-/16-bit I/O port Input data can be read/output data can be written in 1-bit units.	A0
PAL1	142	C4			A1
PAL2	141	A3			A2
PAL3	140	D4			A3
PAL4	139	B4			A4
PAL5	138	A4			A5
PAL6	137	D5			A6
PAL7	136	C5			A7
PAL8	133	B6			A8
PAL9	132	A6			A9
PAL10	131	D6			A10
PAL11	130	C7			A11
PAL12	129	A7			A12
PAL13	128	B7			A13
PAL14	127	D7			A14
PAL15	126	A8			A15

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(3/4)

Pin Name	Pin No.		I/O	Function	Alternate Function
	GJ	F1			
PAH0	123	D8	I/O	Port AH 8-/10-bit I/O port Input data can be read/output data can be written in 1-bit units.	A16
PAH1	122	A9			A17
PAH2	121	B9			A18
PAH3	120	C9			A19
PAH4	119	D9			A20
PAH5	118	B10			A21
PAH6	117	C10			A22
PAH7	116	D10			A23
PAH8	115	A11			A24
PAH9	114	B11			A25
PDL0	17	G3	I/O	Port DL 8-/16-bit I/O port Input data can be read/output data can be written in 1-bit units.	AD0
PDL1	16	H4			AD1
PDL2	15	F1			AD2
PDL3	14	F2			AD3
PDL4	13	F3			AD4
PDL5	12	E1			AD5
PDL6	11	G4			AD6
PDL7	10	E2			AD7
PDL8	7	E3			AD8
PDL9	6	C2			AD9
PDL10	5	D2			AD10
PDL11	4	E4			AD11
PDL12	3	B2			AD12
PDL13	2	C3			AD13
PDL14	1	D3			AD14
PDL15	144	A2			AD15
PCS0	107	D12	I/O	Port CS 8-bit I/O port Input data can be read/output data can be written in 1-bit units.	$\overline{CS0}$
PCS1	106	B13			$\overline{CS1}$
PCS2	105	C13			$\overline{CS2}$/$\overline{IOWR}$
PCS3	104	C12			$\overline{CS3}$
PCS4	103	E12			$\overline{CS4}$
PCS5	102	D13			$\overline{CS5}$/$\overline{IORD}$
PCS6	101	E11			$\overline{CS6}$
PCS7	100	E13			$\overline{CS7}$

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(4/4)

Pin Name	Pin No.		I/O	Function	Alternate Function
	GJ	F1			
PCT0	97	F13	I/O	Port CT 6-bit I/O port Input data can be read/output data can be written in 1-bit units.	$\overline{\text{LBE}}$/$\overline{\text{LWR}}$/LDQM
PCT1	96	F14			$\overline{\text{UBE}}$/$\overline{\text{UWR}}$/UDQM
PCT4	95	F11			$\overline{\text{RD}}$
PCT5	94	G12			$\overline{\text{WR}}$/$\overline{\text{WE}}$
PCT6	93	G14			ASTB
PCT7	92	G13			$\overline{\text{BCYST}}$
PCM0	91	G11	I/O	Port CM 5-bit I/O port Input data can be read/output data can be written in 1-bit units.	$\overline{\text{WAIT}}$
PCM1	90	H14			BUSCLK
PCM2	89	H13			$\overline{\text{HLDAK}}$
PCM3	88	H12			$\overline{\text{HLDRQ}}$
PCM4	87	H11			$\overline{\text{REFRQ}}$
PCD0	111	D11	I/O	Port CD 4-bit I/O port Input data can be read/output data can be written in 1-bit units.	SDCKE
PCD1	110	B12			SDCLK
PCD2	109	A13			$\overline{\text{SDCAS}}$
PCD3	108	A14			$\overline{\text{SDRAS}}$
PBD0	42	N4	I/O	Port BD 4-bit I/O port Input data can be read/output data can be written in 1-bit units.	$\overline{\text{DMAAK0}}$
PBD1	41	P3			$\overline{\text{DMAAK1}}$
PBD2	40	M4			$\overline{\text{DMAAK2}}$
PBD3	39	N3			$\overline{\text{DMAAK3}}$

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(2) Non-port pins

(1/6)

Pin Name	Pin No.		I/O	Function	Alternate Function
	GJ	F1			
A0	143	B3	Output	26-bit address bus for external memory	PAL0
A1	142	C4			PAL1
A2	141	A3			PAL2
A3	140	D4			PAL3
A4	139	B4			PAL4
A5	138	A4			PAL5
A6	137	D5			PAL6
A7	136	C5			PAL7
A8	133	B6			PAL8
A9	132	A6			PAL9
A10	131	D6			PAL10
A11	130	C7			PAL11
A12	129	A7			PAL12
A13	128	B7			PAL13
A14	127	D7			PAL14
A15	126	A8			PAL15
A16	123	D8			PAH0
A17	122	A9			PAH1
A18	121	B9			PAH2
A19	120	C9			PAH3
A20	119	D9			PAH4
A21	118	B10			PAH5
A22	117	C10			PAH6
A23	116	D10			PAH7
A24	115	A11			PAH8
A25	114	B11			PAH9
AD0	17	G3	I/O	16-bit address/data bus for external memory	PDL0
AD1	16	H4			PDL1
AD2	15	F1			PDL2
AD3	14	F2			PDL3
AD4	13	F3			PDL4
AD5	12	E1			PDL5
AD6	11	G4			PDL6
AD7	10	E2			PDL7
AD8	7	E3			PDL8
AD9	6	C2			PDL9
AD10	5	D2			PDL10
AD11	4	E4			PDL11

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(2/6)

Pin Name	Pin No.		I/O	Function	Alternate Function
	GJ	F1			
AD12	3	B2	I/O	16-bit address/data bus for external memory	PDL12
AD13	2	C3			PDL13
AD14	1	D3			PDL14
AD15	144	A2			PDL15
ADTRG	83	J11	Input	A/D converter external trigger input	$\overline{\text{INTP137}}$/P37
ANI0	80	K12	Input	Analog inputs for A/D converter	P70
ANI1	79	K11			P71
ANI2	78	L14			P72
ANI3	77	L13			P73
ANI4	76	L12			P74
ANI5	75	M13			P75
ANI6	74	M12			P76
ANI7	73	L11			P77
ANO0	69	P13	Output	Analog outputs for D/A converter	P80
ANO1	68	N11			P81
ASCK0	53	N7	Input	UARTA0 to UARTA2 serial baud rate clock input	SCK0/P42
ASCK1	50	L7			SCK1/P45
ASCK2	47	L6			SCK2/$\overline{\text{INTP132}}$/P32
ASTB	93	G14	Output	Address strobe output of external data bus	PCT6
AV_{DD0}	72	L10	–	Positive power supply for A/D converter (3.3 V)	–
AV_{DD1}	67	N13	–	Positive power supply for D/A converter (3.3 V)	–
AV_{SS0}	71	M11	–	Ground potential for A/D converter	–
AV_{SS1}	70	N12	–	Ground potential for D/A converter	–
$\overline{\text{BCYST}}$	92	G13	Output	Bus cycle start output	PCT7
BUSCLK	90	H14	Output	Bus clock output	PCM1
CKSEL	60	M9	Input	Clock generator operating mode specification	–
$\overline{\text{CS0}}$	107	D12	Output	Chip select output	PCS0
$\overline{\text{CS1}}$	106	B13			PCS1
$\overline{\text{CS2}}$	105	C13			PCS2/$\overline{\text{IOWR}}$
$\overline{\text{CS3}}$	104	C12			PCS3
$\overline{\text{CS4}}$	103	E12			PCS4
$\overline{\text{CS5}}$	102	D13			PCS5/$\overline{\text{IORD}}$
$\overline{\text{CS6}}$	101	E11			PCS6
$\overline{\text{CS7}}$	100	E13			PCS7
CV_{DD}	56	P9	–	Positive power supply for OSC pin (3.3 V)	–
CV_{SS}	59	N10	–	Ground potential for OSC pin	–
$\overline{\text{DMAAK0}}$	42	N4	Output	DMA transfer acknowledge output	PBD0
$\overline{\text{DMAAK1}}$	41	P3			PBD1
$\overline{\text{DMAAK2}}$	40	M4			PBD2
$\overline{\text{DMAAK3}}$	39	N3			PBD3

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(3/6)

Pin Name	Pin No.		I/O	Function	Alternate Function
	GJ	F1			
$\overline{\text{DMARQ0}}$	33	M3	Input	DMA transfer request input	INTP11/TCLR10/$\overline{\text{INTP004}}$/P04
$\overline{\text{DMARQ1}}$	34	M2			TCUD10/INTP10/$\overline{\text{INTP005}}$/P05
$\overline{\text{DMARQ2}}$	35	L3			TMS/$\overline{\text{INTP106}}$/P06
$\overline{\text{DMARQ3}}$	36	N2			TCK/$\overline{\text{INTP107}}$/P07
EV_{DD}	Note 1	Note 1	–	Positive power supply for external pin (3.3 V)	–
EV_{SS}	Note 2	Note 2	–	Ground potential for external pin	–
EVTP0	19	G2	Input	TMP0 to TMP2 external event count input	TOP00/$\overline{\text{INTP000}}$/TIP0/INTPP00/P00
EVTP1	44	P4			TOP10/INTPP10/TIP1/$\overline{\text{INTP021}}$/P21
EVTP2	86	J13			TOP20/INTPP20/TIP2/$\overline{\text{INTP050}}$/P50
EVTQ	20	H1	Input	TMQ0 external event count input	TOQB3/$\overline{\text{INTP115}}$/P15
$\overline{\text{HLDAK}}$	89	H13	Output	Bus hold acknowledge output	PCM2
$\overline{\text{HLDRQ}}$	88	H12	Input	Bus hold request input	PCM3
INTP10	34	M2	Input	TMENC10 external capture trigger input	$\overline{\text{DMARQ1}}$/TCUD10/$\overline{\text{INTP005}}$/P05
INTP11	33	M3			$\overline{\text{DMARQ0}}$/TCLR10/$\overline{\text{INTP004}}$/P04
$\overline{\text{INTP000}}$	19	G2	Input	External maskable interrupt request input	TOP00/EVTP0/TIP0/INTPP00/P00
$\overline{\text{INTP001}}$	18	G1			TOP01/INTPP01/P01
$\overline{\text{INTP004}}$	33	M3			$\overline{\text{DMARQ0}}$/INTP11/TCLR10/P04
$\overline{\text{INTP005}}$	34	M2			$\overline{\text{DMARQ1}}$/TCUD10/INTP10/P05
$\overline{\text{INTP106}}$	35	L3			$\overline{\text{DMARQ2}}$/TMS/P06
$\overline{\text{INTP107}}$	36	N2			$\overline{\text{DMARQ3}}$/TCK/P07
$\overline{\text{INTP010}}$	27	J3			TOQB1/INTPQ0/TOQ0/P10
$\overline{\text{INTP011}}$	26	K1			TOQT1/INTPQ1/TOQ1/P11
$\overline{\text{INTP012}}$	25	J2			TOQT2/INTPQ2/TOQ2/P12
$\overline{\text{INTP013}}$	22	H3			TOQT3/INTPQ3/TOQ3/P13
$\overline{\text{INTP114}}$	21	H2			TOQB2/TIQ/P14
$\overline{\text{INTP115}}$	20	H1			TOQB3/EVTQ/P15
$\overline{\text{INTP021}}$	44	P4			TOP10/INTPP10/EVTP1/TIP1/P21
$\overline{\text{INTP022}}$	43	L5			TOP11/INTPP11/P22
$\overline{\text{INTP124}}$	31	K3			$\overline{\text{TC0}}$/P24
$\overline{\text{INTP125}}$	30	L2			$\overline{\text{TC1}}$/TIUD10/TO10/P25
$\overline{\text{INTP126}}$	29	K4			$\overline{\text{TC2}}$/TDI/P26
$\overline{\text{INTP130}}$	49	N6			TXD2/SO2/P30
$\overline{\text{INTP131}}$	48	M6			RXD2/SI2/P31
$\overline{\text{INTP132}}$	47	L6			ASCK2/SCK2/P32

Notes 1. GJ: 8, 37, 98, 112, 134
F1: A12, C6, F4, F12, P1

2. GJ: 9, 38, 99, 113, 135
F1: A1, A5, B1, B5, B14, C1, C11, C14, D1, D14, E14, L1, M1, N1, P2, P5

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(4/6)

Pin Name	Pin No.		I/O	Function	Alternate Function
	GJ	F1			
$\overline{\text{INTP133}}$	46	N5	Input	External maskable interrupt request input	TXD3/SDA[Note]/P33
$\overline{\text{INTP134}}$	45	M5			RXD3/SCL[Note]/P34
$\overline{\text{INTP137}}$	83	J11			ADTRG/P37
$\overline{\text{INTP050}}$	86	J13			TOP20/INTPP20/EVTP2/TIP2/P50
$\overline{\text{INTP051}}$	85	J12			TOP21/INTPP21/P51
INTPP00	19	G2	Input	TMP0 to TMP2 external capture trigger input	TOP00/$\overline{\text{INTP000}}$/EVTP0/TIP0/P00
INTPP01	18	G1			TOP01/$\overline{\text{INTP001}}$/P01
INTPP10	44	P4			TOP10/EVTP1/TIP1/$\overline{\text{INTP021}}$/P21
INTPP11	43	L5			TOP11/$\overline{\text{INTP022}}$/P22
INTPP20	86	J13			TOP20/EVTP2/TIP2/$\overline{\text{INTP050}}$/P50
INTPP21	85	J12			TOP21/$\overline{\text{INTP051}}$/P51
INTPQ0	27	J3	Input	TMQ0 external capture trigger input	TOQB1/$\overline{\text{INTP010}}$/TOQ0/P10
INTPQ1	26	K1			TOQT1/$\overline{\text{INTP011}}$/TOQ1/P11
INTPQ2	25	J2			TOQT2/$\overline{\text{INTP012}}$/TOQ2/P12
INTPQ3	22	H3			TOQT3/$\overline{\text{INTP013}}$/TOQ3/P13
$\overline{\text{IORD}}$	102	D13	Output	I/O read strobe output	PCS5/$\overline{\text{CS5}}$
$\overline{\text{IOWR}}$	105	C13	Output	I/O write strobe output	PCS2/$\overline{\text{CS2}}$
$\overline{\text{LBE}}$	97	F13	Output	External data bus byte enable output (D0 to D7)	PCT0/$\overline{\text{LWR}}$/LDQM
LDQM	97	F13	Output	I/O mask signal output for SDRAM (D0 to D7)	PCT0/$\overline{\text{LBE}}$/$\overline{\text{LWR}}$
$\overline{\text{LWR}}$	97	F13	Output	External data bus write strobe output (D0 to D7)	PCT0/$\overline{\text{LBE}}$/LDQM
MODE0	64	L9	Input	Operation mode specification	–
MODE1	65	M10			–
NMI	84	J14	Input	Non-maskable interrupt request input	P20
PSEL	61	P8	Input	Input frequency select signal input in PLL mode	–
$\overline{\text{RD}}$	95	F11	Output	External data bus read strobe output	PCT4
$\overline{\text{REFRQ}}$	87	H11	Output	Refresh request output for SDRAM	PCM4
$\overline{\text{RESET}}$	66	P12	Input	System reset input	–
RXD0	54	P7	Input	UARTA0 to UARTA3 serial receive data input	SI0/P41
RXD1	51	P6			SI1/P44
RXD2	48	M6			SI2/$\overline{\text{INTP131}}$/P31
RXD3	45	M5			SCL[Note]/$\overline{\text{INTP134}}$/P34
SCK0	53	N7	I/O	CSIB0 to CSIB2 serial clock I/O	ASCK0/P42
SCK1	50	L7			ASCK1/P45
SCK2	47	L6			ASCK2/$\overline{\text{INTP132}}$/P32
SCL[Note]	45	M5	I/O	I^2C serial clock I/O	RXD3/$\overline{\text{INTP134}}$/P34
SDA[Note]	46	N5	I/O	I^2C data I/O	TXD3/$\overline{\text{INTP133}}$/P33
$\overline{\text{SDCAS}}$	109	A13	Output	Column address strobe output for SDRAM	PCD2
SDCKE	111	D11	Output	Clock enable output for SDRAM	PCD0
SDCLK	110	B12	Output	Clock output for SDRAM	PCD1

Note I^2C bus versions (Y products) only (see **Table 1-1**)

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(5/6)

Pin Name	Pin No.		I/O	Function	Alternate Function
	GJ	F1			
$\overline{\text{SDRAS}}$	108	A14	Output	Row address strobe output for SDRAM	PCD3
SI0	54	P7	Input	CSIB0 to CSIB2 serial receive data input	RXD0/P41
SI1	51	P6			RXD1/P44
SI2	48	M6			RXD2/$\overline{\text{INTP131}}$/P31
SO0	55	L8	Output	CSIB0 to CSIB2 serial transmit data output	TXD0/P40
SO1	52	M7			TXD1/P43
SO2	49	N6			TXD2/$\overline{\text{INTP130}}$/P30
$\overline{\text{TC0}}$	31	K3	Output	DMA transfer end (terminal count) output	$\overline{\text{INTP124}}$/P24
$\overline{\text{TC1}}$	30	L2			TIUD10/TO10/$\overline{\text{INTP125}}$/P25
$\overline{\text{TC2}}$	29	K4			TDI/$\overline{\text{INTP126}}$/P26
$\overline{\text{TC3}}$	28	K2			TDO/P27
TCK	36	N2	Input	Debug clock input for N-Wire type emulator	$\overline{\text{DMARQ3}}$/$\overline{\text{INTP107}}$/P07
TCLR10	33	M3	Input	Clear signal input to TMENC10	$\overline{\text{DMARQ0}}$/INTP11/$\overline{\text{INTP004}}$/P04
TCUD10	34	M2	Input	Count operation switching signal for TMENC10	$\overline{\text{DMARQ1}}$/INTP10/$\overline{\text{INTP005}}$/P05
TDI	29	K4	Input	Debug data input for N-Wire type emulator	$\overline{\text{TC2}}$/$\overline{\text{INTP126}}$/P26
TDO	28	K2	Output	Debug data output for N-Wire type emulator	$\overline{\text{TC3}}$/P27
TIP0	19	G2	Input	TMP0 to TMP2 external timer trigger input	TOP00/$\overline{\text{INTP000}}$/EVTP0/INTPP00/P00
TIP1	44	P4			TOP10/INTPP10/EVTP1/$\overline{\text{INTP021}}$/P21
TIP2	86	J13			TOP20/INTPP20/EVTP2/$\overline{\text{INTP050}}$/P50
TIQ	21	H2	Input	TMQ0 external timer trigger input	TOQB2/$\overline{\text{INTP114}}$/P14
TIUD10	30	L2	Input	External count clock input of TMENC10	$\overline{\text{TC1}}$/TO10/$\overline{\text{INTP125}}$/P25
TMS	35	L3	Input	Debug mode select for N-Wire type emulator	$\overline{\text{DMARQ2}}$/$\overline{\text{INTP106}}$/P06
TO10	30	L2	Output	TMENC10 pulse signal output	$\overline{\text{TC1}}$/TIUD10/$\overline{\text{INTP125}}$/P25
TOP00	19	G2	Output	TMP0 to TMP2 pulse signal output	$\overline{\text{INTP000}}$/EVTP0/TIP0/INTPP00/P00
TOP01	18	G1			$\overline{\text{INTP001}}$/INTPP01/P01
TOP10	44	P4			INTPP10/EVTP1/TIP1/$\overline{\text{INTP021}}$/P21
TOP11	43	L5			INTPP11/$\overline{\text{INTP022}}$/P22
TOP20	86	J13			INTPP20/EVTP2/TIP2/$\overline{\text{INTP050}}$/P50
TOP21	85	J12			INTPP21/$\overline{\text{INTP051}}$/P51
TOQ0	27	J3	Output	TMQ0 pulse signal output	TOQB1/$\overline{\text{INTP010}}$/INTPQ0/P10
TOQ1	26	K1			TOQT1/$\overline{\text{INTP011}}$/INTPQ1/P11
TOQ2	25	J2			TOQT2/$\overline{\text{INTP012}}$/INTPQ2/P12
TOQ3	22	H3			TOQT3/$\overline{\text{INTP013}}$/INTPQ3/P13

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(6/6)

Pin Name	Pin No.		I/O	Function	Alternate Function
	GJ	F1			
TOQB1	27	J3	Output	Pulse signal output for 6-phase PWM	$\overline{\text{INTP010}}$/INTPQ0/TOQ0/P10
TOQB2	21	H2			$\overline{\text{INTP114}}$/TIQ/P14
TOQB3	20	H1			$\overline{\text{INTP115}}$/EVTQ/P15
TOQT1	26	K1			$\overline{\text{INTP011}}$/INTPQ1/TOQ1/P11
TOQT2	25	J2			$\overline{\text{INTP012}}$/INTPQ2/TOQ2/P12
TOQT3	22	H3			$\overline{\text{INTP013}}$/INTPQ3/TOQ3/P13
$\overline{\text{TRST}}$	32	L4	Input	Debug reset input for N-Wire type emulator	–
TXD0	55	L8	Output	UARTA0 to UARTA3 serial transmit data output	SO0/P40
TXD1	52	M7			SO1/P43
TXD2	49	N6			SO2/$\overline{\text{INTP130}}$/P30
TXD3	46	N5			SDA[Note 1]/$\overline{\text{INTP133}}$/P33
$\overline{\text{UBE}}$	96	F14	Output	External data bus byte enable output (D8 to D15)	PCT1/$\overline{\text{UWR}}$/UDQM
UDQM	96	F14	Output	I/O mask signal output for SDRAM (D8 to D15)	PCT1/$\overline{\text{UBE}}$/$\overline{\text{UWR}}$
$\overline{\text{UWR}}$	96	F14	Output	External data bus write strobe output (D8 to D15)	PCT1/$\overline{\text{UBE}}$/UDQM
V_{DD}	**Note 2**	**Note 2**	–	Positive power supply for internal units (2.5 V)	–
V_{SS}	**Note 3**	**Note 3**	–	Ground potential for internal units	–
$\overline{\text{WAIT}}$	91	G11	Input	External wait request input	PCM0
$\overline{\text{WE}}$	94	G12	Output	Write enable output for SDRAM	PCT5/$\overline{\text{WR}}$
$\overline{\text{WR}}$	94	G12	Output	Write strobe output for external data bus	PCT5/$\overline{\text{WE}}$
X1	58	P10	Input	Crystal connection for system clock oscillator/external clock input (X2 is open when external clock is input)	–
X2	57	N9	–		–

Notes 1. I^2C bus versions (Y products) only (see **Table 1-1**)

2. GJ: 23, 62, 81, 124
F1: C8, J1, K14, M8

3. GJ: 24, 63, 82, 125
F1: B8, J4, K13, N8

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

2.2 Pin Status

The status of each pin after reset, in power-save mode (HALT, IDLE, or software STOP mode), and during DMA transfer, refresh, and bus hold (TH) is shown below.

Operating Status / Pin	Reset (Single-Chip Mode)	IDLE Mode/Software STOP Mode	HALT Mode/During DMA Transfer/ Refresh	Bus Hold (TH)[Note 2]
A0 to A15 (PAL0 to PAL15)	Hi-Z	Hi-Z	Operating	Hi-Z
A16 to A25 (PAH0 to PAH9)	Hi-Z	Hi-Z	Operating	Hi-Z
AD0 to AD15 (PDL0 to PDL15)	Hi-Z	Hi-Z	Operating	Hi-Z
$\overline{CS0}$ to $\overline{CS7}$ (PCS0 to PCS7)	Hi-Z	SELF	Operating	Hi-Z
$\overline{IOWR}$ (PCS2)	–	H	Operating	Hi-Z
$\overline{IORD}$ (PCS5)	–	H	Operating	Hi-Z
$\overline{LWR}$, $\overline{UWR}$ (PCT0, PCT1)	Hi-Z	H	Operating	Hi-Z
$\overline{LBE}$, $\overline{UBE}$ (PCT0, PCT1)	–	H	Operating	Hi-Z
LDQM, UDQM (PCT0, PCT1)	–	H	Operating	Hi-Z
$\overline{RD}$ (PCT4)	Hi-Z	H	Operating	Hi-Z
$\overline{WR}$ (PCT5)	Hi-Z	H	Operating	Hi-Z
$\overline{WE}$ (PCT5)	–	H	Operating	Hi-Z
ASTB (PCT6)	Hi-Z	H	Operating	Hi-Z
$\overline{BCYST}$ (PCT7)	Hi-Z	H	Operating	Hi-Z
$\overline{WAIT}$ (PCM0)	Hi-Z	Input not sampled	Operating	Input not sampled
BUSCLK (PCM1)	Hi-Z	L	Operating	Operating
$\overline{HLDAK}$ (PCM2)	Hi-Z	H	Operating	L
$\overline{HLDRQ}$ (PCM3)	Hi-Z	Input not sampled	Operating	Operating
$\overline{REFRQ}$ (PCM4)	Hi-Z	L[Note 1]	Operating	Operating
SDCKE (PCD0)	Hi-Z	L	Operating	H
SDCLK (PCD1)	Hi-Z	L	Operating	Operating
$\overline{SDCAS}$ (PCD2)	Hi-Z	SELF	Operating	Hi-Z
$\overline{SDRAS}$ (PCD3)	Hi-Z	SELF	Operating	Hi-Z
$\overline{DMAAK0}$ to $\overline{DMAAK3}$ (PBD0 to PBD3)	Hi-Z	H	Operating	H

Notes 1. High-level output when the SDRAM controller is not used.

2. The pin set in the port mode holds the status immediately before.

Remark Hi-Z: High-impedance
H: High-level output
L: Low-level output
SELF: Self-refresh state when pins are connected to SDRAM

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

It is recommended that 1 to 10 kΩ resistors be used when connecting to V_{DD} or V_{SS} via resistors.

(1/4)

Pin Name	Alternate-Function Pin Name	Pin No.		I/O Circuit Type	Recommended Connection
		GJ	F1		
P00	TOP00/$\overline{\text{INTP000}}$/EVTP0/TIP0/INTPP00	19	G2	5-K	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P01	TOP01/$\overline{\text{INTP001}}$/INTPP01	18	G1		
P04	$\overline{\text{DMARQ0}}$/INTP11/TCLR10/$\overline{\text{INTP004}}$	33	M3		
P05	$\overline{\text{DMARQ1}}$/TCUD10/INTP10/$\overline{\text{INTP005}}$	34	M2		
P06	$\overline{\text{DMARQ2}}$/TMS/$\overline{\text{INTP106}}$	35	L3		
P07	$\overline{\text{DMARQ3}}$/TCK/$\overline{\text{INTP107}}$	36	N2		
P10	TOQB1/$\overline{\text{INTP010}}$/INTPQ0/TOQ0	27	J3		
P11	TOQT1/$\overline{\text{INTP011}}$/INTPQ1/TOQ1	26	K1		
P12	TOQT2/$\overline{\text{INTP012}}$/INTPQ2/TOQ2	25	J2		
P13	TOQT3/$\overline{\text{INTP013}}$/INTPQ3/TOQ3	22	H3		
P14	TOQB2/$\overline{\text{INTP114}}$/TIQ	21	H2		
P15	TOQB3/$\overline{\text{INTP115}}$/EVTQ	20	H1		
P20	NMI	84	J14	2	Independently connect to EV_{SS} via a resistor.
P21	TOP10/INTPP10/EVTP1/TIP1/$\overline{\text{INTP021}}$	44	P4	5-K	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P22	TOP11/INTPP11/$\overline{\text{INTP022}}$	43	L5		
P24	$\overline{\text{TC0}}$/$\overline{\text{INTP124}}$	31	K3		
P25	$\overline{\text{TC1}}$/TIUD10/TO10/$\overline{\text{INTP125}}$	30	L2		
P26	$\overline{\text{TC2}}$/TDI/$\overline{\text{INTP126}}$	29	K4		
P27	$\overline{\text{TC3}}$/TDO	28	K2	5	
P30	TXD2/SO2/$\overline{\text{INTP130}}$	49	N6	5-K	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P31	RXD2/SI2/$\overline{\text{INTP131}}$	48	M6		
P32	ASCK2/SCK2/$\overline{\text{INTP132}}$	47	L6		
P33	TXD3/SDA[Note]/$\overline{\text{INTP133}}$	46	N5		
P34	RXD3/SCL[Note]/$\overline{\text{INTP134}}$	45	M5		
P37	ADTRG/$\overline{\text{INTP137}}$	83	J11		
P40	TXD0/SO0	55	L8	5	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P41	RXD0/SI0	54	P7	5-K	
P42	ASCK0/SCK0	53	N7		
P43	TXD1/SO1	52	M7	5	
P44	RXD1/SI1	51	P6	5-K	
P45	ASCK1/SCK1	50	L7		

Note I^2C bus versions (Y products) only (see **Table 1-1**)

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(2/4)

Pin Name	Alternate-Function Pin Name	Pin No.		I/O Circuit Type	Recommended Connection
		GJ	F1		
P50	TOP20/INTPP20/EVTP2/TIP2/$\overline{\text{INTP050}}$	86	J13	5-K	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P51	TOP21/INTPP21/$\overline{\text{INTP051}}$	85	J12		
P70	ANI0	80	K12	9	Independently connect to AV_{DD0} or AV_{SS0} via a resistor.
P71	ANI1	79	K11		
P72	ANI2	78	L14		
P73	ANI3	77	L13		
P74	ANI4	76	L12		
P75	ANI5	75	M13		
P76	ANI6	74	M12		
P77	ANI7	73	L11		
P80	ANO0	69	P13	34	Independently connect to AV_{DD1} or AV_{SS1} via a resistor.
P81	ANO1	68	N11		
PAL0	A0	143	B3	5	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
PAL1	A1	142	C4		
PAL2	A2	141	A3		
PAL3	A3	140	D4		
PAL4	A4	139	B4		
PAL5	A5	138	A4		
PAL6	A6	137	D5		
PAL7	A7	136	C5		
PAL8	A8	133	B6		
PAL9	A9	132	A6		
PAL10	A10	131	D6		
PAL11	A11	130	C7		
PAL12	A12	129	A7		
PAL13	A13	128	B7		
PAL14	A14	127	D7		
PAL15	A15	126	A8		
PAH0	A16	123	D8		
PAH1	A17	122	A9		
PAH2	A18	121	B9		
PAH3	A19	120	C9		
PAH4	A20	119	D9		
PAH5	A21	118	B10		
PAH6	A22	117	C10		
PAH7	A23	116	D10		
PAH8	A24	115	A11		
PAH9	A25	114	B11		

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(3/4)

Pin Name	Alternate-Function Pin Name	Pin No.		I/O Circuit Type	Recommended Connection
		GJ	F1		
PDL0	AD0	17	G3	5	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
PDL1	AD1	16	H4		
PDL2	AD2	15	F1		
PDL3	AD3	14	F2		
PDL4	AD4	13	F3		
PDL5	AD5	12	E1		
PDL6	AD6	11	G4		
PDL7	AD7	10	E2		
PDL8	AD8	7	E3		
PDL9	AD9	6	C2		
PDL10	AD10	5	D2		
PDL11	AD11	4	E4		
PDL12	AD12	3	B2		
PDL13	AD13	2	C3		
PDL14	AD14	1	D3		
PDL15	AD15	144	A2		
PCS0	$\overline{CS0}$	107	D12		
PCS1	$\overline{CS1}$	106	B13		
PCS2	$\overline{CS2}$/$\overline{IOWR}$	105	C13		
PCS3	$\overline{CS3}$	104	C12		
PCS4	$\overline{CS4}$	103	E12		
PCS5	$\overline{CS5}$/$\overline{IORD}$	102	D13		
PCS6	$\overline{CS6}$	101	E11		
PCS7	$\overline{CS7}$	100	E13		
PCT0	$\overline{LBE}$/$\overline{LWR}$/LDQM	97	F13		
PCT1	$\overline{UBE}$/$\overline{UWR}$/UDQM	96	F14		
PCT4	$\overline{RD}$	95	F11		
PCT5	$\overline{WR}$/$\overline{WE}$	94	G12		
PCT6	ASTB	93	G14		
PCT7	$\overline{BCYST}$	92	G13		
PCM0	$\overline{WAIT}$	91	G11		
PCM1	BUSCLK	90	H14		
PCM2	$\overline{HLDAK}$	89	H13		
PCM3	$\overline{HLDRQ}$	88	H12		
PCM4	$\overline{REFRQ}$	87	H11		
PCD0	SDCKE	111	D11		
PCD1	SDCLK	110	B12		
PCD2	$\overline{SDCAS}$	109	A13		
PCD3	$\overline{SDRAS}$	108	A14		

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

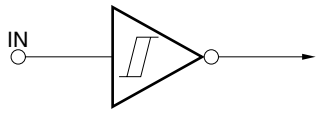
(4/4)

Pin Name	Alternate-Function Pin Name	Pin No.		I/O Circuit Type	Recommended Connection
		GJ	F1		
PBD0	$\overline{\text{DMAAK0}}$	42	N4	5	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
PBD1	$\overline{\text{DMAAK1}}$	41	P3		
PBD2	$\overline{\text{DMAAK2}}$	40	M4		
PBD3	$\overline{\text{DMAAK3}}$	39	N3		
$\overline{\text{TRST}}$	–	32	L4	2	Independently connect to EV_{SS} via a resistor.
$\overline{\text{RESET}}$	–	66	P12		–
MODE0	–	64	L9		–
MODE1	–	65	M10		–
PSEL	–	61	P8		–
CKSEL	–	60	M9		–
X2	–	57	N9	–	Leave open.

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

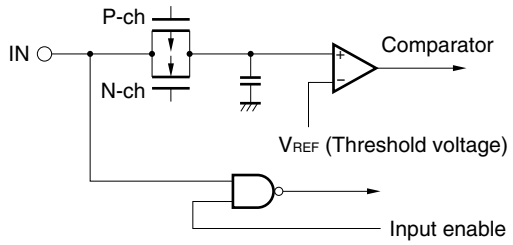
2.4 Pin I/O Circuits

Type 2

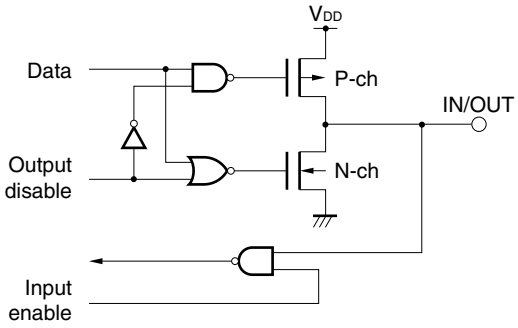


Schmitt-triggered input with hysteresis characteristics

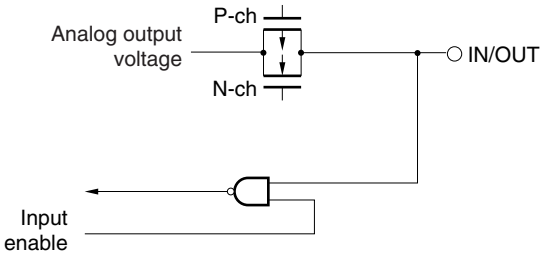
Type 9



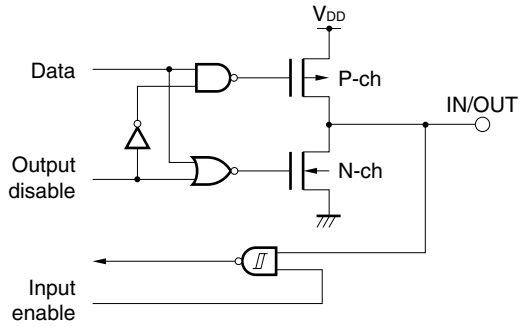
Type 5



Type 34



Type 5-K



Caution Type 2 or 5-K pins have hysteresis characteristics when their alternate function is used in the input mode, but do not have hysteresis characteristics when they are used in the port mode.

CHAPTER 3 CPU FUNCTION

The CPU of the V850E/MA3 is based on RISC architecture and executes almost all the instructions in one clock cycle using 5-stage pipeline control.

3.1 Features

- Minimum instruction execution time: 12.5 ns (@80 MHz internal operation)
- Memory space Program space: 64 MB linear
  Data space: 4 GB linear
- Thirty-two 32-bit general-purpose registers
- Internal 32-bit architecture
- Five-stage pipeline control
- Multiply/divide instructions
- Saturated operation instructions
- One-clock 32-bit shift instruction
- Load/store instruction with long/short instruction format
- Four types of bit manipulation instructions
  - SET1
  - CLR1
  - NOT1
  - TST1

3.2 CPU Register Set

The registers of the V850E/MA3 can be classified into two categories: a general-purpose program register set and a dedicated system register set. All the registers have a 32-bit width.

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

Figure 3-1. CPU Register Set

(1) Program register set

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

31	0
PC	(Program counter)

(2) System register set

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

3.2.1 Program register set

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

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

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

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

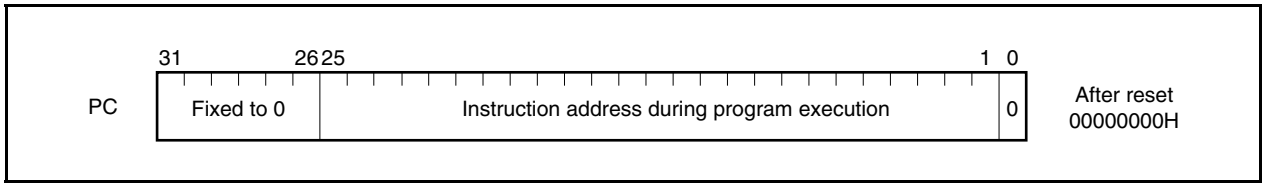
Table 3-1. General-Purpose Registers

Name	Usage	Operation
r0	Zero register	Always holds 0
r1	Assembler-reserved register	Working register for generating 32-bit immediate data
r2	Address/data variable register (when r2 is not used by the real-time OS)	
r3	Stack pointer	Used to generate stack frame when function is called
r4	Global pointer	Used to access global variable in data area
r5	Text pointer	Register to indicate the start of the text area (where program code is located)
r6 to r29	Address/data variable registers	
r30	Element pointer	Base pointer when memory is accessed
r31	Link pointer	Used by compiler when calling function

(2) Program counter (PC)

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

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



3.2.2 System register set

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

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

Table 3-2. System Register Numbers

No.	System Register Name	Operand Specification	
		LDSR Instruction	STSR Instruction
0	Status saving register during interrupt (EIPC)[Note 1]	○	○
1	Status saving register during interrupt (EIPSW)[Note 1]	○	○
2	Status saving register during NMI (FEPC)	○	○
3	Status saving register during NMI (FEPSW)	○	○
4	Interrupt source register (ECR)	×	○
5	Program status word (PSW)	○	○
6 to 15	Reserved for future function expansion (operations that access these register numbers cannot be guaranteed).	×	×
16	Status saving register during CALLT execution (CTPC)	○	○
17	Status saving register during CALLT execution (CTPSW)	○	○
18	Status saving register during exception/debug trap (DBPC)	○[Note 2]	○[Note 2]
19	Status saving register during exception/debug trap (DBPSW)	○[Note 2]	○[Note 2]
20	CALLT base pointer (CTBP)	○	○
21, 22	Reserved for future function expansion (operations that access these register numbers cannot be guaranteed).	×	×
23	Program ID register (ASID)	○	○
24 to 31	Reserved for future function expansion (operations that access these register numbers cannot be guaranteed).	×	×

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

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

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

Remark ○: Access allowed
×: Access prohibited

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

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

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

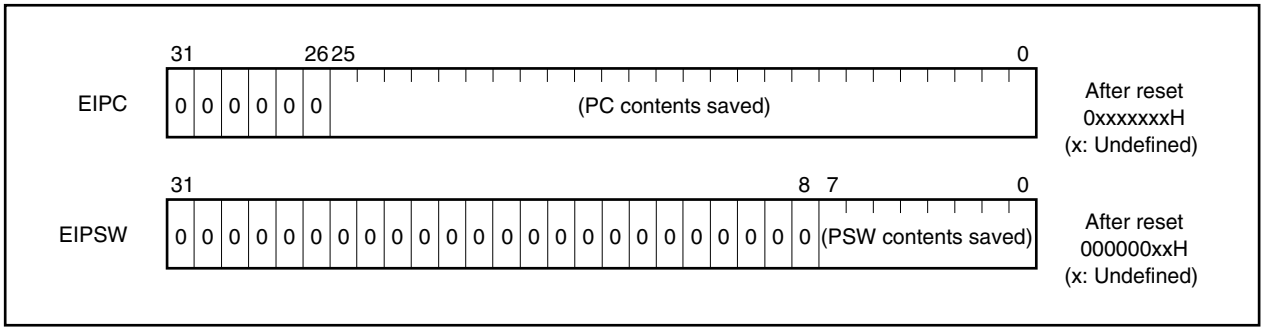
The address of the next instruction following the instruction executed when a software exception or maskable interrupt occurs is saved to EIPC, except for some instructions (see **20.9 Period in Which CPU Does Not Acknowledge Interrupts**).

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.

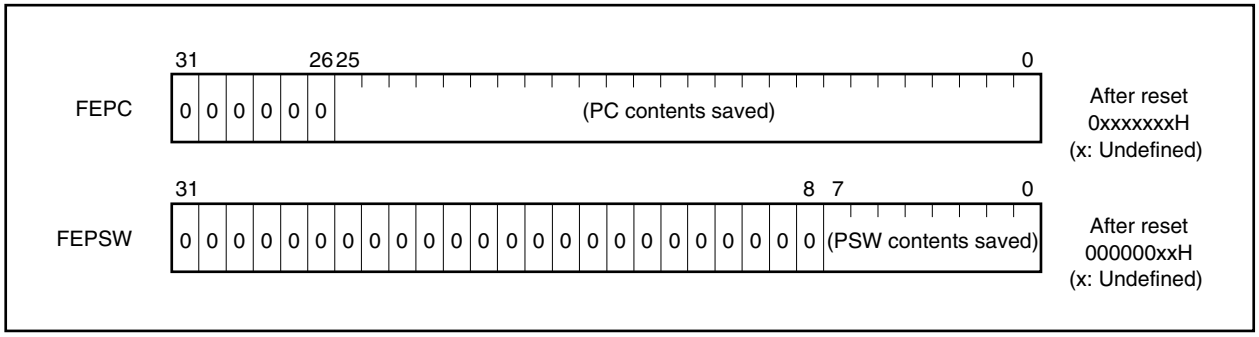
Upon occurrence of a non-maskable interrupt (NMI), the contents of the program counter (PC) are saved to FEPC and the contents of the program status word (PSW) are saved to FEPSW.

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.

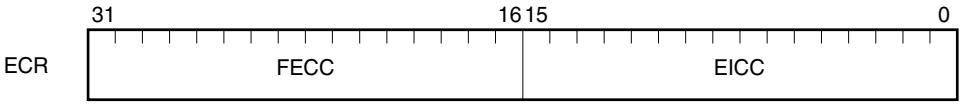
Bits 31 to 26 of FEPC and bits 31 to 8 of FEPSW are reserved (fixed to 0) for future function expansion.

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

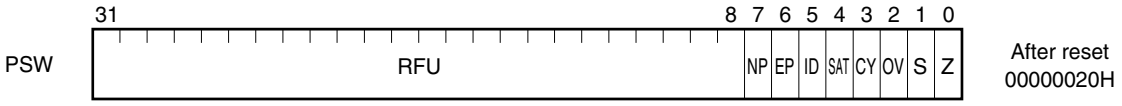
(4) Program status word (PSW)

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)

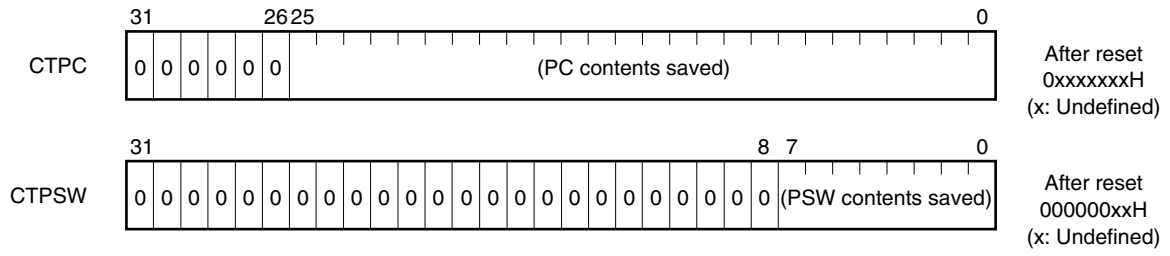
There are two CALLT execution status saving registers, CTPC and CTPSW.

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 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.

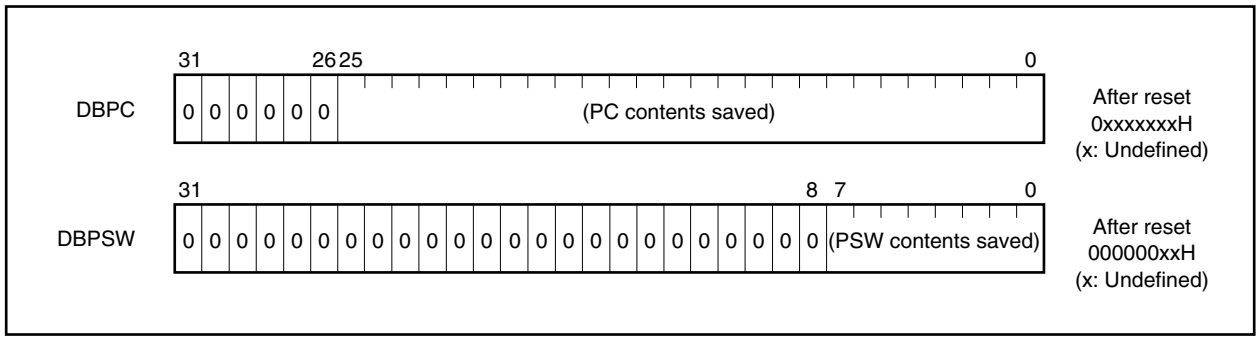
The contents saved to DBPC consist of the address of the next instruction after the instruction executed when an exception trap or debug trap occurs.

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.

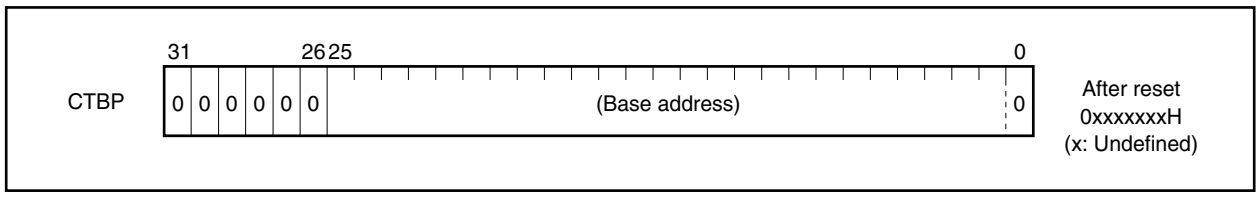
When the DBRET instruction has been executed, the values of DBPC and DBPSW are restored to the PC and PSW, respectively.



(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.

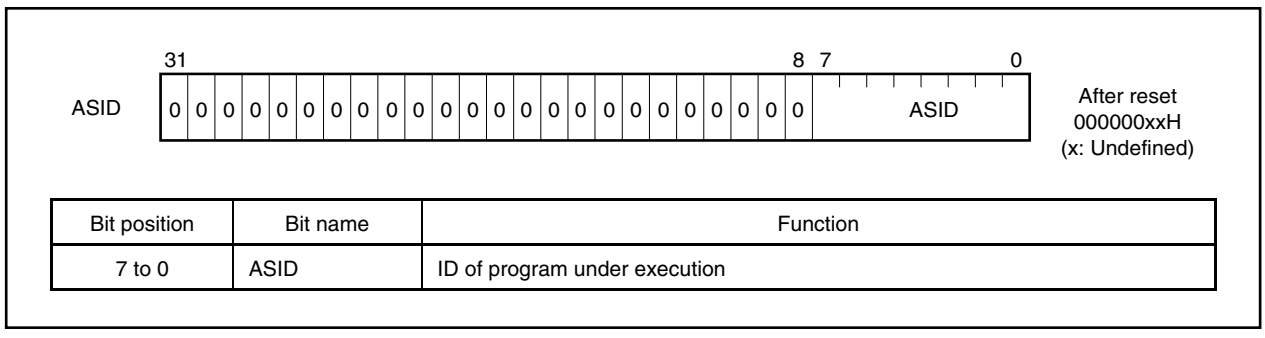


(8) Program ID register (ASID)

The ASID register sets the ID of the program in progress.

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

Caution To use the V850E/MA3, initialize the ASID register to 00H in its initialization routine.



Bit position	Bit name	Function
7 to 0	ASID	ID of program under execution

3.3 Operating Modes

3.3.1 Operating modes

The V850E/MA3 has the following operating modes. Mode specification is carried out using the MODE0 and MODE1 pins.

(1) Normal operation mode

(a) Single-chip mode

Access to the internal ROM is enabled.

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

(2) Flash memory programming mode (flash memory versions only (see Table 1-1))

If this mode is specified, programming the on-chip flash memory by the flash programmer is enabled.

3.3.2 Operating mode specification

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

MODE1	MODE0	Operation Mode	Remarks
L	L	Normal operation mode (single-chip mode)	Internal ROM area is allocated from address 000000H.
H	L	Flash memory programming mode[Note]	–
Other than above		Setting prohibited	

Note Flash memory versions only (see **Table 1-1**)

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

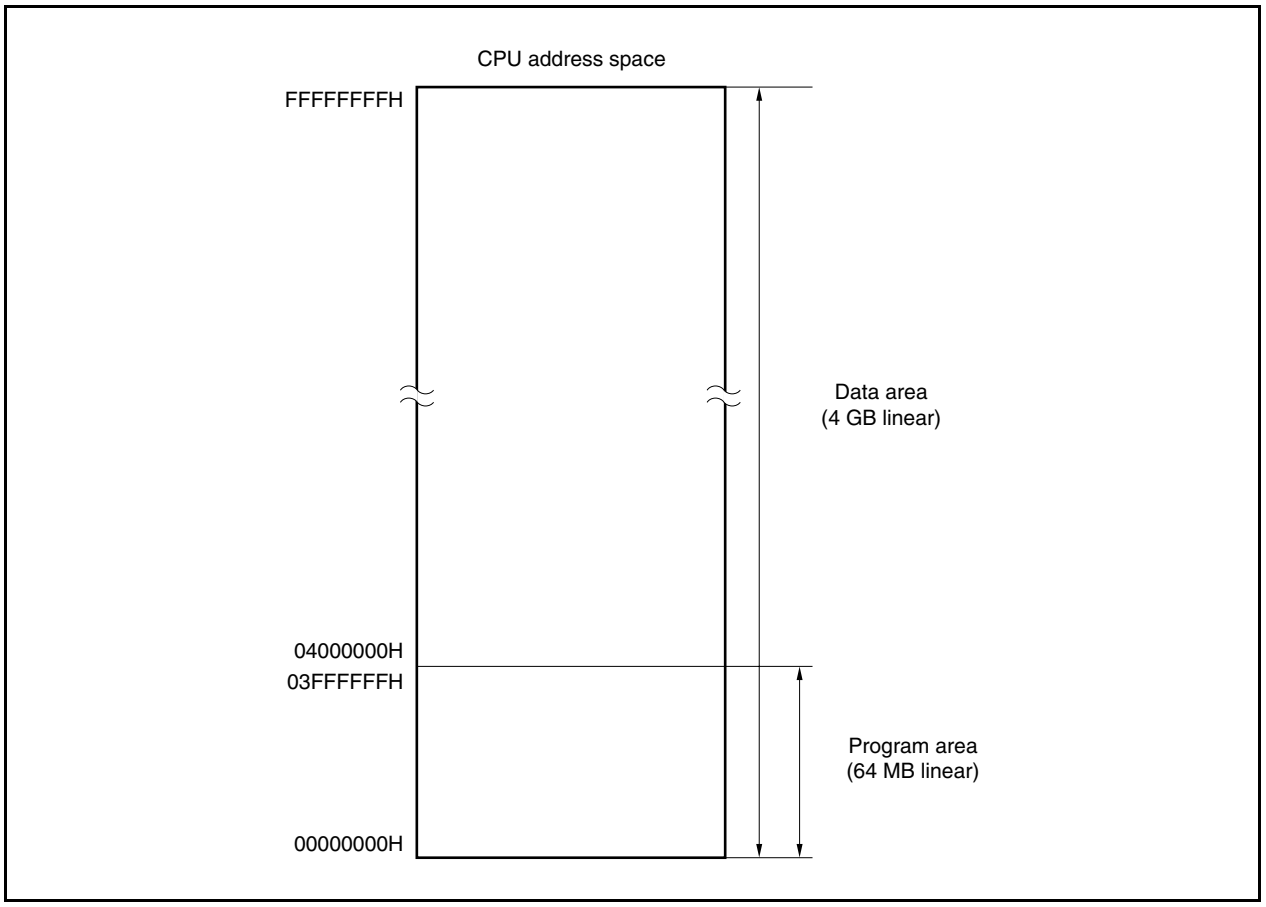
3.4 Address Space

3.4.1 CPU address space

The CPU of the V850E/MA3 has 32-bit architecture and supports up to 4 GB of linear address space (data space) during operand addressing (data access). Also, in instruction address addressing, a maximum of 64 MB of linear address space (program space) is supported.

Figure 3-2 shows the CPU address space.

Figure 3-2. CPU Address Space

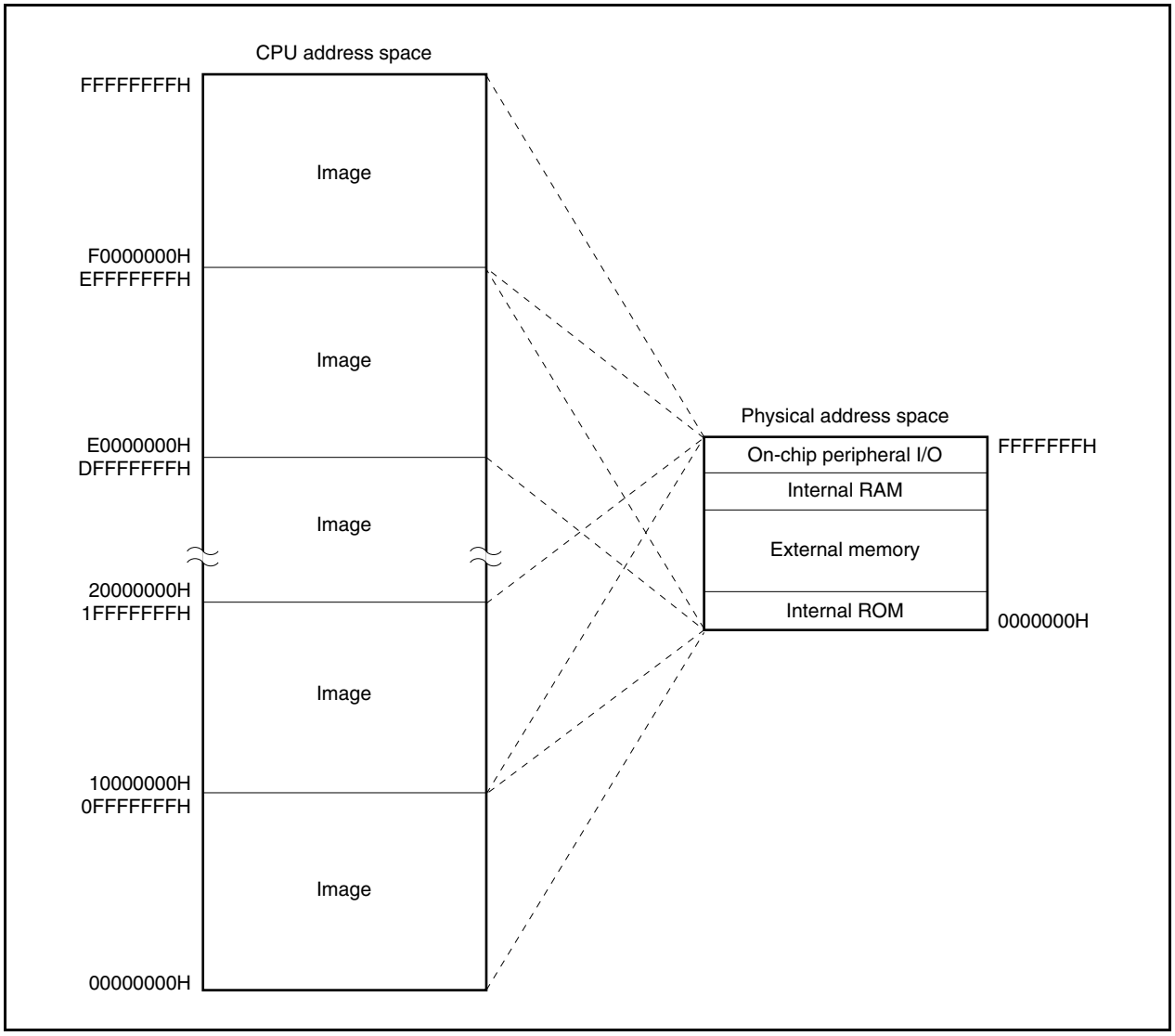


3.4.2 Image

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

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

Figure 3-3. Images on Address Space



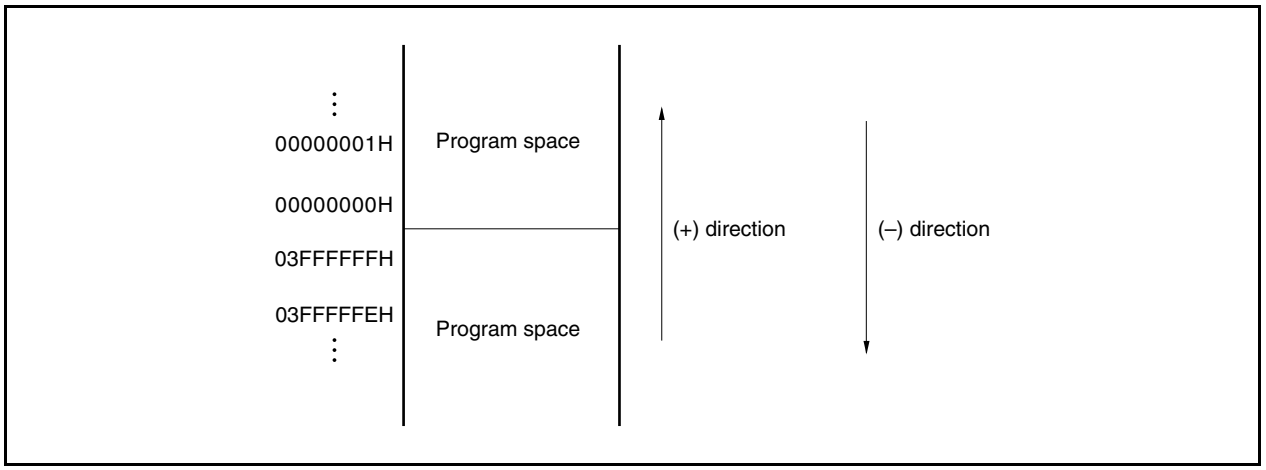
3.4.3 Wraparound of CPU address space

(1) Program space

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

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

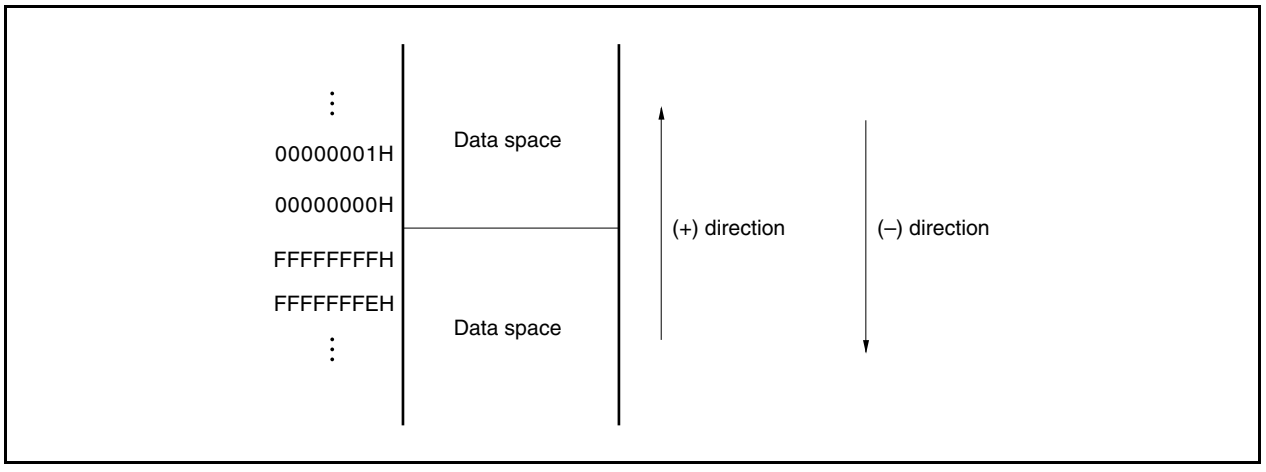
Caution The 4 KB area of 03FFF000H to 03FFFFFFH can be seen as an image of 0FFFF000H to 0FFFFFFFH. This area is access-prohibited. Therefore, do not execute any branch address calculation in which the result will reside in any part of this area.



(2) Data space

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

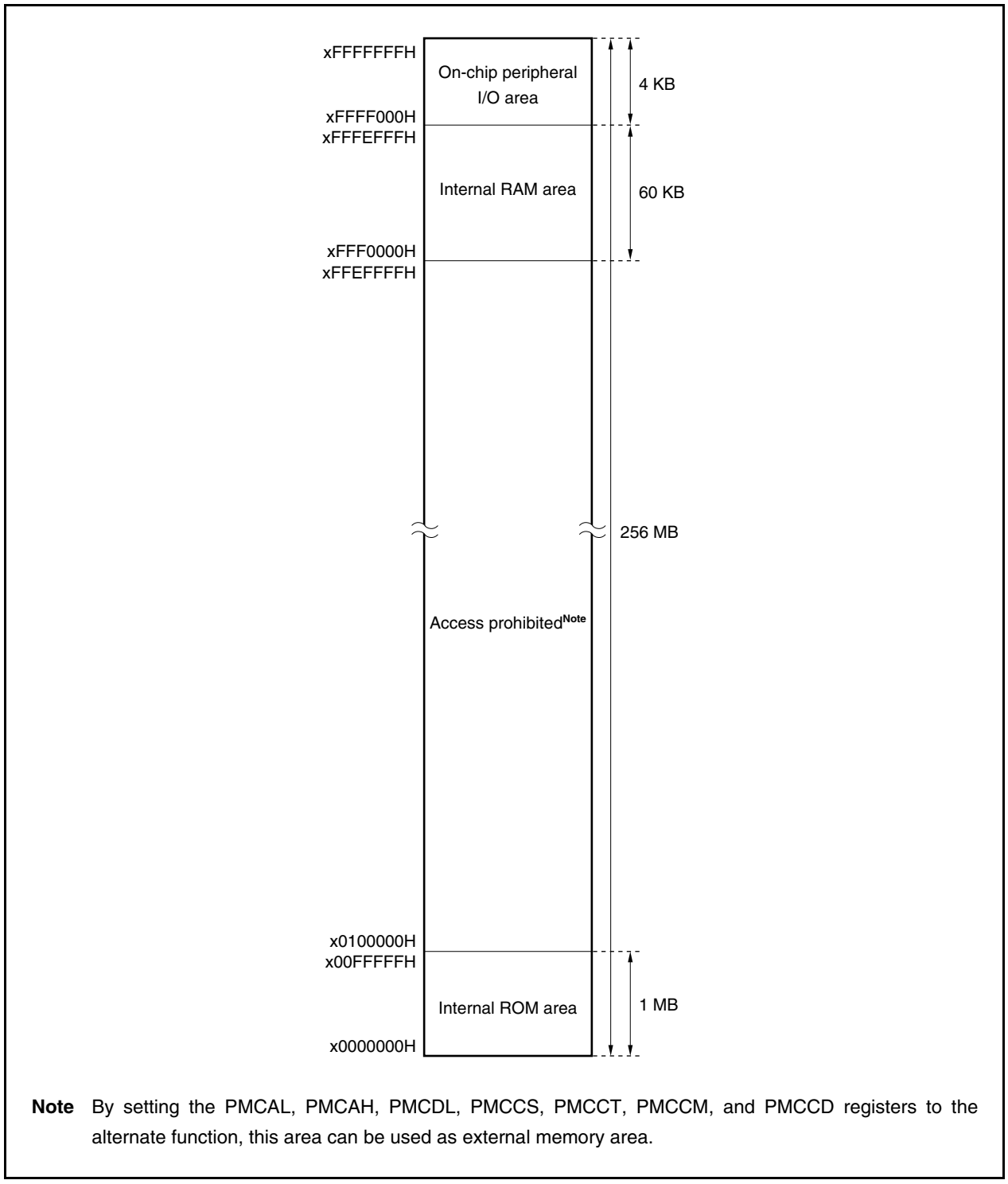
Therefore, the upper-limit address of the program space, address FFFFFFFFH, and the lower-limit address 00000000H are contiguous addresses, and the data space is wrapped around at the boundary of these addresses.



3.4.4 Memory map

The V850E/MA3 reserves areas as shown in Figure 3-4.

Figure 3-4. Memory Map



Note By setting the PMCAL, PMCAH, PMCDL, PMCCS, PMCCT, PMCCM, and PMCCD registers to the alternate function, this area can be used as external memory area.

3.4.5 Area

(1) Internal ROM area

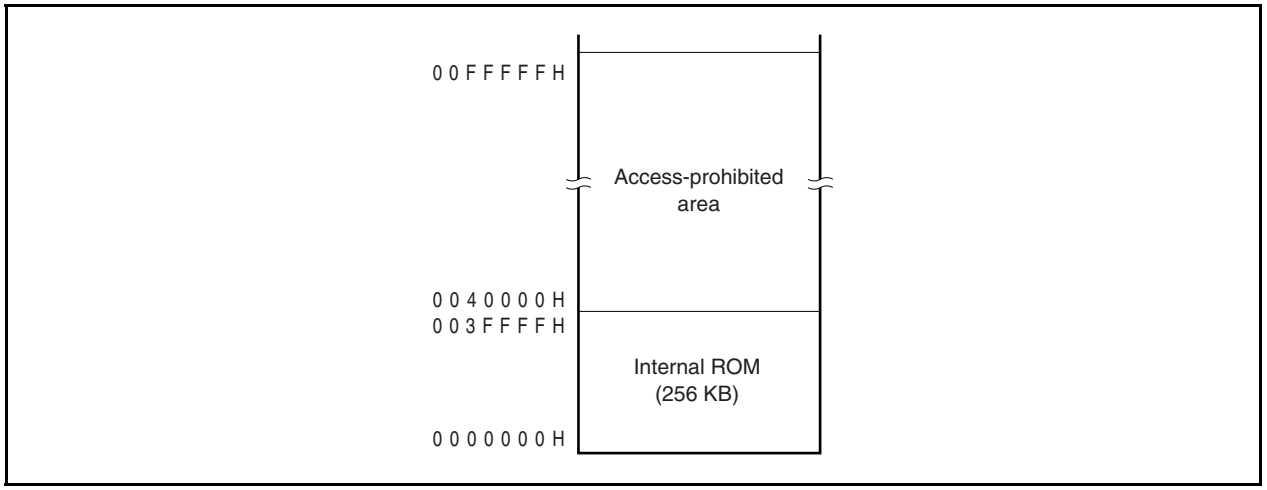
1 MB of addresses 0000000H to 00FFFFFH is reserved as an internal ROM area.

(a) Internal ROM (256 KB)

256 KB are allocated to addresses 0000000H to 003FFFFH in the following versions.
Accessing addresses 0040000H to 00FFFFFH is prohibited.

<R> • μPD703131A, 703131AY, 703132A, 703132AY, 703136A, 703136AY

Figure 3-5. Internal ROM Area (256 KB)

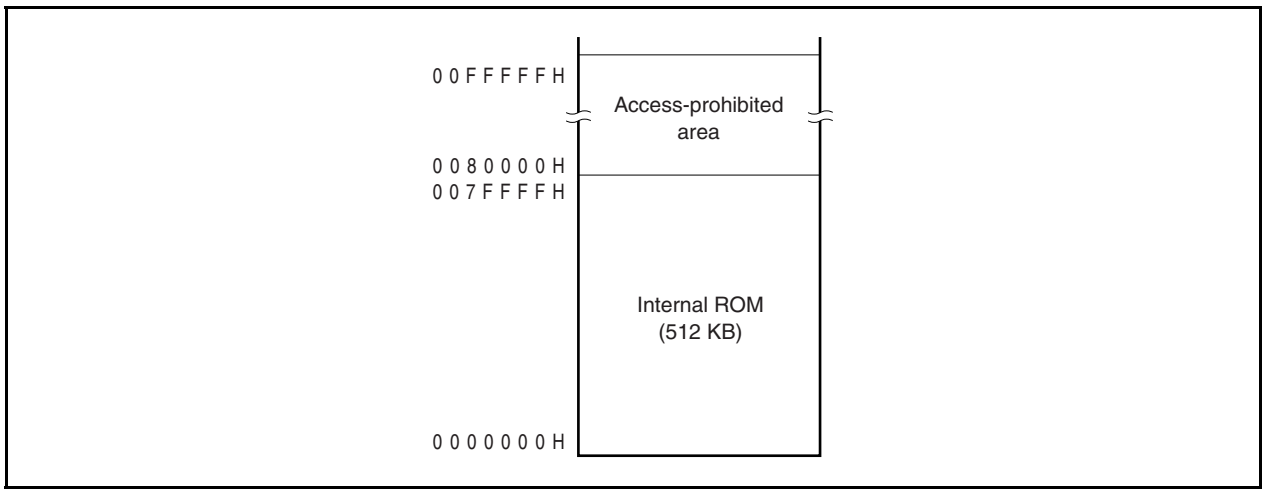


(b) Internal ROM (512 KB)

512 KB are allocated to addresses 0000000H to 007FFFFH in the following versions.
Accessing addresses 0080000H to 00FFFFFH is prohibited.

<R> • μPD703133A, 703133AY, 703134A, 703134AY, 70F3134A, 70F3134AY

Figure 3-6. Internal ROM Area (512 KB)



(2) Internal RAM area

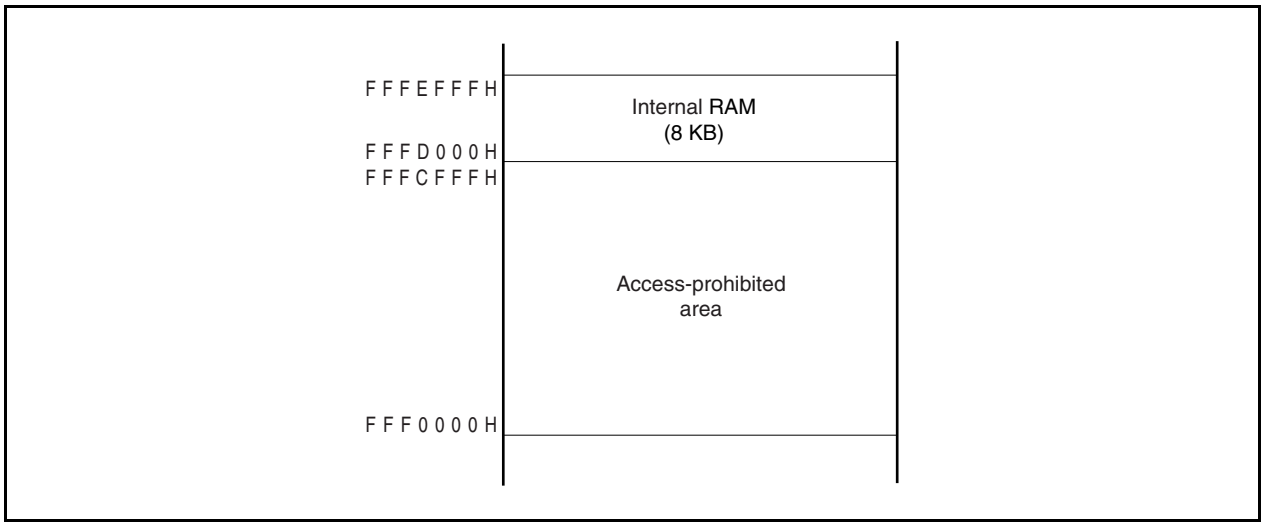
60 KB of addresses FFF0000H to FFFEFFFH are reserved as the internal RAM area.

<R> **(a) Internal RAM (8 KB)**

8 KB are allocated to addresses FFFD000H to FFFEFFFH of the following versions.
Accessing addresses FFF0000H to FFFCFFFH is prohibited.

- μPD703136A, 703136AY

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

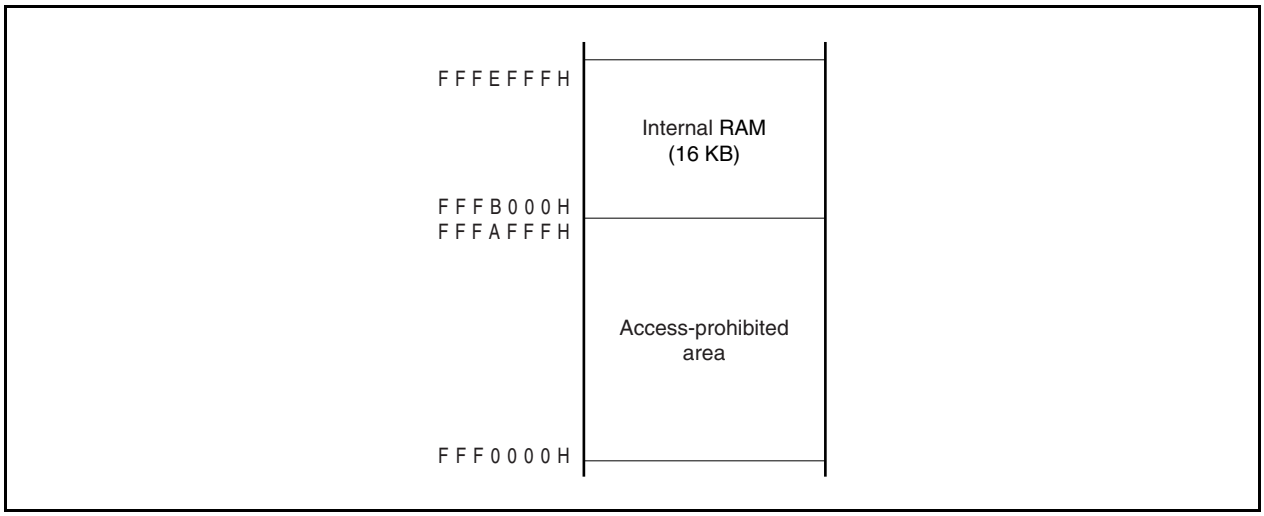


(b) Internal RAM (16 KB)

16 KB are allocated to addresses FFFB000H to FFFEFFFH of the following versions.
Accessing addresses FFF0000H to FFFAFFFH is prohibited.

<R> - μPD703131A, 703131AY, 703133A, 703133AY

Figure 3-8. Internal RAM Area (16 KB)

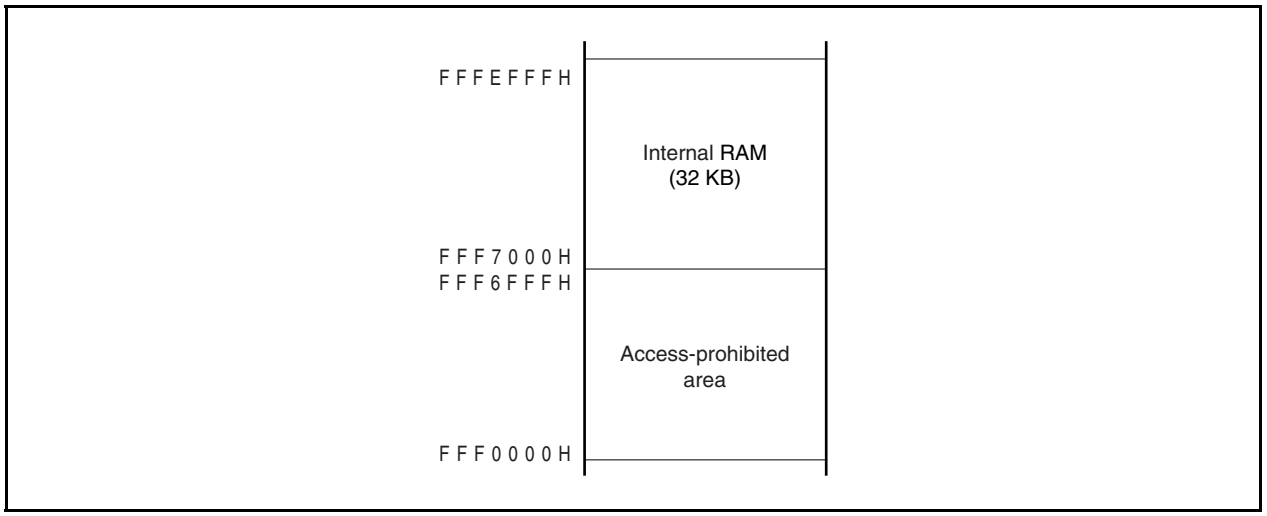


(c) Internal RAM (32 KB)

32 KB are allocated to addresses FFF7000H to FFFEFFFH of the following versions.
Accessing addresses FFF0000H to FFF6FFFH is prohibited.

<R> • μPD703132A, 703132AY, 703134A, 703134AY, 70F3134A, 70F3134AY

Figure 3-9. Internal RAM Area (32 KB)

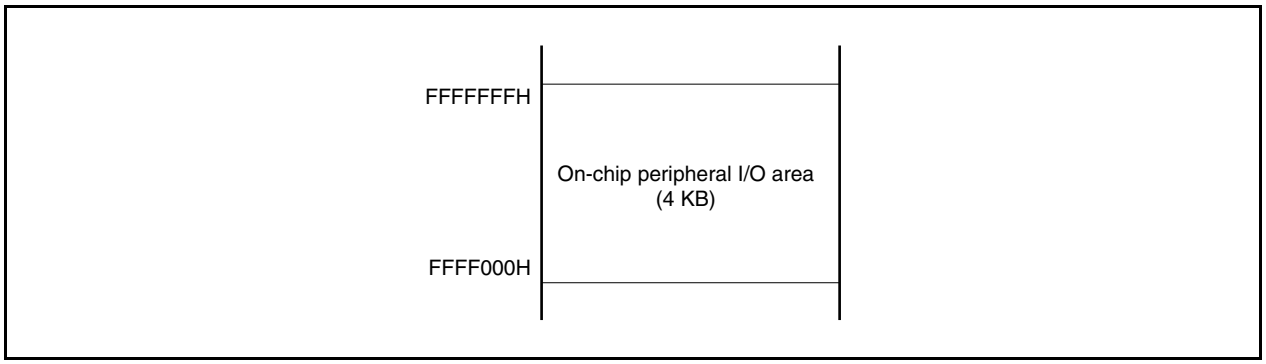


(3) On-chip peripheral I/O area

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

An image of addresses FFFF000H to FFFFFFFH can be seen at addresses 3FFF000H to 3FFFFFFH[Note].

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



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

Cautions 1. In the V850E/MA3, if a register is word accessed, halfword access is performed twice in the order of lower address, then higher address of the word area, disregarding the lower 2 bits of the address.

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

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

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

(4) External memory area

256 MB are available for external memory area. The lower 64 MB can be used as program/data area and the higher 192 MB as data area.

Access to the external memory area uses the chip select signal assigned to each memory block (access is carried out in the CS unit set by the CSC0 and CSC1 registers).

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

3.4.6 External memory expansion

By setting the PMCn register to the alternate function, an external memory device can be connected to the external memory space using each pin of ports AL, AH, DL, CS, CT, CM, and CD. Each register is set by selecting the alternate function for each pin of these ports using the PMCn register (n = AL, AH, DL, CS, CT, CM, CD).

After reset, since the internal ROM area is accessed, each pin of ports AL, AH, DL, CS, CT, CM, and CD enters the port mode and external devices cannot be used.

To use external memory, set the PMCn register.

3.4.7 Recommended use of address space

The architecture of the V850E/MA3 requires that a register that serves as a pointer be secured for address generation in operand data accessing of data space. Operand data access from an instruction can be directly executed at the address in this pointer register ±32 KB. However, because the general-purpose registers that can be used as a pointer register are limited, by minimizing the deterioration of address calculation performance when changing the pointer value, the number of usable general-purpose registers for handling variables is maximized, and the program size can be saved.

(1) Program space

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

(2) Data space

With the V850E/MA3, a 256 MB physical address space is seen as 16 images in the 4 GB CPU address space. The highest bit (bit 25) of this 26-bit address is assigned as an address sign-extended to 32 bits.

(a) Application of wraparound

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

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

Example For μPD703132A

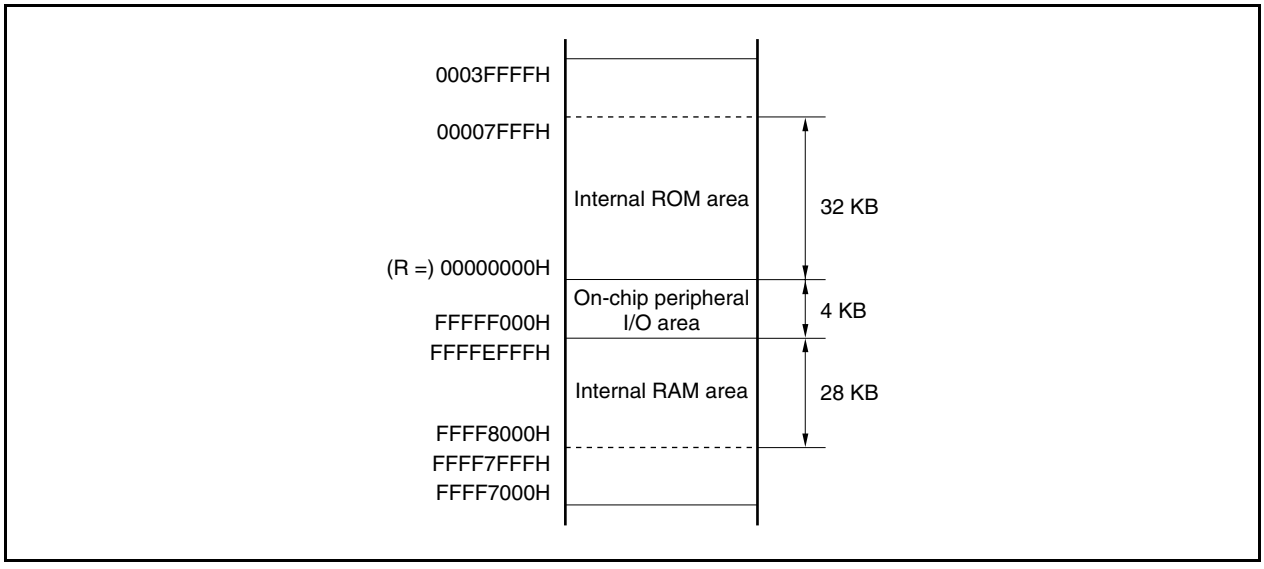
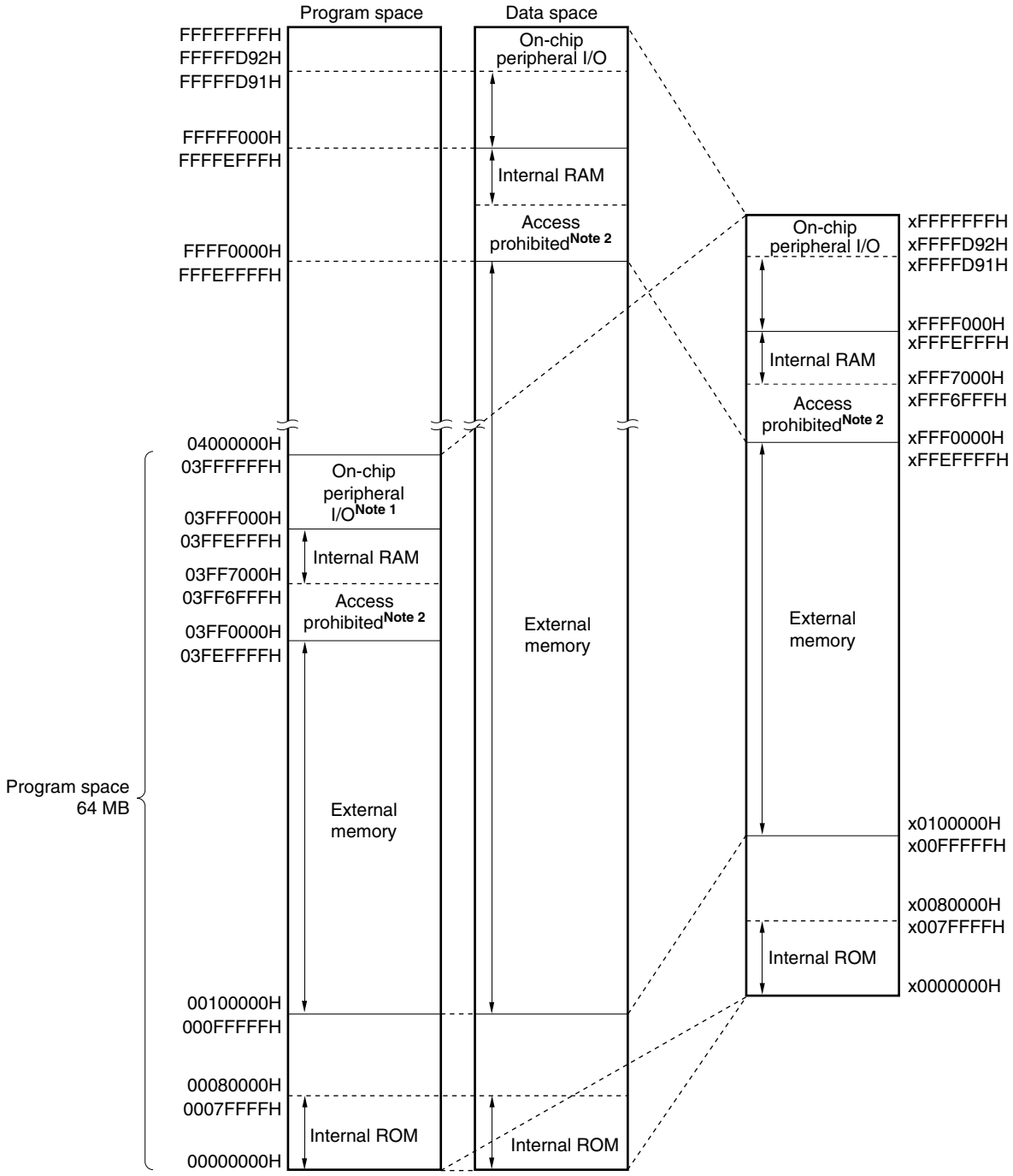


Figure 3-10. Recommended Memory Map



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

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

Remarks 1. The arrows indicate the recommended area.

2. This is a recommended memory map when the μPD703134A is set to the single-chip mode, and used in external expansion mode.

3.4.8 On-chip peripheral I/O registers

(1/11)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF000H	Port AL register	PAL	R/W			√	Undefined
FFFFF000H	Port ALL register	PALL	R/W	√	√		Undefined
FFFFF001H	Port ALH register	PALH	R/W	√	√		Undefined
FFFFF002H	Port AH register	PAH	R/W			√	Undefined
FFFFF002H	Port AHL register	PAHL	R/W	√	√		Undefined
FFFFF003H	Port AHH register	PAHH	R/W	√	√		Undefined
FFFFF004H	Port DL register	PDL	R/W			√	Undefined
FFFFF004H	Port DLL register	PDLL	R/W	√	√		Undefined
FFFFF005H	Port DLH register	PDLH	R/W	√	√		Undefined
FFFFF008H	Port CS register	PCS	R/W	√	√		Undefined
FFFFF00AH	Port CT register	PCT	R/W	√	√		Undefined
FFFFF00CH	Port CM register	PCM	R/W	√	√		Undefined
FFFFF00EH	Port CD register	PCD	R/W	√	√		Undefined
FFFFF012H	Port BD register	PBD	R/W	√	√		Undefined
FFFFF020H	Port AL mode register	PMAL	R/W			√	FFFFH
FFFFF020H	Port AL mode register L	PMALL	R/W	√	√		FFH
FFFFF021H	Port AL mode register H	PMALH	R/W	√	√		FFH
FFFFF022H	Port AH mode register	PMAH	R/W			√	FFFFH
FFFFF022H	Port AH mode register L	PMAHL	R/W	√	√		FFH
FFFFF023H	Port AH mode register H	PMAHH	R/W	√	√		FFH
FFFFF024H	Port DL mode register	PMDL	R/W			√	FFFFH
FFFFF024H	Port DL mode register L	PMDLL	R/W	√	√		FFH
FFFFF025H	Port DL mode register H	PMDLH	R/W	√	√		FFH
FFFFF028H	Port CS mode register	PMCS	R/W	√	√		FFH
FFFFF02AH	Port CT mode register	PMCT	R/W	√	√		FFH
FFFFF02CH	Port CM mode register	PMCM	R/W	√	√		FFH
FFFFF02EH	Port CD mode register	PMCD	R/W	√	√		FFH
FFFFF032H	Port BD mode register	PMBD	R/W	√	√		FFH
FFFFF040H	Port AL mode control register	PMCAL	R/W			√	0000H
FFFFF040H	Port AL mode control register L	PMCALL	R/W	√	√		00H
FFFFF041H	Port AL mode control register H	PMCALH	R/W	√	√		00H
FFFFF042H	Port AH mode control register	PMCAH	R/W			√	0000H
FFFFF042H	Port AH mode control register L	PMCAHL	R/W	√	√		00H
FFFFF043H	Port AH mode control register H	PMCAHH	R/W	√	√		00H
FFFFF044H	Port DL mode control register	PMCDL	R/W			√	0000H
FFFFF044H	Port DL mode control register L	PMCDLL	R/W	√	√		00H
FFFFF045H	Port DL mode control register H	PMCDLH	R/W	√	√		00H

(2/11)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF048H	Port CS mode control register	PMCCS	R/W	√	√		00H
FFFFF049H	Port CS function control register	PFCCS	R/W	√	√		00H
FFFFF04AH	Port CT mode control register	PMCCT	R/W	√	√		00H
FFFFF04BH	Port CT function control register	PFCCT	R/W	√	√		00H
FFFFF04CH	Port CM mode control register	PMCCM	R/W	√	√		00H
FFFFF04EH	Port CD mode control register	PMCCD	R/W	√	√		00H
FFFFF052H	Port BD mode control register	PMCBD	R/W	√	√		00H
FFFFF060H	Chip area select control register 0	CSC0	R/W			√	2C11H
FFFFF062H	Chip area select control register 1	CSC1	R/W			√	2C11H
FFFFF068H	Endian configuration register	BEC	R/W			√	0000H
FFFFF06EH	System wait control register	VSWC	R/W		√		77H
FFFFF080H	DMA source address register 0L	DSA0L	R/W			√	Undefined
FFFFF082H	DMA source address register 0H	DSA0H	R/W			√	Undefined
FFFFF084H	DMA destination address register 0L	DDA0L	R/W			√	Undefined
FFFFF086H	DMA destination address register 0H	DDA0H	R/W			√	Undefined
FFFFF088H	DMA source address register 1L	DSA1L	R/W			√	Undefined
FFFFF08AH	DMA source address register 1H	DSA1H	R/W			√	Undefined
FFFFF08CH	DMA destination address register 1L	DDA1L	R/W			√	Undefined
FFFFF08EH	DMA destination address register 1H	DDA1H	R/W			√	Undefined
FFFFF090H	DMA source address register 2L	DSA2L	R/W			√	Undefined
FFFFF092H	DMA source address register 2H	DSA2H	R/W			√	Undefined
FFFFF094H	DMA destination address register 2L	DDA2L	R/W			√	Undefined
FFFFF096H	DMA destination address register 2H	DDA2H	R/W			√	Undefined
FFFFF098H	DMA source address register 3L	DSA3L	R/W			√	Undefined
FFFFF09AH	DMA source address register 3H	DSA3H	R/W			√	Undefined
FFFFF09CH	DMA destination address register 3L	DDA3L	R/W			√	Undefined
FFFFF09EH	DMA destination address register 3H	DDA3H	R/W			√	Undefined
FFFFF0C0H	DMA transfer count register 0	DBC0	R/W			√	Undefined
FFFFF0C2H	DMA transfer count register 1	DBC1	R/W			√	Undefined
FFFFF0C4H	DMA transfer count register 2	DBC2	R/W			√	Undefined
FFFFF0C6H	DMA transfer count register 3	DBC3	R/W			√	Undefined
FFFFF0D0H	DMA addressing control register 0	DADC0	R/W			√	0000H
FFFFF0D2H	DMA addressing control register 1	DADC1	R/W			√	0000H
FFFFF0D4H	DMA addressing control register 2	DADC2	R/W			√	0000H
FFFFF0D6H	DMA addressing control register 3	DADC3	R/W			√	0000H
FFFFF0E0H	DMA channel control register 0	DCHC0	R/W	√	√		00H
FFFFF0E2H	DMA channel control register 1	DCHC1	R/W	√	√		00H
FFFFF0E4H	DMA channel control register 2	DCHC2	R/W	√	√		00H
FFFFF0E6H	DMA channel control register 3	DCHC3	R/W	√	√		00H

(3/11)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF100H	Interrupt mask register 0	IMR0	R/W			√	FFFFH
FFFFF100H	Interrupt mask register 0L	IMR0L	R/W	√	√		FFH
FFFFF101H	Interrupt mask register 0H	IMR0H	R/W	√	√		FFH
FFFFF102H	Interrupt mask register 1	IMR1	R/W			√	FFFFH
FFFFF102H	Interrupt mask register 1L	IMR1L	R/W	√	√		FFH
FFFFF103H	Interrupt mask register 1H	IMR1H	R/W	√	√		FFH
FFFFF104H	Interrupt mask register 2	IMR2	R/W			√	FFFFH
FFFFF104H	Interrupt mask register 2L	IMR2L	R/W	√	√		FFH
FFFFF105H	Interrupt mask register 2H	IMR2H	R/W	√	√		FFH
FFFFF106H	Interrupt mask register 3	IMR3	R/W			√	FFFFH
FFFFF106H	Interrupt mask register 3L	IMR3L	R/W	√	√		FFH
FFFFF107H	Interrupt mask register 3H	IMR3H	R/W	√	√		FFH
FFFFF110H	Interrupt control register 0	WDTIC	R/W	√	√		47H
FFFFF112H	Interrupt control register 1	P00IC0	R/W	√	√		47H
FFFFF114H	Interrupt control register 2	P00IC1	R/W	√	√		47H
FFFFF116H	Interrupt control register 3	P00IC4	R/W	√	√		47H
FFFFF118H	Interrupt control register 4	P00IC5	R/W	√	√		47H
FFFFF11AH	Interrupt control register 5	P10IC6	R/W	√	√		47H
FFFFF11CH	Interrupt control register 6	P10IC7	R/W	√	√		47H
FFFFF11EH	Interrupt control register 7	P01IC0	R/W	√	√		47H
FFFFF120H	Interrupt control register 8	P01IC1	R/W	√	√		47H
FFFFF122H	Interrupt control register 9	P01IC2	R/W	√	√		47H
FFFFF124H	Interrupt control register 10	P01IC3	R/W	√	√		47H
FFFFF126H	Interrupt control register 11	P11IC4	R/W	√	√		47H
FFFFF128H	Interrupt control register 12	P11IC5	R/W	√	√		47H
FFFFF12AH	Interrupt control register 13	P02IC1	R/W	√	√		47H
FFFFF12CH	Interrupt control register 14	P02IC2	R/W	√	√		47H
FFFFF12EH	Interrupt control register 15	P12IC4	R/W	√	√		47H
FFFFF130H	Interrupt control register 16	P12IC5	R/W	√	√		47H
FFFFF132H	Interrupt control register 17	P12IC6	R/W	√	√		47H
FFFFF134H	Interrupt control register 18	P13IC0	R/W	√	√		47H
FFFFF136H	Interrupt control register 19	P13IC1	R/W	√	√		47H
FFFFF138H	Interrupt control register 20	P13IC2	R/W	√	√		47H
FFFFF13AH	Interrupt control register 21	P13IC3	R/W	√	√		47H
FFFFF13CH	Interrupt control register 22	P13IC4	R/W	√	√		47H
FFFFF13EH	Interrupt control register 23	P13IC7	R/W	√	√		47H
FFFFF140H	Interrupt control register 24	P05IC0	R/W	√	√		47H

(4/11)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF142H	Interrupt control register 25	P05IC1	R/W	√	√		47H
FFFFF144H	Interrupt control register 26	CMICD0	R/W	√	√		47H
FFFFF146H	Interrupt control register 27	CMICD1	R/W	√	√		47H
FFFFF148H	Interrupt control register 28	CMICD2	R/W	√	√		47H
FFFFF14AH	Interrupt control register 29	CMICD3	R/W	√	√		47H
FFFFF14CH	Interrupt control register 30	CM10IC0	R/W	√	√		47H
FFFFF14EH	Interrupt control register 31	CM10IC1	R/W	√	√		47H
FFFFF150H	Interrupt control register 32	OVPIC0	R/W	√	√		47H
FFFFF152H	Interrupt control register 33	OVQIC	R/W	√	√		47H
FFFFF154H	Interrupt control register 34	OVPIC1	R/W	√	√		47H
FFFFF156H	Interrupt control register 35	OVPIC2	R/W	√	√		47H
FFFFF158H	Interrupt control register 36	DMAIC0	R/W	√	√		47H
FFFFF15AH	Interrupt control register 37	DMAIC1	R/W	√	√		47H
FFFFF15CH	Interrupt control register 38	DMAIC2	R/W	√	√		47H
FFFFF15EH	Interrupt control register 39	DMAIC3	R/W	√	√		47H
FFFFF160H	Interrupt control register 40	SEIC0	R/W	√	√		47H
FFFFF162H	Interrupt control register 41	SRIC0	R/W	√	√		47H
FFFFF164H	Interrupt control register 42	STIC0	R/W	√	√		47H
FFFFF166H	Interrupt control register 43	SEIC1	R/W	√	√		47H
FFFFF168H	Interrupt control register 44	SRIC1	R/W	√	√		47H
FFFFF16AH	Interrupt control register 45	STIC1	R/W	√	√		47H
FFFFF16CH	Interrupt control register 46	SEIC2	R/W	√	√		47H
FFFFF16EH	Interrupt control register 47	SRIC2	R/W	√	√		47H
FFFFF170H	Interrupt control register 48	STIC2	R/W	√	√		47H
FFFFF172H	Interrupt control register 49	SEIC3	R/W	√	√		47H
FFFFF174H	Interrupt control register 50	SRIC3	R/W	√	√		47H
FFFFF176H	Interrupt control register 51	STIC3	R/W	√	√		47H
FFFFF178H	Interrupt control register 52	ADIC	R/W	√	√		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 mode register 0	ADM0	R/W	√	√		00H
FFFFF201H	A/D converter mode register 1	ADM1	R/W		√		07H
FFFFF202H	A/D converter mode register 2	ADM2	R/W	√	√		02H
FFFFF210H	A/D conversion result register 0 (10BIT)	ADCR0	R			√	0000H
FFFFF212H	A/D conversion result register 1 (10BIT)	ADCR1	R			√	0000H
FFFFF214H	A/D conversion result register 2 (10BIT)	ADCR2	R			√	0000H
FFFFF216H	A/D conversion result register 3 (10BIT)	ADCR3	R			√	0000H
FFFFF218H	A/D conversion result register 4 (10BIT)	ADCR4	R			√	0000H

(5/11)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF21AH	A/D conversion result register 5 (10BIT)	ADCR5	R			√	0000H
FFFFF21CH	A/D conversion result register 6 (10BIT)	ADCR6	R			√	0000H
FFFFF21EH	A/D conversion result register 7 (10BIT)	ADCR7	R			√	0000H
FFFFF220H	A/D conversion result register 0H (8BIT)	ADCR0H	R		√		00H
FFFFF221H	A/D conversion result register 1H (8BIT)	ADCR1H	R		√		00H
FFFFF222H	A/D conversion result register 2H (8BIT)	ADCR2H	R		√		00H
FFFFF223H	A/D conversion result register 3H (8BIT)	ADCR3H	R		√		00H
FFFFF224H	A/D conversion result register 4H (8BIT)	ADCR4H	R		√		00H
FFFFF225H	A/D conversion result register 5H (8BIT)	ADCR5H	R		√		00H
FFFFF226H	A/D conversion result register 6H (8BIT)	ADCR6H	R		√		00H
FFFFF227H	A/D conversion result register 7H (8BIT)	ADCR7H	R		√		00H
FFFFF288H	A/D trigger select register	ADTS	R/W	√	√		01H
FFFFF2C0H	D/A conversion value setting register 0	DA0CS0	R/W		√		00H
FFFFF2C1H	D/A conversion value setting register 1	DA0CS1	R/W		√		00H
FFFFF2C2H	D/A converter mode register	DA0M	R/W	√	√		00H
FFFFF400H	Port 0 register	P0	R/W	√	√		Undefined
FFFFF402H	Port 1 register	P1	R/W	√	√		Undefined
FFFFF404H	Port 2 register	P2	R/W	√	√		Undefined
FFFFF406H	Port 3 register	P3	R/W	√	√		Undefined
FFFFF408H	Port 4 register	P4	R/W	√	√		Undefined
FFFFF40AH	Port 5 register	P5	R/W	√	√		Undefined
FFFFF40EH	Port 7 register	P7	R	√	√		Undefined
FFFFF410H	Port 8 register	P8	R	√	√		Undefined
FFFFF420H	Port 0 mode register	PM0	R/W	√	√		FFH
FFFFF422H	Port 1 mode register	PM1	R/W	√	√		FFH
FFFFF424H	Port 2 mode register	PM2	R/W	√	√		FFH
FFFFF426H	Port 3 mode register	PM3	R/W	√	√		FFH
FFFFF428H	Port 4 mode register	PM4	R/W	√	√		FFH
FFFFF42AH	Port 5 mode register	PM5	R/W	√	√		FFH
FFFFF440H	Port 0 mode control register	PMC0	R/W	√	√		00H
FFFFF442H	Port 1 mode control register	PMC1	R/W	√	√		00H
FFFFF444H	Port 2 mode control register	PMC2	R/W	√	√		01H
FFFFF446H	Port 3 mode control register	PMC3	R/W	√	√		00H
FFFFF448H	Port 4 mode control register	PMC4	R/W	√	√		00H
FFFFF44AH	Port 5 mode control register	PMC5	R/W	√	√		00H
FFFFF44EH	Port 7 mode control register	PMC7	R/W	√	√		00H
FFFFF460H	Port 0 function control register	PFC0	R/W	√	√		00H
FFFFF462H	Port 1 function control register	PFC1	R/W	√	√		00H

(6/11)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF464H	Port 2 function control register	PFC2	R/W	√	√		00H
FFFFF466H	Port 3 function control register	PFC3	R/W	√	√		00H
FFFFF468H	Port 4 function control register	PFC4	R/W	√	√		00H
FFFFF46AH	Port 5 function control register	PFC5	R/W	√	√		00H
FFFFF480H	Bus cycle type configuration register 0	BCT0	R/W			√	8888H
FFFFF482H	Bus cycle type configuration register 1	BCT1	R/W			√	8888H
FFFFF484H	Data wait control register 0	DWC0	R/W			√	7777H
FFFFF486H	Data wait control register 1	DWC1	R/W			√	7777H
FFFFF488H	Bus cycle control register	BCC	R/W			√	FFFFH
FFFFF48AH	Address setup wait control register	ASC	R/W			√	FFFFH
FFFFF48CH	Bus cycle period control register	BCP	R/W		√		00H
FFFFF48EH	Local bus sizing control register	LBS	R/W			√	5555H
FFFFF494H	DMA flyby transfer wait control register	FWC	R/W			√	7777H
FFFFF496H	DMA flyby transfer idle control register	FIC	R/W			√	3333H
FFFFF498H	Bus mode control register	BMC	R/W	√	√		01H
FFFFF49AH	Page ROM configuration register	PRC	R/W			√	7000H
FFFFF49CH	Write access synchronization control register	WAS	W		√		00H
FFFFF49EH	Address hold wait control register	AHC	R/W			√	FFFFH
FFFFF4A4H	SDRAM configuration register 1	SCR1	R/W			√	30C0H
FFFFF4A6H	SDRAM refresh control register 1	RFS1	R/W			√	0000H
FFFFF4ACH	SDRAM configuration register 3	SCR3	R/W			√	30C0H
FFFFF4AEH	SDRAM refresh control register 3	RFS3	R/W			√	0000H
FFFFF4B0H	SDRAM configuration register 4	SCR4	R/W			√	30C0H
FFFFF4B2H	SDRAM refresh control register 4	RFS4	R/W			√	0000H
FFFFF4B8H	SDRAM configuration register 6	SCR6	R/W			√	30C0H
FFFFF4BAH	SDRAM refresh control register 6	RFS6	R/W			√	0000H
FFFFF540H	Timer D0	TMD0	R			√	0000H
FFFFF542H	Compare register D0	CMD0	R/W			√	0000H
FFFFF544H	Timer mode control register D0	TMCD0	R/W	√	√		00H
FFFFF550H	Timer D1	TMD1	R			√	0000H
FFFFF552H	Compare register D1	CMD1	R/W			√	0000H
FFFFF554H	Timer mode control register D1	TMCD1	R/W	√	√		00H
FFFFF560H	Timer D2	TMD2	R			√	0000H
FFFFF562H	Compare register D2	CMD2	R/W			√	0000H
FFFFF564H	Timer mode control register D2	TMCD2	R/W	√	√		00H
FFFFF570H	Timer D3	TMD3	R			√	0000H
FFFFF572H	Compare register D3	CMD3	R/W			√	0000H

(7/11)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF574H	Timer mode control register D3	TMCD3	R/W	√	√		00H
FFFFF5C0H	Timer ENC10	TMENC10	R/W			√	0000H
FFFFF5C2H	Compare register 100	CM100	R/W			√	0000H
FFFFF5C4H	Compare register 101	CM101	R/W			√	0000H
FFFFF5C6H	Capture/compare register 100	CC100	R/W			√	0000H
FFFFF5C8H	Capture/compare register 101	CC101	R/W			√	0000H
FFFFF5CAH	Capture/compare control register 10	CCR10	R/W		√		00H
FFFFF5CBH	Timer unit mode register 10	TUM10	R/W		√		00H
FFFFF5CCH	Timer control register 10	TMC10	R/W	√	√		00H
FFFFF5CDH	Valid edge select register 10	SESA10	R/W		√		00H
FFFFF5CEH	Prescaler mode register 10	PRM10	R/W		√		07H
FFFFF5CFH	Status register 10	STATUS10	R	√	√		00H
FFFFF600H	TMQ0 control register 0	TQ0CTL0	R/W	√	√		00H
FFFFF601H	TMQ0 control register 1	TQ0CTL1	R/W	√	√		00H
FFFFF602H	TMQ0 I/O control register 0	TQ0IOC0	R/W	√	√		00H
FFFFF603H	TMQ0 I/O control register 1	TQ0IOC1	R/W	√	√		00H
FFFFF604H	TMQ0 I/O control register 2	TQ0IOC2	R/W	√	√		00H
FFFFF605H	TMQ0 option register 0	TQ0OPT0	R/W	√	√		00H
FFFFF606H	TMQ0 capture/compare register 0	TQ0CCR0	R/W			√	0000H
FFFFF608H	TMQ0 capture/compare register 1	TQ0CCR1	R/W			√	0000H
FFFFF60AH	TMQ0 capture/compare register 2	TQ0CCR2	R/W			√	0000H
FFFFF60CH	TMQ0 capture/compare register 3	TQ0CCR3	R/W			√	0000H
FFFFF60EH	TMQ0 counter read buffer register	TQ0CNT	R			√	0000H
FFFFF620H	TMQ0 option register 1	TQ0OPT1	R/W	√	√		00H
FFFFF621H	TMQ0 option register 2	TQ0OPT2	R/W	√	√		00H
FFFFF622H	TMQ0 I/O control register 3	TQ0IOC3	R/W	√	√		A8H
FFFFF624H	TMQ0 dead time compare register	TQ0DTC	R/W			√	0000H
FFFFF630H	High-impedance output control register 0	HZA0CTL0	R/W	√	√		00H
FFFFF631H	High-impedance output control register 1	HZA0CTL1	R/W	√	√		00H
FFFFF640H	TMP0 control register 0	TP0CTL0	R/W	√	√		00H
FFFFF641H	TMP0 control register 1	TP0CTL1	R/W	√	√		00H
FFFFF642H	TMP0 I/O control register 0	TP0IOC0	R/W	√	√		00H
FFFFF643H	TMP0 I/O control register 1	TP0IOC1	R/W	√	√		00H
FFFFF644H	TMP0 I/O control register 2	TP0IOC2	R/W	√	√		00H
FFFFF645H	TMP0 option register 0	TP0OPT0	R/W	√	√		00H
FFFFF646H	TMP0 capture/compare register 0	TP0CCR0	R/W			√	0000H
FFFFF648H	TMP0 capture/compare register 1	TP0CCR1	R/W			√	0000H
FFFFF64AH	TMP0 counter read buffer register	TP0CNT	R			√	0000H

(8/11)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1	8	16	
FFFFF660H	TMP1 control register 0	TP1CTL0	R/W	√	√		00H
FFFFF661H	TMP1 control register 1	TP1CTL1	R/W	√	√		00H
FFFFF662H	TMP1 I/O control register 0	TP1IOC0	R/W	√	√		00H
FFFFF663H	TMP1 I/O control register 1	TP1IOC1	R/W	√	√		00H
FFFFF664H	TMP1 I/O control register 2	TP1IOC2	R/W	√	√		00H
FFFFF665H	TMP1 option register 0	TP1OPT0	R/W	√	√		00H
FFFFF666H	TMP1 capture/compare register 0	TP1CCR0	R/W			√	0000H
FFFFF668H	TMP1 capture/compare register 1	TP1CCR1	R/W			√	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	R/W	√	√		00H
FFFFF682H	TMP2 I/O control register 0	TP2IOC0	R/W	√	√		00H
FFFFF683H	TMP2 I/O control register 1	TP2IOC1	R/W	√	√		00H
FFFFF684H	TMP2 I/O control register 2	TP2IOC2	R/W	√	√		00H
FFFFF685H	TMP2 option register 0	TP2OPT0	R/W	√	√		00H
FFFFF686H	TMP2 capture/compare register 0	TP2CCR0	R/W			√	0000H
FFFFF688H	TMP2 capture/compare register 1	TP2CCR1	R/W			√	0000H
FFFFF68AH	TMP2 counter read buffer register	TP2CNT	R			√	0000H
FFFFF6C0H	Oscillation stabilization time select register	OSTS	R/W		√		01H
FFFFF6C1H	Watchdog timer clock select register	WDCS	R/W		√		00H
FFFFF6C2H	Watchdog timer mode register	WDTM	R/W	√	√		00H
FFFFF700H	Port 0 function control expansion register	PFCE0	R/W	√	√		00H
FFFFF702H	Port 1 function control expansion register	PFCE1	R/W	√	√		00H
FFFFF704H	Port 2 function control expansion register	PFCE2	R/W	√	√		00H
FFFFF706H	Port 3 function control expansion register	PFCE3	R/W	√	√		00H
FFFFF70AH	Port 5 function control expansion register	PFCE5	R/W	√	√		00H
FFFFF802H	System status register	SYS	R/W	√	√		00H
<R> FFFFF804H	Chip select signal delay control register	CSDC	R/W	√	√		00H
FFFFF810H	DMA trigger factor register 0	DTFR0	R/W	√Note 1	√		00H
FFFFF812H	DMA trigger factor register 1	DTFR1	R/W	√Note 1	√		00H
FFFFF814H	DMA trigger factor register 2	DTFR2	R/W	√Note 1	√		00H
FFFFF816H	DMA trigger factor register 3	DTFR3	R/W	√Note 1	√		00H
FFFFF820H	Power save mode register	PSMR	R/W	√	√		00H
FFFFF822H	Clock control register	CKC	R/W	√	√		00H
FFFFF828H	Processor clock control register	PCC	R/W	√	√		00H
FFFFF82AH	Watchdog timer reset status register	WDRES	R/W	√Note 2	√		00H

Notes 1. Only the seventh bit can be manipulated in bit units.

2. This register can be manipulated in bit units only when it is read.

(9/11)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1	8	16	32	
FFFFF840H	Correction address register 0	CORAD0	R/W				√	00000000H
FFFFF840H	Correction address register 0L	CORAD0L	R/W			√		0000H
FFFFF842H	Correction address register 0H	CORAD0H	R/W			√		0000H
FFFFF844H	Correction address register 1	CORAD1	R/W				√	00000000H
FFFFF844H	Correction address register 1L	CORAD1L	R/W			√		0000H
FFFFF846H	Correction address register 1H	CORAD1H	R/W			√		0000H
FFFFF848H	Correction address register 2	CORAD2	R/W				√	00000000H
FFFFF848H	Correction address register 2L	CORAD2L	R/W			√		0000H
FFFFF84AH	Correction address register 2H	CORAD2H	R/W			√		0000H
FFFFF84CH	Correction address register 3	CORAD3	R/W				√	00000000H
FFFFF84CH	Correction address register 3L	CORAD3L	R/W			√		0000H
FFFFF84EH	Correction address register 3H	CORAD3H	R/W			√		0000H
FFFFF880H	Correction control register	CORCN	R/W	√	√			00H
FFFFF8A0H	DMA terminal count output control register	DTOC	R/W		√			01H
FFFFF8A8H	DMA interface control register	DIFC	R/W		√			00H
FFFFF8ACH	DMAAK width control register	DAKW	R/W		√			00H
FFFFFA00H	UARTA0 control register 0	UA0CTL0	R/W	√	√			10H
FFFFFA01H	UARTA0 control register 1	UA0CTL1	R/W		√			00H
FFFFFA02H	UARTA0 control register 2	UA0CTL2	R/W		√			FFH
FFFFFA03H	UARTA0 option control register 0	UA0OPT0	R/W	√	√			14H
FFFFFA04H	UARTA0 status register	UA0STR	R/W	√	√			00H
FFFFFA06H	UARTA0 receive data register	UA0RX	R		√			FFH
FFFFFA07H	UARTA0 transmit data register	UA0TX	R/W		√			FFH
FFFFFA10H	UARTA1 control register 0	UA1CTL0	R/W	√	√			10H
FFFFFA11H	UARTA1 control register 1	UA1CTL1	R/W		√			00H
FFFFFA12H	UARTA1 control register 2	UA1CTL2	R/W		√			FFH
FFFFFA13H	UARTA1 option control register 0	UA1OPT0	R/W	√	√			14H
FFFFFA14H	UARTA1 status register	UA1STR	R/W	√	√			00H
FFFFFA16H	UARTA1 receive data register	UA1RX	R		√			FFH
FFFFFA17H	UARTA1 transmit data register	UA1TX	R/W		√			FFH
FFFFFA20H	UARTA2 control register 0	UA2CTL0	R/W	√	√			10H
FFFFFA21H	UARTA2 control register 1	UA2CTL1	R/W		√			00H
FFFFFA22H	UARTA2 control register 2	UA2CTL2	R/W		√			FFH
FFFFFA23H	UARTA2 option control register 0	UA2OPT0	R/W	√	√			14H
FFFFFA24H	UARTA2 status register	UA2STR	R/W	√	√			00H
FFFFFA26H	UARTA2 receive data register	UA2RX	R		√			FFH
FFFFFA27H	UARTA2 transmit data register	UA2TX	R/W		√			FFH
FFFFFA30H	UARTA3 control register 0	UA3CTL0	R/W	√	√			10H

(10/11)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation				After Reset
				1	8	16	32	
FFFFFA31H	UARTA3 control register 1	UA3CTL1	R/W		√			00H
FFFFFA32H	UARTA3 control register 2	UA3CTL2	R/W		√			FFH
FFFFFA33H	UARTA3 option control register 0	UA3OPT0	R/W	√	√			14H
FFFFFA34H	UARTA3 status register	UA3STR	R/W	√	√			00H
FFFFFA36H	UARTA3 receive data register	UA3RX	R		√			FFH
FFFFFA37H	UARTA3 transmit data register	UA3TX	R/W		√			FFH
FFFFFC00H	External interrupt falling edge specification register 0	INTF0	R/W	√	√			F3H
FFFFFC02H	External interrupt falling edge specification register 1	INTF1	R/W	√	√			3FH
FFFFFC04H	External interrupt falling edge specification register 2	INTF2	R/W	√	√			76H
FFFFFC06H	External interrupt falling edge specification register 3	INTF3	R/W	√	√			9FH
FFFFFC0AH	External interrupt falling edge specification register 5	INTF5	R/W	√	√			03H
FFFFFC1EH	NMI falling edge specification register	NMIF	R/W	√	√			00H
FFFFFC20H	External interrupt rising edge specification register 0	INTR0	R/W	√	√			00H
FFFFFC22H	External interrupt rising edge specification register 1	INTR1	R/W	√	√			00H
FFFFFC24H	External interrupt rising edge specification register 2	INTR2	R/W	√	√			00H
FFFFFC26H	External interrupt rising edge specification register 3	INTR3	R/W	√	√			00H
FFFFFC2AH	External interrupt rising edge specification register 5	INTR5	R/W	√	√			00H
FFFFFC3EH	NMI rising edge specification register	NMIR	R/W	√	√			00H
FFFFFD00H	CSIB0 control register 0	CB0CTL0	R/W	√	√			01H
FFFFFD01H	CSIB0 control register 1	CB0CTL1	R/W	√	√			00H
FFFFFD02H	CSIB0 control register 2	CB0CTL2	R/W		√			00H
FFFFFD03H	CSIB0 status register	CB0STR	R/W	√	√			00H
FFFFFD04H	CSIB0 receive data register (16 bits)	CB0RX	R			√		0000H
FFFFFD04H	CSIB0 receive data register L (8 bits)	CB0RXL	R		√			00H
FFFFFD06H	CSIB0 transmit data register (16 bits)	CB0TX	R/W			√		0000H
FFFFFD06H	CSIB0 transmit data register L (8 bits)	CB0TXL	R/W		√			00H
FFFFFD10H	CSIB1 control register 0	CB1CTL0	R/W	√	√			01H
FFFFFD11H	CSIB1 control register 1	CB1CTL1	R/W	√	√			00H
FFFFFD12H	CSIB1 control register 2	CB1CTL2	R/W		√			00H
FFFFFD13H	CSIB1 status register	CB1STR	R/W	√	√			00H
FFFFFD14H	CSIB1 receive data register (16 bits)	CB1RX	R			√		0000H
FFFFFD14H	CSIB1 receive data register L (8 bits)	CB1RXL	R		√			00H
FFFFFD16H	CSIB1 transmit data register (16 bits)	CB1TX	R/W			√		0000H
FFFFFD16H	CSIB1 transmit data register L (8 bits)	CB1TXL	R/W		√			00H
FFFFFD20H	CSIB2 control register 0	CB2CTL0	R/W	√	√			01H
FFFFFD21H	CSIB2 control register 1	CB2CTL1	R/W	√	√			00H
FFFFFD22H	CSIB2 control register 2	CB2CTL2	R/W		√			00H
FFFFFD23H	CSIB2 status register	CB2STR	R/W	√	√			00H
FFFFFD24H	CSIB2 receive data register (16 bits)	CB2RX	R			√		0000H
FFFFFD24H	CSIB2 receive data register L (8 bits)	CB2RXL	R		√			00H
FFFFFD26H	CSIB2 transmit data register (16 bits)	CB2TX	R/W			√		0000H
FFFFFD26H	CSIB2 transmit data register L (8 bits)	CB2TXL	R/W		√			00H

(11/11)

	Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
					1	8	16	
	FFFFFD80H	IIC shift register[Note]	IIC	R/W		√		00H
	FFFFFD82H	IIC control register[Note]	IICC	R/W	√	√		00H
	FFFFFD83H	Slave address register[Note]	SVA	R/W		√		00H
	FFFFFD84H	IIC clock select register[Note]	IICCL	R/W	√	√		00H
	FFFFFD85H	IIC function expansion register[Note]	IICX	R/W	√	√		00H
	FFFFFD86H	IIC status register[Note]	IICS	R	√	√		00H
	FFFFFD8AH	IIC flag register[Note]	IICF	R/W	√	√		00H
<R>	FFFFFD90H	Prescaler mode register[Note]	PRSM	R/W	√	√		00H
<R>	FFFFFD91H	Prescaler compare register[Note]	PRSCM	R/W		√		00H

Note I^2C bus versions (Y products) only (see **Table 1-1)**

3.4.9 Special registers

Special registers are registers that are protected from being written with illegal data due to a program hang-up. The V850E/MA3 has the following five special registers.

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

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

(1) Setting data to special registers

Set data to the special registers in the following sequence.

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

[Example] When using CKC register (system clock setting)

```
<1>  MOV   0x03, r10
<2>  ST.B  r10, PRCMD[r0]      ; Write PRCMD register.
<3>  ST.B  r10, CKC[r0]        ; Set CKC register.
(next instruction)
```

No special sequence is necessary for reading a specific register.

Cautions 1. A store instruction for the PRCMD register does not acknowledge interrupts. This coding is made on assumption that <2> and <3> above are executed by the program with consecutive store instructions. If another instruction is set between <2> and <3>, the above sequence may become ineffective when the interrupt is acknowledged by that instruction, and a malfunction of the program may result.

<R> **2. Although the data written to the PRCMD register is dummy data, use the same register as the general-purpose register used in setting the special register (<3> in [Example]) by using the store instruction for writing to the PRCMD register (<2> in [Example]). The same method should be applied when using a general-purpose register for addressing.**
An example of setting the special register (<3> in Example) by using the bit manipulation instruction is shown below.

```
CLR1 0, WDRES[r0]
```

3. Before executing this processing, terminate all DMA transfer operations.
4. To set the IDLE mode or software STOP mode (PSC.STB bit = 1), see 21.7 Procedure for Setting or Restoring from IDLE and Software STOP Modes.

(2) Command register (PRCMD)

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

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

After reset: Undefined W Address: FFFFF1FCH

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

(3) System status register (SYS)

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

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

Reset input clears 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	Detects protection error
0	Protection error did not occur.
1	Protection error occurred.

The PRERR flag operates under the following conditions.

(a) Set condition (PRERR flag = 1)

(i) When data is written to a special register without writing anything to the PRCMD register (when <3> is executed without executing <2> in **3.4.9 (1) Setting data to special registers**)

(ii) 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 **3.4.9 (1) Setting data to special registers** is not the setting of a special register)

<R> (iii) When 1 is written to the PRERR flag

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

(b) Clear condition (PRERR flag = 0)

(i) When 0 is written to the PRERR flag

(ii) When the system is reset

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

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

<R> **3.4.10 System wait control register (VSWC)**

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

Access to on-chip peripheral I/O registers of the V850E1 CPU core is basically made in 3 clocks; however, in the V850E/MA3, waits may be required in addition to those 3 clocks. Set the values shown below to the VSWC register according to the operating frequency that is used.

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

Operating Frequency (f_{CPU})	Set Value of VSWC	Number of Waits for On-Chip Peripheral I/O Register Access
5 MHz ≤ f_{CPU} ≤ 33 MHz	11H	2
33 MHz < f_{CPU} ≤ 50 MHz	12H	3
50 MHz < f_{CPU} ≤ 80 MHz	24H	6

Remark When a register the includes status flags that indicate the statuses of the on-chip peripheral functions (register such as the STATUS10 register) or a register (TMENC10, etc.) 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.

3.4.11 Cautions

(1) Registers to be set first

When using the V850E/MA3, the following registers must be set in the beginning.

- System wait control register (VSWC)
  (See **3.4.10 System wait control register (VSWC)**)
- Clock control register (CKC)
  (See **7.3 (2) Clock control register (CKC)**)
- Program ID register (ASID)
  (See **3.2.2 (8) Program ID register (ASID)**.)

After setting VSWC, CKC, and ASID set other registers as necessary.

To use the external bus, initialize each register in the following sequence after setting the above registers.

<1> Set each pin to the alternate-function mode by setting each port-related register.

<2> Select a chip select space by using CSCn register (n = 0, 1).

<3> Specify the type of memory of each chip select space by using BCTn register.

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

(a) Description

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

Instruction <1>

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

Instruction <2>

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

<Example>

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

(b) Countermeasure

<1> When compiler (CA850) is used

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

<2> For assembler

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

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

CHAPTER 4 PORT FUNCTIONS

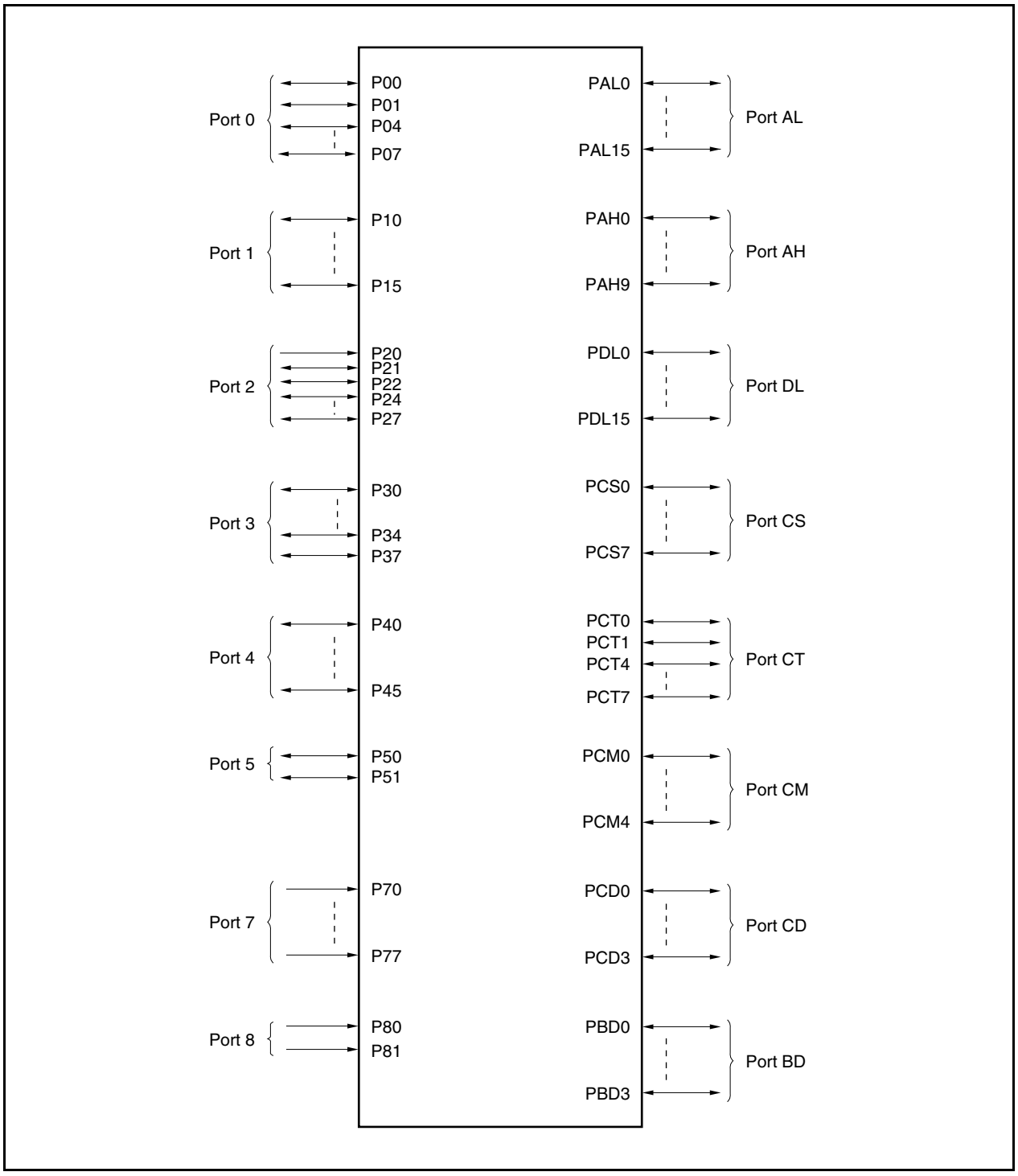
4.1 Features

- I/O ports: 112
- Input and output can be specified in 1-bit units.

4.2 Port Basic Configuration

The V850E/MA3 incorporates a total of 112 I/O ports labeled ports 0 to 5, 7, 8, AL, AH, DL, CS, CT, CM, CD, and BD. The port configuration is shown below.

Figure 4-1. Port Configuration



4.3 Port Configuration

Table 4-1. Port Configuration

Item	Configuration
Control registers	Port n register (Pn: n = 0 to 5, 7, 8, AL, AH, DL, CS, CT, CM, CD, BD) Port n mode register (PMn: n = 0 to 5, AL, AH, DL, CS, CT, CM, CD, BD) Port n mode control register (PMCn: n = 0 to 5, 7, AL, AH, DL, CS, CT, CM, CD, BD) Port n function control register (PFCn: n = 0 to 5, CS, CT) Port 1 function control expansion register (PFCEn: n = 0 to 3, 5)
Ports	Input-only: 11 I/O: 101

(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, according to each register setting.

<R>

Table 4-2. Writing/Reading Pn Register

Setting of PMCn Register	Setting of PMn Register	Writing to Pn Register	Reading from Pn Register
Port mode (PMCnm bit = 0)	Output mode (PMnm bit = 0)	Data is written to the output latch[Note]. The contents of the output latch are output from the pins.	The value of the output latch is read.
	Input mode (PMnm bit = 1)	Data is written to the output latch[Note]. The pin status is not affected.	The pin status is read.
Alternate-function mode (PMCnm bit = 1)	Output mode (PMnm bit = 0)	Data is written to the output latch[Note]. The pin status is not affected. The pin operates as an alternate-function pin.	The output status of the alternate function is read.
	Input mode (PMnm bit = 1)		The pin status is read.

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

(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 (in port 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[Note] 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

Note 01H for PMC2 register

(4) 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

(5) 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

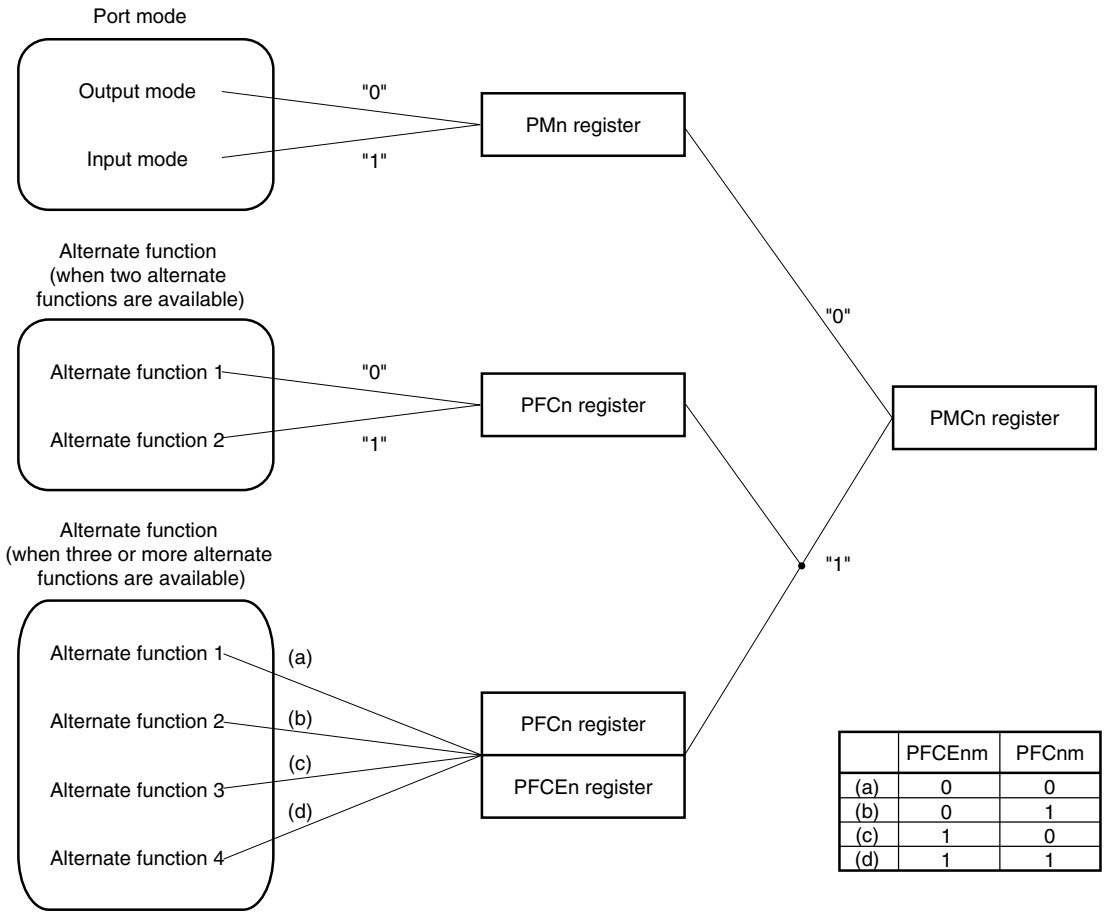
	7	6	5	4	3	2	1	0
PFCn	PFCn7	PFCn6	PFCn5	PFCn4	PFCn3	PFCn2	PFCn1	PFCn0

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

(6) 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

Remark For switching to the alternate function, see **4.6.1 (1) Procedure to change mode from port mode to alternate-function mode**.

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 4-3. Alternate-Function Pins of Port 0

Pin Name	Pin No.		Alternate-Function Pin	I/O
	GJ	F1		
P00	19	G2	$\overline{\text{INTP000}}$/TOP00/EVTP0/TIP0/INTPP00	I/O
P01	18	G1	$\overline{\text{INTP001}}$/TOP01/INTPP01	
P04	33	M3	$\overline{\text{INTP004}}$/$\overline{\text{DMARQ0}}$/INTP11/TCLR10	
P05	34	M2	$\overline{\text{INTP005}}$/$\overline{\text{DMARQ1}}$/INTP10/TCUD10	
P06	35	L3	$\overline{\text{INTP106}}$/$\overline{\text{DMARQ2}}$/TMS[Note]	
P07	36	N2	$\overline{\text{INTP107}}$/$\overline{\text{DMARQ3}}$/TCK[Note]	

Note The TMS and TCK pins are for on-chip debugging. To use the P06 and P07 pins as P06/$\overline{\text{INTP106}}$/$\overline{\text{DMARQ2}}$ and P07/$\overline{\text{INTP107}}$/$\overline{\text{DMARQ3}}$, be sure to input a low level to the $\overline{\text{TRST}}$ pin. If a high level is input to the $\overline{\text{TRST}}$ pin, the values set to the P0, PM0, PMC0, and PFC0 registers become invalid, and the P06 and P07 pins function as the TMS and TCK pins.

Caution P00, P01, and P04 to P07 have hysteresis characteristics when their alternate function is used in the input mode, but not when they are used in the port mode.

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(1) Registers

(a) Port 0 register (P0)

After reset: Undefined R/W Address: FFFFF400H

	7	6	5	4	3	2	1	0
P0	P07	P06	P05	P04	0	0	P01	P00

P0n	Control of output data (in output mode) (n = 0, 1, 4 to 7)
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	PM07	PM06	PM05	PM04	1	1	PM01	PM00

PM0n	Control of input/output mode (in port mode) (n = 0, 1, 4 to 7)
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	PMC07	PMC06	PMC05	PMC04	0	0	PMC01	PMC00

PMC07	Specification of operation mode of P07 pin
0	I/O port
1	$\overline{\text{INTP107}}$ input/$\overline{\text{DMARQ3}}$ input

PMC06	Specification of operation mode of P06 pin
0	I/O port
1	$\overline{\text{INTP106}}$ input/$\overline{\text{DMARQ2}}$ input

PMC05	Specification of operation mode of P05 pin
0	I/O port
1	$\overline{\text{INTP005}}$ input/$\overline{\text{DMARQ1}}$ input/INTP10 input/TCUD10 input

PMC04	Specification of operation mode of P04 pin
0	I/O port
1	$\overline{\text{INTP004}}$ input/$\overline{\text{DMARQ0}}$ input/INTP11 input/TCLR10 input

PMC01	Specification of operation mode of P01 pin
0	I/O port
1	$\overline{\text{INTP001}}$ input/TOP01 output/INTPP01 input

PMC00	Specification of operation mode of P00 pin
0	I/O port
1	$\overline{\text{INTP000}}$ input/TOP00 output/EVTP0 input/TIP0 input/INTPP00 input

Caution The P06/$\overline{\text{INTP106}}$/$\overline{\text{DMARQ2}}$ pin functions alternately as the N-Wire control signal TMS, and the P07/$\overline{\text{INTP107}}$/$\overline{\text{DMARQ3}}$ pin functions alternately as the N-Wire control signal TCK. These alternate functions cannot be debugged when the N-Wire type emulator is used.

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

After reset: 00H R/W Address: FFFFF700H

	7	6	5	4	3	2	1	0
PFCE0	0	0	PFCE05	PFCE04	0	0	PFCE01	PFCE00

Remark For the specification of the alternate function, see **4.3.1 (1) (f) Setting of alternate functions of port 0 pins**.

(e) Port 0 function control register (PFC0)

After reset: 00H R/W Address: FFFFF460H

	7	6	5	4	3	2	1	0
PFC0	PFC07	PFC06	PFC05	PFC04	0	0	PFC01	PFC00

Remark For the specification of the alternate function, see **4.3.1 (1) (f) Setting of alternate functions of port 0 pins**.

(f) Setting of alternate functions of port 0 pins

PFC07	Specification of Alternate Function of P07 Pin
0	$\overline{\text{INTP107}}$ input
1	$\overline{\text{DMARQ3}}$ input

PFC06	Specification of Alternate Function of P06 Pin
0	$\overline{\text{INTP106}}$ input
1	$\overline{\text{DMARQ2}}$ input

PFCE05	PFC05	Specification of Alternate Function of P05 Pin
0	0	$\overline{\text{INTP005}}$ input
0	1	$\overline{\text{DMARQ1}}$ input
1	0	INTP10 input/TCUD10 input
1	1	Setting prohibited

PFCE04	PFC04	Specification of Alternate Function of P04 Pin
0	0	$\overline{\text{INTP004}}$ input
0	1	$\overline{\text{DMARQ0}}$ input
1	0	INTP11 input/TCLR10 input
1	1	Setting prohibited

PFCE01	PFC01	Specification of Alternate Function of P01 Pin
0	0	$\overline{\text{INTP001}}$ input
0	1	TOP01 output
1	0	INTPP01 input
1	1	Setting prohibited

PFCE00	PFC00	Specification of Alternate Function of P00 Pin
0	0	$\overline{\text{INTP000}}$ input
0	1	TOP00 output
1	0	EVTP0 input/TIP0 input/INTPP00 input
1	1	Setting prohibited

(2) Block diagram

Figure 4-3. Block Diagram of P00 Pin

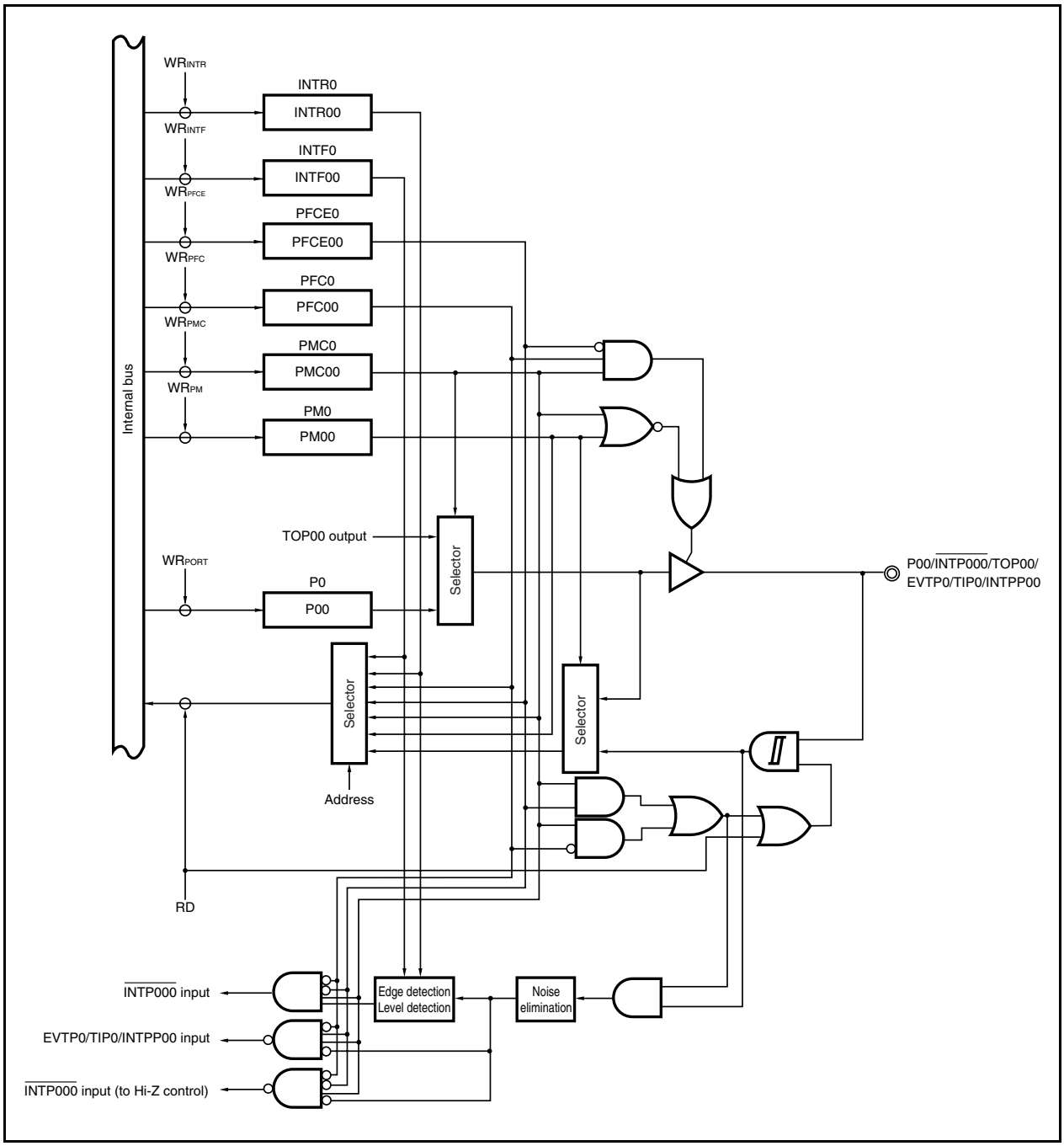


Figure 4-4. Block Diagram of P01 Pin

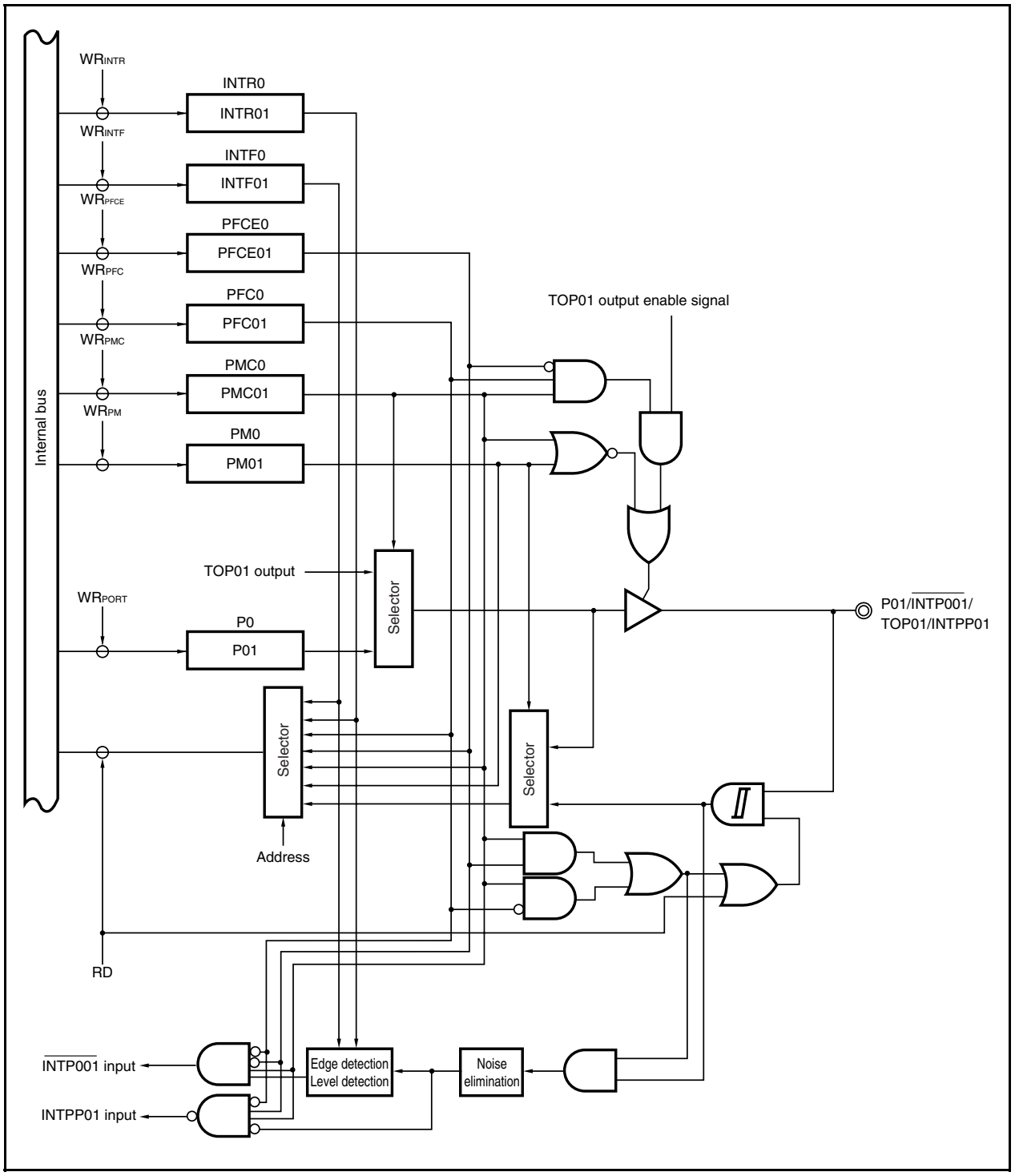
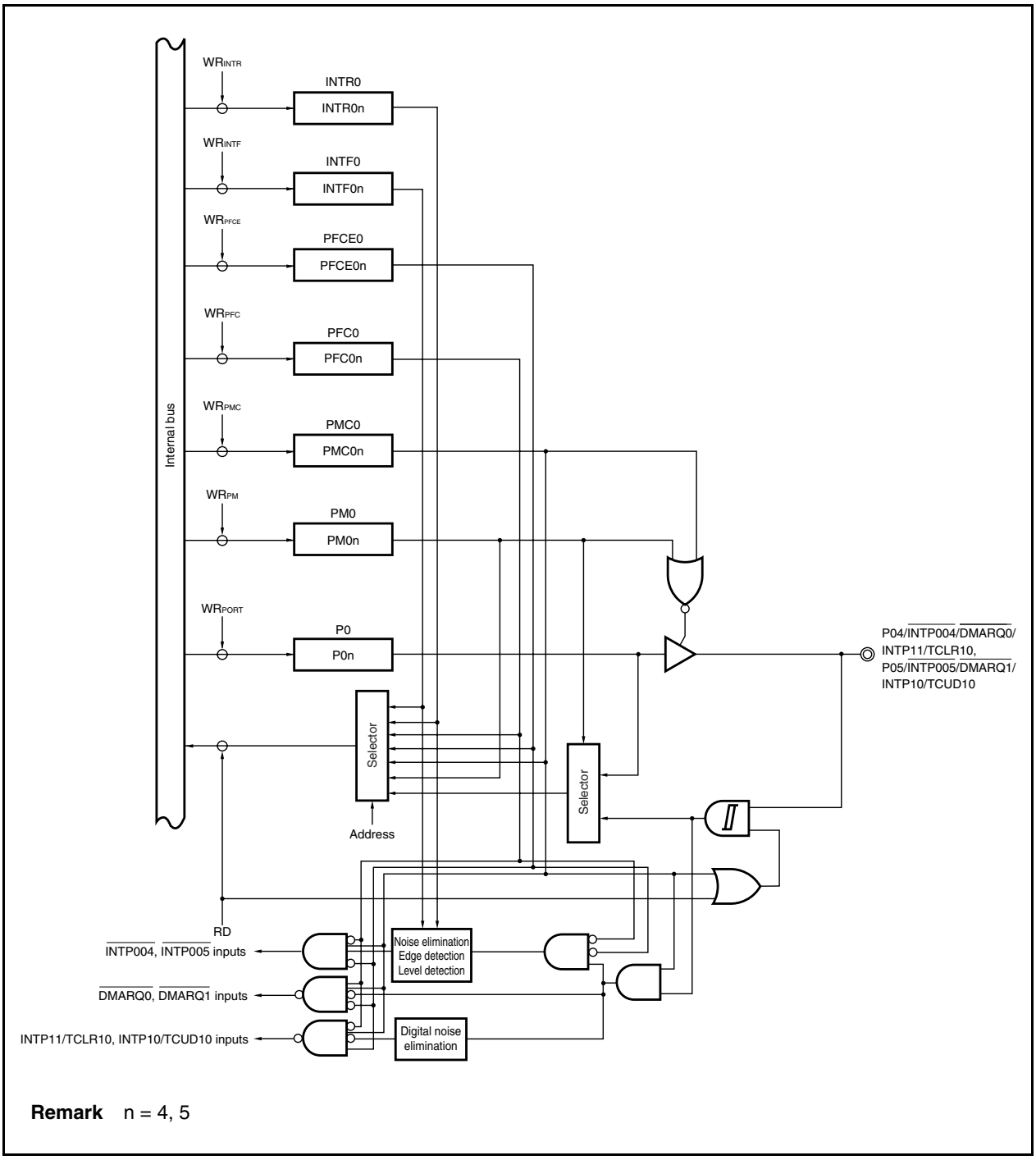
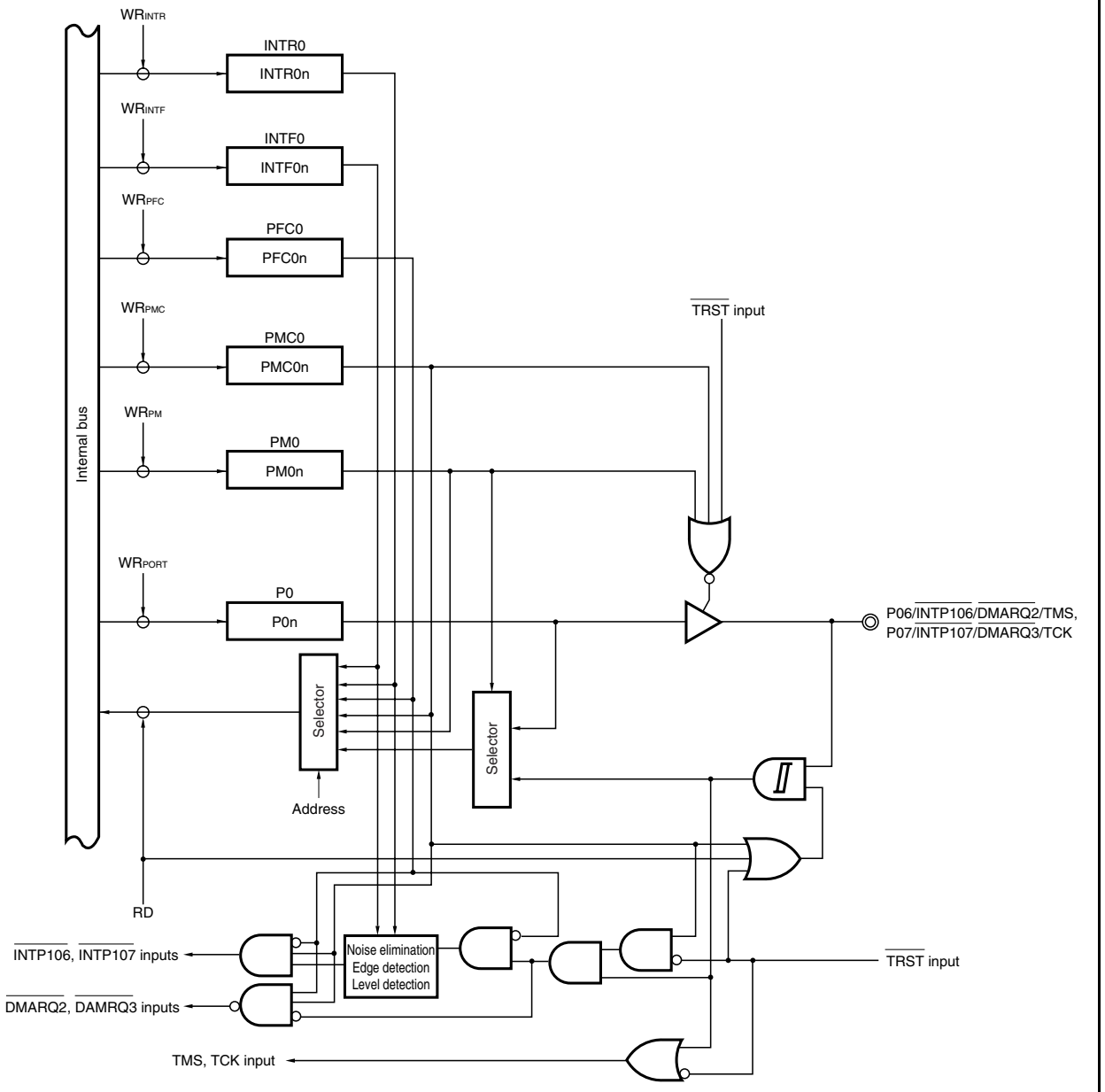


Figure 4-5. Block Diagram of P04 and P05 Pins



Remark n = 4, 5

Figure 4-6. Block Diagram of P06 and P07 Pins



Remark n = 6, 7

4.3.2 Port 1

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

Port 1 has an alternate function as the following pins.

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

Pin Name	Pin No.		Alternate-Function Pin	I/O
	GJ	F1		
P10	27	J3	$\overline{\text{INTP010}}$/TOQB1/INTPQ0/TOQ0	I/O
P11	26	K1	$\overline{\text{INTP011}}$/TOQT1/INTPQ1/TOQ1	
P12	25	J2	$\overline{\text{INTP012}}$/TOQT2/INTPQ2/TOQ2	
P13	22	H3	$\overline{\text{INTP013}}$/TOQT3/INTPQ3/TOQ3	
P14	21	H2	$\overline{\text{INTP114}}$/TOQB2/TIQ	
P15	20	H1	$\overline{\text{INTP115}}$/TOQB3/EVTQ	

Caution P10 to P15 have hysteresis characteristics when their alternate function is used in the input mode, but not when they are used in the port mode.

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(1) Registers

(a) Port 1 register (P1)

After reset: Undefined R/W Address: FFFFF402H

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

P1n	Control of output data (in output mode) (n = 0 to 5)
0	Output 0.
1	Output 1.

(b) Port 1 mode register (PM1)

After reset: FFH R/W Address: FFFFF422H

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

PM1n	Control of input/output mode (in port mode) (n = 0 to 5)
0	Output mode
1	Input mode

(c) Port 1 mode control register (PMC1)

After reset: 00H R/W Address: FFFFF442H

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

PMC15	Specification of operation mode of P15 pin
0	I/O port
1	$\overline{\text{INTP115}}$ input/TOQB3 output/EVTQ input

PMC14	Specification of operation mode of P14 pin
0	I/O port
1	$\overline{\text{INTP114}}$ input/TOQB2 output/TIQ input

PMC13	Specification of operation mode of P13 pin
0	I/O port
1	$\overline{\text{INTP013}}$ input/TOQT3 output/INTPQ3 input/TOQ3 output

PMC12	Specification of operation mode of P12 pin
0	I/O port
1	$\overline{\text{INTP012}}$ input/TOQT2 output/INTPQ2 input/TOQ2 output

PMC11	Specification of operation mode of P11 pin
0	I/O port
1	$\overline{\text{INTP011}}$ input/TOQT1 output/INTPQ1 input/TOQ1 output

PMC10	Specification of operation mode of P10 pin
0	I/O port
1	$\overline{\text{INTP010}}$ input/TOQB1 output/INTPQ0 input/TOQ0 output

(d) 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	PFCE15	PFCE14	PFCE13	PFCE12	PFCE11	PFCE10

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

(e) Port 1 function control register (PFC1)

After reset: 00H R/W Address: FFFFF462H

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

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

(f) Setting of alternate functions of port 1 pins

PFCE15	PFC15	Specification of Alternate Function of P15 Pin
0	0	$\overline{\text{INTP115}}$ input
0	1	TOQB3 output
1	0	EVTQ input
1	1	Setting prohibited

PFCE14	PFC14	Specification of Alternate Function of P14 Pin
0	0	$\overline{\text{INTP114}}$ input
0	1	TOQB2 output
1	0	TIQ input
1	1	Setting prohibited

PFCE13	PFC13	Specification of Alternate Function of P13 Pin
0	0	$\overline{\text{INTP013}}$ input
0	1	TOQT3 output
1	0	INTPQ3 input
1	1	TOQ3 output

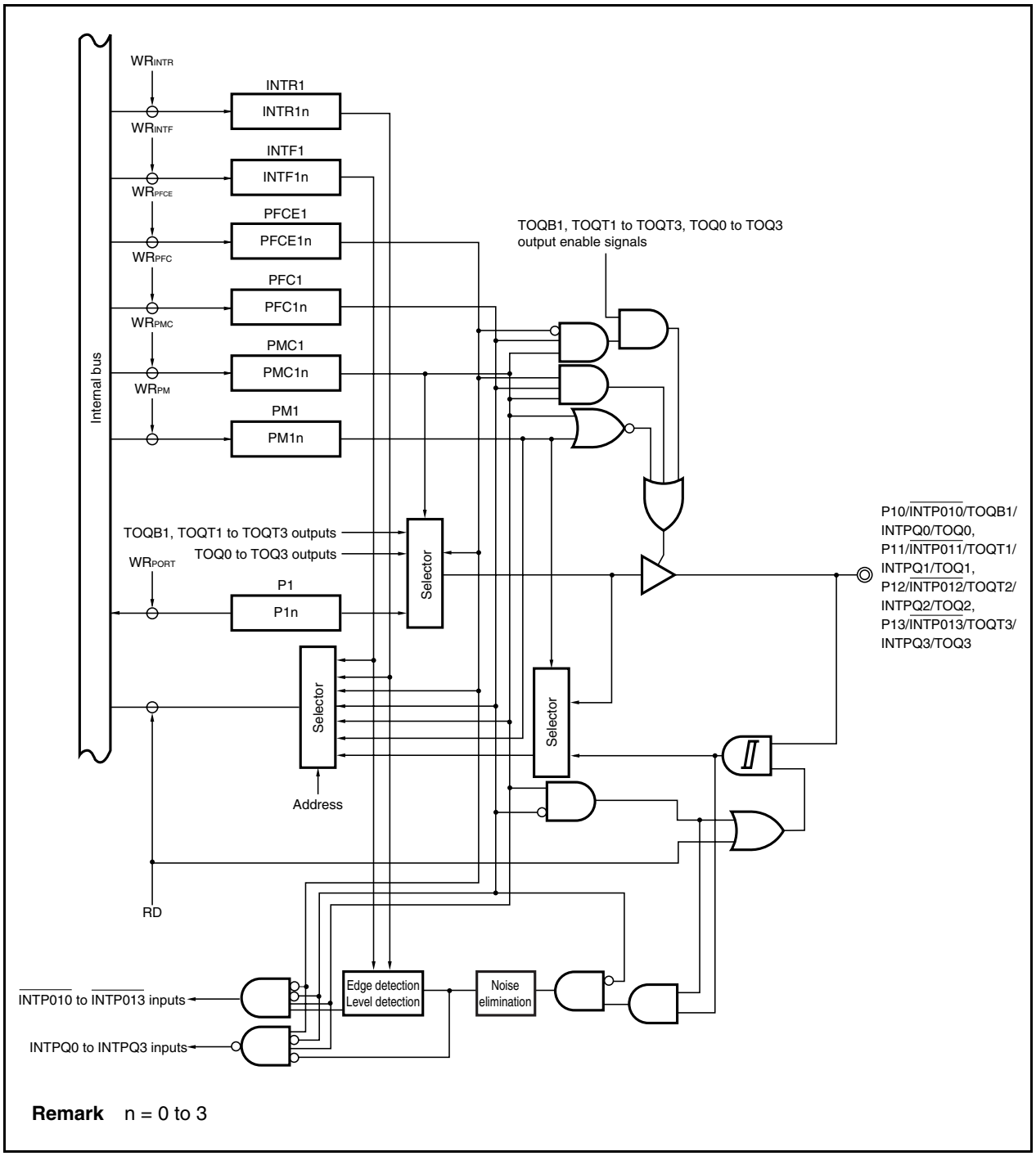
PFCE12	PFC12	Specification of Alternate Function of P12 Pin
0	0	$\overline{\text{INTP012}}$ input
0	1	TOQT2 output
1	0	INTPQ2 input
1	1	TOQ2 output

PFCE11	PFC11	Specification of Alternate Function of P11 Pin
0	0	$\overline{\text{INTP011}}$ input
0	1	TOQT1 output
1	0	INTPQ1 input
1	1	TOQ1 output

PFCE10	PFC10	Specification of Alternate Function of P10 Pin
0	0	$\overline{\text{INTP010}}$ input
0	1	TOQB1 output
1	0	INTPQ0 input
1	1	TOQ0 output

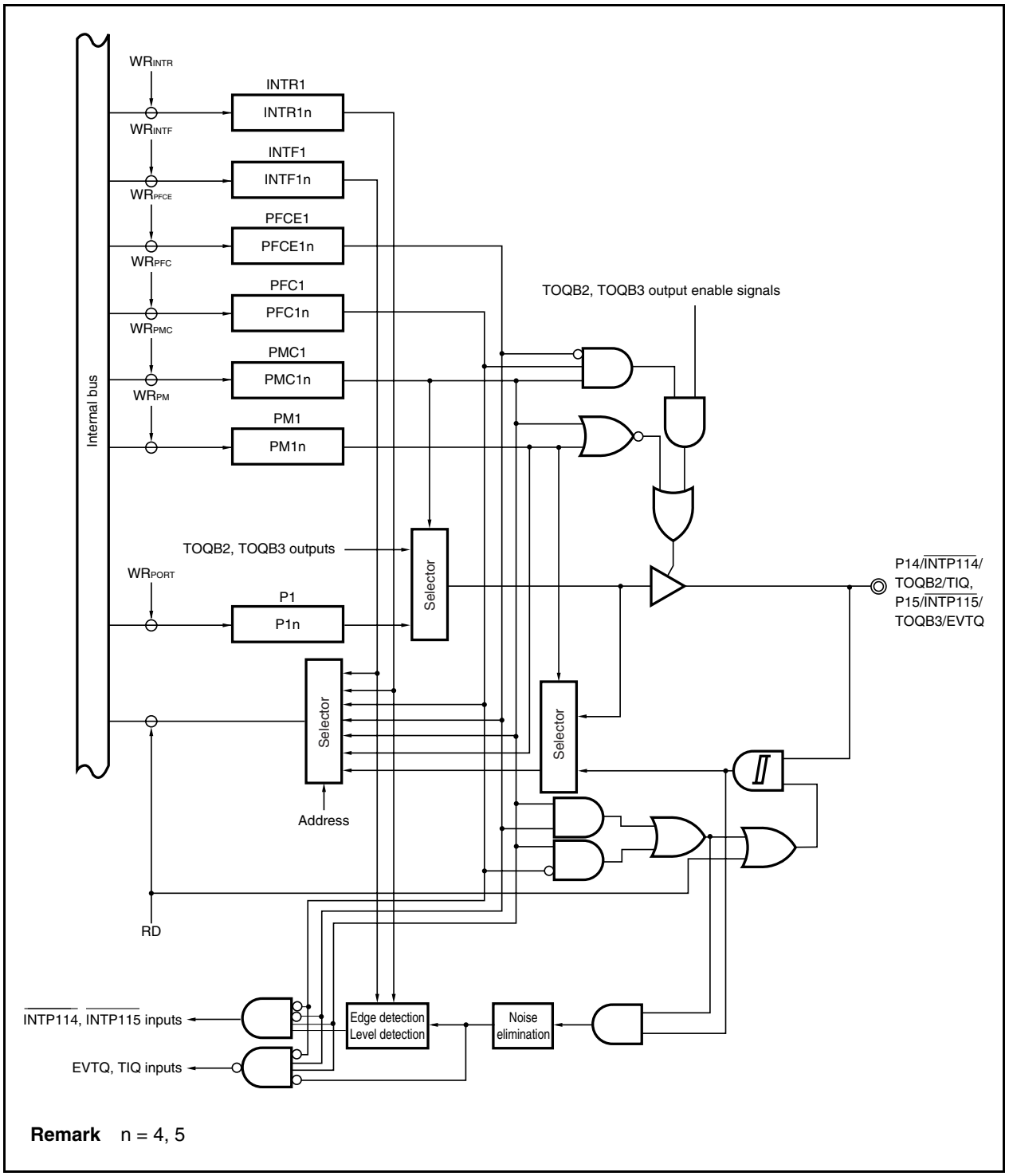
(2) Block diagram

Figure 4-7. Block Diagram of P10 to P13 Pins



Remark n = 0 to 3

Figure 4-8. Block Diagram of P14 and P15 Pins



Remark n = 4, 5

4.3.3 Port 2

Port 2 can be set in the input or output mode in 1-bit units, except P20, which is input-only.

P20 always functions as an NMI pin. The level of the NMI pin can be read by reading the P2.P20 bit.

Port 2 functions alternately as the following pins.

Port 2 has an alternate function as the following pins.

Table 4-5. Alternate-Function Pins of Port 2

Pin Name	Pin No.		Alternate-Function Pin	I/O
	GJ	F1		
P20	84	J14	NMI	Input
P21	44	P4	$\overline{\text{INTP021}}$/TOP10/EVTP1/TIP1/INTPP10	I/O
P22	43	L5	$\overline{\text{INTP022}}$/TOP11/INTPP11	
P24	31	K3	$\overline{\text{INTP124}}$/$\overline{\text{TC0}}$	
P25	30	L2	$\overline{\text{INTP125}}$/$\overline{\text{TC1}}$/TIUD10/TO10	
P26	29	K4	$\overline{\text{INTP126}}$/$\overline{\text{TC2}}$/TDI[Note]	
P27	28	K2	$\overline{\text{TC3}}$/TDO[Note]	

Note The TDI and TDO pins are for on-chip debugging. To use the P26 and P27 pins as P26/$\overline{\text{INTP126}}$/$\overline{\text{TC2}}$ and P27/$\overline{\text{TC3}}$, be sure to input a low level to the $\overline{\text{TRST}}$ pin. If a high level is input to the $\overline{\text{TRST}}$ pin, the values set to the P2, PM2, PMC2, and PFC2 registers become invalid, and the P26 and P27 pins function as the TDI and TDO pins.

Caution P20 to P22 and P24 to P26 have hysteresis characteristics when their alternate function is used in the input mode, but not when they are used in the port mode.

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(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	0	P22	P21	P20

P2n	Control of output data (in output mode) (n = 0 to 2, 4 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	1	PM22	PM21	1

PM2n	Control of input/output mode (in port mode) (n = 1, 2, 4 to 7)
0	Output mode
1	Input mode

(c) Port 2 mode control register (PMC2)

After reset: 01H R/W Address: FFFFF444H

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

PMC27	Specification of operation mode of P27 pin
0	I/O port
1	$\overline{TC3}$ output

PMC26	Specification of operation mode of P26 pin
0	I/O port
1	$\overline{INTP126}$ input/$\overline{TC2}$ output

PMC25	Specification of operation mode of P25 pin
0	I/O port
1	$\overline{INTP125}$ input/$\overline{TC1}$ output/TIUD10 input/TO10 output

PMC24	Specification of operation mode of P24 pin
0	I/O port
1	$\overline{INTP124}$ input/$\overline{TC0}$ output

PMC22	Specification of operation mode of P22 pin
0	I/O port
1	$\overline{INTP022}$ input/TOP11 output/INTPP11 input

PMC21	Specification of operation mode of P21 pin
0	I/O port
1	$\overline{INTP021}$ input/TOP10 output/EVTP1 input/TIP1 input/INTPP10 input

Caution The P26/$\overline{INTP126}$/$\overline{TC2}$ pin functions alternately as the N-Wire control signal TDI, and the P27/$\overline{TC3}$ pin functions alternately as the N-Wire control signal TDO. These alternate functions cannot be debugged when the N-Wire type emulator is used.

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

After reset: 00H R/W Address: FFFFF704H

	7	6	5	4	3	2	1	0
PFCE2	0	0	PFCE25	0	0	PFCE22	PFCE21	0

Remark For the specification of the alternate function, see **4.3.3 (1) (f) Setting of alternate functions of port 2 pins**.

(e) Port 2 function control register (PFC2)

After reset: 00H R/W Address: FFFFF464H

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

Remark For the specification of the alternate function, see **4.3.3 (1) (f) Setting of alternate functions of port 2 pins**.

(f) Setting of alternate functions of port 2 pins

PFC26	Specification of Alternate Function of P26 Pin
0	$\overline{\text{INTP126}}$ input
1	$\overline{\text{TC2}}$ output

PFCE25	PFC25	Specification of Alternate Function of P25 Pin
0	0	$\overline{\text{INTP125}}$ input
0	1	$\overline{\text{TC1}}$ output
1	0	TIUD10 input
1	1	TO10 output

PFC24	Specification of Alternate Function of P24 Pin
0	$\overline{\text{INTP124}}$ input
1	$\overline{\text{TC0}}$ output

PFCE22	PFC22	Specification of Alternate Function of P22 Pin
0	0	$\overline{\text{INTP022}}$ input
0	1	TOP11 output
1	0	INTPP11 input
1	1	Setting prohibited

PFCE21	PFC21	Specification of Alternate Function of P21 Pin
0	0	$\overline{\text{INTP021}}$ input
0	1	TOP10 output
1	0	EVTP1 input/TIP1 input/INTPP10 input
1	1	Setting prohibited

(2) Block diagram

Figure 4-9. Block Diagram of P20 Pin

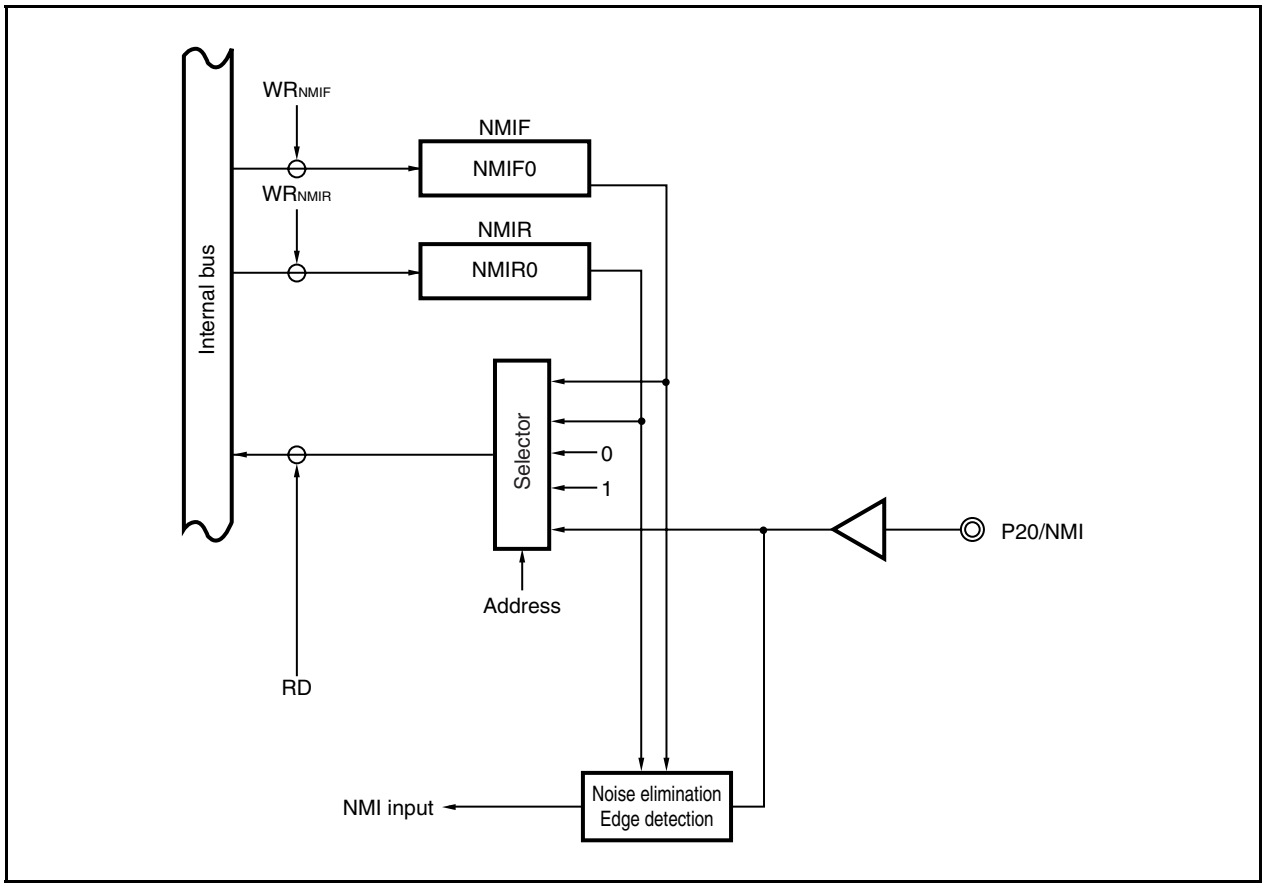
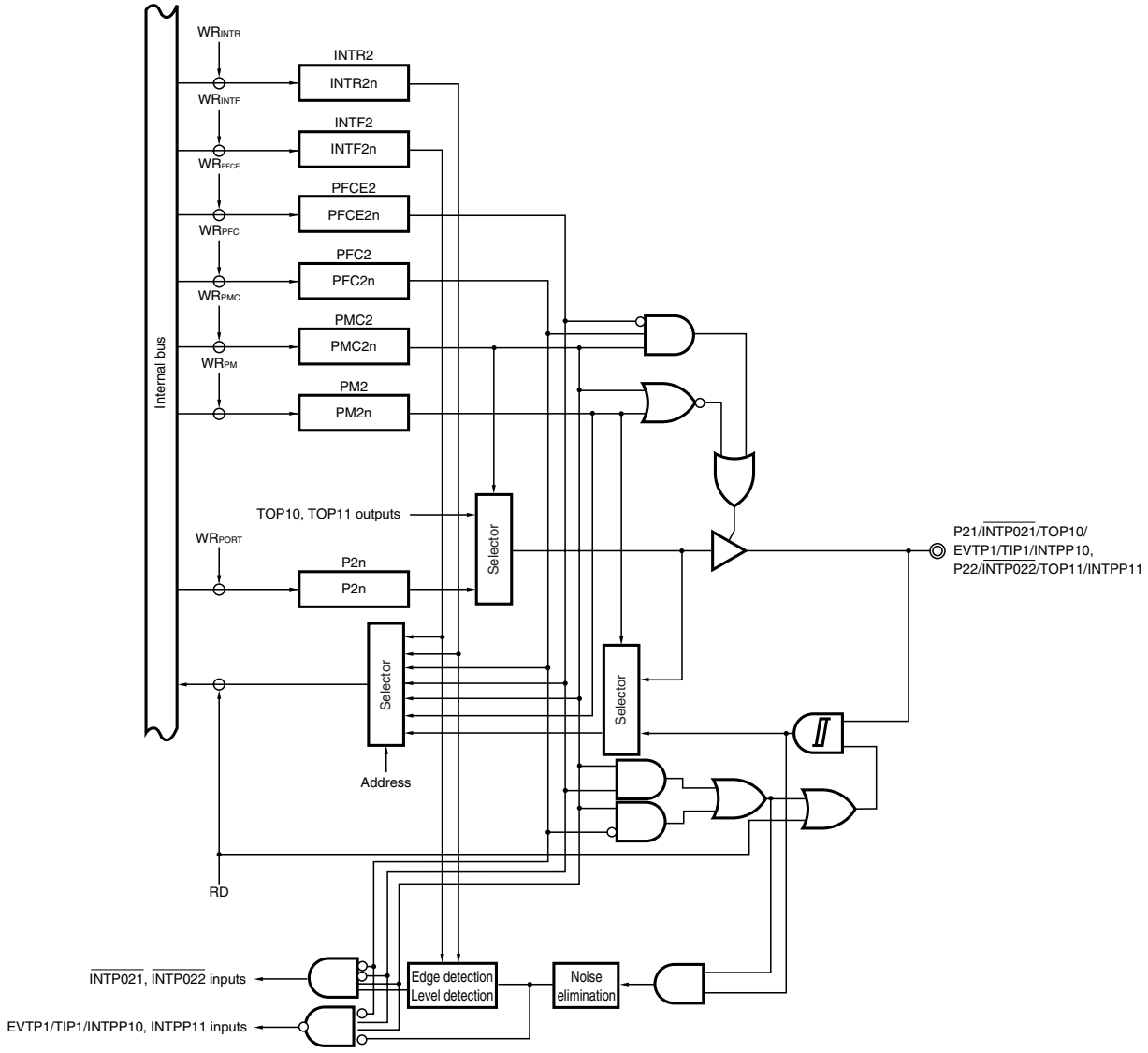


Figure 4-10. Block Diagram of P21 and P22 Pins



Remark n = 1, 2

Figure 4-11. Block Diagram of P24 Pin

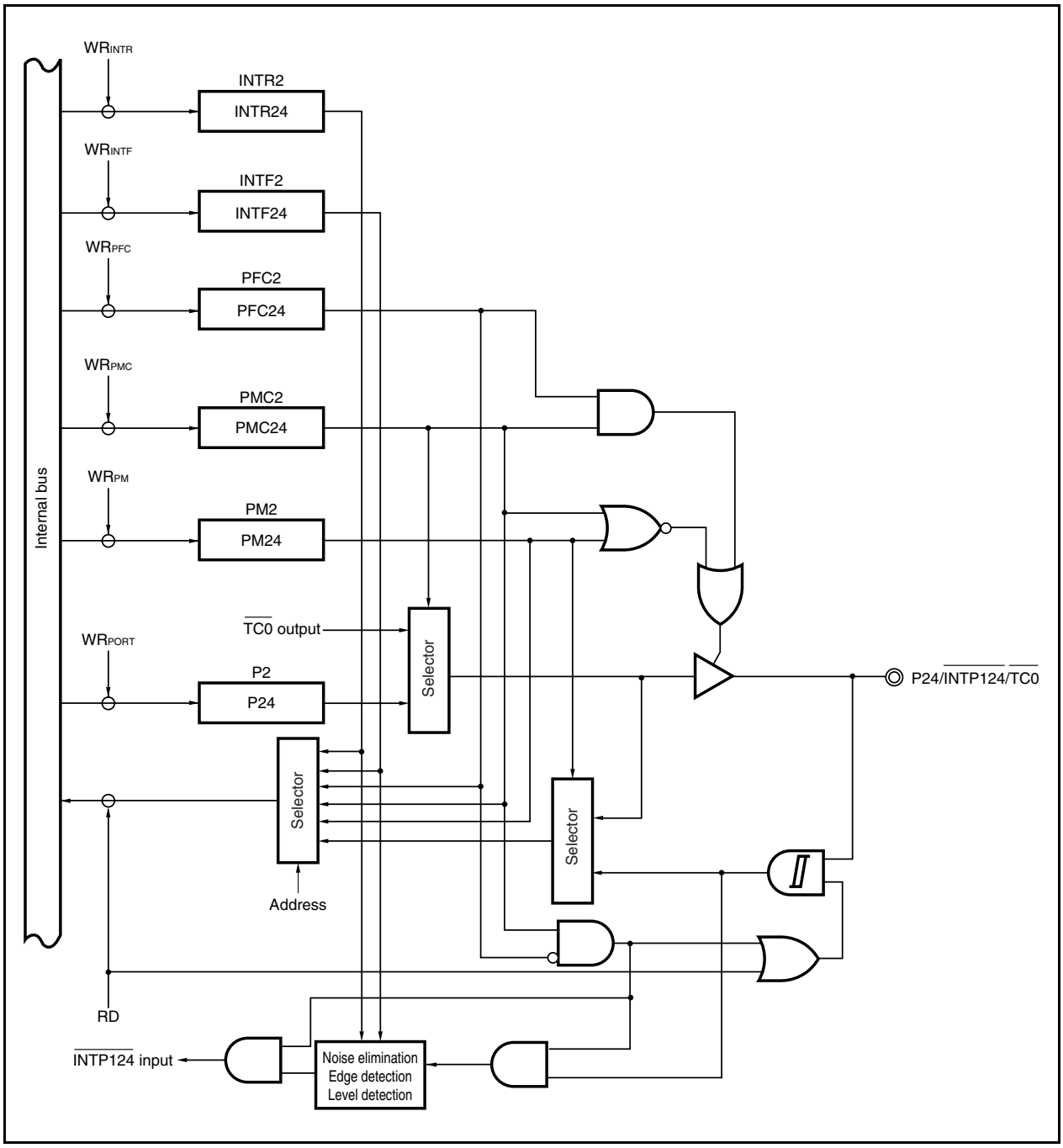


Figure 4-12. Block Diagram of P25 Pin

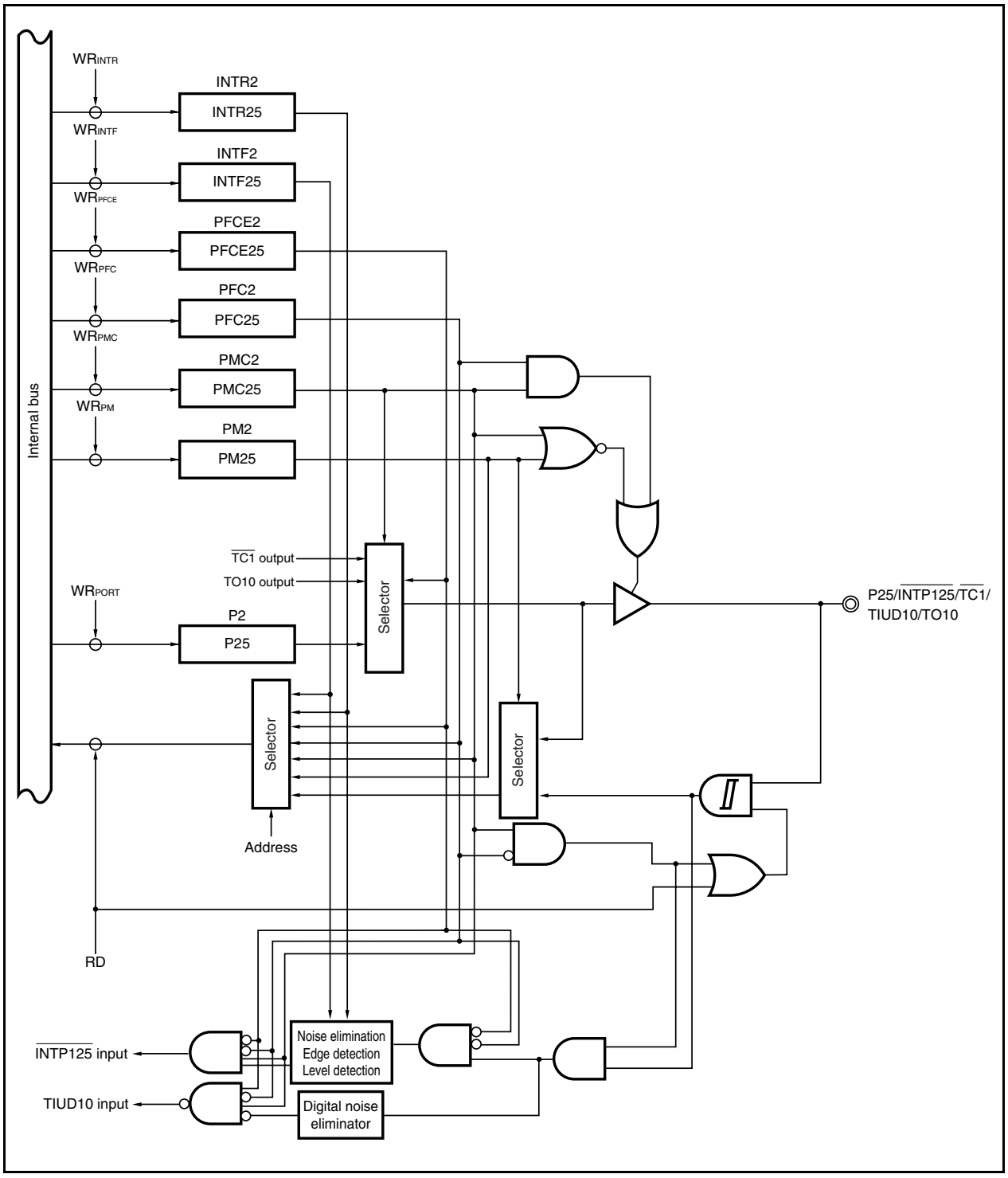


Figure 4-13. Block Diagram of P26 Pin

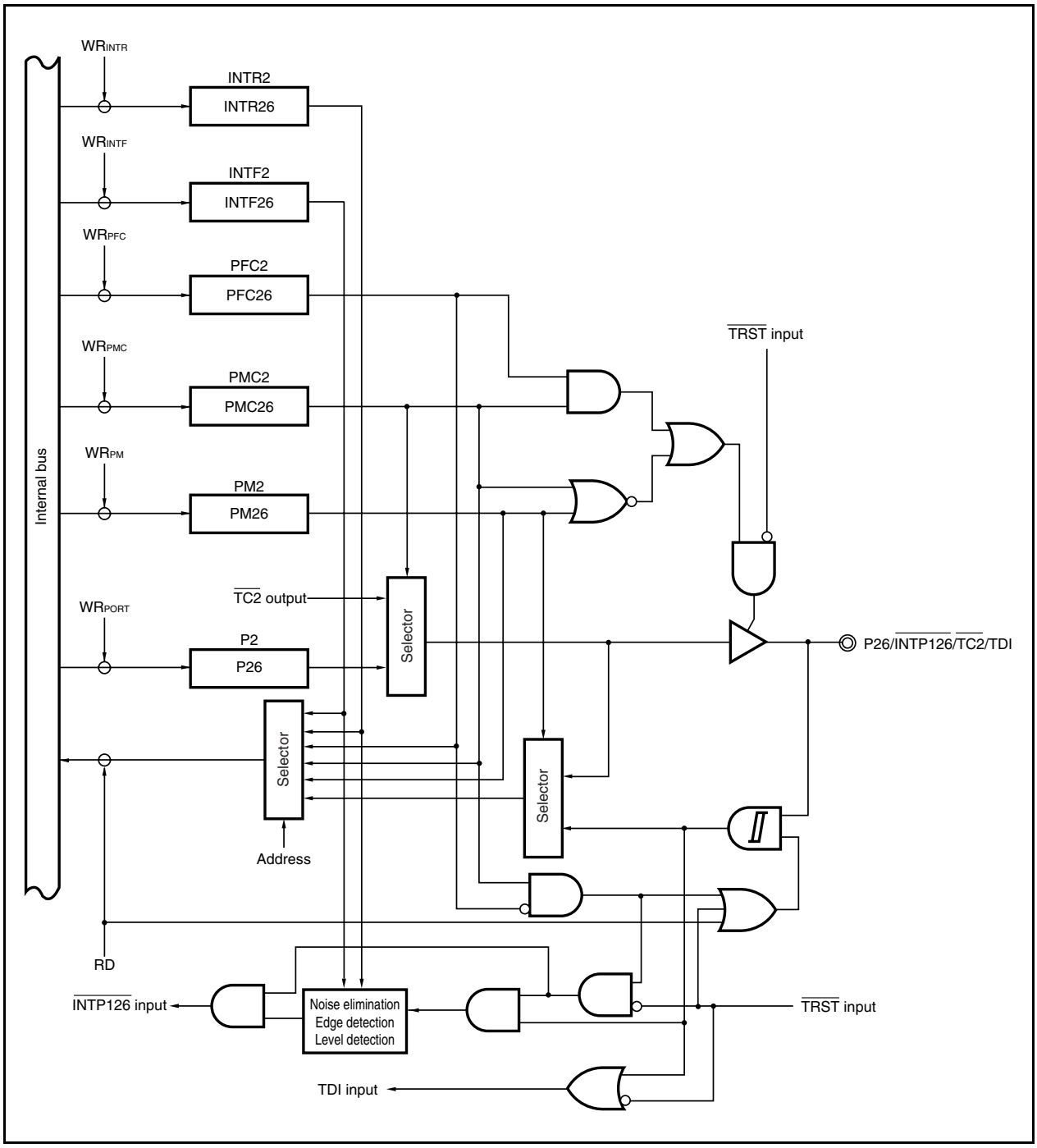
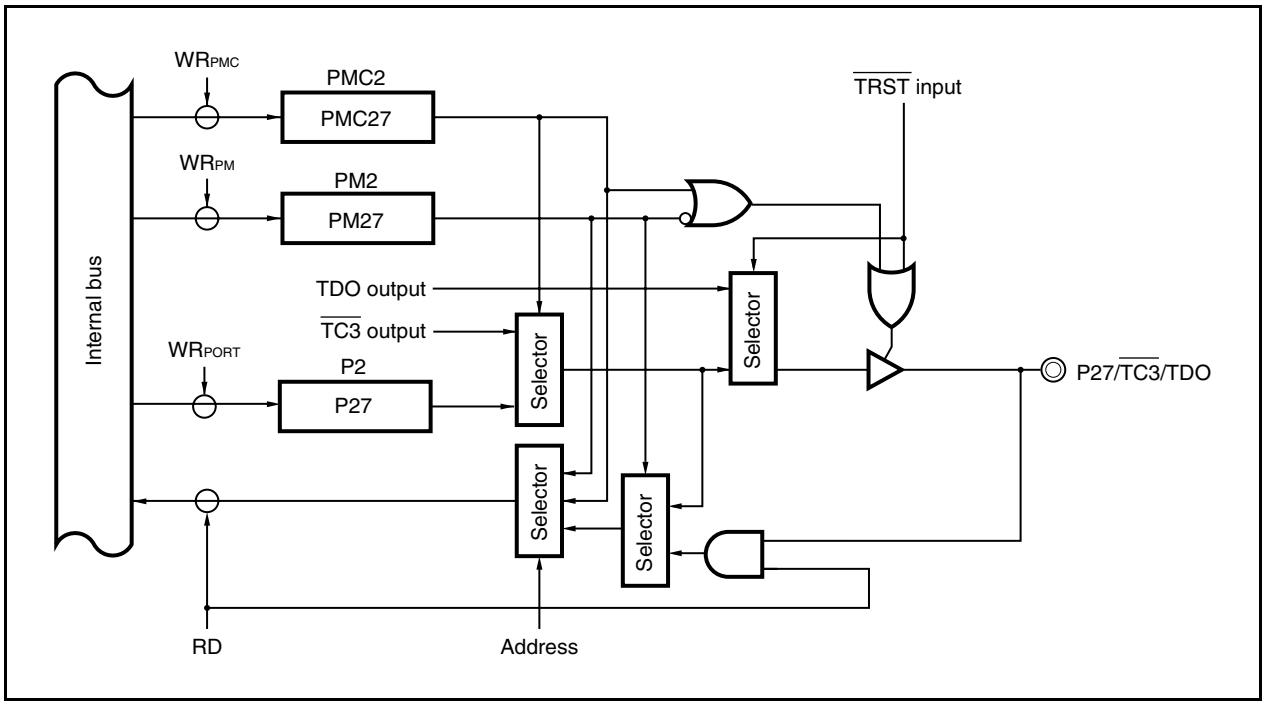


Figure 4-14. Block Diagram of P27 Pin



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 4-6. Alternate-Function Pins of Port 3

Pin Name	Pin No.		Alternate-Function Pin	I/O
	GJ	F1		
P30	49	N6	$\overline{\text{INTP130}}$/TXD2/SO2	I/O
P31	48	M6	$\overline{\text{INTP131}}$/RXD2/SI2	
P32	47	L6	$\overline{\text{INTP132}}$/ASCK2/SCK2	
P33	46	N5	$\overline{\text{INTP133}}$/TXD3/SDA[Note]	
P34	45	M5	$\overline{\text{INTP134}}$/RXD3/SCL[Note]	
P37	83	J11	$\overline{\text{INTP137}}$/ADTRG	

Note I^2C bus versions (Y products) only (see **Table 1-1**)
When the P33 and P34 pins are used as the SDA and SCL pins, respectively, they function as dummy open-drain output pins (P-ch is always off).

Caution P30 to P34 and P37 have hysteresis characteristics when their alternate function is used in the input mode, but not when they are used in the port mode.

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(1) Registers

(a) Port 3 register (P3)

After reset: Undefined R/W Address: FFFFF406H

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

P3n	Control of output data (in output mode) (n = 0 to 4, 7)
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	PM37	1	1	PM34	PM33	PM32	PM31	PM30

PM3n	Control of input/output mode (in port mode) (n = 0 to 4, 7)
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	PMC37	0	0	PMC34	PMC33	PMC32	PMC31	PMC30

PMC37	Specification of operation mode of P37 pin
0	I/O port
1	$\overline{\text{INTP137}}$ input/ADTRG output

PMC34	Specification of operation mode of P34 pin
0	I/O port
1	$\overline{\text{INTP134}}$ input/RXD3 input/SCL[Note] I/O

PMC33	Specification of operation mode of P33 pin
0	I/O port
1	$\overline{\text{INTP133}}$ input/TXD3 output/SDA[Note] I/O

PMC32	Specification of operation mode of P32 pin
0	I/O port
1	$\overline{\text{INTP132}}$ input/ASCK2 input/SCK2 I/O

PMC31	Specification of operation mode of P31 pin
0	I/O port
1	$\overline{\text{INTP131}}$ input/RXD2 input/SI2 input

PMC30	Specification of operation mode of P30 pin
0	I/O port
1	$\overline{\text{INTP130}}$ input/TXD2 output/SO2 output

Note I^2C bus versions (Y products) only (see **Table 1-1**)

When the P33 and P34 pins are used as the SDA and SCL pins, respectively, they function as dummy open-drain output pins (P-ch is always off).

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

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	0	0	0	PFCE34	PFCE33	PFCE32	PFCE31	PFCE30

Remark For the specification of the alternate function, see **4.3.4 (1) (f) Setting of alternate functions of port 3 pins**.

(e) 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	PFC34	PFC33	PFC32	PFC31	PFC30

Remark For the specification of the alternate function, see **4.3.4 (1) (f) Setting of alternate functions of port 3 pins**.

(f) Setting of alternate functions of port 3 pins

PFCE34	PFC34	Specification of Alternate Function of P34 Pin
0	0	$\overline{\text{INTP134}}$ input
0	1	RXD3 input
1	0	SCL[Note] I/O
1	1	Setting prohibited

PFCE33	PFC33	Specification of Alternate Function of P33 Pin
0	0	$\overline{\text{INTP133}}$ input
0	1	TXD3 output
1	0	SDA[Note] I/O
1	1	Setting prohibited

PFCE32	PFC32	Specification of Alternate Function of P32 Pin
0	0	$\overline{\text{INTP132}}$ input
0	1	ASCK2 input
1	0	SCK2 I/O
1	1	Setting prohibited

PFCE31	PFC31	Specification of Alternate Function of P31 Pin
0	0	$\overline{\text{INTP131}}$ input
0	1	RXD2 input
1	0	SI2 input
1	1	Setting prohibited

PFCE30	PFC30	Specification of Alternate Function of P30 Pin
0	0	$\overline{\text{INTP130}}$ input
0	1	TXD2 output
1	0	SO2 output
1	1	Setting prohibited

Note I^2C bus versions (Y products) only (see **Table 1-1**)

When the P33 and P34 pins are used as the SDA and SCL pins, respectively, they function as dummy open-drain output pins (P-ch is always off).

(2) Block diagram

Figure 4-15. Block Diagram of P30 Pin

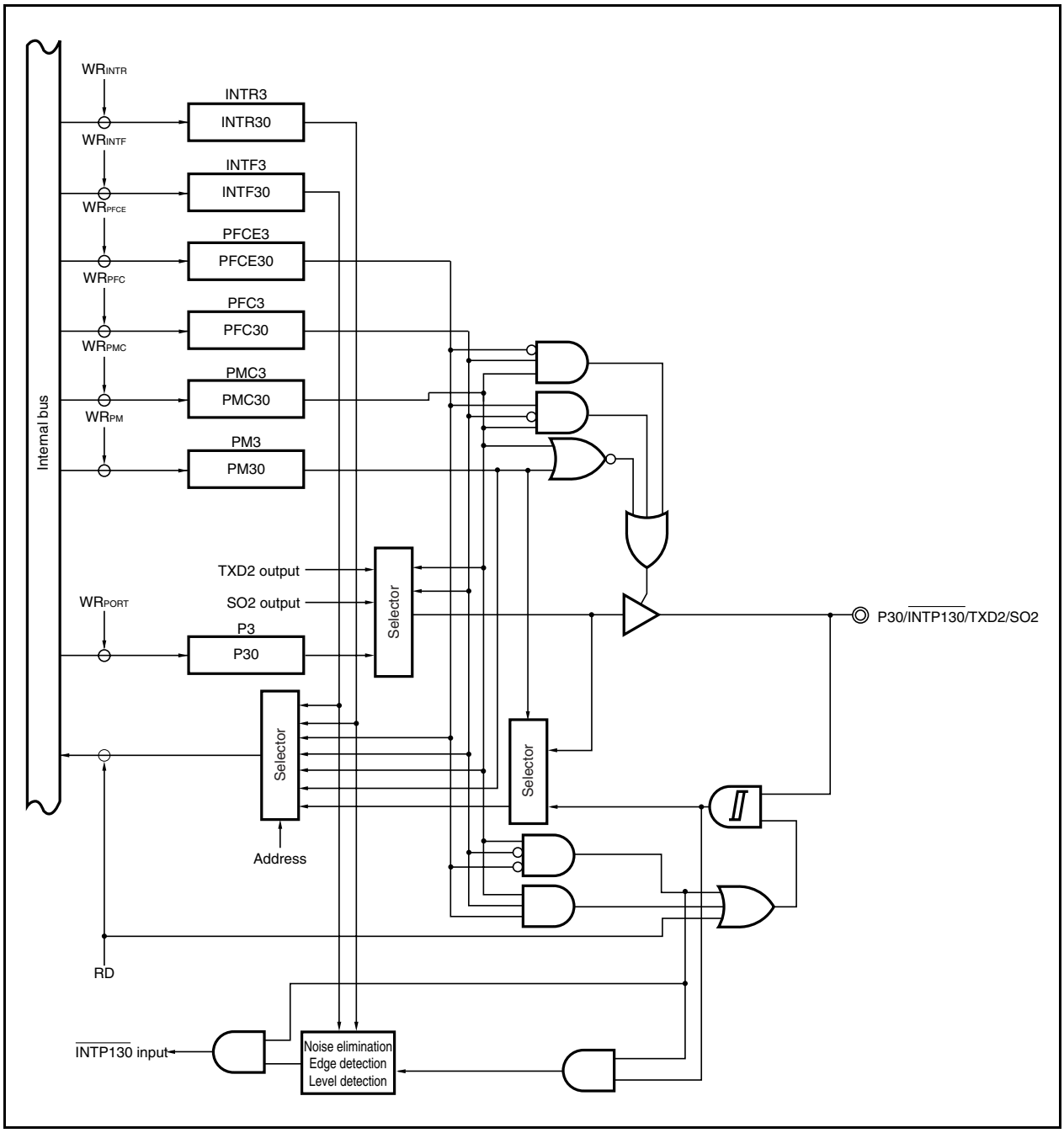


Figure 4-16. Block Diagram of P31 Pin

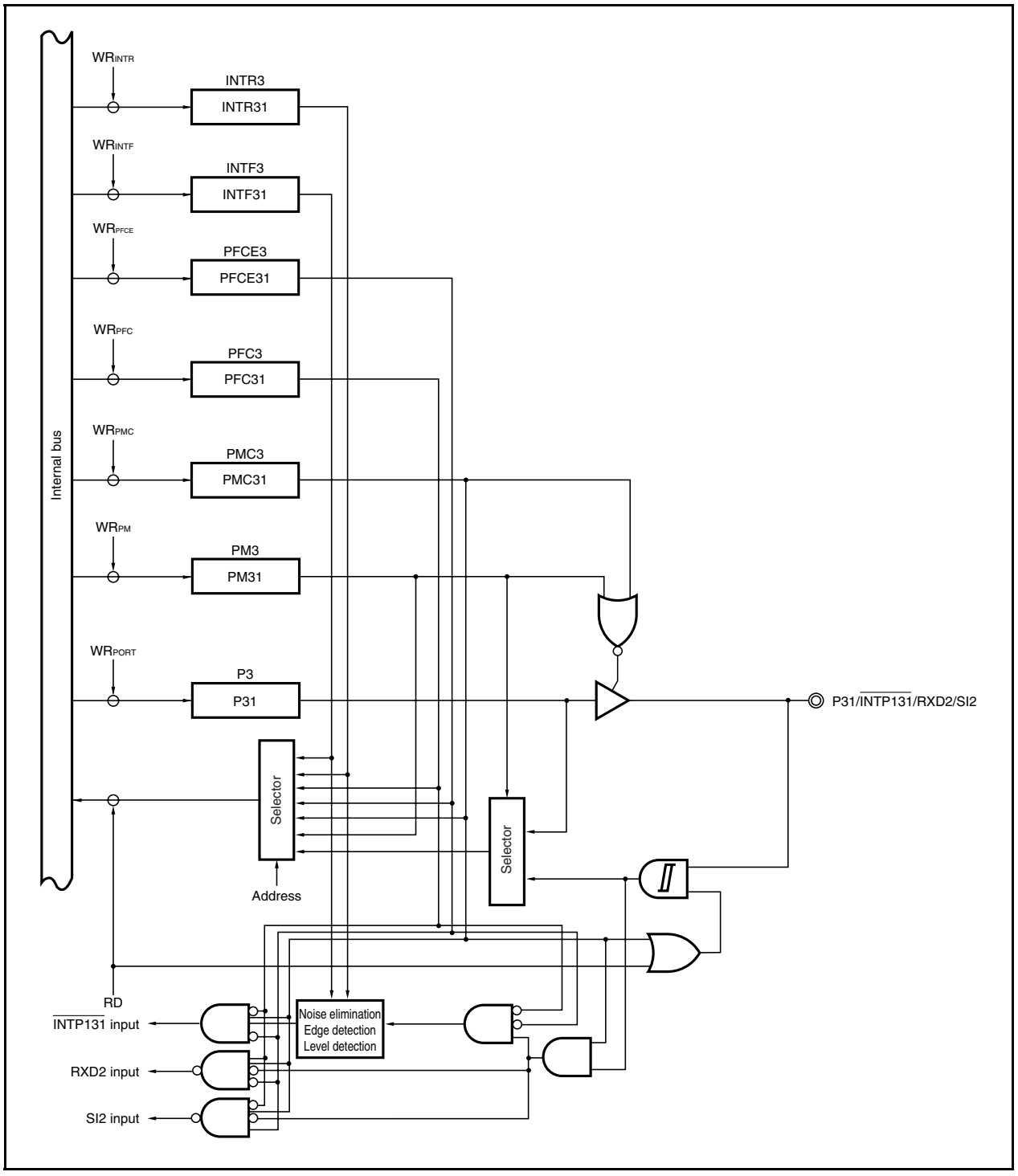


Figure 4-17. Block Diagram of P32 Pin

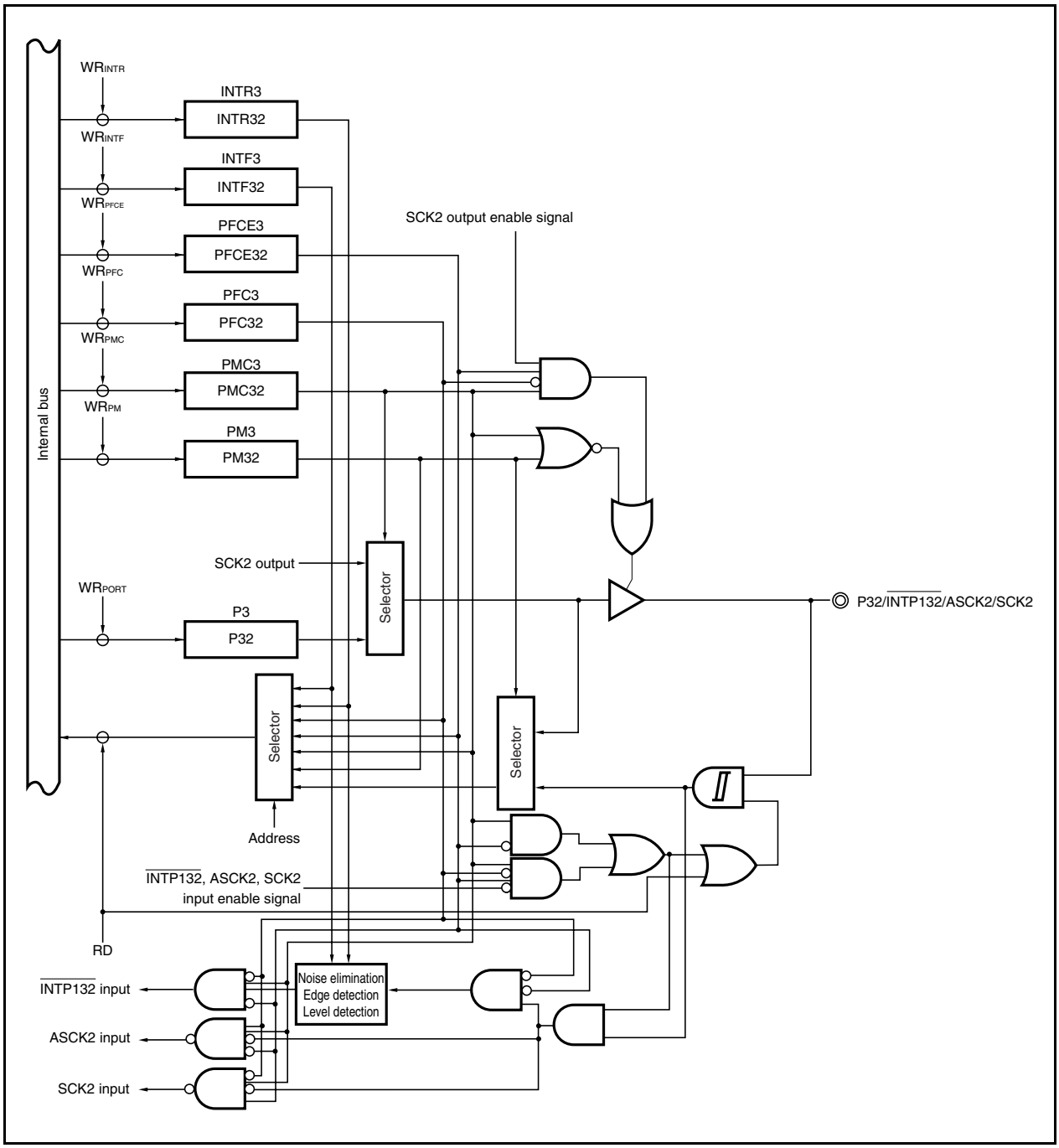
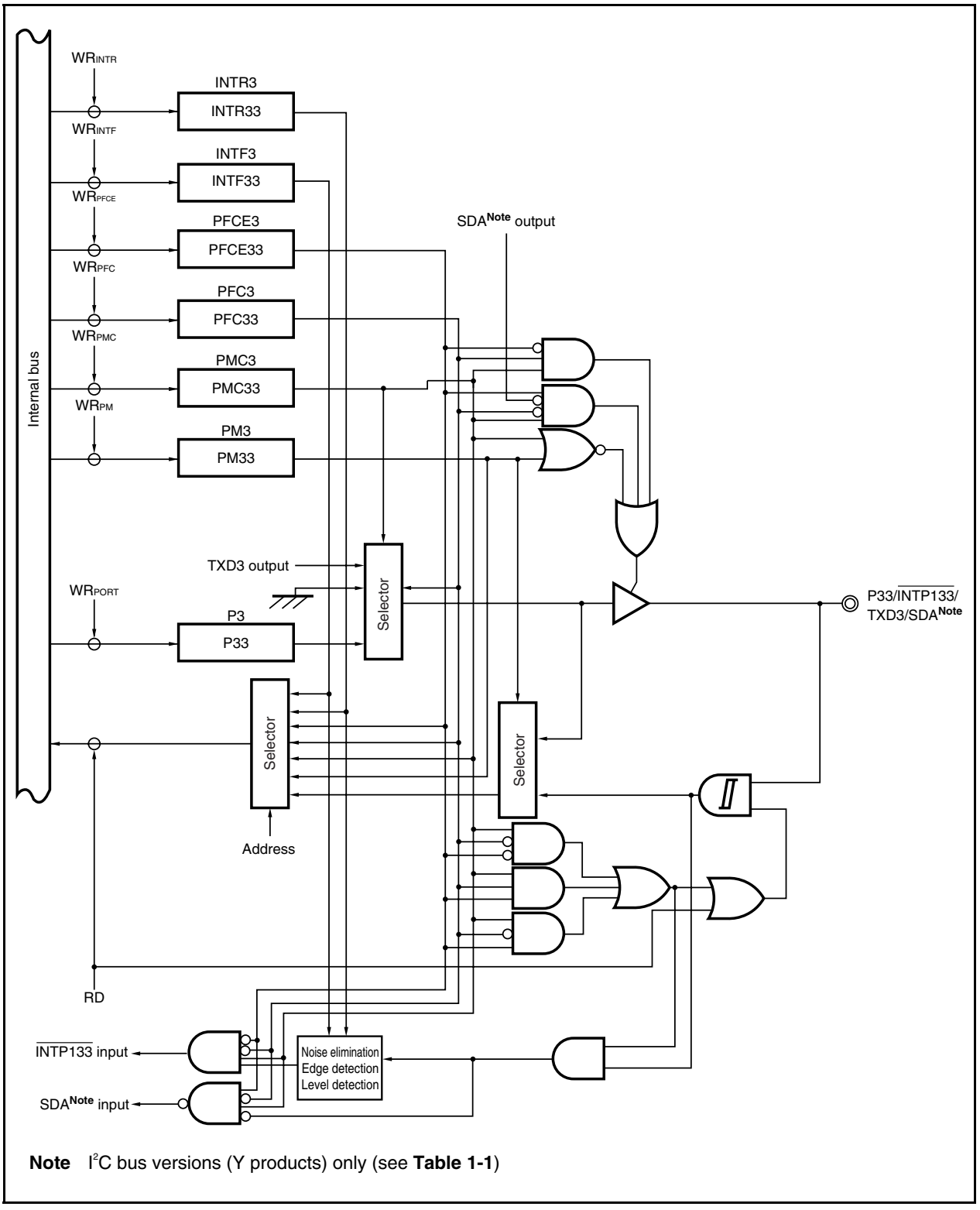
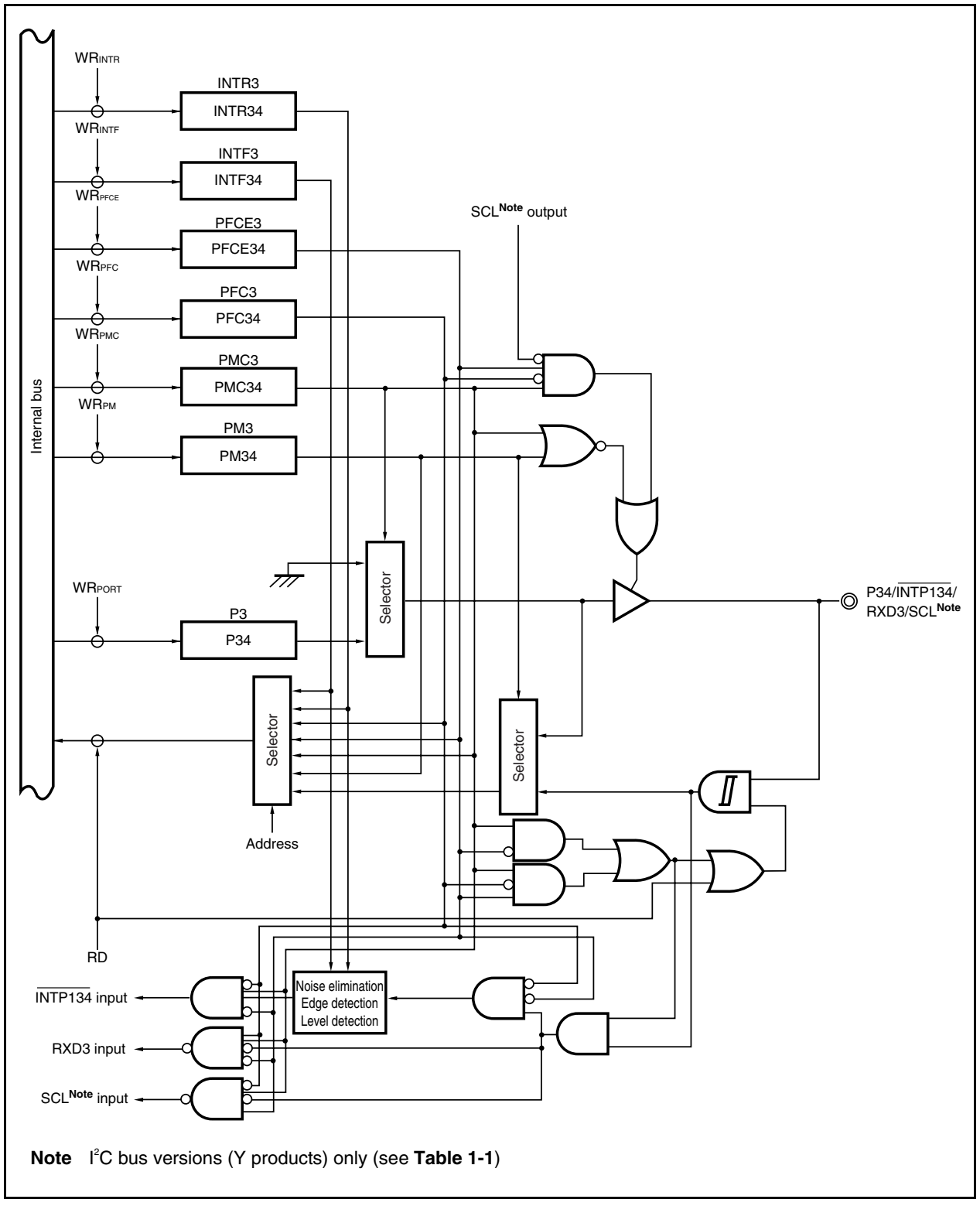


Figure 4-18. Block Diagram of P33 Pin



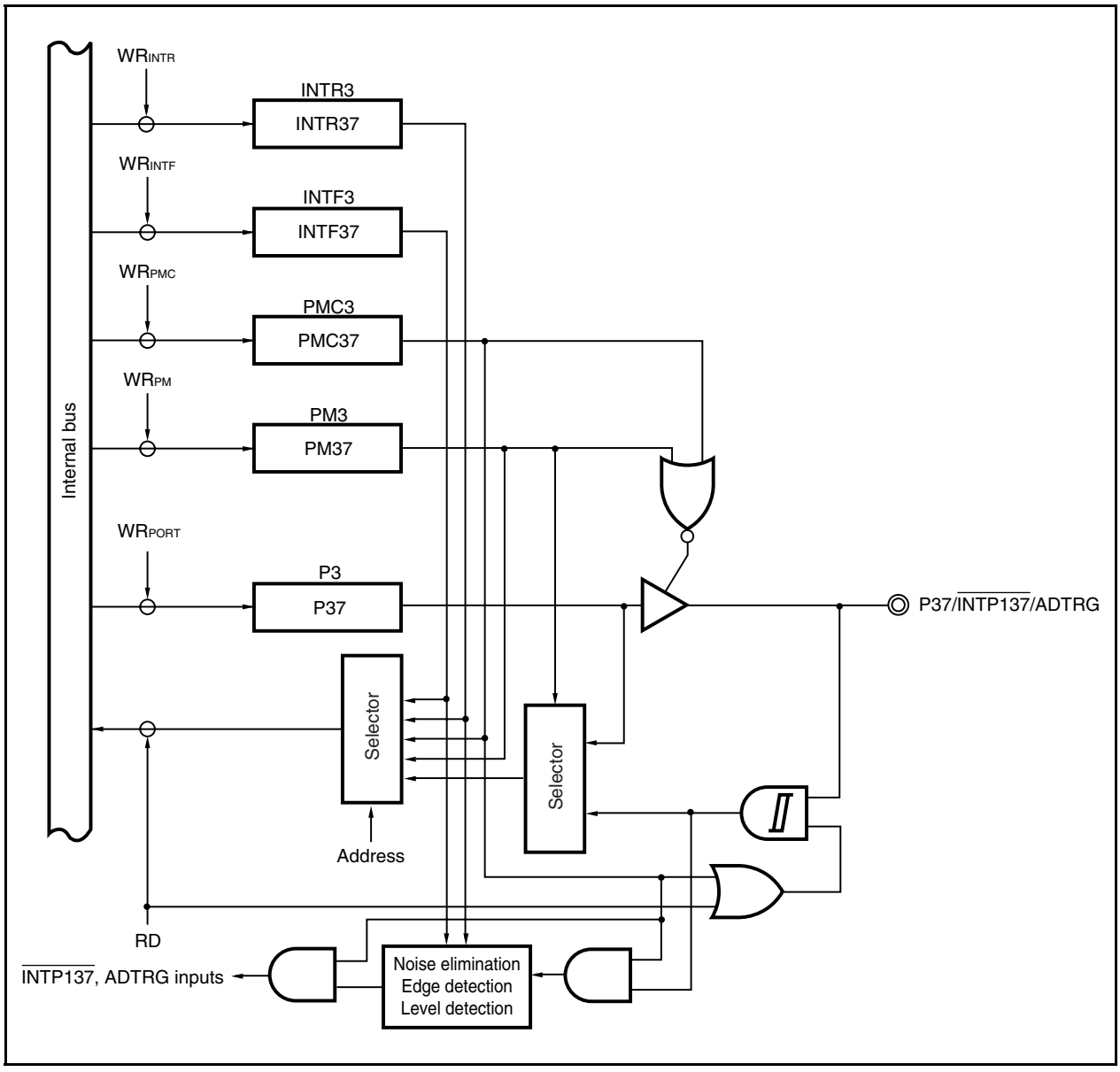
Note I^2C bus versions (Y products) only (see **Table 1-1**)

Figure 4-19. Block Diagram of P34 Pin



Note I^2C bus versions (Y products) only (see **Table 1-1**)

Figure 4-20. Block Diagram of P37 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 4-7. Alternate-Function Pins of Port 4

Pin Name	Pin No.		Alternate-Function Pin	I/O
	GJ	F1		
P40	55	L8	SO0/TXD0	I/O
P41	54	P7	SI0/RXD0	
P42	53	N7	SCK0/ASCK0	
P43	52	M7	SO1/TXD1	
P44	51	P6	SI1/RXD1	
P45	50	L7	SCK1/ASCK1	

Caution P41, P42, P44, and P45 have hysteresis characteristics when their alternate function is used in the input mode, but not when they are used in the port mode.

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(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	P45	P44	P43	P42	P41	P40

P4n	Control of output data (in output mode) (n = 0 to 5)
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	PM45	PM44	PM43	PM42	PM41	PM40

PM4n	Control of input/output mode (in port mode) (n = 0 to 5)
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	PMC45	PMC44	PMC43	PMC42	PMC41	PMC40

PMC45	Specification of operation mode of P45 pin
0	I/O port
1	SCK1 I/O/ASCK1 input

PMC44	Specification of operation mode of P44 pin
0	I/O port
1	SI1 input/RXD1 input

PMC43	Specification of operation mode of P43 pin
0	I/O port
1	SO1 output/TXD1 output/

PMC42	Specification of operation mode of P42 pin
0	I/O port
1	SCK0 I/O/ASCK0 input

PMC41	Specification of operation mode of P41 pin
0	I/O port
1	SI0 input/RXD0 input

PMC40	Specification of operation mode of P40 pin
0	I/O port
1	SO0 output/TXD0 output

(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	PFC45	PFC44	PFC43	PFC42	PFC41	PFC40

PFC45	Specification of alternate function of P45 pin
0	SCK1 I/O
1	ASCK1 input

PFC44	Specification of alternate function of P44 pin
0	SI1 input
1	RXD1 input

PFC43	Specification of alternate function of P43 pin
0	SO1 output
1	TXD1 output

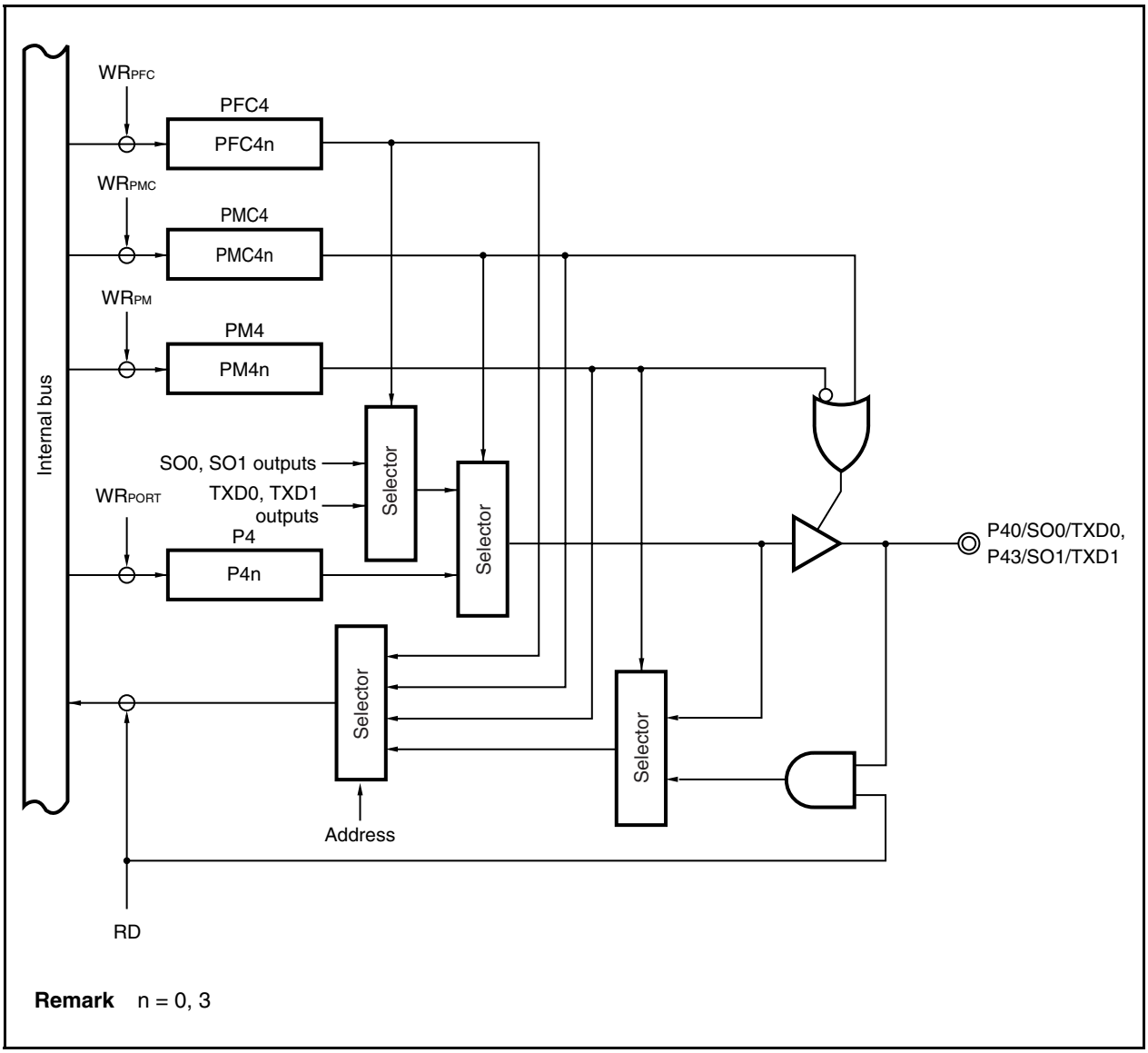
PFC42	Specification of alternate function of P42 pin
0	SCK0 I/O
1	ASCK0 input

PFC41	Specification of alternate function of P41 pin
0	SI0 input
1	RXD0 input

PFC40	Specification of alternate function of P40 pin
0	SO0 output
1	TXD0 output

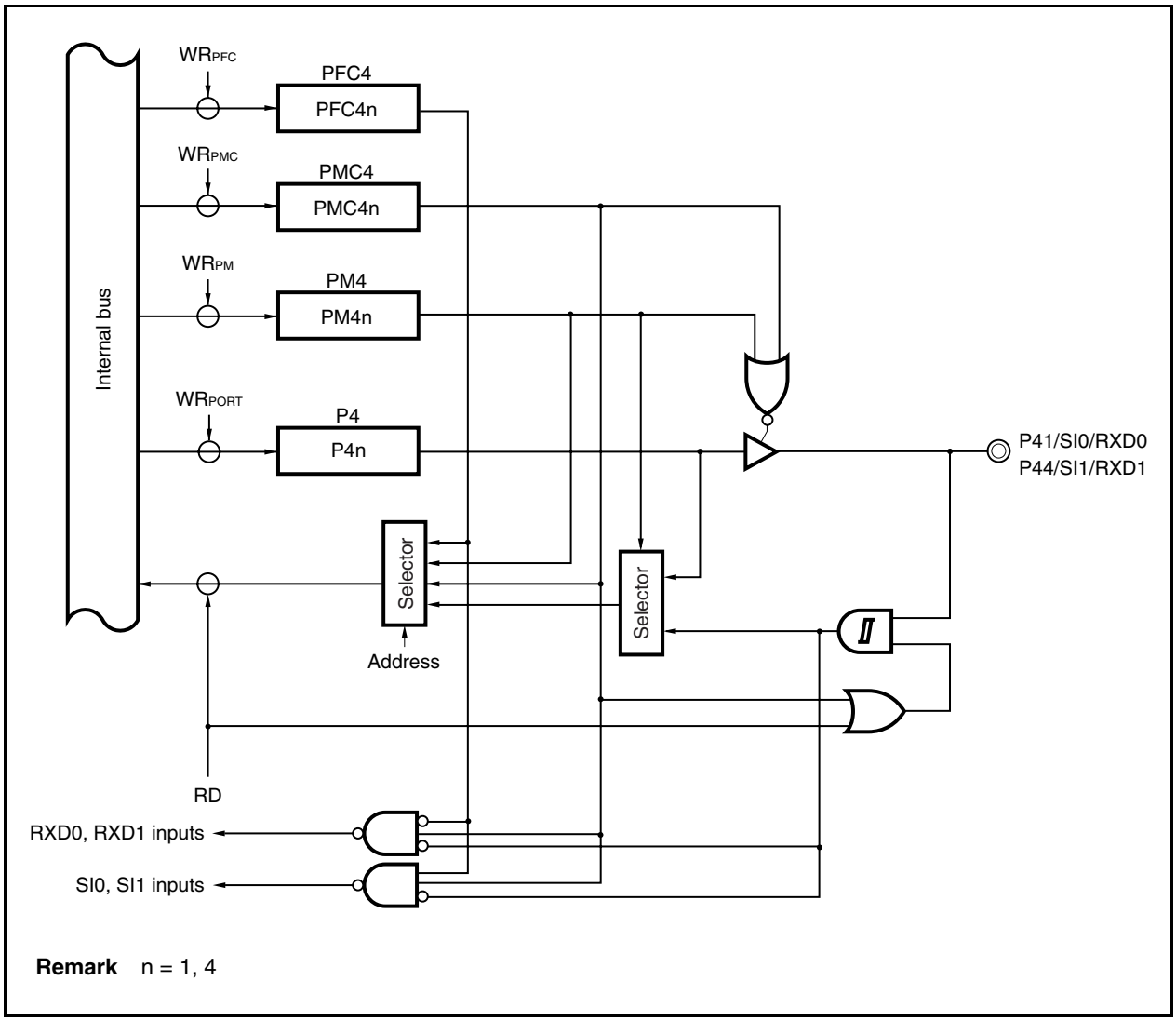
(2) Block diagram

Figure 4-21. Block Diagram of P40 and P43 Pins



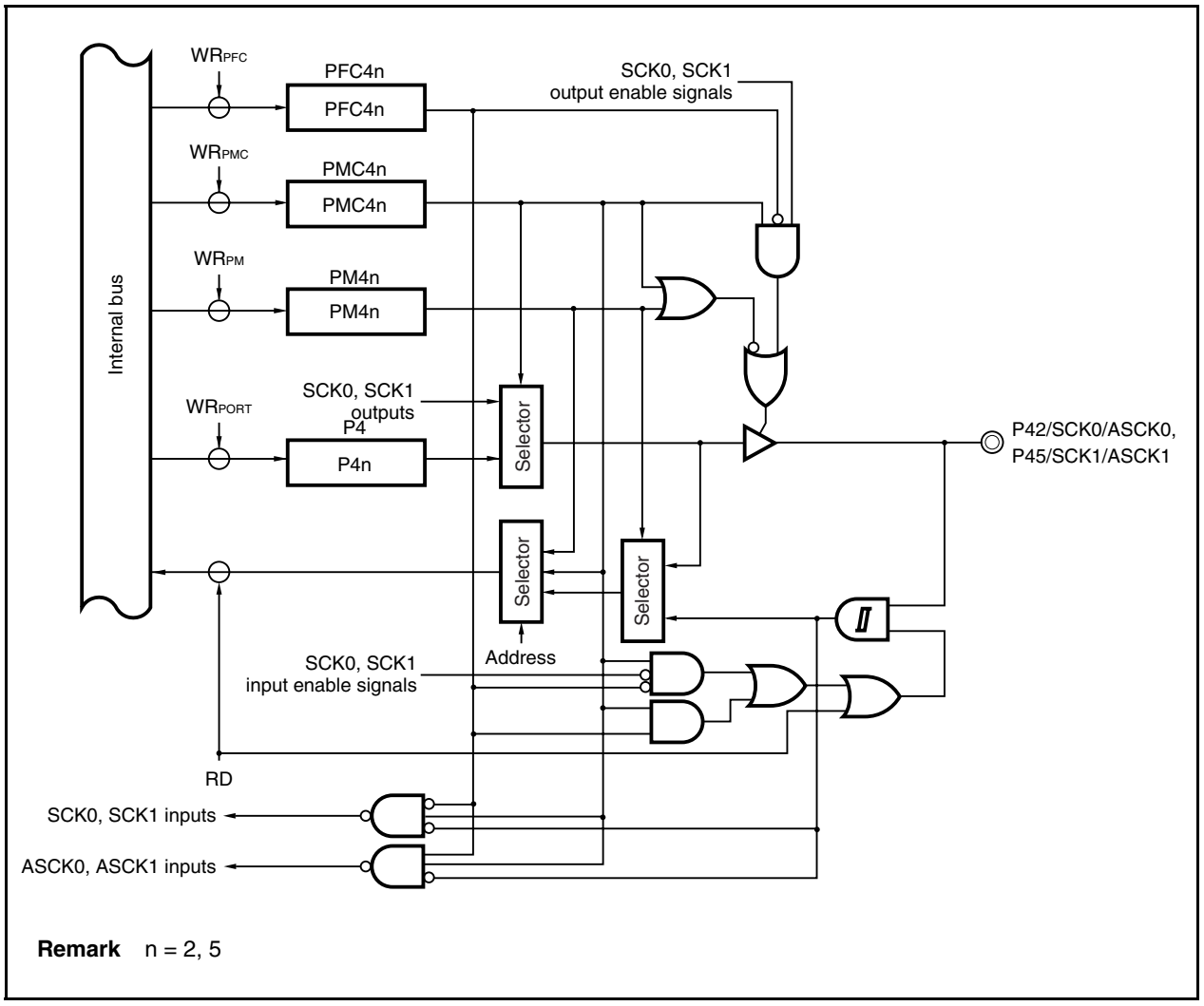
Remark n = 0, 3

Figure 4-22. Block Diagram of P41 and P44 Pins



Remark n = 1, 4

Figure 4-23. Block Diagram of P42 and P45 Pins



Remark n = 2, 5

4.3.6 Port 5

Port 5 can be set to the input or output mode in 1-bit units.

Port 5 has an alternate function as the following pins.

Table 4-8. Alternate-Function Pins of Port 5

Pin Name	Pin No.		Alternate-Function Pin	I/O
	GJ	F1		
P50	86	J13	$\overline{\text{INTP050}}$/TOP20/EVTP2/TIP2/INTPP20	I/O
P51	85	J12	$\overline{\text{INTP051}}$/TOP21/INTPP21	

Caution P50 and P51 have hysteresis characteristics when their alternate function is used in the input mode, but not when they are used in the port mode.

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(1) Registers

(a) Port 5 register (P5)

After reset: Undefined R/W Address: FFFFF40AH

	7	6	5	4	3	2	1	0
P5	0	0	0	0	0	0	P51	P50

P5n	Control of output data (in output mode) (n = 0, 1)
0	Output 0.
1	Output 1.

(b) Port 5 mode register (PM5)

After reset: FFH R/W Address: FFFFF42AH

	7	6	5	4	3	2	1	0
PM5	1	1	1	1	1	1	PM51	PM50

PM5n	Control of input/output mode (in port mode) ((n = 0, 1)
0	Output mode
1	Input mode

(c) Port 5 mode control register (PMC5)

After reset: 00H R/W Address: FFFFF44AH

	7	6	5	4	3	2	1	0
PMC5	0	0	0	0	0	0	PMC51	PMC50

PMC51	Specification of operation mode of P51 pin
0	I/O port
1	$\overline{\text{INTP051}}$ input/TOP21 output/INTPP21 input

PMC50	Specification of operation mode of P51 pin
0	I/O port
1	$\overline{\text{INTP050}}$ input/TOP20 output/EVTP2 input/TIP2 input/INTPP20 input

(d) Port 5 function control expansion register (PFCE5)

After reset: 00H R/W Address: FFFFF70AH

	7	6	5	4	3	2	1	0
PFCE5	0	0	0	0	0	0	PFCE51	PFCE50

Remark For the specification of the alternate function, see **4.3.6 (1) (f) Setting of alternate functions of port 5 pins**.

(e) Port 5 function control register (PFC5)

After reset: 00H R/W Address: FFFFF46AH

	7	6	5	4	3	2	1	0
PFC5	0	0	0	0	0	0	PFC51	PFC50

Remark For the specification of the alternate function, see **4.3.6 (1) (f) Setting of alternate functions of port 5 pins**.

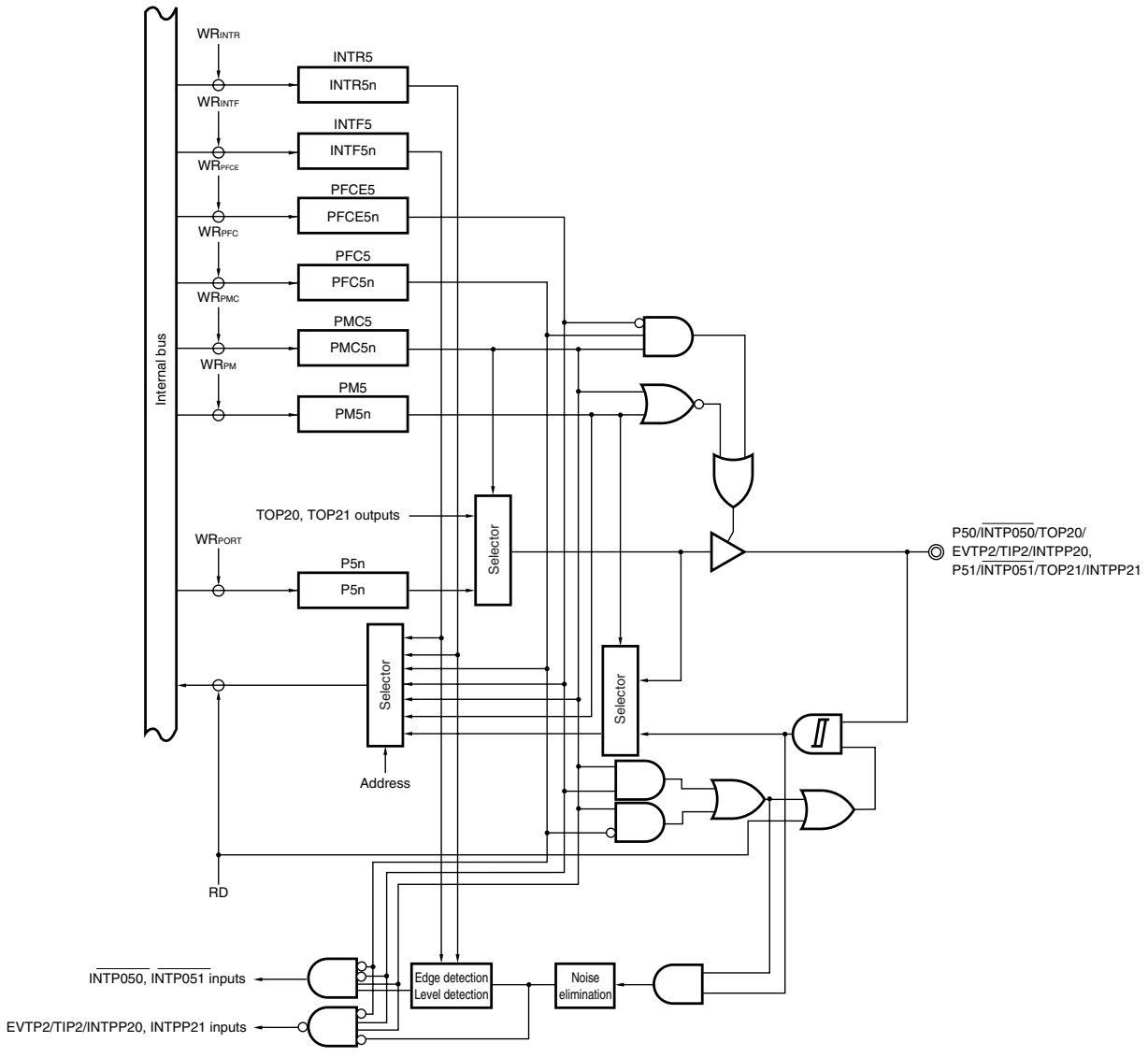
(f) Setting of alternate functions of port 5 pins

PFCE51	PFC51	Specification of Alternate Function of P51
0	0	$\overline{\text{INTP051}}$ input
0	1	TOP21 output
1	0	INTPP21 input
1	1	Setting prohibited

PFCE50	PFC50	Specification of Alternate Function of P50
0	0	$\overline{\text{INTP050}}$ input
0	1	TOP20 output
1	0	EVTP2 input/TIP2 input/INTPP20 input
1	1	Setting prohibited

(2) Block diagram

Figure 4-24. Block Diagram of P50 and P51 Pins



Remark n = 0, 1

4.3.7 Port 7

All of the port 7 pins are fixed to the input mode.

Port 7 has an alternate function as the following pins.

Table 4-9. Alternate-Function Pins of Port 7

Pin Name	Pin No.		Alternate-Function Pin	I/O
	GJ	F1		
P70	80	K12	ANI0	Input
P71	79	K11	ANI1	
P72	78	L14	ANI2	
P73	77	L13	ANI3	
P74	76	L12	ANI4	
P75	75	M13	ANI5	
P76	74	M12	ANI6	
P77	73	L11	ANI7	

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(1) Registers

(a) Port 7 register (P7)

After reset: Undefined R Address: FFFFF40EH

	7	6	5	4	3	2	1	0
P7	P77	P76	P75	P74	P73	P72	P71	P70

P7n	Input data read (n = 0 to 7)
0	Input 0.
1	Input 1.

Caution Do not read the P7 register during A/D conversion.

(b) Port 7 mode control register (PMC7)

After reset: 00H R/W Address: FFFFF44EH

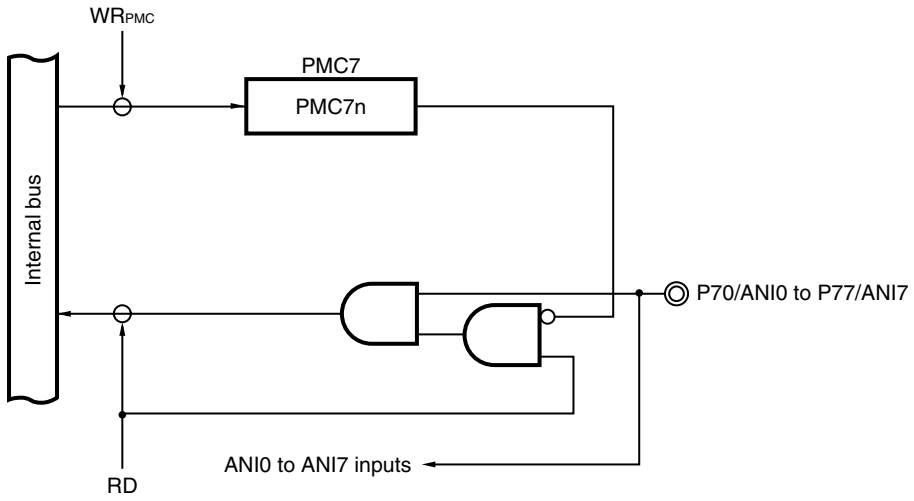
	7	6	5	4	3	2	1	0
PMC7	PMC77	PMC76	PMC75	PMC74	PMC73	PMC72	PMC71	PMC70

PMC7n	Specification of operation mode of P7n pin (n = 0 to 7)
0	Input port
1	ANIn input

Caution Do not change to the port mode during A/D conversion.

(2) Block diagram

Figure 4-25. Block Diagram of P70 to P77 Pins



Remark n = 0 to 7

4.3.8 Port 8

Both the port 8 pins are fixed to the input mode.

Port 8 has an alternate function as the following pins.

Table 4-10. Alternate-Function Pins of Port 8

Pin Name	Pin No.		Alternate-Function Pin	I/O
	GJ	F1		
P80	69	P13	ANO0	Input
P81	68	N11	ANO1	

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)

F1: 161-pin plastic FBGA (13 × 13)

(1) Registers

(a) Port 8 register (P8)

After reset: Undefined R Address: FFFFF410H

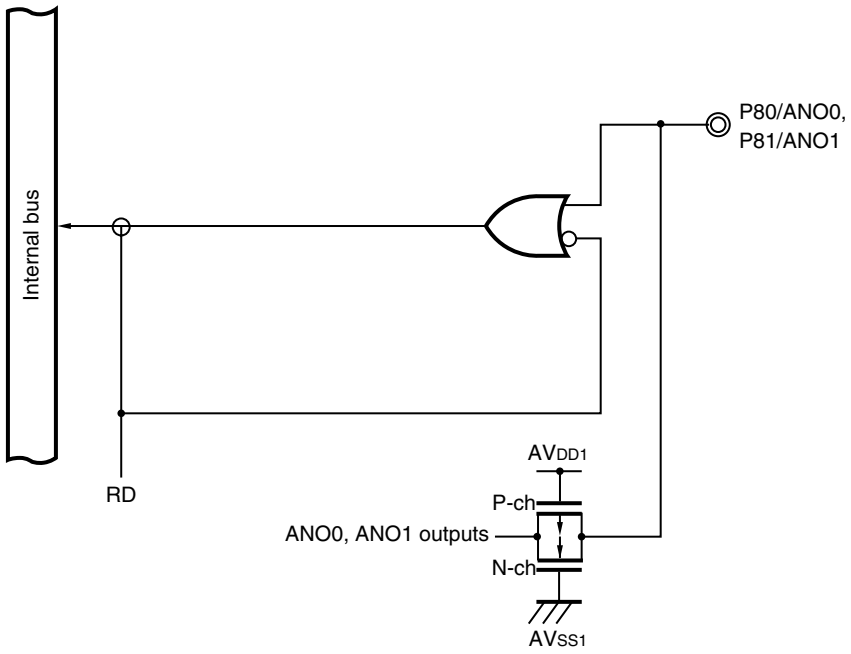
	7	6	5	4	3	2	1	0
P8	0	0	0	0	0	0	P81	P80

P8n	Input data read (n = 0, 1)
0	Input 0.
1	Input 1.

Caution Do not read the P8 register during D/A conversion. When using port 8 as an input port, be sure to stop the D/A conversion operation.

(2) Block diagram

Figure 4-26. Block Diagram of P80 and P81 Pins



Remark n = 0, 1

4.3.9 Port AL

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

Port AL has an alternate function as the following pins.

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

Pin Name	Pin No.		Alternate-Function Pin	I/O
	GJ	F1		
PAL0	143	B3	A0	I/O
PAL1	142	C4	A1	
PAL2	141	A3	A2	
PAL3	140	D4	A3	
PAL4	139	B4	A4	
PAL5	138	A4	A5	
PAL6	137	D5	A6	
PAL7	136	C5	A7	
PAL8	133	B6	A8	
PAL9	132	A6	A9	
PAL10	131	D6	A10	
PAL11	130	C7	A11	
PAL12	129	A7	A12	
PAL13	128	B7	A13	
PAL14	127	D7	A14	
PAL15	126	A8	A15	

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)

F1: 161-pin plastic FBGA (13 × 13)

(1) Registers

(a) Port AL register (PAL)

After reset: Undefined R/W Address: PAL FFFFF000H,
PALL FFFFF000H, PALH FFFFF001H

	15	14	13	12	11	10	9	8
PAL (PALH[Note])	PAL15	PAL14	PAL13	PAL12	PAL11	PAL10	PAL9	PAL8

	7	6	5	4	3	2	1	0
(PALL)	PAL7	PAL6	PAL5	PAL4	PAL3	PAL2	PAL1	PAL0

PALn	Control of output data (in output mode) (n = 0 to 15)
0	Output 0.
1	Output 1.

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

Remark The PAL register can be read or written in 16-bit units.
When the higher 8 bits of the PAL register are used as the PALH register, and the lower 8 bits, as the PALL register, these registers can be read or written in 8-bit or 1-bit units.

(b) Port AL mode register (PMAL)

After reset: FFFFH R/W Address: PMAL FFFFF020H,
PMALL FFFFF020H, PMALH FFFFF021H

	15	14	13	12	11	10	9	8
PMAL (PMALH[Note])	PMAL15	PMAL14	PMAL13	PMAL12	PMAL11	PMAL10	PMAL9	PMAL8

	7	6	5	4	3	2	1	0
(PMALL)	PMAL7	PMAL6	PMAL5	PMAL4	PMAL3	PMAL2	PMAL1	PMAL0

PMALn	Specification of input/output mode (in port mode) (n = 0 to 15)
0	Output mode
1	Input mode

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

Remark The PMAL register can be read or written in 16-bit units.
When the higher 8 bits of the PMAL register are used as the PMALH register, and the lower 8 bits, as the PMALL register, these registers can be read or written in 8-bit or 1-bit units.

(c) Port AL mode control register (PMCAL)

After reset: 0000H R/W Address: PMCAL FFFFF040H,
PMCALL FFFFF040H, PMCALH FFFFF041H

	15	14	13	12	11	10	9	8
PMCAL (PMCALH[Note])	PMCAL15	PMCAL14	PMCAL13	PMCAL12	PMCAL11	PMCAL10	PMCAL9	PMCAL8

	7	6	5	4	3	2	1	0
(PMCALL)	PMCAL7	PMCAL6	PMCAL5	PMCAL4	PMCAL3	PMCAL2	PMCAL1	PMCAL0

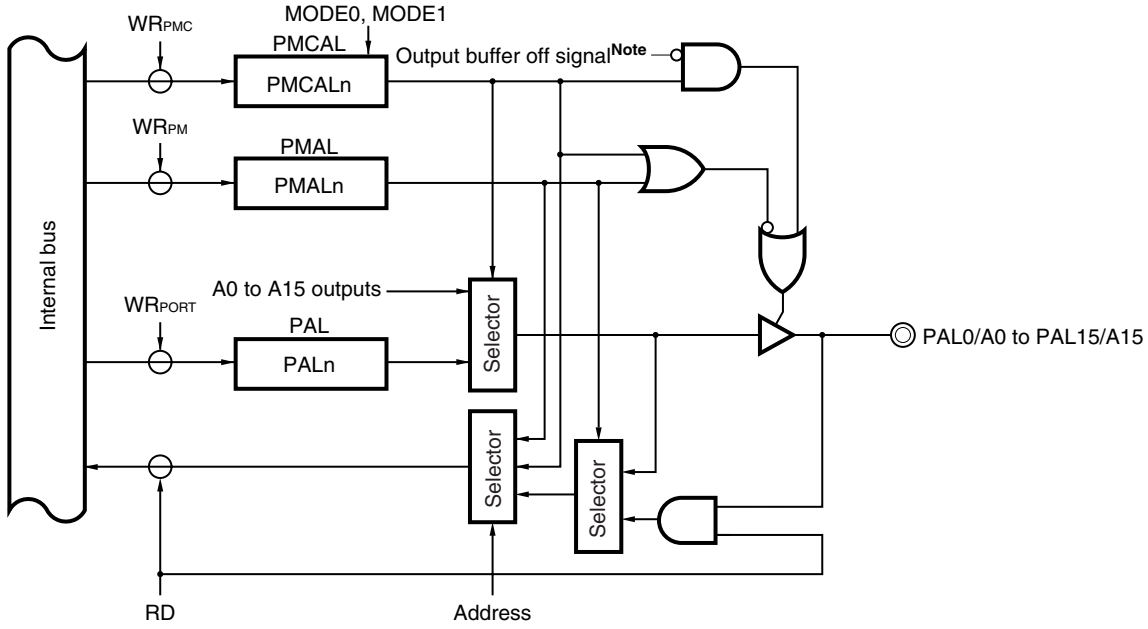
PMCALn	Specification of operation mode of PALn pin (n = 0 to 15)
0	I/O port
1	An output

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

Remark The PMCAL register can be read or written in 16-bit units.
When the higher 8 bits of the PMCAL register are used as the PMCALH register, and the lower 8 bits, as the PMCALL register, these registers can be read or written in 8-bit or 1-bit units.

(2) Block diagram

Figure 4-27. Block Diagram of PAL0 to PAL15 Pins



Note Signal that is active in the IDLE and software STOP modes, or during bus hold.

Remark n = 0 to 15

4.3.10 Port AH

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

Port AH has an alternate function as the following pins.

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

Pin Name	Pin No.		Alternate-Function Pin	I/O
	GJ	F1		
PAH0	123	D8	A16	I/O
PAH1	122	A9	A17	
PAH2	121	B9	A18	
PAH3	120	C9	A19	
PAH4	119	D9	A20	
PAH5	118	B10	A21	
PAH6	117	C10	A22	
PAH7	116	D10	A23	
PAH8	115	A11	A24	
PAH9	114	B11	A25	

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(1) Registers

(a) Port AH register (PAH)

After reset: Undefined R/W Address: PAH FFFFF002H,
PAHL FFFFF002H, PAHH FFFFF003H

	15	14	13	12	11	10	9	8
PAH (PAHH[Note])	0	0	0	0	0	0	PAH9	PAH8

	7	6	5	4	3	2	1	0
(PAHL)	PAH7	PAH6	PAH5	PAH4	PAH3	PAH2	PAH1	PAH0

PAHn	Control of output data (in output mode) (n = 0 to 9)
0	Output 0.
1	Output 1.

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

Remark The PAH register can be read or written in 16-bit units.
When the higher 8 bits of the PAH register are used as the PAHH register, and the lower 8 bits, as the PAHL register, these registers can be read or written in 8-bit or 1-bit units.

(b) Port AH mode register (PMAH)

After reset: FFFFH R/W Address: PMAH FFFFF022H,
PMAHL FFFFF022H, PMAHH FFFFF023H

	15	14	13	12	11	10	9	8
PMAH (PMAHH[Note])	1	1	1	1	1	1	PMAH9	PMAH8

	7	6	5	4	3	2	1	0
(PMAHL)	PMAH7	PMAH6	PMAH5	PMAH4	PMAH3	PMAH2	PMAH1	PMAH0

PMAHn	Control of input/output mode (in port mode) (n = 0 to 9)
0	Output mode
1	Input mode

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

Remark The PMAH register can be read or written in 16-bit units.
When the higher 8 bits of the PMAH register are used as the PMAHH register, and the lower 8 bits, as the PMAHL register, these registers can be read or written in 8-bit or 1-bit units.

(c) Port AH mode control register (PMCAH)

After reset: 0000H R/W Address: PMCAH FFFFF042H,
PMCAHL FFFFF042H, PMCAHH FFFFF043H

	15	14	13	12	11	10	9	8
PMCAH (PMCAHH[Note])	0	0	0	0	0	0	PMCAH9	PMCAH8

	7	6	5	4	3	2	1	0
(PMCAHL)	PMCAH7	PMCAH6	PMCAH5	PMCAH4	PMCAH3	PMCAH2	PMCAH1	PMCAH0

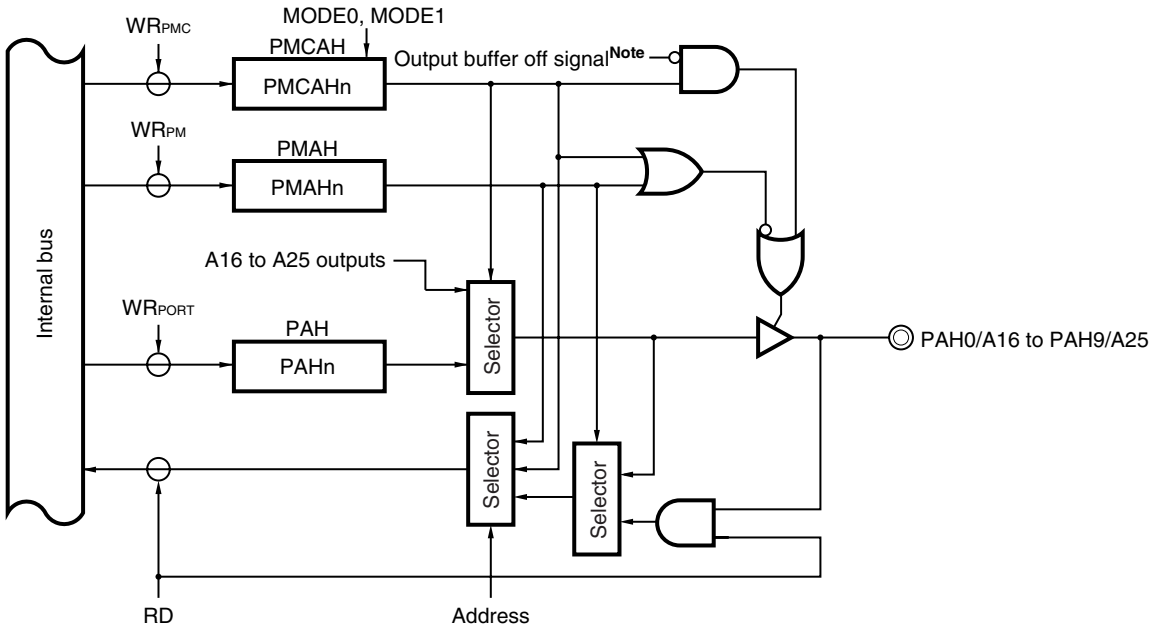
PMCAHn	Specification of operation mode of PAHn pin (n = 0 to 9)
0	I/O port
1	Am output (m = 16 to 25)

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

Remark The PMCAH register can be read or written in 16-bit units.
When the higher 8 bits of the PMCAH register are used as the PMCAHH register, and the lower 8 bits, as the PMCAHL register, these registers can be read or written in 8-bit or 1-bit units.

(2) Block diagram

Figure 4-28. Block Diagram of PAH0 to PAH9 Pins



Note Signal that is active in the IDLE and software STOP modes, or during bus hold.

Remark n = 0 to 9

4.3.11 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 4-13. Alternate-Function Pins of Port DL

Pin Name	Pin No.		Alternate-Function Pin	I/O
	GJ	F1		
PDL0	17	G3	AD0	I/O
PDL1	16	H4	AD1	
PDL2	15	F1	AD2	
PDL3	14	F2	AD3	
PDL4	13	F3	AD4	
PDL5	12	E1	AD5	
PDL6	11	G4	AD6	
PDL7	10	E2	AD7	
PDL8	7	E3	AD8	
PDL9	6	C2	AD9	
PDL10	5	D2	AD10	
PDL11	4	E4	AD11	
PDL12	3	B2	AD12	
PDL13	2	C3	AD13	
PDL14	1	D3	AD14	
PDL15	144	A2	AD15	

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(1) Registers

(a) Port DL register (PDL)

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

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

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

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

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

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

(b) Port DL mode register (PMDL)

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

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

	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 15)
0	Output mode
1	Input mode

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

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

(c) Port DL mode control register (PMCDL)

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

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

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

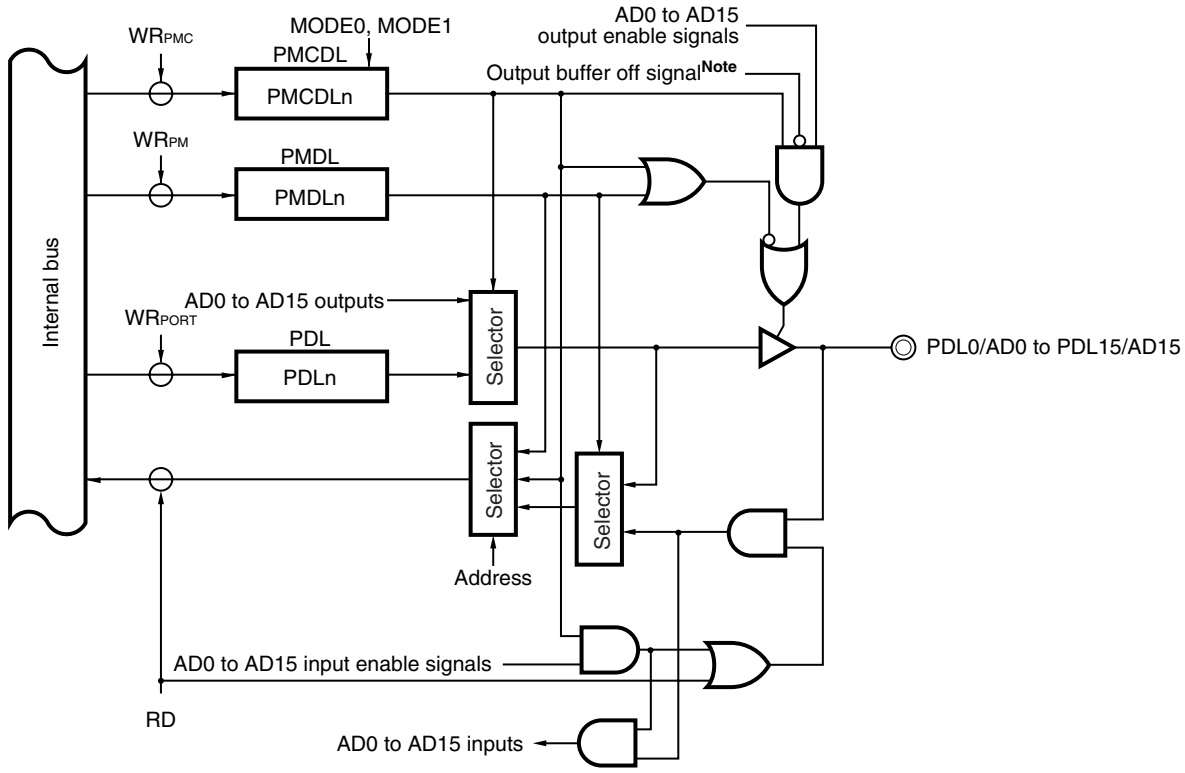
PMCDLn	Specification of operation mode of PDLn pin (n = 0 to 15)
0	I/O port
1	ADn I/O

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

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

(2) Block diagram

Figure 4-29. Block Diagram of PDL0 to PDL15 Pins



Note Signal that is active in the IDLE and software STOP modes, or during bus hold.

Remark n = 0 to 15

4.3.12 Port CS

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

Port CS has an alternate function as the following pins.

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

Pin Name	Pin No.		Alternate-Function Pin	I/O
	GJ	F1		
PCS0	107	D12	$\overline{CS0}$	I/O
PCS1	106	B13	$\overline{CS1}$	
PCS2	105	C13	$\overline{CS2}$/$\overline{IOWR}$	
PCS3	104	C12	$\overline{CS3}$	
PCS4	103	E12	$\overline{CS4}$	
PCS5	102	D13	$\overline{CS5}$/$\overline{IORD}$	
PCS6	101	E11	$\overline{CS6}$	
PCS7	100	E13	$\overline{CS7}$	

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(1) Registers

(a) Port CS register (PCS)

After reset: Undefined R/W Address: FFFFF008H

	7	6	5	4	3	2	1	0
PCS	PCS7	PCS6	PCS5	PCS4	PCS3	PCS2	PCS1	PCS0

PCSn	Control of output data (in output mode) (n = 0 to 7)
0	Output 0.
1	Output 1.

(b) Port CS mode register (PMCS)

After reset: FFH R/W Address: FFFFF028H

	7	6	5	4	3	2	1	0
PMCS	PMCS7	PMCS6	PMCS5	PMCS4	PMCS3	PMCS2	PMCS1	PMCS0

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

(c) Port CS mode control register (PMCCS)

After reset: 00H R/W Address: FFFFF048H

	7	6	5	4	3	2	1	0
PMCCS	PMCCS7	PMCCS6	PMCCS5	PMCCS4	PMCCS3	PMCCS2	PMCCS1	PMCCS0

PMCCS7	Specification of operation mode of PCS7 pin
0	I/O port
1	$\overline{CS7}$ output

PMCCS6	Specification of operation mode of PCS6 pin
0	I/O port
1	$\overline{CS6}$ output

PMCCS5	Specification of operation mode of PCS5 pin
0	I/O port
1	$\overline{CS5}$ output/$\overline{IORD}$ output

PMCCS4	Specification of operation mode of PCS4 pin
0	I/O port
1	$\overline{CS4}$ output

PMCCS3	Specification of operation mode of PCS3 pin
0	I/O port
1	$\overline{CS3}$ output

PMCCS2	Specification of operation mode of PCS2 pin
0	I/O port
1	$\overline{CS2}$ output/$\overline{IOWR}$ output

PMCCS1	Specification of operation mode of PCS1 pin
0	I/O port
1	$\overline{CS1}$ output

PMCCS0	Specification of operation mode of PCS0 pin
0	I/O port
1	$\overline{CS0}$ output

(d) Port CS function control register (PFCCS)

After reset: 00H R/W Address: FFFFF049H

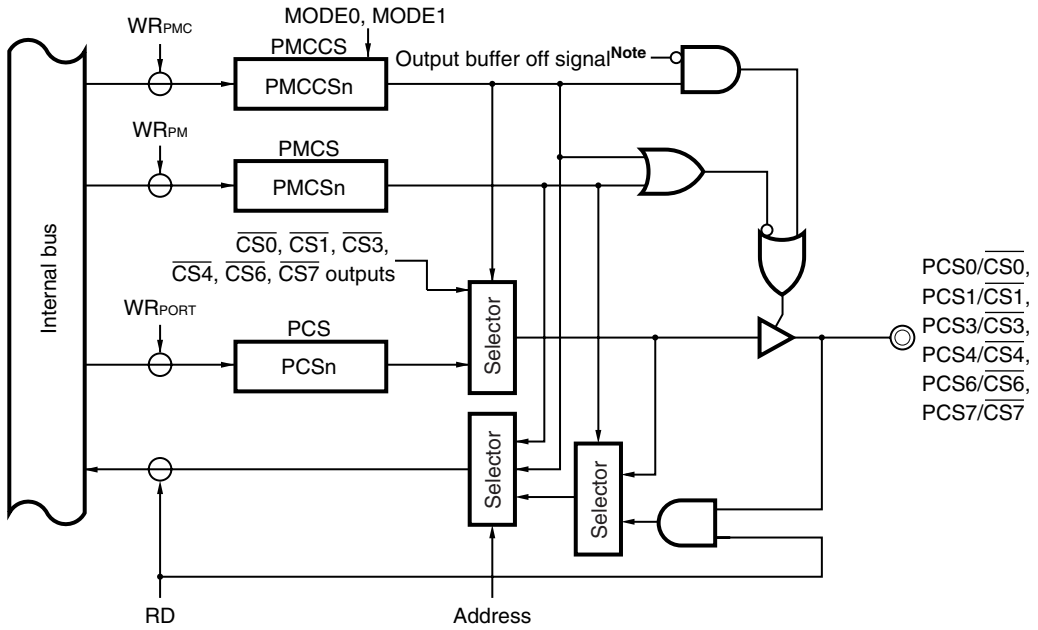
	7	6	5	4	3	2	1	0
PFCCS	0	0	PFCCS5	0	0	PFCCS2	0	0

PFCCS5	Specification of alternate function of PCS5 pin
0	$\overline{CS5}$ output
1	$\overline{IORD}$ output

PFCCS2	Specification of alternate function of PCS2 pin
0	$\overline{CS2}$ output
1	$\overline{IOWR}$ output

(2) Block diagram

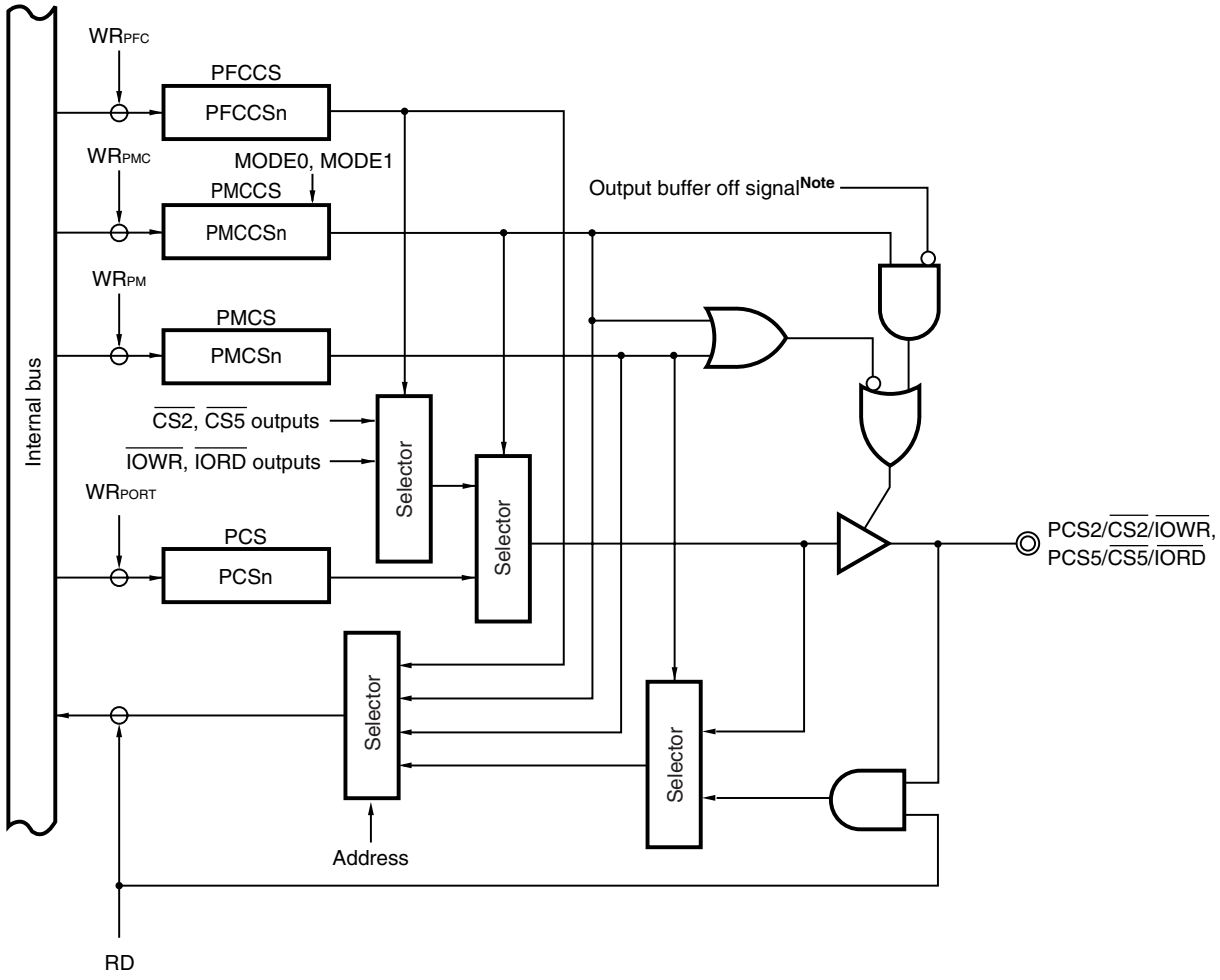
Figure 4-30. Block Diagram of PCS0, PCS1, PCS3, PCS4, PCS6, and PCS7 Pins



Note Signal that is active during bus hold.

Remark n = 0, 1, 3, 4, 6, 7

Figure 4-31. Block Diagram of PCS2 and PCS5 Pins



Note Signal that is active during bus hold.

Remark n = 2, 5

4.3.13 Port CT

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

Port CT has an alternate function as the following pins.

Table 4-15 Alternate-Function Pins of Port CT

Pin Name	Pin No.		Alternate-Function Pin	I/O
	GJ	F1		
PCT0	97	F13	$\overline{\text{LWR}}$/LDQM/$\overline{\text{LBE}}$	I/O
PCT1	96	F14	$\overline{\text{UWR}}$/UDQM/$\overline{\text{UBE}}$	
PCT4	95	F11	$\overline{\text{RD}}$	
PCT5	94	G12	$\overline{\text{WR}}$/$\overline{\text{WE}}$	
PCT6	93	G14	ASTB	
PCT7	92	G13	$\overline{\text{BCYST}}$	

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(1) Registers

(a) Port CT register (PCT)

After reset: Undefined R/W Address: FFFFF00AH

	7	6	5	4	3	2	1	0
PCT	PCT7	PCT6	PCT5	PCT4	0	0	PCT1	PCT0

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

(b) Port CT mode register (PMCT)

After reset: FFH R/W Address: FFFFF02AH

	7	6	5	4	3	2	1	0
PMCT	PMCT7	PMCT6	PMCT5	PMCT4	1	1	PMCT1	PMCT0

PMCTn	Control of input/output mode (in port mode) (n = 0, 1, 4 to 7)
0	Output mode
1	Input mode

(c) Port CT mode control register (PMCCT)

After reset: 00H R/W Address: FFFFF04AH

	7	6	5	4	3	2	1	0
PMCCT	PMCCT7	PMCCT6	PMCCT5	PMCCT4	0	0	PMCCT1	PMCCT0

PMCCT7	Specification of operation mode of PCT7 pin
0	I/O port
1	$\overline{\text{BCYST}}$ output

PMCCT6	Specification of operation mode of PCT6 pin
0	I/O port
1	ASTB output

PMCCT5	Specification of operation mode of PCT5 pin
0	I/O port
1	$\overline{\text{WR}}$ output/$\overline{\text{WE}}$ output

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

PMCCT1	Specification of operation mode of PCT1 pin
0	I/O port
1	$\overline{\text{UWR}}$ output/UDQM output/$\overline{\text{UBE}}$ output

PMCCT0	Specification of operation mode of PCT0 pin
0	I/O port
1	$\overline{\text{LWR}}$ output/LDQM output/$\overline{\text{LBE}}$ output

(d) Port CT function control register (PFCCT)

After reset: 00H R/W Address: FFFFF04BH

	7	6	5	4	3	2	1	0
PFCCT	0	0	0	0	0	0	PFCCT1	PFCCT0

PFCCT1	Specification of alternate function of PCT1 pin
0	$\overline{\text{UWR}}$ output/UDQM output[Note 1]
1	$\overline{\text{UBE}}$ output/UDQM output[Note 1]

PFCCT0	Specification of alternate function of PCT0 pin
0	$\overline{\text{LWR}}$ output/LDQM output[Note 2]
1	$\overline{\text{LBE}}$ output/LDQM output[Note 2]

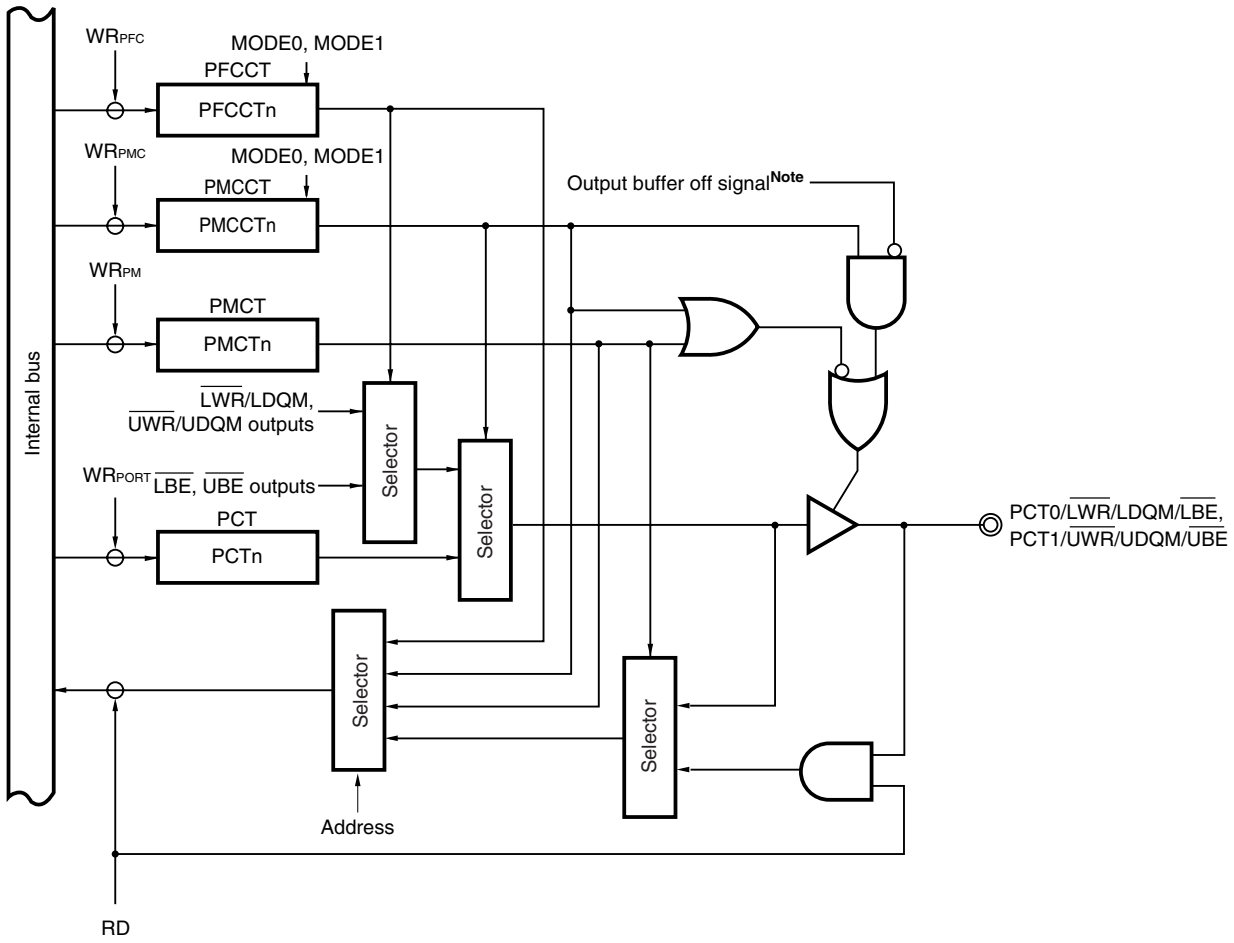
Notes 1. The $\overline{\text{UWR}}$ output or UDQM output, and $\overline{\text{UBE}}$ output or UDQM output are automatically selected when target memory is accessed.

2. The $\overline{\text{LWR}}$ output or LDQM output, and $\overline{\text{LBE}}$ output or LDQM output are automatically selected when the target memory is accessed.

Caution The $\overline{\text{xDQM}}$ signal differs in timing between when $\overline{\text{xWR}}$ output/$\overline{\text{xDQM}}$ output is selected and when $\overline{\text{xBE}}$ output/$\overline{\text{xDQM}}$ output is selected. However, this signal can be connected to SDRAM without problem regardless of which output is selected. For the output timing of the $\overline{\text{xDQM}}$ signal, see timing charts of 6.3.5 SDRAM access (Figure 6-12) (x = U or L).

(2) Block diagram

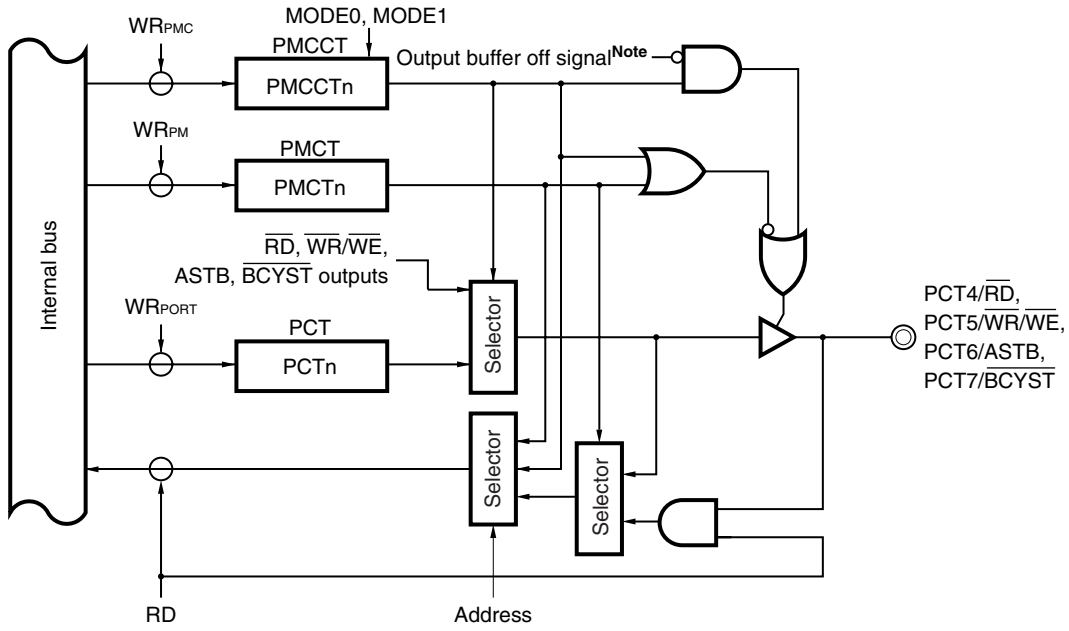
Figure 4-32. Block Diagram of PCT0 and PCT1 Pins



Note Signal that is active during bus hold.

Remark n = 0, 1

Figure 4-33. Block Diagram of PCT4 to PCT7 Pins



Note Signal that is active during bus hold.

Remark n = 4 to 7

4.3.14 Port CM

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

Port CM has an alternate function as the following pins.

Table 4-16. Alternate-Function Pins of Port CM

Pin Name	Pin No.		Alternate-Function Pin	I/O
	GJ	F1		
PCM0	91	G11	$\overline{\text{WAIT}}$	I/O
PCM1	90	H14	BUSCLK	
PCM2	89	H13	$\overline{\text{HLDAK}}$	
PCM3	88	H12	$\overline{\text{HLDRQ}}$	
PCM4	87	H11	$\overline{\text{REFRQ}}$	

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)

F1: 161-pin plastic FBGA (13 × 13)

(1) Registers

(a) Port CM register (PCM)

After reset: Undefined R/W Address: FFFFF00CH

	7	6	5	4	3	2	1	0
PCM	0	0	0	PCM4	PCM3	PCM2	PCM1	PCM0

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

(b) Port CM mode register (PMCM)

After reset: FFH R/W Address: FFFFF02CH

	7	6	5	4	3	2	1	0
PMCM	1	1	1	PMCM4	PMCM3	PMCM2	PMCM1	PMCM0

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

(c) Port CM mode control register (PMCCM)

After reset: 00H R/W Address: FFFFF04CH

	7	6	5	4	3	2	1	0
PMCCM	0	0	0	PMCCM4	PMCCM3	PMCCM2	PMCCM1	PMCCM0

PMCCM4	Specification of operation mode of PCM4 pin
0	I/O port
1	$\overline{\text{REFRQ}}$ output

PMCCM3	Specification of operation mode of PCM3 pin
0	I/O port
1	$\overline{\text{HLDRQ}}$ input

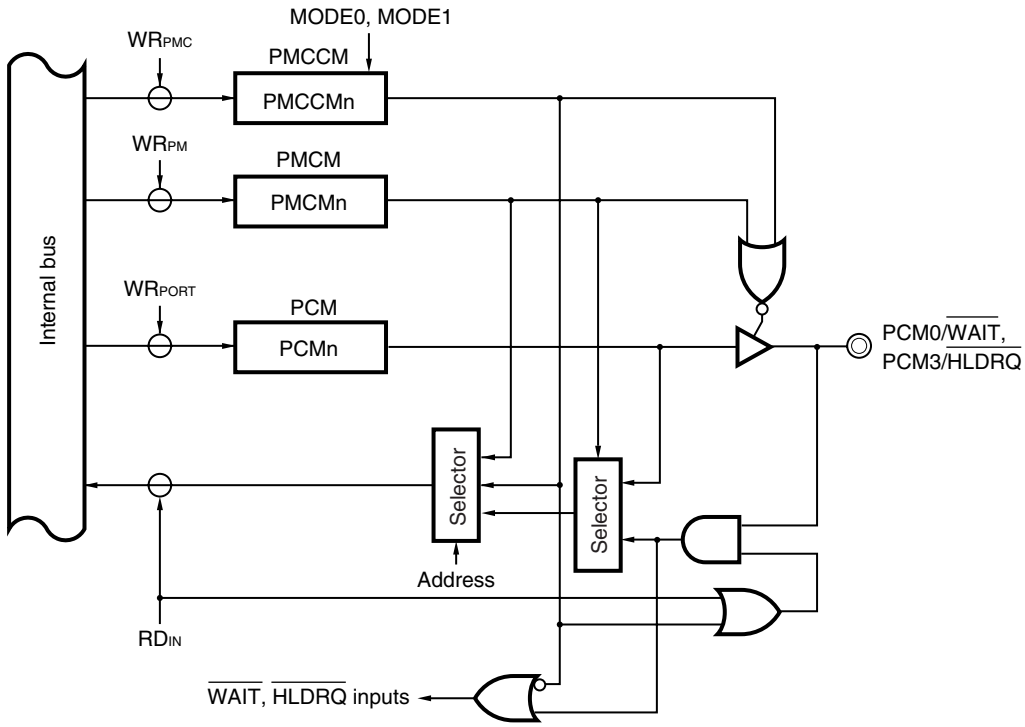
PMCCM2	Specification of operation mode of PCM2 pin
0	I/O port
1	$\overline{\text{HLDAK}}$ output

PMCCM1	Specification of operation mode of PCM1 pin
0	I/O port
1	BUSCLK output

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

(2) Block diagram

Figure 4-34. Block Diagram of PCM0 and PCM3 Pins



Remark n = 0, 3

Figure 4-35. Block Diagram of PCM1 Pin

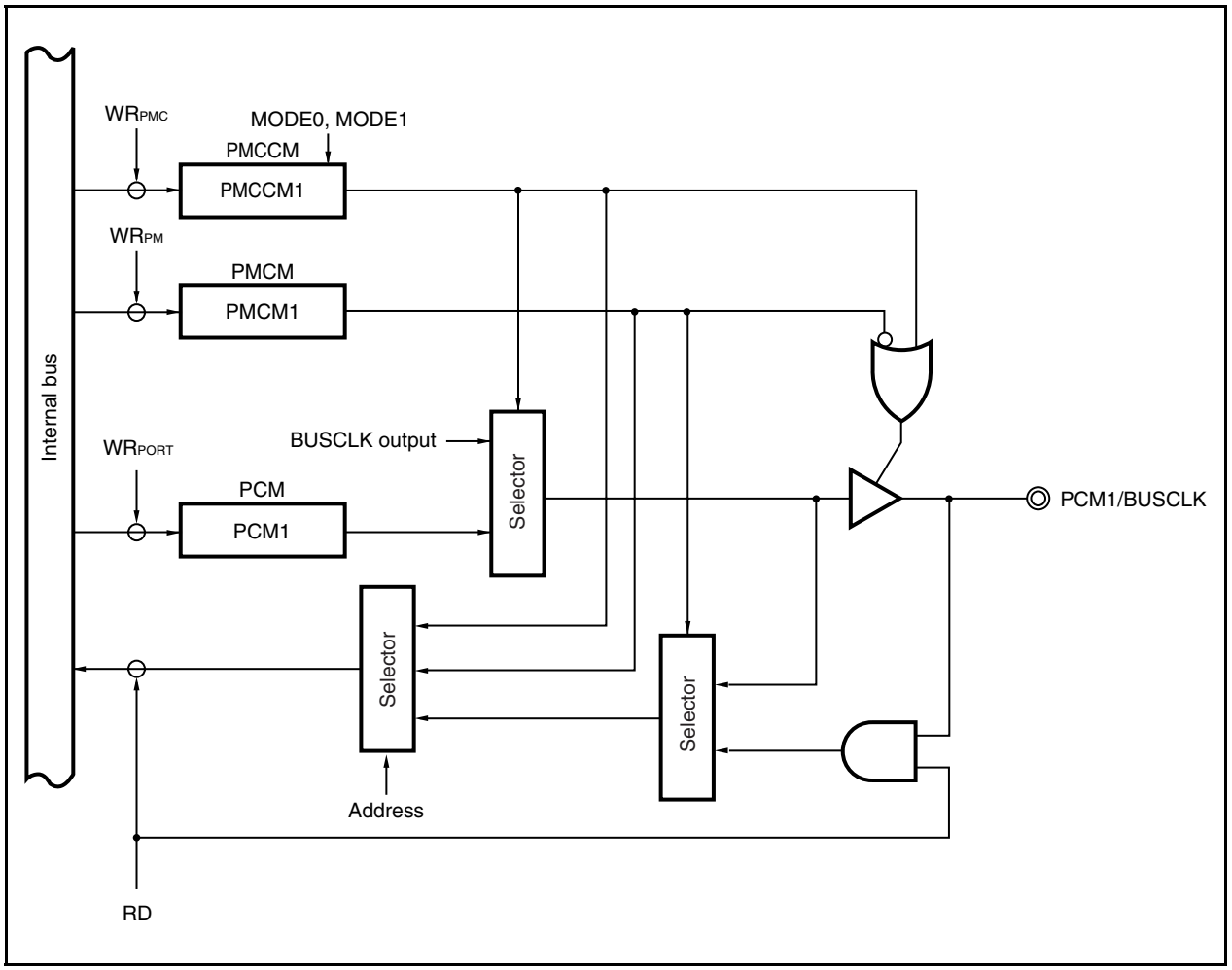
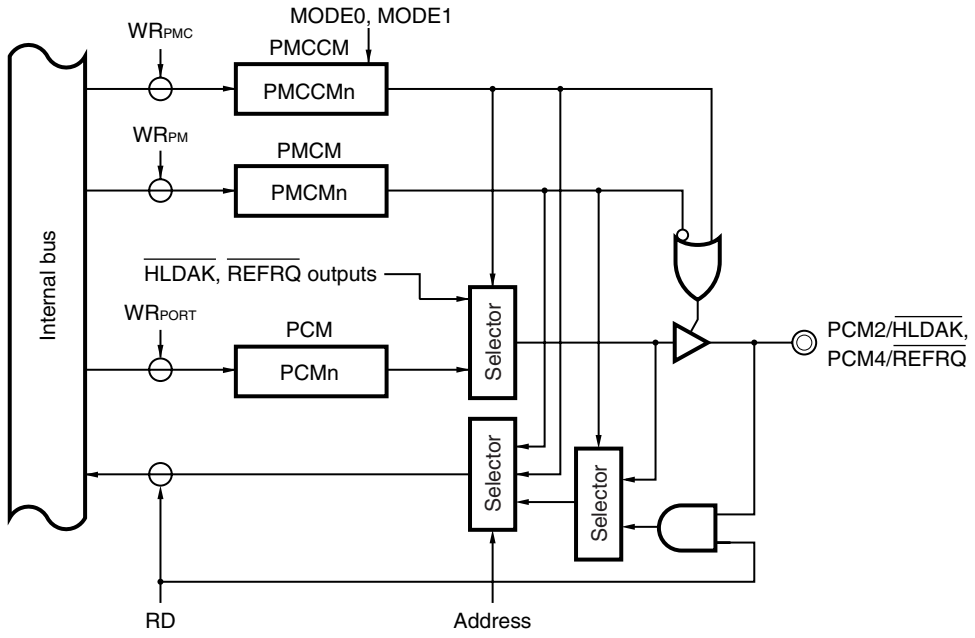


Figure 4-36. Block Diagram of PCM2 and PCM4 Pins



Remark n = 2, 4

4.3.15 Port CD

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

Port CD has an alternate function as the following pins.

Table 4-17. Alternate-Function Pins of Port CD

Pin Name	Pin No.		Alternate-Function Pin	I/O
	GJ	F1		
PCD0	111	D11	SDCKE	I/O
PCD1	110	B12	SDCLK	
PCD2	109	A13	$\overline{\text{SDCAS}}$	
PCD3	108	A14	$\overline{\text{SDRAS}}$	

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(1) Registers

(a) Port CD register (PCD)

After reset: Undefined R/W Address: FFFFF00EH

	7	6	5	4	3	2	1	0
PCD	0	0	0	0	PCD3	PCD2	PCD1	PCD0

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

(b) Port CD mode register (PMCD)

After reset: FFH R/W Address: FFFFF02EH

	7	6	5	4	3	2	1	0
PMCD	1	1	1	1	PMCD3	PMCD2	PMCD1	PMCD0

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

(c) Port CD mode control register (PMCCD)

After reset: 00H R/W Address: FFFFF04EH

	7	6	5	4	3	2	1	0
PMCCD	0	0	0	0	PMCCD3	PMCCD2	PMCCD1	PMCCD0

PMCCD3	Specification of operation mode of PCD3 pin
0	I/O port
1	$\overline{\text{SDRAS}}$ output

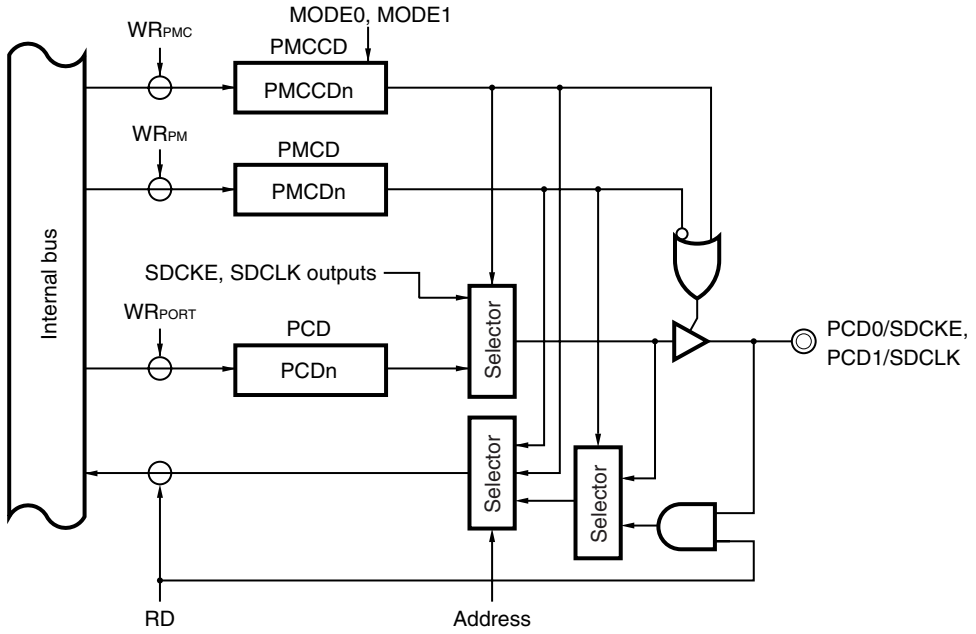
PMCCD2	Specification of operation mode of PCD2 pin
0	I/O port
1	$\overline{\text{SDCAS}}$ output

PMCCD1	Specification of operation mode of PCD1 pin
0	I/O port
1	SDCLK output

PMCCD0	Specification of operation mode of PCD0 pin
0	I/O port
1	SDCKE output

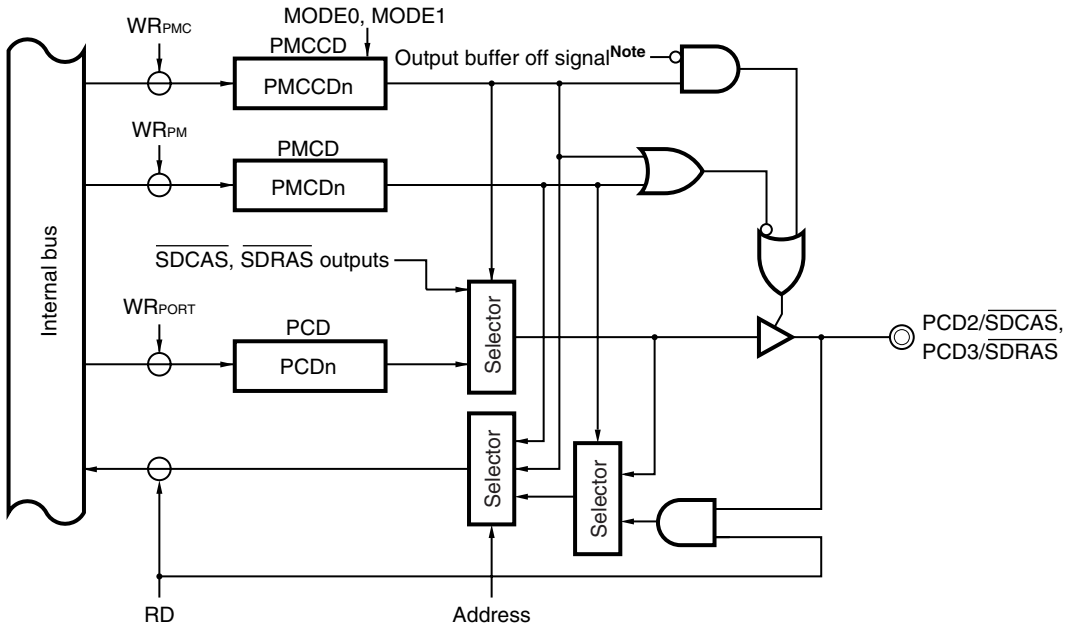
(2) Block diagram

Figure 4-37. Block Diagram of PCD0 and PCD1 Pins



Remark n = 0, 1

Figure 4-38. Block Diagram of PCD2 and PCD3 Pins



Remark n = 2, 3

4.3.16 Port BD

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

Port BD has an alternate function as the following pins.

Table 4-18. Alternate-Function Pins of Port BD

Pin Name	Pin No.		Alternate-Function Pin	I/O
	GJ	F1		
PBD0	42	N4	$\overline{\text{DMAAK0}}$	I/O
PBD1	41	P3	$\overline{\text{DMAAK1}}$	
PBD2	40	M4	$\overline{\text{DMAAK2}}$	
PBD3	39	N3	$\overline{\text{DMAAK3}}$	

Remark GJ: 144-pin plastic LQFP (fine pitch) (20 × 20)
F1: 161-pin plastic FBGA (13 × 13)

(1) Registers

(a) Port BD register (PBD)

After reset: Undefined R/W Address: FFFFF012H

	7	6	5	4	3	2	1	0
PBD	0	0	0	0	PBD3	PBD2	PBD1	PBD0

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

(b) Port BD mode register (PMBD)

After reset: FFH R/W Address: FFFFF032H

	7	6	5	4	3	2	1	0
PMBD	1	1	1	1	PMBD3	PMBD2	PMBD1	PMBD0

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

(c) Port BD mode control register (PMCBD)

After reset: 00H R/W Address: FFFFF052H

	7	6	5	4	3	2	1	0
PMCBD	0	0	0	0	PMCBD3	PMCBD2	PMCBD1	PMCBD0

PMCBD3	Specification of operation mode of PBD3 pin
0	I/O port
1	$\overline{\text{DMAAK3}}$ output

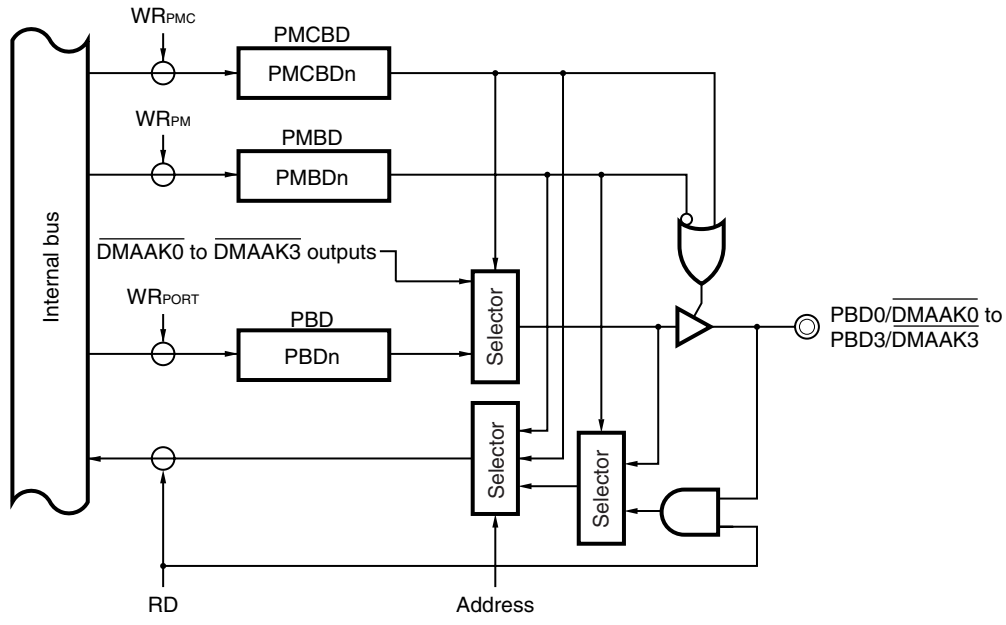
PMCBD2	Specification of operation mode of PBD2 pin
0	I/O port
1	$\overline{\text{DMAAK2}}$ output

PMCBD1	Specification of operation mode of PBD1 pin
0	I/O port
1	$\overline{\text{DMAAK1}}$ output

PMCBD0	Specification of operation mode of PBD0 pin
0	I/O port
1	$\overline{\text{DMAAK0}}$ output

(2) Block diagram

Figure 4-39. Block Diagram of PBD0 to PBD3 Pins



Remark n = 0 to 3

4.4 Setting to Use Alternate Function of Port Pin

Set the port pins as shown in Table 4-19 to use their alternate function.

Table 4-19. Using Alternate Function of Port Pins (1/9)

Pin Name	Alternate Function		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 Bits (Registers)
	Name	I/O						
P00	$\overline{\text{INTP000}}$	Input	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	PFCE00 = 0	PFC00 = 0	INTR00 (INTR0), INTF00 (INTF0)
	TOP00	Output	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	PFCE00 = 0	PFC00 = 1	
	EVTP00	Input	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	PFCE00 = 1	PFC00 = 0	
	TIP00	Input	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	PFCE00 = 1	PFC00 = 0	
	INTPP00	Input	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	PFCE00 = 1	PFC00 = 0	
P01	$\overline{\text{INTP001}}$	Input	P01 = Setting not required	PM01 = Setting not required	PMC01 = 1	PFCE01 = 0	PFC01 = 0	INTR01 (INTR0), INTF01 (INTF0)
	TOP01	Output	P01 = Setting not required	PM01 = Setting not required	PMC01 = 1	PFCE01 = 0	PFC01 = 1	
	INTPP01	Input	P01 = Setting not required	PM01 = Setting not required	PMC01 = 1	PFCE01 = 1	PFC01 = 0	
P04	$\overline{\text{INTP004}}$	Input	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	PFCE04 = 0	PFC04 = 0	INTR04 (INTR0), INTF04 (INTF0)
	$\overline{\text{DMARQ0}}$	Input	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	PFCE04 = 0	PFC04 = 1	
	INTP11	Input	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	PFCE04 = 1	PFC04 = 0	
	TCLR10	Input	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	PFCE04 = 1	PFC04 = 0	
P05	$\overline{\text{INTP005}}$	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	PFCE05 = 0	PFC05 = 0	INTR05 (INTR0), INTF05 (INTF0)
	$\overline{\text{DMARQ1}}$	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	PFCE05 = 0	PFC05 = 1	
	INTP10	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	PFCE05 = 1	PFC05 = 0	
	TCUD10	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	PFCE05 = 1	PFC05 = 0	

Table 4-19. Using Alternate Function of Port Pins (2/9)

Pin Name	Alternate Function		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 Bits (Registers)
	Name	I/O						
P06	$\overline{\text{INTP106}}$	Input	P06 = Setting not required	PM06 = Setting not required	PMC06 = 1	–	PFC06 = 0	INTR06 (INTR0), INTF06 (INTF0)
	$\overline{\text{DMARQ2}}$	Input	P06 = Setting not required	PM06 = Setting not required	PMC06 = 1	–	PFC06 = 1	
	TMS[Note]	Input	P06 = Setting not required	PM06 = Setting not required	PMC06 = Setting not required	–	PFC06 = Setting not required	
P07	$\overline{\text{INTP107}}$	Input	P07 = Setting not required	PM07 = Setting not required	PMC07 = 1	–	PFC07 = 0	INTR07 (INTR0), INTF07 (INTF0)
	$\overline{\text{DMARQ3}}$	Input	P07 = Setting not required	PM07 = Setting not required	PMC07 = 1	–	PFC07 = 1	
	TCK[Note]	Input	P07 = Setting not required	PM07 = Setting not required	PMC07 = Setting not required	–	PFC07 = Setting not required	
P10	$\overline{\text{INTP010}}$	Input	P10 = Setting not required	PM10 = Setting not required	PMC10 = 1	PFCE10 = 0	PFC10 = 0	INTR10 (INTR1), INTF10 (INTF1)
	TOQB1	Output	P10 = Setting not required	PM10 = Setting not required	PMC10 = 1	PFCE10 = 0	PFC10 = 1	
	INTPQ0	Input	P10 = Setting not required	PM10 = Setting not required	PMC10 = 1	PFCE10 = 1	PFC10 = 0	
	TOQ0	Output	P10 = Setting not required	PM10 = Setting not required	PMC10 = 1	PFCE10 = 1	PFC10 = 1	
P11	$\overline{\text{INTP011}}$	Input	P11 = Setting not required	PM11 = Setting not required	PMC11 = 1	PFCE11 = 0	PFC11 = 0	INTR11 (INTR1), INTF11 (INTF1)
	TOQT1	Output	P11 = Setting not required	PM11 = Setting not required	PMC11 = 1	PFCE11 = 0	PFC11 = 1	
	INTPQ1	Input	P11 = Setting not required	PM11 = Setting not required	PMC11 = 1	PFCE11 = 1	PFC11 = 0	
	TOQ1	Output	P11 = Setting not required	PM11 = Setting not required	PMC11 = 1	PFCE11 = 1	PFC11 = 1	
P12	$\overline{\text{INTP012}}$	Input	P12 = Setting not required	PM12 = Setting not required	PMC12 = 1	PFCE12 = 0	PFC12 = 0	INTR12 (INTR1), INTF12 (INTF1)
	TOQT2	Output	P12 = Setting not required	PM12 = Setting not required	PMC12 = 1	PFCE12 = 0	PFC12 = 1	
	INTPQ2	Input	P12 = Setting not required	PM12 = Setting not required	PMC12 = 1	PFCE12 = 1	PFC12 = 0	
	TOQ2	Output	P12 = Setting not required	PM12 = Setting not required	PMC12 = 1	PFCE12 = 1	PFC12 = 1	
P13	$\overline{\text{INTP013}}$	Input	P13 = Setting not required	PM13 = Setting not required	PMC13 = 1	PFCE13 = 0	PFC13 = 0	INTR13 (INTR1), INTF13 (INTF1)
	TOQT3	Output	P13 = Setting not required	PM13 = Setting not required	PMC13 = 1	PFCE13 = 0	PFC13 = 1	
	INTPQ3	Input	P13 = Setting not required	PM13 = Setting not required	PMC13 = 1	PFCE13 = 1	PFC13 = 0	
	TOQ3	Output	P13 = Setting not required	PM13 = Setting not required	PMC13 = 1	PFCE13 = 1	PFC13 = 1	

Note The TMS and TCK pins are for on-chip debugging. To use the P06 and P07 pins as P06/$\overline{\text{INTP106}}$/$\overline{\text{DMARQ2}}$ and P07/$\overline{\text{INTP107}}$/$\overline{\text{DMARQ3}}$, be sure to input a low level to the $\overline{\text{TRST}}$ pin. If a high level is input to the $\overline{\text{TRST}}$ pin, the values set to the P0, PM0, PMC0, and PFC0 registers become invalid, and the P06 and P07 pins function as the TMS and TCK pins.

Table 4-19. Using Alternate Function of Port Pins (3/9)

Pin Name	Alternate Function		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 Bits (Registers)
	Name	I/O						
P14	$\overline{\text{INTP114}}$	Input	P14 = Setting not required	PM14 = Setting not required	PMC14 = 1	PFCE14 = 0	PFC14 = 0	INTR14 (INTR1), INTF14 (INTF1)
	TOQB2	Output	P14 = Setting not required	PM14 = Setting not required	PMC14 = 1	PFCE14 = 0	PFC14 = 1	
	TIQ	Input	P14 = Setting not required	PM14 = Setting not required	PMC14 = 1	PFCE14 = 1	PFC14 = 0	
P15	$\overline{\text{INTP115}}$	Input	P15 = Setting not required	PM15 = Setting not required	PMC15 = 1	PFCE15 = 0	PFC15 = 0	INTR15 (INTR1), INTF15 (INTF1)
	TOQB3	Output	P15 = Setting not required	PM15 = Setting not required	PMC15 = 1	PFCE15 = 0	PFC15 = 1	
	EVTQ	Input	P15 = Setting not required	PM15 = Setting not required	PMC15 = 1	PFCE15 = 1	PFC15 = 0	
P20	NMI	Input	P20 = Setting not required	–	–	–	–	NMIF0 (NMIF), NMIR0 (NMIR)
P21	$\overline{\text{INTP021}}$	Input	P21 = Setting not required	PM21 = Setting not required	PMC21 = 1	PFCE21 = 0	PFC21 = 0	INTR21 (INTR2), INTF21 (INTF2)
	TOP10	Output	P21 = Setting not required	PM21 = Setting not required	PMC21 = 1	PFCE21 = 0	PFC21 = 1	
	EVTP1	Input	P21 = Setting not required	PM21 = Setting not required	PMC21 = 1	PFCE21 = 1	PFC21 = 0	
	TIP1	Input	P21 = Setting not required	PM21 = Setting not required	PMC21 = 1	PFCE21 = 1	PFC21 = 0	
	INTPP10	Input	P21 = Setting not required	PM21 = Setting not required	PMC21 = 1	PFCE21 = 1	PFC21 = 0	
P22	$\overline{\text{INTP022}}$	Input	P22 = Setting not required	PM22 = Setting not required	PMC22 = 1	PFCE22 = 0	PFC22 = 0	INTR22 (INTR2), INTF22 (INTF2)
	TOP11	Output	P22 = Setting not required	PM22 = Setting not required	PMC22 = 1	PFCE22 = 0	PFC22 = 1	
	INTPP11	Input	P22 = Setting not required	PM22 = Setting not required	PMC22 = 1	PFCE22 = 1	PFC22 = 0	
P24	$\overline{\text{INTP124}}$	Input	P24 = Setting not required	PM24 = Setting not required	PMC24 = 1	–	PFC24 = 0	INTR24 (INTR2), INTF24 (INTF2)
	$\overline{\text{TC0}}$	Output	P24 = Setting not required	PM24 = Setting not required	PMC24 = 1	–	PFC24 = 1	
P25	$\overline{\text{INTP125}}$	Input	P25 = Setting not required	PM25 = Setting not required	PMC25 = 1	PFCE25 = 0	PFC25 = 0	INTR25 (INTR2), INTF25 (INTF2)
	$\overline{\text{TC1}}$	Output	P25 = Setting not required	PM25 = Setting not required	PMC25 = 1	PFCE25 = 0	PFC25 = 1	
	TIUD10	Input	P25 = Setting not required	PM25 = Setting not required	PMC25 = 1	PFCE25 = 1	PFC25 = 0	
	TO10	Output	P25 = Setting not required	PM25 = Setting not required	PMC25 = 1	PFCE25 = 1	PFC25 = 1	

Table 4-19. Using Alternate Function of Port Pins (4/9)

Pin Name	Alternate Function		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 Bits (Registers)
	Name	I/O						
P26	$\overline{\text{INTP126}}$	Input	P26 = Setting not required	PM26 = Setting not required	PMC26 = 1	–	PFC26 = 0	INTR26 (INTR2), INTF26 (INTF2)
	$\overline{\text{TC2}}$	Output	P26 = Setting not required	PM26 = Setting not required	PMC26 = 1	–	PFC26 = 1	
	TDI[Note 1]	Input	P26 = Setting not required	PM26 = Setting not required	PMC26 = Setting not required	–	PFC26 = Setting not required	
P27	$\overline{\text{TC3}}$	Output	P27 = Setting not required	PM27 = Setting not required	PMC27 = 1	–	–	
	TDO[Note 1]	Output	P27 = Setting not required	PM27 = Setting not required	PMC27 = Setting not required	–	–	
P30	$\overline{\text{INTP130}}$	Input	P30 = Setting not required	PM30 = Setting not required	PMC30 = 1	PFCE30 = 0	PFC30 = 0	INTR30 (INTR3), INTF30 (INTF3)
	TXD2	Output	P30 = Setting not required	PM30 = Setting not required	PMC30 = 1	PFCE30 = 0	PFC30 = 1	
	SO2	Output	P30 = Setting not required	PM30 = Setting not required	PMC30 = 1	PFCE30 = 1	PFC30 = 0	
P31	$\overline{\text{INTP131}}$	Input	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	PFCE31 = 0	PFC31 = 0	INTR31 (INTR3), INTF31 (INTF3)
	RXD2	Input	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	PFCE31 = 0	PFC31 = 1	
	SI2	Input	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	PFCE31 = 1	PFC31 = 0	
P32	$\overline{\text{INTP132}}$	Input	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 0	PFC32 = 0	INTR32 (INTR3), INTF32 (INTF3)
	ASCK2	Input	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 0	PFC32 = 1	
	SCK2	I/O	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 1	PFC32 = 0	
P33	$\overline{\text{INTP133}}$	Input	P33 = Setting not required	PM33 = Setting not required	PMC33 = 1	PFCE33 = 0	PFC33 = 0	INTR33 (INTR3), INTF33 (INTF3)
	TXD3	Output	P33 = Setting not required	PM33 = Setting not required	PMC33 = 1	PFCE33 = 0	PFC33 = 1	
	SDA[Note 2]	I/O	P33 = Setting not required	PM33 = Setting not required	PMC33 = 1	PFCE33 = 1	PFC33 = 0	
P34	$\overline{\text{INTP134}}$	Input	P34 = Setting not required	PM34 = Setting not required	PMC34 = 1	PFCE34 = 0	PFC34 = 0	INTR34 (INTR3), INTF34 (INTF3)
	RXD3	Input	P34 = Setting not required	PM34 = Setting not required	PMC34 = 1	PFCE34 = 0	PFC34 = 1	
	SCL[Note 2]	I/O	P34 = Setting not required	PM34 = Setting not required	PMC34 = 1	PFCE34 = 1	PFC34 = 0	
P37	$\overline{\text{INTP137}}$	Input	P37 = Setting not required	PM37 = Setting not required	PMC37 = 1	–	–	INTR37 (INTR3), INTF37 (INTF3)
	ADTRG	Output	P37 = Setting not required	PM37 = Setting not required	PMC37 = 1	–	–	

Notes 1. TDI and TDO pins are for on-chip debugging. To use the P26 and P27 pins as P26/$\overline{\text{INTP126}}$/$\overline{\text{TC2}}$ and P27/$\overline{\text{TC3}}$, be sure to input a low level to the $\overline{\text{TRST}}$ pin. If a high level is input to the $\overline{\text{TRST}}$ pin, the values set to the P2, PM2, PMC2, and PFC2 registers become invalid, and the P06 and P07 pins function as the TDI and TDO pins.

2. I^2C bus versions (Y products) (see **Table 1-1**)

When the P33 and P34 pins are used as the SDA and SCL pins, respectively, they function as dummy open-drain output pins (P-ch is always off).

Table 4-19. Using Alternate Function of Port Pins (5/9)

Pin Name	Alternate Function		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 Bits (Registers)
	Name	I/O						
P40	SO0	Output	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	–	PFC40 = 0	
	TXD0	Output	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	–	PFC40 = 1	
P41	SI0	Input	P41 = Setting not required	PM41 = Setting not required	PMC41 = 1	–	PFC41 = 0	
	RXD0	Input	P41 = Setting not required	PM41 = Setting not required	PMC41 = 1	–	PFC41 = 1	
P42	SCK0	I/O	P42 = Setting not required	PM42 = Setting not required	PMC42 = 1	–	PFC42 = 0	
	ASCK0	Input	P42 = Setting not required	PM42 = Setting not required	PMC42 = 1	–	PFC42 = 1	
P43	SO1	Output	P43 = Setting not required	PM43 = Setting not required	PMC43 = 1	–	PFC43 = 0	
	TXD1	Output	P43 = Setting not required	PM43 = Setting not required	PMC43 = 1	–	PFC43 = 1	
P44	SI1	Input	P44 = Setting not required	PM44 = Setting not required	PMC44 = 1	–	PFC44 = 0	
	RXD1	Input	P44 = Setting not required	PM44 = Setting not required	PMC44 = 1	–	PFC44 = 1	
P45	SCK1	I/O	P45 = Setting not required	PM45 = Setting not required	PMC45 = 1	–	PFC45 = 0	
	ASCK1	Input	P45 = Setting not required	PM45 = Setting not required	PMC45 = 1	–	PFC45 = 1	
P50	$\overline{\text{INTP050}}$	Input	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 0	PFC50 = 0	INTR50 (INTR5), INTF50 (INTF5)
	TOP20	Output	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 0	PFC50 = 1	
	EVTP2	Input	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 1	PFC50 = 0	
	TIP2	Input	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 1	PFC50 = 0	
	INTPP20	Input	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 1	PFC50 = 0	
P51	$\overline{\text{INTP051}}$	Input	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 0	PFC51 = 0	INTR51 (INTR5), INTF51 (INTF5)
	TOP21	Output	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 0	PFC51 = 1	
	INTPP21	Input	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 1	PFC51 = 0	

Table 4-19. Using Alternate Function of Port Pins (6/9)

Pin Name	Alternate Function		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 Bits (Registers)
	Name	I/O						
P70	ANI0	Input	P70 = Setting not required	–	PMC70 = 1	–	–	
P71	ANI1	Input	P71 = Setting not required	–	PMC71 = 1	–	–	
P72	ANI2	Input	P72 = Setting not required	–	PMC72 = 1	–	–	
P73	ANI3	Input	P73 = Setting not required	–	PMC73 = 1	–	–	
P74	ANI4	Input	P74 = Setting not required	–	PMC74 = 1	–	–	
P75	ANI5	Input	P75 = Setting not required	–	PMC75 = 1	–	–	
P76	ANI6	Input	P76 = Setting not required	–	PMC76 = 1	–	–	
P77	ANI7	Input	P77 = Setting not required	–	PMC77 = 1	–	–	
P80	ANO0	Output	P80 = Setting not required	–	–	–	–	
P81	ANO1	Output	P81 = Setting not required	–	–	–	–	
PAL0	A0	Output	PAL0 = Setting not required	PMAL0 = Setting not required	PMCAL0 = 1	–	–	
PAL1	A1	Output	PAL1 = Setting not required	PMAL1 = Setting not required	PMCAL1 = 1	–	–	
PAL2	A2	Output	PAL2 = Setting not required	PMAL2 = Setting not required	PMCAL2 = 1	–	–	
PAL3	A3	Output	PAL3 = Setting not required	PMAL3 = Setting not required	PMCAL3 = 1	–	–	
PAL4	A4	Output	PAL4 = Setting not required	PMAL4 = Setting not required	PMCAL4 = 1	–	–	
PAL5	A5	Output	PAL5 = Setting not required	PMAL5 = Setting not required	PMCAL5 = 1	–	–	
PAL6	A6	Output	PAL6 = Setting not required	PMAL6 = Setting not required	PMCAL6 = 1	–	–	
PAL7	A7	Output	PAL7 = Setting not required	PMAL7 = Setting not required	PMCAL7 = 1	–	–	
PAL8	A8	Output	PAL8 = Setting not required	PMAL8 = Setting not required	PMCAL8 = 1	–	–	
PAL9	A9	Output	PAL9 = Setting not required	PMAL9 = Setting not required	PMCAL9 = 1	–	–	
PAL10	A10	Output	PAL10 = Setting not required	PMAL10 = Setting not required	PMCAL10 = 1	–	–	
PAL11	A11	Output	PAL11 = Setting not required	PMAL11 = Setting not required	PMCAL11 = 1	–	–	
PAL12	A12	Output	PAL12 = Setting not required	PMAL12 = Setting not required	PMCAL12 = 1	–	–	

Table 4-19. Using Alternate Function of Port Pins (7/9)

Pin Name	Alternate Function		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 Bits (Registers)
	Name	I/O						
PAL13	A13	Output	PAL13 = Setting not required	PMAL13 = Setting not required	PMCAL13 = 1	–	–	
PAL14	A14	Output	PAL14 = Setting not required	PMAL14 = Setting not required	PMCAL14 = 1	–	–	
PAL15	A15	Output	PAL15 = Setting not required	PMAL15 = Setting not required	PMCAL15 = 1	–	–	
PAH0	A16	Output	PAH0 = Setting not required	PMAH0 = Setting not required	PMCAH0 = 1	–	–	
PAH1	A17	Output	PAH1 = Setting not required	PMAH1 = Setting not required	PMCAH1 = 1	–	–	
PAH2	A18	Output	PAH2 = Setting not required	PMAH2 = Setting not required	PMCAH2 = 1	–	–	
PAH3	A19	Output	PAH3 = Setting not required	PMAH3 = Setting not required	PMCAH3 = 1	–	–	
PAH4	A20	Output	PAH4 = Setting not required	PMAH4 = Setting not required	PMCAH4 = 1	–	–	
PAH5	A21	Output	PAH5 = Setting not required	PMAH5 = Setting not required	PMCAH5 = 1	–	–	
PAH6	A22	Output	PAH6 = Setting not required	PMAH6 = Setting not required	PMCAH6 = 1	–	–	
PAH7	A23	Output	PAH7 = Setting not required	PMAH7 = Setting not required	PMCAH7 = 1	–	–	
PAH8	A24	Output	PAH8 = Setting not required	PMAH8 = Setting not required	PMCAH8 = 1	–	–	
PAH9	A25	Output	PAH9 = Setting not required	PMAH9 = Setting not required	PMCAH9 = 1	–	–	
PDL0	AD0	I/O	PDL0 = Setting not required	PMDL0 = Setting not required	PMCDL0 = 1	–	–	
PDL1	AD1	I/O	PDL1 = Setting not required	PMDL1 = Setting not required	PMCDL1 = 1	–	–	
PDL2	AD2	I/O	PDL2 = Setting not required	PMDL2 = Setting not required	PMCDL2 = 1	–	–	
PDL3	AD3	I/O	PDL3 = Setting not required	PMDL3 = Setting not required	PMCDL3 = 1	–	–	
PDL4	AD4	I/O	PDL4 = Setting not required	PMDL4 = Setting not required	PMCDL4 = 1	–	–	
PDL5	AD5	I/O	PDL5 = Setting not required	PMDL5 = Setting not required	PMCDL5 = 1	–	–	
PDL6	AD6	I/O	PDL6 = Setting not required	PMDL6 = Setting not required	PMCDL6 = 1	–	–	
PDL7	AD7	I/O	PDL7 = Setting not required	PMDL7 = Setting not required	PMCDL7 = 1	–	–	
PDL8	AD8	I/O	PDL8 = Setting not required	PMDL8 = Setting not required	PMCDL8 = 1	–	–	
PDL9	AD9	I/O	PDL9 = Setting not required	PMDL9 = Setting not required	PMCDL9 = 1	–	–	

Table 4-19. Using Alternate Function of Port Pins (8/9)

Pin Name	Alternate Function		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 Bits (Registers)
	Name	I/O						
PDL10	AD10	I/O	PDL10 = Setting not required	PMDL10 = Setting not required	PMCDL10 = 1	–	–	
PDL11	AD11	I/O	PDL11 = Setting not required	PMDL11 = Setting not required	PMCDL11 = 1	–	–	
PDL12	AD12	I/O	PDL12 = Setting not required	PMDL12 = Setting not required	PMCDL12 = 1	–	–	
PDL13	AD13	I/O	PDL13 = Setting not required	PMDL13 = Setting not required	PMCDL13 = 1	–	–	
PDL14	AD14	I/O	PDL14 = Setting not required	PMDL14 = Setting not required	PMCDL14 = 1	–	–	
PDL15	AD15	I/O	PDL15 = Setting not required	PMDL15 = Setting not required	PMCDL15 = 1	–	–	
PCS0	$\overline{\text{CS0}}$	Output	PCS0 = Setting not required	PMCS0 = Setting not required	PMCCS0 = 1	–	–	
PCS1	$\overline{\text{CS1}}$	Output	PCS1 = Setting not required	PMCS1 = Setting not required	PMCCS1 = 1	–	–	
PCS2	$\overline{\text{CS2}}$	Output	PCS2= Setting not required	PMCS2= Setting not required	PMCCS2 = 1	–	PFCCS2 = 0	
	$\overline{\text{IOWR}}$	Output	PCS2 = Setting not required	PMCS2 = Setting not required	PMCCS2 = 1	–	PFCCS2 = 1	
PCS3	$\overline{\text{CS3}}$	Output	PCS3 = Setting not required	PMCS3 = Setting not required	PMCCS3 = 1	–	–	
PCS4	$\overline{\text{CS4}}$	Output	PCS4 = Setting not required	PMCS4 = Setting not required	PMCCS4 = 1	–	–	
PCS5	$\overline{\text{CS5}}$	Output	PCS5 = Setting not required	PMCS5 = Setting not required	PMCCS5 = 1	–	PFCCS5 = 0	
	$\overline{\text{IORD}}$	Output	PCS5 = Setting not required	PMCS5 = Setting not required	PMCCS5 = 1	–	PFCCS5 = 1	
PCS6	$\overline{\text{CS6}}$	Output	PCS6 = Setting not required	PMCS6 = Setting not required	PMCCS6 = 1	–	–	
PCS7	$\overline{\text{CS7}}$	Output	PCS7 = Setting not required	PMCS7 = Setting not required	PMCCS7 = 1	–	–	
PCT0	$\overline{\text{LWR}}$	Output	PCT0 = Setting not required	PMCT0 = Setting not required	PMCCT0 = 1	–	PFCCT0 = 0, **Note**	
	LDQM	Output	PCT0 = Setting not required	PMCT0 = Setting not required	PMCCT0 = 1	–	PFCCT0 = 0/1, **Note**	
	$\overline{\text{LBE}}$	Output	PCT0 = Setting not required	PMCT0 = Setting not required	PMCCT0 = 1	–	PFCCT0 = 1, **Note**	
PCT1	$\overline{\text{UWR}}$	Output	PCT1 = Setting not required	PMCT1 = Setting not required	PMCCT1 = 1	–	PFCCT1 = 0, **Note**	
	UDQM	Output	PCT1 = Setting not required	PMCT1 = Setting not required	PMCCT1 = 1	–	PFCCT1 = 0/1, **Note**	
	$\overline{\text{UBE}}$	Output	PCT1 = Setting not required	PMCT1 = Setting not required	PMCCT1 = 1	–	PFCCT1 = 1, **Note**	

Note $\overline{\text{xWR}}$ output or xDQM output, or $\overline{\text{xBE}}$ output and xDQM output are automatically selected when the target memory is accessed (x = U or L).

Table 4-19. Using Alternate Function of Port Pins (9/9)

Pin Name	Alternate Function		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 Bits (Registers)
	Name	I/O						
PCT4	$\overline{RD}$	Output	PCT4 = Setting not required	PMCT4 = Setting not required	PMCCT4 = 1	–	–	
PCT5	$\overline{WR}$	Output	PCT5 = Setting not required	PMCT5 = Setting not required	PMCCT5 = 1	–	–	
	$\overline{WE}$	Output	PCT5 = Setting not required	PMCT5 = Setting not required	PMCCT5 = 1	–	–	
PCT6	ASTB	Output	PCT6 = Setting not required	PMCT6 = Setting not required	PMCCT6 = 1	–	–	
PCT7	$\overline{BCYST}$	Output	PCT7 = Setting not required	PMCT7 = Setting not required	PMCCT7 = 1	–	–	
PCM0	$\overline{WAIT}$	Input	PCM0 = Setting not required	PMCM0 = Setting not required	PMCCM0 = 1	–	–	
PCM1	BUSCLK	Output	PCM1 = Setting not required	PMCM1 = Setting not required	PMCCM1 = 1	–	–	
PCM2	$\overline{HLDAK}$	Output	PCM2 = Setting not required	PMCM2 = Setting not required	PMCCM2 = 1	–	–	
PCM3	$\overline{HLDRQ}$	Input	PCM3 = Setting not required	PMCM3 = Setting not required	PMCCM3 = 1	–	–	
PCM4	$\overline{REFRQ}$	Output	PCM4 = Setting not required	PMCM4 = Setting not required	PMCCM4 = 1	–	–	
PCD0	SDCKE	Output	PCD0 = Setting not required	PMCD0 = Setting not required	PMCCD0 = 1	–	–	
PCD1	SDCLK	Output	PCD1 = Setting not required	PMCD1 = Setting not required	PMCCD1 = 1	–	–	
PCD2	$\overline{SDCAS}$	Output	PCD2 = Setting not required	PMCD2 = Setting not required	PMCCD2 = 1	–	–	
PCD3	$\overline{SDRAS}$	Output	PCD3 = Setting not required	PMCD3 = Setting not required	PMCCD3 = 1	–	–	
PBD0	$\overline{DMAAK0}$	Output	PBD0 = Setting not required	PMBD0 = Setting not required	PMCBD0 = 1	–	–	
PBD1	$\overline{DMAAK1}$	Output	PBD1 = Setting not required	PMBD1 = Setting not required	PMCBD1 = 1	–	–	
PBD2	$\overline{DMAAK2}$	Output	PBD2 = Setting not required	PMBD2 = Setting not required	PMCBD2 = 1	–	–	
PBD3	$\overline{DMAAK3}$	Output	PBD3 = Setting not required	PMBD3 = Setting not required	PMCBD3 = 1	–	–	

4.5 Noise Eliminator

4.5.1 Interrupt input pin

The following timing controller used to secure the noise elimination time is provided for the NMI and port pins that operate in the alternate-function mode when the valid edge is input. Input signals that change within the noise elimination time are not internally acknowledged.

Table 4-20. Noise Elimination Time of Interrupt Input Pins

Pin	Noise Elimination Time
NMI	Analog delay (80 ns typ.)
$\overline{\text{INTP000}}$	
$\overline{\text{INTP001}}$	
$\overline{\text{INTP004}}$	
$\overline{\text{INTP005}}$	
$\overline{\text{INTP010}}$	
$\overline{\text{INTP011}}$	
$\overline{\text{INTP012}}$	
$\overline{\text{INTP013}}$	
$\overline{\text{INTP021}}$	
$\overline{\text{INTP022}}$	
$\overline{\text{INTP050}}$	
$\overline{\text{INTP051}}$	
$\overline{\text{INTP106}}$	
$\overline{\text{INTP107}}$	
$\overline{\text{INTP114}}$	
$\overline{\text{INTP115}}$	
$\overline{\text{INTP124}}$	
$\overline{\text{INTP125}}$	
$\overline{\text{INTP126}}$	
$\overline{\text{INTP130}}$	
$\overline{\text{INTP131}}$	
$\overline{\text{INTP132}}$	
$\overline{\text{INTP133}}$	
$\overline{\text{INTP134}}$	
$\overline{\text{INTP137}}$	

Cautions 1. The above non-maskable and maskable interrupt pins are used to release the standby mode. A timing circuit that controls the clock is not employed because the internal system clock is stopped in the standby mode.

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

4.5.2 Timer ENC1 input pins

The following noise filter that operates via clock sampling is provided for the pins of timer ENC1, which operates when the valid edge is input. Input signals that change within the noise elimination time are not internally acknowledged.

Table 4-21. Noise Elimination Time of Timer ENC1 Input Pins

Pin	Noise Elimination Time
INTP10/TCUD10 INTP11/TCLR10 TIUD10	$f_{XX} \times 3$

Cautions 1. The noise filter of the above pins cannot acknowledge an input signal when the CPU clock is stopped because it uses clock sampling.

2. The noise eliminator is valid only when used as INTP10/TCUD10, INTP11/TCLR10, and TIUD10.

4.5.3 Timer P and timer Q input pins

The following timing controller used to secure the noise elimination time is provided for the pins of timers P and Q that operate when the valid edge is input. An input signal that changes within the noise elimination time is not internally acknowledged.

Table 4-22. Noise Elimination Time of Timer P and Timer Q Input Pins

Pin	Noise Elimination Time
EVTP0/TIP0/INTPP00 INTPP01 EVTP1/TIP1/INTPP10 INTPP11 EVTP2/TIP2/INTPP20 INTPP21	Analog delay (80 ns typ.)
INTPQ0 INTPQ1 INTPQ2 INTPQ3 EVTQ TIQ	

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

4.6 Cautions

4.6.1 Cautions on setting port pins

(1) Procedure to change mode from port mode to alternate-function mode

Change the mode of a port pin that functions as an output or I/O pin in the alternate-function mode to the alternate-function mode using the following procedure (except port 8).

<1> Set the inactive level of the signal to be output in the alternate-function mode to the corresponding bit of port n (n = 0 to 5, 7, AL, AH, DL, CS, CT, CM, CD, BD).
<2> Select the alternate-function mode by using the port n mode control register (PMCn).

If <1> is not performed, the contents of port n may be momentarily output when the mode is changed from the port mode to the alternate-function mode.

(2) Changing mode to external interrupt input mode

When the mode is changed from the port mode to the external interrupt input mode ($\overline{\text{INTPa}}$), an external interrupt may be generated if an incorrect valid edge is detected. Disable (hold pending) (xxICn.xxMKn bit = 1) interrupt servicing by using the interrupt mask flag of the target interrupt control register, and perform the following procedure.

(1) Set the PFCE register.
(2) Set the PFC register.
(3) Set the PMC register[Note].
(4) Set the INTF and INTR registers[Note].
(5) Clear the interrupt request flag (xxICn.xxIFn bit = 0)[Note].
(6) Clear the interrupt mask flag (xxICn.xxMKn bit = 0).

Note Make sure that 500 ns or more elapse between (3) and (5).
It takes the level signal of the pin 80 ns (TYP.) to reach the internal circuit after the PMC register has been set in (3) above, because the analog noise eliminator of the external interrupt input pin ($\overline{\text{INTPa}}$) operates.

Remarks 1. To change the mode from the external interrupt input ($\overline{\text{INTPa}}$) mode to the port mode or the mode of other alternate-function mode, mask the external interrupt input with the xxICn.xxMKn bit.

2. xx: Identification name of each peripheral unit (see **Table 20-2**.)
n: Peripheral unit number (see **Table 20-2**.)
a = 000, 001, 004, 005, 010 to 013, 021, 022, 050, 051, 106, 107, 114, 115, 124 to 126, 130 to 134, 137

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

When a 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.

<R> Therefore, it is recommended to rewrite (using ST/SS instruction) the output latch when switching a port from input mode to output mode (including switching from alternate-function mode to port mode).

<Example> When PCS0 pin is an output port, PCS1 to PCS7 pins are input ports (all pin statuses are high level), and the value of the port latch is 00H, if the output of PCS0 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 V850E/MA3.

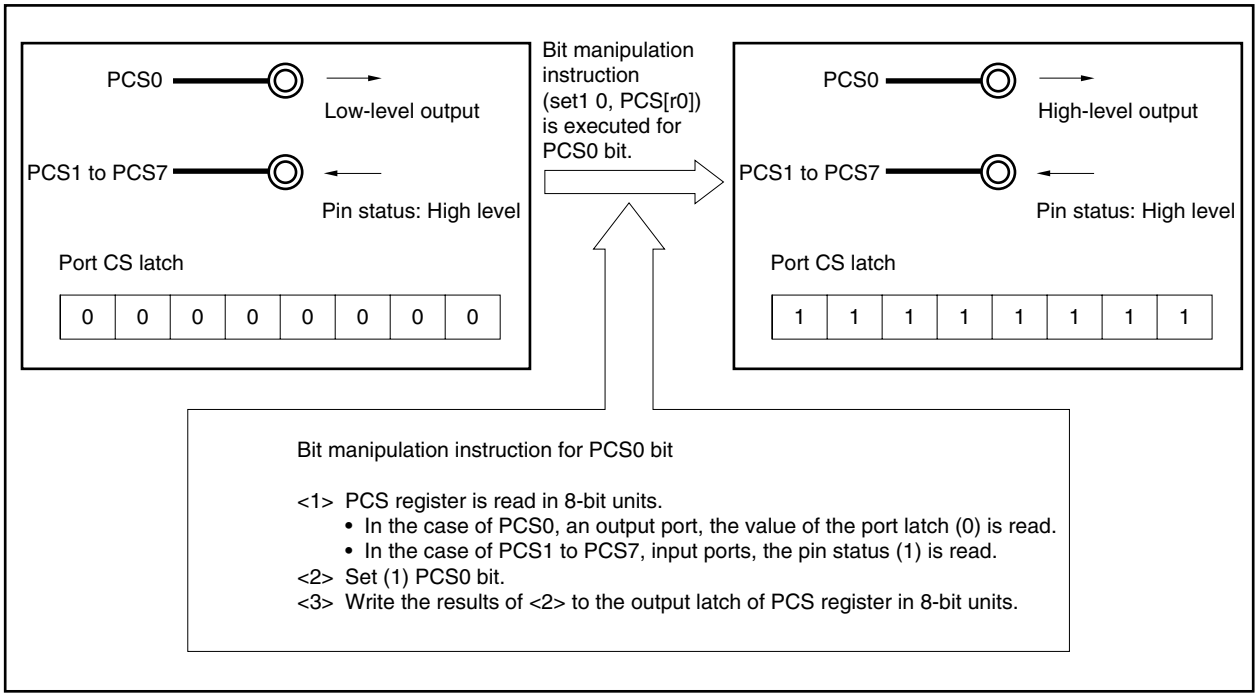
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the value of the output latch (0) of PCS0 pin, which is an output port, is read, while the pin statuses of PCS1 to PCS7 pins, which are input ports, are read. If the pin statuses of PCS1 to PCS7 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 4-40. Bit Manipulation Instruction (PCS0 Pin)



4.6.3 Hysteresis characteristics

In port mode, the following port pins do not have hysteresis characteristics.

P00, P01, P04 to P07
P10 to P15
P20 to P22, P24 to P26
P30 to P34, P37
P41, P42, P44, P45
P50, P51

CHAPTER 5 BUS CONTROL FUNCTION

The V850E/MA3 is provided with an external bus interface function via which external memories such as ROM and RAM, and I/O can be connected.

5.1 Features

- ○ 16-bit/8-bit data bus sizing function
- ○ 8-space chip select function
- ○ Wait function
  - Programmable wait function of up to 7 states for each memory block
  - External wait function using $\overline{\text{WAIT}}$ pin
- ○ Idle state insertion function
- ○ Bus arbitration function
- ○ Bus hold function
- ○ Connectable to external device with alternate-function port pin
- ○ Separate bus mode/multiplexed bus mode selectable

5.2 Bus Control Pins

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

Bus Control Pin (Function in Alternate-Function Mode)	Function in Port Mode	Port Mode/Alternate-Function Mode Switching Registers
Address/data bus (AD0 to AD15)	PDL0 to PDL15 (port DL)	PMCDL
Address bus (A0 to A15)	PAL0 to PAL15 (port AL)	PMCAL
Address bus (A16 to A25)	PAH0 to PAH9 (port AH)	PMCAH
Chip select ($\overline{\text{CS0}}$ to $\overline{\text{CS7}}$, $\overline{\text{IOWR}}$, $\overline{\text{IORD}}$)	PCS0 to PCS7 (port CS)	PMCCS
SDRAM synchronous control (SDCKE, SDCLK)	PCD0, PCD1 (port CD)	PMCCD
SDRAM control ($\overline{\text{SDCAS}}$, $\overline{\text{SDRAS}}$)	PCD2, PCD3 (port CD)	
Read/write control ($\overline{\text{LBE}}$/$\overline{\text{LWR}}$/LDQM, $\overline{\text{UBE}}$/$\overline{\text{UWR}}$/UDQM, $\overline{\text{RD}}$, $\overline{\text{WR}}$/$\overline{\text{WE}}$)	PCT0, PCT1, PCT4, PCT5 (port CT)	PMCCT
Address strobe control (ASTB)	PCT6 (port CT)	
Bus cycle start ($\overline{\text{BCYST}}$)	PCT7 (port CT)	
External wait control ($\overline{\text{WAIT}}$)	PCM0 (port CM)	PMCCM
Bus clock (BUSCLK)	PCM1 (port CM)	
Bus hold control ($\overline{\text{HLDRQ}}$, $\overline{\text{HLDAK}}$)	PCM2, PCM3 (port CM)	
SDRAM refresh control ($\overline{\text{REFRQ}}$)	PCM4 (port CM)	

(1) A0 to A15

These are the address output pins of the lower 16 bits of the address bus's 26-bit address when the external memory is accessed.

The output changes in synchronization with the rising edge of the BUSCLK signal in the T1 state. In the idle state (TI), the address of the bus cycle immediately before is retained.

(2) A16 to A25

These are the address output pins of the higher 10 bits of the address bus's 26-bit address when the external memory is accessed.

The output changes in synchronization with the rising edge of the BUSCLK signal in the T1 state. In the idle state (TI), the address of the bus cycle immediately before is retained.

(3) AD0 to AD15

These pins form a multiplexed address/data bus when the external memory is accessed. In the multiplexed bus mode, this bus outputs an address and inputs/outputs data. It inputs/outputs data in the separate bus mode.

(4) $\overline{\text{CS0}}$ to $\overline{\text{CS7}}$

These are the chip select signal output pins for the SRAM, external ROM, external peripheral I/O, page ROM, and SDRAM area.

The $\overline{\text{CSn}}$ signal is assigned to memory block n (n = 0 to 7).

It becomes active while the bus cycle that accesses the corresponding memory block is activated.

In the idle state (TI), it becomes inactive.

(5) $\overline{\text{IOWR}}$

This is a write strobe signal output pin for external I/O during DMA flyby transfer. It indicates whether the bus cycle currently being executed is a write cycle for external I/O during DMA flyby transfer.

Note that if the IOEN bit of the bus cycle period control register (BCP) is set (1), this signal can be output even in the normal SRAM, external ROM, or external I/O cycle.

(6) $\overline{\text{IORD}}$

This is a read strobe signal output pin for external I/O during DMA flyby transfer. It indicates whether the bus cycle currently being executed is a read cycle for external I/O during DMA flyby transfer.

Note that if the IOEN bit of the BCP register is set (1), this signal can be output even in the normal SRAM, external ROM, or external I/O cycle.

(7) $\overline{\text{LWR}}$

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

For the data bus, the lower byte (D0 to D7) becomes valid. If the bus cycle is a lower memory write, it becomes active at the falling edge of the BUSCLK signal in the T1 state and becomes inactive at the falling edge of the BUSCLK signal in the T2 state (in separate bus mode).

(8) $\overline{\text{UWR}}$

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

For the data bus, the higher byte (D8 to D15) becomes valid. If the bus cycle is a higher byte memory write, it becomes active at the falling edge of the BUSCLK signal in the T1 state and becomes inactive at the falling edge of the BUSCLK signal in the T2 state (in separate bus mode).

(9) LDQM

This is a pin for outputting the data bus control signal to SDRAM. For the data bus, the lower byte (D0 to D7) is valid. This signal controls SDRAM output disable during a read operation, and SDRAM byte masking during a write operation.

(10) UDQM

This is a pin for outputting the data bus control signal to SDRAM. For the data bus, the higher byte (D8 to D15) is valid. This signal controls SDRAM output disable during a read operation, and SDRAM byte masking during a write operation.

(11) $\overline{\text{LBE}}$

This is a signal output pin that enables the lower byte (D0 to D7) of the external data bus.

(12) $\overline{\text{UBE}}$

This is a signal output pin that enables the higher byte (D8 to D15) of the external data bus.

(13) $\overline{\text{RD}}$

This is a strobe signal output pin that indicates that the bus cycle currently being executed is a read cycle for the SRAM, external ROM, external peripheral I/O, or page ROM area.

(14) $\overline{\text{WE}}$

This is a enable signal output pin that indicates that the bus cycle currently being executed is a write cycle for the SDRAM area.

(15) $\overline{\text{WR}}$

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

It becomes active at the falling edge of the BUSCLK signal in the T1 state and becomes inactive at the falling edge of the BUSCLK signal in the T2 state (in separate bus mode).

(16) ASTB

This pin outputs a latch strobe signal for the external address bus.

The output signal goes low at the falling edge of the BUSCLK signal in the T1 state of a bus cycle, and goes high at the falling edge of the BUSCLK signal in the T3 state.

(17) $\overline{\text{BCYST}}$

This is a status signal output pin that shows the start of the bus cycle. It becomes active for 1 clock cycle from the start of each cycle. In the idle state (TI), it becomes inactive.

(18) $\overline{\text{WAIT}}$

This is the control signal input pin from which a data wait is inserted in the bus cycle. The $\overline{\text{WAIT}}$ signal can be input asynchronously to the BUSCLK signal. When the BUSCLK signal rises, sampling is executed (in separate bus mode). If the set/hold time is not satisfied within the sampling timing, wait insertion may not be executed.

(19) $\overline{\text{HLDAK}}$

This is the acknowledge signal output pin that indicates the high impedance status for the address bus, data bus, and control bus when the V850E/MA3 receives a bus hold request.

While this signal is active, the impedance of the address bus, data bus, and control bus becomes high and the bus mastership is transferred to the external bus master.

(20) $\overline{\text{HLDRQ}}$

This is the input pin through which an external device requests the V850E/MA3 to release the address bus, data bus, and control bus. The $\overline{\text{HLDRQ}}$ signal can be input asynchronously to the BUSCLK signal. When this pin is active, the address bus, data bus, and control bus are set to the high impedance status. This occurs either when the V850E/MA3 completes execution of the current bus cycle or immediately if no bus cycle is being executed, then the $\overline{\text{HLDAK}}$ signal is activated and the bus is released.

In order to make the bus hold state secure, keep the $\overline{\text{HLDRQ}}$ signal active until the $\overline{\text{HLDAK}}$ signal is output.

(21) $\overline{\text{REFRQ}}$

This is the refresh request signal output pin for SDRAM.

This signal becomes active during the refresh cycle. Also, during bus hold, it becomes active when a refresh request is generated and informs the external bus master that a refresh request was generated.

(22) BUSCLK

This is a clock output pin for external bus interface.

(23) SDCKE

This is the SDRAM clock enable output signal. It becomes inactive in self-refresh and standby mode.

(24) SDCLK

This is a clock output pin dedicated to SDRAM. It always outputs a clock of the same frequency as that of BUSCLK.

(25) $\overline{\text{SDCAS}}$

This is a command output signal for SDRAM.

(26) $\overline{\text{SDRAS}}$

This is a command output signal for SDRAM.

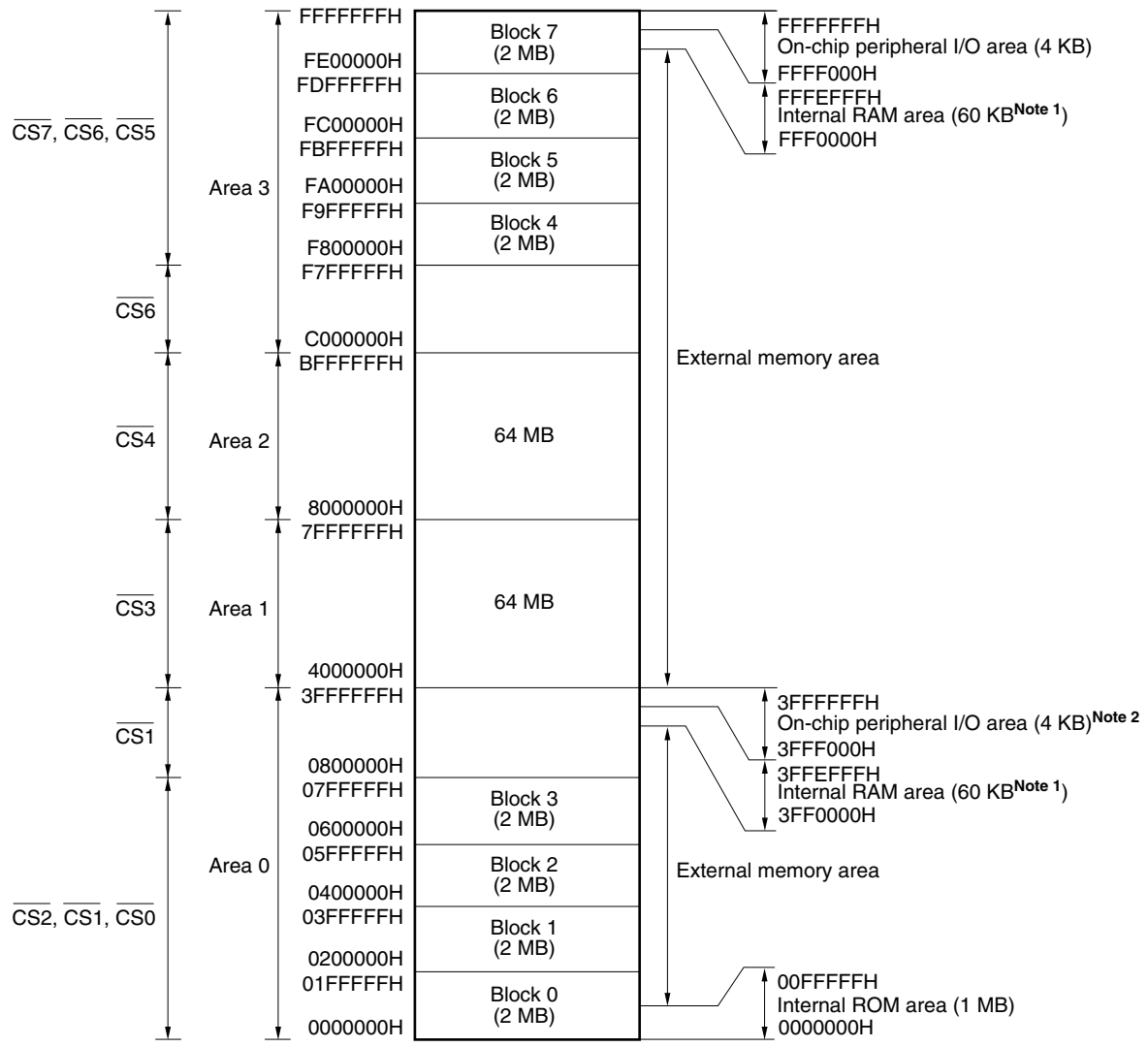
5.2.1 Pin status when internal ROM, internal RAM, or on-chip peripheral I/O is accessed

The address bus outputs a low level when the internal ROM, internal RAM, or on-chip peripheral I/O is accessed. The data bus goes into a high-impedance state without outputting anything. The external bus control signal is deasserted.

5.3 Memory Block Function

The 256 MB memory space is divided into memory blocks of 2 MB and 64 MB. The programmable wait function and bus cycle operation mode for each of these blocks can be independently controlled in one-block units.

Note that the 64 MB space of 0000000H to 3FFFFFH can be used as a program area.



<R> **Notes 1.** 8/16/32 KB (see **Table 1-1**)

2. Accessing addresses 3FFF000H to 3FFFFFFH is prohibited. To access the on-chip peripheral I/O, specify addresses FFFF000H to FFFFFFFH.

5.3.1 Chip select control function

Each memory block can be divided by using the CSC0 and CSC1 registers to control the chip select signal.

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

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

Each bit of this register becomes valid when set to 1.

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

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

If both the CSC0.CS0n and CSC0.CS2n bits are cleared to 00, $\overline{CS1}$ is output to the corresponding block (n = 0 to 3).

Similarly, if both the CSC1.CS5n and CSC1.CS7n bits are cleared to 0, $\overline{CS6}$ is output to the corresponding block (n = 0 to 3).

Reset input sets these registers to 2C11H.

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

After reset: 2C11H R/W Address: CSC0 FFFFF060H, CSC1 FFFFF062H

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

Caution For details of the CSnm bit, see Table 5-1 Specification of Chip Select Signal ($\overline{CSn}$) (n = 0 to 7, m = 0 to 3).

Table 5-1. Specification of Chip Select Signal ($\overline{CSn}$)

CSnm Bit	CS Operation
CS00	$\overline{CS0}$ is output when block 0 is accessed.
CS01	$\overline{CS0}$ is output when block 1 is accessed.
CS02	$\overline{CS0}$ is output when block 2 is accessed.
CS03	$\overline{CS0}$ is output when block 3 is accessed.
CS10 to CS13	Setting is meaningless.
CS20	$\overline{CS2}$ is output when block 0 is accessed.
CS21	$\overline{CS2}$ is output when block 1 is accessed.
CS22	$\overline{CS2}$ is output when block 2 is accessed.
CS23	$\overline{CS2}$ is output when block 3 is accessed.
CS30 to CS33	Setting is meaningless.
CS40 to CS43	Setting is meaningless.
CS50	$\overline{CS5}$ is output when block 7 is accessed.
CS51	$\overline{CS5}$ is output when block 6 is accessed.
CS52	$\overline{CS5}$ is output when block 5 is accessed.
CS53	$\overline{CS5}$ is output when block 4 is accessed.
CS60 to CS63	Setting is meaningless.
CS70	$\overline{CS7}$ is output when block 7 is accessed.
CS71	$\overline{CS7}$ is output when block 6 is accessed.
CS72	$\overline{CS7}$ is output when block 5 is accessed.
CS73	$\overline{CS7}$ is output when block 4 is accessed.

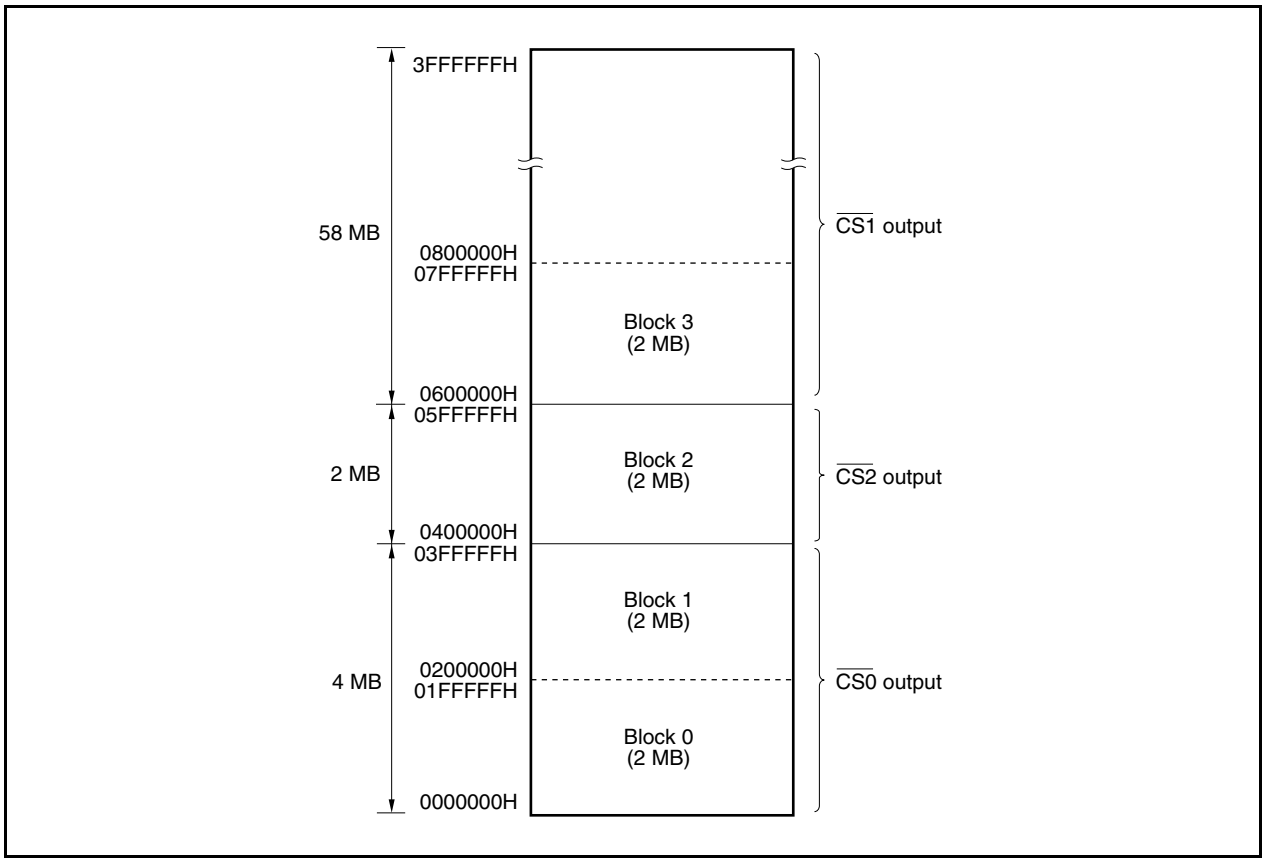
Caution The chip select signal ($\overline{CSn}$) can be enabled by setting the CSnm bit to 1 (n = 0 to 7, m = 0 to 3).

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

If the CSC0 register is set to 0703H, $\overline{CS0}$ and $\overline{CS2}$ are output to both block 0 and 1. Because $\overline{CS0}$ takes precedence over $\overline{CS2}$, however, $\overline{CS0}$ is output when an address of block 0 or 1 is accessed.

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

Figure 5-1. Example Where CSC0 Register Is Set to 0703H



5.4 Bus Cycle Type Control Function

The V850E/MA3 can directly connect the following external devices to each memory block.

- SRAM, external ROM, external I/O
- Page ROM
- SDRAM

The external device to be connected is specified by using the BCT0 and BCT1 registers.

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

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

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

Reset input sets these registers to 8888H.

Cautions 1. Write to the BCT0 and BCT1 registers after reset, and then do not change the set values (however, the MEn bit value can be changed). Also, do not access an external memory area other than the one for this initialization routine until the initial settings of the BCT0 and BCT1 registers are complete. However, external memory areas whose initial settings are complete may be accessed.

2. To connect external I/O, page ROM, or SDRAM, set the bus mode of the CSn space to the separate bus mode (n = 0 to 7).

3. When accessing a CSn space set in the multiplexed bus mode, do not use the $\overline{\text{IORD}}$ and $\overline{\text{IOWR}}$ signals (n = 0 to 7).

4. In a system that uses both the multiplexed bus mode and separate bus mode, set the number of address setup wait states corresponding to the CSn space set in the multiplexed bus mode to 1 or more (n = 0 to 7).

After reset: 8888H R/W Address: BCT0 FFFFF480H, BCT1 FFFFF482H

	15	14	13	12	11	10	9	8
BCT0	ME3	MM3	BT31	BT30	ME2	MM2	BT21	BT20
$\overline{CSn}$ signal			$\overline{CS3}$				$\overline{CS2}$	

	7	6	5	4	3	2	1	0
	ME1	MM1	BT11	BT10	ME0	MM0	BT01	BT00
$\overline{CSn}$ signal			$\overline{CS1}$				$\overline{CS0}$	

	15	14	13	12	11	10	9	8
BCT1	ME7	MM7	BT71	BT70	ME6	MM6	BT61	BT60
$\overline{CSn}$ signal			$\overline{CS7}$				$\overline{CS6}$	

	7	6	5	4	3	2	1	0
	ME5	MM5	BT51	BT50	ME4	MM4	BT41	BT40
$\overline{CSn}$ signal			$\overline{CS5}$				$\overline{CS4}$	

MEn	Enable the operation of the memory controller in each CSn space (n = 0 to 7)
0	Operation disabled
1	Operation enabled

MMn	Set the bus mode of each CSn space (n = 0 to 7)
0	Separate bus mode
1	Multiplexed bus mode

BTn1	BTn0	Specify the device to be connected to the $\overline{CSn}$ signal (n = 0 to 7)
0	0	SRAM, external I/O
0	1	Page ROM
1	0	Setting prohibited
1	1	SDRAM: n = 1, 3, 4, 6 Setting prohibited: n = 0, 2, 5, 7

5.4.2 Chip select signal delay control register (CSDC)

<R> **(1) Chip select signal delay control register (CSDC)**

If the external device of the SRAM (including external I/O) interface is accessed immediately after SDRAM is accessed, the external device of the SRAM may be written by mistake. If there is a possibility that the external device of the SRAM interface is written by mistake in a system that uses both the SDRAM and external device of the SRAM (including external I/O) interface, delay the falling of the $\overline{CSn}$ signal of SRAM (including external I/O) one clock by using this CSDC register (n = 0, 4, 6, or 7).

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

Reset input clears this register to 00H.

Cautions 1. Be sure to allocate an external device that may be written by mistake to the CSn space of $\overline{CS0}$, $\overline{CS4}$, $\overline{CS6}$, or $\overline{CS7}$. Also set the type of the memory of the CSn space to which the external device is allocated to SRAM (BTn1, BTn0 bits of BCT0, BCT1 registers = 00) (n = 0, 4, 6, or 7).

2. In the CSn space where the CSDCn bit is set to 1, be sure to insert one or more address setup wait states (the necessary number of wait states + 1), by using the ASC register (n = 0, 4, 6, or 7).

3. Do not change the CSDCn bit for the CSn space where the program currently under execution is allocated (n = 0, 4, 6, or 7).

After reset: 00H R/W Address: FFFFF804H

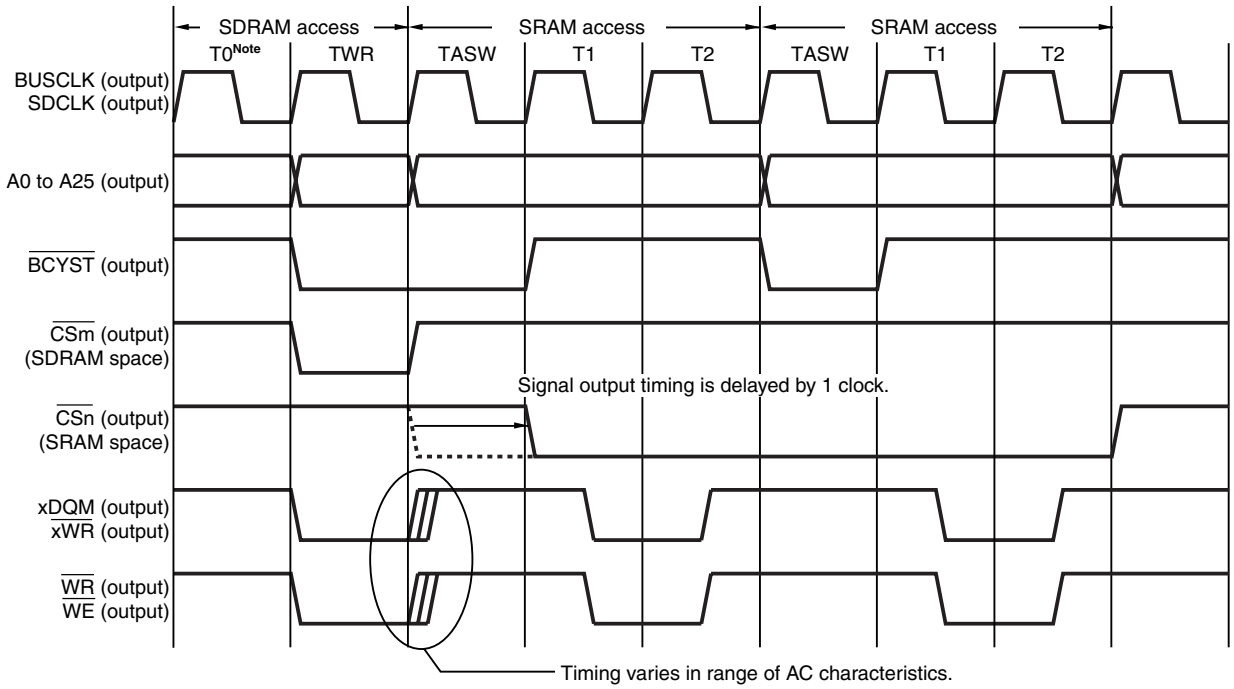
	7	6	5	4	3	2	1	0
CSDC	CSDC7	CSDC6	0	CSDC4	0	0	0	CSDC0

Remark When the CSDCn bit is set to 1, the falling of the corresponding $\overline{CSn}$ signal is delayed by one clock (n = 0, 4, 6, or 7). The output timing of the signals other than the $\overline{CSn}$ signal remains unchanged.

An example of timing chart where the CSDC.CSDCn bit is set to 1 is shown below.

Figure 5-2. Example of SRAM Access Timing Immediately After SDRAM Access (If SRAM Is Accessed Two Times in a Row After SDRAM Is Accessed)

(a) SRAM setting (address setup wait inserted, CSDC.CSDCn bit = 1, xDQM/$\overline{\text{xBE}}$/$\overline{\text{xWR}}$ pin as xDQM/$\overline{\text{xWR}}$ output)



Caution The timing is the same even when the xDQM/$\overline{\text{xBE}}$/$\overline{\text{xWR}}$ pin is used as xDQM/$\overline{\text{xBE}}$ output pin.

Note This state is inserted between bus cycles (T0).

Remark m = 1, 3, 4, or 6
n = 0, 4, 6, or 7
m ≠ n
x = U or L

5.5 Bus Access

5.5.1 Number of clocks for access

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

Bus Cycle Type / Resource (Bus Width)	Instruction Fetch	Operand Data Access
Internal ROM (32 bits)	1	7
Internal RAM (32 bits)	1[Note]	1

Note 2 if a conflict with a data access occurs.

Remark Unit: Clocks/access

5.5.2 Bus sizing function

The bus sizing function controls the data bus width of each CS space. The data bus width is set by using the LBS register.

(1) Local bus sizing control register (LBS)

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

Reset input sets this register to 5555H.

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

2. Only the following signal is asserted if the bus width is 8 bits.
$\overline{\text{LWR}}$: To access SRAM, external ROM, or external I/O (write cycle)

After reset: 5555H R/W Address: FFFFF48EH

	15	14	13	12	11	10	9	8
LBS	0	LB70	0	LB60	0	LB50	0	LB40
$\overline{\text{CSn}}$ signal		$\overline{\text{CS7}}$		$\overline{\text{CS6}}$		$\overline{\text{CS5}}$		$\overline{\text{CS4}}$

	7	6	5	4	3	2	1	0
	0	LB30	0	LB20	0	LB10	0	LB00
$\overline{\text{CSn}}$ signal		$\overline{\text{CS3}}$		$\overline{\text{CS2}}$		$\overline{\text{CS1}}$		$\overline{\text{CS0}}$

LBn0	Setting of data bus width of a CSn space (n = 0 to 7)
0	8 bits
1	16 bits

Caution Be sure to clear bits 15, 13, 11, 9, 7, 5, 3, and 1 to "0". The operation cannot be guaranteed if these bits are set to 1.

5.5.3 Endian control function

The endian control function is used to specify whether the word data in the memory of the CS space selected by the chip select signal ($\overline{CS0}$ to $\overline{CS7}$) is processed in big-endian or little-endian mode. The endianness is selected by using the BEC register.

Caution The following areas are fixed to little-endian mode and thus the setting of the BEC register is invalid.

- **On-chip peripheral I/O area**
- **Internal ROM area**
- **Internal RAM area**
- **Program fetch area of external memory**

(1) Endian configuration register (BEC)

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

Reset input clears this register to 0000H.

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

After reset: 0000H R/W Address: FFFFF068H

	15	14	13	12	11	10	9	8
BEC	0	BE70	0	BE60	0	BE50	0	BE40
$\overline{CSn}$ signal		$\overline{CS7}$		$\overline{CS6}$		$\overline{CS5}$		$\overline{CS4}$

	7	6	5	4	3	2	1	0
	0	BE30	0	BE20	0	BE10	0	BE00
$\overline{CSn}$ signal		$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$

BEn0	Specification of endian mode of each CSn space (n = 0 to 7)
0	Little-endian mode
1	Big-endian mode

Caution Be sure to clear bits 15, 13, 11, 9, 7, 5, 3, and 1 to "0". The operation cannot be guaranteed if these bits are set to 1.

Figure 5-3. Big-Endian Address in Word

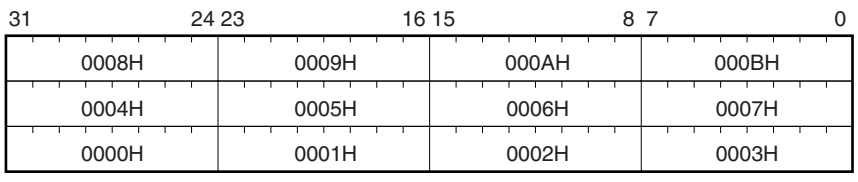
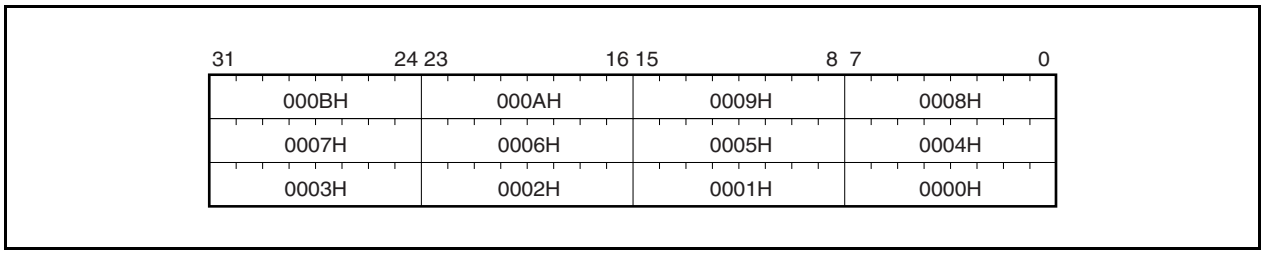


Figure 5-4. Little-Endian Address in Word



5.5.4 Restrictions on big endianness with NEC Electronics development tools

(1) When debugger (ID850) is used

Only the indication of the memory window supports big endian.

(2) When compiler (CA850) is used

(a) Restriction of C language

(i) The variables to be located in the big-endian space have the following restrictions.

- union cannot be used.
- bitfield cannot be used.
- Access by cast (changing access size) cannot be used.
- Variables with an initial value cannot be used.

(ii) Because the access size may be changed as a result of optimization, the following optimization suppressing options must be specified.

- For wide-area optimization block (opt850): –Wo, –XTb
- For machine type dependent optimization block (impr850):
  –Wi, +arg_reg_opt = OFF, +stld_trans_opt = OFF

If an access by casting or masking/shifting is not used[Note], the above optimization suppressing options do not have to be specified.

Note This applies if a pattern that causes the following optimization is not used. However, it is extremely difficult for the user to completely check this when each pattern (especially, machine type dependent optimization blocks) is combined. It is therefore recommended to specify the above optimization suppressing options.

[Wide area optimization block-related]

- Setting 1 bit by using *bit or*
  int i;
  i ^ = 1;
- Clearing 1 bit by using *bit and*
  i & = ~1;
- Negating 1 bit by using *bit xor*
  i ^ = 1;
- Testing 1 bit by using *bit and*
  if(i & 1);

[Machine type dependent optimization block-related]

Use to enable accessing the same variable in a different size

- Cast
- Mask
- Shift

Example:
```
int i, *ip;
char c;
  ⋮
c =*((char*)ip);
  ⋮
c = 0xff & i;
  ⋮
i = (i<<24) >>24;
```

(b) Restriction of assembly language

Area securing quasi directives (.hword, .word, .float, and .shword) of a size other than byte size cannot be used for variables to be located in the big-endian space.

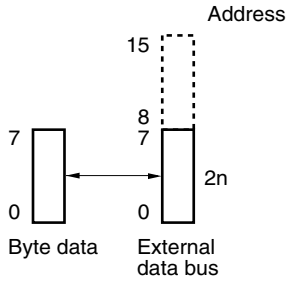
5.5.5 Bus width

The V850E/MA3 accesses the on-chip peripheral I/O and external memory in 8-bit, 16-bit, or 32-bit units. The operations when V850E/MA3 accesses the on-chip peripheral I/O and external memory are described below. All data are sequentially accessed, starting from the lowest bit.

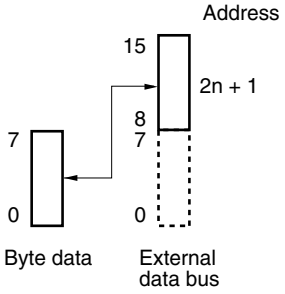
(1) Byte access (8 bits)

(a) 16-bit data bus width (little endian)

<1> Access to even address (2n)

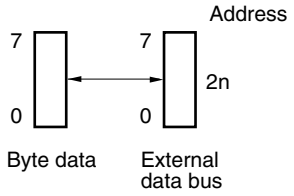


<2> Access to odd address (2n + 1)

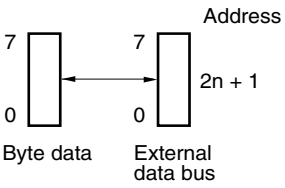


(b) 8-bit data bus width (little endian)

<1> Access to even address (2n)

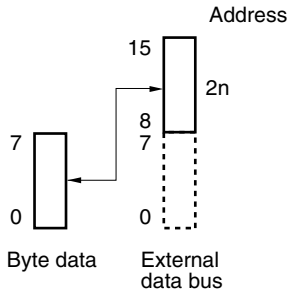


<2> Access to odd address (2n + 1)

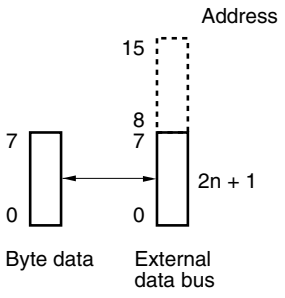


(c) 16-bit data bus width (big endian)

<1> Access to even address (2n)

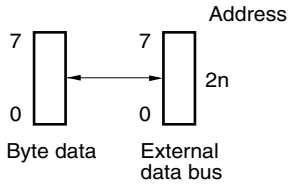


<2> Access to odd address (2n + 1)

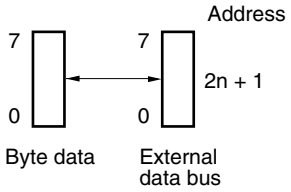


(d) 8-bit data bus width (big endian)

<1> Access to even address (2n)



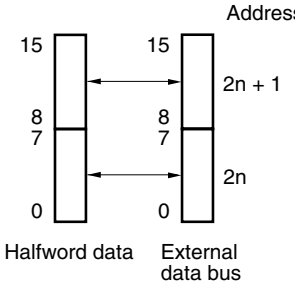
<2> Access to odd address (2n + 1)



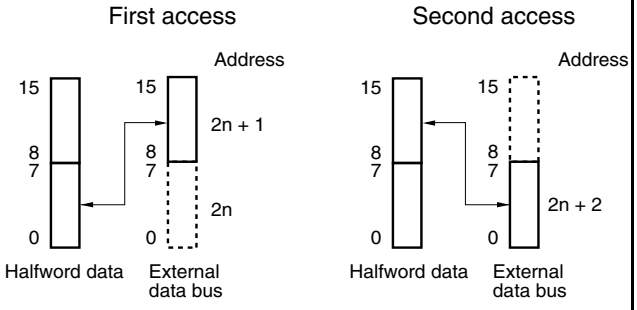
(2) Halfword access (16 bits)

(a) With 16-bit data bus width (little endian)

<1> Access to even address (2n)



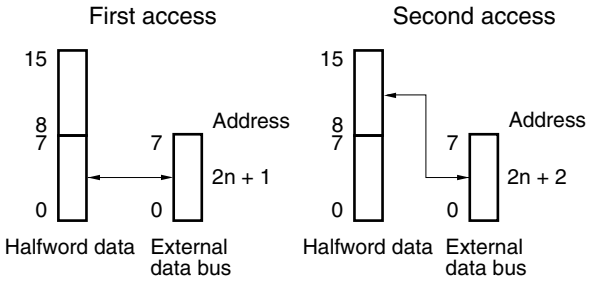
<2> Access to odd address (2n + 1)



(b) 8-bit data bus width (little endian)

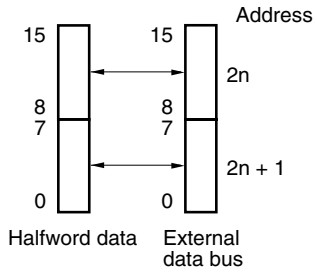
<1> Access to even address (2n)

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

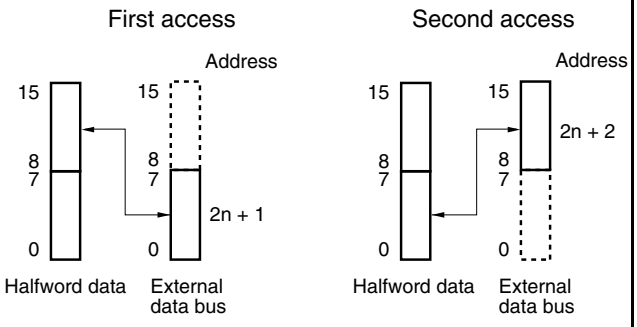


(c) With 16-bit data bus width (big endian)

<1> Access to even address (2n)



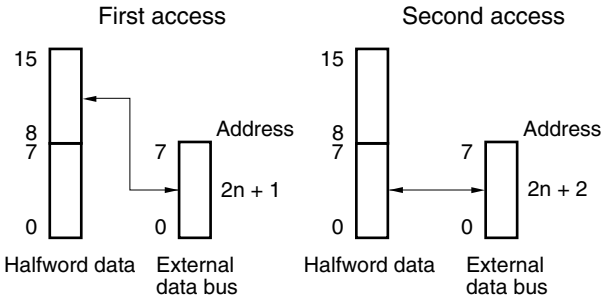
<2> Access to odd address (2n + 1)



(d) 8-bit data bus width (big endian)

<1> Access to even address (2n)

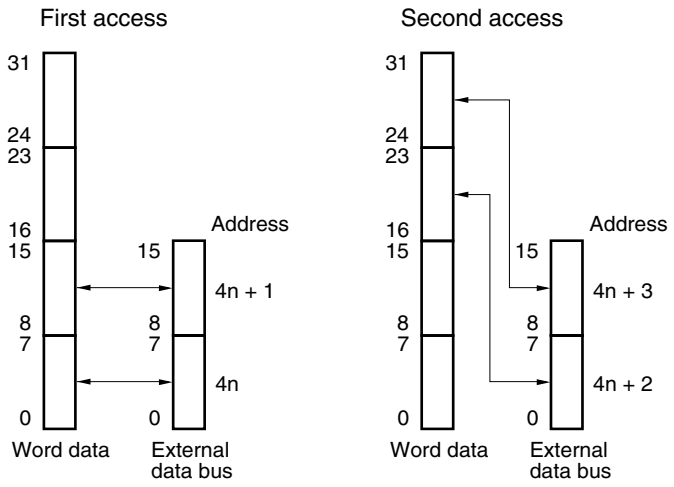
<2> Access to odd address (2n + 1)



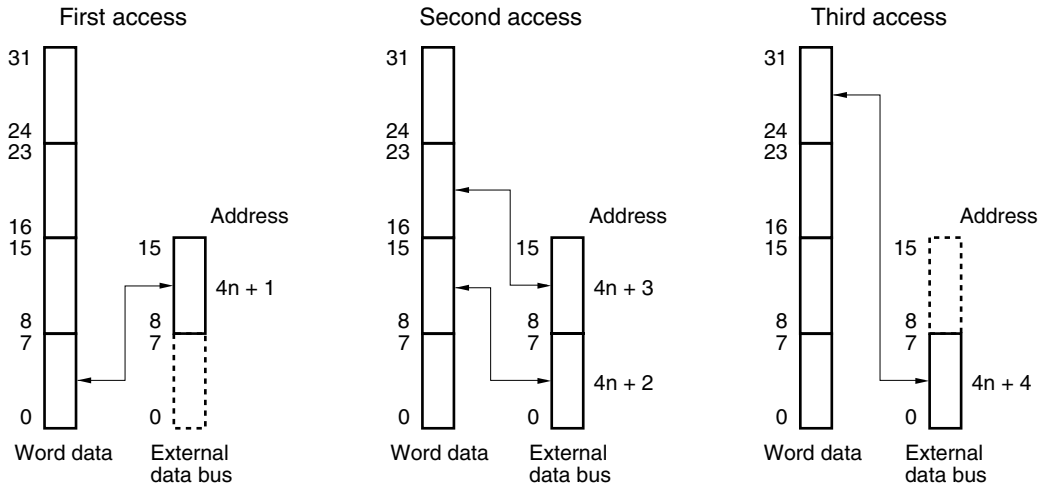
(3) Word access (32 bits)

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

<1> Access to address (4n)

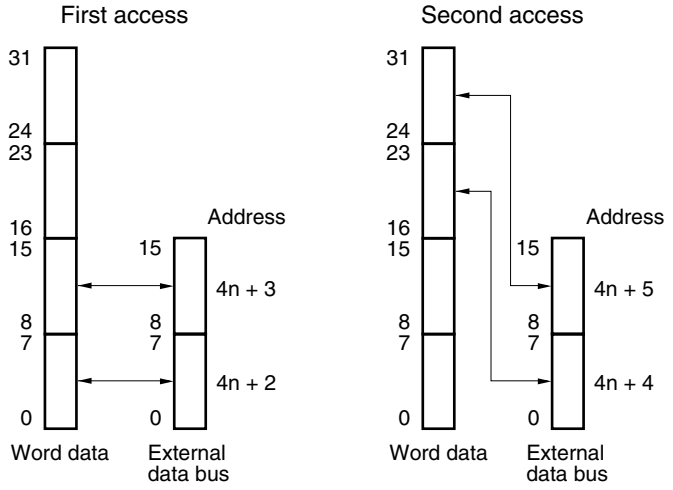


<2> Access to address (4n + 1)

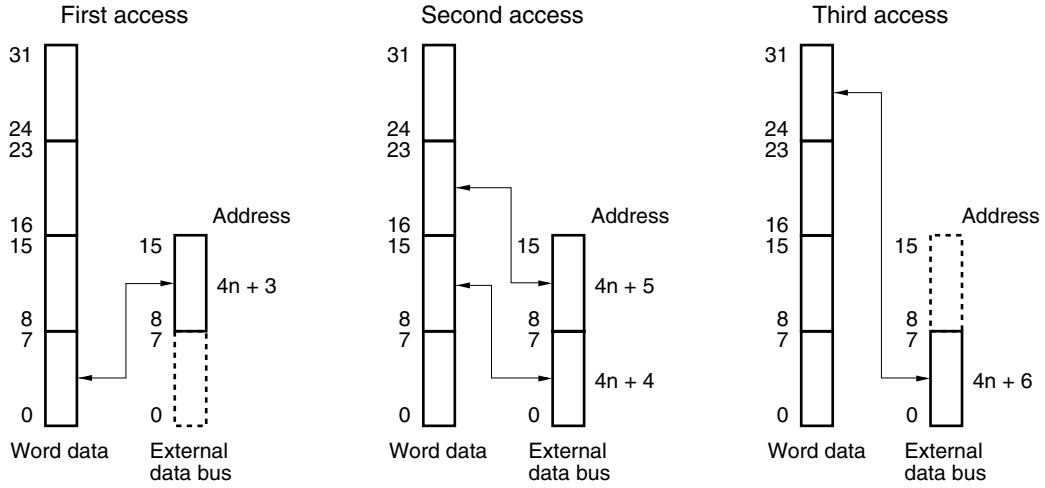


(a) 16-bit data bus width (little endian) (2/2)

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

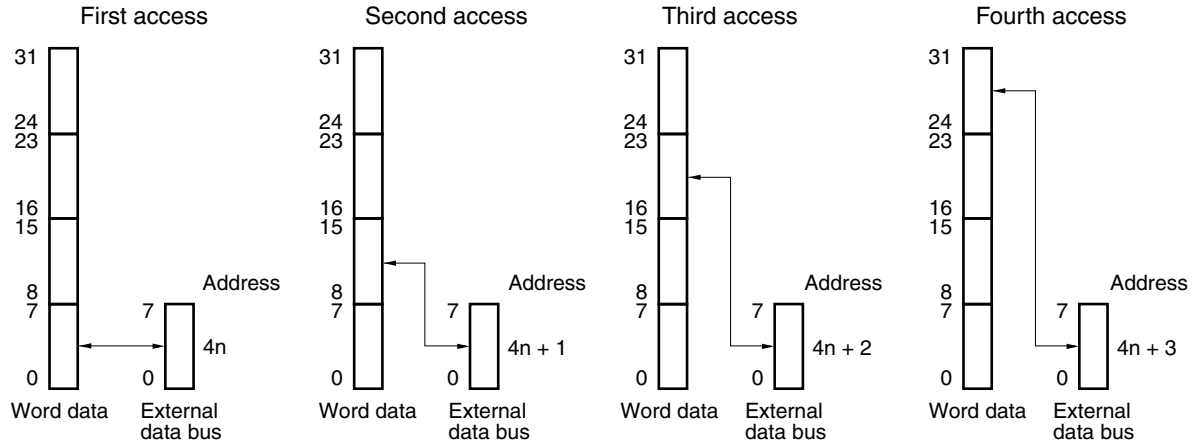


<4> Access to address (4n + 3)

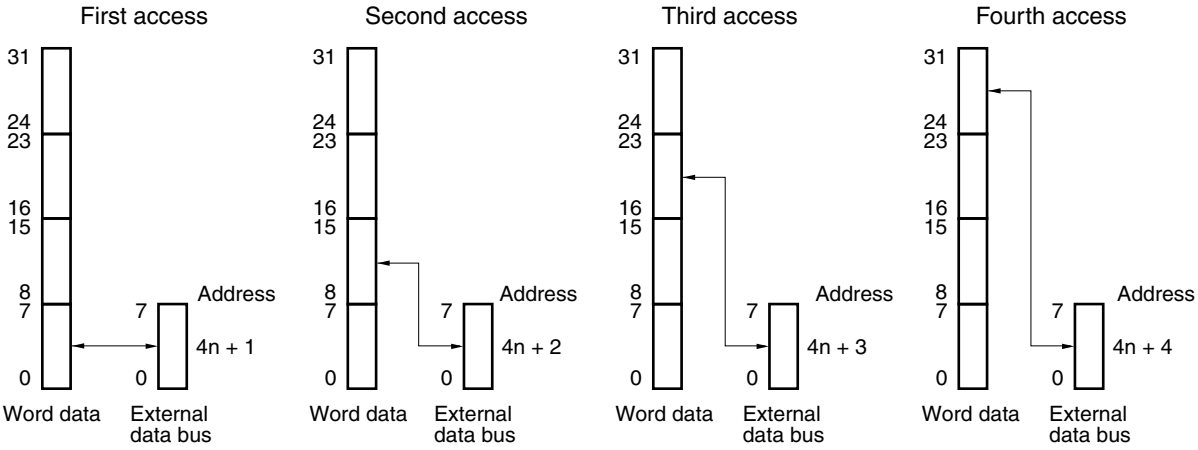


(b) 8-bit data bus width (little endian) (1/2)

<1> Access to address (4n)

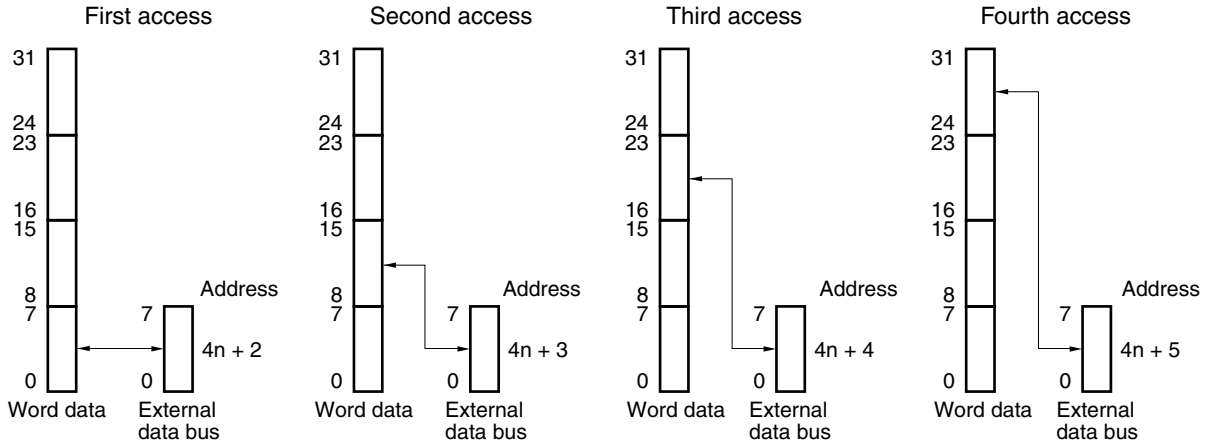


<2> Access to address (4n + 1)

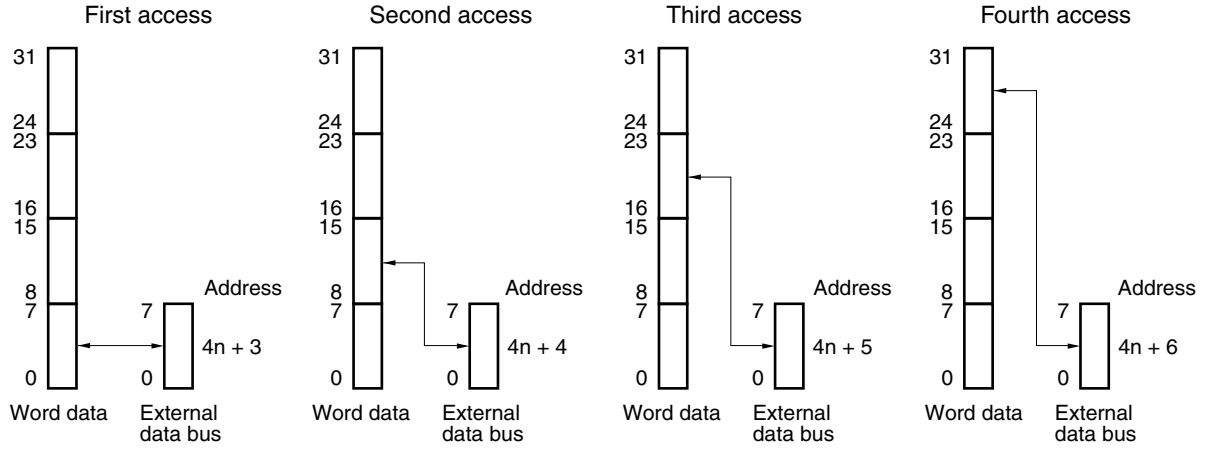


(b) 8-bit data bus width (little endian) (2/2)

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

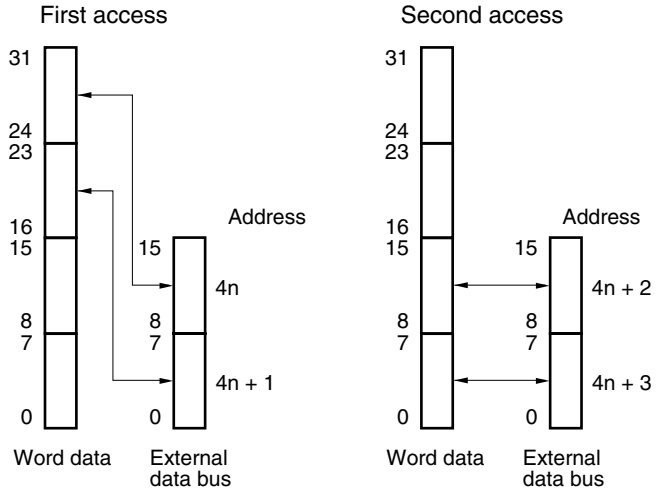


<4> Access to address (4n + 3)

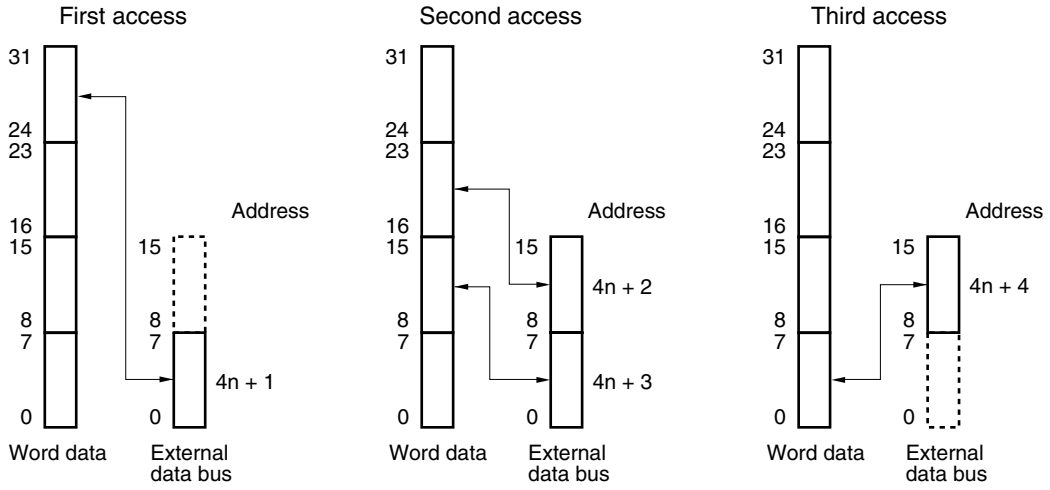


(c) 16-bit data bus width (big endian) (1/2)

<1> Access to address (4n)

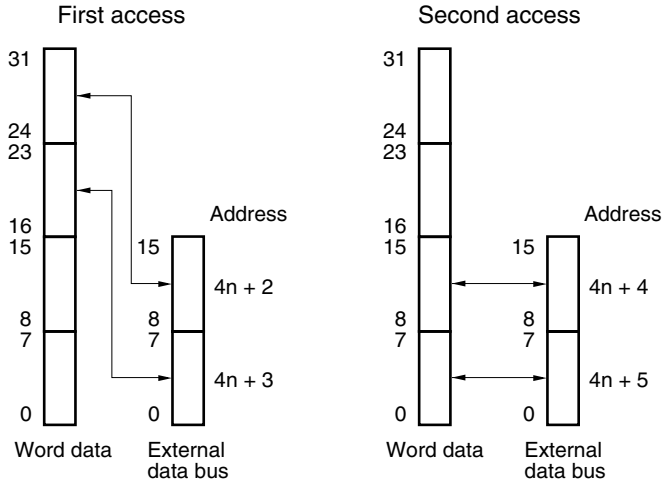


<2> Access to address (4n + 1)

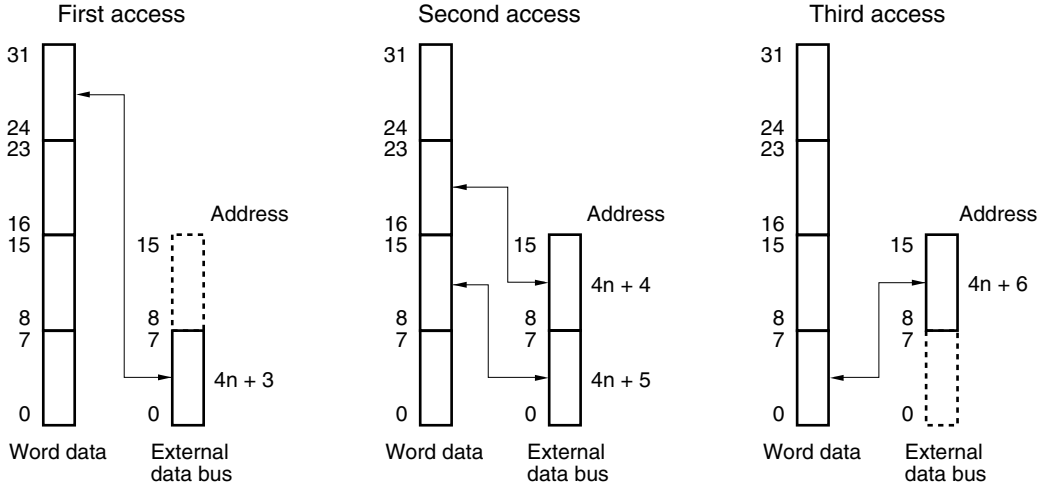


(c) 16-bit data bus width (big endian) (2/2)

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

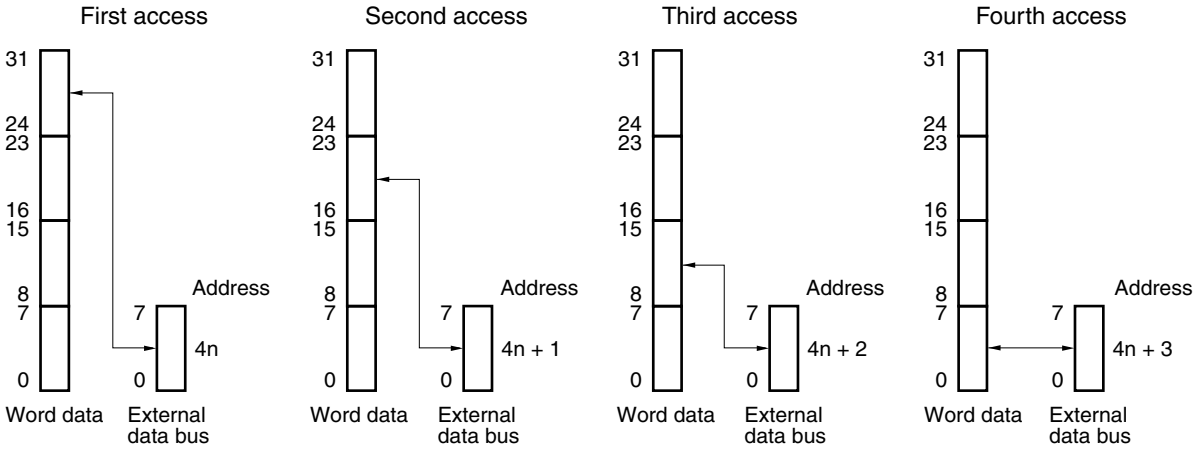


<4> Access to address (4n + 3)

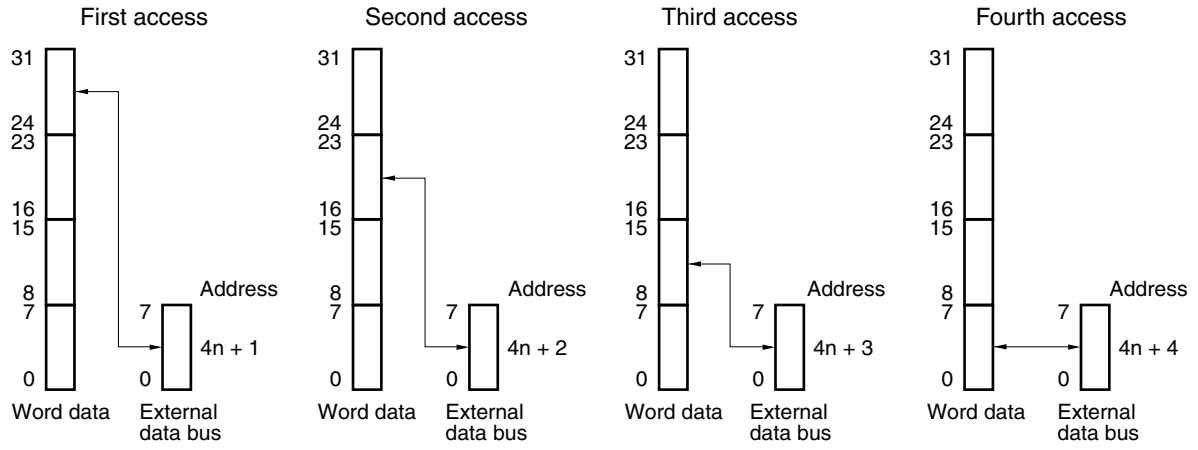


(d) 8-bit data bus width (big endian) (1/2)

<1> Access to address (4n)

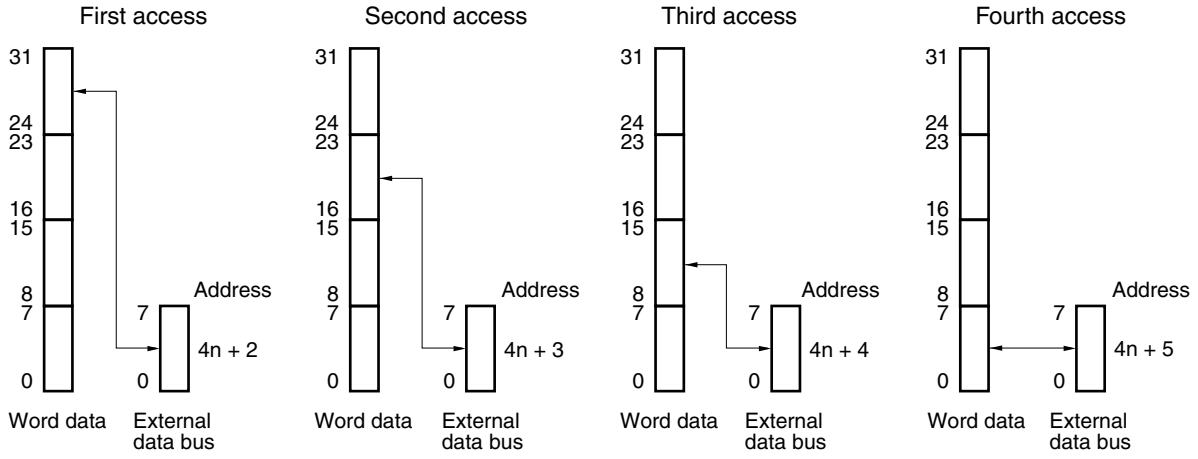


<2> Access to address (4n + 1)

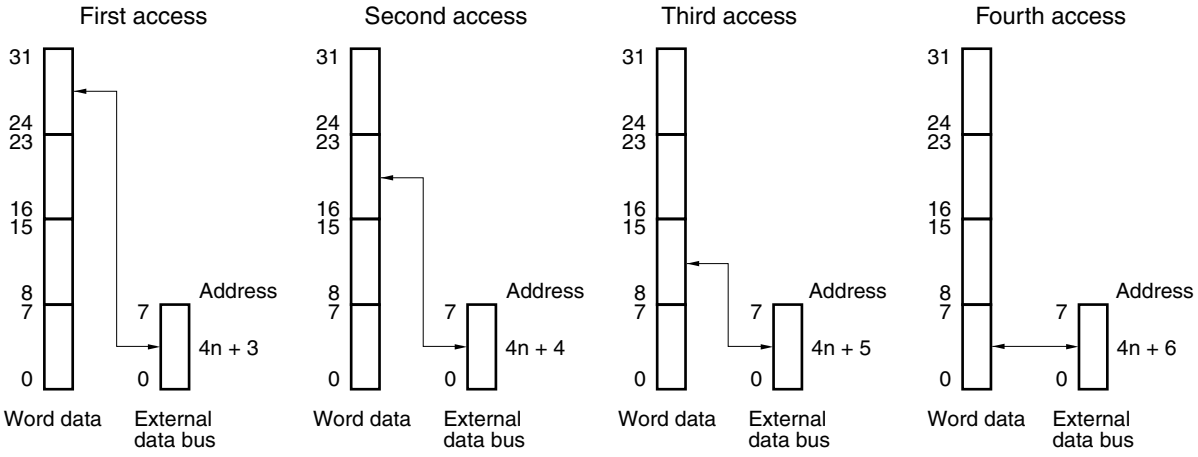


(d) 8-bit data bus width (big endian) (2/2)

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



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



5.6 Write Buffer Function

The V850E/MA3 has an on-chip write buffer of 4 words (128 bits). The write buffer stores data if a write cycle cannot be executed while the external bus is occupied[Note]. The next instructions are speculatively executed until the write buffer becomes full. The write buffer is valid for all the external memory areas. If a write request is generated while the write buffer is full, the next instruction execution is postponed until there is a vacancy in the write buffer.

While data is being stored in the write buffer (when a write operation to the external memory has not been completed), DMA flyby and bus hold requests are not acknowledged (DMA flyby and bus hold requests are acknowledged and an enable signal is generated after all the data of the write buffers has been written to the external memory).

Note The external bus is occupied when there is a bus cycle currently under execution.

Cautions 1. Because the write buffer consists of four stages, the write buffer becomes full after 4 bytes (32 bits) when a byte write operation is executed. Similarly, the conditions under which the write buffer becomes full vary due to an address miss-align access, etc.

2. When data is written to an external device, the write operation to the external device may not be executed even when a CPU write operation has been completed by the write buffer. The CPU can access the on-chip peripheral I/O registers after the write operation has been completed even if the write buffer exists. Therefore, if it is necessary to change the value of an on-chip peripheral I/O register after completion of execution of an external memory cycle, write 00H to the WAS register before writing the on-chip peripheral I/O register whose value is to be changed. When writing an on-chip peripheral I/O register other than the WAS register without writing 00H to the WAS register, the register value may be changed before completion of the external memory cycle.

3. If an read access to the external device occurs when data exist in the write buffer, reading from the external device is executed after the writing all data in the write buffer to the external device.

4. During 2-cycle transfer that writes data to the external device, the write operation to the external device may not be completed even if DCHCn.TCn bit = 1 (DMA transfer completion) is read by the write buffer (n = 0 to 3). If it is necessary to change the value of an on-chip peripheral I/O register after completion of DMA transfer (completion of a write operation to the external device), perform either of the following operations.

- **Monitor the $\overline{\text{TCn}}$ signal (the $\overline{\text{TCn}}$ signal becomes active in synchronization with a write operation to the external device).**
- **After detecting setting (to 1) of the DCHCn.TCn bit, write 00H to the WAS register and then change the value of the on-chip peripheral I/O register. If the value of an on-chip peripheral I/O register other than the WAS register is changed without writing 00H to the WAS register, the value of the on-chip peripheral I/O register may be changed before completion of DMA transfer.**

(1) Write access synchronization control register (WAS)

When an external device is written, even if the write operation by the CPU via the write buffer is complete, writing to the external device may not be complete. The WAS register is used to complete writing all data in the write buffer to the external device. See **5.6 Write Buffer Function** for details.

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

After reset: Undefined W Address: FFFFF49CH

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

Caution Be sure to write 00H to the WAS register.
Operation cannot be guaranteed if value other than 00H is written.

5.7 Bus Clock Control Function

(1) Bus mode control register (BMC)

The BMC register is used to set the division ratio of the bus clock (BUSCLK) with respect to the internal system clock.

When this register is written, BUSCLK stops once at the low level. BUSCLK resumes operation using the divided clock that is set after BUSCLK was stopped. While BUSCLK is stopped, the operation of the RFSn register of SDRAM also stops (n = 1, 3, 4, 6).

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

Reset input sets this register to 01H.

Cautions 1. Write to the BMC register after reset, and then do not change the set value.

2. Be sure to write to the BMC register after setting the VSWC register to x7H (x: VSWC register value before the set value of the BMC register is changed). After changing the set value of the BMC register, re-set the value of the VSWC register to the value before the set value of the BMC register was changed.

[Setting procedure]

<1> Save the set value of the VSWC register.

<2> Set x7H to the VSWC register (x: value of the VSWC register before the set value of the BMC register is changed).

<3> Set the BMC register.

(In the meantime, BUSCLK once stops.)

<4> Restore the saved set value of the VSWC register.

3. The maximum operating frequency of the external bus interface of the V850E/MA3 is 50 MHz. Do not set a bus clock (BUSCLK) or perform a setting sequence that may exceed this frequency.

After reset: 01H R/W Address: FFFFF498H

	7	6	5	4	3	2	1	0
BMC	0	0	0	0	0	0	CKM1	CKM0

CKM1	CKM0	Specification of division ratio of BUSCLK with respect to f_{CLK}
0	0	f_{CLK}
0	1	$f_{CLK}/2$
1	0	$f_{CLK}/3$
1	1	$f_{CLK}/4$

Caution Be sure to clear bits 7 to 2 to "0". The operation cannot be guaranteed if these bits are set to 1.

Remark f_{CLK}: Internal system clock

5.8 Wait Function

5.8.1 Programmable wait function

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

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

The number of wait states can be programmed by using the DWC0 and DWC1 registers. Immediately after system reset, 7 data wait states are inserted for all the blocks.

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

Reset input sets these registers to 7777H.

Note Separate bus mode: SRAM read/write cycle
Page ROM read cycle (off page)
Multiplexed bus mode: SRAM read/write cycle

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

2. The setting of the DWC0 and DWC1 registers is invalid in the following cases (wait control is performed by each memory controller).
- **On-page access to page ROM**
- **Access to SDRAM**

3. Write to the DWC0 and DWC1 registers after reset, and then do not change the set values. Also, do not access an external memory area until the initial settings of the DWC0 and DWC1 registers are complete. However, external memory areas whose initial settings are complete may be accessed.

After reset: 7777H R/W Address: DWC0 FFFFF484H, DWC1 FFFFF486H

	15	14	13	12	11	10	9	8
DWC0	0	DW32	DW31	DW30	0	DW22	DW21	DW20
$\overline{\text{CSn}}$ signal			$\overline{\text{CS3}}$				$\overline{\text{CS2}}$	

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

	15	14	13	12	11	10	9	8
DWC1	0	DW72	DW71	DW70	0	DW62	DW61	DW60
$\overline{\text{CSn}}$ signal			$\overline{\text{CS7}}$				$\overline{\text{CS6}}$	

	7	6	5	4	3	2	1	0
	0	DW52	DW51	DW50	0	DW42	DW41	DW40
$\overline{\text{CSn}}$ signal			$\overline{\text{CS5}}$				$\overline{\text{CS4}}$	

DWn2	DWn1	DWn0	Specification of number of wait states to be inserted in CSn space (n = 0 to 7)
0	0	0	Not inserted
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

(2) Address setup wait control register (ASC)

The V850E/MA3 can insert address setup wait states at the beginning of a read/write cycle of SRAM cycle and a read cycle of page ROM cycle in the separate mode, and at the beginning of a read/write cycle of SRAM in the multiplexed bus mode (the setting of the ASC register is invalid in the SDRAM cycle).

The number of address setup wait states can be set for each CS space by using the ASC register.

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

Reset input sets this register to FFFFH.

Cautions 1. An address setup wait state is not inserted in the internal ROM area, internal RAM area, and on-chip peripheral I/O area.

2. The external wait function that is effected by the $\overline{\text{WAIT}}$ pin is invalid when an address setup wait state is inserted.

3. Write to the ASC register after reset, and then do not change the set value.

4. The address setup wait setting value is valid during the DMA flyby transfer.

After reset: FFFFH R/W Address: FFFFF48AH

	15	14	13	12	11	10	9	8
ASC	AC71	AC70	AC61	AC60	AC51	AC50	AC41	AC40
$\overline{\text{CSn}}$ signal		$\overline{\text{CS7}}$		$\overline{\text{CS6}}$		$\overline{\text{CS5}}$		$\overline{\text{CS4}}$

	7	6	5	4	3	2	1	0
	AC31	AC30	AC21	AC20	AC11	AC10	AC01	AC00
$\overline{\text{CSn}}$ signal		$\overline{\text{CS3}}$		$\overline{\text{CS2}}$		$\overline{\text{CS1}}$		$\overline{\text{CS0}}$

ACn1	ACn0	Specification of number of address setup wait states in CSn space (n = 0 to 7)
0	0	Not inserted
0	1	1
1	0	2
1	1	3

(3) Address hold wait control register (AHC)

The AHC register inserts an address hold wait state between the T1 cycle and T2 cycle of a read/write cycle of the SRAM cycle in the multiplexed bus mode.

The number of address hold wait states to be inserted can be specified for each CS space by using this register.

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

Reset input sets this register to FFFFH.

Cautions 1. An address hold wait state is not inserted in the internal ROM area, internal RAM area, and on-chip peripheral I/O area.

2. The AHC register can be set only in the multiplexed bus mode.

3. The external wait function that is effected by the $\overline{\text{WAIT}}$ pin is invalid when an address hold wait state is inserted.

4. Write to the AHC register after reset, and then do not change the set values.

After reset: FFFFH R/W Address: FFFFF49EH

	15	14	13	12	11	10	9	8
AHC	AH71	AH70	AH61	AH60	AH51	AH50	AH41	AH40
$\overline{\text{CSn}}$ signal		$\overline{\text{CS7}}$		$\overline{\text{CS6}}$		$\overline{\text{CS5}}$		$\overline{\text{CS4}}$

	7	6	5	4	3	2	1	0
	AH31	AH30	AH21	AH20	AH11	AH10	AH01	AH00
$\overline{\text{CSn}}$ signal		$\overline{\text{CS3}}$		$\overline{\text{CS2}}$		$\overline{\text{CS1}}$		$\overline{\text{CS0}}$

AHn1	AHn0	Specification of number of address hold wait states in CSn space (n = 0 to 7)
0	0	Not inserted
0	1	1
1	0	2
1	1	3

(4) Bus cycle period control register (BCP)

The V850E/MA3 can enable or disable the operations of $\overline{\text{IORD}}$ and $\overline{\text{IOWR}}$ in the SRAM, external ROM, and external I/O cycles.

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

Reset input clears this register to 00H.

Cautions 1. The $\overline{\text{IORD}}$ and $\overline{\text{IOWR}}$ signals are output, regardless of the setting of the IOEN bit, during flyby DMA transfer to transfer data to/from the SRAM, external ROM, and external I/O. The setting of the IOEN bit is meaningless in the page ROM cycle.

2. Write to the BCP register after reset, and then do not change the set values.

After reset: 00H R/W Address: FFFFF48CH

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

IOEN	$\overline{\text{IORD}}$, $\overline{\text{IOWR}}$ operation enable in the SRAM, external ROM, and external I/O cycles
0	Disables the operations of $\overline{\text{IORD}}$, $\overline{\text{IOWR}}$.
1	Enables the operations of $\overline{\text{IORD}}$, $\overline{\text{IOWR}}$.

Caution Be sure to clear bits 7 to 4 and 2 to 0 to "0". The operation cannot be guaranteed if these bits are set to 1.

(5) DMA flyby transfer wait control register (FWC)

The FWC register sets the number of data wait states for channel n during DMA flyby transfer (n = 0 to 3).

The set value of this register becomes valid during DMA flyby transfer, and the set values of the DWC0, DWC1, and PRC registers become invalid.

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

Reset input sets this register to 7777H.

Cautions 1. Wait states cannot be programmed for accessing the internal ROM and internal RAM areas, and these areas are always accessed without a wait state. Wait states cannot be programmed for accessing the on-chip peripheral I/O area, and only the wait states that are inserted when each peripheral function accesses this area are controlled.

2. Write to the FWC register after reset, and then do not change the set values.

After reset: 7777H R/W Address: FFFFF494H

	15	14	13	12	11	10	9	8
FWC	0	FW32	FW31	FW30	0	FW22	FW21	FW20
DMA channel		DMA3				DMA2		

	7	6	5	4	3	2	1	0
	0	FW12	FW11	FW10	0	FW02	FW01	FW00
DMA channel		DMA1				DMA0		

FWn2	FWn1	FWn0	Specification of number of data waits in CSn space during DMA flyby transfer (n = 0 to 3)
0	0	0	Not inserted
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

5.8.2 External wait function

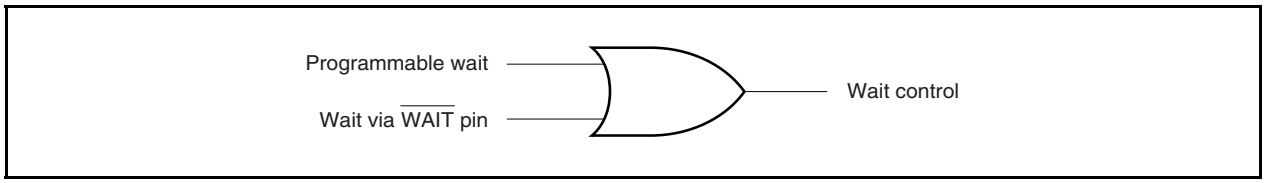
To synchronize an extremely slow external memory, I/O, or asynchronous system, any number of wait states can be inserted in the bus cycle by using the external wait pin ($\overline{\text{WAIT}}$).

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

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

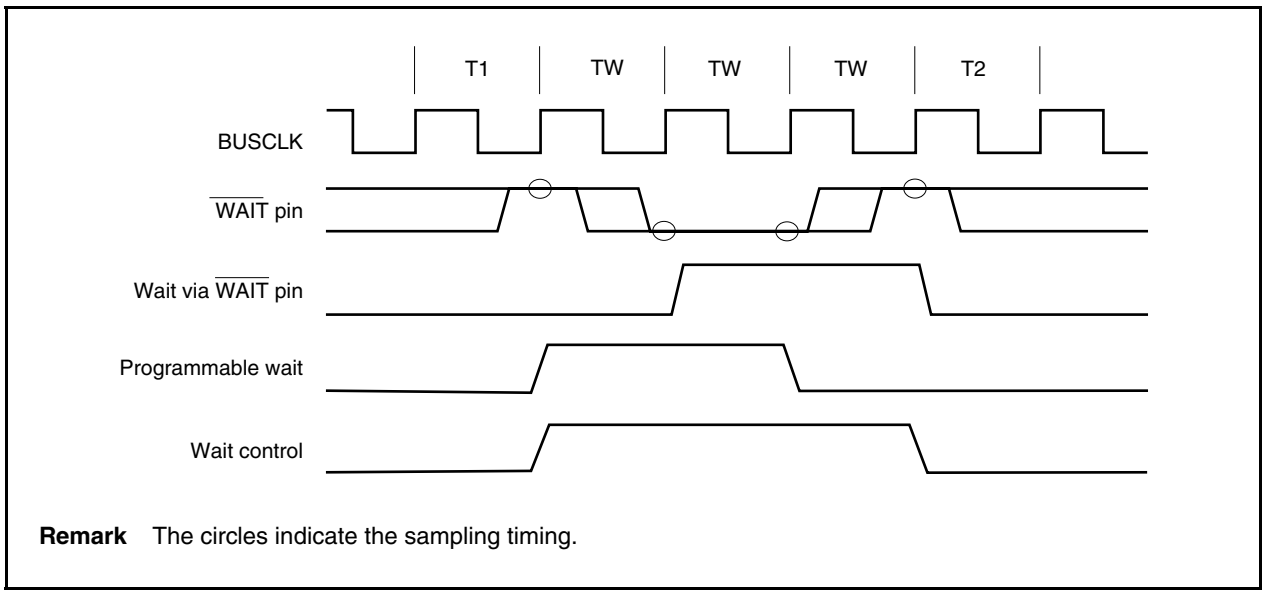
5.8.3 Relationship between programmable wait and external wait

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



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

Figure 5-5. Example of Inserting Wait States



Remark The circles indicate the sampling timing.

5.8.4 Bus cycle for which wait function is valid

The V850E/MA3 can specify the number of wait states in accordance with the type of the memory specified for each memory block. The bus cycle for which the wait function is valid and the register that specifies the number of wait states are shown below.

Table 5-2. Bus Cycle in Which Wait Function Is Valid

(a) In separate bus mode

Bus Cycle		Wait Type	Setting of Programmable Wait			Wait by $\overline{\text{WAIT}}$ Pin
			Register	Bit	Number of Waits	
SRAM, external ROM, external I/O cycle		Address setup wait	ASC	ACn1, ACn0	0 to 3	× (invalid)
		Data wait	DWC0, DWC1	DWn2 to DWn0	0 to 7	√ (valid)
Page ROM cycle		Address setup wait	ASC	ACn1, ACn0	0 to 3	× (invalid)
	Off-page	Data access wait	DWC0, DWC1	DWn2 to DWn0	0 to 7	√ (valid)
	On-page	Data access wait	PRC	PRW2 to PRW0	0 to 7	√ (valid)
SDRAM cycle		Row address precharge	SCRm	BCWm1, BCWm0	1 to 3	× (invalid)
DMA flyby transfer cycle	External I/O ↔ SRAM	Flyby transfer wait	FWC	FWa2 to FWa0	0 to 7	√ (valid)
		Address setup wait	ASC	ACn1, ACn0	0 to 3	√ (invalid)

(b) In multiplexed bus mode

Bus Cycle	Wait Type	Setting of Programmable Wait			Wait by $\overline{\text{WAIT}}$ Pin
		Register	Bit	Number of Waits	
SRAM, external ROM cycle	Address setup wait	ASC	ACn1, ACn0	0 to 3	× (invalid)
	Address hold wait	AHC	AHn1, AHn0	0 to 3	× (invalid)
	Data access wait	DWC0, DWC1	DWn2 to DWn0	0 to 7	√ (valid)

Remark n = 0 to 7
m = 1, 3, 4, 6
a = 0 to 3

5.9 Idle State Insertion Function

(1) Bus cycle control register (BCC)

To establish interfacing with a low-speed memory easily and to secure the data output float delay time during a read access to each CS space, the specified number of idle states (TI) can be inserted after the T2 state in the bus cycle to be started. The bus cycle following the T2 state starts after the inserted idle state.
The timing of inserting an idle state is as follows.

(a) In separate bus mode

- After read cycle or write cycle of SRAM, external I/O, and external ROM
- After read cycle of page ROM
- After read cycle of SDRAM

(b) In multiplexed bus mode

- After read cycle or write cycle of SRAM and external ROM

Whether the idle state is to be inserted can be programmed by using the BCC register. An idle state is inserted for all the blocks immediately after system reset. For the timing of inserting the idle state, see the access timing of each memory in chapter 6.
This register can be read or written in 16-bit units.
Reset input sets this register to FFFFH.

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

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

3. The chip select signal ($\overline{CSn}$) is not asserted in the idle state (n = 0 to 7).

After reset: FFFFH R/W Address: FFFFF488H

	15	14	13	12	11	10	9	8
BCC	BC71	BC70	BC61	BC60	BC51	BC50	BC41	BC40
$\overline{CSn}$ signal	$\overline{CS7}$		$\overline{CS6}$		$\overline{CS5}$		$\overline{CS4}$	

	7	6	5	4	3	2	1	0
	BC31	BC30	BC21	BC20	BC11	BC10	BC01	BC00
$\overline{CSn}$ signal	$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$	

BCn1	BCn0	Specification of number of idle states in CSn space (n = 0 to 7)
0	0	Not inserted
0	1	1
1	0	2
1	1	3

(2) DMA flyby transfer idle control register (FIC)

The FIC register specifies the number of idle states during DMA flyby transfer for DMA channel n (n = 0 to 3). The idle state is inserted at the end of the DMA flyby transfer. During the DMA flyby transfer, the set value of this register becomes valid, and the set value of the bus cycle control register (BCC) becomes invalid.

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

Reset input sets this register to 3333H.

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

2. Write to the FIC register after reset, and then do not change the set values.

After reset: 3333H R/W Address: FFFFF496H

	15	14	13	12	11	10	9	8
FIC	0	0	FI31	FI30	0	0	FI21	FI20
DMA channel			DMA3				DMA2	

	7	6	5	4	3	2	1	0
	0	0	FI11	FI10	0	0	FI01	FI00
DMA channel			DMA1				DMA0	

FIn1	FIn0	Specification of number of idle states in CSn space during DMA flyby transfer (n = 0 to 3)
0	0	Not inserted
0	1	1
1	0	2
1	1	3

5.10 Bus Hold Function

5.10.1 Functional outline

The $\overline{\text{HLDAK}}$ and $\overline{\text{HLDRQ}}$ functions are valid if the PCM2 and PCM3 pins are set to their alternate functions.

When the $\overline{\text{HLDRQ}}$ pin is asserted (low level), indicating that another bus master has requested bus mastership, the external address/data bus and each strobe pin go into a high-impedance state and the bus is released (bus hold status). If the request for the bus mastership is cleared and the $\overline{\text{HLDRQ}}$ pin is deasserted (high level), driving these pins is started again.

During the bus hold period, execution of the internal operation of the V850E/MA3 is continued until the external memory or an on-chip peripheral I/O register is accessed.

The bus hold status is indicated by assertion of the $\overline{\text{HLDAK}}$ pin (low level). It takes the $\overline{\text{HLDAK}}$ pin at least two clocks to be asserted (low level) after the $\overline{\text{HLDRQ}}$ pin is asserted (low level).

The bus hold function enables the configuration of a multiprocessor type system in which two or more bus masters exist.

Status	Data Bus Width	Access Type	Timing at Which Bus Hold Request Is Not Acknowledged
CPU bus lock	16 bits	Word access to even address	Between first and second access
		Word access to odd address	Between first and second access
			Between second and third access
		Halfword access to odd address	Between first and second access
	8 bits	Word access	Between first and second access
			Between second and third access
			Between third and fourth access
		Halfword access	Between first and second access
Read-modify-write access of bit manipulation instruction	–	–	Between read access and write access

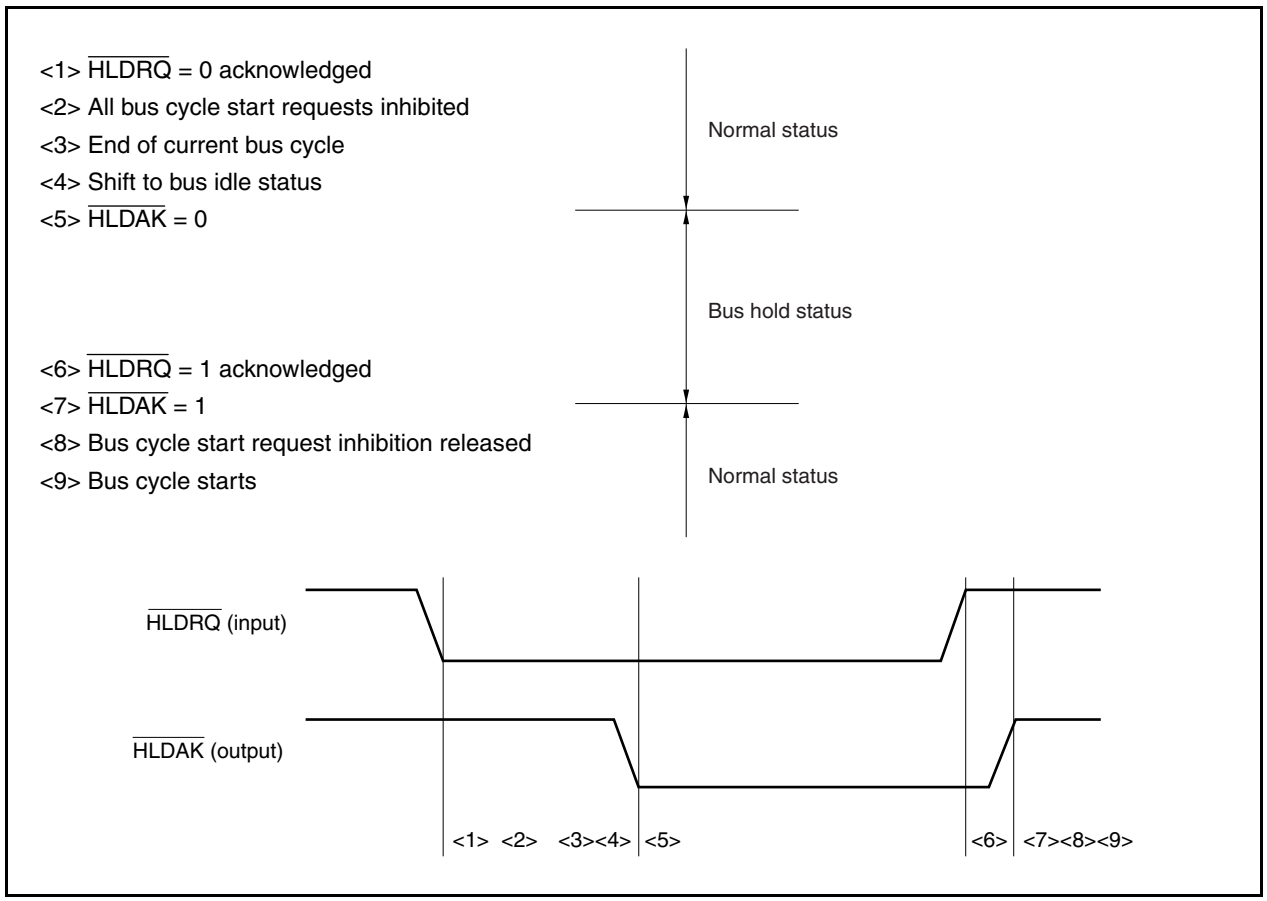
Cautions 1. If the external bus master accesses SDRAM during bus hold, make sure that the external bus master executes the all bank precharge command.

To clear the bus hold status, the CPU always executes the all bank precharge command. The external bus master must not change the value of the command register of SDRAM during bus hold.

2. The $\overline{\text{HLDRQ}}$ function is invalid during the reset period. The $\overline{\text{HLDAK}}$ pin is asserted immediately after the $\overline{\text{RESET}}$ pin has been deasserted after the $\overline{\text{RESET}}$ and $\overline{\text{HLDRQ}}$ pins were asserted at the same time or after one clock address cycle is inserted. If an external bus master other than the V850E/MA3 is connected, perform arbitration upon power application by using the $\overline{\text{RESET}}$ signal.

5.10.2 Bus hold procedure

The bus hold status transition procedure is shown below.



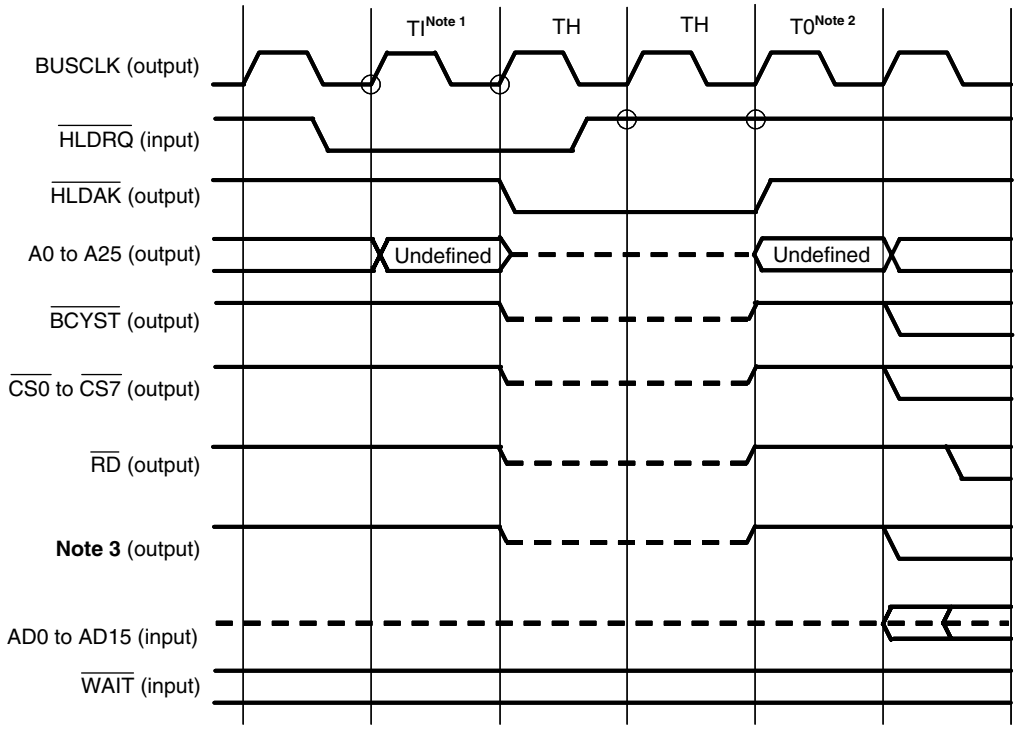
5.10.3 Operation in power save mode

Because the internal system clock is stopped in the software STOP and IDLE modes, the bus hold status is not entered even if the $\overline{\text{HLDRQ}}$ pin is asserted.

In the HALT mode, the $\overline{\text{HLDAK}}$ pin is asserted as soon as the $\overline{\text{HLDRQ}}$ pin is asserted, and the bus hold status is entered. When the $\overline{\text{HLDRQ}}$ pin is later deasserted, the $\overline{\text{HLDAK}}$ pin is also deasserted, and the bus hold status is cleared and return to the HALT mode.

5.10.4 Bus hold timing

(1) When bus hold request is issued without bus cycle generated (BMC register = 01H)



Notes 1. Idle state (TI) independent of BCC register setting

2. State (T0) inserted between bys cycles

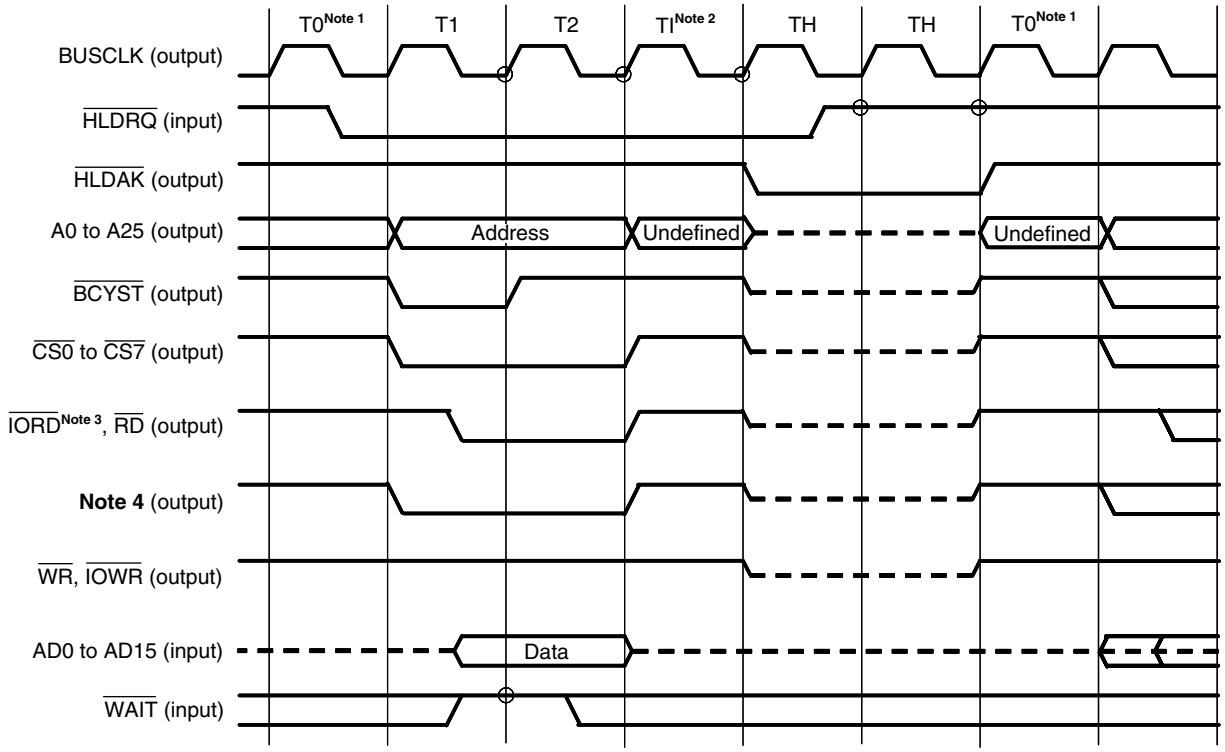
3. LWR/LBE/LDQM, UWR/UBE/UDQM, WE/WR, IORD, IOWR, SDRAS, SDCAS

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

5.10.5 Bus hold timing (SRAM)

(1) SRAM (During read, no idle states inserted, BMC register = 01H)



Notes 1. State (T0) inserted between bys cycles

2. Idle state (TI) independent of BCC register setting

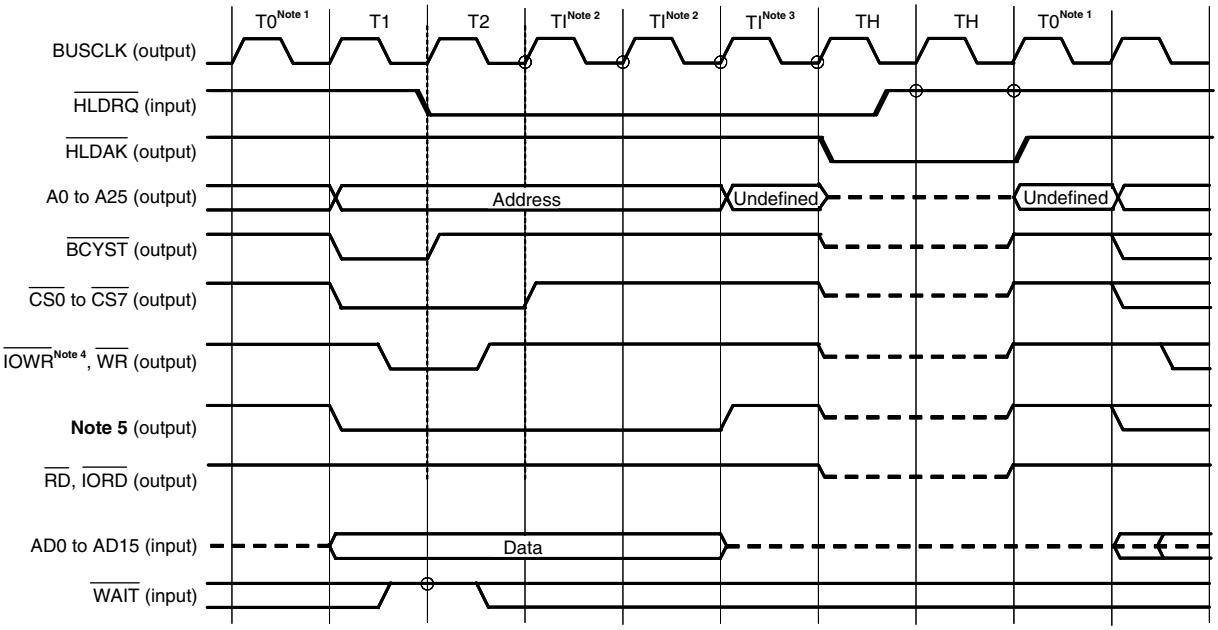
3. When BCP.IOEN bit is set (1).

4. UBE, LBE

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

(2) SRAM (During write, two idle states inserted, BMC register = 01H)



Notes 1. State (T0) inserted between bys cycles

2. Idle state (TI) inserted according to the BCC register setting

3. Idle state (TI) independent of the BCC register setting

4. When BCP.IOEN bit is set (1).

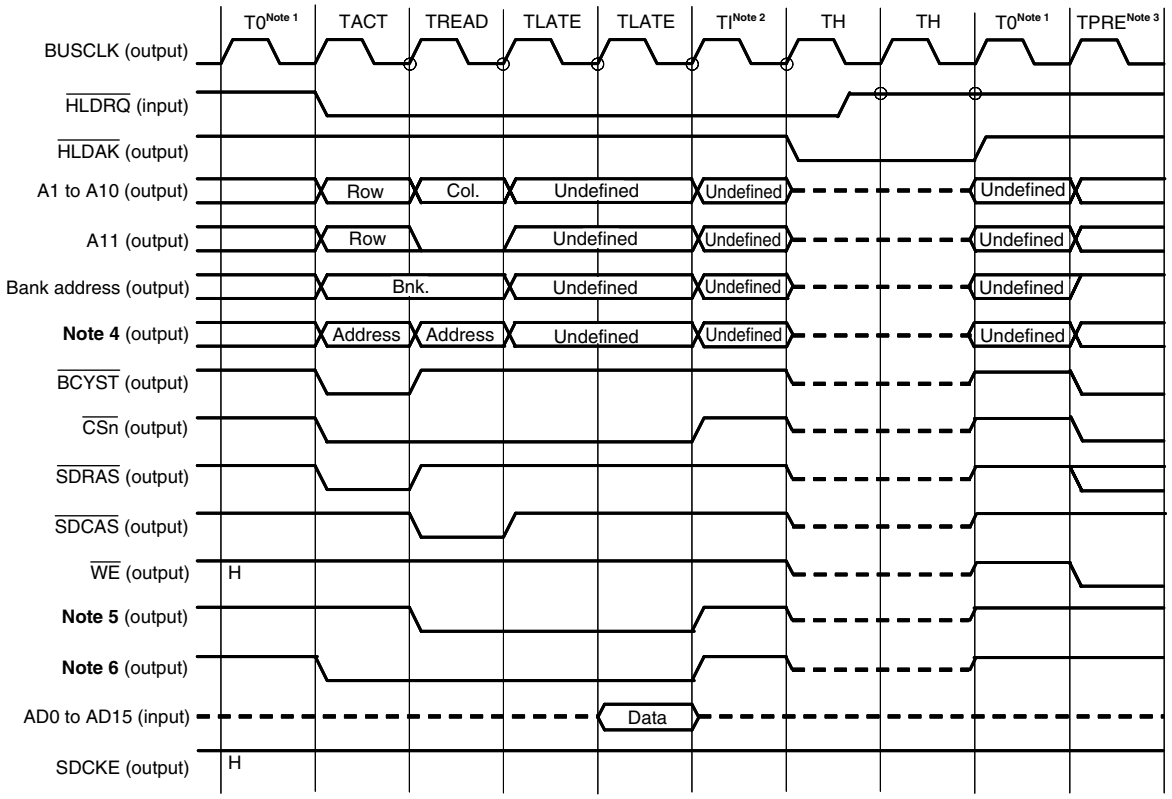
5. $\overline{UBE}$, $\overline{LBE}$

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

5.10.6 Bus hold timing (SDRAM)

(1) SDRAM (During read, latency = 2, no idle states inserted, 16-bit bus width halfword access, BMC register = 01H)



Notes 1. State (T0) inserted between bys cycles

2. Idle state (TI) inserted according to the BCC register setting

3. Always executes the all bank precharge command.

4. Addresses other than bank address, A11, and A1 to A10

5. When $\overline{\text{xWR}}$ output mode/xDQM output mode is set (PFCCT.PFCCTm bit = 0)

6. When $\overline{\text{xBE}}$ output mode/xDQM output mode is set (PFCCT.PFCCTm bit = 1)

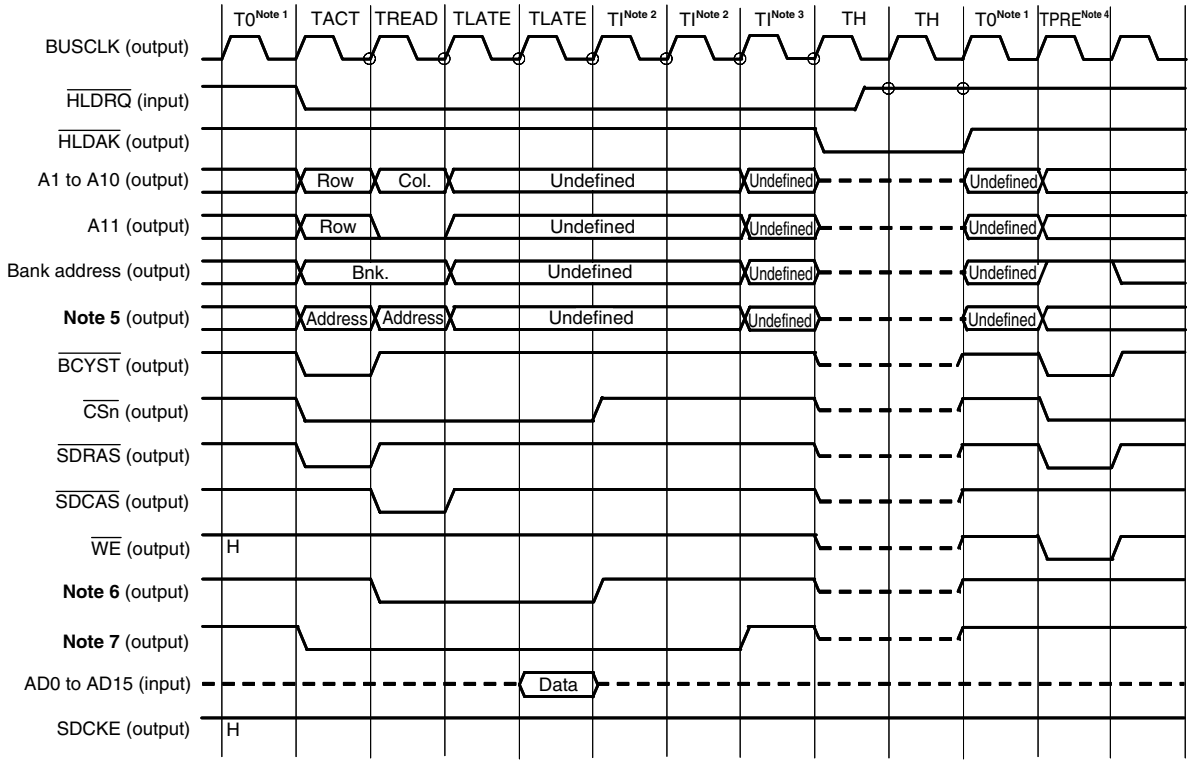
Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

3. n = 1, 3, 4, or 6
m = 0 or 1
xx = U or L

4. Bnk.: Bank address
Col.: Column address
Row: Row address

(2) SDRAM (During read, latency = 2, two idle states inserted, 16-bit bus width halfword access, BMC register = 01H)



Notes 1. State (T0) inserted between bys cycles

2. Idle state (TI) inserted according to the BCC register setting

3. Idle state (TI) independent of the BCC register setting

4. Always executes the all bank precharge command.

5. Addresses other than bank address, A11, and A1 to A10

6. When $\overline{\text{xWR}}$ output mode/xDQM output mode is set (PFCCT.PFCCTm bit = 0)

7. When $\overline{\text{xBE}}$ output mode/xDQM output mode is set (PFCCT.PFCCTm bit = 1)

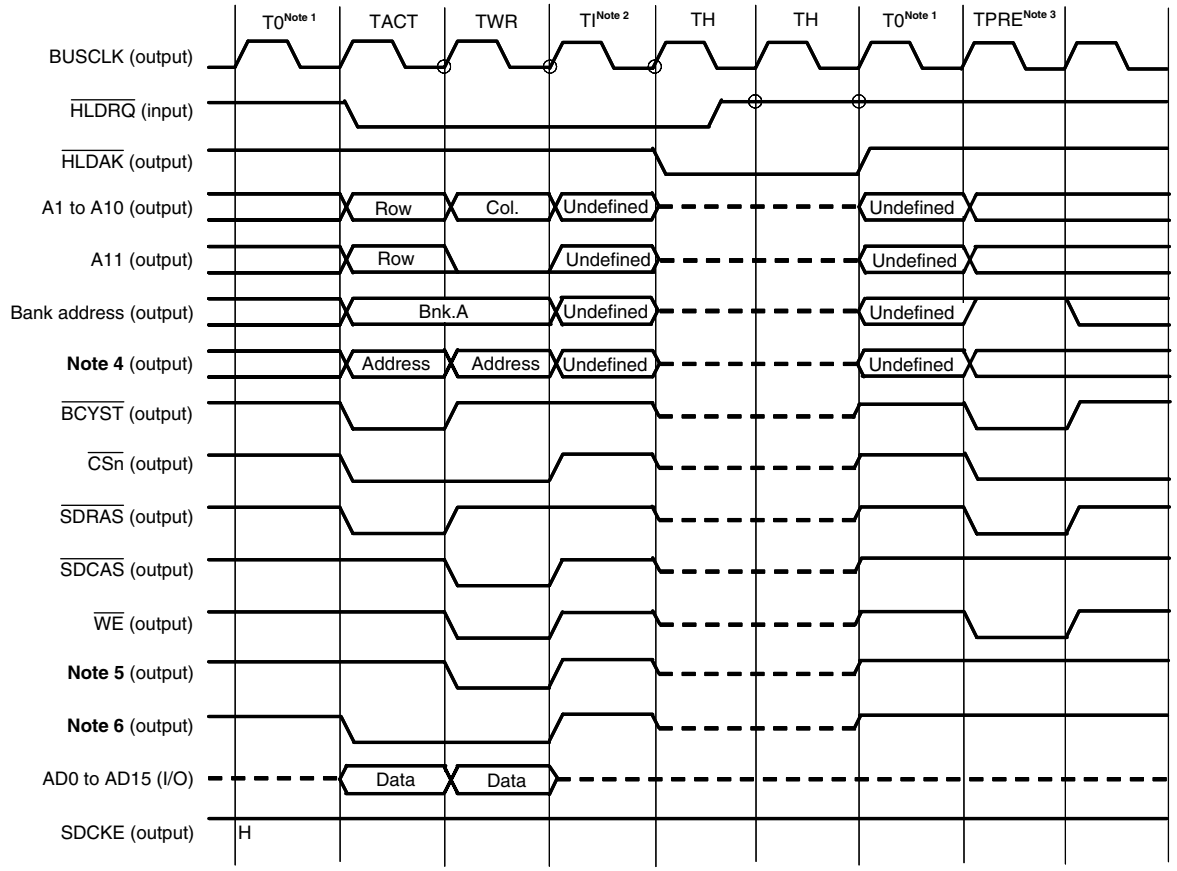
Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

3. n = 1, 3, 4, or 6
m = 0 or 1
xx = U or L

4. Bnk.: Bank address
Col.: Column address
Row: Row address

(3) SDRAM (During write, BMC register = 01H)



Notes 1. State (T0) inserted between bys cycles

2. Idle state (TI) inserted according to the BCC register setting

3. Always executes the all bank precharge command.

4. Addresses other than bank address, A11, and A1 to A10

5. When $\overline{\text{xWR}}$ output mode/xDQM output mode is set (PFCCT.PFCCTm bit = 0)

6. When $\overline{\text{xBE}}$ output mode/xDQM output mode is set (PFCCT.PFCCTm bit = 1)

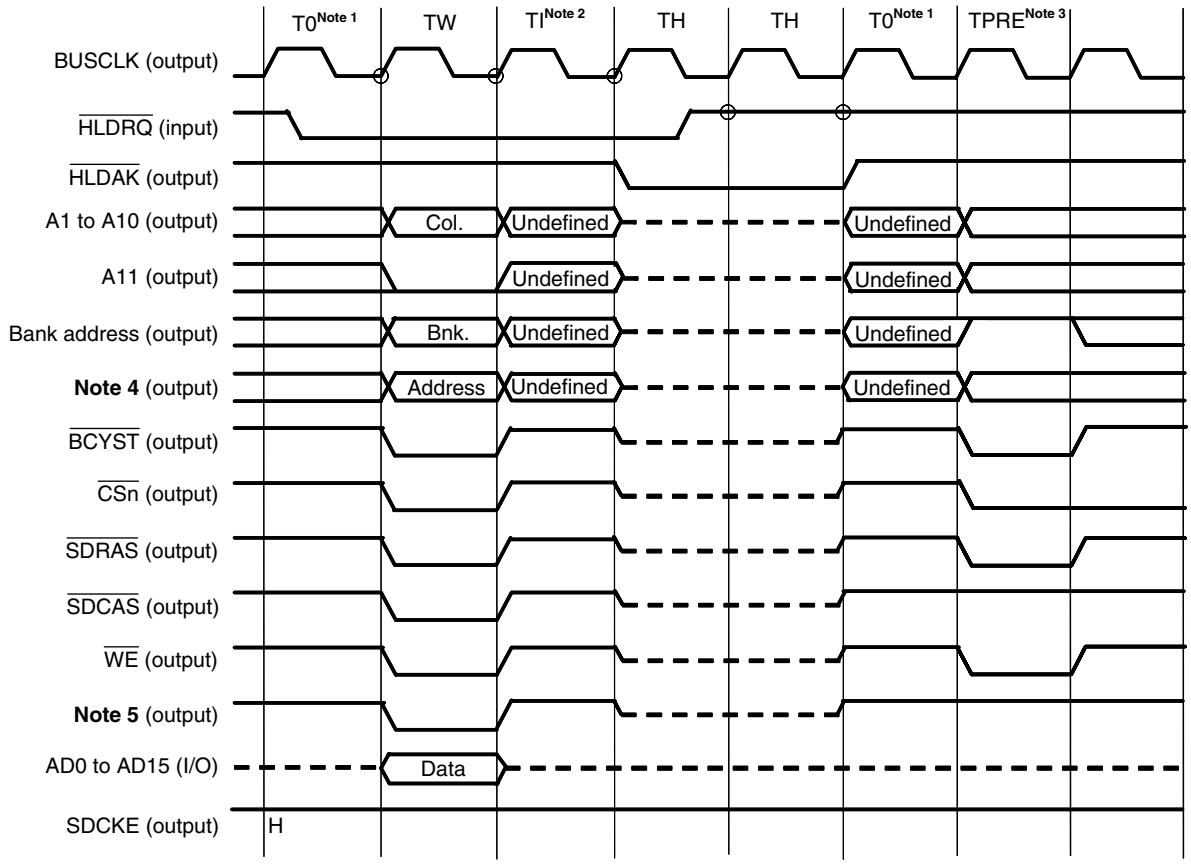
Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

3. n = 1, 3, 4, or 6
m = 0 or 1
xx = U or L

4. Bnk.: Bank address
Col.: Column address
Row: Row address

(4) SDRAM (During write, BMC register = 01H, when bus hold request is acknowledged during on-page access)



Notes 1. State (T0) inserted between bys cycles

2. Idle state (TI) independent of the BCC register setting

3. Always executes the all bank precharge command.

4. Addresses other than bank address, A11, and A1 to A10

5. UDQM, LDQM

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

3. n = 1, 3, 4, or 6

4. Bnk.: Bank address
Col.: Column address

5.11 Bus Priority

Bus hold, instruction fetch, operand data accesses, DMA cycles, and refresh cycles are executed in the external bus cycle.

Bus hold has the highest priority, followed by refresh cycle, DMA cycle, operand data access, and instruction fetch.

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

Table 5-3. Bus Priority

Priority	External Bus Cycle	Bus Master
High ↕ Low	Bus hold	External device
	Refresh cycle	SDRAM controller
	DMA cycle	DMA controller
	Operand data access	CPU
	Instruction fetch	CPU

5.12 Boundary Operation Conditions

5.12.1 Program space

Do not branch to the on-chip peripheral I/O area. If branch is executed, undefined data is fetched, and data is not fetched from the external memory.

5.12.2 Data space

The V850E/MA3 has an address misalign function.

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

(1) Halfword-length data access

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

(2) Word-length data access

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

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

CHAPTER 6 MEMORY ACCESS CONTROL FUNCTION

6.1 SRAM, External ROM, External I/O Interface

6.1.1 Features

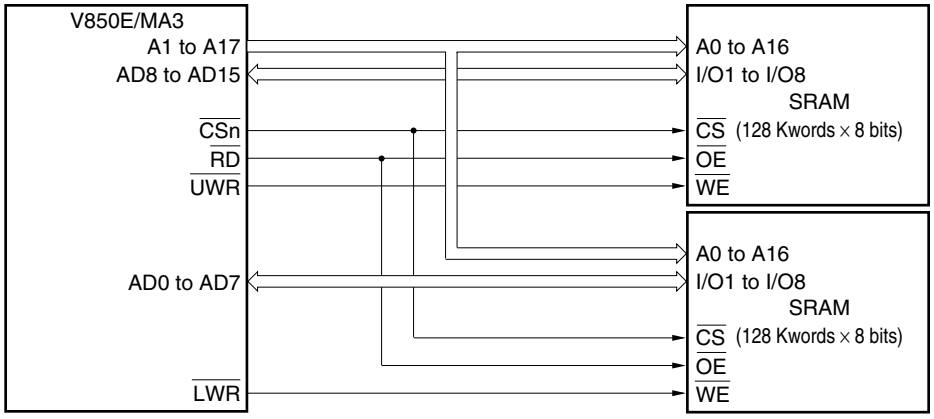
- SRAM is accessed in a minimum of 2 states.
- Up to 7 states of programmable data waits can be inserted by setting the DWC0 and DWC1 registers (DMA flyby transfer: FWC register).
- Data wait can be controlled via $\overline{\text{WAIT}}$ pin input.
- Up to 3 idle states can be inserted after a read/write cycle by setting the BCC register (DMA flyby transfer: FIC register).
- Up to 3 address setup wait states can be inserted by setting the ASC register.
- DMA flyby transfer can be activated (SRAM → external I/O, external I/O → SRAM)
- Supports separate bus mode/multiplexed bus mode.

6.1.2 SRAM connection

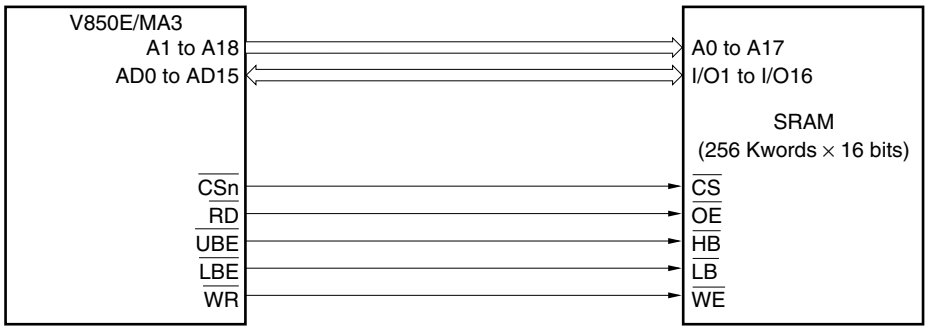
Examples of connection to SRAM are shown below.

Figure 6-1. Examples of Connection to SRAM (in Separate Bus Mode)

(a) When data bus width is 8 bits



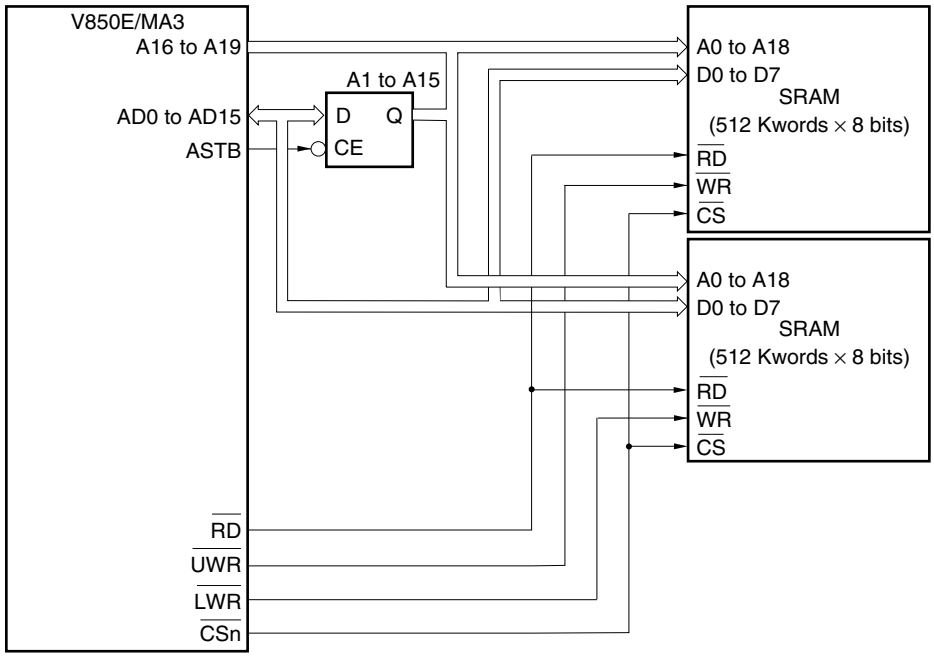
(b) When data bus width is 16 bits



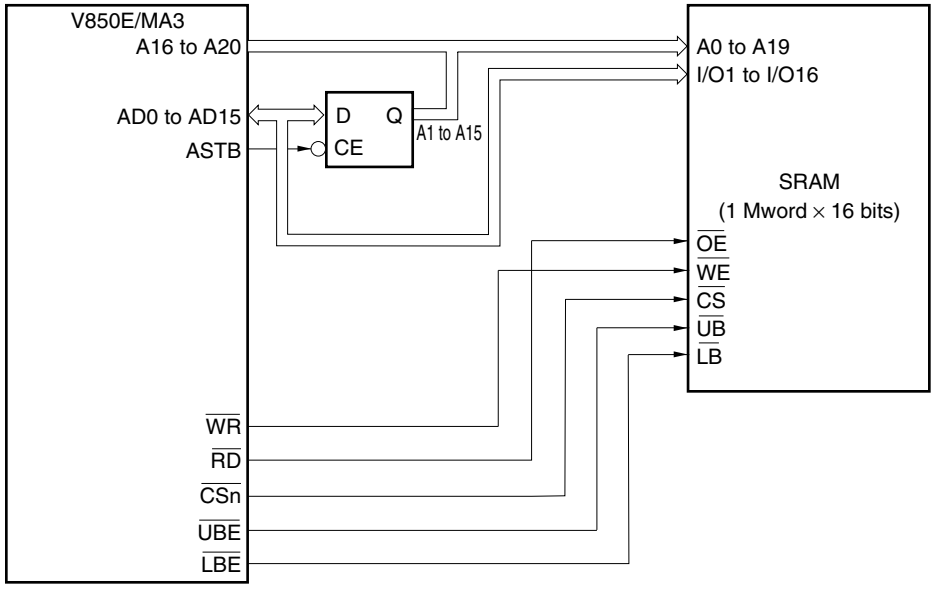
Remark n = 0 to 7

Figure 6-2. Examples of Connection to SRAM (in Multiplexed Bus Mode)

(a) When data bus width is 8 bits



(b) When data bus width is 16 bits



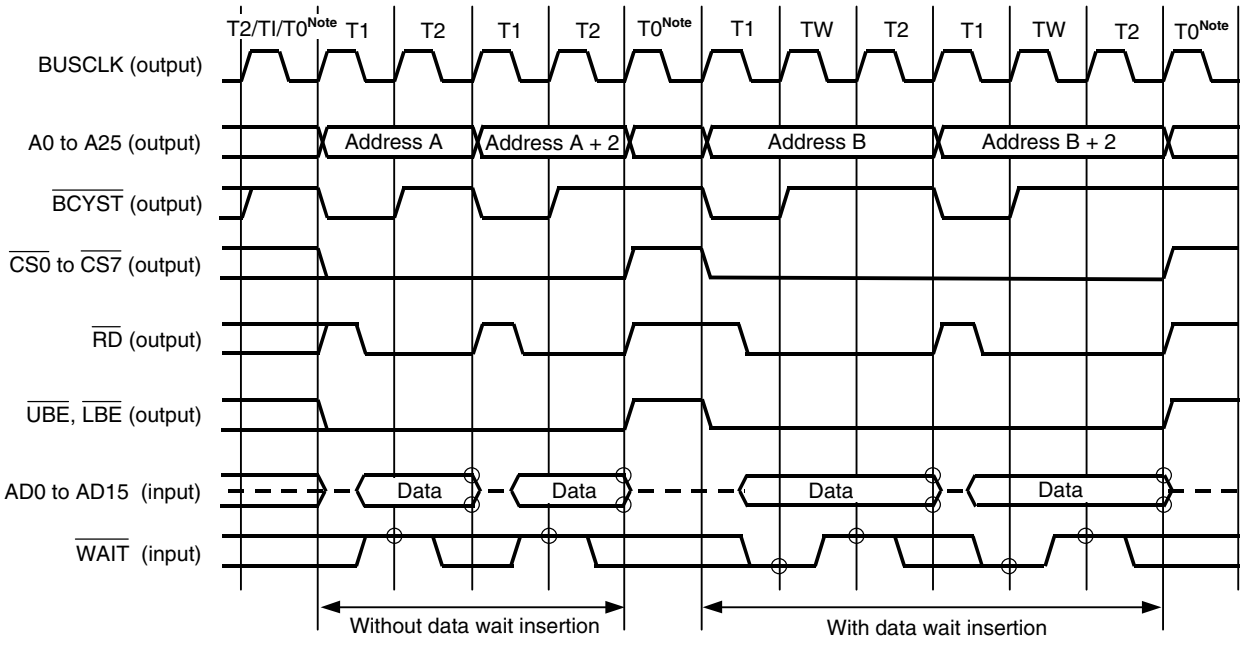
Remark n = 0 to 7

6.1.3 SRAM, external ROM, external I/O access

(1) Timing in separate bus mode

Figure 6-3. SRAM, External ROM, External I/O Access Timing (in Separate Bus Mode) (1/12)

(a) Read (successive 32-bit access)



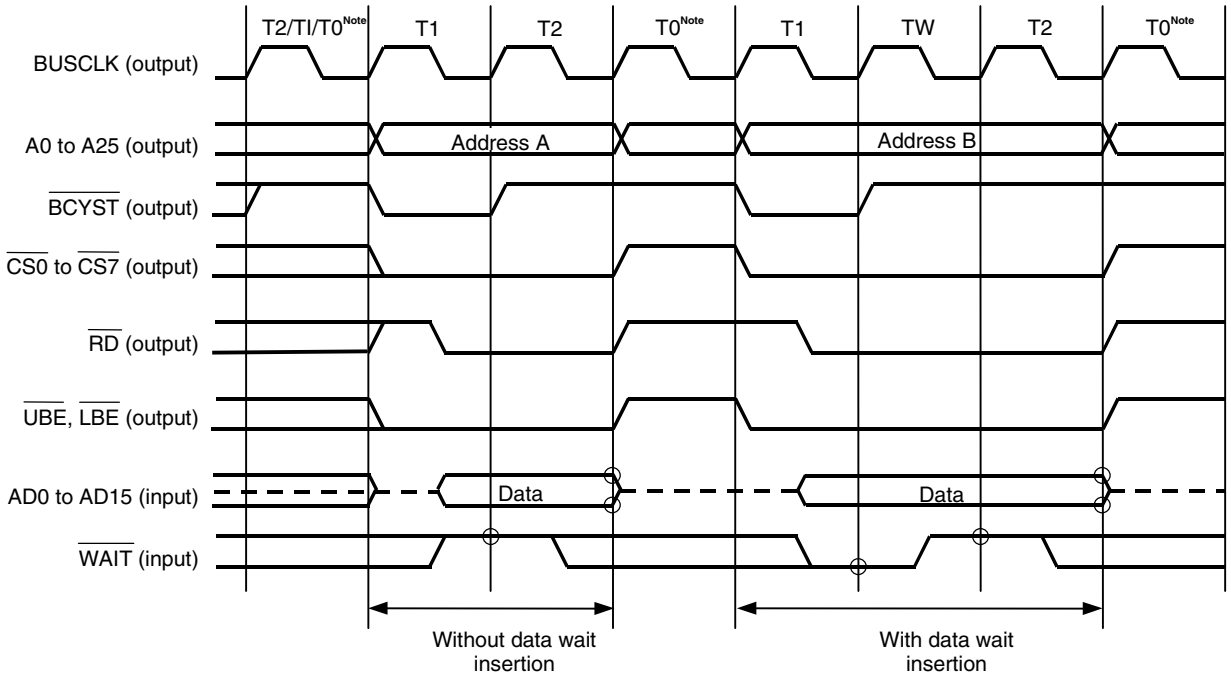
Note State (T0) inserted between bus cycles

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-3. SRAM, External ROM, External I/O Access Timing (in Separate Bus Mode) (2/12)

(b) Read (successive 16-bit access)



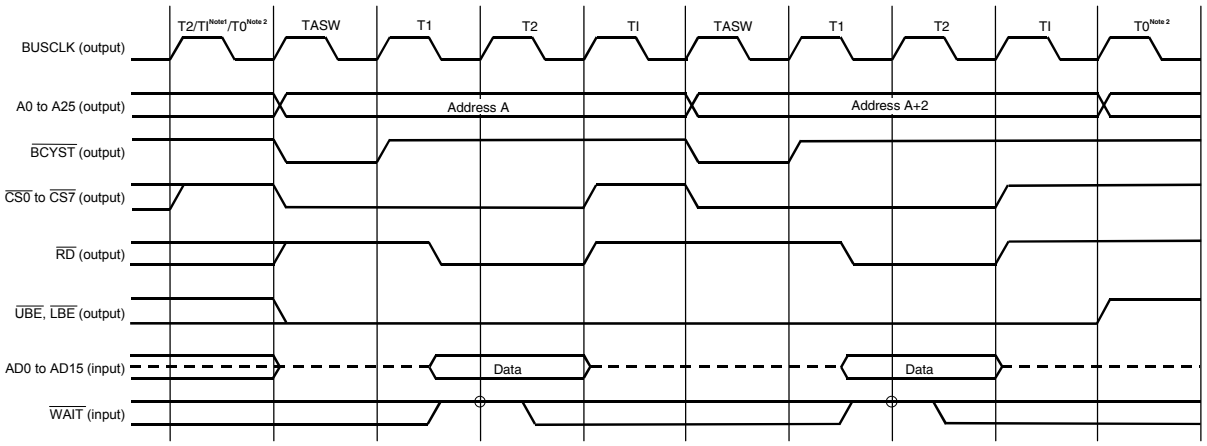
Note State (T0) inserted between bus cycles

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-3. SRAM, External ROM, External I/O Access Timing (in Separate Bus Mode) (3/12)

(c) Read (32-bit access, address setup wait, idle state inserted)



Notes 1. If the chip select signal is not the same, the idle state may not be inserted, depending on the setting.

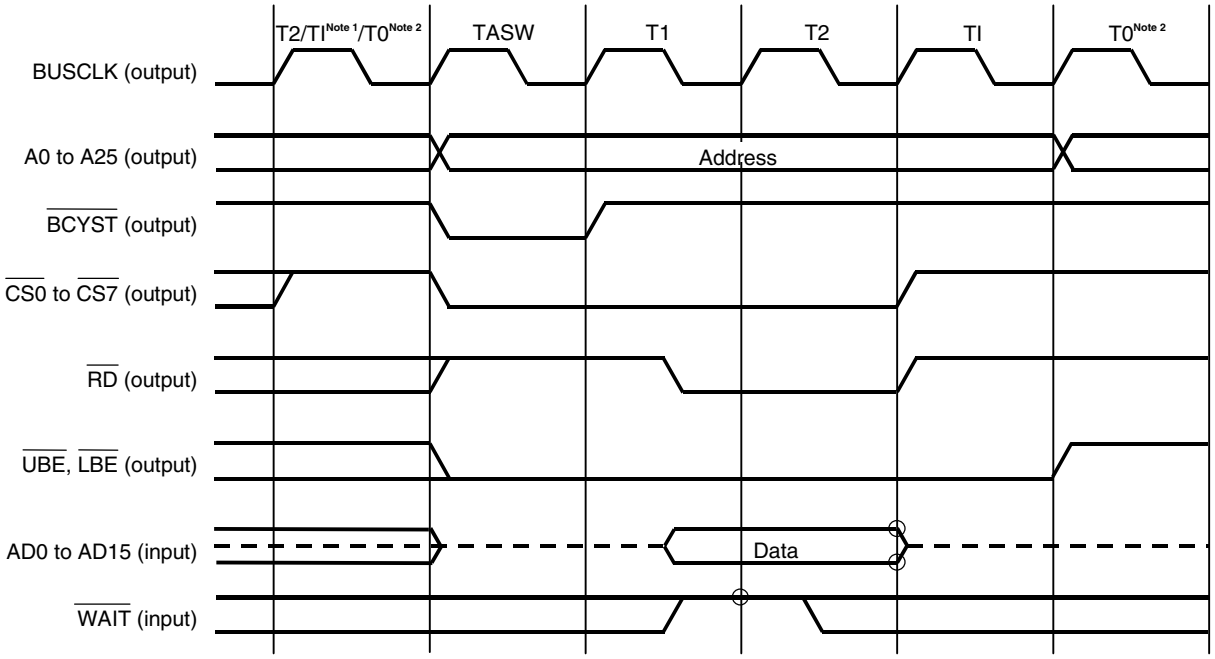
2. State (T0) inserted between bus cycles

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-3. SRAM, External ROM, External I/O Access Timing (in Separate Bus Mode) (4/12)

(d) Read (16-bit access, address setup wait, idle state inserted)



Notes 1. If the chip select signal is not the same, the idle state may not be inserted, depending on the setting.

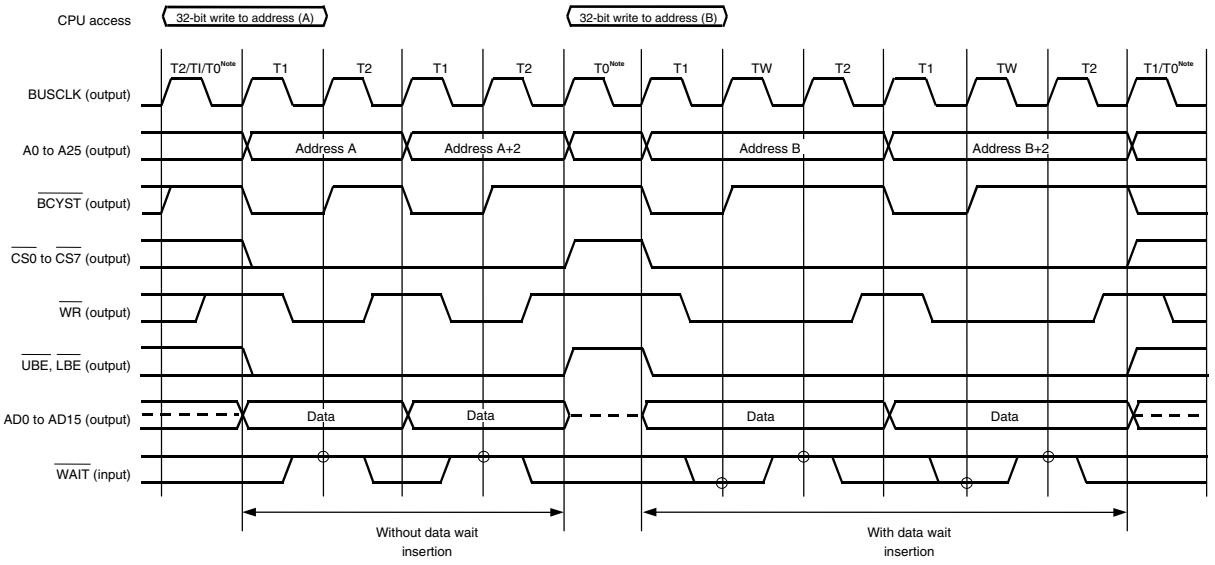
2. State (T0) inserted between bus cycles

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-3. SRAM, External ROM, External I/O Access Timing (in Separate Bus Mode) (5/12)

(e) Write (32-bit access (1/2))



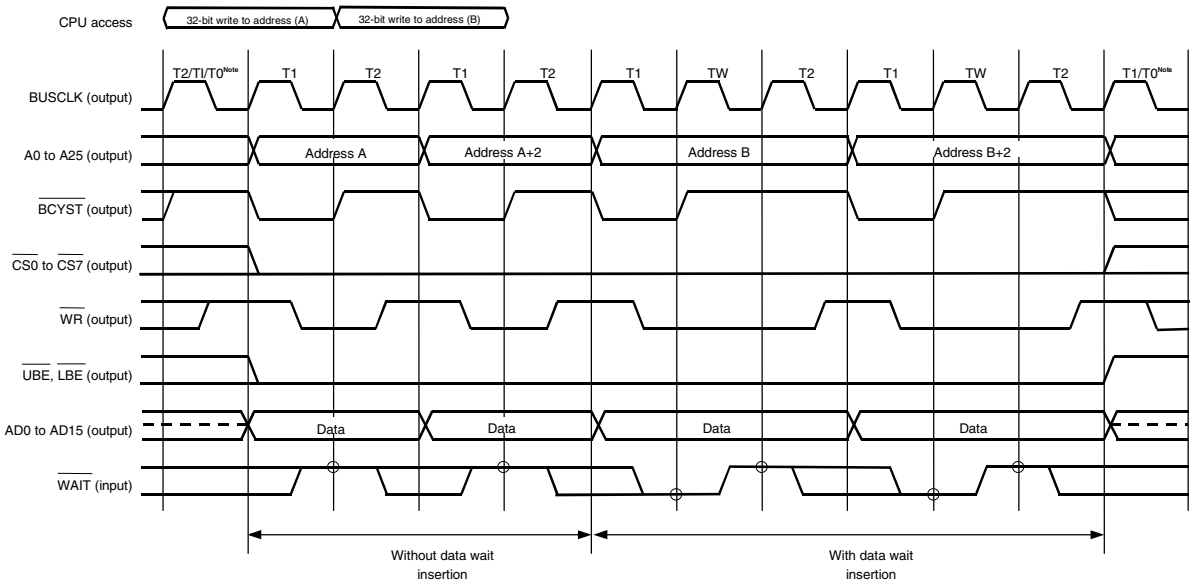
Note State (T0) inserted between bus cycles

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-3. SRAM, External ROM, External I/O Access Timing (in Separate Bus Mode) (6/12)

(e) Write (32-bit access (2/2))



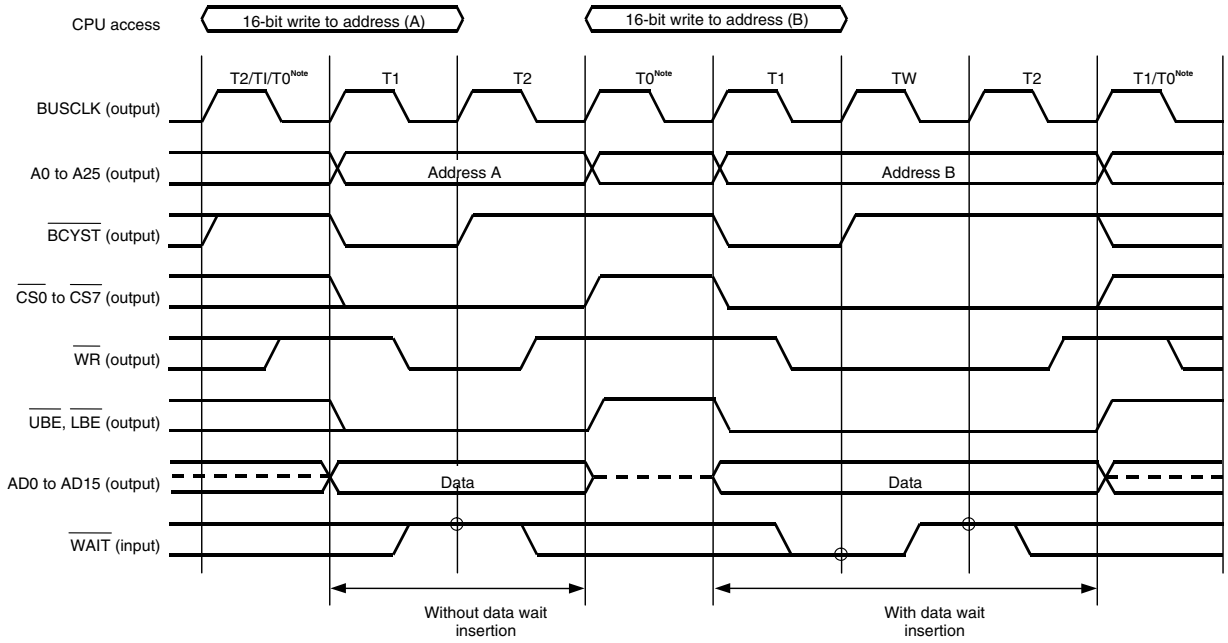
Note State (T0) inserted between bus cycles

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-3. SRAM, External ROM, External I/O Access Timing (in Separate Bus Mode) (7/12)

(f) Write (16-bit access (1/2))



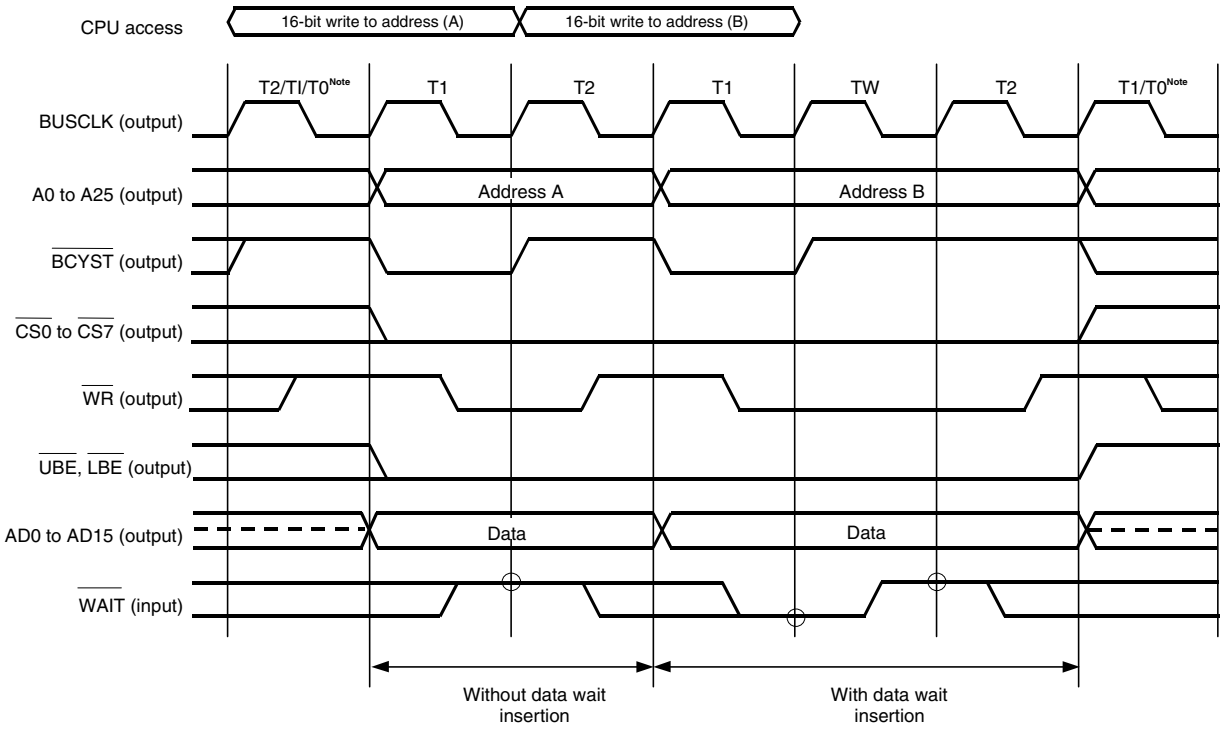
Note State (T0) inserted between bus cycles

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-3. SRAM, External ROM, External I/O Access Timing (in Separate Bus Mode) (8/12)

(f) Write (16-bit access (2/2))



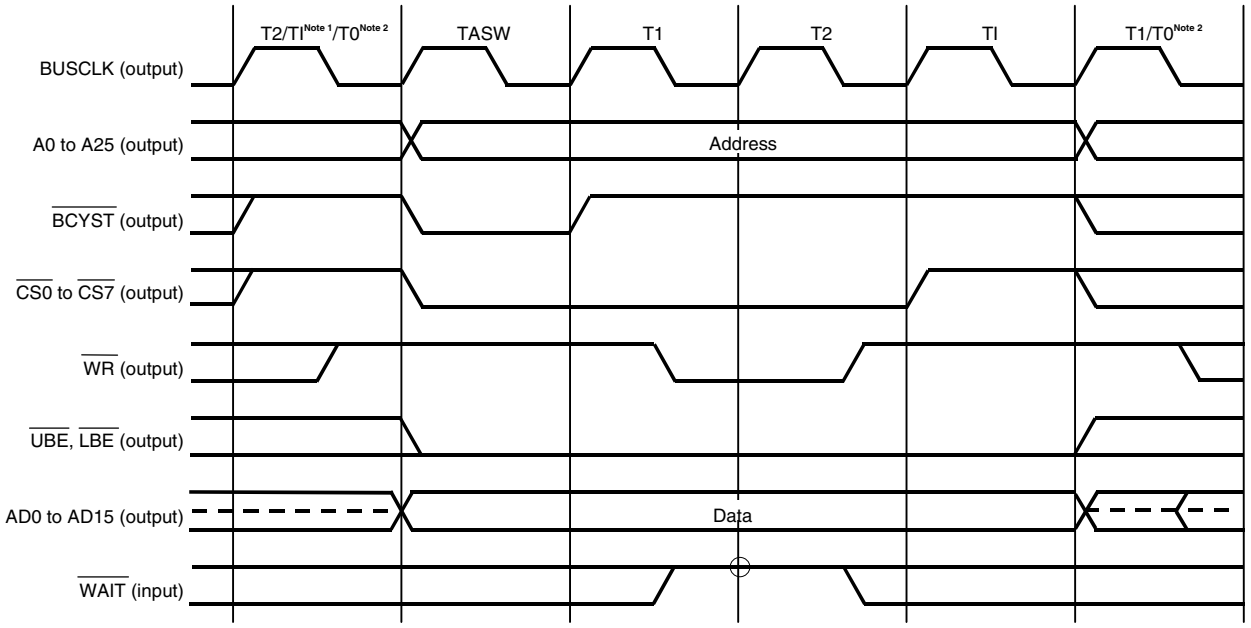
Note State (T0) inserted between bus cycles

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-3. SRAM, External ROM, External I/O Access Timing (in Separate Bus Mode) (9/12)

(g) Write (16-bit access, address setup wait, idle state inserted)



Notes 1. If the chip select signal is not the same, the idle state may not be inserted, depending on the setting.

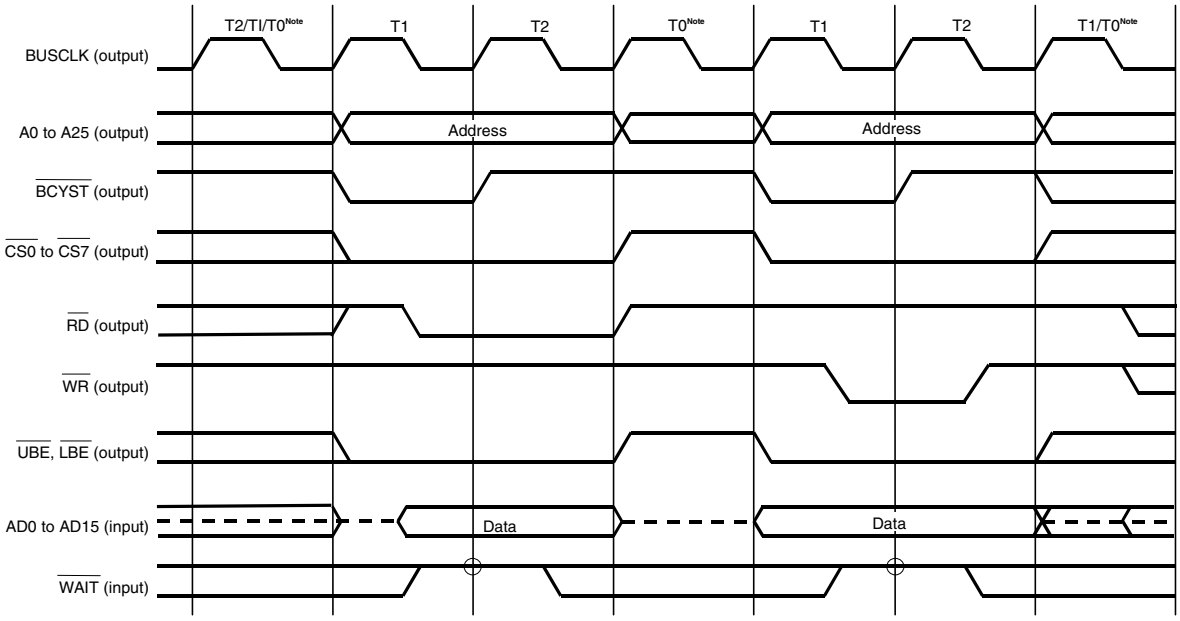
2. State (T0) inserted between bus cycles

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-3. SRAM, External ROM, External I/O Access Timing (in Separate Bus Mode) (10/12)

(h) Read → write (16-bit read access → 16-bit write access)



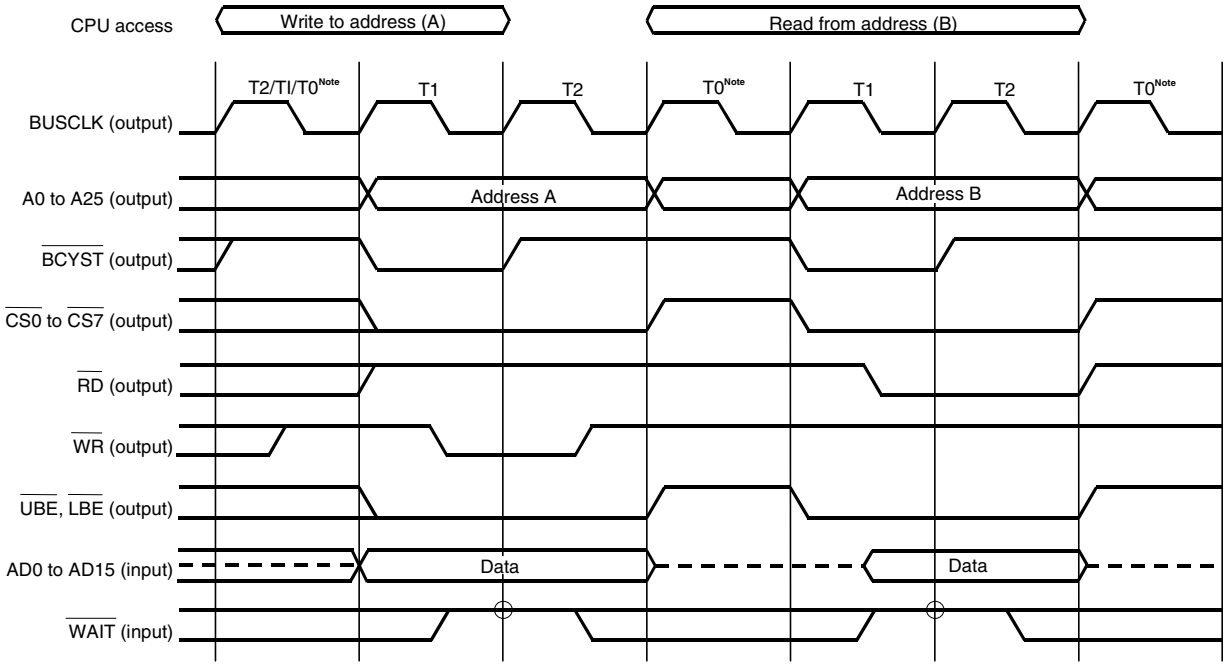
Note State (T0) inserted between bus cycles

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-3. SRAM, External ROM, External I/O Access Timing (in Separate Bus Mode) (11/12)

(i) Write → read (16-bit write access → 16-bit read access (1/2))



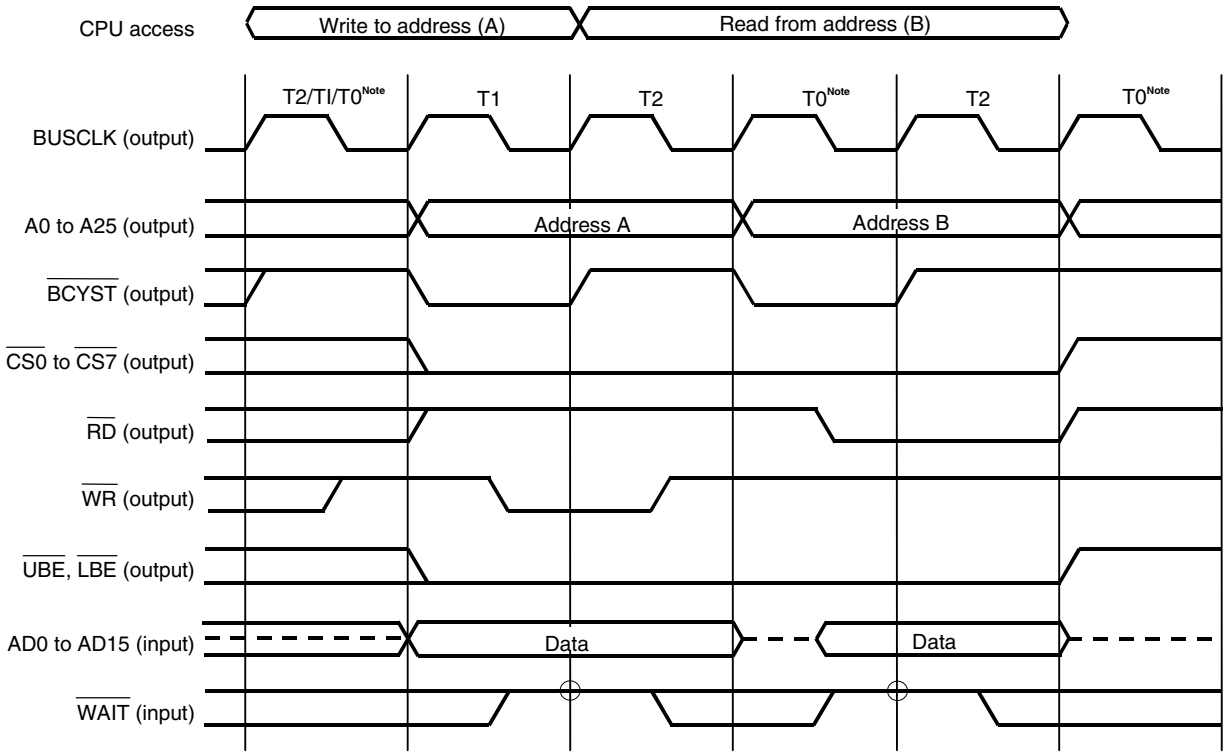
Note State (T0) inserted between bus cycles

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-3. SRAM, External ROM, External I/O Access Timing (in Separate Bus Mode) (12/12)

(i) Write → read (16-bit write access → 16-bit read access (2/2))



Note State (T0) inserted between bus cycles

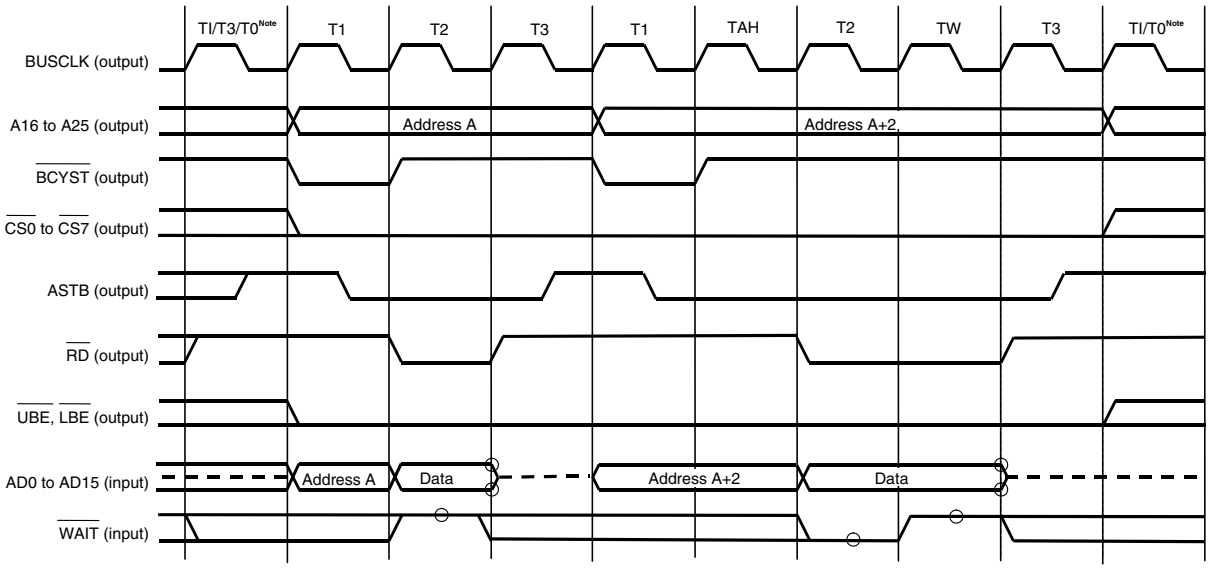
Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

(2) Timing in multiplexed bus mode

Figure 6-4. SRAM, External ROM Access Timing (in Multiplexed Bus Mode) (1/5)

(a) Read (32-bit access)



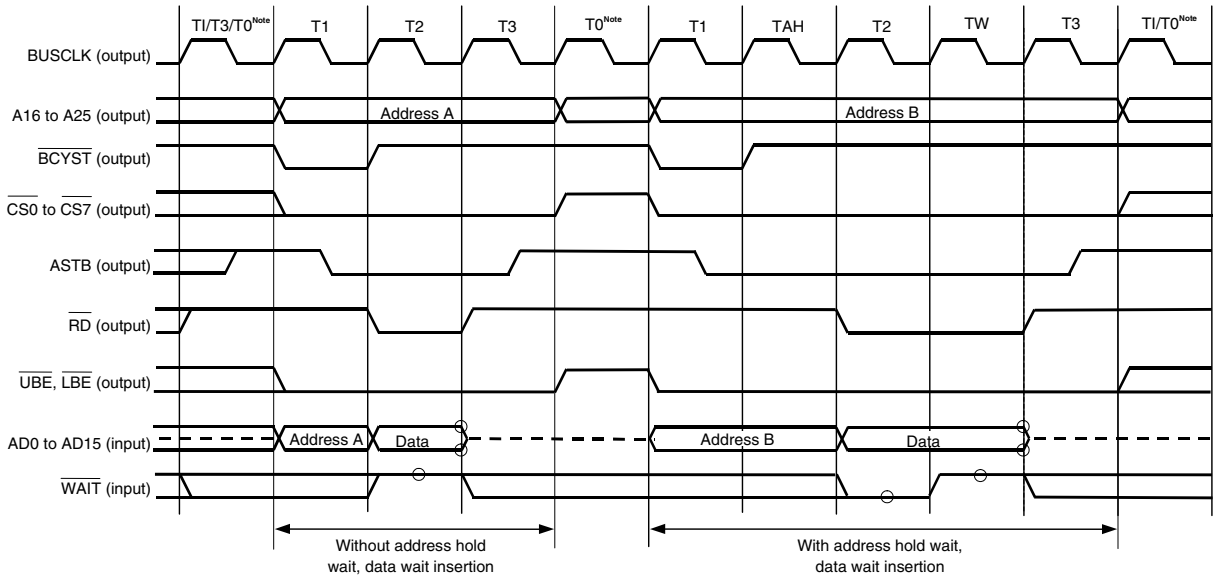
Note State (T0) inserted between bus cycles

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-4. SRAM, External ROM Access Timing (in Multiplexed Bus Mode) (2/5)

(b) Read (successive 16-bit access)



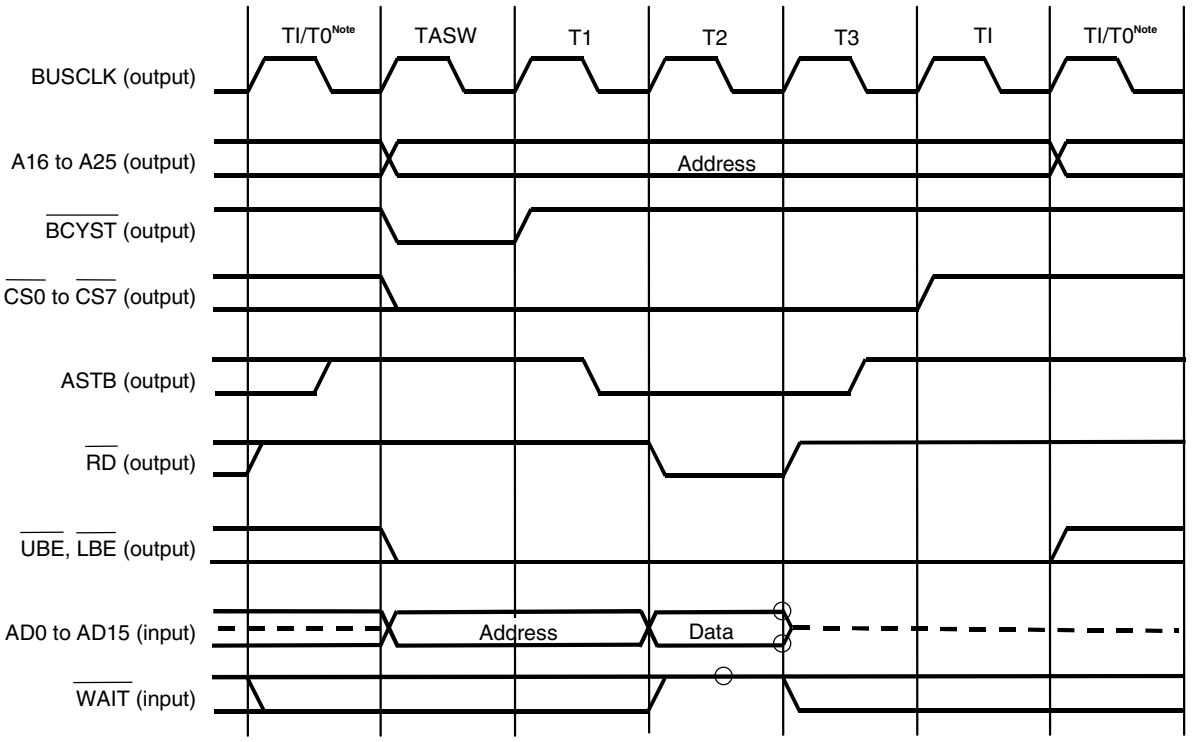
Note State (T0) inserted between bus cycles

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-4. SRAM, External ROM Access Timing (in Multiplexed Bus Mode) (3/5)

(c) Read (address setup wait, idle state inserted)



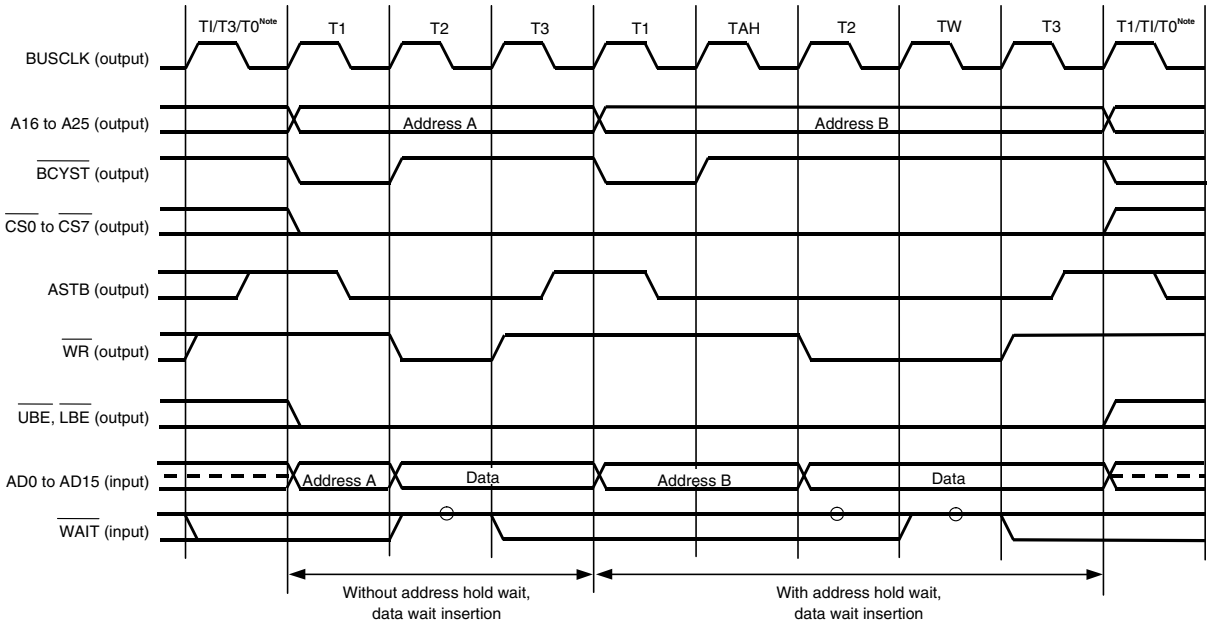
Note State (T0) inserted between bus cycles

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-4. SRAM, External ROM Access Timing (in Multiplexed Bus Mode) (4/5)

(d) Write (successive 16-bit access)



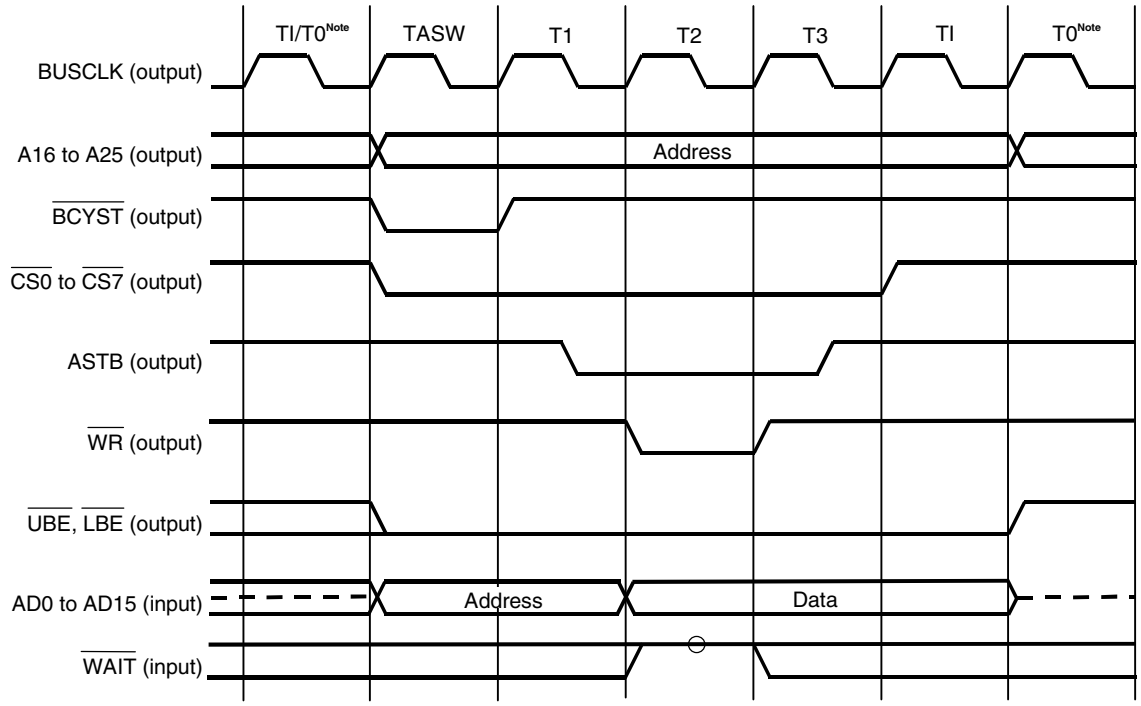
Note State (T0) inserted between bus cycles

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-4. SRAM, External ROM Access Timing (in Multiplexed Bus Mode) (5/5)

(e) Write (16-bit access, address setup wait, idle state inserted)



Note State (T0) inserted between bus cycles

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

6.2 Page ROM Controller (ROMC)

The page ROM controller (ROMC) is provided for accessing ROM (page ROM) with a page access function.

6.2.1 Features

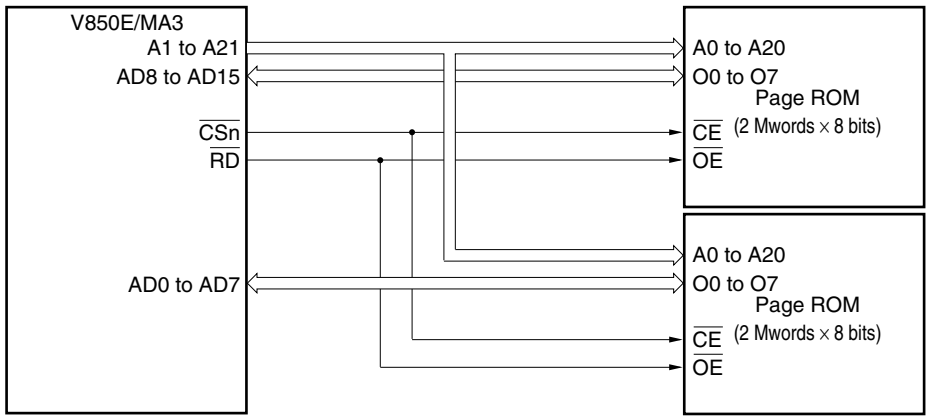
- Direct connection to 8-bit/16-bit page ROM supported
- Page ROM is accessed in a minimum of 2 states.
- On-page judgment function
- Up to 7 states of programmable data waits can be inserted by setting the following registers.
  - During on-page cycle: PRC register
  - During off-page cycle: DWC0 and DWC1 registers
  - During DMA flyby cycle: FWC register
- Waits can be controlled via $\overline{\text{WAIT}}$ pin input.
- DMA flyby cycle can be activated (page ROM → external I/O)
- SRAM write cycle is started when a write cycle request is issued to a CSn space (n = 0 to 7) where page ROM is located

6.2.2 Page ROM connection

Examples of connection to page ROM are shown below.

Figure 6-5. Examples of Connection to Page ROM

(a) When data bus width is 8 bits



(b) When data bus width is 16 bits

Remark n = 0 to 7

6.2.3 On-page

On-page access in the page ROM cycle is generated in the following conditions.

- 8-bit bus width halfword access
  16-bit access from even address boundary [off-page → on-page]
- 8-bit bus width word access
  32-bit access from word boundary [off-page → on-page → on-page → on-page]
  32-bit access from halfword boundary [off-page → on-page → off-page → on-page]
  32-bit access from odd address [off-page → off-page → on-page → off-page]
- 16-bit bus width word access
  32-bit access from word boundary [off-page → on-page]

6.2.4 Page ROM configuration register (PRC)

This register sets the wait state in the on-page cycle and sets the number of waits corresponding to the connected page ROM on-page access time, as well as to the system clock.

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

Reset input sets this register to 7000H.

Caution Write to the PRC register after reset, and then do not change the set values. Also, do not access an external memory area until the initial setting of the PRC register is complete. However, it is possible to access external memory areas whose initialization settings are complete.

After reset: 7000H R/W Address: FFFFF49AH

	15	14	13	12	11	10	9	8
PRC	0	PRW2	PRW1	PRW0	0	0	0	0
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0

PRW2	PRW1	PRW0	Specification of number of waits corresponding to system clock during on-page access
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

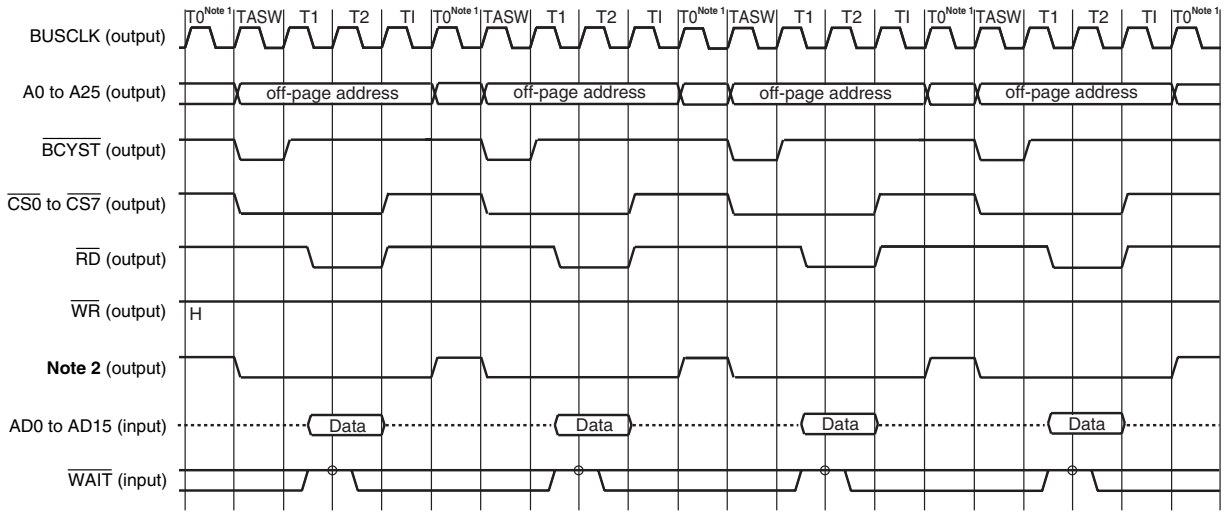
Cautions 1. For off-page access, the waits set by DWC0 and DWC1 registers are inserted.

2. Be sure to clear bits 15 and 11 to 0 to "0". The operation cannot be guaranteed if these bits are set to 1.

6.2.5 Page ROM access

Figure 6-6. Page ROM Access Timing (1/3)

(a) Read (address setup wait, idle state inserted, 16-bit bus width halfword access)



Notes 1. State (T0) inserted between bus cycles.

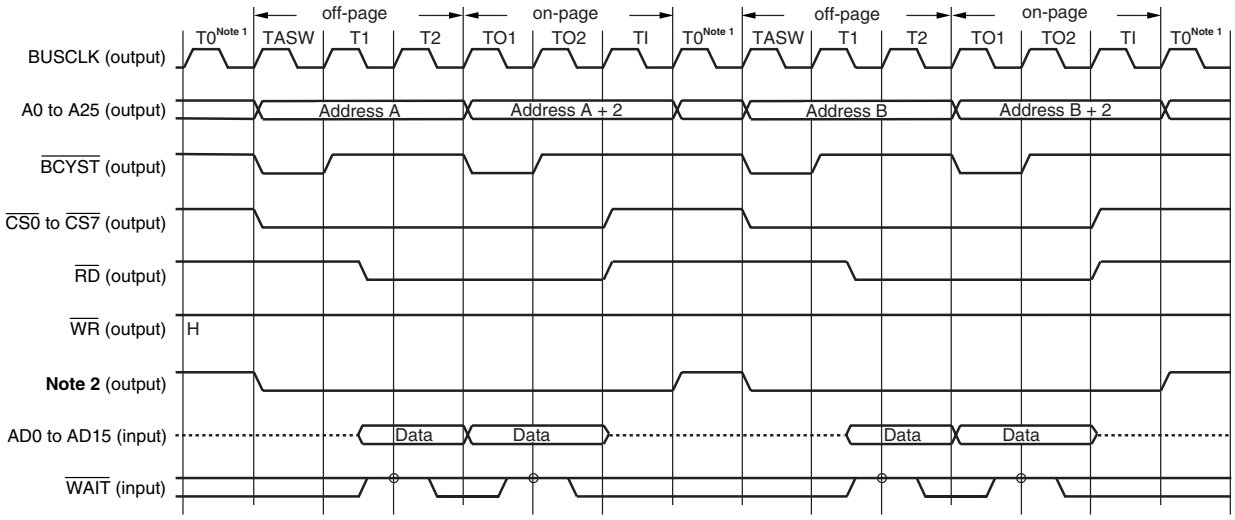
2. $\overline{UBE}$, $\overline{LBE}$

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-6. Page ROM Access Timing (2/3)

(b) Read (address setup wait, idle state inserted, 16-bit bus width word access)



Notes 1. State (T0) inserted between bus cycles.

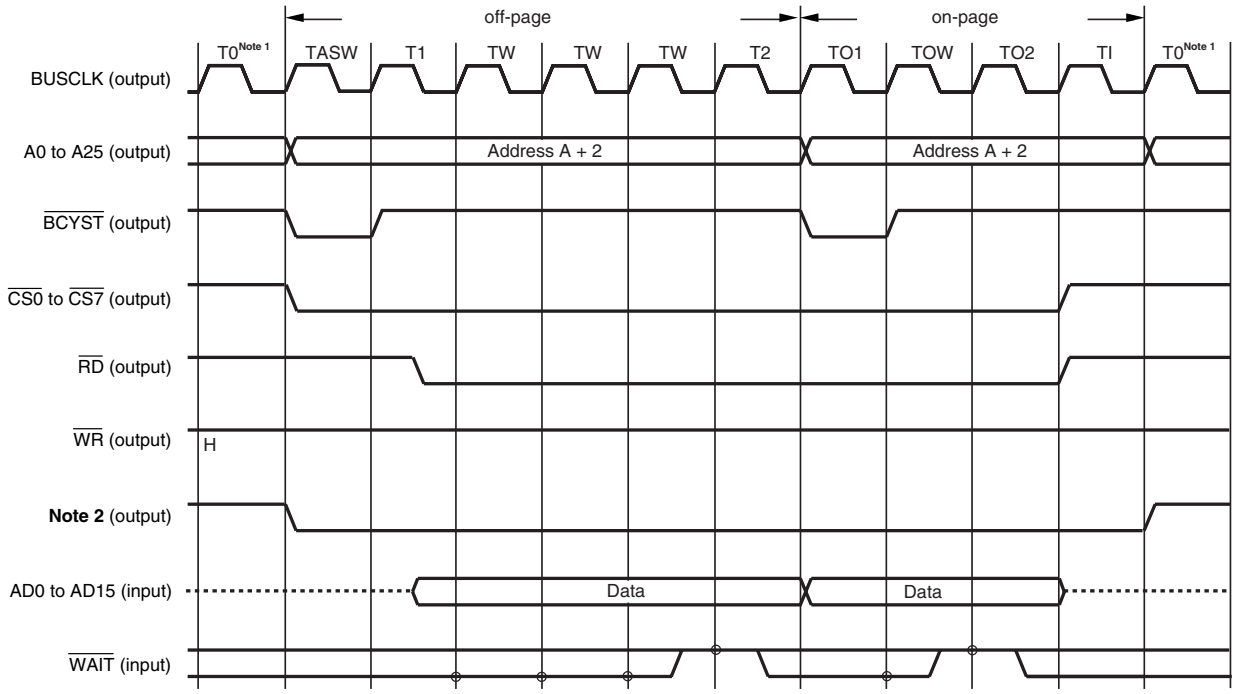
2. $\overline{UBE}$, $\overline{LBE}$

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

Figure 6-6. Page ROM Access Timing (3/3)

(c) Read (address setup wait, idle state inserted, data wait inserted, on-page data wait inserted, 16-bit bus width word access)



Notes 1. State (T0) inserted between bus cycles.

2. $\overline{UBE}$, $\overline{LBE}$

Remarks 1. The circles indicate the sampling timing.

2. The broken lines indicate the high-impedance state.

6.3 DRAM Controller (SDRAM)

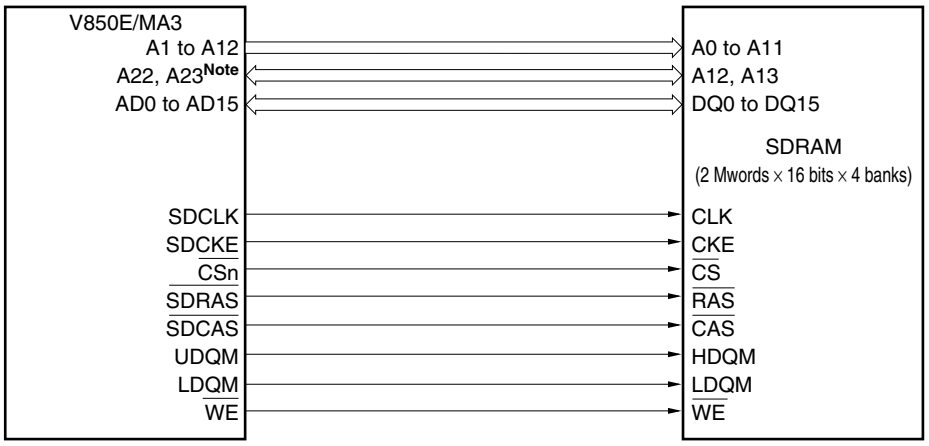
6.3.1 Features

- Burst length: 1
- Wrap type: Sequential
- CAS latency: 1, 2, and 3 supported
- 4 types of SDRAM can be assigned to 4 memory blocks.
- Row and column address multiplex widths can be changed.
- Waits (0 to 3 waits) can be inserted between the bank active command and the read/write command.
- Supports CBR (automatic) refresh and self-refresh.

6.3.2 SDRAM connection

An example of connection to SDRAM is shown below.

Figure 6-7. Example of Connection to SDRAM



Note The address signals to be used differ depending on the SDRAM product.

Remark n = 1, 3, 4, 6

6.3.3 Address multiplex function

The row address output in the SDRAM cycle is multiplexed as shown in Figure 6-8 (a) (n = 1, 3, 4, 6) according to the value of the SCRn.SAWn0 and SCRn.SAWn1 bits. The column address output in the SDRAM cycle is multiplexed as shown in Figure 6-8 (b) (n = 1, 3, 4, 6) according to the value of the SCRn.SSOn0 and SCRn.SSOn1 bits. In Figures 6-8 (a) and (b), a0 to a25 indicate the addresses output from the CPU, and A0 to A25 indicate the address pins of the V850E/MA3.

Figure 6-8. Row Address/Column Address Output (1/2)

(a) Row address output

Address pin	A25 to A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Row address (SAWn1, SAWn0 bits = 11)	a25 to a18	a17	a16	a15	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11
Row address (SAWn1, SAWn0 bits = 10)	a25 to a18	a17	a16	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10
Row address (SAWn1, SAWn0 bits = 01)	a25 to a18	a17	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10	a9
Row address (SAWn1, SAWn0 bits = 00)	a25 to a18	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10	a9	a8

Remark n = 1, 3, 4, 6

(b) Column address output (using all-bank precharge command)

Address pin	A25 to A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Column address (SSOn1, SSOn0 bits = 00)	a25 to a18	a17	a16	a15	a14	a13	a12	a11	1	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
Column address (SSOn1, SSOn0 bits = 01)	a25 to a18	a17	a16	a15	a14	a13	a12	1	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0

Remark n = 1, 3, 4, 6

(c) Column address output (using register write command)

Address pin	A25 to A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Column address (SSOn1, SSOn0 bits = 00)	0	0	0	0	0	0	0	0	0	0	0	0	LTM2	LTM1	LTM0	0	0	0	0
Column address (SSOn1, SSOn0 bits = 01)	0	0	0	0	0	0	0	0	0	0	0	LTM2	LTM1	LTM0	0	0	0	0	0

Remark n = 1, 3, 4, 6

Figure 6-8. Row Address/Column Address Output (2/2)

(d) Column address output (using read/write command)

Address pin	A25 to A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Column address (SSOn1, SSOn0 bits = 00)	a25 to a18	a17	a16	a15	a14	a12	a11	a10	0	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
Column address (SSOn1, SSOn0 bits = 01)	a25 to a18	a17	a16	a15	a14	a12	a11	0	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0

Remark n = 1, 3, 4, 6

(1) Output of each address and connection of SDRAM

The set contents of the SCRn register, physical address, address output from the V850E/MA3, and connection between the V850E/MA3 and SDRAM at each data bus width (8 bits or 16 bits) are described below.

(a) At 8-bit data bus width

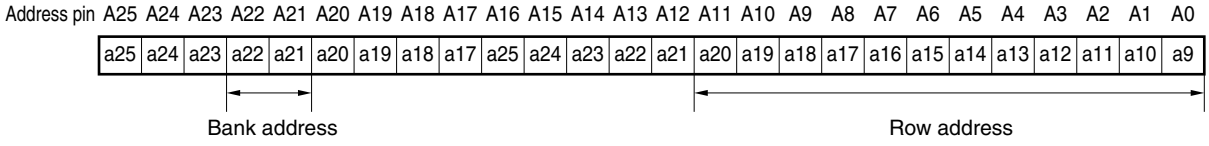
An example of connecting 64 Mb SDRAM (2 Mwords × 8 bits × 4 banks) at 8-bit data bus width is shown below.

- Set contents of SCRn register
  SSOn1, SSOn0 bits = 00: Data bus width = 8 bits
  RAWn1, RAWn0 bits = 01: Row address width = 12 bits
  SAWn1, SAWn0 bits = 01: Column address width = 9 bits

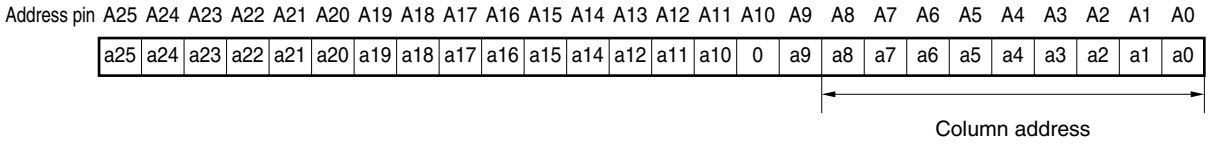
- Physical address
  A22, A21: Bank address
  A20 to A9: Row address
  A8 to A0: Column address

- Address output from V850E/MA3
  A22, A21: Bank address
  A11 to A0: Row address (12 bits), column address (9 bits)

(a) Row address, bank address output (with active command)



(b) Column address output (with read/write command)



- Connection of V850E/MA3 and SDRAM
  A22, A21 (V850E/MA3) → BA0 (A13), BA1 (A12) (SDRAM)
  A11 to A0 (V850E/MA3) → A11 to A0 (SDRAM)

(b) At 16-bit data bus width

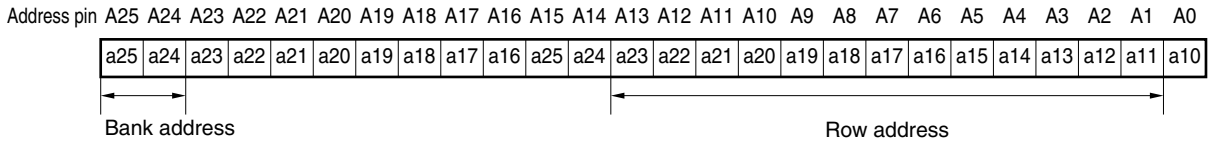
An example of connecting 512 Mb SDRAM (8 Mwords × 16 bits × 4 banks) at 16-bit data bus width is shown below.

- Set contents of SCRn register
  SSOn1, SSOn0 bits = 01: Data bus width = 16 bits
  RAWn1, RAWn0 bits = 10: Row address width = 13 bits
  SAWn1, SAWn0 bits = 10: Column address width = 10 bits

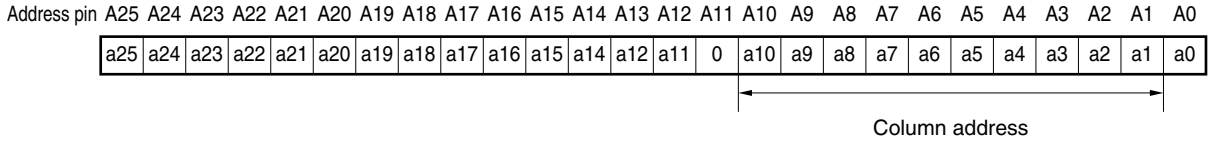
- Physical address
  A25, A24: Bank address
  A23 to A11: Row address
  A10 to A1: Column address

- Address output from V850E/MA3
  A25, A24: Bank address
  A13 to A1: Row address (13 bits), column address (10 bits)

(a) Row address, bank address output (with active command)



(b) Column address output (with read/write command)



- Connection of V850E/MA3 and SDRAM
  A25, A24 (V850E/MA3) → BA0 (A14), BA1 (A13) (SDRAM)
  A13 to A1 (V850E/MA3) → A12 to A0 (SDRAM)

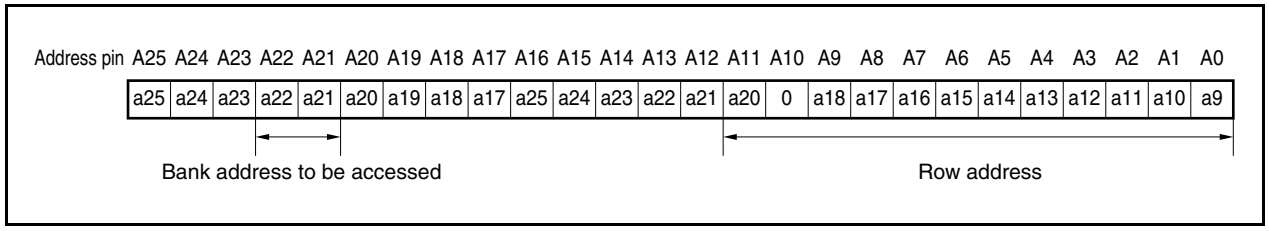
(2) Bank address output

The V850E/MA3 precharges a bank to access by using a bank precharge command when a row address is output immediately after page change. After bank change, it precharges the bank previously accessed when a column address is output. Therefore, a bank is precharged both when a row address is output and when a column address is output. To connect SDRAM with the contents described in **6.3.3 (1) (a) At 8-bit data bus width**, always connect the pins (A22 and A21) of the V850E/MA3 that outputs bank addresses to the bank address pins (A13 and A12) of SDRAM.

An example of address output by the bank precharge command when the page or bank is changed if SDRAM is connected with the contents described in **6.3.3 (1) (a) At 8-bit data bus width** is shown below.

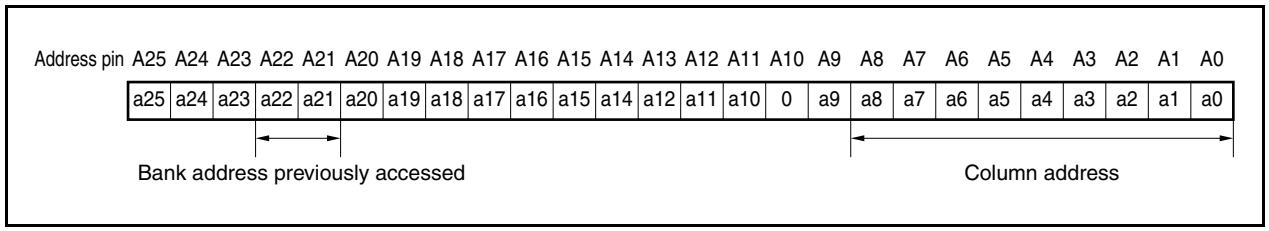
(a) When page is changed (at 8-bit data bus width)

When the page is changed, the precharge command outputs the physical addresses to be accessed (a20 and a18 to a9) to the A11 and A9 to A0 pins, and the bank addresses to be accessed (a22 and a21) to the A22 and A21 pins.



(b) When bank is changed (at 8-bit data bus width)

When the bank is changed, the bank precharge command outputs the physical addresses to be accessed (a8 to a0) to the A8 to A0 pins, and the bank addresses previously accessed (a22 and a21) to the A22 and A21 pins.



The bits that determine the precharge mode (A10: 8-bit data bus width, A11: 16-bit data bus width, A12: 32-bit data bus width) output a high level when the all bank precharge command is executed, and a low level when any other command is executed for precharging.

6.3.4 SDRAM configuration registers 1, 3, 4, 6 (SCR1, SCR3, SCR4, SCR6)

These registers specify the number of waits and the address multiplex width. The SCRn register corresponds to $\overline{CSn}$ (n = 1, 3, 4, 6). For example, to connect SDRAM to $\overline{CS1}$, set the SCR1 register.

These registers can be read or written in 16-bit units. The WCFn bit is read-only.

Reset input sets these registers to 30C0H.

Cautions 1. An SDRAM read/write cycle is not generated prior to executing a register write operation. Access SDRAM after reading the value of the SCRn register and confirming that the WCFn bit is set to 1.

2. To write to the SCRn register again following access to SDRAM, clear the BCT0.MEn and BCT1.MEn bits to 0, and then set it to 1 again before performing access (n = 1, 3, 4, 6).

3. Do not execute instructions to write to the SCRn register continuously. Be sure to insert another instruction between commands to write to the SCRn register.

4. Start accessing SDRAM after all the SCRn registers have been set. Set the RFSn register before setting the SCRn register (n = 1, 3, 4, 6).

(1/2)

After reset: 30C0H R/W Address: SCR1 FFFFF4A4H, SCR3 FFFFF4ACH, SCR4 FFFFF4B0H, SCR6 FFFFF4B8H

SCRn (n = 1, 3, 4, 6)

15	14	13	12	11	10	9	8
0	LTMn2	LTMn1	LTMn0	0	0	0	WCFn
7	6	5	4	3	2	1	0
BCWn1	BCWn0	SSOn1	SSOn0	RAWn1	RAWn0	SAWn1	SAWn0

LTMn2	LTMn1	LTMn0	Specification of CAS latency value when reading
0	0	1	1
0	1	0	2
0	1	1	3
Other than above			Setting prohibited

WCFn	Register write command completion flag to SDRAM after SCRn register setting
0	Setting not complete
1	Setting complete
Set (1) when a register write command occurs.	

BCWn1	BCWn0	Specification of number of wait states to be inserted
0	0	Setting prohibited
0	1	1
1	0	2
1	1	3
Specify the number of wait states inserted from the bank active command to a read/write command, or from the precharge command to the bank active command.		

Caution Be sure to clear bits 15, 11, 10, and 9 to "0". The operation cannot be guaranteed if these bits are set to 1.

Remark n = 1, 3, 4, 6

(2/2)

SSOn1	SSOn0	Specification of address shift width during on-page judgment
0	0	0 bits (external data bus width: 8 bits)
0	1	1 bit (external data bus width: 16 bits)Note
1	0	Setting prohibited
1	1	Setting prohibited
If the external data bus width is set to 16 bits, the system does not use the lower address (A0). Set these bits in accordance with the contents of the LBS register corresponding to $\overline{CSn}$.		

RAWn1	RAWn0	Specification of row address width
0	0	11 bits
0	1	12 bits
1	0	13 bitsNote
1	1	Setting prohibited

SAWn1	SAWn0	Specification of address multiplex width (column address width) during SDRAM access
0	0	8 bits
0	1	9 bits
1	0	10 bits
1	1	11 bitsNote

Note The following setting is prohibited because the upper limit of the address is exceeded.

SSOn1	SSOn0	RAWn1	RAWn0	SAWn1	SAWn0	Setting
0	1	1	0	1	1	Data bus width: 16 bits Row address width: 13 bits Column address width: 11 bits

Remark n = 1, 3, 4, 6

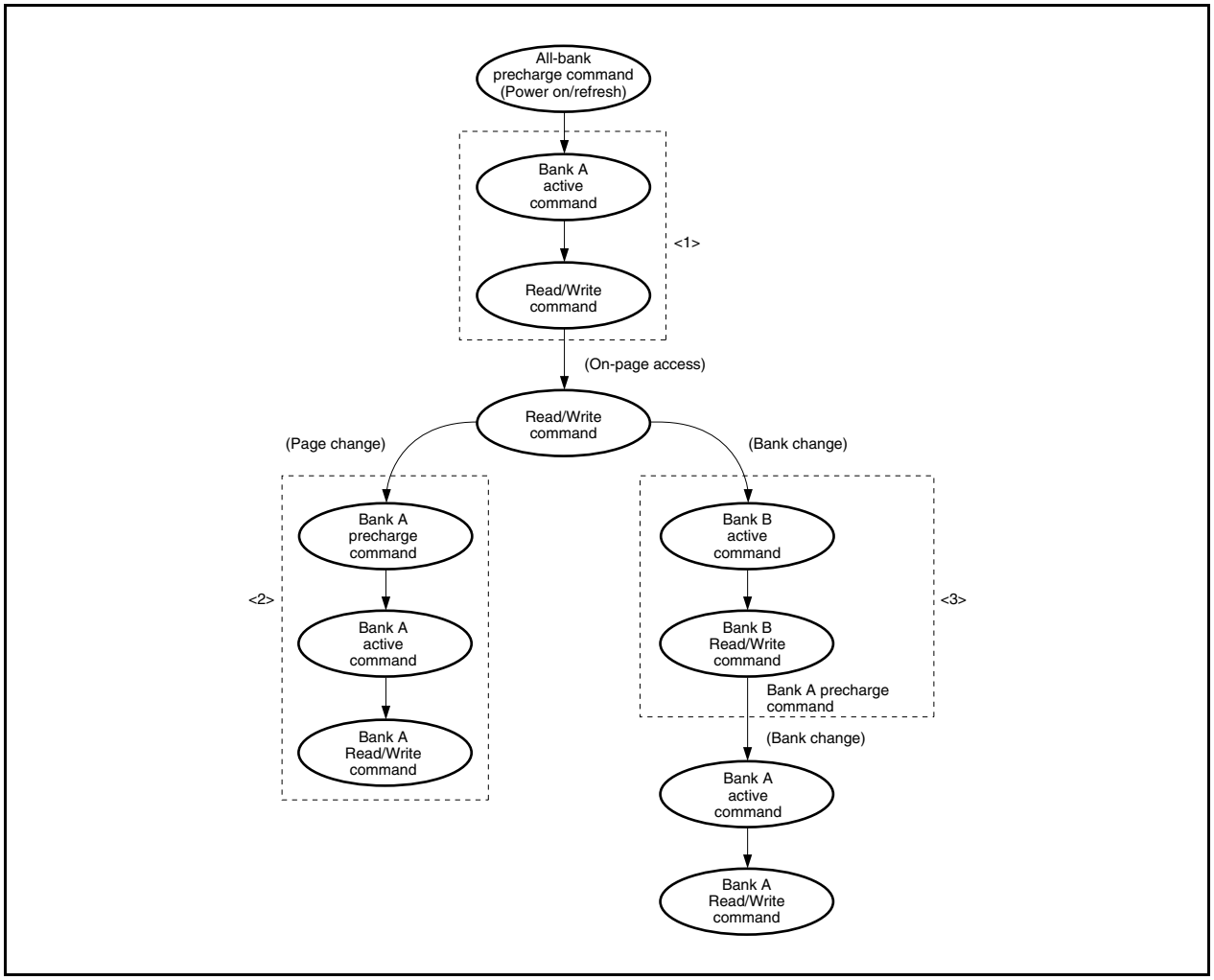
6.3.5 SDRAM access

During power-on or a refresh operation, the all-bank precharge command is always issued for SDRAM. When accessing SDRAM after that, therefore, the active command and read/write command are issued in that order (see <1> in **Figure 6-9**).

If a page change occurs following this, the precharge command, active command, and read/write command are issued in that order (see <2> in **Figure 6-9**).

If a bank change occurs, the active command and read/write command for the bank to be accessed next are issued in that order. Following this read/write command, the precharge command for the bank that was accessed before the bank currently being accessed will be issued (see <3> in **Figure 6-9**).

Figure 6-9. State Transition of SDRAM Access



(1) SDRAM single read cycle

The SDRAM single read cycle is a cycle for reading from SDRAM by executing a load instruction (LD) for the SDRAM area, by fetching an instruction, or by 2-cycle DMA transfer.

In the SDRAM single read cycle, the active command (ACT) and read command (RD) are issued to SDRAM in that order. During on-page access, however, only the read command is issued and the precharge command and active command are not issued. When a page change occurs in the same bank, the precharge command (PRE) is issued before the active command.

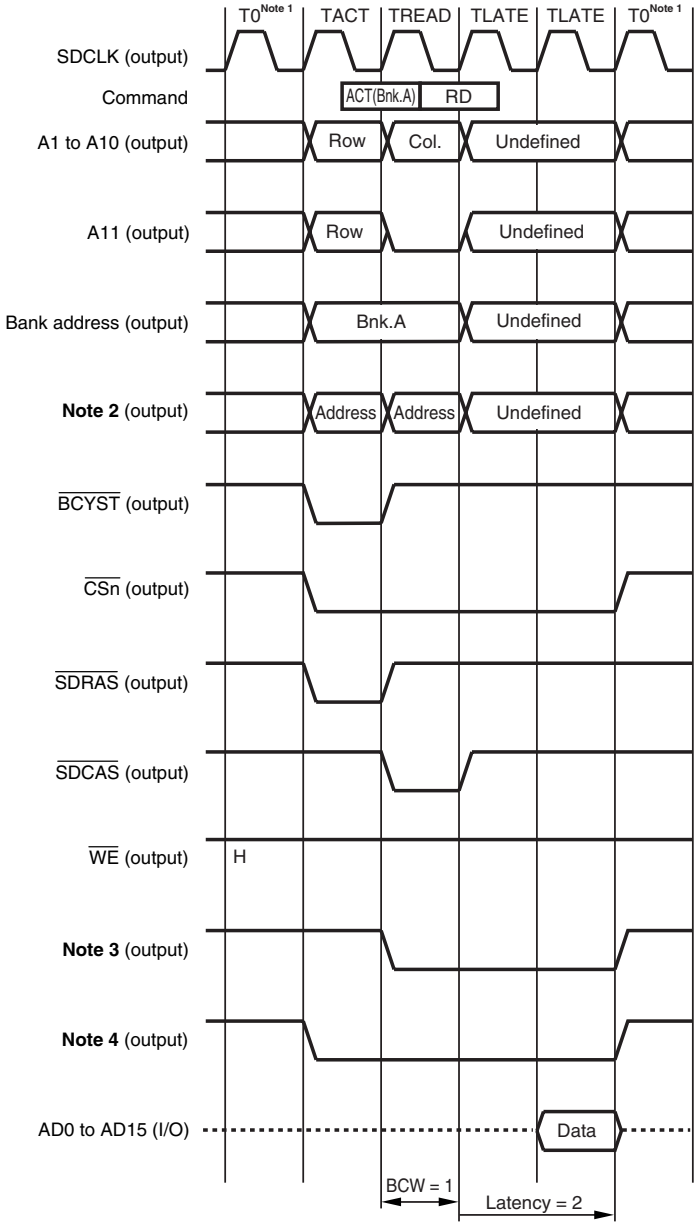
A one-state T0 cycle is always inserted immediately before all read commands activated by the CPU.

The number of idle states (TI) set by the BCC register are inserted after the read cycle (no idle states are inserted, however, if the BCC.BCn1 and BCC.BCn0 bits are 00) (n = 1, 3, 4, 6). The following shows the SDRAM single read cycle timing.

Caution When executing a write access to SRAM or external I/O after read accessing SDRAM, data conflict may occur depending on the SDRAM data output float delay time. In such a case, avoid data conflict by inserting an idle state in the SDRAM space via a setting in the BCC register.

Figure 6-10. SDRAM Single Read Cycle (1/5)

(a) Off-page access (latency = 2, BCW = 1, 16-bit bus width halfword access)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

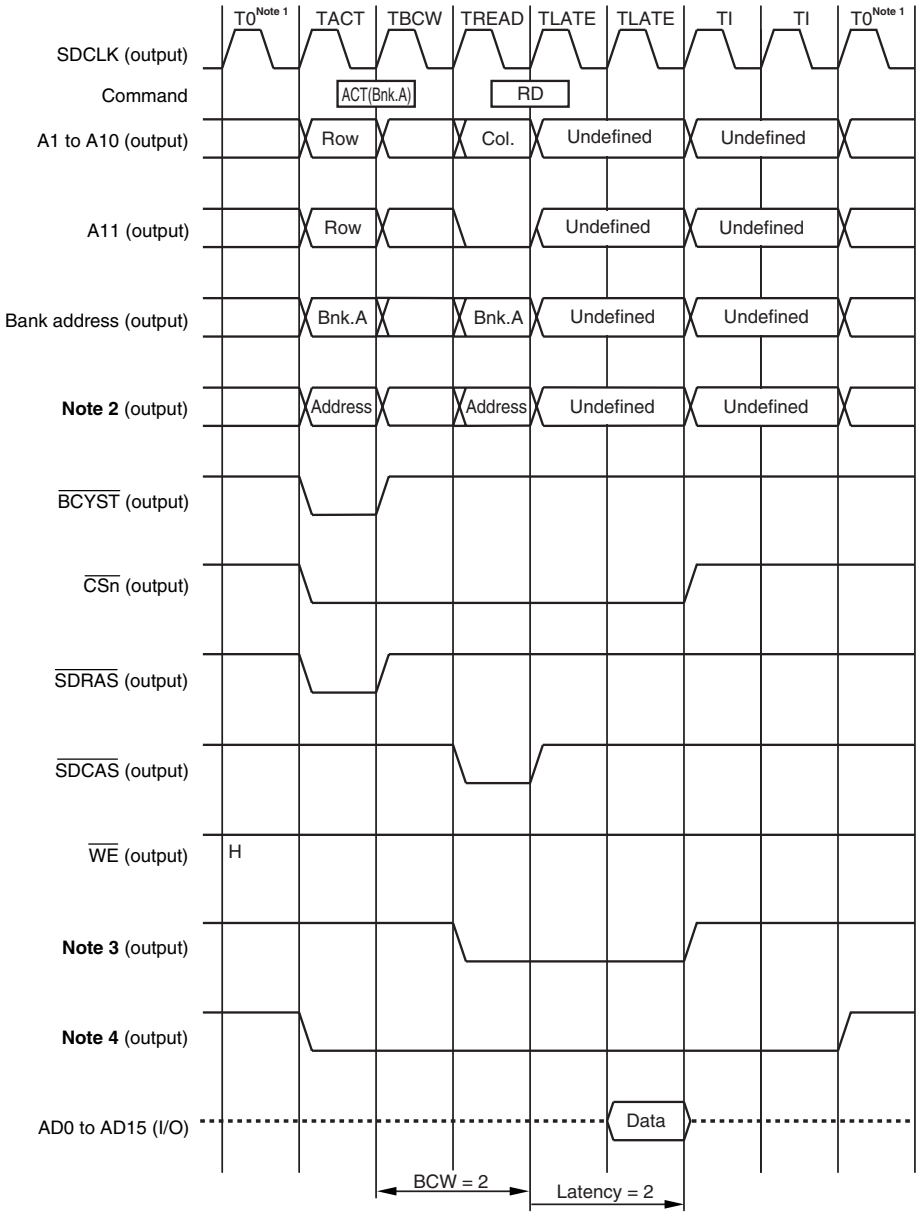
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address
Row: Row address

Figure 6-10. SDRAM Single Read Cycle (2/5)

(b) Off-page access (latency = 2, BCW = 2, two idle state inserted, 16-bit bus width halfword access)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

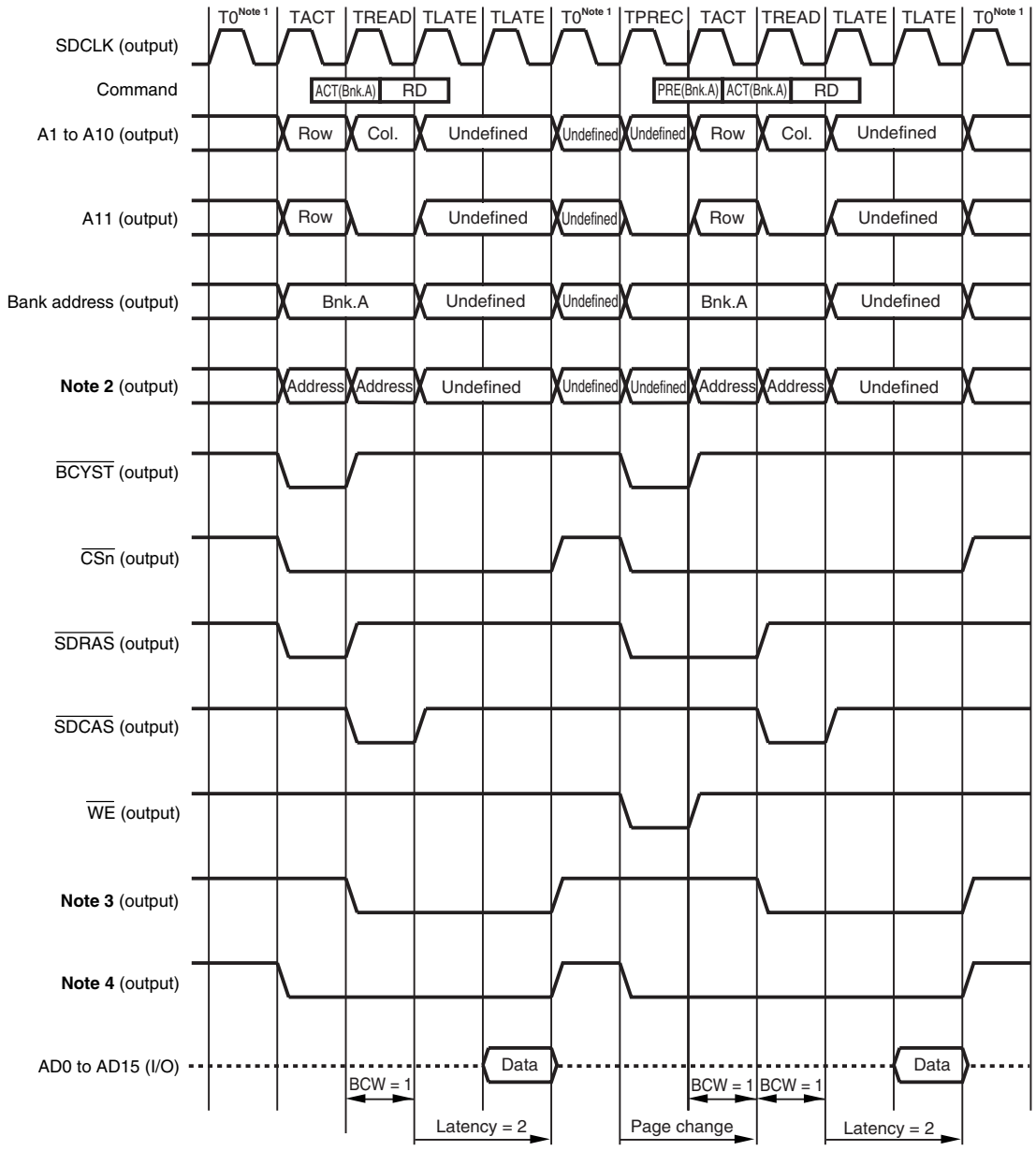
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address
Row: Row address

Figure 6-10. SDRAM Single Read Cycle (3/5)

(c) Off-page access (latency = 2, page change, BCW = 1, 16-bit bus width halfword access)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

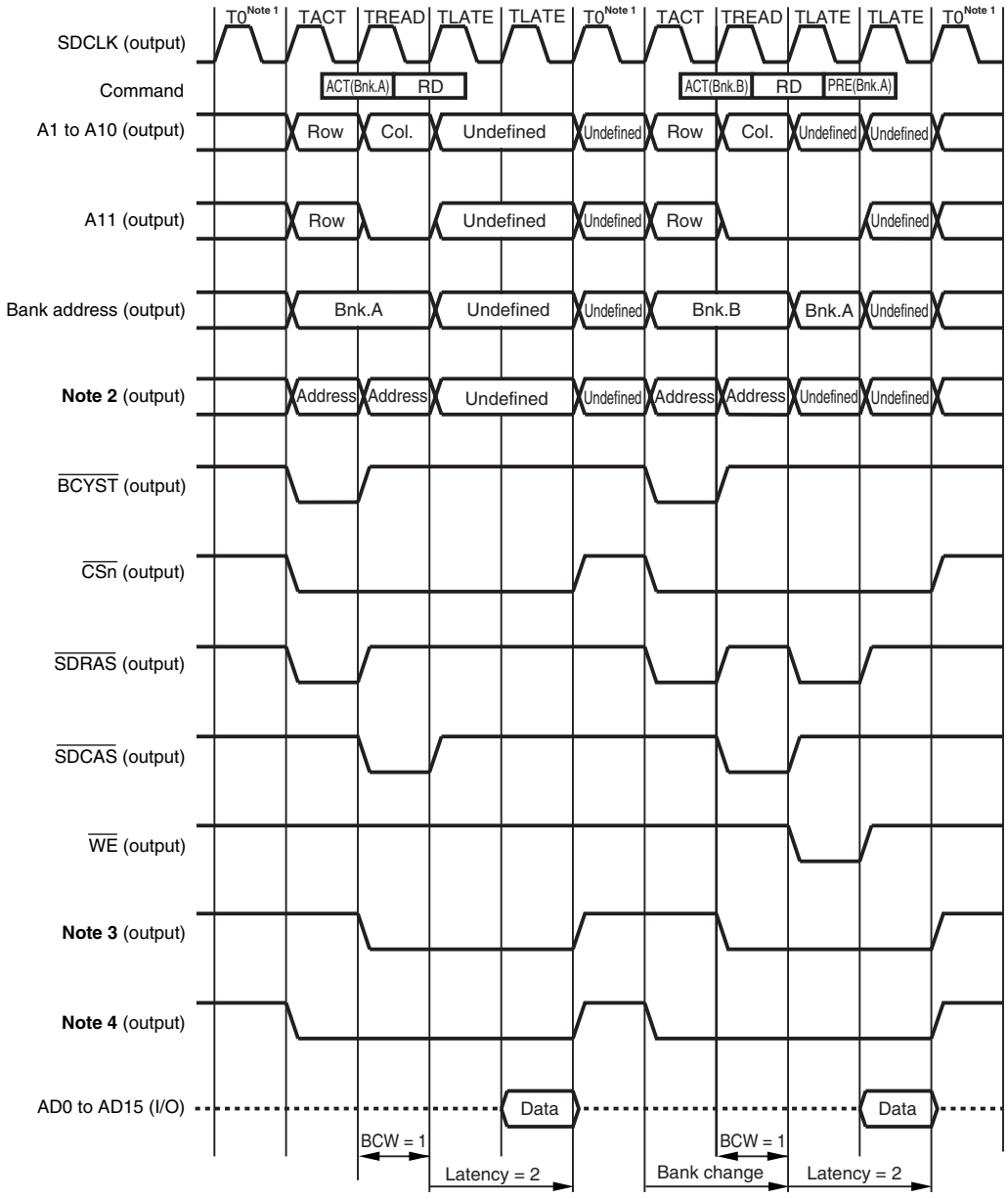
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address
Row: Row address

Figure 6-10. SDRAM Single Read Cycle (4/5)

(d) Off-page access (latency = 2, bank change, BCW = 1, 16-bit bus width halfword access)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

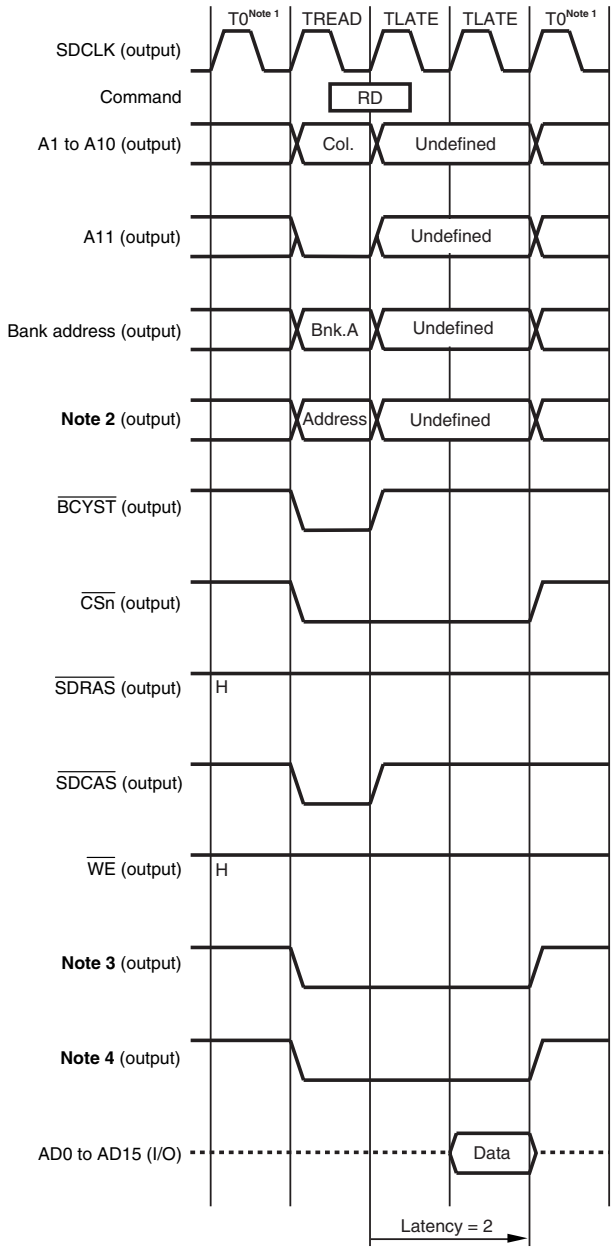
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address
Row: Row address

Figure 6-10. SDRAM Single Read Cycle (5/5)

(e) On-page access (latency = 2, 16-bit bus width halfword access)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address
Row: Row address

(2) SDRAM single write cycle

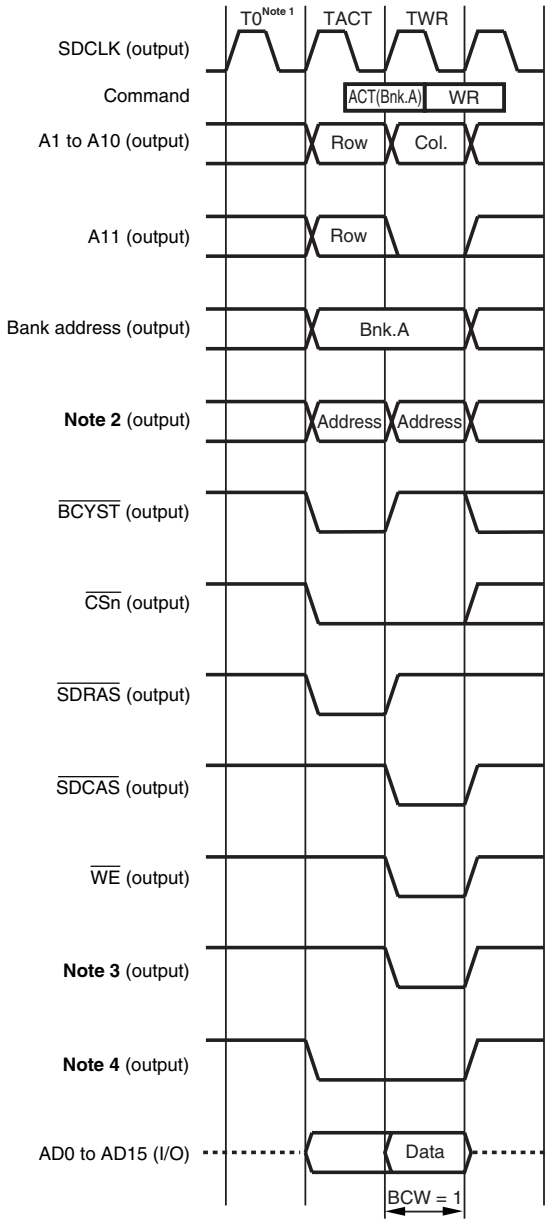
The SDRAM single write cycle is a cycle for writing to SDRAM by executing a write instruction (ST) for the SDRAM area or by 2-cycle DMA transfer.

In the SDRAM single write cycle, the active command (ACT) and write command (WR) are issued to SDRAM in that order. During on-page access, however, only the write command is issued and the precharge command and active command are not issued. When a page change occurs in the same bank, the precharge command (PRE) is issued before the active command.

The following shows the SDRAM single write cycle timing.

Figure 6-11. SDRAM Single Write Cycle (1/6)

(a) Off-page access (BCW = 1, 16-bit bus width halfword access)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

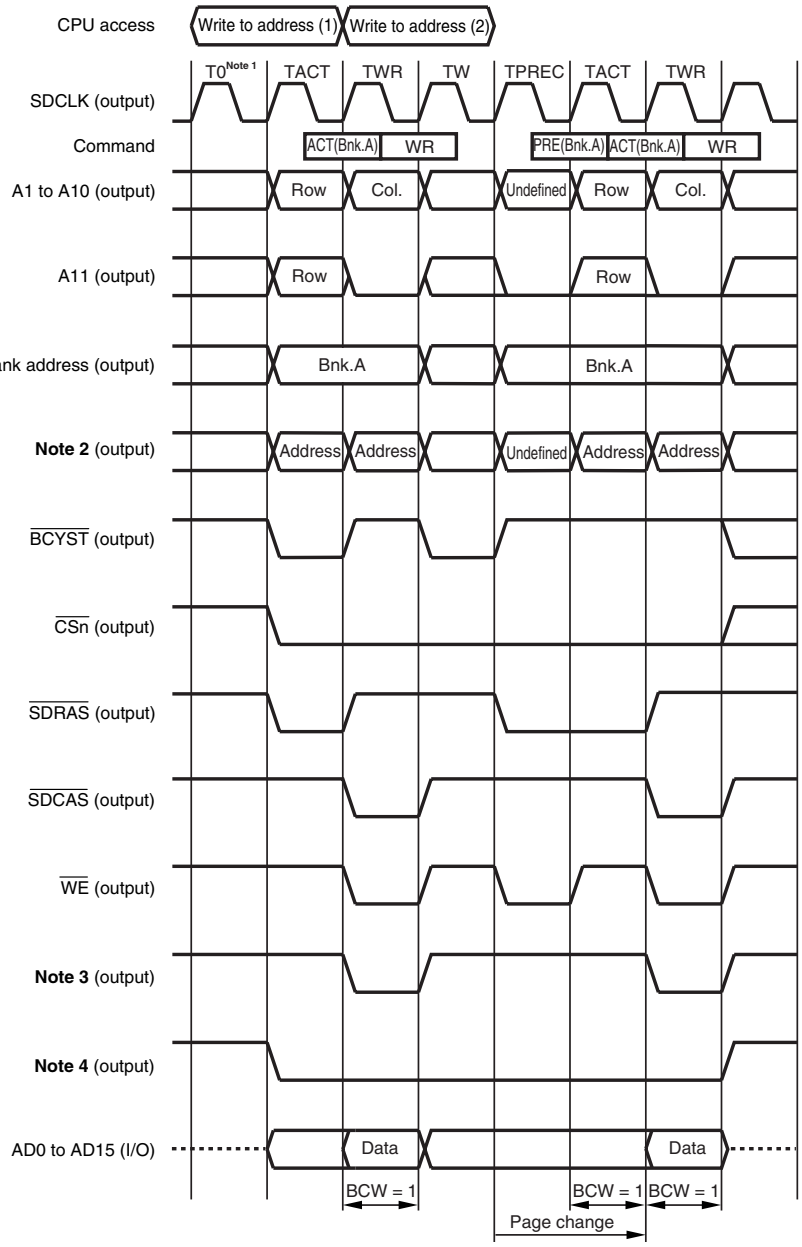
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address
Row: Row address

Figure 6-11. SDRAM Single Write Cycle (2/6)

(b) Off-page access (BCW = 1, page change, 16-bit bus width halfword access) (1/2)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. In $\overline{xWR}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{xBE}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

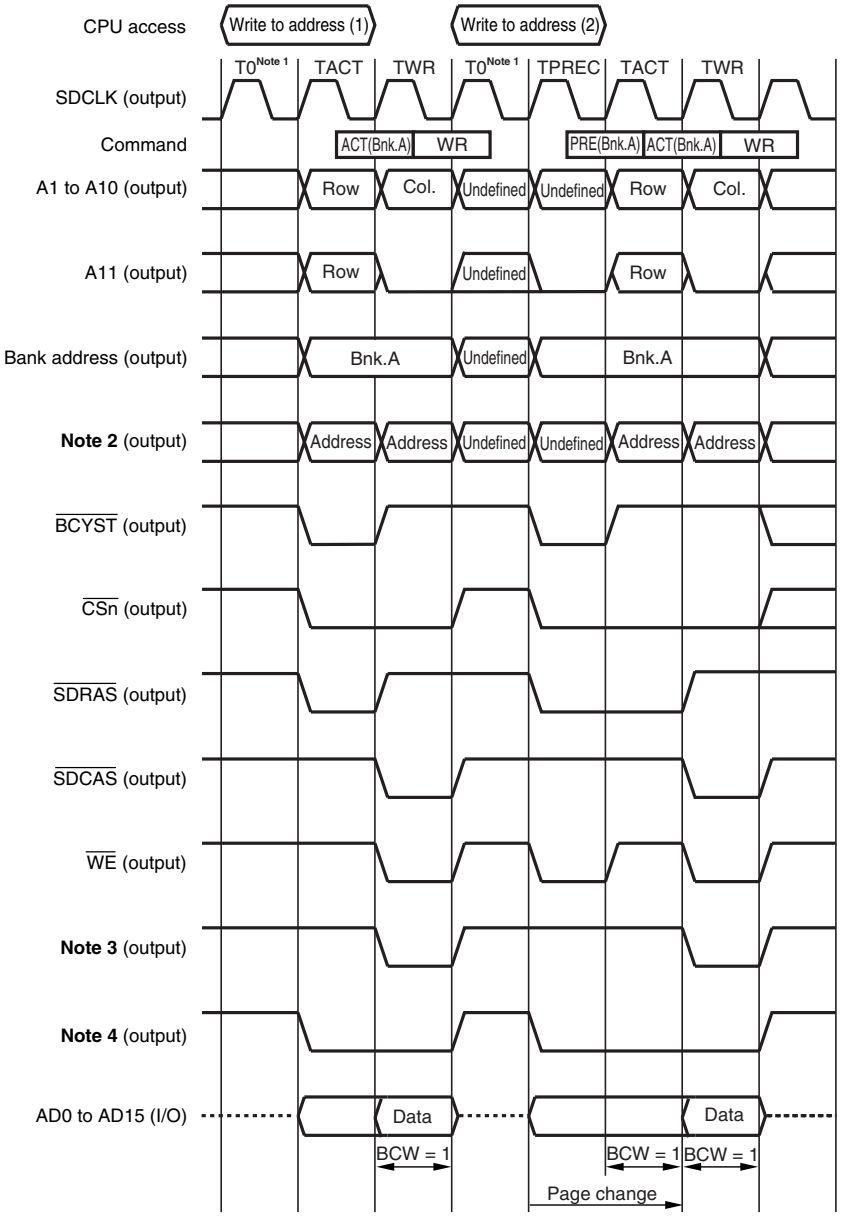
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address
Row: Row address

Figure 6-11. SDRAM Single Write Cycle (3/6)

(b) Off-page access (BCW = 1, page change, 16-bit bus width halfword access) (2/2)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

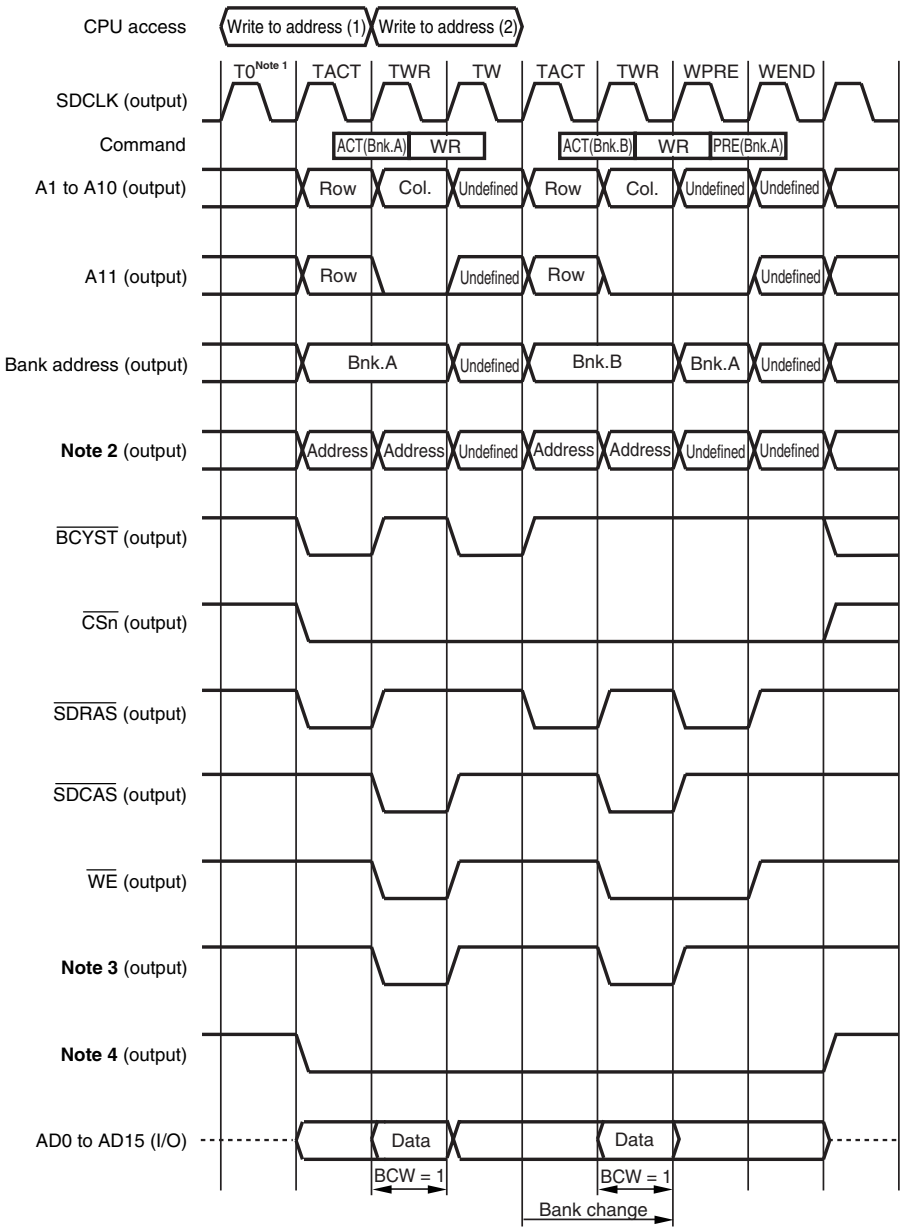
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address
Row: Row address

Figure 6-11. SDRAM Single Write Cycle (4/6)

(c) Off-page access (BCW = 1, bank change, 16-bit bus width halfword access) (1/2)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

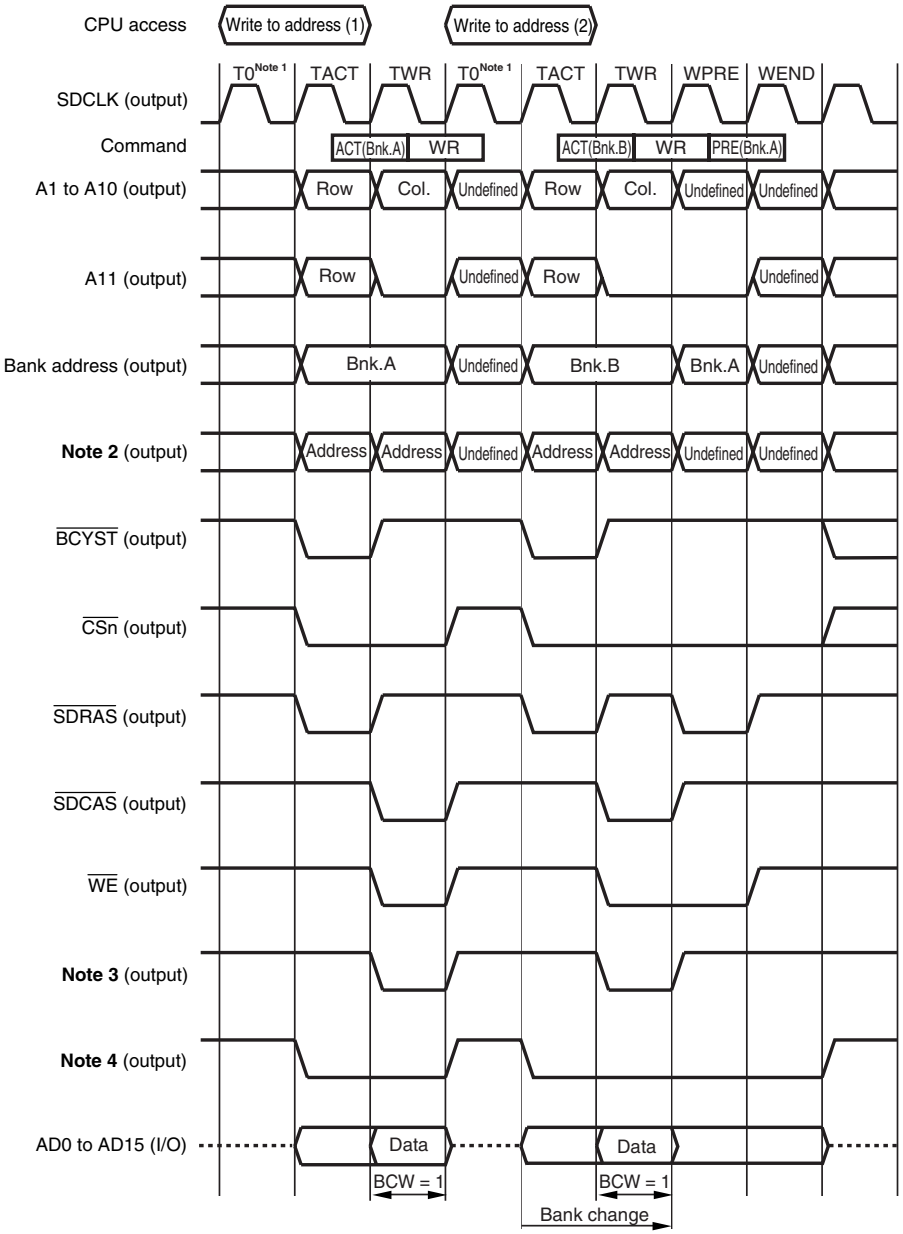
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address
Row: Row address

Figure 6-11. SDRAM Single Write Cycle (5/6)

(c) Off-page access (BCW = 1, bank change, 16-bit bus width halfword access) (2/2)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

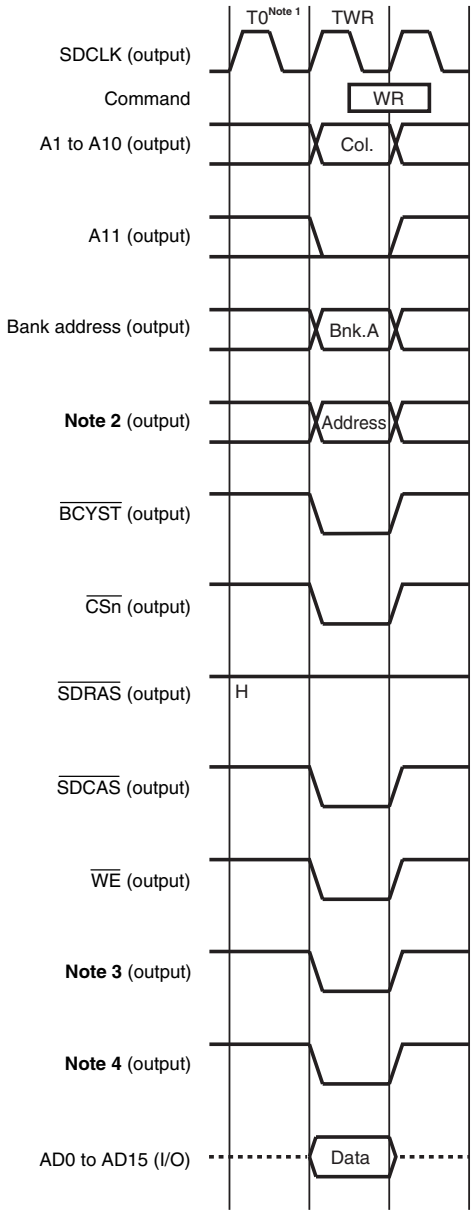
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address
Row: Row address

Figure 6-11. SDRAM Single Write Cycle (6/6)

(d) On-page access (16-bit bus width halfword access) (1/2)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address
Row: Row address

(3) SDRAM access timing control

The SDRAM access timing can be controlled by the SCRn register (n = 1, 3, 4, 6). For details, see **6.3.4 SDRAM configuration registers 1, 3, 4, 6 (SCR1, SCR3, SCR4, SCR6)**.

Caution Wait control by the $\overline{\text{WAIT}}$ pin is not available during SDRAM access.

(a) Number of waits from bank active command to read/write command

The number of wait states from bank active command issuance to read/write command issuance can be set by setting the SCRn.BCWn1 and SCRn.BCWn0 bits.

BCWn1, BCWn0 bits = 01: 1 wait
BCWn1, BCWn0 bits = 10: 2 waits
BCWn1, BCWn0 bits = 11: 3 waits

(b) Number of waits from precharge command to bank active command

The number of wait states from precharge command issuance to bank active command issuance can be set by setting the SCRn.BCWn1 and SCRn.BCWn0 bits.

BCWn1, BCWn0 bits = 01: 1 wait
BCWn1, BCWn0 bits = 10: 2 waits
BCWn1, BCWn0 bits = 11: 3 waits

(c) CAS latency setting when read

The CAS latency during a read operation can be set by setting the SCRn.LTMn2 to SCRn.LTMn0 bits.

LTMn2 to LTMn0 bits = 001: Latency = 1
LTMn2 to LTMn0 bits = 010: Latency = 2
LTMn2 to LTMn0 bits = 011: Latency = 3

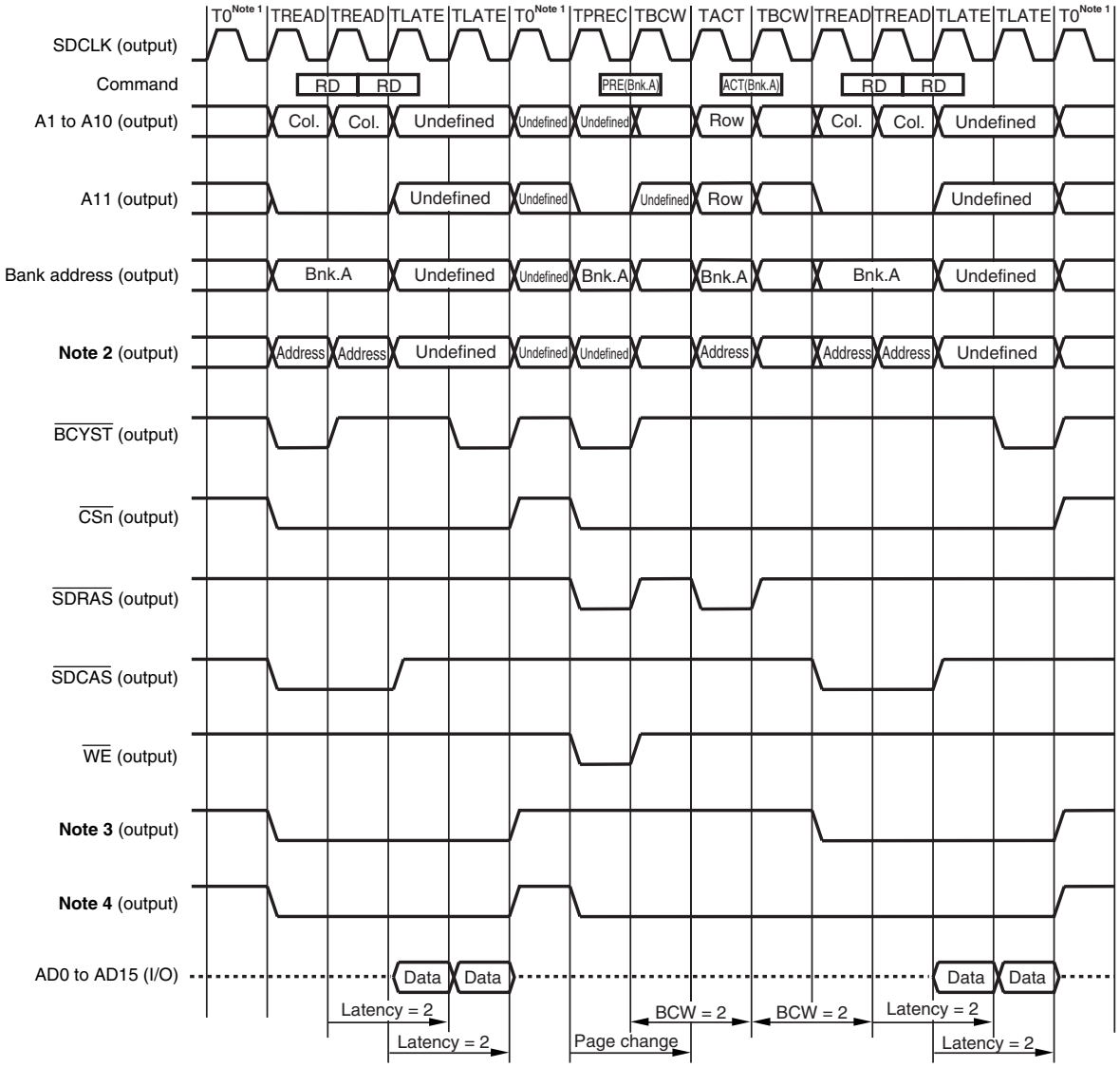
(d) Number of waits from refresh command to next command

The number of wait states from refresh command issuance to next command issuance can be set by setting the SCRn.BCWn1 and SCRn.BCWn0 bits. The number of wait states becomes four times the value set by BCWn1 and BCWn0 bits.

BCWn1, BCWn0 bits = 01: 4 waits
BCWn1, BCWn0 bits = 10: 8 waits
BCWn1, BCWn0 bits = 11: 12 waits

Figure 6-12. SDRAM Access Timing (1/6)

(a) Read (on-page → page change, 16-bit bus width word access, BCW = 2, latency = 2)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

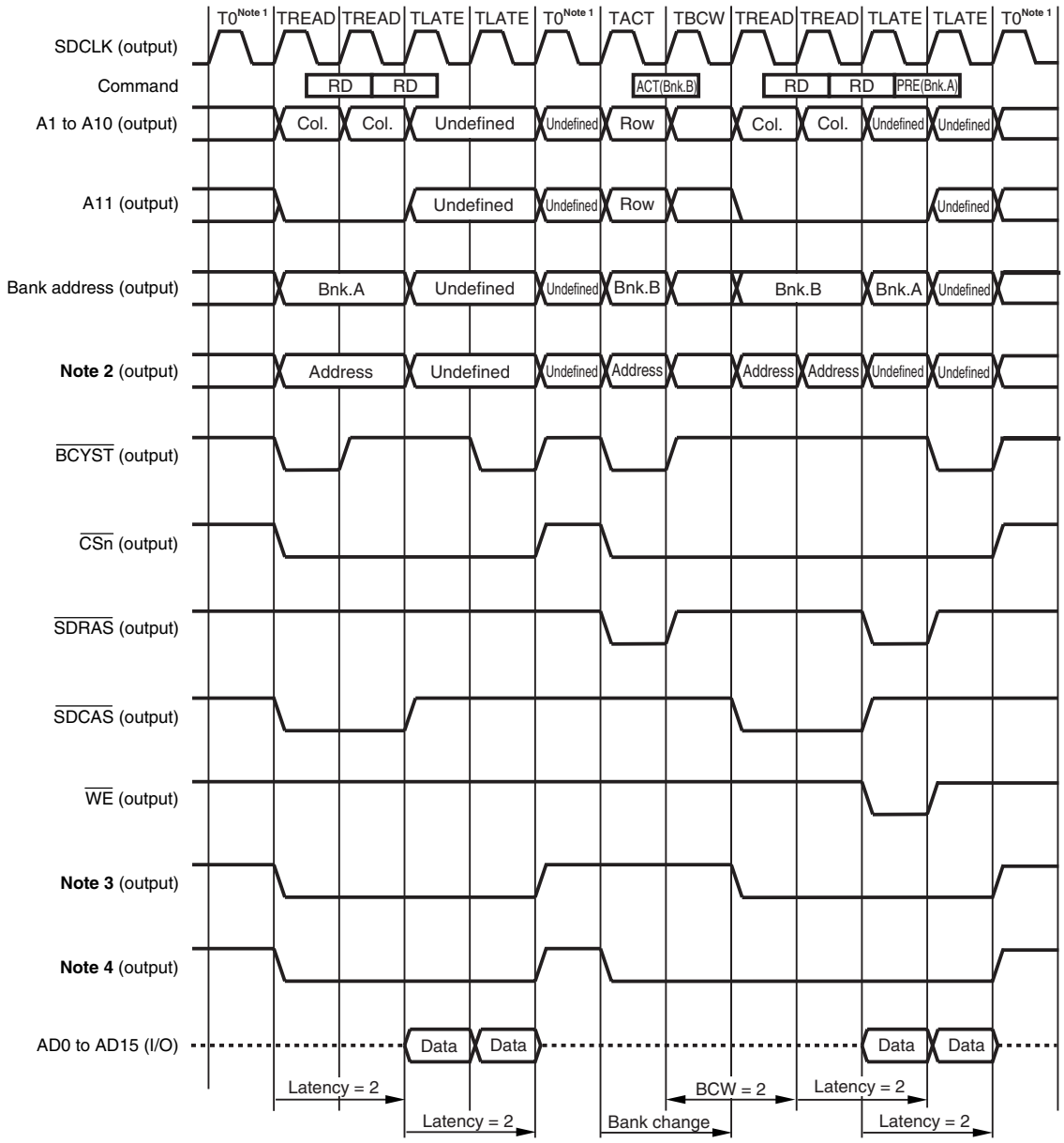
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address
Row: Row address

Figure 6-12. SDRAM Access Timing (2/6)

(b) Read (on-page → bank change, 16-bit bus width word access, BCW = 2, latency = 2)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

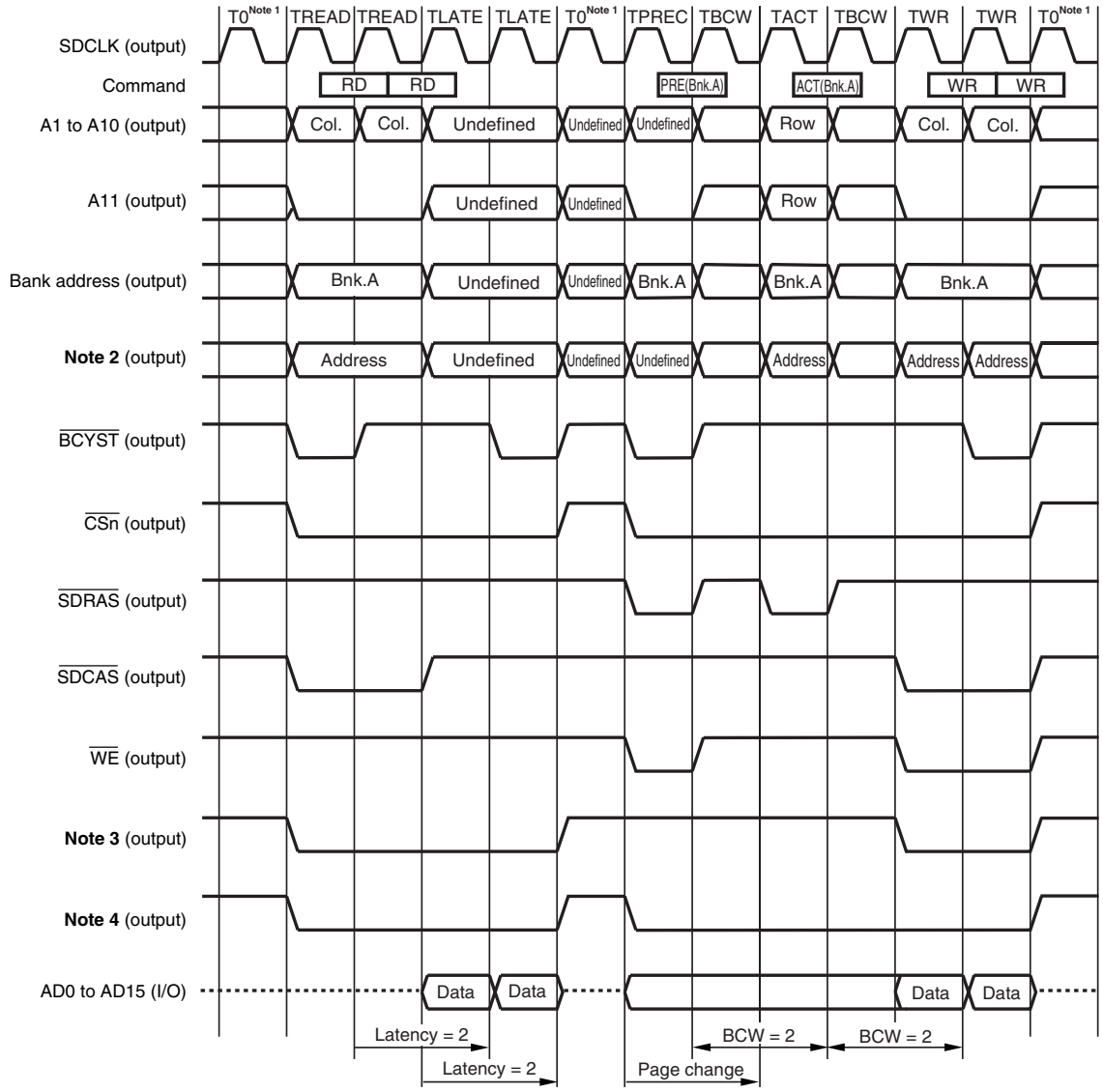
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address
Row: Row address

Figure 6-12. SDRAM Access Timing (3/6)

(c) Read → write (on-page → page change, 16-bit bus width word access, BCW = 2, latency = 2)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

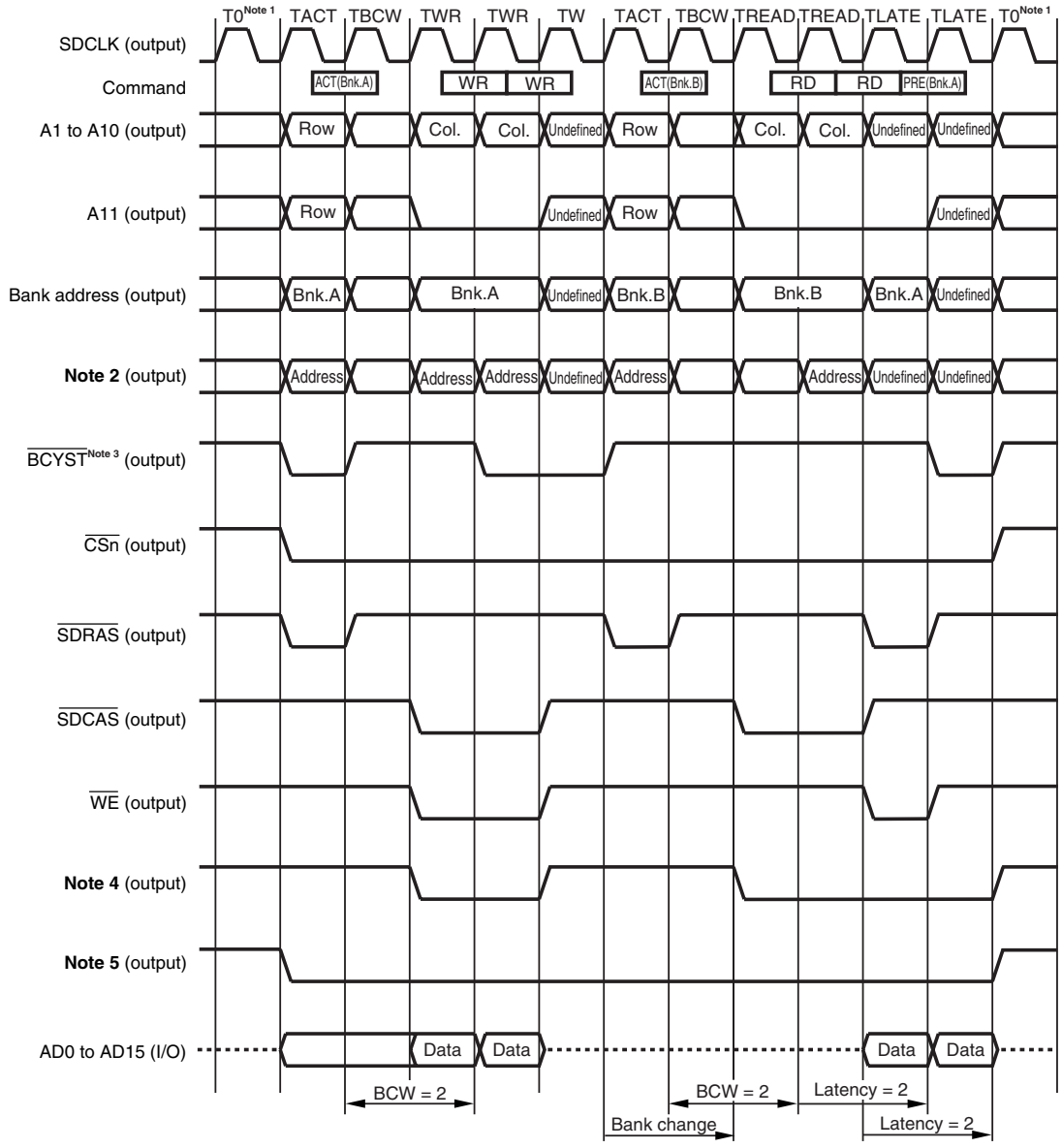
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address
Row: Row address

Figure 6-12. SDRAM Access Timing (4/6)

(d) Write → Read (bank change, 16-bit bus width word access, BCW = 2, latency = 2)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. Glitch may occur if $\overline{\text{BCYST}}$ outputs low levels successively.

4. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

5. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

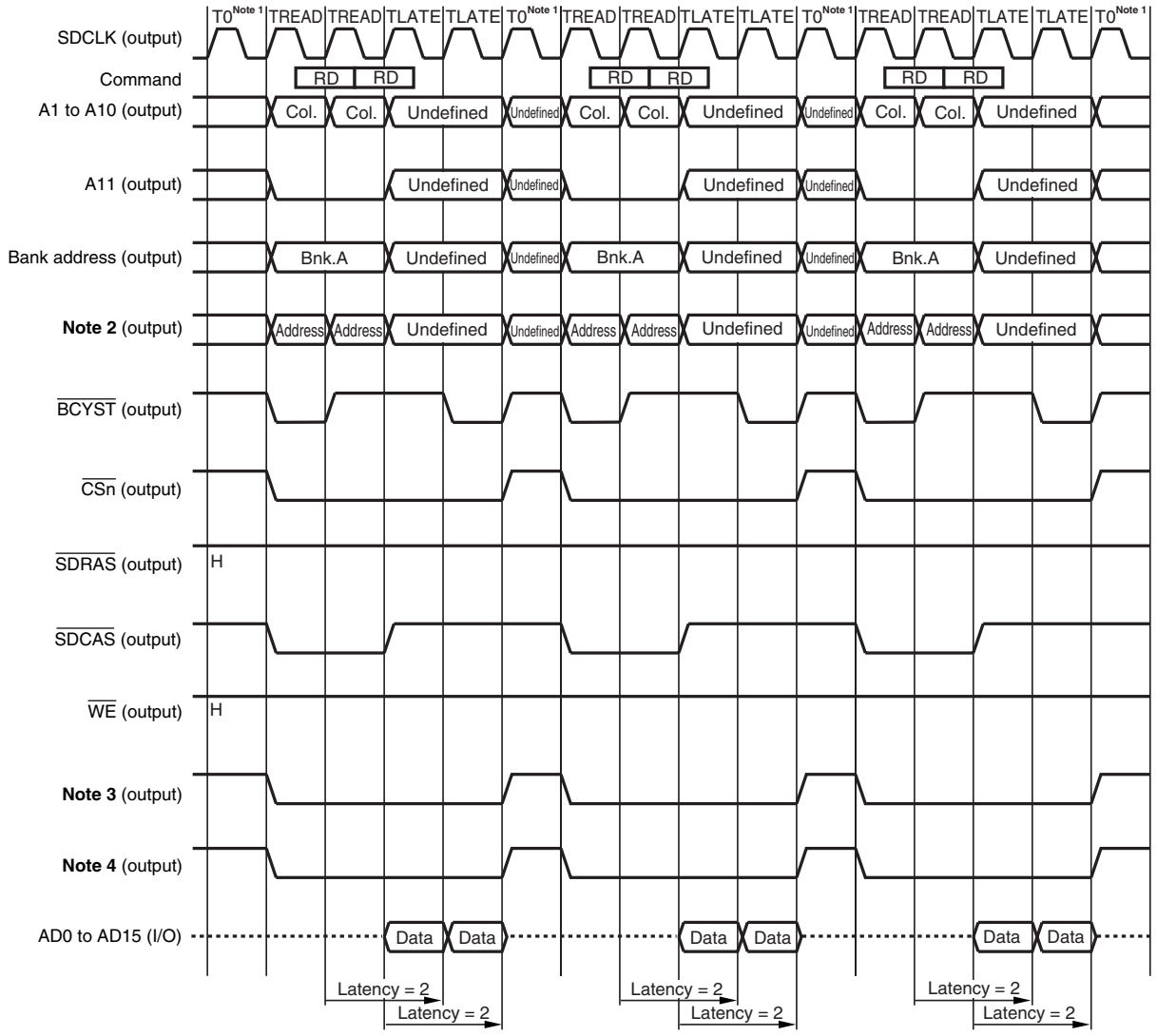
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address
Row: Row address

Figure 6-12. SDRAM Access Timing (5/6)

(e) Read (on-page successive access, 16-bit bus width word access, latency = 2)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

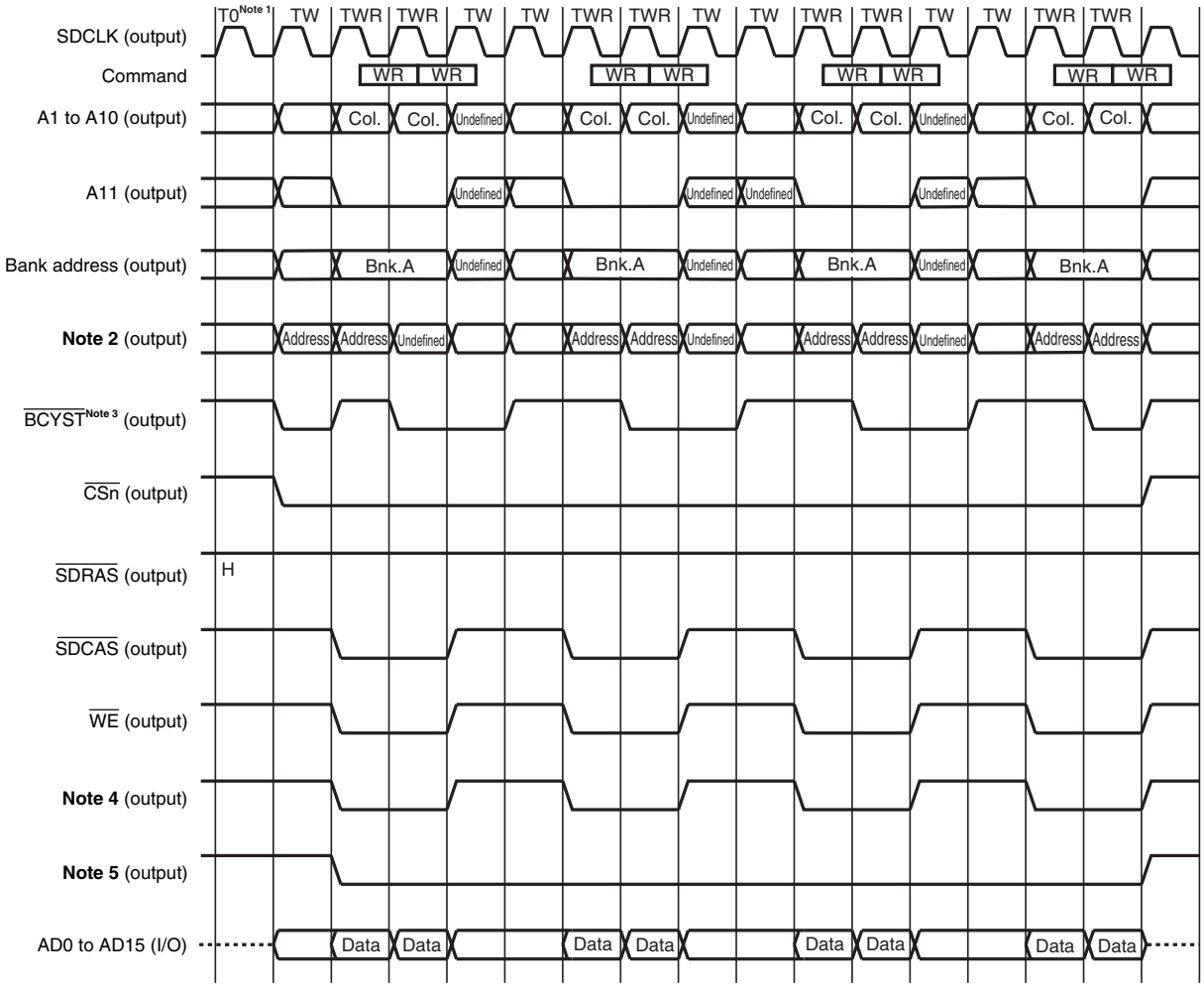
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address

Figure 6-12. SDRAM Access Timing (6/6)

(f) Write (on-page successive access, 16-bit bus width word access)



Notes 1. State (T0) inserted between bus cycles

2. Addresses other than bank address, A1 to A10, and A11

3. Glitch may occur if $\overline{\text{BCYST}}$ outputs low levels successively.

4. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

5. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

3. Bnk.: Bank address
Col.: Column address

6.3.6 Refresh control function

The V850E/MA3 can generate a refresh cycle. The refresh cycle is set by the RFS1, RFS3, RFS4, and RFS6 registers. The RFSn register corresponds to $\overline{CSn}$ (n = 1, 3, 4, 6). For example, to connect SDRAM to $\overline{CS1}$, set the RFS1 register.

When another bus master occupies the external bus, the DRAM controller cannot occupy the external bus. In this case, the DRAM controller issues a refresh request to the bus master by changing the $\overline{REFRQ}$ signal to active (low level).

During a refresh operation, the address bus retains the state it was in just before the refresh cycle.

(1) SDRAM refresh control registers 1, 3, 4, 6 (RFS1, RFS3, RFS4, RFS6)

These registers are used to enable or disable a refresh and set the refresh interval. The refresh interval is determined by the following calculation formula.

Refresh interval (μs) = Refresh count clock (T_{RCY}) $\times$ Interval factor

The refresh count clock and interval factor are determined by the RFSn.RCCn1 and RFSn.RCCn0 bits and RFSn.RIN5n to RFSn.RIN0n bits, respectively.

Note that n corresponds to the register number (1, 3, 4, 6) of the SCR1, SCR3, SCR4, and SCR6 registers.

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

Reset input clears these registers to 0000H.

Cautions 1. Write the RFSn register after reset and do not change its value after that. However, the set value of the RFSn register can be changed only if it is necessary to change the refresh interval of SDRAM as a result of changing the set value of the CKC register or PCC register (internal system clock (f_{CLK})). Do not access the external memory area until the initial setting of the RFSn register is completed.

2. Change the setting of the RFSn register in the following procedure (n = 1, 3, 4, or 6).

<1> Clear the BCTm.MEn bit to 0 (m = 0 or 1, n = 1, 3, 4, or 6).

<2> Clear the RFSn.RENn bit to 0.

<3> Change the clock cycle.

<4> Set the MEn bit to 1.

<5> Set a new refresh interval to the RFSn register. Clear the RENn bit to 0.

<6> Write a value same as that currently set to the SCRn register to the SCRn register.

<7> Confirm that the SCRn.WCFn bit is set to 1 and then set the RENn bit to 1. Write a value set in <5> above to the bits other than the RENn bit.

<8> SDRAM can now be accessed.

For how to change the clock cycle, see 7.3 (1) Processor clock control register (PCC) and 7.3 (2) Clock control register (CKC).

3. To change the refresh interval, consider and set a value at which refreshing can be performed in time even while the refresh interval is changed. For details, see 6.3.6 (1) (a) Notes on changing refresh interval.

After reset: 0000H R/W Address: RFS1 FFFFF4A6H, RFS3 FFFFF4AEH, RFS4 FFFFF4B2H, RFS6 FFFFF4BAH

	15	14	13	12	11	10	9	8
RFSn	RENn	0	0	0	0	0	RCCn1	RCCn0
(n = 1, 3, 4, 6)	7	6	5	4	3	2	1	0
	0	0	RINn5	RINn4	RINn3	RINn2	RINn1	RINn0

RENn	CBR (automatic) refresh enable
0	Refresh disabled
1	Refresh enabled

RCCn1	RCCn0	Specification of refresh count clock (T_{RCY})
0	0	32/BUSCLK
0	1	128/BUSCLK
1	0	256/BUSCLK
1	1	Setting prohibited

RINn5	RINn4	RINn3	RINn2	RINn1	RINn0	Interval factor
0	0	0	0	0	0	1
0	0	0	0	0	1	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	0	63
1	1	1	1	1	1	64

Sets the interval factor of the interval timer for the generation of the refresh timing using the RINn5 to RINn0 bits.

Table 6-1. Example of Interval Factor Settings

Specified Refresh Interval Value (μs)	Refresh Count Clock (T_{RCY})	Interval Factor Value[Notes 1, 2]		
		BUSCLK = 30 MHz	BUSCLK = 40 MHz	BUSCLK = 50 MHz
15.6	32/BUSCLK	14 (14.9)	19 (15.2)	24 (15.4)
	128/BUSCLK	3 (12.8)	4 (12.8)	6 (15.4)
	256/BUSCLK	1 (8.5)	2 (12.8)	3 (15.4)
7.8	32/BUSCLK	7 (7.5)	9 (7.2)	12 (7.7)
	128/BUSCLK	1 (4.3)	2 (6.4)	3 (7.7)
	256/BUSCLK	–	1 (6.4)	1 (5.1)

Notes 1. The interval factor is set by the RFSn.RINn0 to RFSn.RINn5 bits (n = 1, 3, 4, 6).

2. The values in parentheses are the calculated values for the refresh interval (μs).
Refresh interval (μs) = Refresh count clock (T_{RCY}) × Interval factor

The V850E/MA3 can automatically generate a CBR (automatic) refresh cycle and a self-refresh cycle.

(a) Notes on changing refresh interval

Figure 6-13 shows the internal status and external bus status while the refresh interval is changed when SDRAM is connected to $\overline{CS1}$ and $\overline{CS3}$. In this case, SDRAM connected to $\overline{CS1}$ is not refreshed during periods <1>, <3>, and <4>, and SDRAM connected to $\overline{CS3}$ is not refreshed during periods <1>, <2>, and <4>.

Periods <1> and <4> may extend or shorten depending on the status of the internal system bus or on a factor such as a bus hold request from the external bus. Periods <2> and <3> changes depending on the set value of the RFSn register.

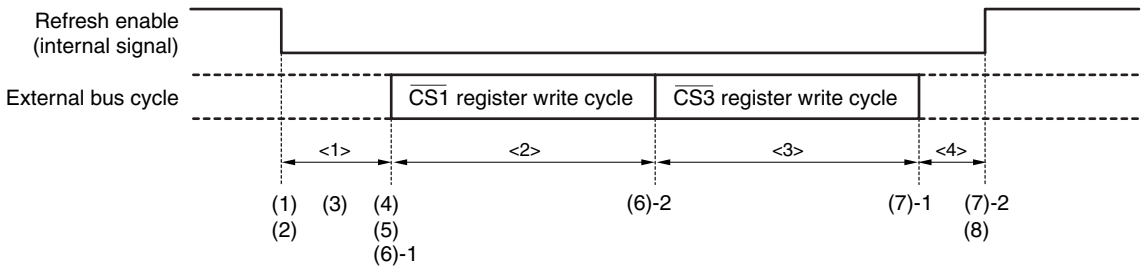
To change the refresh interval, therefore, periods <1> to <4> in Figure 6-13 during which SDRAM is not refreshed must be taken into consideration and a value to set to the RFSn register must be determined. Even when SDRAM is connected only to one $\overline{CS}$ area, the refresh interval value must be changed, taking periods <1> to <4> in Figure 6-13 into consideration.

Cautions 1. The refresh command is issued eight times in the register write cycle, regardless of the set value of the RENn bit (n = 1, 3, 4, 6).

2. While the RENn bit is 0, the $\overline{REFRQ}$ signal does not operate. This should be noted when the external bus master references the $\overline{REFRQ}$ signal.

<R> **3. While the BCTm.MEn bit is 0 (n = 1, 3 when m = 0, n = 4, 6 when m = 1), the self refresh status is not set even when the IDLE or software STOP mode is set.**

Figure 6-13. Internal Status and External Bus Status When Refresh Interval Is Changed (If SDRAM Is Connected to $\overline{CS1}$ and $\overline{CS3}$)

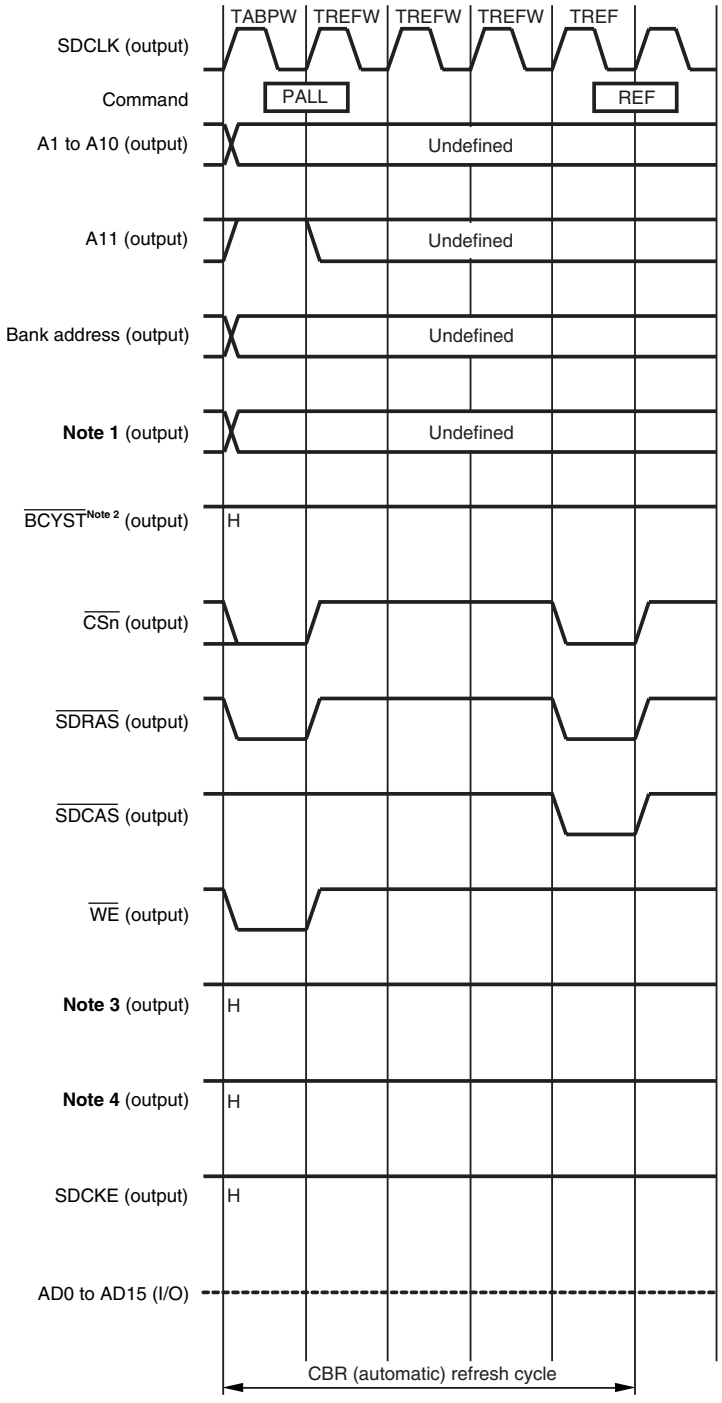


(1) Clear the BCT0.ME1 and BCT3.ME3 bits to 0.
(2) Clear the RFS1.REN1 and RFS3.REN3 bits to 0.
(3) Change the clock cycle.
(4) Set the ME1 and ME3 bits to 1.
(5) Set a new refresh interval to the RFS1 and RFS3 registers. However, clear the REN1 and REN3 bits to 0.
(6)-1 Write a value same as that currently set to the SCR1 register to the SCR1 register.
(6)-2 Write a value same as that currently set to the SCR3 register to the SCR3 register.
(7)-1 Confirm that the SCR1.WCF1 and SCR3.WCF3 bits are set to 1.
(7)-2 Set the REN1 and REN3 bits to 1. Write a value set in (5) to the bits other than the REN1 and REN3 bits.
(8) SDRAM can be accessed.

(2) CBR (automatic) refresh cycle

In the CBR (automatic) refresh cycle, the auto-refresh command (REF) is issued four clocks after the precharge command for all banks (PALL) is issued.

Figure 6-14. CBR (Automatic) Refresh Cycle (16-Bit Bus Width)



Notes 1. Addresses other than bank address, A1 to A10, and A11

2. Glitch may be generated if $\overline{\text{BCYST}}$ outputs high levels in the CBR (automatic) refresh cycle.

3. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

6.3.7 Self-refresh control function

In the case of transition to the IDLE or software STOP mode, the DRAM controller generates the self-refresh cycle.

Caution The internal ROM and internal RAM can be accessed even in the self-refresh cycle. However, access to an on-chip peripheral I/O register or external device is held pending until the self-refresh cycle is released.

To release the self-refresh cycle, use one of the three methods below.

(1) Release by NMI input

(a) In the case of self-refresh cycle in IDLE mode

To release the self-refresh cycle, make the $\overline{\text{SDRAS}}$ and $\overline{\text{SDCAS}}$ signals inactive immediately.

(b) In the case of self-refresh cycle in software STOP mode

To release the self-refresh cycle, make the $\overline{\text{SDRAS}}$ and $\overline{\text{SDCAS}}$ signals inactive after stabilizing oscillation.

(2) Release by INTP0n input (n = 000, 001, 004, 005, 010 to 013, 021, 022, 050, 051, 106, 107, 114, 115, 124 to 126, 130 to 134, 137)

(a) In the case of self-refresh cycle in IDLE mode

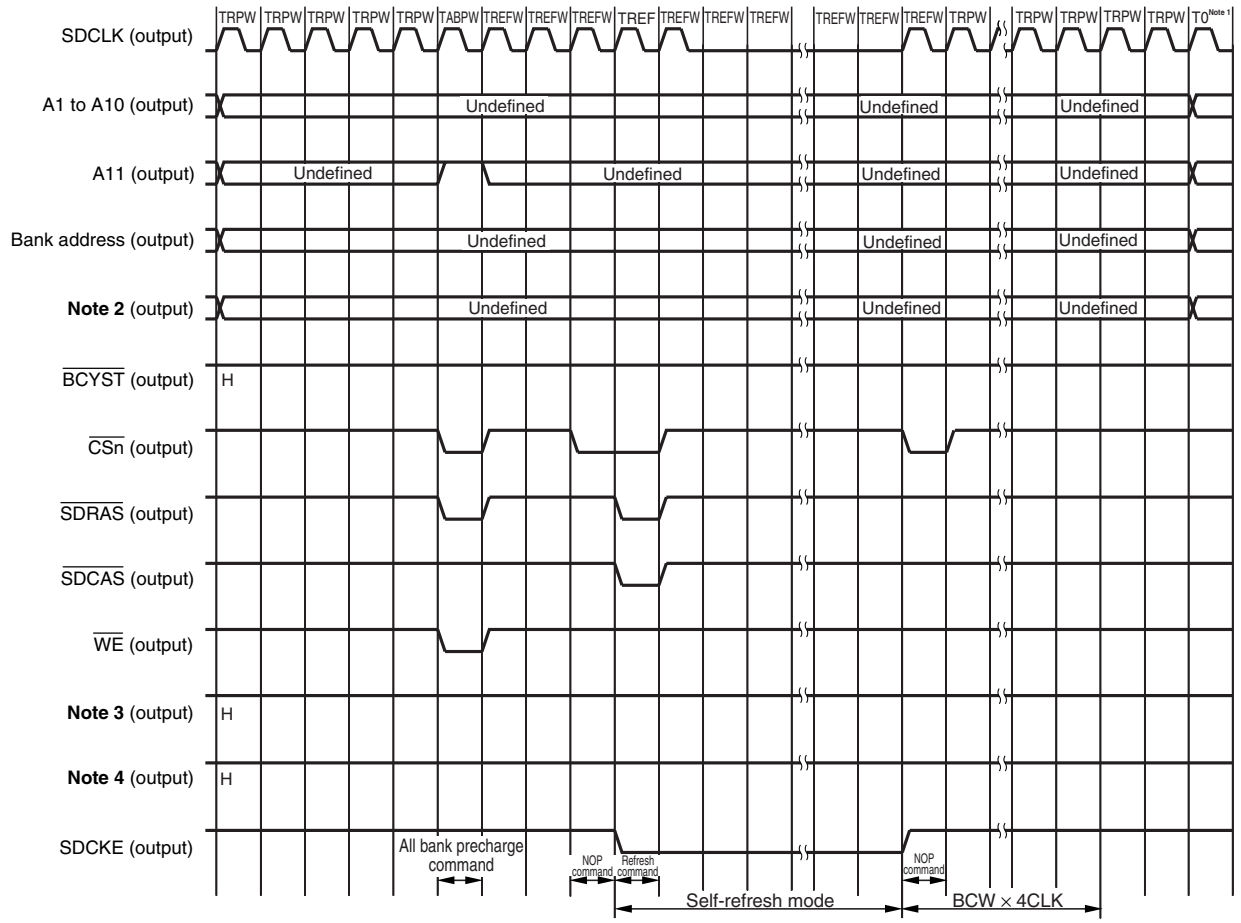
To release the self-refresh cycle, make the $\overline{\text{SDRAS}}$ and $\overline{\text{SDCAS}}$ signals inactive immediately.

(b) In the case of self-refresh cycle in software STOP mode

To release the self-refresh cycle, make the $\overline{\text{SDRAS}}$ and $\overline{\text{SDCAS}}$ signals inactive after stabilizing oscillation.

(3) Release by $\overline{\text{RESET}}$ input

Figure 6-15. Self Timing (16-Bit Bus Width)



Notes 1. State (T0) inserted between bys cycles

2. Addresses other than bank address, A1 to A10, and A11

3. In $\overline{\text{xWR}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 0)

4. In $\overline{\text{xBE}}$ output mode/xDQM output mode (PFCCT.PFCCTm bit = 1)

Remarks 1. Wait states of "BCW × 4 CLK = Number of wait states set by SCRn.BCWn1 and SCRn.BCWn0 bits × 4 CLK" is inserted.

2. n = 1, 3, 4, or 6, m = 0 or 1, x = U or L

6.3.8 SDRAM initialization sequence

Be sure to initialize SDRAM when applying power. Initialize SDRAM in the following procedure.

(1) Set the registers of SDRAM (other than SDRAM configuration register n (SCRn), SDRAM refresh control register n (RFSn)).

- Bus cycle type configuration registers 0, 1 (BCT0, BCT1)
- Bus cycle control register (BCC)

(2) Set other than RENn bit of SDRAM refresh control registers 1, 3, 4, 6 (RFS1, RFS3, RFS4, RFS6). Clear the RENn bit to 0.

(3) Set SDRAM configuration registers 1, 3, 4, 6 (SCR1, SCR3, SCR4, SCR6). When writing data to the SCRn register, the following commands are issued for SDRAM in the order shown below.

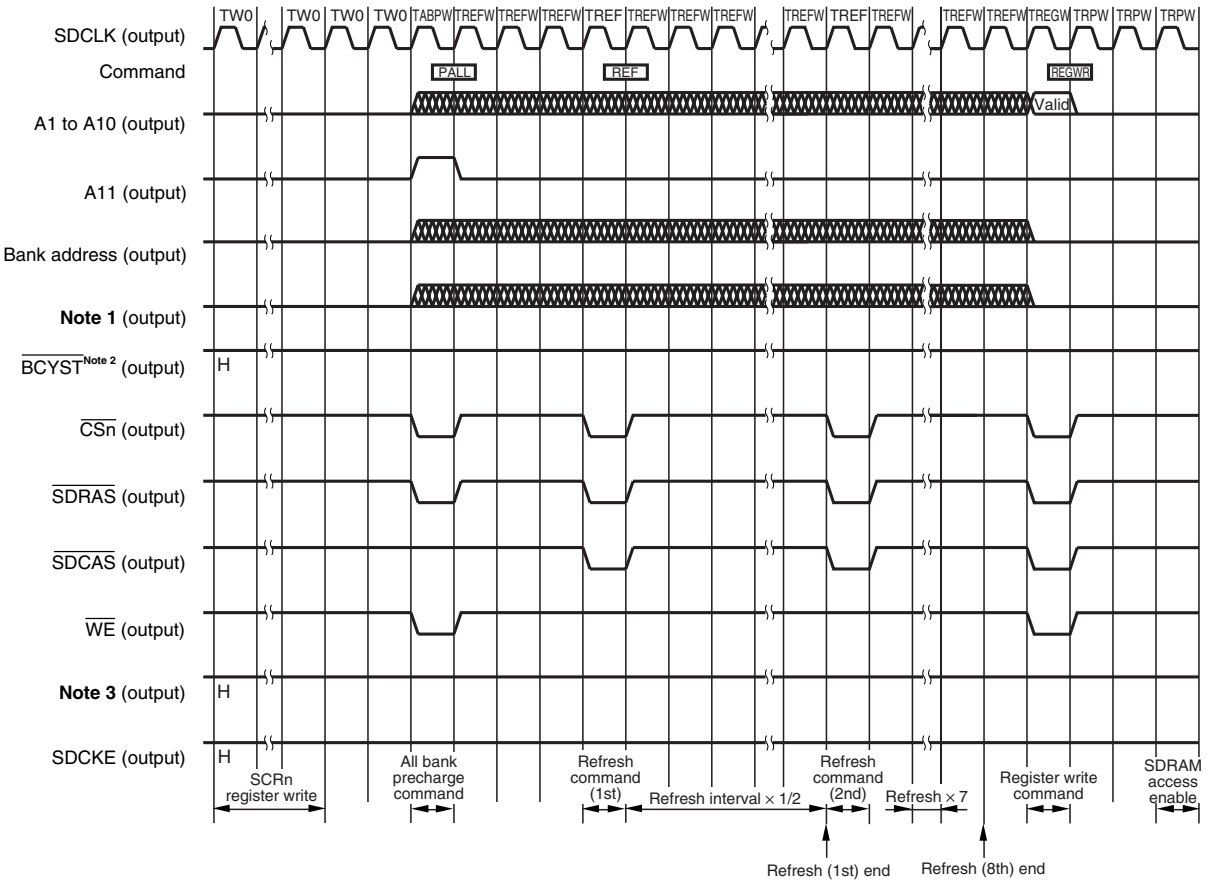
- All-bank precharge command
- Refresh command (8 times)
- Command that is used to set a mode register

(4) After confirming that all the SDRAM initialization is complete using the SCRn.WCFn bit, set the RFSn.REN bit to 1.
Set the value set in step (2) to other bits than the RENn bit.

Caution If it is necessary to make the input levels of the UDQM and LDQM pins high until initialization of SDRAM is complete, do not change the set values of the PFCCT1 and PFCCT0 bits of the PFCCT register and do not write to the external device until initialization of SDRAM is complete.

An example of timing of a SDRAM register write operation is shown below.

Figure 6-16. SDRAM Register Write Operation Timing (16-Bit Bus Width)



Notes 1. Addresses other than bank address, A1 to A10, and A11

2. Glitch may be generated if $\overline{\text{BCYST}}$ outputs high levels in the SDRAM initialization sequence.

3. UDQM, LDQM

Remark n = 1, 3, 4, or 6

CHAPTER 7 CLOCK GENERATOR

7.1 Overview

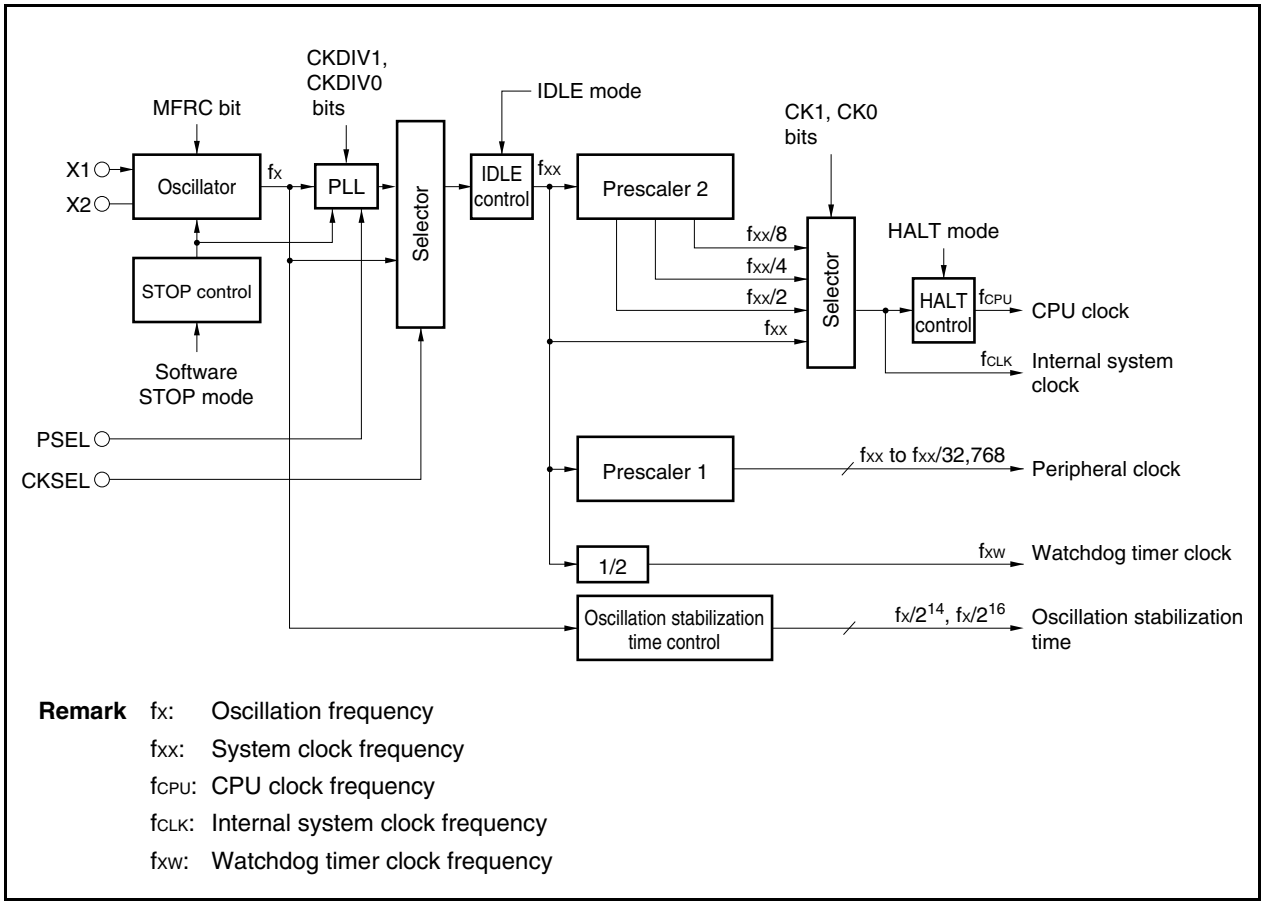
The features of clock generator are as follows.

- Oscillator
  - f_X = 4 to 8 MHz (in PLL mode)
  - f_X = 5 to 25 MHz (in clock-through mode)
- Multiply (×1.25/2.5/5/10) 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

7.2 Configuration

Figure 7-1. Clock Generator



Remark f_X: Oscillation frequency
f_{XX}: System clock frequency
f_{CPU}: CPU clock frequency
f_{CLK}: Internal system clock frequency
f_{XW}: Watchdog timer clock frequency

(1) Oscillator

The main resonator oscillates the following frequencies (f_X):

- f_X = 4 to 8 MHz (in PLL mode)
- f_X = 5 to 25 MHz (in clock-through mode)

(2) STOP control

This circuit generates a control signal that stops oscillation of the oscillator.
Oscillation of the oscillator is stopped in the software STOP mode.

(3) IDLE control

A control signal that stops clock supply to the internal circuits other than the oscillator is generated.
In the IDLE mode, clock supply to the internal circuits other than the oscillator is stopped.

(4) HALT control

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

(5) PLL

This circuit multiplies the clock (f_X) generated by the oscillator by 1.25, 2.5, 5, or 10.
It operates in two modes: clock-through mode in which f_X is output as is, and PLL mode in which a multiplied clock is output. These modes can be selected by the CKSEL pin input.
Change the setting of the PSEL pin according to the frequency band.

(6) Prescaler 1

This prescaler generates the clock (f_{XX} to f_{XX}/32,768) to be supplied to the following on-chip peripheral functions: TMP0 to TMP2, TMQ0, TMD0 to TMD3, TMENC10, UARTA0 to UARTA3, CSIB0 to CSIB2, I^2C, ADC, and DAC.

(7) Prescaler 2

This circuit divides the system clock (f_{XX}).
The clock generated by prescaler 2 (f_{XX} to f_{XX}/8) is supplied to the selector that generates the CPU clock (f_{CPU}) and internal system clock (f_{CLK}).
f_{CLK} is the clock supplied to the INTC, ROM correction, ROM, and RAM blocks.

(8) Oscillation stabilization time control

The time until oscillation is stabilized after the software STOP mode has been released by an interrupt request signal is counted.
The count clock can be selected from $2^{14}/f_X$ and $2^{16}/f_X$.

7.3 Control Registers

The clock generator is controlled by following five registers.

- Processor clock control register (PCC)
- Clock control register (CKC)
- Power save control register (PSC)
- Power save mode register (PSMR)
- Oscillation stabilization time select register (OSTS)

(1) Processor clock control register (PCC)

The PCC register is a special register. Data can be written to this register only in combination of specific sequences (see **3.4.9 Special registers**).

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF828H

	7	6	<5>	4	3	2	1	0
PCC	0	0	MFRC	0	0	0	CK1	CK0

MFRC	Use of main clock on-chip feedback resistor
0	Used (resonator is connected to X1, X2 pins)
1	Not used (external clock is connected to X1 pin)

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 4, 6, and 7 to "0".

2. After the CK1 and CK0 bits have been set, the clock is switched within 10 CPU clocks (f_{CPU}).

(2) Clock control register (CKC)

The CKC register is an 8-bit register that controls the system clock (fxx) in PLL mode. It can be written to only in a specific sequence combination so that it cannot easily be overwritten by mistake due to an inadvertent program loop.

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

Reset input clears this register to 00H.

Caution Do not change the CKDIV1 and CKDIV0 bits in clock-through mode.

After reset: 00H R/W Address: FFFFF822H

	7	6	5	4	3	2	1	0
CKC	0	0	0	0	0	0	CKDIV1	CKDIV0

CKDIV1	CKDIV0	System clock (fxx)
0	0	1.25 × fx
0	1	2.5 × fx
1	0	5 × fx
1	1	10 × fx

Cautions 1. The system clock frequency switches in 10 system clocks (fxx) after the CKDIV1 and CKDIV0 bits are set.

2. Be sure to clear bits 2 to 7 to "0".

Remark fx: Oscillation frequency

Example Clock generator settings

Operation Mode	CKSEL Pin	CKC Register		Input Clock (fx)	System Clock (fxx)
		CKDIV0 Bit	CKDIV0 Bit		
Clock-through mode	High-level input	×	×	25 MHz	25 MHz
PLL mode	Low-level input	0	0	8 MHz	10 MHz
		0	1	8 MHz	20 MHz
		1	0	8 MHz	40 MHz
		1	1	8 MHz	80 MHz

Remark ×: don't care

(3) 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 **3.4.9 Special registers**).

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

Reset input clears this register to 00H.

After reset: 00H R/W After reset: FFFFF1FEH

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

NMIM	Control of non-maskable interrupt request (NMI) from NMI pin[Note 1]
0	Standby mode release enabled by NMI request
1	Standby mode release disabled by NMI request

INTM	Control of all maskable interrupt requests[Note 1] (INTxx[Note 2])
0	Standby mode release enabled by INTxx request
1	Standby mode release disabled by INTxx request

STB	Setting operation mode[Note 3]
0	Normal mode
1	Standby mode

Notes 1. Setting these bits is valid only in the IDLE/software STOP mode.

2. For details, see **Tables 20-1 Interrupt Source List**.

3. For the setting procedure, see **21.7 Procedure for Setting and Restoring from IDLE and Software STOP Modes**.

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

2. To set the IDLE mode or software STOP mode, set the PSMR.PSM bit first and then set the STB bit to 1.

(4) Power save mode register (PSMR)

The PSMR register is an 8-bit register that controls the operation status in the power save mode and the clock operation.

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

Reset input clears this register to 00H.

After reset: 00H R/W After reset: FFFFF820H

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

PSM	Specification of operation in software standby mode
0	IDLE mode
1	Software STOP mode

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

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

(5) Oscillation stabilization time select register (OSTS)

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

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

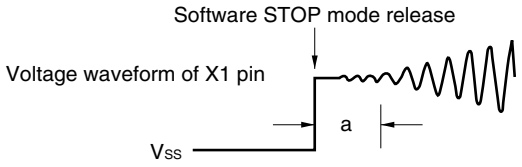
Reset input sets this register to 01H.

After reset: 01H R/W Address: FFFFF6C0H

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

OSTS0		Selection of oscillation stabilization time	
		fx	
		8 MHz	6 MHz
0	$2^{14}/f_x$	2.048 ms	2.731 ms
1	$2^{16}/f_x$	8.192 ms	10.923 ms

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



2. Set the OSTS register to 01H when using an oscillator.

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

4. When the software STOP mode is released by $\overline{\text{RESET}}$ pin input, the oscillation stabilization time does not elapse. Secure the oscillation stabilization time by the low-level width of the $\overline{\text{RESET}}$ pin input.

Remark fx: Oscillation frequency

7.4 Operation

7.4.1 Operation of each clock

The following table shows the operation status of each clock.

Table 7-1. Each Clock Operation Status

Clock Source		Power Save Mode	Oscillator (f_X)	PLL	Internal System Clock (f_{CLK})	Peripheral Clock (f_{XX} to f_{XX}/32,768)	CPU Clock (f_{CPU})	Watchdog Timer Clock (f_{XW})
PLL mode	Oscillation with resonator	Normal operation	√	√	√	√	√	√
		HALT mode	√	√	√	√	×	√
		IDLE mode	√	√	×	×	×	×
		Software STOP mode	×	×	×	×	×	×
		Oscillation stabilization period	√	√	×	×	×	√
		Reset period	√	×	×	×	×	×
	External clock	Normal operation	×	√	√	√	√	√
		HALT mode	×	√	√	√	×	√
		IDLE mode	×	√	×	×	×	×
		Software STOP mode	×	×	×	×	×	×
		Oscillation stabilization period	×	√	×	×	×	√
		Reset period	×	×	×	×	×	×
Clock-through mode	External clock	Normal operation	×	×	√	√	√	√
		HALT mode	×	×	√	√	×	√
		IDLE mode	×	×	×	×	×	×
		Software STOP mode	×	×	×	×	×	×
		Oscillation stabilization period	×	×	×	×	×	√
		Reset period	×	×	√Note	√Note	√Note	√Note

Note The clock is supplied during the reset period but the operations of the CPU and all the peripheral functions are stopped.

Remark √: Operating
×: Stopped

7.4.2 External clock input function

An external clock can be directly input to the oscillator. In this case, input the clock signal only to the X1 pin (leave the X2 pin open). Set the MFRC bit of the PCC register to 1 (to cut the feedback resistor). Note, however, that oscillation stabilization time is inserted even in the external clock mode.

7.5 PLL Function

7.5.1 Overview

The CPU and the operating clock of the on-chip peripheral function can be switched between output of the oscillation frequency multiplied by 1.25, 2.5, 5, or 10, and clock-through mode.

When PLL function is used: Input clock = 4 to 8 MHz (f_{XX}: 5 to 80 MHz)
Clock-through mode: Input clock = 5 to 25 MHz (f_{XX}: 5 to 25 MHz)

7.5.2 Selecting system clock

In the V850E/MA3, the system clock is selected according to the input levels of the CKSEL and PSEL pins, and the setting of the CKC register.

The following system clocks can be selected.

Table 7-2. Selecting System Clock

CKSEL	PSEL	Internal Clock Selection		Input Clock Frequency (f_X)	CKC Register		System Clock Frequency (f_{XX})
					CKDIV1	CKDIV0	
L	L	PLL Mode	Low-frequency mode	4.0 to 5.5 MHz	0	0	5 to 6.875 MHz
					0	1	10 to 13.75 MHz
					1	0	20 to 27.5 MHz
					1	1	40 to 55 MHz
L	H		High-frequency mode	5.5 to 8.0 MHz	0	0	6.875 to 10 MHz
					0	1	13.75 to 20 MHz
					1	0	27.5 to 40 MHz
					1	1	55 to 80 MHz
H	×	Clock-through mode		5.0 to 25.0 MHz	×	×	5 to 25 MHz

Caution Fix the input levels of the CKSEL and PSEL pins during the reset period and do not change the levels during operation. Otherwise, the operation will not be guaranteed.

Remark ×: don't care

7.5.3 PLL mode

In the PLL mode, the oscillation frequency (fx) is multiplied by the PLL to generate a system clock (fxx). fxx can be selected from fx multiplied by 1.25, 2.5, 5, or 10, according to the setting of the clock control register (CKC).

Fix the input level of the PSEL pin to the high or low level according to the value of the oscillation frequency (fx).

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. The V850E/MA3 automatically secures the following lockup time after release of reset.

Lockup time = 2^{14}/fx + α = Approx. 2.048 ms (fx = 8 MHz) to Approx. 4.096 ms (fx = 4 MHz)

7.5.4 Clock-through mode

In the clock-through mode, a system clock (fxx) of the same frequency as the oscillation frequency (fx) is generated.

The V850E/MA3 requires the following time until the CPU starts operation after release of reset.

CPU operation start time = 2^{16}/fx + α = Approx. 2.62144 ms (fx = 25 MHz) to Approx. 13.1072 ms (fx = 5 MHz)

CHAPTER 8 16-BIT TIMER/EVENT COUNTER P (TMP)

Timer P (TMP) is a 16-bit timer/event counter.
The V850E/MA3 has three timer/event counter channels, TMP0 to TMP2.

8.1 Overview

An outline of TMPn is shown below.

- Clock selection: 8 ways
- Capture/trigger input pins: 2
- External event count input pins: 1
- External trigger input pins: 1
- Timer/counters: 1
- Capture/compare registers: 2
- Capture/compare match interrupt request signals: 2
- Timer output pins: 2

Remark n = 0 to 2

8.2 Functions

TMPn has the following functions.

- Interval timer
- External event counter
- External trigger pulse output
- One-shot pulse output
- PWM output
- Free-running timer
- Pulse width measurement
- Timer tuning operation function (TMP2 only)

Remark n = 0 to 2

8.3 Configuration

TMPn includes the following hardware.

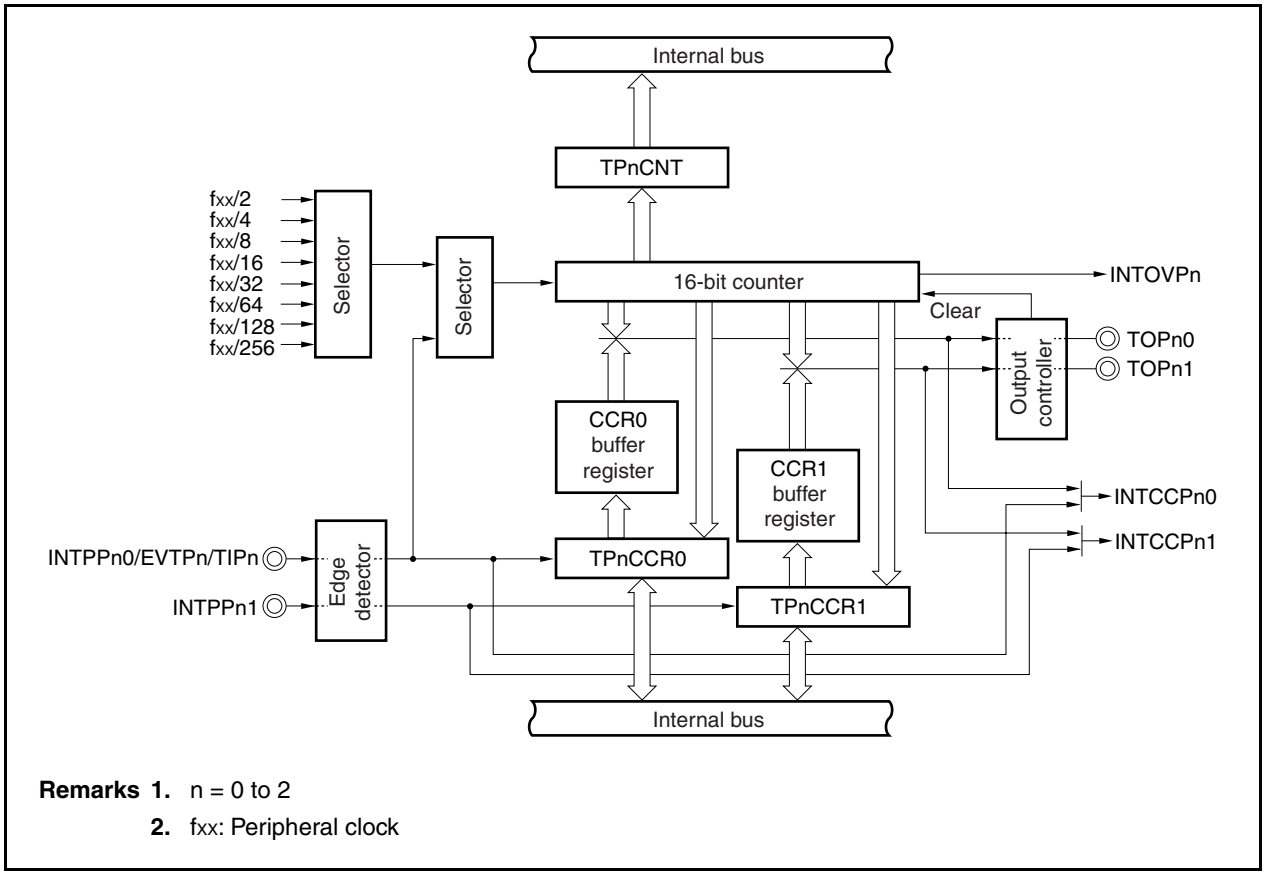
Table 8-1. Configuration of TMPn

Item	Configuration
Timer register	16-bit counter
Registers	TMPn capture/compare registers 0, 1 (TPnCCR0, TPnCCR1) TMPn counter read buffer register (TPnCNT) CCR0 and CCR1 buffer registers
Timer input	Total 12 (INTPP00, INTPP01, INTPP10, INTPP11, INTPP20, INTPP21, EVTP0 to EVTP2, TIP0 to TIP2 pins)[Note]
Timer output	Total 6 (TOP00, TOP01, TOP10, TOP11, TOP20, TOP21 pins)[Note]
Control registers	TMPn control registers 0, 1 (TPnCTL0, TPnCTL1) TMPn I/O control registers 0 to 2 (TPnIOC0 to TPnIOC2) TMPn option register 0 (TPnOPT0)

Note INTPPn0/EVTPn/TIPn pin functions alternately as a capture trigger input pin (INTPPn0), external event count input pin (EVTPn), external trigger input pin (TIPn), and timer output pin (TOPn0). INTPPn1 pin functions alternately as a capture trigger input pin (INTPPn1) and timer output pin (TOPn1).

Remark n = 0 to 2

Figure 8-1. TMPn Block Diagram



Remarks 1. n = 0 to 2

2. 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 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.

Reset input clears the TPnCE bit to 0.

(2) CCR0 buffer register

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

When the 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 (INTCCPn0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is cleared to 0000H after reset, and the TPnCCR0 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 (INTCCPn1) is generated.

The CCR1 buffer register cannot be read or written directly.

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

(4) Edge detector

This circuit detects the valid edges input to the INTPPn0/EVTPn/TIPn and INTPPn1 pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TPnIOC1 and TPnIOC2 registers.

(5) Output controller

This circuit controls the output of the TOPn0 and TOPn1 pins. The output controller is controlled by the TPnIOC0 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.

8.4 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 input clears 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

TPnCTL0 (n = 0 to 2)

<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	$f_{XX}/2$
0	0	1	$f_{XX}/4$
0	1	0	$f_{XX}/8$
0	1	1	$f_{XX}/16$
1	0	0	$f_{XX}/32$
1	0	1	$f_{XX}/64$
1	1	0	$f_{XX}/128$
1	1	1	$f_{XX}/256$

<R> **Note** TPnOPT0.TPnOVF bit and 16-bit counter are reset simultaneously. Moreover, timer outputs (TOPn0 and TOPn1 pins) are reset to the TPnIOC0 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 set 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 input clears this register to 00H.

(1/2)

After reset: 00H R/W Address: TP0CTL1 FFFFF641H, TP1CTL1 FFFFF661H, TP2CTL1 FFFFF681H

	7	6	5	4	3	2	1	0
TPnCTL1 (n = 0 to 2)	TP2SYE[Note]	TPnEST	TPnEEE	0	0	TPnMD2	TPnMD1	TPnMD0

TP2SYE[Note]	Operation mode selection
0	TMP2 single mode
1	Tuning operation mode (see **12.4.5**)
TMP2 can be used only as an A/D conversion start trigger factor of an A/D converter during the tuning operation. In the tuning operation mode, this bit always operates in synchronization with TMQ0.	

TPnEST	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 TPnEST bit as the trigger. • In external trigger pulse output mode: A PWM waveform is output with writing 1 to the TPnEST bit as the trigger.
The read value of the TPnEST bit is always 0.	

TPnEEE	Count clock selection
0	Disable operation with external event count input (EVTPn pin). (Perform counting with the count clock selected by the TPnCTL0.TPnCKS0 to TPnCTL0.TPnCKS2 bits.)
1	Enable operation with external event count input (EVTPn pin). (Perform counting at the valid edge of the external event count input signal (EVTPn pin).)
The TPnEEE bit selects whether counting is performed with the internal count clock or the valid edge of the external event count input.	

Note This bit can only be set in TMP2. Be sure to clear bit 7 of TMP0 and TMP1 to 0. For details of tuning operation mode, see **CHAPTER 12 MOTOR CONTROL FUNCTION**.

(2/2)

TPnMD2	TPnMD1	TPnMD0	Timer mode selection
0	0	0	Interval timer mode
0	0	1	External event count mode
0	1	0	External trigger pulse output mode
0	1	1	One-shot pulse output mode
1	0	0	PWM output mode
1	0	1	Free-running timer mode
1	1	0	Pulse width measurement mode
1	1	1	Setting prohibited

Cautions 1. The TPnEST 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 TPnEEE bit.

<R> **3. Set the TP2SYE, TPnEEE, 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 set bits 3 and 4 to "0".

(3) TMPn I/O control register 0 (TPnIOC0)

The TPnIOC0 register is an 8-bit register that controls the timer output (TOPn0, TOPn1 pins).

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: TP0IOC0 FFFFF642H, TP1IOC0 FFFFF662H
TP2IOC0 FFFFF682H

	7	6	5	4	3	<2>	1	<0>
TPnIOC0	0	0	0	0	TPnOL1	TPnOE1	TPnOL0	TPnOE0

(n = 0 to 2)

TPnOL1	TOPn1 pin output level setting[Note]
0	TOPn1 pin starts output at high level.
1	TOPn1 pin starts output at low level.

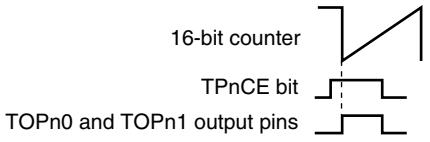
TPnOE1	TOPn1 pin output setting
0	Timer output prohibited • Low level is output from the TOPn1 pin when the TPnOL1 bit = 0. • High level is output from the TOPn1 pin when the TPnOL1 bit = 1.
1	Timer output enabled (A pulse is output from the TOPn1 pin.)

TPnOL0	TOPn0 pin output level setting[Note]
0	TOPn0 pin starts output at high level.
1	TOPn0 pin starts output at low level.

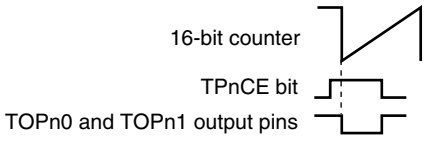
TPnOE0	TOPn0 pin output setting
0	Timer output prohibited • Low level is output from the TOPn0 pin when the TPnOL0 bit = 0. • High level is output from the TOPn0 pin when the TPnOL0 bit = 1.
1	Timer output enabled (A pulse is output from the TOPn0 pin.)

Note The output level of the timer output pins (TOPn0 and TOPn1) specified by the TPnOLa bit is shown below (a = 0, 1).

• When TPnOLa bit = 0



• When TPnOLa bit = 1



<R> **Cautions 1. If the setting of the TPnIOC0 register is changed when TOPn0 and TOPn1 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 TPnOL1, TPnOE1, TPnOL0, and TPnOE0 bits when the TPnCTL0.TPnCE bit = 0. (The same value can be written when the TPnCE bit = 1.) If rewriting was mistakenly performed, clear the TPnCE bit to 0 and then set the bits again.

3. Even if the TPnOL0 or TPnOL1 bit is manipulated when the TPnCE, TPnOE0, and TPnOE1 bits are 0, the output level of the TOPn0 and TOPn1 pins changes.

(4) TMPn I/O control register 1 (TPnIOC1)

The TPnIOC1 register is an 8-bit register that controls the valid edge for the capture trigger input signals (INTPPn0, INTPPn1 pins).

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: TP0IOC1 FFFFF643H, TP1IOC1 FFFFF663H, TP2IOC1 FFFFF683H

	7	6	5	4	3	2	1	0
TPnIOC1	0	0	0	0	TPnIS3	TPnIS2	TPnIS1	TPnIS0

(n = 0 to 2)

TPnIS3	TPnIS2	Capture trigger input signal (INTPPn1 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

TPnIS1	TPnIS0	Capture trigger input signal (INTPPn0 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 TPnIS3 to TPnIS0 bits when the TPnCTL0.TPnCE bit = 0. (The same value can be written when the TPnCE bit = 1.) If rewriting was mistakenly performed, clear the TPnCE bit to 0 and then set the bits again.

<R> **2. The TPnIS3 to TPnIS0 bits are valid only in the free-running timer mode (only when the TPnOPT0.TPnCCS1 and TPnOPT0. TPnCCS0 bits = 11) and the pulse width measurement mode. In all other modes, a capture operation is not possible.**

(5) TMPn I/O control register 2 (TPnIOC2)

The TPnIOC2 register is an 8-bit register that controls the valid edge for the external event count input signal (EVTPn pin) and external trigger input signal (TIPn pin).

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: TP0IOC2 FFFFF644H, TP1IOC2 FFFFF664H, TP2IOC2 FFFFF684H

	7	6	5	4	3	2	1	0
TPnIOC2	0	0	0	0	TPnEES1	TPnEES0	TPnETS1	TPnETS0

(n = 0 to 2)

TPnEES1	TPnEES0	External event count input signal (EVTPn 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

TPnETS1	TPnETS0	External trigger input signal (TIPn 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 TPnEES1, TPnEES0, TPnETS1, and TPnETS0 bits when the TPnCTL0.TPnCE bit = 0. (The same value can be written when the TPnCE bit = 1.) If rewriting was mistakenly performed, clear the TPnCE bit to 0 and then set the bits again.

2. The TPnEES1 and TPnEES0 bits are valid only when the TPnCTL1.TPnEEE bit = 1 or when the external event count mode (the TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 bits = 001) has been set.

3. The TPnETS1 and TPnETS0 bits are valid only in the external trigger pulse 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 input clears this register to 00H.

After reset: 00H R/W Address: TP0OPT0 FFFFF645H, TP1OPT0 FFFFF665H, TP2OPT0 FFFFF685H

	7	6	5	4	3	2	1	<0>
TPnOPT0 (n = 0 to 2)	0	0	TPnCCS1	TPnCCS0	0	0	0	TPnOVF

<R>

TPnCCS1	TPnCCR1 register capture/compare selection
0	Compare register selected
1	Capture register selected (cleared by TPnCTL0.TPnCE bit = 0)
The TPnCCS1 bit setting is valid only in the free-running timer mode.	

<R>

TPnCCS0	TPnCCR0 register capture/compare selection
0	Compare register selected
1	Capture register selected (cleared by TPnCTL0.TPnCE bit = 0)
The TPnCCS0 bit setting is valid only in the free-running timer mode.	

TPnOVF	TMPn overflow 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 (INTOVPn) is generated at the same time that the TPnOVF bit is set to 1. The INTOVPn 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 are read when the TPnOVF bit = 1.
- Before clearing the TPnOVF bit to 0 after generation of the INTOVPn 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.

Cautions 1. Rewrite the TPnCCS1 and TPnCCS0 bits when the TPnCE bit = 0. (The same value can be written when the TPnCE bit = 1.) If rewriting was mistakenly performed, clear the TPnCE bit to 0 and then set the bits again.

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

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

The TPnCCR0 register is a 16-bit register that can be used as a capture register or compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TPnOPT0.TPnCCS0 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 input clears this register to 0000H.

After reset: 0000H R/W Address: TP0CCR0 FFFFF646H, TP1CCR0 FFFFF666H, TP2CCR0 FFFFF686H

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

(n = 0 to 2)

(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 (INTCCPn0) is generated. If TOPn0 pin output is enabled at this time, the output of the TOPn0 pin is inverted.

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.

<R> The compare register is not cleared by setting the TPnCTL0.TPnCE bit to 0.

(b) Function as capture register

When the TPnCCR0 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 TPnCCR0 register if the valid edge of the capture trigger input pin (INTPPn0 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TPnCCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (INTPPn0) is detected.

Even if the capture operation and reading the TPnCCR0 register conflict, the correct value of the TPnCCR0 register can be read.

<R> The capture register is cleared by setting the TPnCTL0.TPnCE bit to 0.

Remark n = 0 to 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 8-2. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write[Note]
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write[Note]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	None

Note Writing to the TPnCCR1 register is the trigger.

Remark For anytime write and batch write, see **8.6 (2) Anytime write and batch write**.

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

The TPnCCR1 register is a 16-bit register that can be used as a capture register or compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TPnOPT0.TPnCCS1 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 input clears this register to 0000H.

After reset: 0000H R/W Address: TP0CCR1 FFFFF648H, TP1CCR1 FFFFF668H, TP2CCR1 FFFFF688H

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

(n = 0 to 2)

(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 (INTCCPn1) is generated. If TOPn1 pin output is enabled at this time, the output of the TOPn1 pin is inverted.

<R> The compare register is not cleared by setting the TPnCTL0.TPnCE bit to 0.

(b) Function as capture register

When the TPnCCR1 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 TPnCCR1 register if the valid edge of the capture trigger input pin (INTPPn1 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TPnCCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (INTPPn1 pin) is detected.

Even if the capture operation and reading the TPnCCR1 register conflict, the correct value of the TPnCCR1 register can be read.

<R> The capture register is cleared by setting the TPnCTL0.TPnCE bit to 0.

Remark n = 0 to 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 8-3. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write[Note]
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write[Note]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	None

Note Writing to the TPnCCR1 register is the trigger.

Remark For anytime write and batch write, see **8.6 (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 timer 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 TPnCNT register is cleared to 0000H after reset, and the TPnCE bit is cleared to 0.

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

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

(n = 0 to 2)

8.5 Timer Output Operations

The following table shows the operations and output levels of the TOPn0 and TOPn1 pins.

Table 8-4. Timer Output Control in Each Mode

Operation Mode	TOPn1 Pin	TOPn0 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 n = 0 to 2

Table 8-5. Truth Table of TOPn0 and TOPn1 Pins Under Control of Timer Output Control Bits

TPnIOC0.TPnOLa Bit	TPnIOC0.TPnOEa Bit	TPnCTL0.TPnCE Bit	Level of TOPna Pin
0	0	×	Low-level output
	1	0	Low-level output
		1	Low level immediately before counting, high level after counting is started
1	0	×	High-level output
	1	0	High-level output
		1	High level immediately before counting, low level after counting is started

Remark n = 0 to 2
a = 0, 1

8.6 Operation

The functions of TMPn are shown below.

Operation	TPnCTL1.TPnEST Bit (Software Trigger Bit)	TIPn Pin (External Trigger Input)	Capture/Compare Register Setting	Compare Register Write Method
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode	Invalid	Invalid	Compare only	Anytime write
External trigger pulse output mode[Note]	Valid	Valid	Compare only	Batch write
One-shot pulse output mode[Note]	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]	Invalid	Invalid	Capture only	Not applicable

Note 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 TPnCTL1.TPnEEE bit to 0).

Remark n = 0 to 2

(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

(a) Counter start operation

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 INTCCPn0 and INTCCPn1 interrupt signals are not generated.

(c) Overflow operation

The 16-bit counter overflows when the counter counts up from FFFFH to 0000H in the free-running mode or pulse width measurement mode. If the counter overflows, the TPnOPT0.TPnOVF bit is set to 1 and an interrupt request signal (INTOVPn) is generated. Note that the INTOVPn 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 (INTOVPn) 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. When the TPnCTL0.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.

- INTCCPn0 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.
- INTCCPn1 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.
- INTOVPn interrupt: This signal functions as an overflow interrupt request signal.

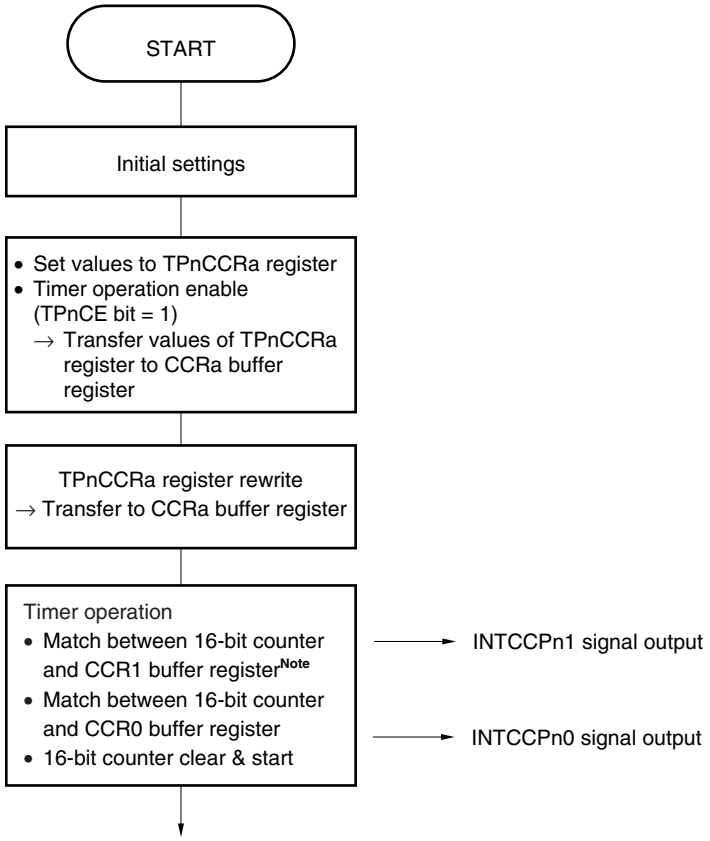
(2) Anytime write and batch write

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 2).

Figure 8-2. Flowchart of Basic Operation for Anytime Write

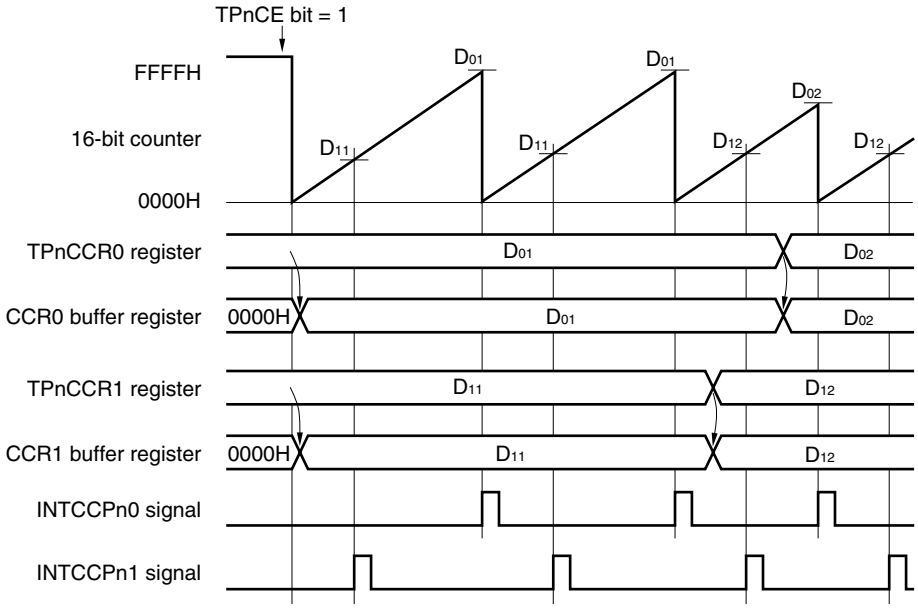


<R> **Note** The 16-bit counter is not cleared upon a match between the 16-bit counter value and the CCR1 buffer register value. It is cleared upon a match between the 16-bit counter value and the CCR0 buffer register value.

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

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

Figure 8-3. 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 2

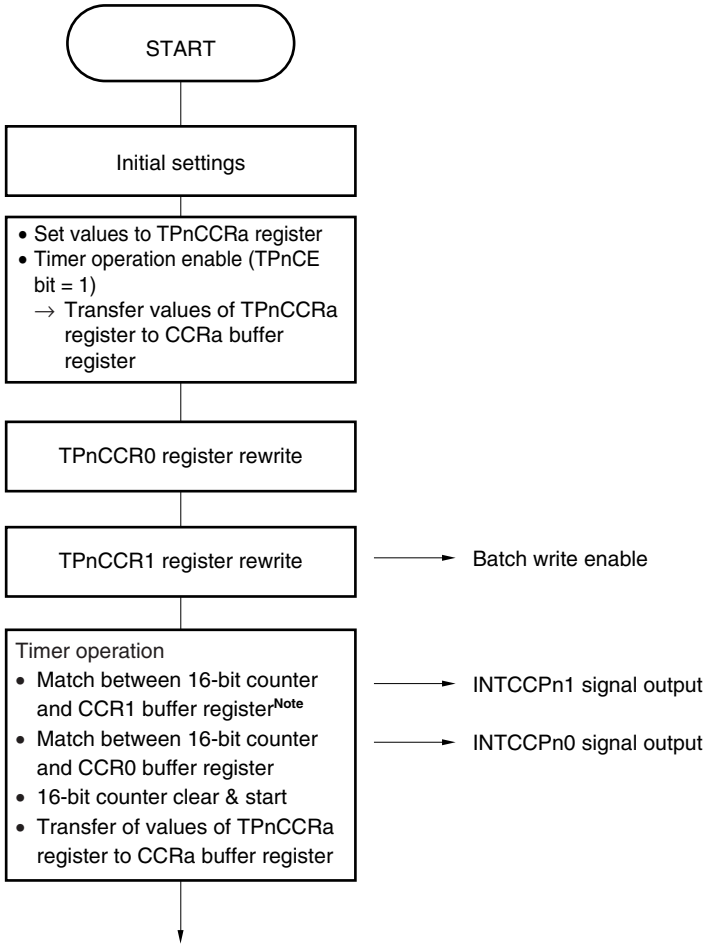
(b) Batch write

In this mode, data is transferred all at once from the TPnCCR0 and TPnCCR1 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 TPnCCR1 register.

Whether to enable or disable the next transfer timing is controlled by writing or not writing to the TPnCCR1 register.

In order for the setting value when the TPnCCR0 and TPnCCR1 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 TPnCCR0 register and then write to the TPnCCR1 register before the 16-bit counter value and the CCR0 buffer register value match. Therefore, the values of the TPnCCR0 and TPnCCR1 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 TPnCCR0 register, also write the same value (same as preset value of the TPnCCR1 register) to the TPnCCR1 register.

Figure 8-4. Flowchart of Basic Operation for Batch Write



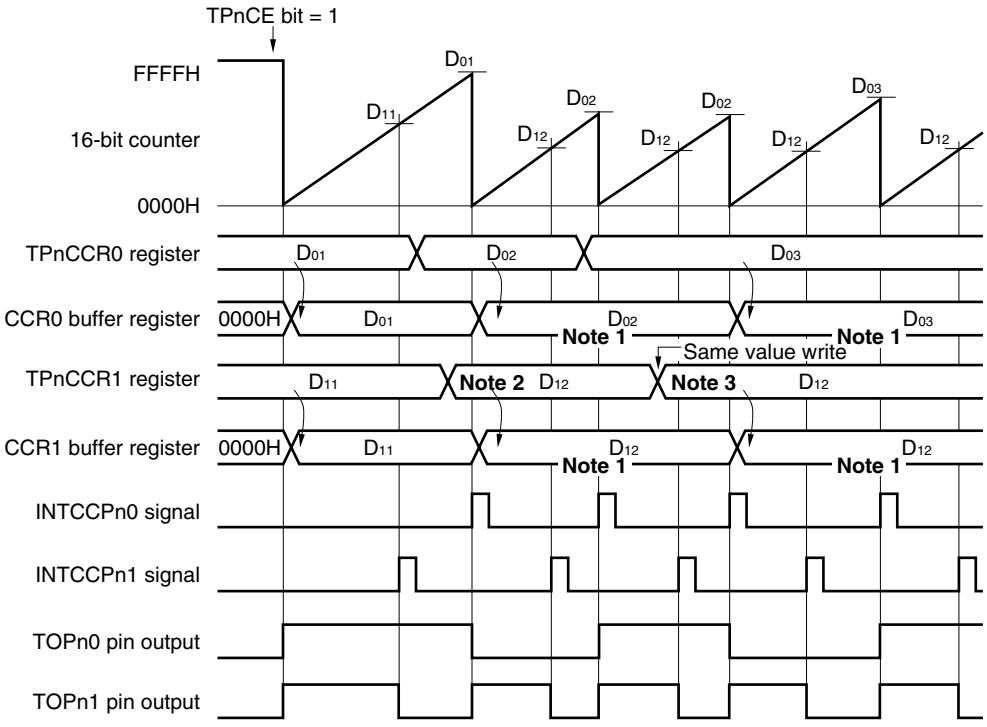
<R> **Note** The 16-bit counter is not cleared upon a match between the 16-bit counter value and the CCR1 buffer register value. It is cleared upon a match between the 16-bit counter value and the CCR0 buffer register value.

Caution Writing to the TPnCCR1 register includes enabling of batch write. Thus, rewrite the TPnCCR1 register after rewriting the TPnCCR0 register.

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

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

Figure 8-5. Timing of Batch Write



Notes 1. Because the TPnCCR1 register was not rewritten, D_{03} is not transferred.

2. Because the TPnCCR1 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 TPnCCR0 register (D_{01}).

3. Because the TPnCCR1 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 TPnCCR0 register (D_{02}).

Remarks 1. D_{01}, D_{02}, D_{03}: Setting values of TPnCCR0 register
D_{11}, D_{12}: Setting values of TPnCCR1 register

2. The above flowchart illustrates the operation in the PWM output mode as an example.

3. n = 0 to 2

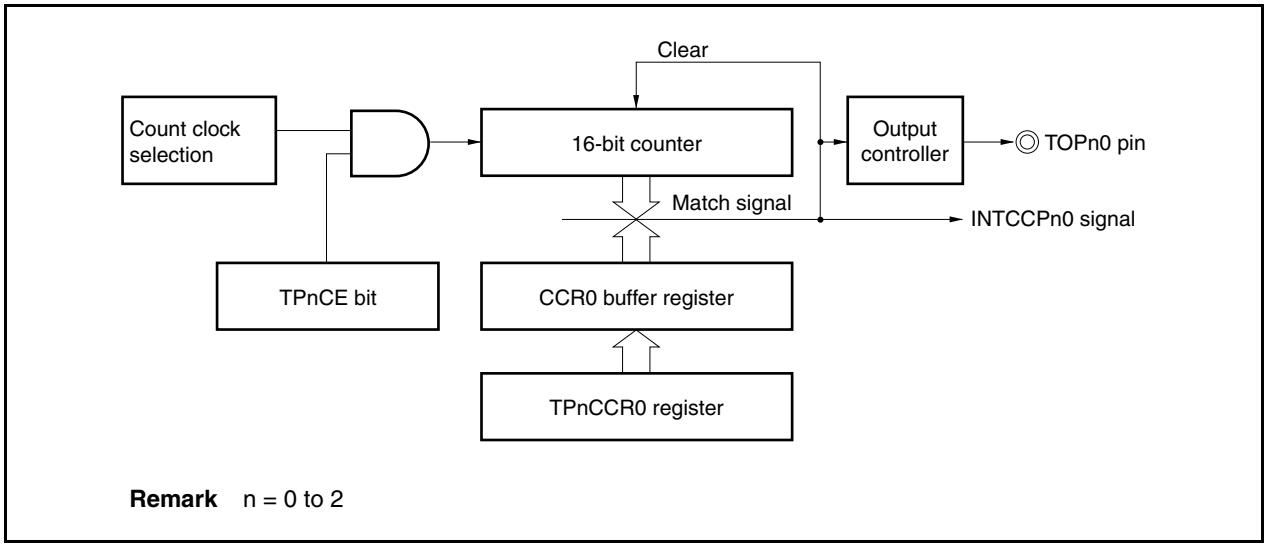
8.6.1 Interval timer mode (TPnMD2 to TPnMD0 bits = 000)

In the interval timer mode, an interrupt request signal (INTCCPn0) 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 TOPn0 pin.

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 (INTCCPn1) is generated. In addition, a PWM waveform with a duty factor of 50%, which is inverted when the INTCCPn1 signal is generated, can be output from the TOPn1 pin.

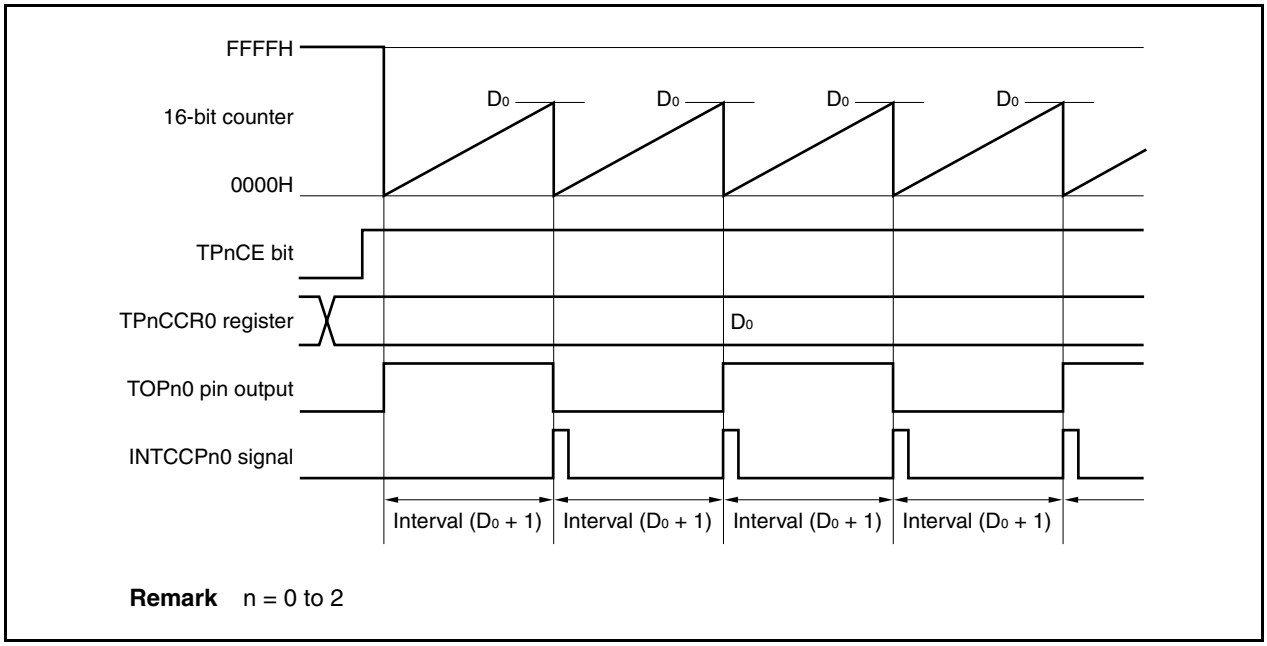
The value of the TPnCCR0 and TPnCCR1 registers can be rewritten even while the timer is operating.

Figure 8-6. Configuration of Interval Timer



Remark n = 0 to 2

Figure 8-7. Basic Timing of Operation in Interval Timer Mode



Remark n = 0 to 2

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 TOPn0 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 TOPn0 pin is inverted, and a compare match interrupt request signal (INTCCPn0) 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 2

Figure 8-8. Register Setting for Interval Timer Mode Operation (1/2)

(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)

	TP2SYE	TPnEST	TPnEEE			TPnMD2	TPnMD1	TPnMD0
<R> TPnCTL1	0	0	0/1[Note]	0	0	0	0	0

0, 0, 0:
Interval timer mode

0: Operate on count clock selected by TPnCKS0 to TPnCKS2 bits
1: Count with external event count input signal

<R> **Note** The TPnEEE bit can be set to 1 only when timer output (TOPn1) is used. However, set the TPnCCR0 and TPnCCR1 registers to the same value.

Figure 8-8. Register Setting for Interval Timer Mode Operation (2/2)

(c) TMPn I/O control register 0 (TPnIOC0)

					TPnOL1	TPnOE1	TPnOL0	TPnOE0
TPnIOC0	0	0	0	0	0/1	0/1	0/1	0/1

0: Disable TOPn0 pin output
1: Enable TOPn0 pin output

Setting of output level with operation of TOPn0 pin disabled
0: Low level
1: High level

0: Disable TOPn1 pin output
1: Enable TOPn1 pin output

Setting of output level with operation of TOPn1 pin disabled
0: Low level
1: High level

<R> **(d) TMPn I/O control register 2 (TPnIOC2)**

					TPnEES1	TPnEES0	TPnETS1	TPnETS0
TPnIOC2	0	0	0	0	0/1 Note	0/1 Note	0	0

Select valid edge of external event count input (EVTPn pin).

Note The TPnEES1 and TPnEES0 bits can be set only when timer output (TOPn1) is used. However, set the TPnCCR0 and TPnCCR1 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

(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 TOPn1 pin output is inverted and a compare match interrupt request signal (INTCCPn1) is generated.

By setting this register to the same value as the value set in the TPnCCR0 register, a PWM waveform with a duty factor of 50% can be output from the TOPn1 pin.

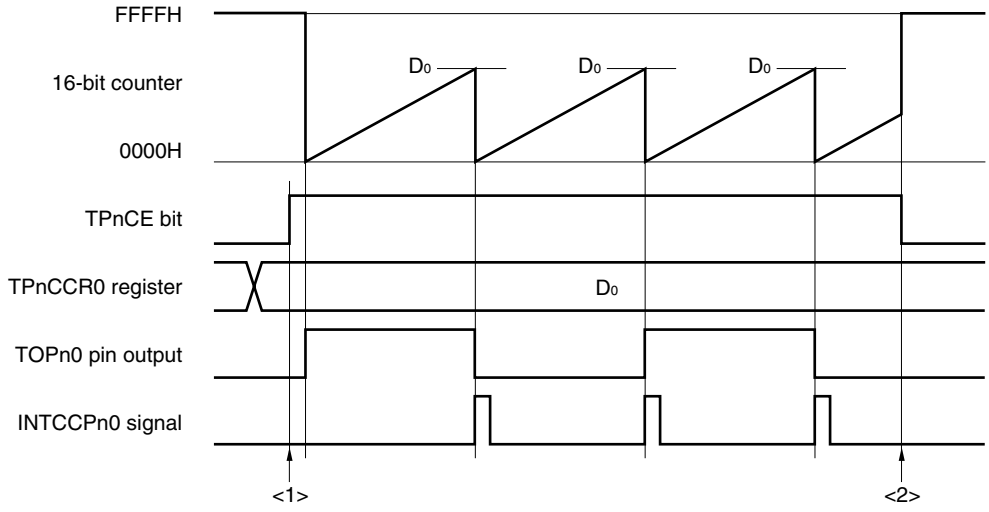
When the TP0CCR1 to TOP2CCR1 registers are not used, it is recommended to set their value to FFFFH. Also mask the register by the interrupt mask flag (P00IC1.P00MK1, P02IC2.P02MK2, P05IC1.P05MK1).

<R> **Remarks 1.** TMPn I/O control register 1 (TPnIOC1) and TMPn option register 0 (TPnOPT0) are not used in the interval timer mode.

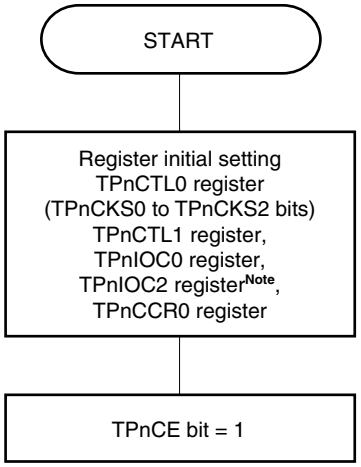
2. n = 0 to 2

(1) Interval timer mode operation flow

Figure 8-9. Software Processing Flow in Interval Timer Mode



<1> Count operation start flow

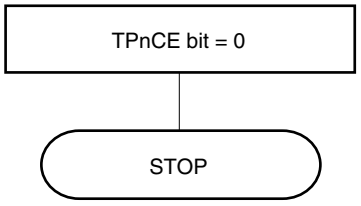


Initial setting of these registers is performed before setting the TPnCE bit to 1.

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

<R> **Note** The TPnEES1 and TPnEES0 bits can be set only when timer output (TOPn1) is used. However, set the TPnCCR0 and TPnCCR1 registers to the same value.

<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 TOPn0 pin is as specified by the TPnIOC0 register.

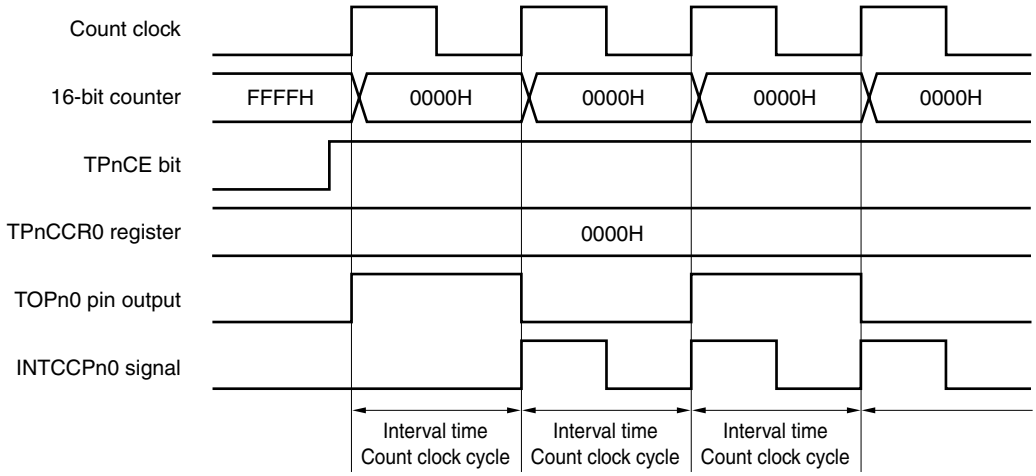
Remark n = 0 to 2

(2) Interval timer mode operation timing

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

If the TPnCCR0 register is set to 0000H, the INTCCPn0 signal is generated at each count clock, and the output of the TOPn0 pin is inverted.

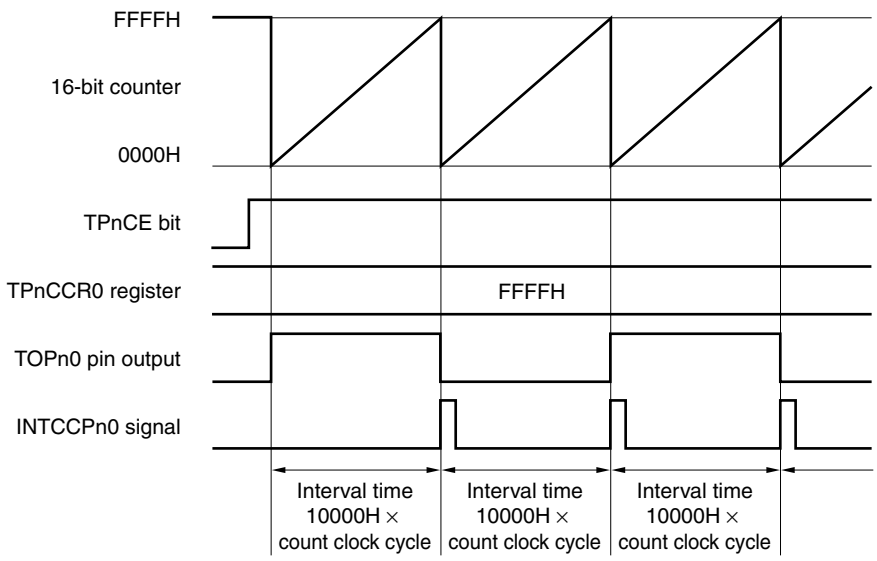
The value of the 16-bit counter is always 0000H.



Remark n = 0 to 2

(b) Operation if TPnCCR0 register is set to FFFFH

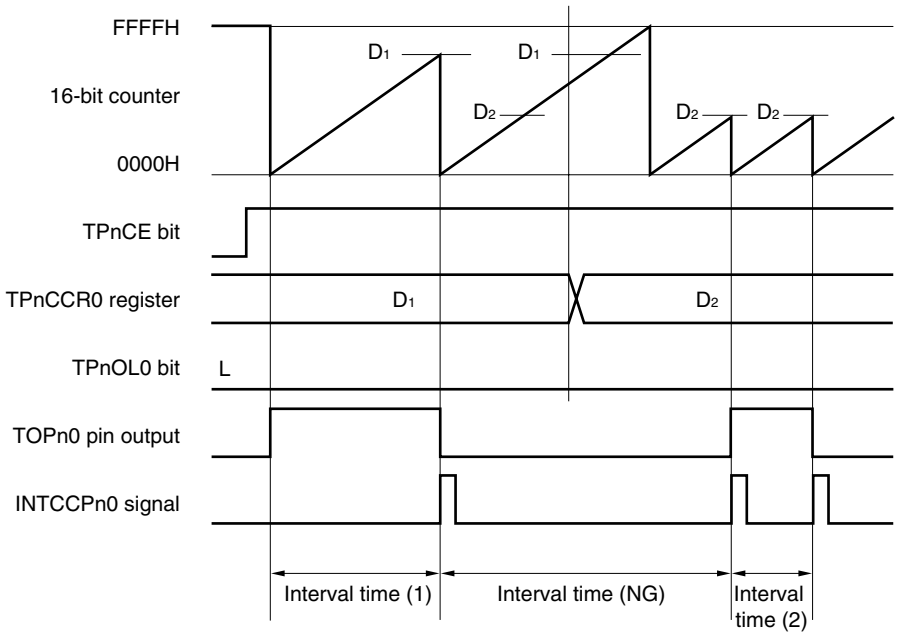
If the TPnCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTCCPn0 signal is generated and the output of the TOPn0 pin is inverted. At this time, an overflow interrupt request signal (INTOVPn) is not generated, nor is the overflow flag (TPnOPT0.TPnOVF bit) set to 1.



Remark n = 0 to 2

(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 2

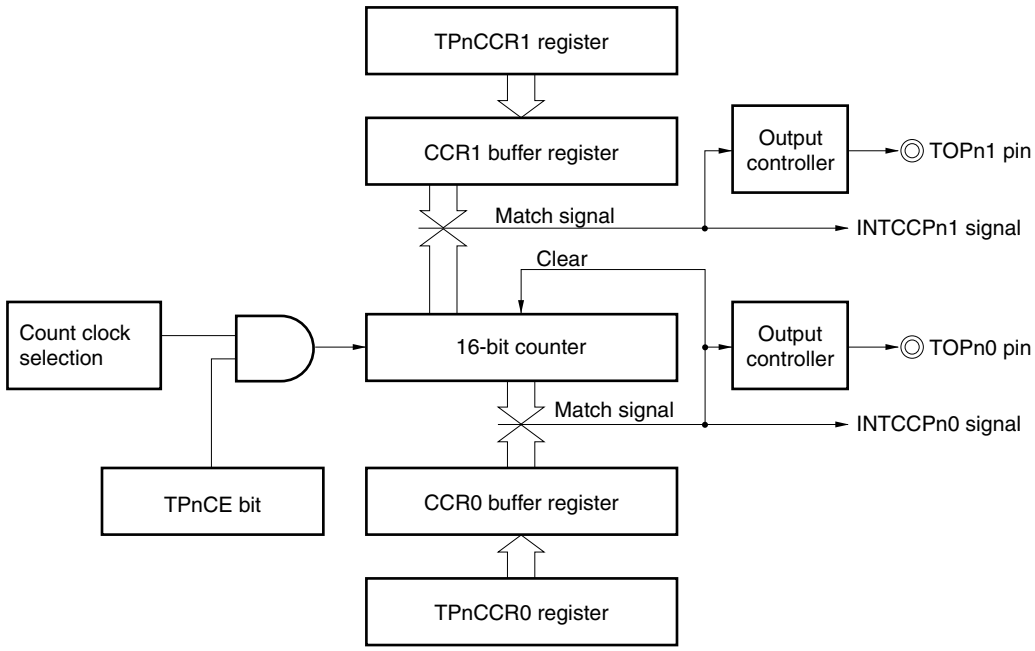
If the value of the TPnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TPnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is 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 INTCCPn0 signal is generated and the output of the TOPn0 pin is inverted.

Therefore, the INTCCPn0 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 8-10. Configuration of TPnCCR1 Register



Remark n = 0 to 2

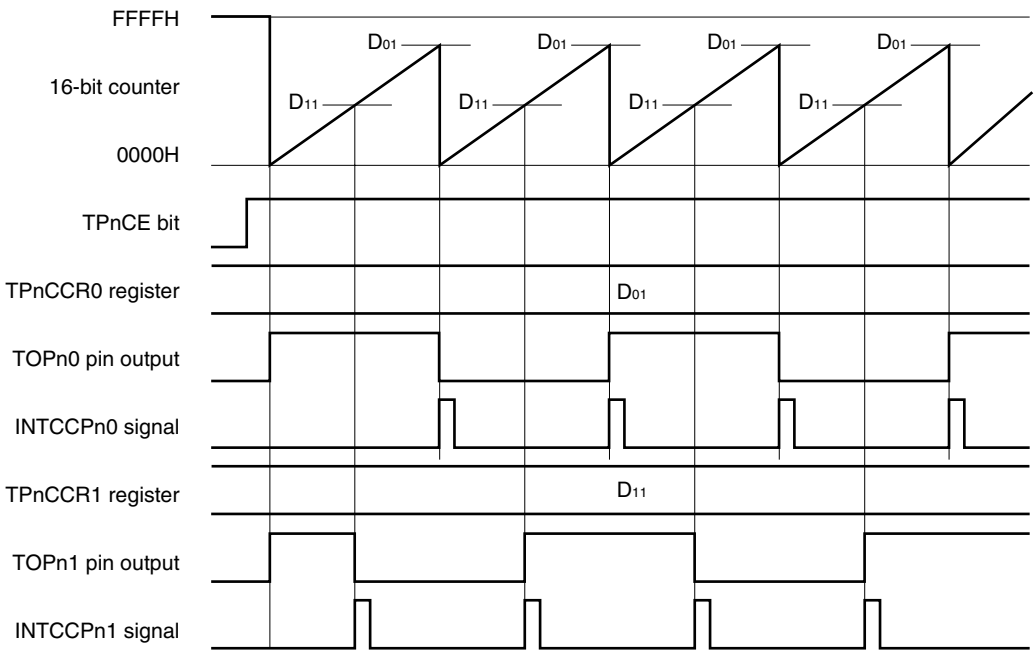
When the TPnCCR1 register is set to the same value as the TPnCCR0 register, the INTCCPn0 signal is generated at the same timing as the INTCCPn1 signal and the TOPn1 pin output is inverted. In other words, a PWM waveform with a duty factor of 50% can be output from the TOPn1 pin.

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 INTCCPn1 signal is generated once per cycle. At the same time, the output of the TOPn1 pin is inverted.

The TOPn1 pin outputs a PWM waveform with a duty factor of 50% after outputting a short-width pulse.

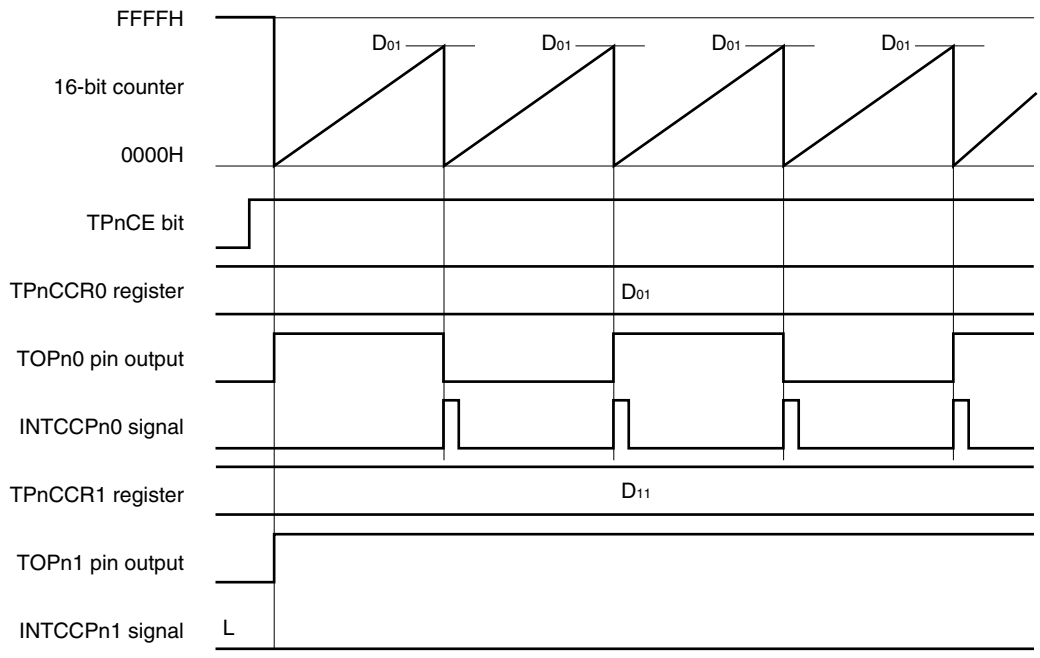
Figure 8-11. Timing Chart When $D_{01} \geq D_{11}$



Remark n = 0 to 2

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 INTCCPn1 signal is not generated, nor is the output of the TOPn1 pin changed.
When the TPnCCR1 register is not used, it is recommended to set its value to FFFFH.

Figure 8-12. Timing Chart When $D_{01} < D_{11}$



Remark n = 0 to 2

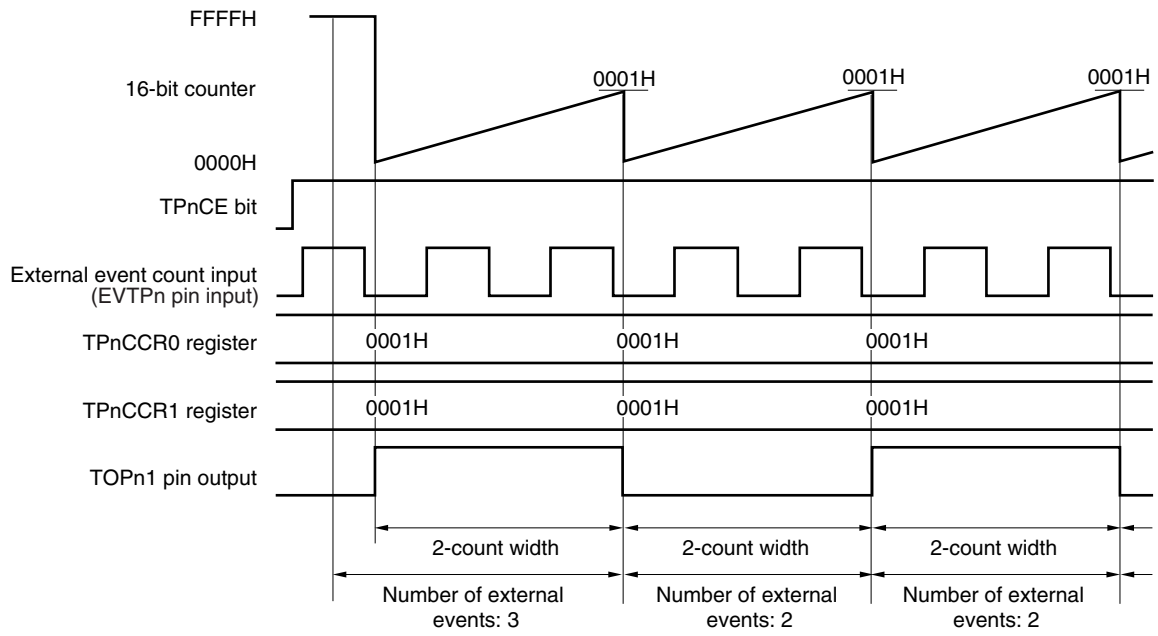
<R> **(3) Operation by external event count input (EVTPn)**

(a) Operation

To count the 16-bit counter at the valid edge of external event count input (EVTPn) in the interval timer mode, the valid edge of the external event count input is necessary once because the 16-bit counter is cleared from FFFFH to 0000H immediately after the TPnCE bit is set from 0 to 1.

When 0001H is set to both the TPnCCR0 and TPnCCR1 registers, the TOPn1 pin output is inverted each time the 16-bit counter counts twice.

The TPnCTL1.TPnEEE bit can be set to 1 in the interval timer mode only when the timer output (TOPn1) is used with the external event count input.



Remark n = 0 to 2

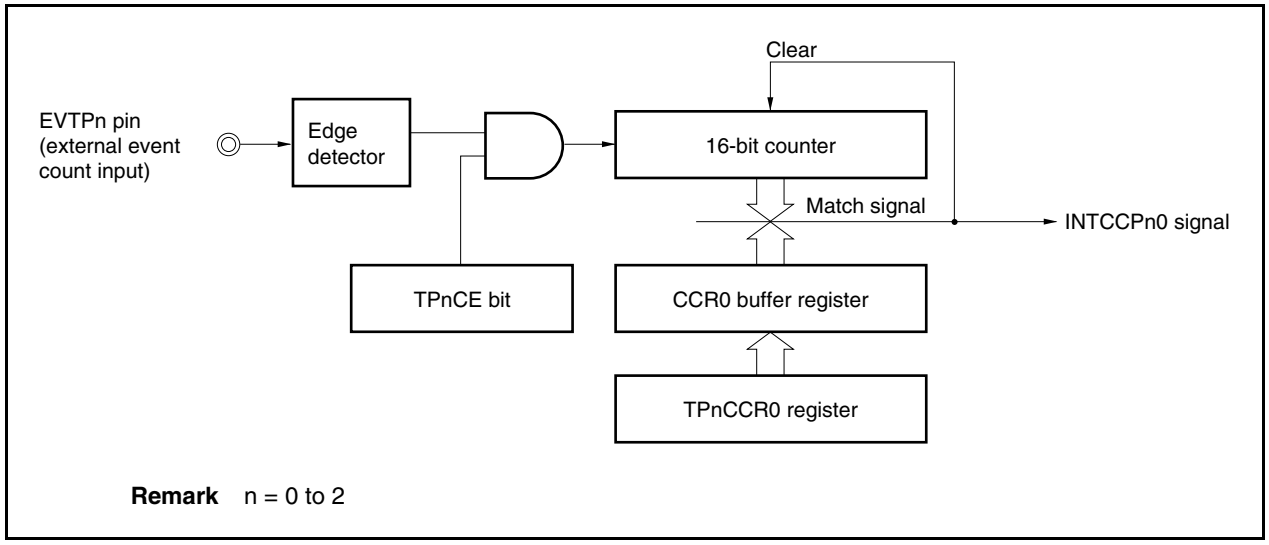
8.6.2 External event count mode (TPnMD2 to TPnMD0 bits = 001)

In the external event count mode, the valid edge of the external event count input (EVTPn) is counted when the TPnCTL0.TPnCE bit is set to 1, and an interrupt request signal (INTCCPn0) is generated each time the number of <R> edges set by the TPnCCR0 register have been counted. The TOPn0 and TOPn1 pins cannot be used. When using the TOPn1 pin for external event count input, set the TPnCTL1.TPnEEE bit to 1 in the interval timer mode (see **8.6.1 (3) Operation by external event count input (EVTPn)**).

The TPnCCR1 register is not used in the external event count mode.

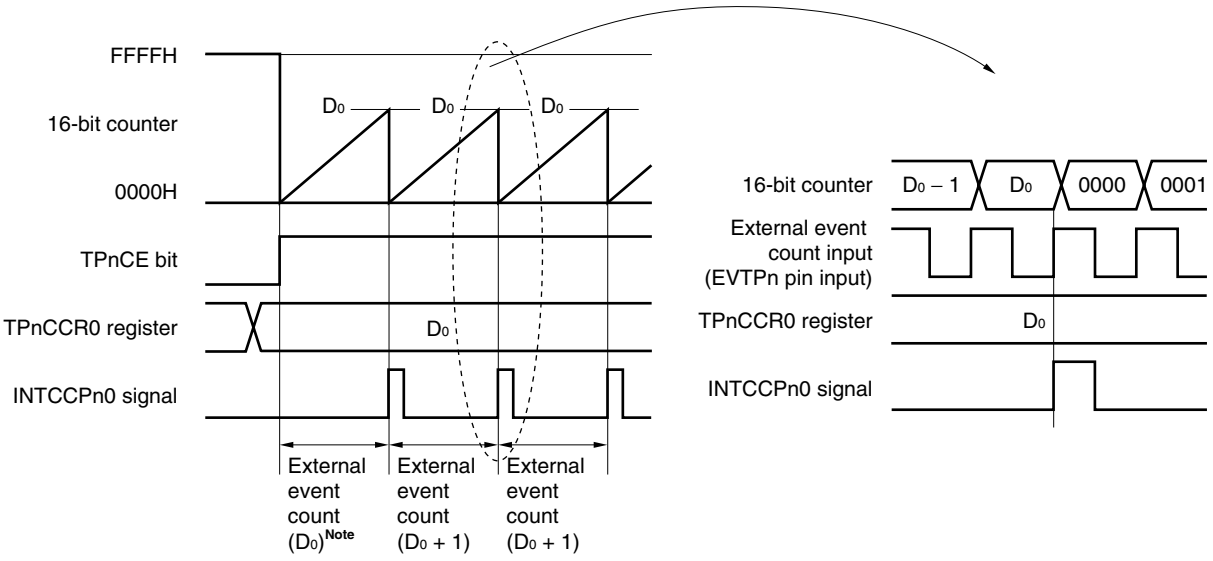
Caution In the external event count mode, the TPnCCR0 and TPnCCR1 registers must not be cleared to 0000H.

Figure 8-13. Configuration in External Event Count Mode



Remark n = 0 to 2

Figure 8-14. Basic Timing in External Event Count Mode



Note In the external event count mode, when the TPnCTL0.TPnCE 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 each 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 the second or subsequent count operation.

Caution This figure shows the basic timing when the rising edge is specified as the valid edge of the external event count input.

Remark n = 0 to 2

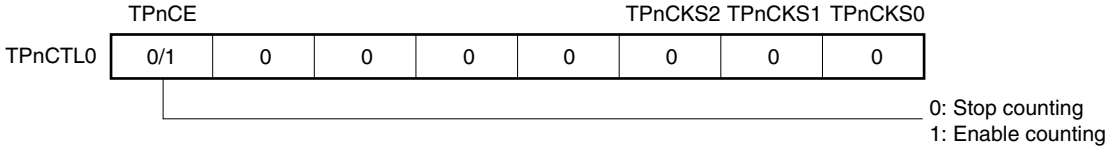
When the TPnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H. The counter counts each time the valid edge of external event count input is detected. Additionally, the set value of the TPnCCR0 register is transferred to the CCR0 buffer register.

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, and a compare match interrupt request signal (INTCCPn0) is generated.

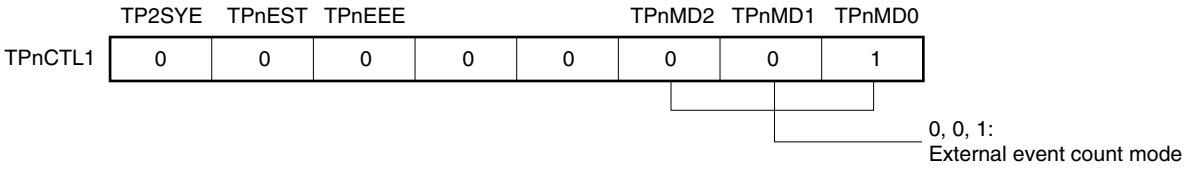
The INTCCPn0 signal is generated for the first time when the valid edge of the external event count input has been detected “value set to TPnCCR0 register” times. After that, the INTCCPn0 signal is generated each time the valid edge of the external event count has been detected “value set to TPnCCR0 register + 1” times.

Figure 8-15. Register Setting for Operation in External Event Count Mode

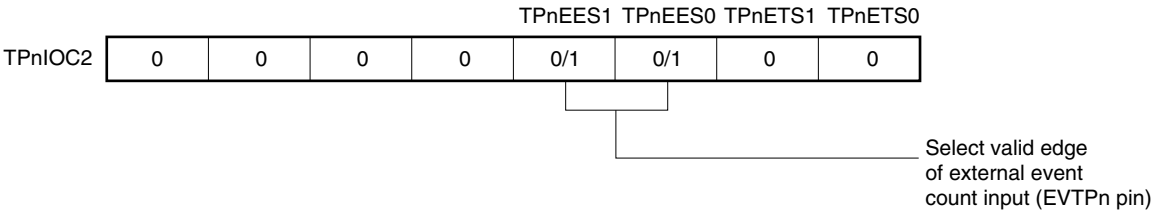
(a) TMPn control register 0 (TPnCTL0)



(b) TMPn control register 1 (TPnCTL1)



(c) TMPn I/O control register 2 (TPnIOC2)



(d) TMPn counter read buffer register (TPnCNT)

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

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

If the TPnCCR0 register is set to D_0, the count is cleared when the number of external events has reached (D_0) and the first compare match interrupt request signal (INTCCPn0) is generated. The second compare match interrupt request signal (INTCCPn0) is generated when the number of external events has reached (D_0 + 1).

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

The TPnCCR1 register is not used in the external event count 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, a compare match interrupt request signal (INTCCPn1) is generated.

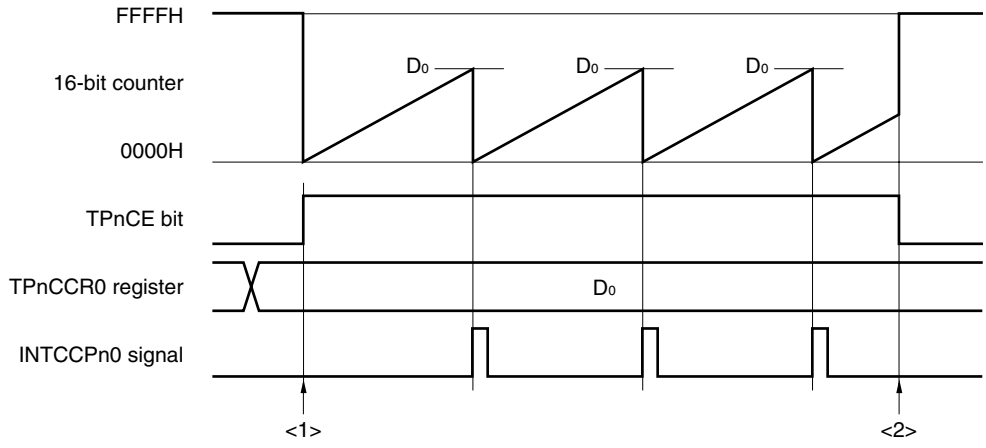
When the TP0CCR1 to TP2CCR1 registers are not used, it is recommended to set their value to FFFFH. Also mask the register by the interrupt mask flag (P00IC1.P00MK1, P02IC2.P02MK2, P05IC1.P05MK1).

Remarks 1. TMPn I/O control register 0 (TPnIOC0), TMPn I/O control register 1 (TPnIOC1), and TMPn option register 0 (TPnOPT0) are not used in the external event count mode.

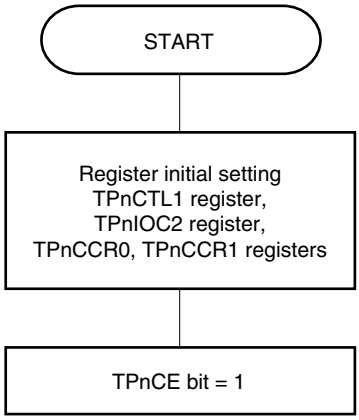
2. n = 0 to 2

(1) External event count mode operation flow

Figure 8-16. Software Processing Flow in External Event Count Mode

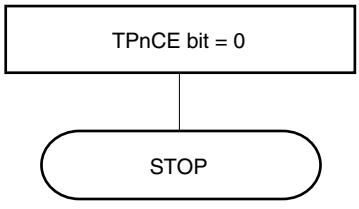


<1> Count operation start flow



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

<2> Count operation stop flow



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

Remark n = 0 to 2

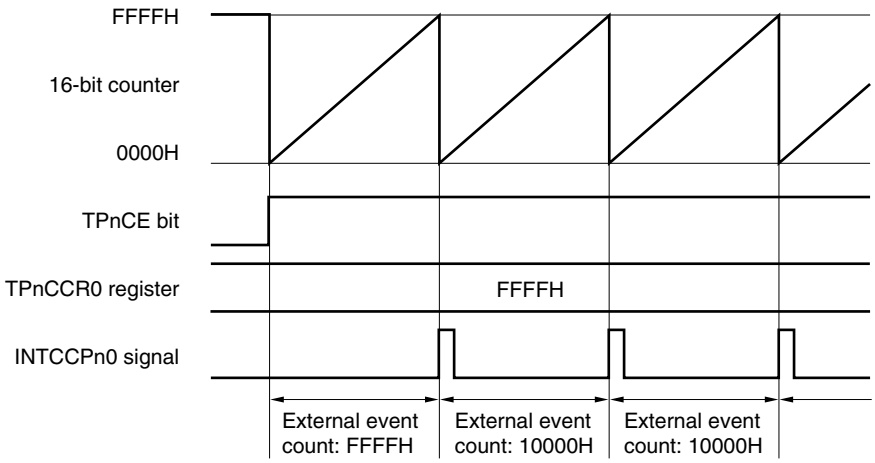
(2) Operation timing in external event count mode

Cautions 1. In the external event count mode, the TPnCCR0 and TPnCCR1 registers must not be cleared to 0000H.

<R> **2. In the external event count mode, use of the timer output (TOPn0, TOPn1) is disabled. If using timer output (TOPn1) with external event count input (EVTPn), set the interval timer mode, and select the operation enabled by the external event count input for the count clock (TPnCTL1.TPnEEE bit = 1) (see 8.6.1 (3) Operation by external event count input (EVTPn)).**

(a) Operation if TPnCCR0 register is set to FFFFH

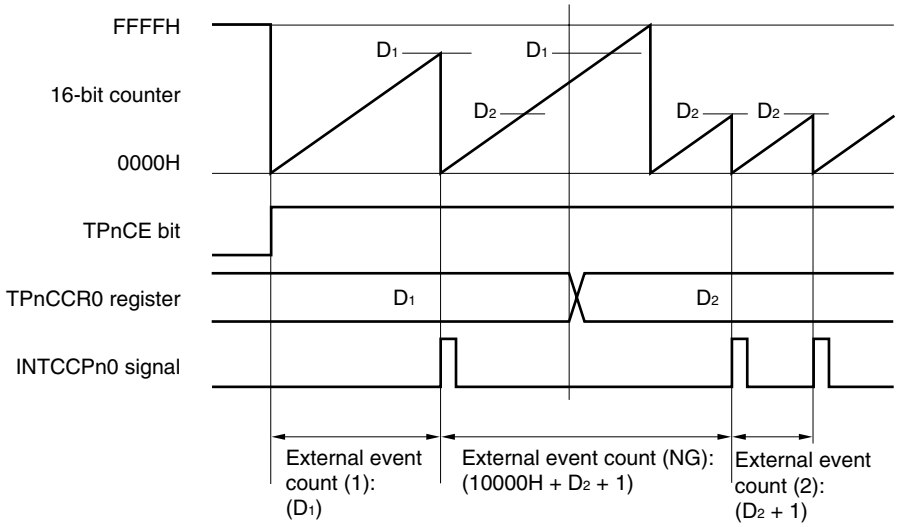
If the TPnCCR0 register is set to FFFFH, the 16-bit counter counts to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTCCPn0 signal is generated. At this time, the TPnOPT0.TPnOVF bit is not set.



Remark n = 0 to 2

(b) Notes on rewriting the TPnCCR0 register

If the value of the TPnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow. When the overflow may occur, stop counting once and then change the set value.



Remark n = 0 to 2

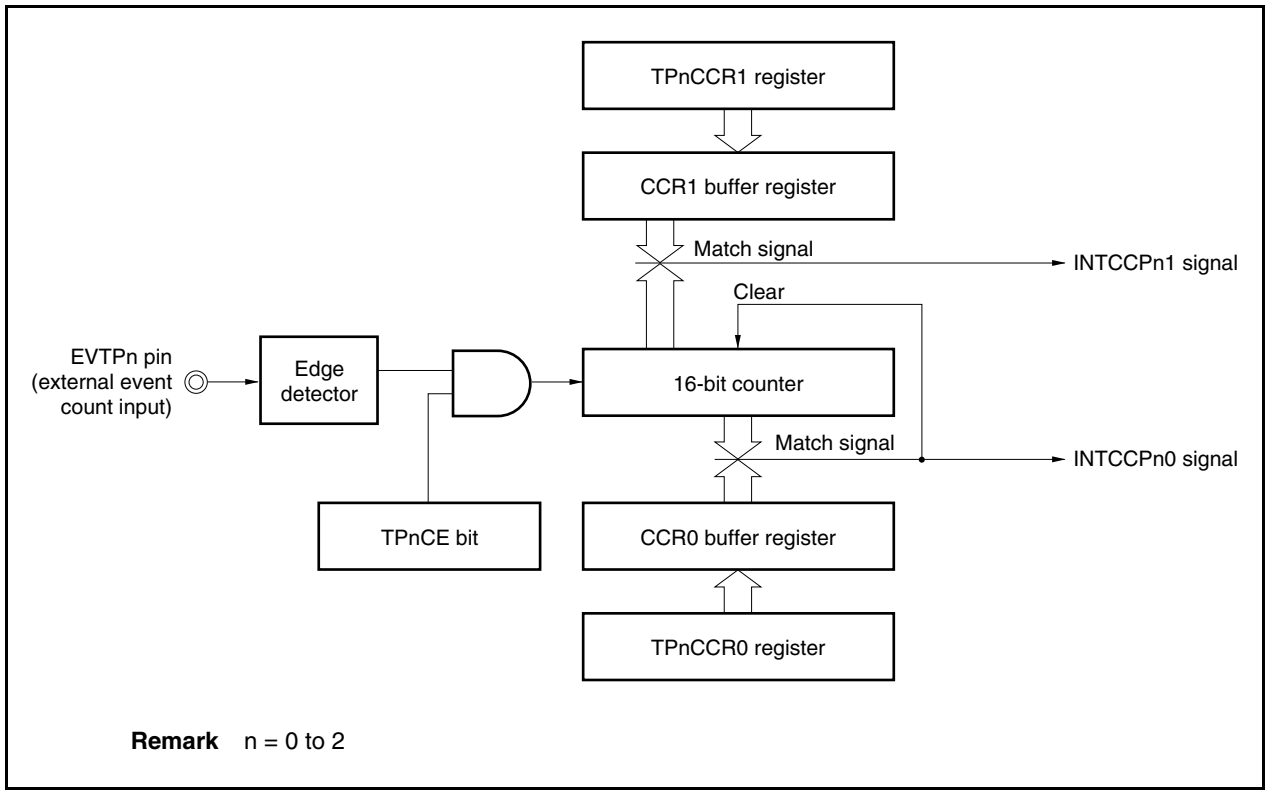
If the value of the TPnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TPnCCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter 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 INTCCPn0 signal is generated.

Therefore, the INTCCPn0 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 TPnCCR1 register

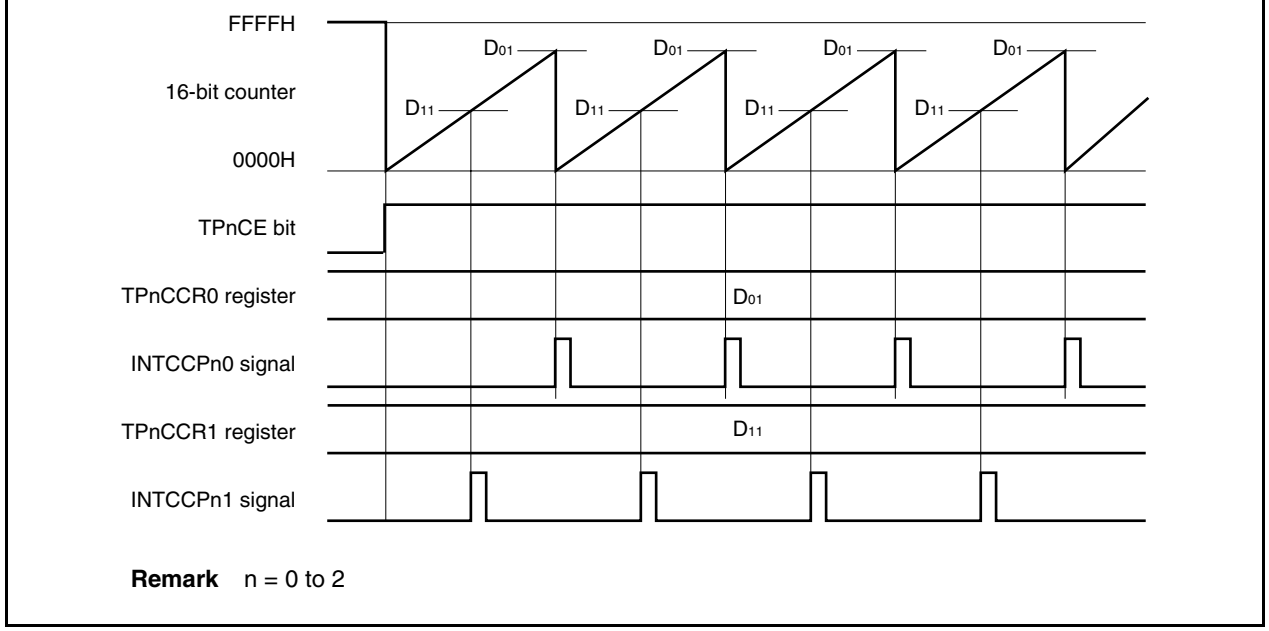
Figure 8-17. Configuration of TPnCCR1 Register



Remark n = 0 to 2

If the set value of the TPnCCR1 register is smaller than the set value of the TPnCCR0 register, the INTCCPn1 signal is generated once per cycle.

Figure 8-18. Timing Chart When $D_{01} \geq D_{11}$

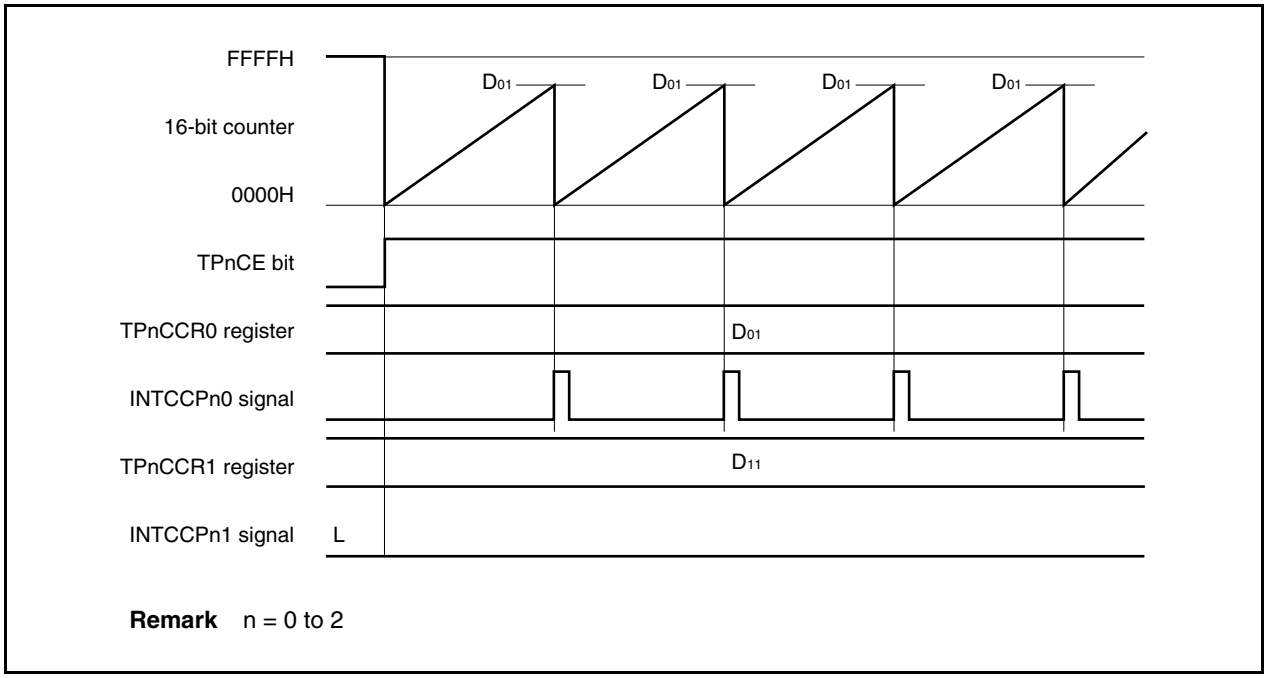


Remark n = 0 to 2

If the set value of the TPnCCR1 register is greater than the set value of the TPnCCR0 register, the INTCCPn1 signal is not generated because the count value of the 16-bit counter and the value of the TPnCCR1 register do not match.

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

Figure 8-19. Timing Chart When $D_{01} < D_{11}$



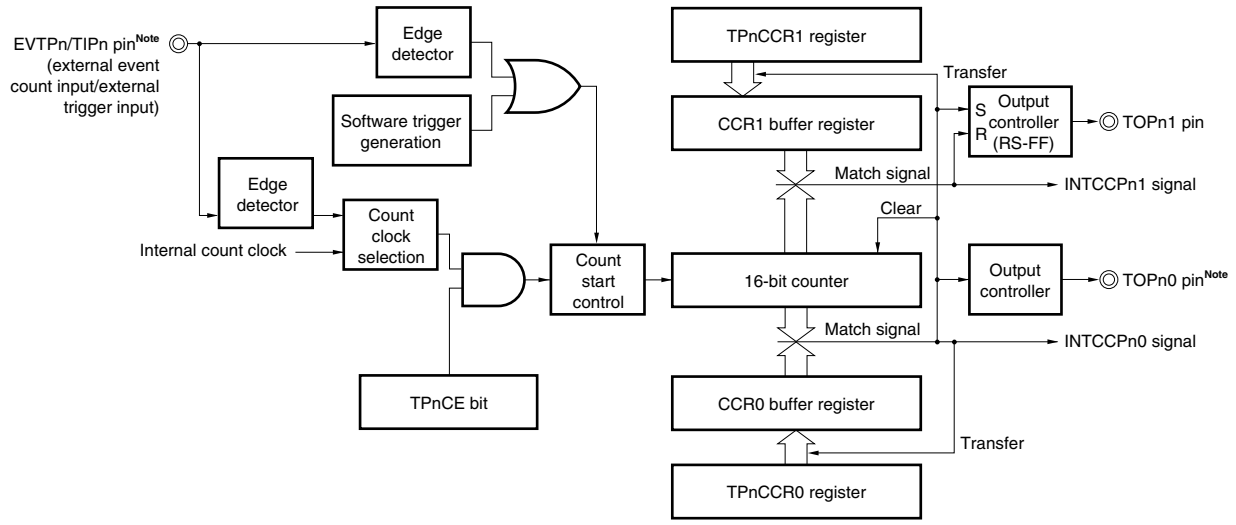
Remark n = 0 to 2

8.6.3 External trigger pulse output mode (TPnMD2 to TPnMD0 bits = 010)

In the external trigger pulse output mode, 16-bit timer/event counter P waits for a trigger when the TPnCTL0.TPnCE bit is set to 1. When the valid edge of an external trigger input (TIPn) is detected, 16-bit timer/event counter P starts counting, and outputs a PWM waveform from the TOPn1 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 TPnCCR0 register + 1 as half its cycle can also be output from the TOPn0 pin.

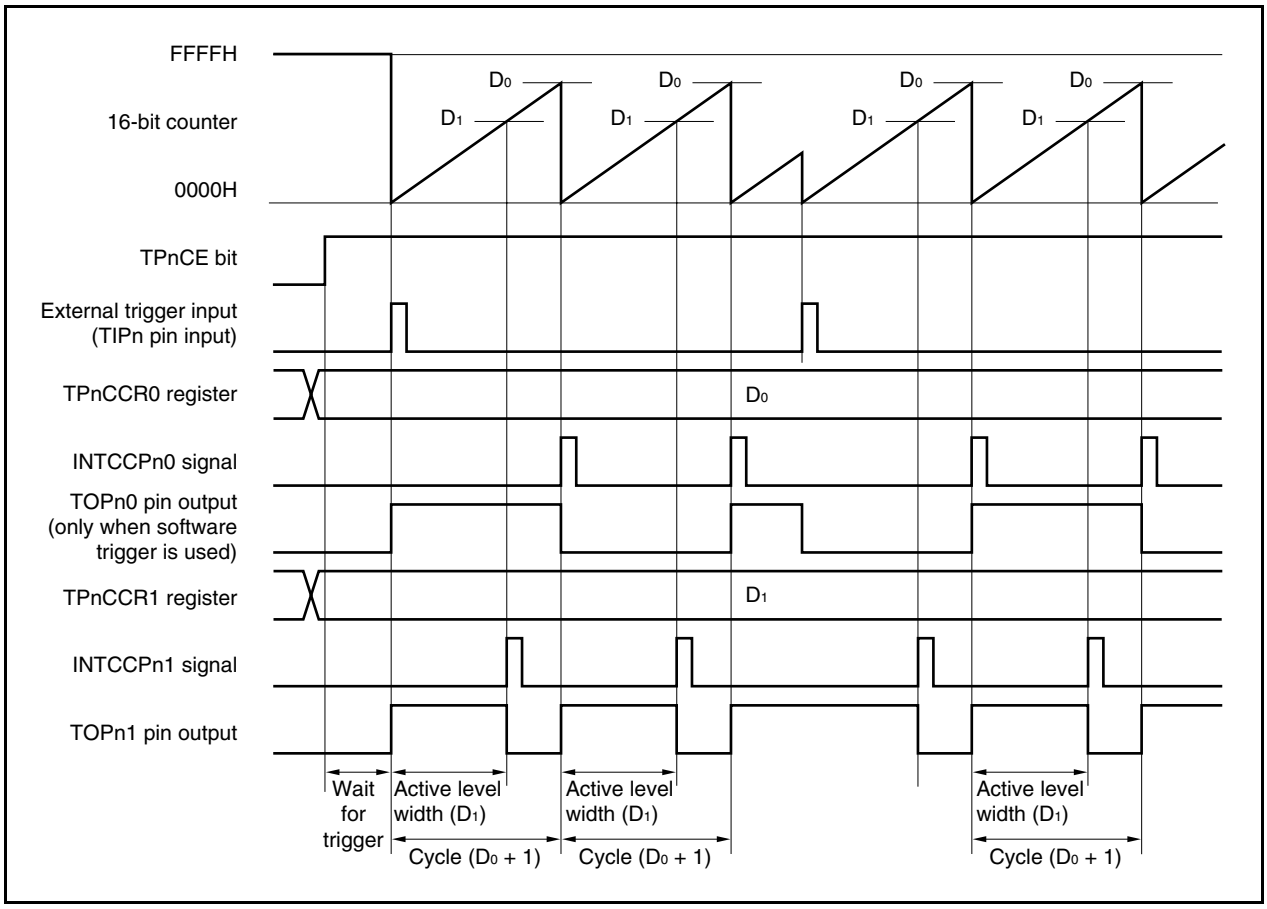
Figure 8-20. Configuration in External Trigger Pulse Output Mode



Note Because the external event count input pin (EVTPn), external trigger input pin (TIPn), and timer output pin (TOPn0) are the same alternate-function pin, two or more functions cannot be used at the same time.

Remark n = 0 to 2

Figure 8-21. Basic Timing in External Trigger Pulse Output Mode



<R> 16-bit timer/event counter P waits for a trigger when the TPnCE 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 wave\form from the TOPn1 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 TOPn0 pin is inverted. The TOPn1 pin outputs a high-level regardless of the status (high/low) when a trigger occurs.)

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

Active level width = (Set value of TPnCCR1 register) × Count clock cycle
Cycle = (Set value of TPnCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TPnCCR1 register)/(Set value of TPnCCR0 register + 1)

The compare match request signal INTCCPn0 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 INTCCPn1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

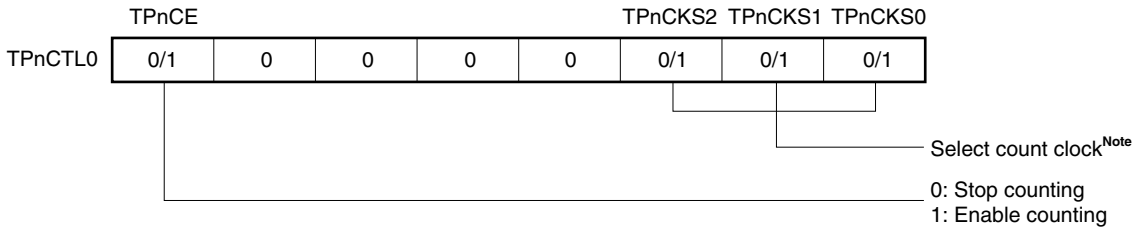
The value set to the TPnCCRa register is transferred to the CCRa buffer register when the count value of the 16-bit counter matches the value of the CCRa buffer register and the 16-bit counter is cleared to 0000H.

The valid edge of an external trigger input (TIPn), or setting the software trigger (TPnCTL1.TPnEST bit) to 1 is used as the trigger.

Remark n = 0 to 2, a = 0, 1

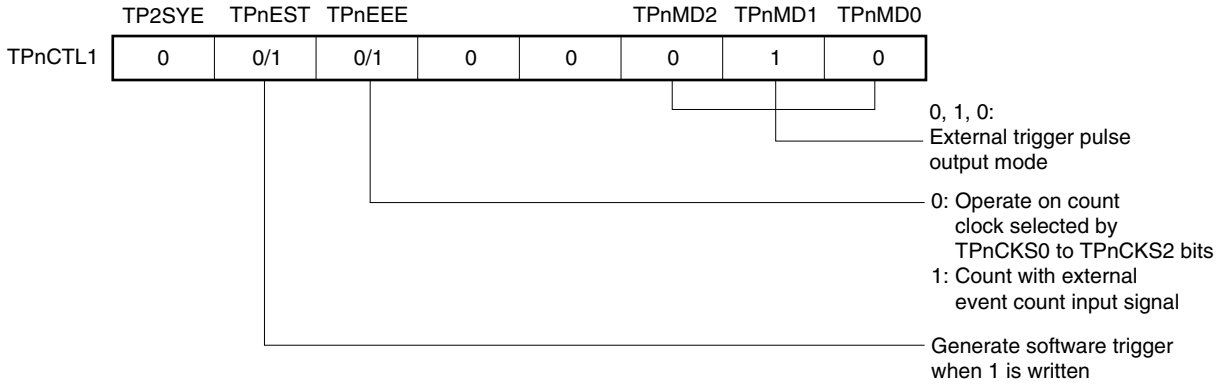
Figure 8-22. Setting of Registers in External Trigger Pulse Output Mode (1/2)

(a) TMPn control register 0 (TPnCTL0)

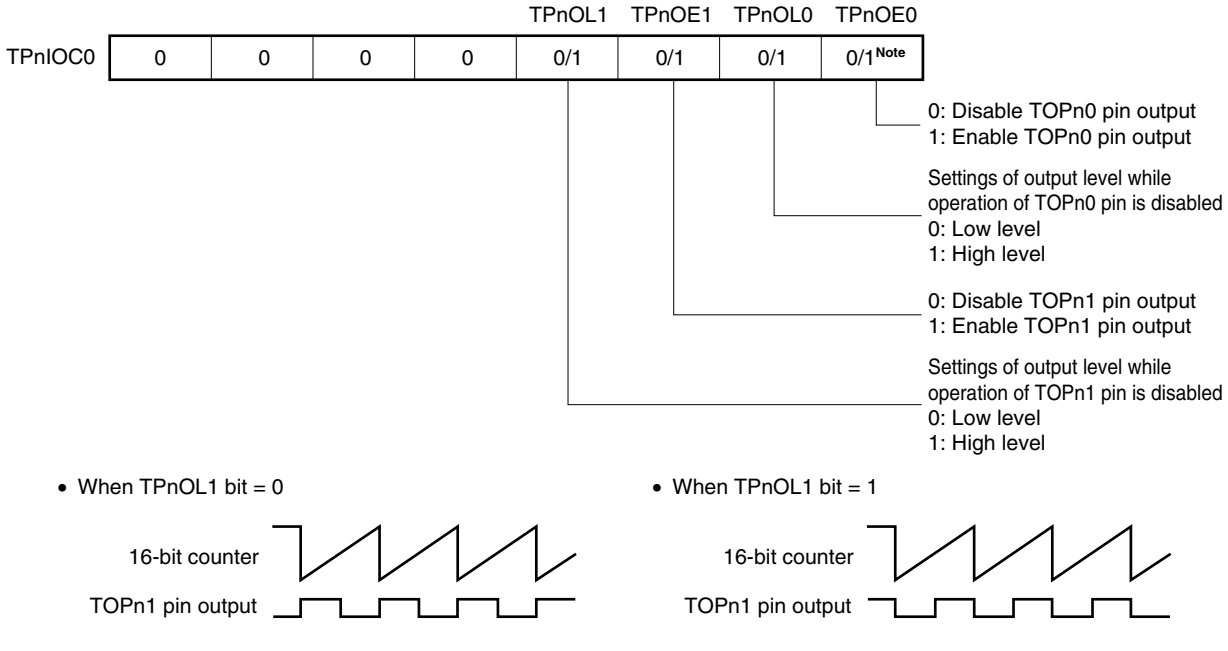


Note The setting is invalid when the TPnCTL1.TPnEEE bit = 1.

(b) TMPn control register 1 (TPnCTL1)



(c) TMPn I/O control register 0 (TPnIOC0)



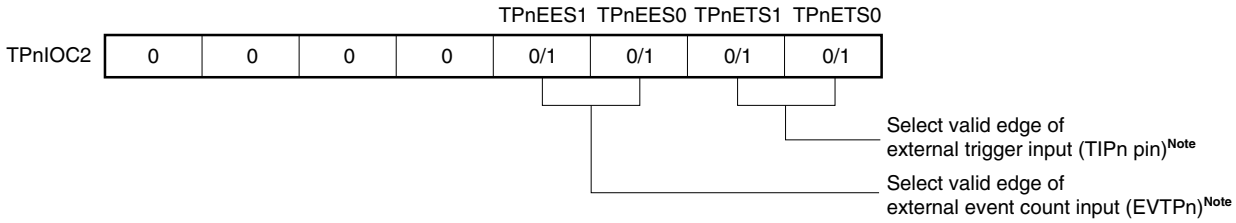
- When TPnOL1 bit = 0

- When TPnOL1 bit = 1

<R> **Note** Clear this bit to 0 when the TOPn0 pin is not used in the external trigger pulse output mode.

Figure 8-22. Setting of Registers in External Trigger Pulse Output Mode (2/2)

(d) TMPn I/O control register 2 (TPnIOC2)



Note Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

(e) TMPn counter read buffer register (TPnCNT)

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

(f) TMPn capture/compare registers 0 and 1 (TPnCCR0 and TPnCCR1)

If D_0 is set to the TPnCCR0 register and D_1 to the TPnCCR1 register, the cycle and active level of the PWM waveform are as follows.

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

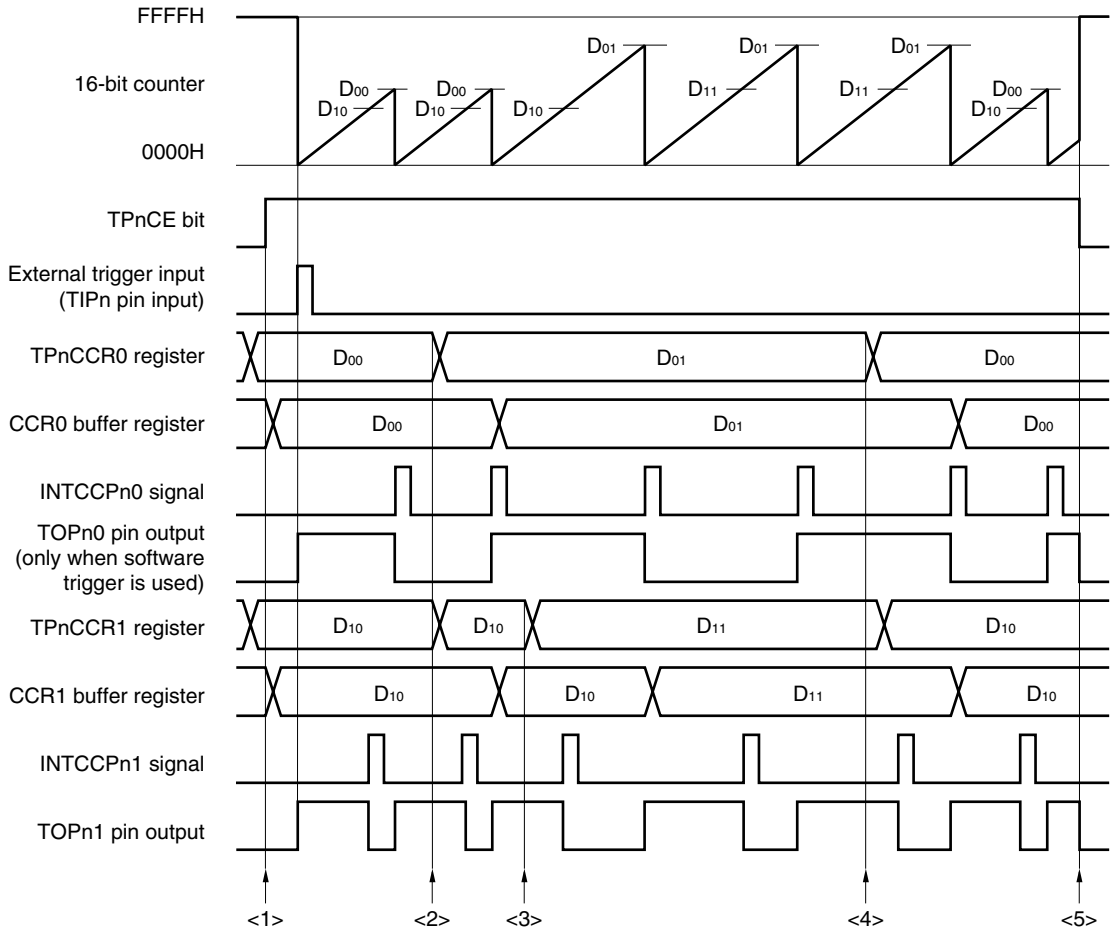
Active level width = $D_1 \times$ Count clock cycle

Remarks 1. TMPn I/O control register 1 (TPnIOC1) and TMPn option register 0 (TPnOPT0) are not used in the external trigger pulse output mode.

2. n = 0 to 2

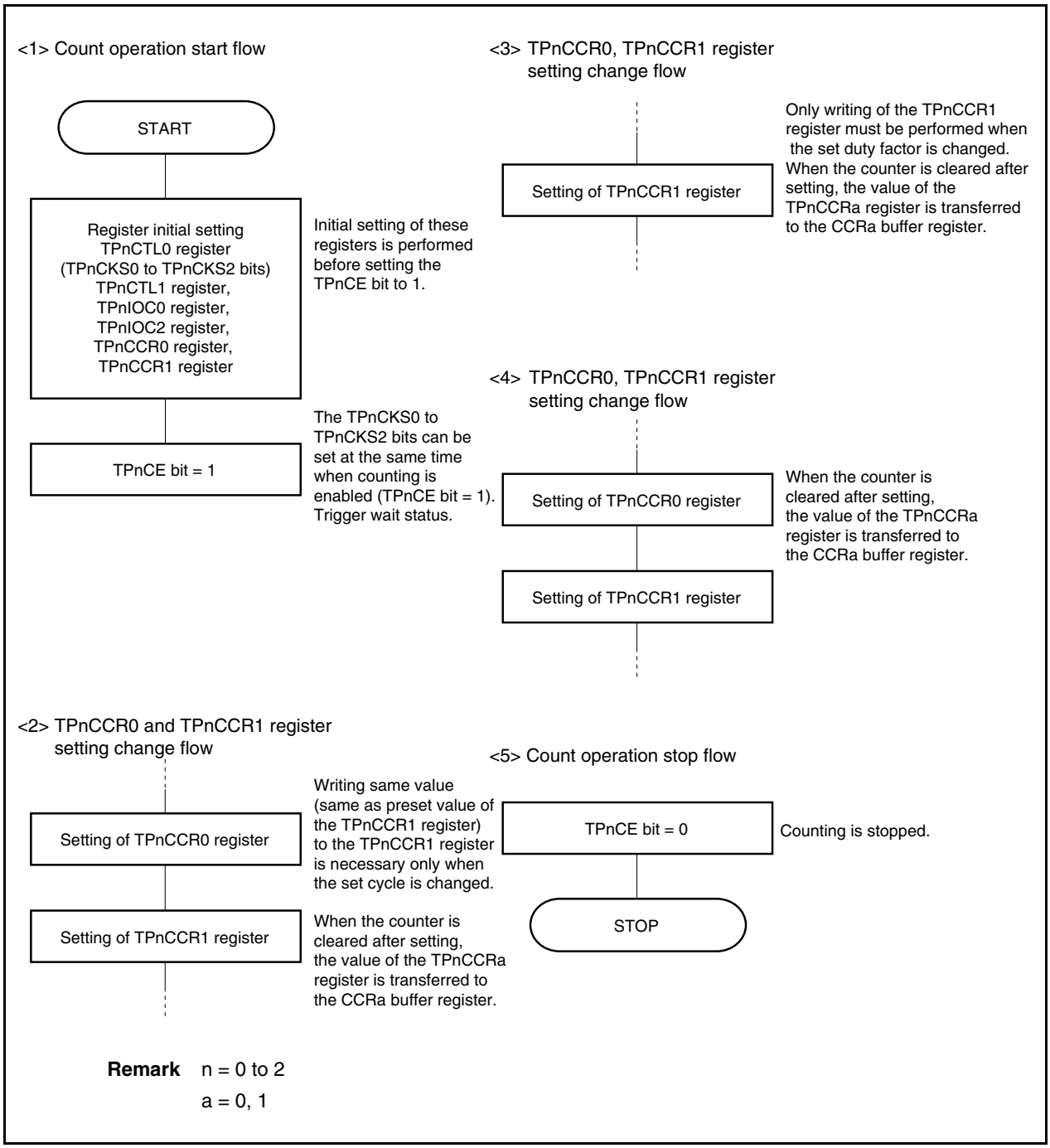
(1) Operation flow in external trigger pulse output mode

Figure 8-23. Software Processing Flow in External Trigger Pulse Output Mode (1/2)



Remark n = 0 to 2

Figure 8-23. Software Processing Flow in External Trigger Pulse Output Mode (2/2)



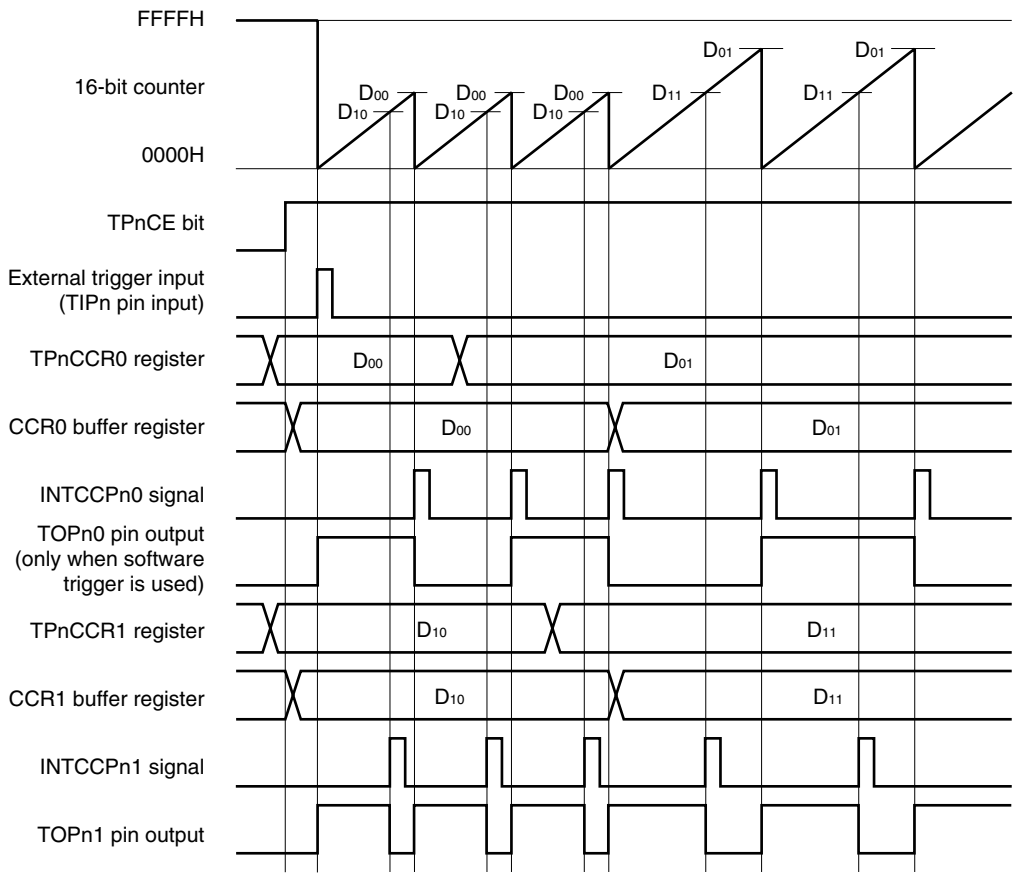
Remark n = 0 to 2
a = 0, 1

(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 TPnCCR1 register last.

Rewrite the TPnCCRa register after writing the TPnCCR1 register after the INTCCPn0 signal is detected.



Remark n = 0 to 2

In order to transfer data from the TPnCCRa register to the CCRa buffer register, the TPnCCR1 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 TPnCCR0 register and then set the active level width to the TPnCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TPnCCR0 register, and then write the same value (same as preset value of the TPnCCR1 register) to the TPnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TPnCCR1 register has to be set.

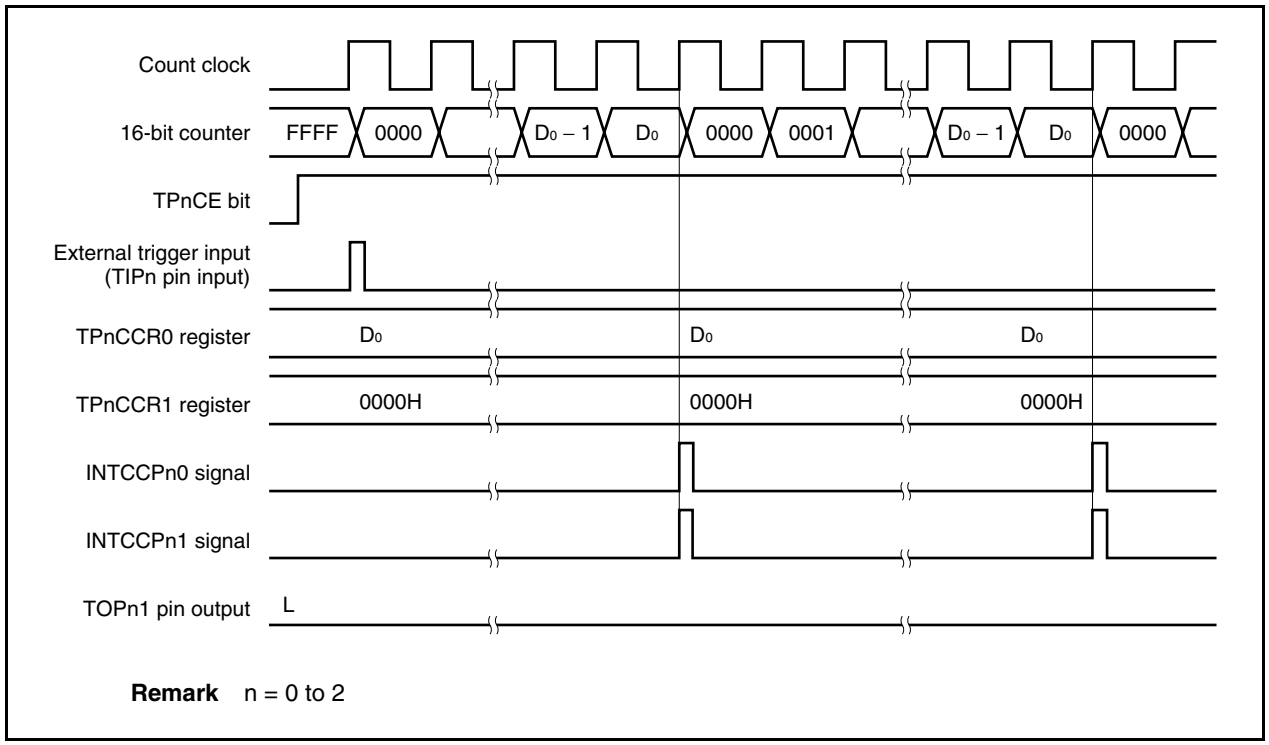
After data is written to the TPnCCR1 register, the value written to the TPnCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TPnCCR0 or TPnCCR1 register again after writing the TPnCCR1 register once, do so after the INTCCPn0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TPnCCRa register to the CCRa buffer register conflicts with writing the TPnCCRa register.

Remark n = 0 to 2
a = 0, 1

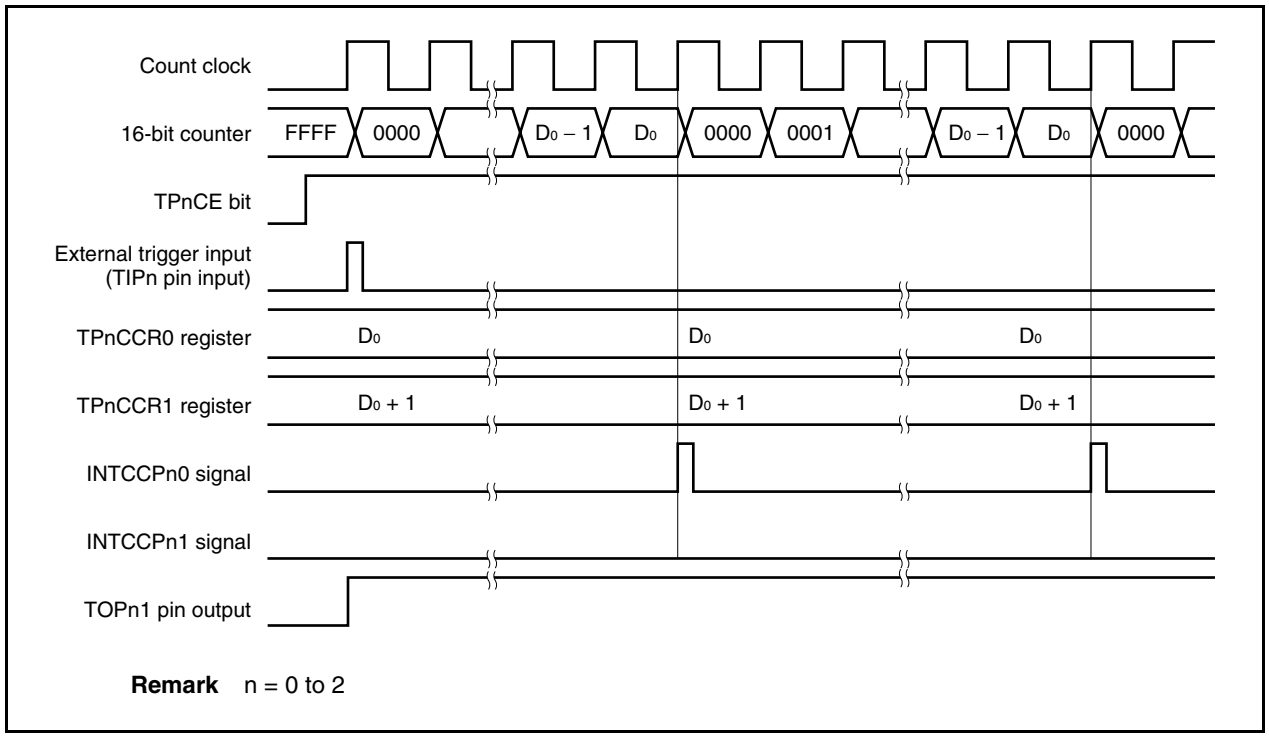
(b) 0%/100% output of PWM waveform

<R> To output a 0% waveform, set the TPnCCR1 register to 0000H. The 16-bit counter is cleared to 0000H and the INTCCPn0 and INTCCPn1 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.



Remark n = 0 to 2

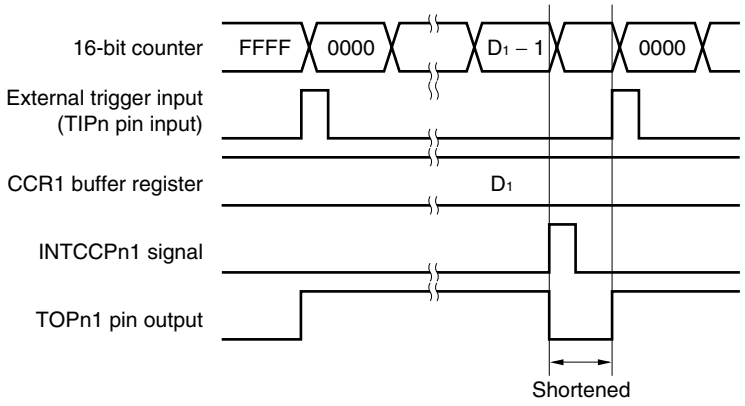
To output a 100% waveform, set a value of (set value of TPnCCR0 register + 1) to the TPnCCR1 register. If the set value of the TPnCCR0 register is FFFFH, 100% output cannot be produced.



Remark n = 0 to 2

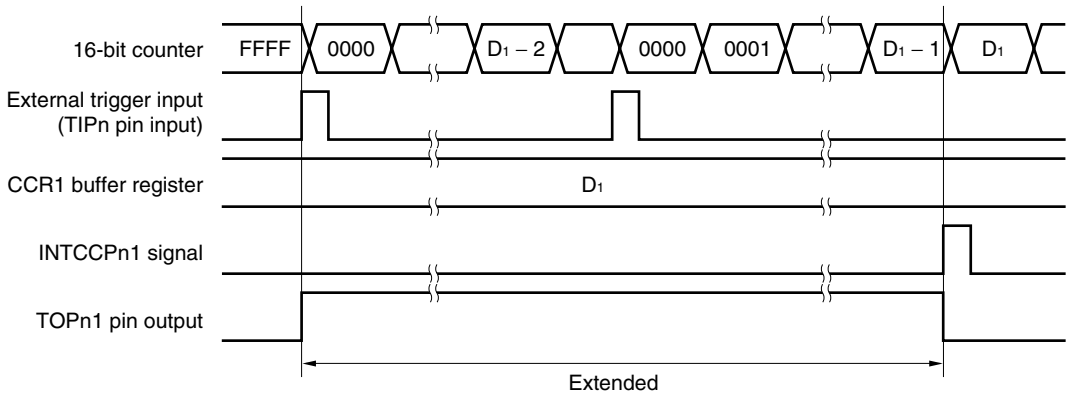
(c) Conflict between trigger detection and match with CCR1 buffer register

If the trigger is detected immediately after the INTCCPn1 signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TOPn1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark n = 0 to 2

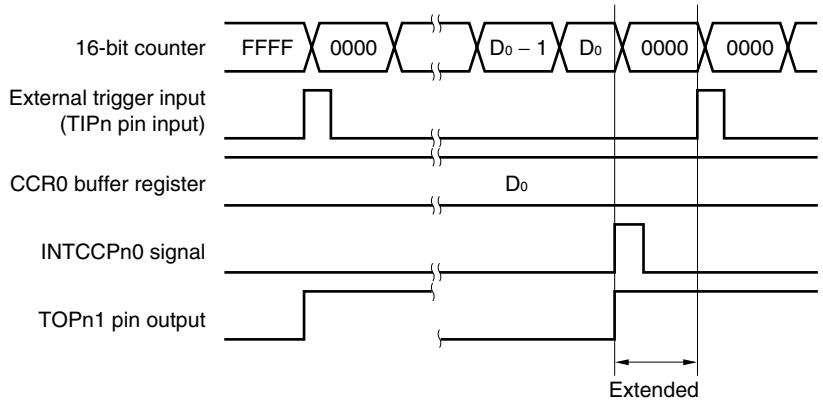
If the trigger is detected immediately before the INTCCPn1 signal is generated, the INTCCPn1 signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOPn1 pin remains active. Consequently, the active period of the PWM waveform is extended.



Remark n = 0 to 2

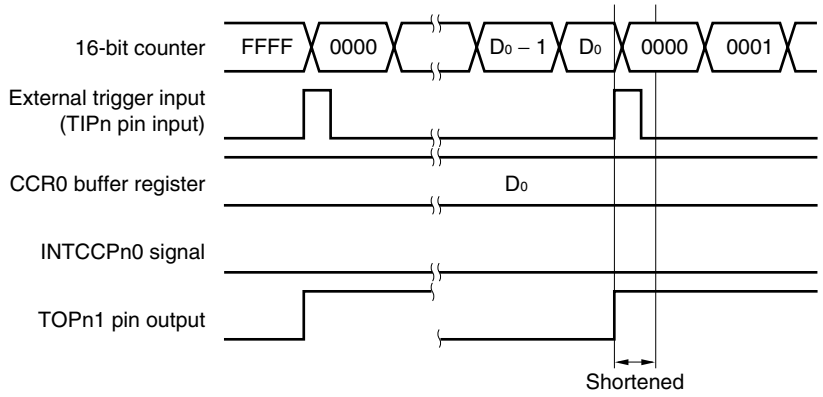
(d) Conflict between trigger detection and match with CCR0 buffer register

If the trigger is detected immediately after the INTCCPn0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOPn1 pin is extended by time from generation of the INTCCPn0 signal to trigger detection.



Remark n = 0 to 2

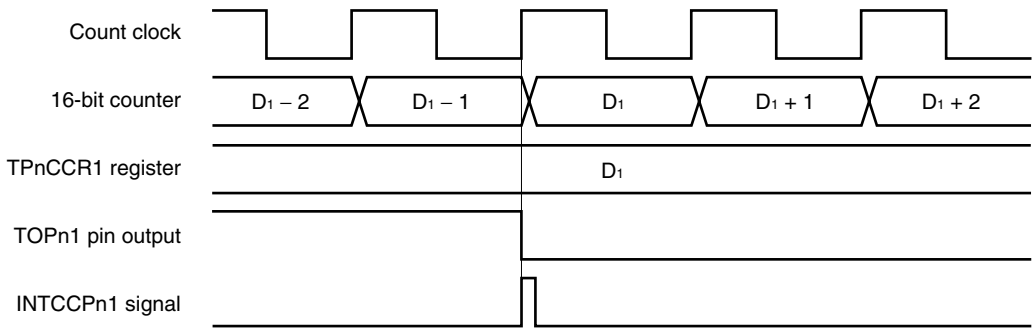
If the trigger is detected immediately before the INTCCPn0 signal is generated, the INTCCPn0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOPn1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark n = 0 to 2

(e) Generation timing of compare match interrupt request signal (INTCCPn1)

The timing of generation of the INTCCPn1 signal in the external trigger pulse output mode differs from the timing of INTCCPn1 signals in other mode; the INTCCPn1 signal is generated when the count value of the 16-bit counter matches the value of the TPnCCR1 register.



Remark n = 0 to 2

Usually, the INTCCPn1 signal is generated in synchronization with the next count-up, after the count value of the 16-bit counter matches the value of the TPnCCR1 register.

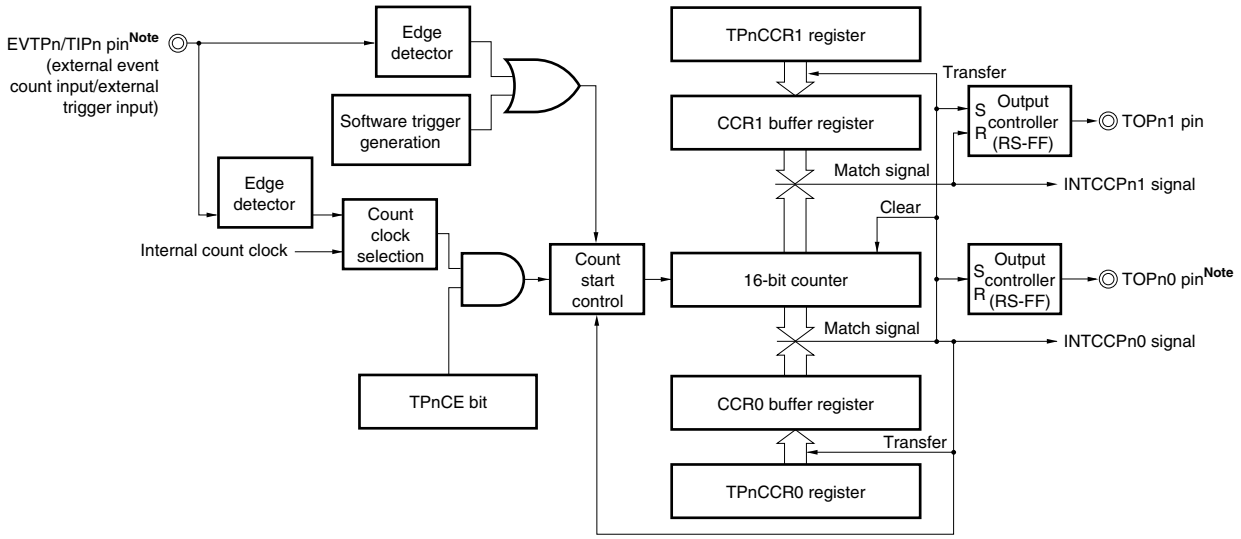
In the external trigger pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the timing of changing the output signal of the TOPn1 pin.

8.6.4 One-shot pulse output mode (TPnMD2 to TPnMD0 bits = 011)

In the one-shot pulse output mode, 16-bit timer/event counter P waits for a trigger when the TPnCTL0.TPnCE bit is set to 1. When the valid edge of an external trigger input (TIPn) is detected, 16-bit timer/event counter P starts counting, and outputs a one-shot pulse from the TOPn1 pin.

Instead of the external trigger, a software trigger can also be generated to output the pulse. When the software trigger is used, the TOPn0 pin outputs the active level while the 16-bit counter is counting, and the inactive level when the counter is stopped (waiting for a trigger).

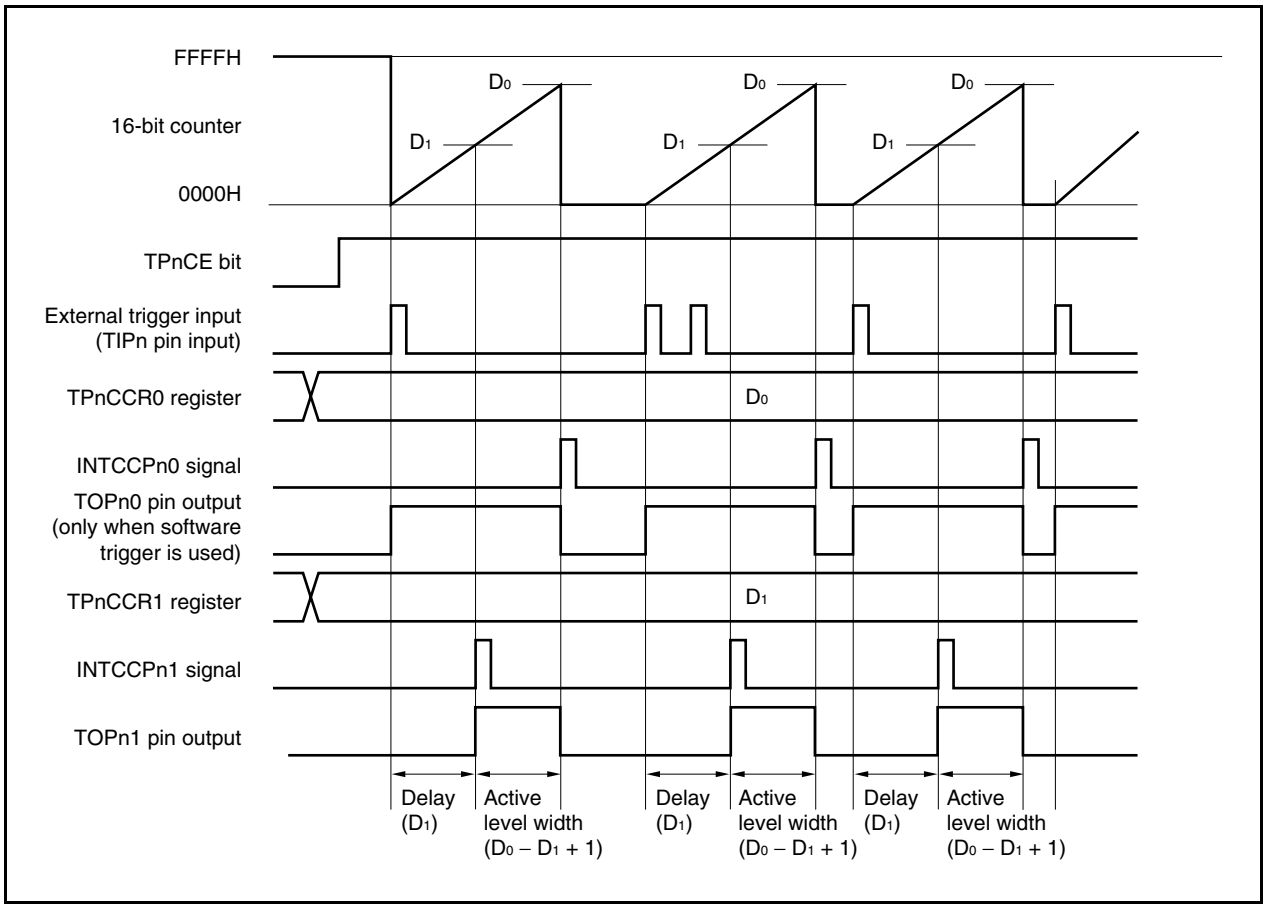
Figure 8-24. Configuration in One-Shot Pulse Output Mode



Note Because the external event count input pin (EVTPn), external trigger input pin (TIPn), and timer output pin (TOPn0) are the same alternate-function pin, two or more functions cannot be used at the same time.

Remark n = 0 to 2

Figure 8-25. Basic Timing in One-Shot Pulse Output Mode



When the TPnCE 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 TOPn1 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 TPnCCR1 register) × Count clock cycle

Active level width = (Set value of TPnCCR0 register − Set value of TPnCCR1 register + 1) × Count clock cycle

The compare match interrupt request signal INTCCRn0 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 INTCCRn1 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 (TIPn pin) or setting the software trigger (TPnCTL1.TPnEST bit) to 1 is used as the trigger.

Remark n = 0 to 2

Figure 8-26. Setting of Registers in One-Shot Pulse Output Mode (1/2)

(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[Note]

0: Stop counting
1: Enable counting

Note The setting is invalid when the TPnCTL1.TPnEEE bit = 1.

(b) TMPn control register 1 (TPnCTL1)

	TP2SYE	TPnEST	TPnEEE			TPnMD2	TPnMD1	TPnMD0
TPnCTL1	0	0/1	0/1	0	0	0	1	1

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

0: Operate on count clock selected by TPnCKS0 to TPnCKS2 bits
1: Count external event input signal

Generate software trigger when 1 is written

(c) TMPn I/O control register 0 (TPnIOC0)

<R>

					TPnOL1	TPnOE1	TPnOL0	TPnOE0
TPnIOC0	0	0	0	0	0/1	0/1	0/1	0/1[Note]

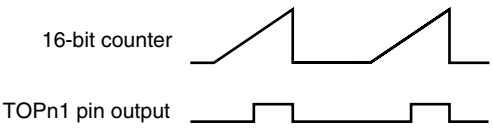
0: Disable TOPn0 pin output
1: Enable TOPn0 pin output

Setting of output level while operation of TOPn0 pin is disabled
0: Low level
1: High level

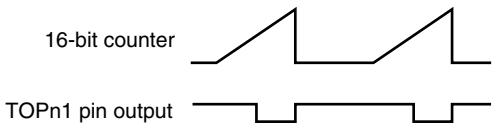
0: Disable TOPn1 pin output
1: Enable TOPn1 pin output

Setting of output level while operation of TOPn1 pin is disabled
0: Low level
1: High level

• When TPnOL1 bit = 0



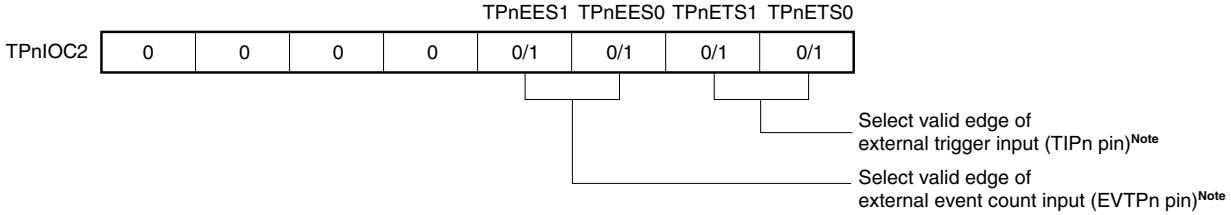
• When TPnOL1 bit = 1



<R> **Note** Clear this bit to 0 when the TOPn0 pin is not used in the one-shot pulse output mode.

Figure 8-26. Setting of Registers in One-Shot Pulse Output Mode (2/2)

(d) TMPn I/O control register 2 (TPnIOC2)



Note Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

(e) TMPn counter read buffer register (TPnCNT)

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

(f) TMPn capture/compare registers 0 and 1 (TPnCCR0 and TPnCCR1)

If D_0 is set to the TPnCCR0 register and D_1 to the TPnCCR1 register, the active level width and output delay period of the one-shot pulse are as follows.

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

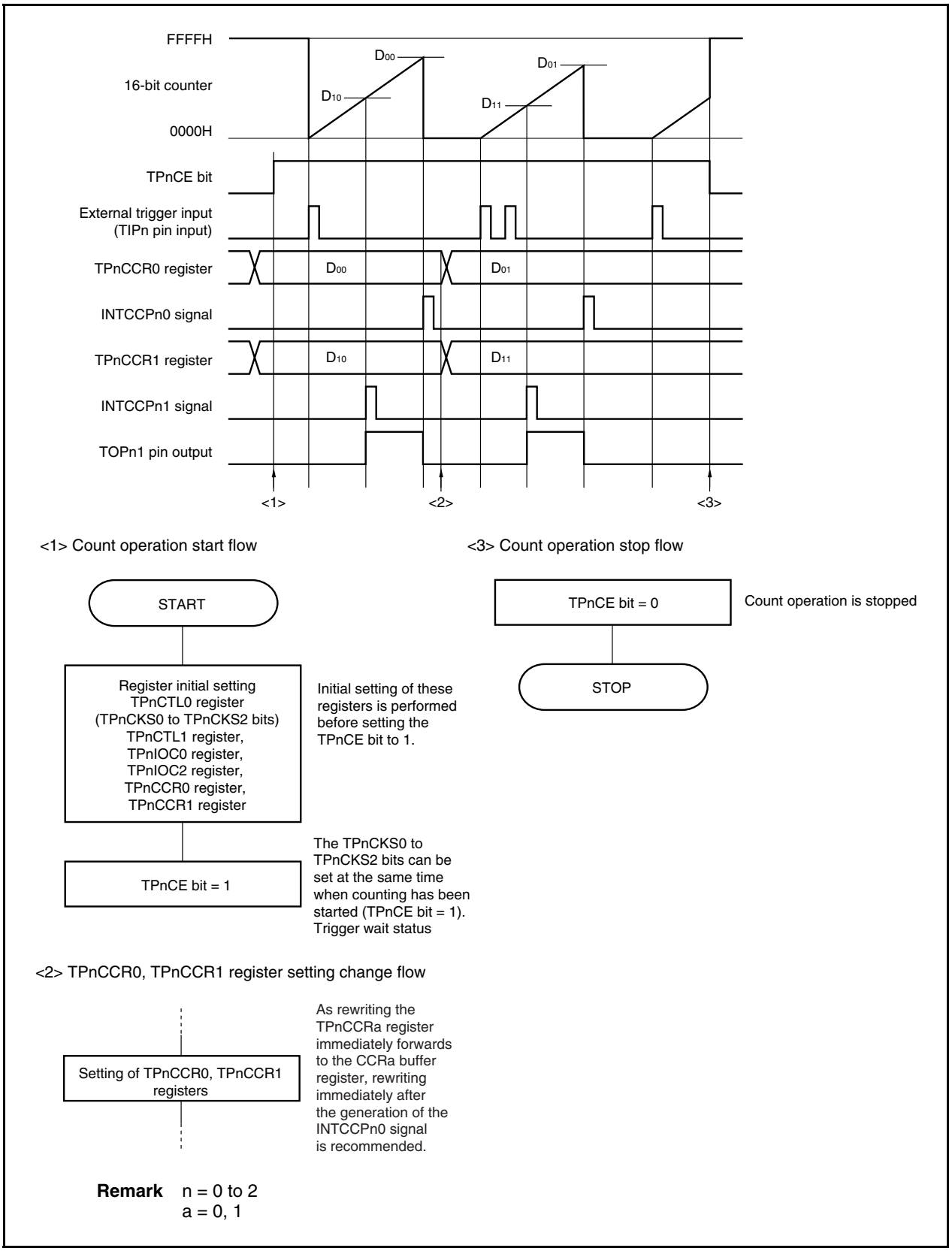
Output delay period = $D_1 \times$ Count clock cycle

Remarks 1. TMPn I/O control register 1 (TPnIOC1) and TMPn option register 0 (TPnOPT0) are not used in the one-shot pulse output mode.

2. n = 0 to 2

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

<R> **Figure 8-27. Software Processing Flow in One-Shot Pulse Output Mode**

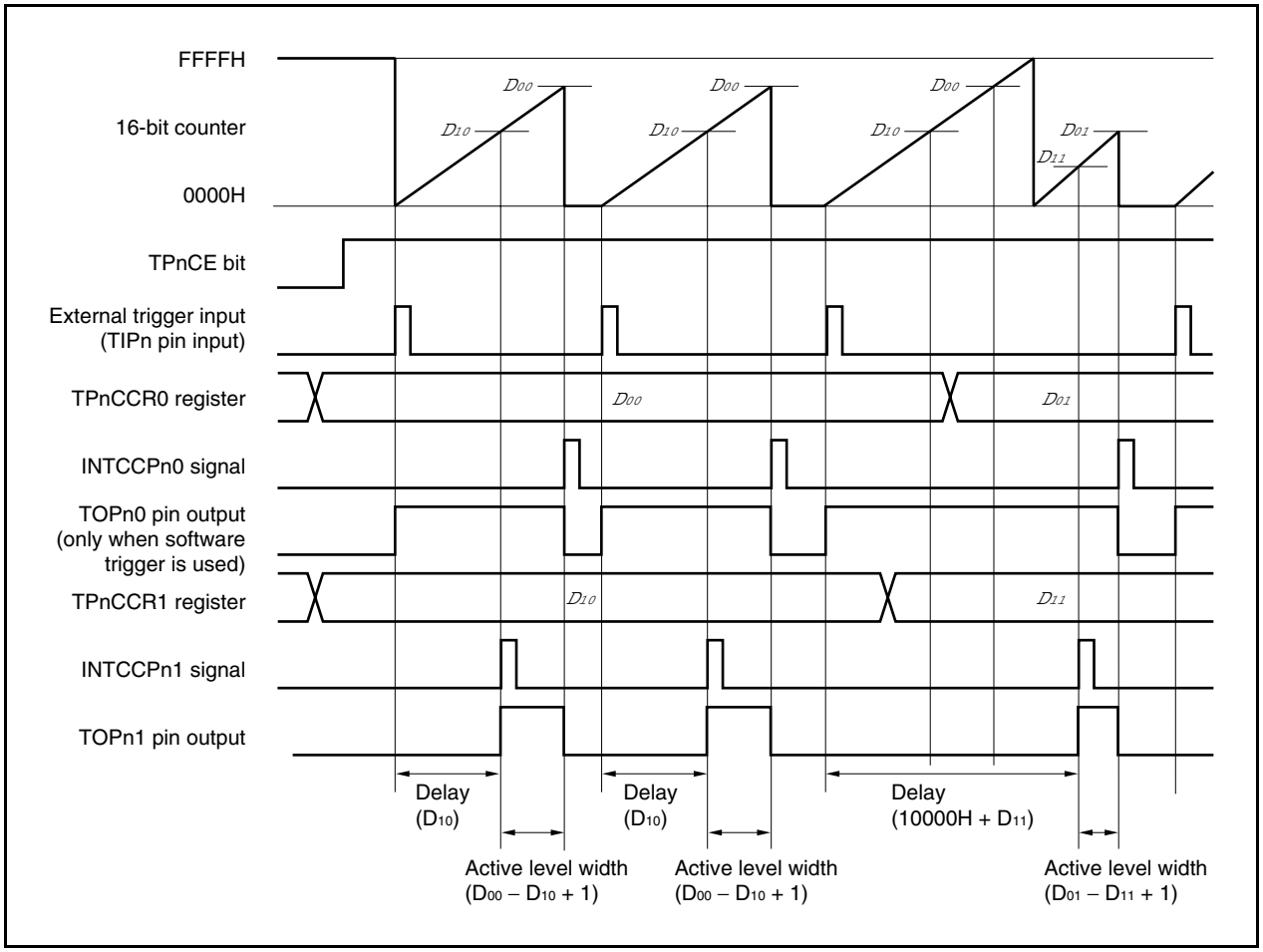


Remark n = 0 to 2
a = 0, 1

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

(a) Note on rewriting TPnCCRa register

If the value of the TPnCCRa 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.



<R>

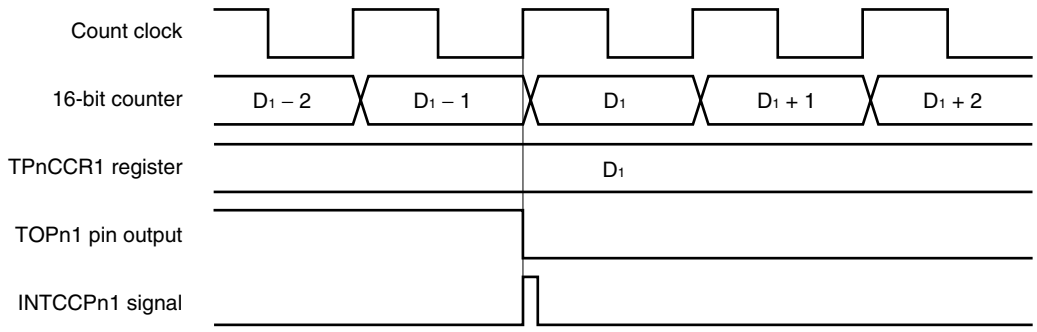
When the TPnCCR0 register is rewritten from D_{00} to D_{01} and the TPnCCR1 register from D_{10} to D_{11} where $D_{00} > D_{01}$ and $D_{10} > D_{11}$, if the TPnCCR1 register is rewritten when the count value of the 16-bit counter is greater than D_{11} and less than D_{10} and if the TPnCCR0 register 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 INTCCPn1 signal and asserts the TOPn1 pin. When the count value matches D_{01}, the counter generates the INTCCPn0 signal, deasserts the TOPn1 pin, and stops counting.

Therefore, the counter may output a pulse with a delay period or active period different from that of the one-shot pulse that is originally expected.

Remark n = 0 to 2, a = 0, 1

(b) Generation timing of compare match interrupt request signal (INTCCPn1)

The generation timing of the INTCCPn1 signal in the one-shot pulse output mode is different from INTCCPn1 signals in other mode; the INTCCPn1 signal is generated when the count value of the 16-bit counter matches the value of the TPnCCR1 register.



Remark n = 0 to 2

Usually, the INTCCPn1 signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TPnCCR1 register.

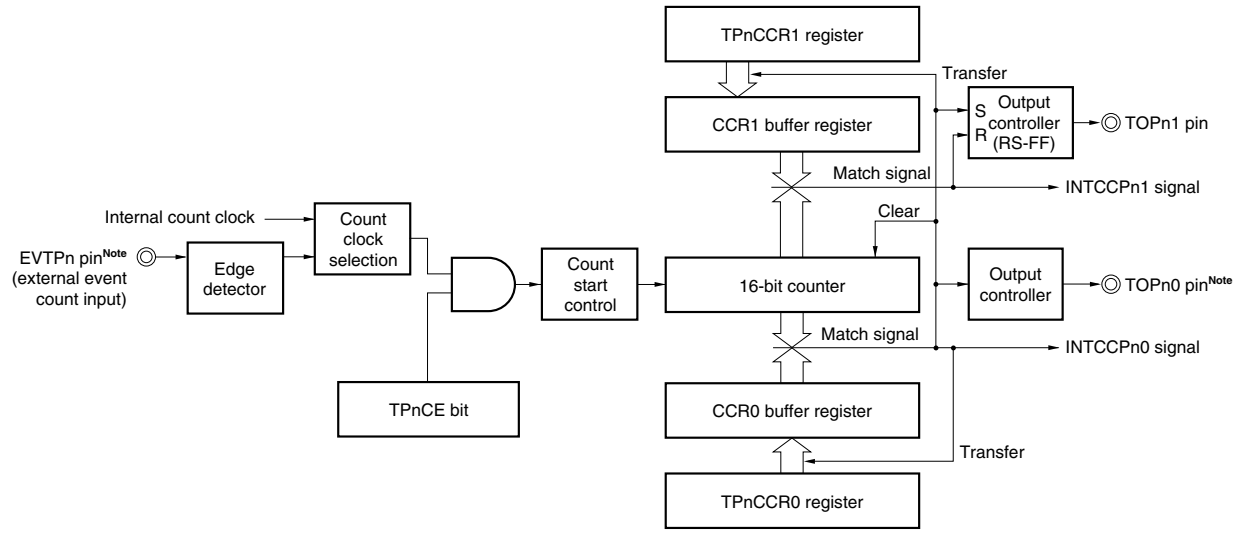
In the one-shot pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the TOPn1 pin.

8.6.5 PWM output mode (TPnMD2 to TPnMD0 bits = 100)

In the PWM output mode, a PWM waveform is output from the TOPn1 pin when the TPnCTL0.TPnCE bit is set to 1.

In addition, a PWM waveform with a duty factor of 50% with the set value of the TPnCCR0 register + 1 as half its cycle is output from the TOPn0 pin.

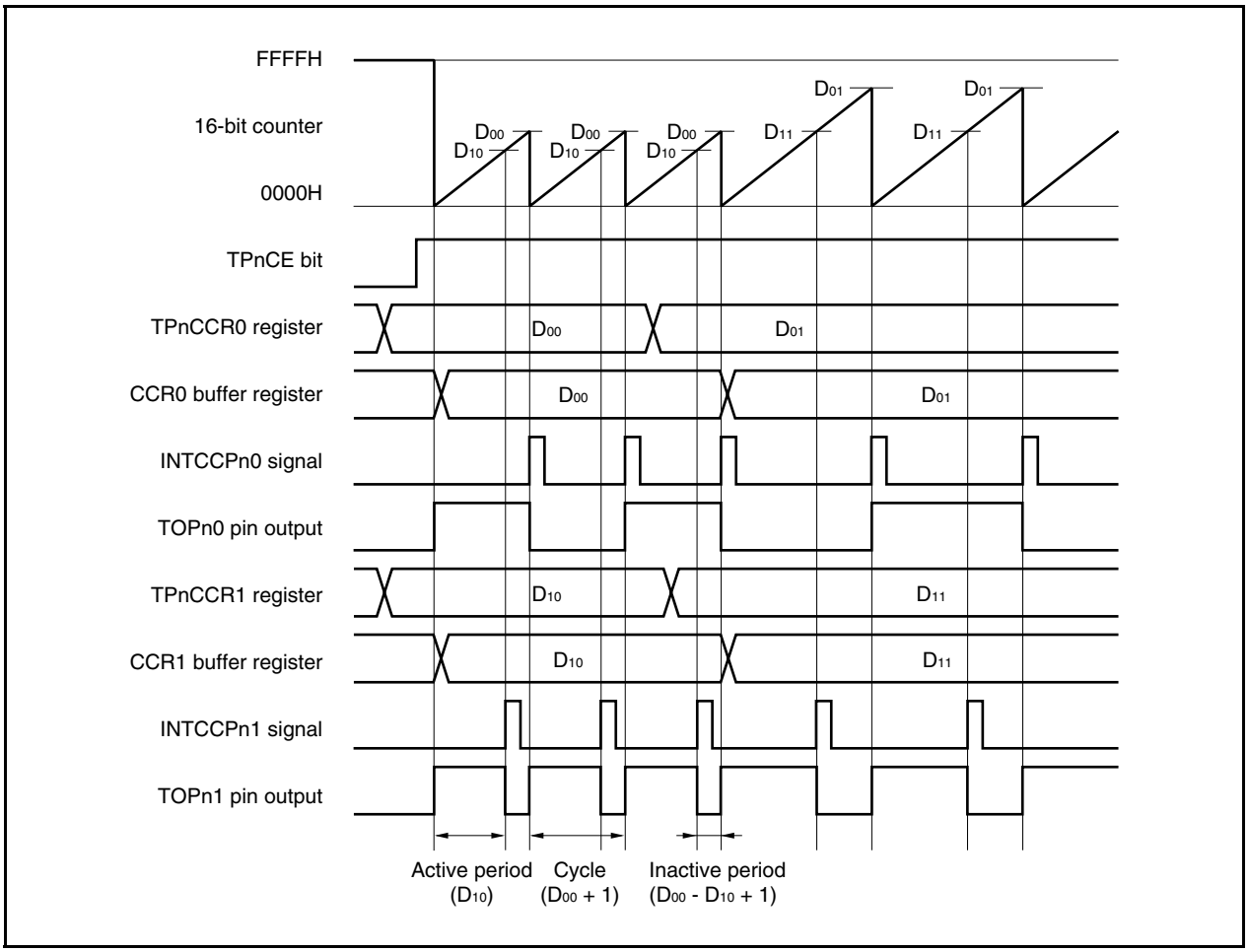
Figure 8-28. Configuration in PWM Output Mode



Note Because the external event count input pin (EVTPn) and timer output pin (TOPn0) are the same alternate-function pin, two or more functions cannot be used at the same time.

Remark n = 0 to 2

Figure 8-29. Basic Timing in PWM Output Mode



When the TPnCE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a PWM waveform from the TOPn1 pin.

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

Active level width = (Set value of TPnCCR1 register) × Count clock cycle
Cycle = (Set value of TPnCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TPnCCR1 register)/(Set value of TPnCCR0 register + 1)

The PWM waveform can be changed by rewriting the TPnCCRa 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 INTCCPn0 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 INTCCPn1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TPnCCRa 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 n = 0 to 2, a = 0, 1

Figure 8-30. Setting of Registers in PWM Output Mode (1/2)

(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[Note]

0: Stop counting
1: Enable counting

Note The setting is invalid when the TPnCTL1.TPnEEE bit = 1.

(b) TMPn control register 1 (TPnCTL1)

	TP2SYE	TPnEST	TPnEEE			TPnMD2	TPnMD1	TPnMD0
TPnCTL1	0	0	0/1	0	0	1	0	0

1, 0, 0:
PWM output mode

0: Operate on count clock selected by TPnCKS0 to TPnCKS2 bits
1: Count with external event count input signal

(c) TMPn I/O control register 0 (TPnIOC0)

<R>

					TPnOL1	TPnOE1	TPnOL0	TPnOE0
TPnIOC0	0	0	0	0	0/1	0/1	0/1	0/1[Note]

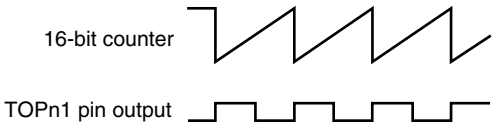
0: Disable TOPn0 pin output
1: Enable TOPn0 pin output

Setting of output level while operation of TOPn0 pin is disabled
0: Low level
1: High level

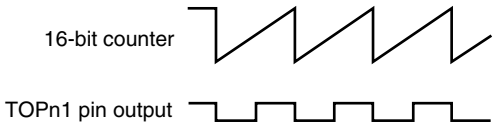
0: Disable TOPn1 pin output
1: Enable TOPn1 pin output

Setting of output level while operation of TOPn1 pin is disabled
0: Low level
1: High level

- When TPnOL1 bit = 0



- When TPnOL1 bit = 1



<R>

Note Clear this bit to 0 when the TOPn0 pin is not used in the PWM output mode.

Figure 8-30. Register Setting in PWM Output Mode (2/2)

(d) TMPn I/O control register 2 (TPnIOC2)

					TPnEES1	TPnEES0	TPnETS1	TPnETS0
TPnIOC2	0	0	0	0	0/1	0/1	0	0

Select valid edge of external event count input (EVTPn pin).

(e) TMPn counter read buffer register (TPnCNT)

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

(f) TMPn capture/compare registers 0 and 1 (TPnCCR0 and TPnCCR1)

If D_0 is set to the TPnCCR0 register and D_1 to the TPnCCR1 register, the cycle and active level of the PWM waveform are as follows.

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

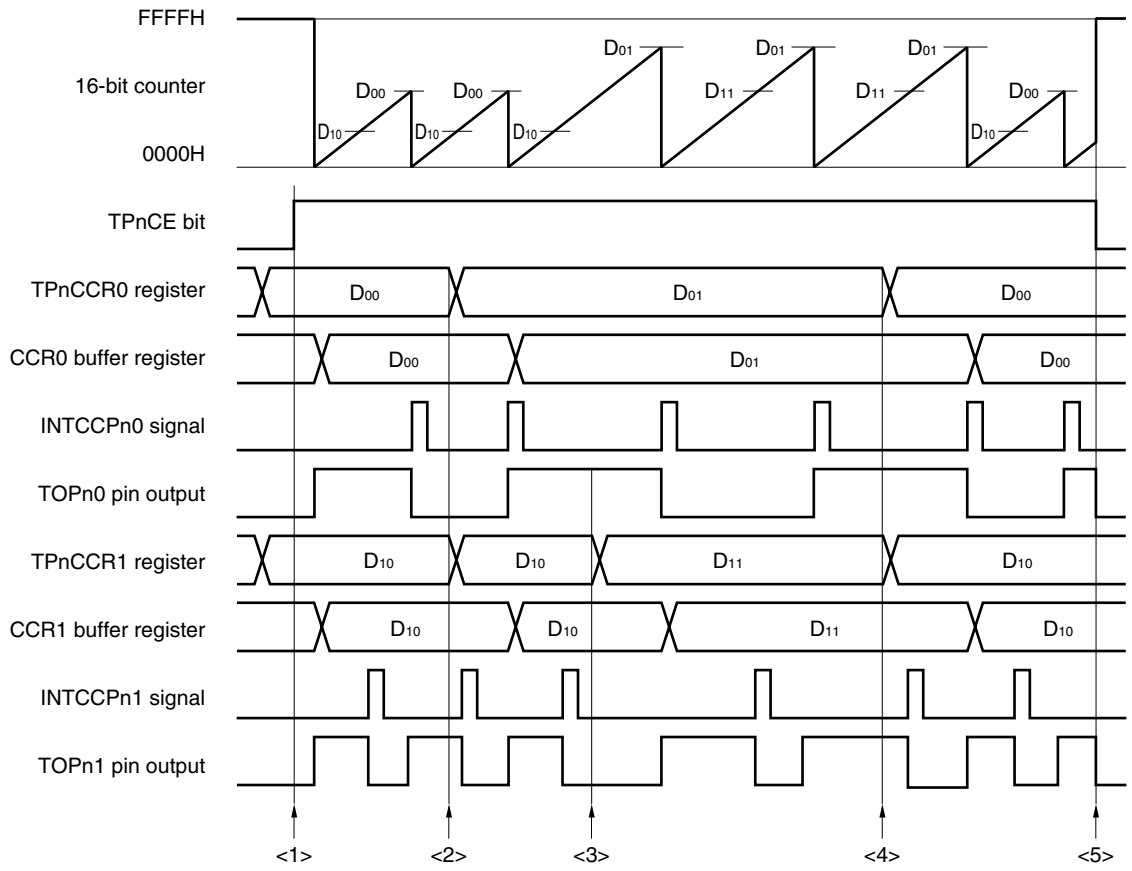
Active level width = $D_1 \times$ Count clock cycle

Remarks 1. TMPn I/O control register 1 (TPnIOC1) and TMPn option register 0 (TPnOPT0) are not used in the PWM output mode.

2. n = 0 to 2

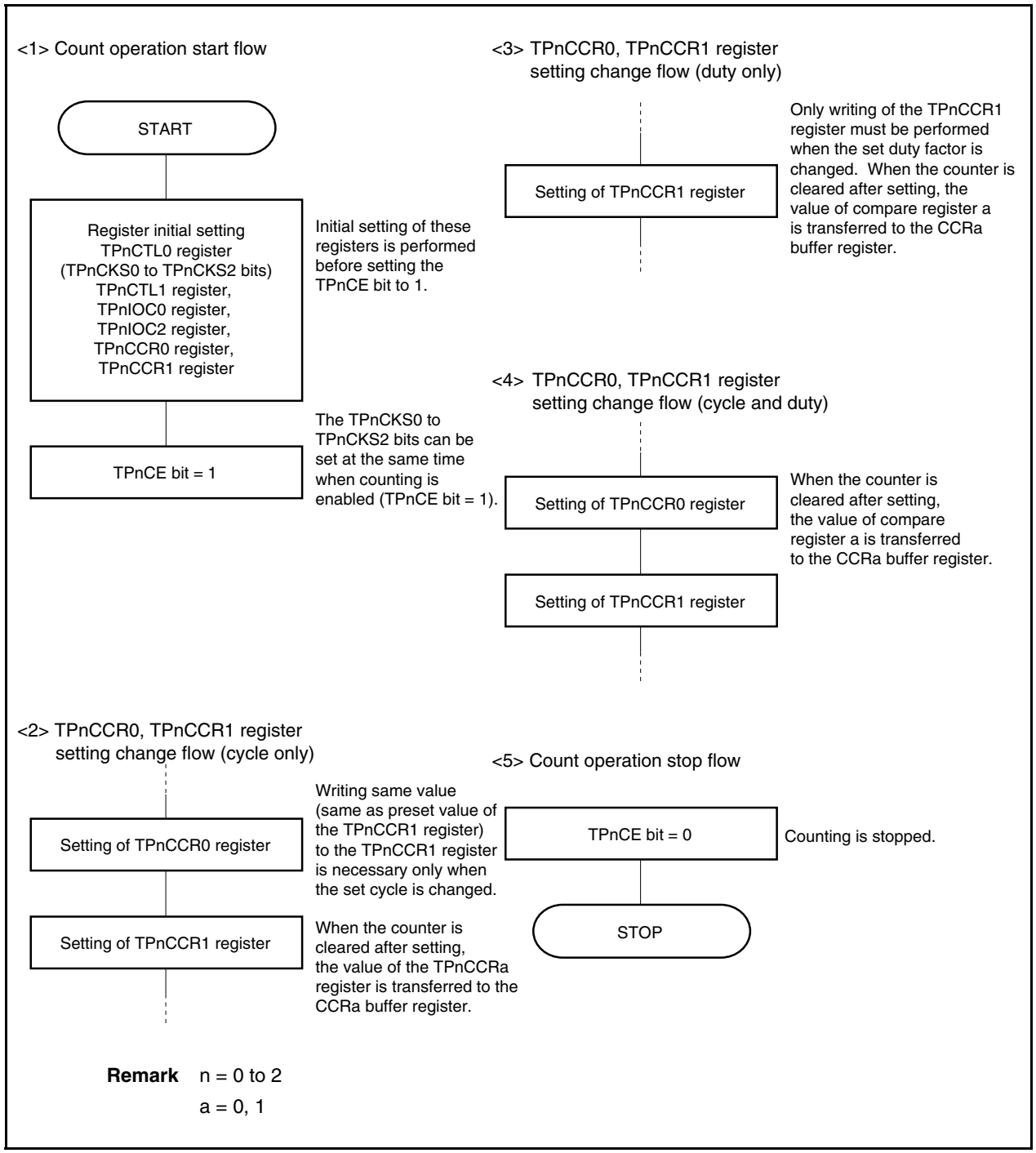
(1) Operation flow in PWM output mode

Figure 8-31. Software Processing Flow in PWM Output Mode (1/2)



Remark n = 0 to 2

Figure 8-31. Software Processing Flow in PWM Output Mode (2/2)



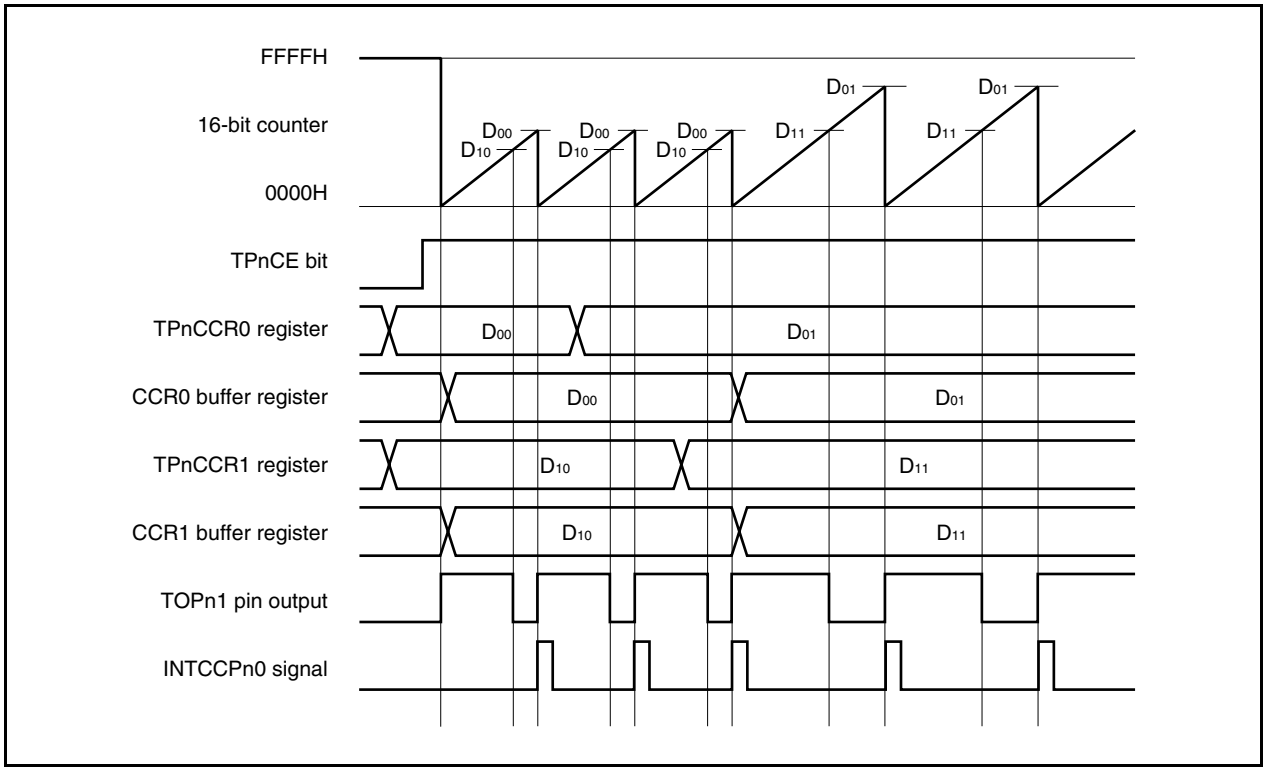
Remark n = 0 to 2
a = 0, 1

(2) PWM output mode operation timing

(a) Changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TPnCCR1 register last.

Rewrite the TPnCCRa register after writing the TPnCCR1 register after the INTCCPn1 signal is detected.



To transfer data from the TPnCCRa register to the CCRa buffer register, the TPnCCR1 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 TPnCCR0 register and then set the active level to the TPnCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TPnCCR0 register, and then write the same value (same as preset value of the TPnCCR1 register) to the TPnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TPnCCR1 register has to be set.

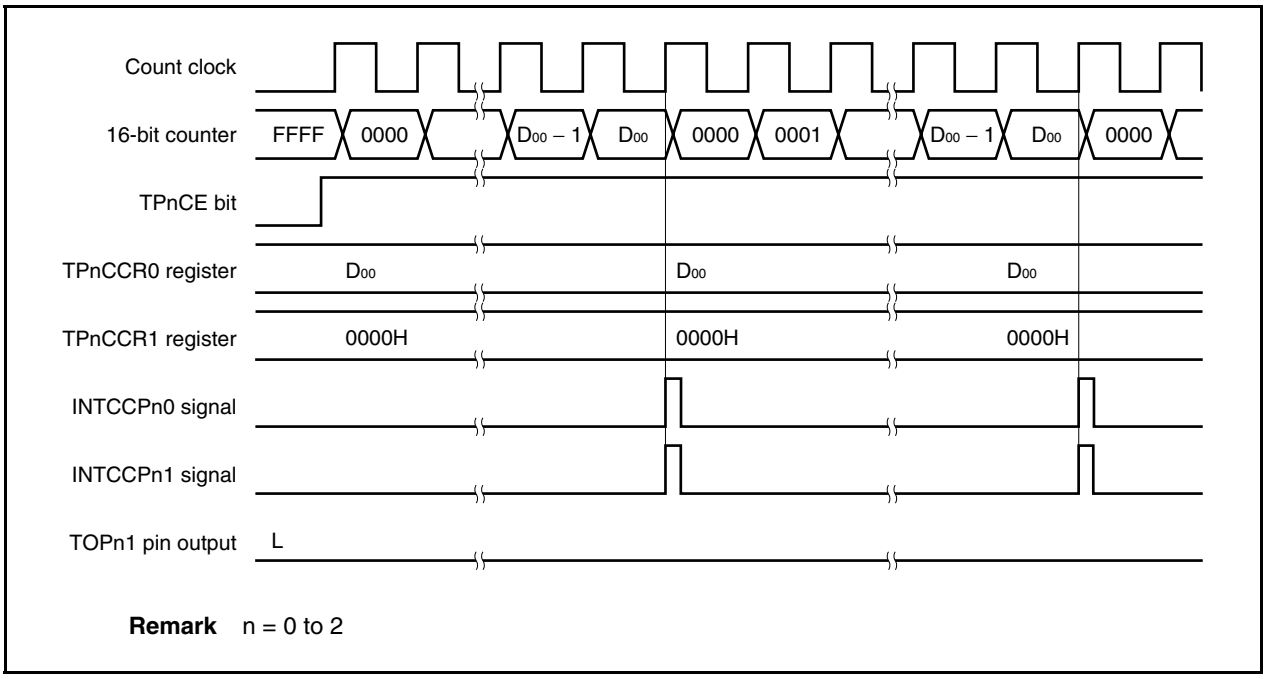
After data is written to the TPnCCR1 register, the value written to the TPnCCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TPnCCR0 or TPnCCR1 register again after writing the TPnCCR1 register once, do so after the INTCCPn0 signal is generated. Otherwise, the value of the CCRa buffer register may become undefined because the timing of transferring data from the TPnCCRa register to the CCRa buffer register conflicts with writing the TPnCCRa register.

Remark n = 0 to 2
a = 0, 1

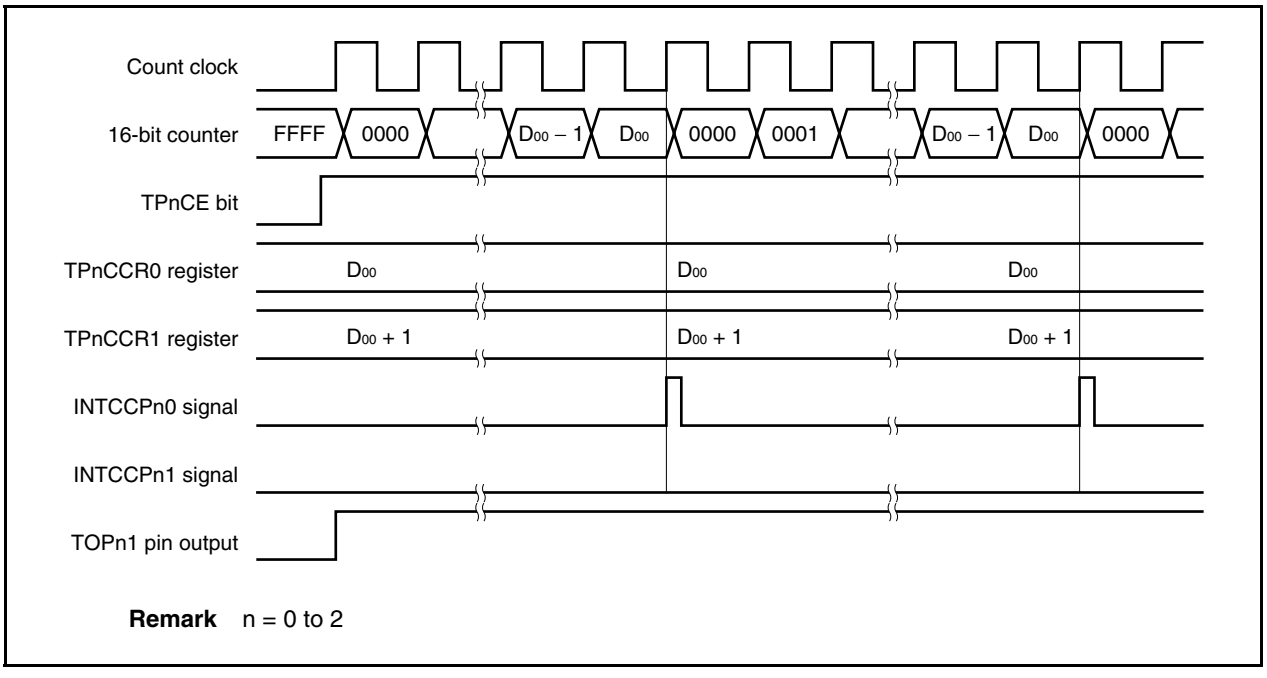
(b) 0%/100% output of PWM waveform

<R> To output a 0% waveform, set the TPnCCR1 register to 0000H. The 16-bit counter is cleared to 0000H and the INTCCPn0 and INTCCPn1 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.



Remark n = 0 to 2

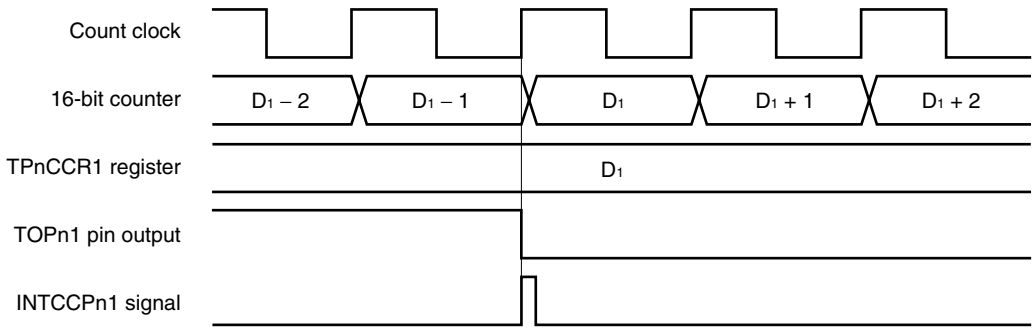
To output a 100% waveform, set a value of (set value of TPnCCR0 register + 1) to the TPnCCR1 register. If the set value of the TPnCCR0 register is FFFFH, 100% output cannot be produced.



Remark n = 0 to 2

(c) Generation timing of compare match interrupt request signal (INTCCPn1)

The timing of generation of the INTCCPn1 signal in the PWM output mode differs from the timing of INTCCPn1 signals in other modes; the INTCCPn1 signal is generated when the count value of the 16-bit counter matches the value of the TPnCCR1 register.



Remark n = 0 to 2

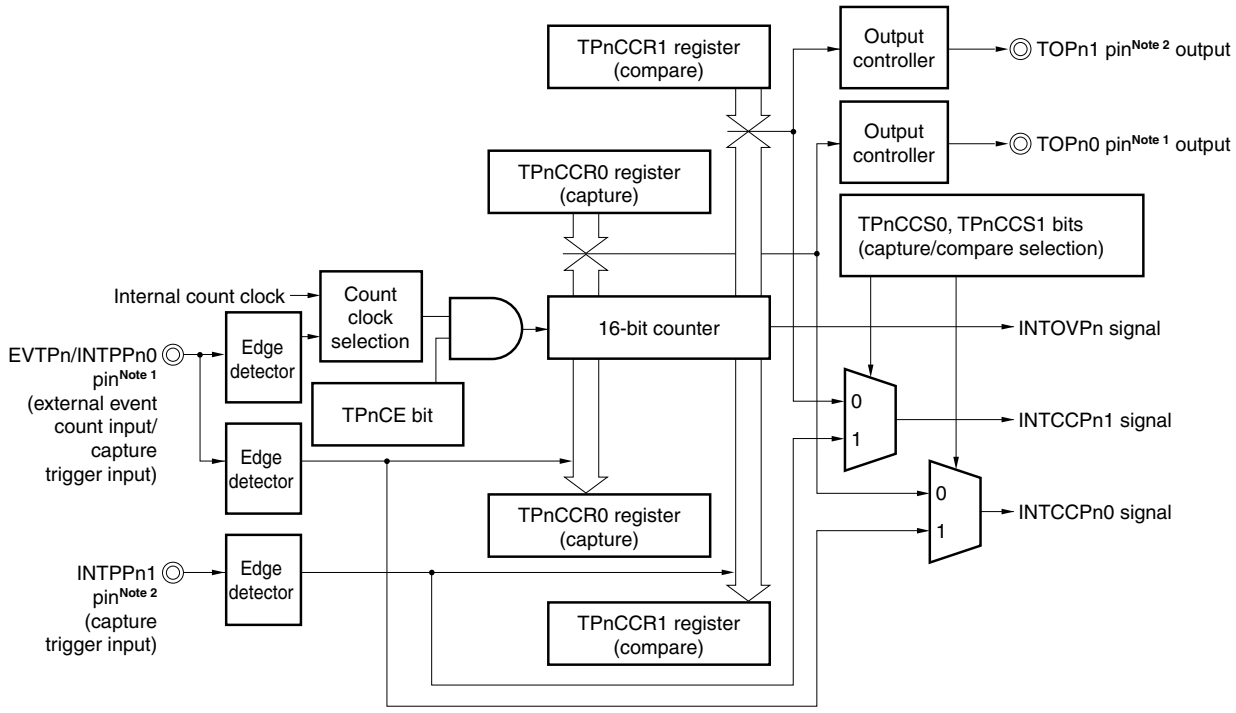
Usually, the INTCCPn1 signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TPnCCR1 register.

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 TOPn1 pin.

8.6.6 Free-running timer mode (TPnMD2 to TPnMD0 bits = 101)

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 TPnCCR0 and TPnCCR1 registers can be used as compare registers or capture registers, depending on the setting of the TPnOPT0.TPnCCS0 and TPnOPT0.TPnCCS1 bits.

Figure 8-32. Configuration in Free-Running Timer Mode



Notes 1. Because the external event count input pin (EVTPn), capture trigger input pin (INTPPn0), and timer output pin (TOPn0) are the same alternate-function pin, two or more functions cannot be used at the same time.

2. Because the capture trigger input pin (INTPPn1) and timer output pin (TOPn1) are the same alternate-function pin, two or more functions cannot be used at the same time.

Remark n = 0 to 2

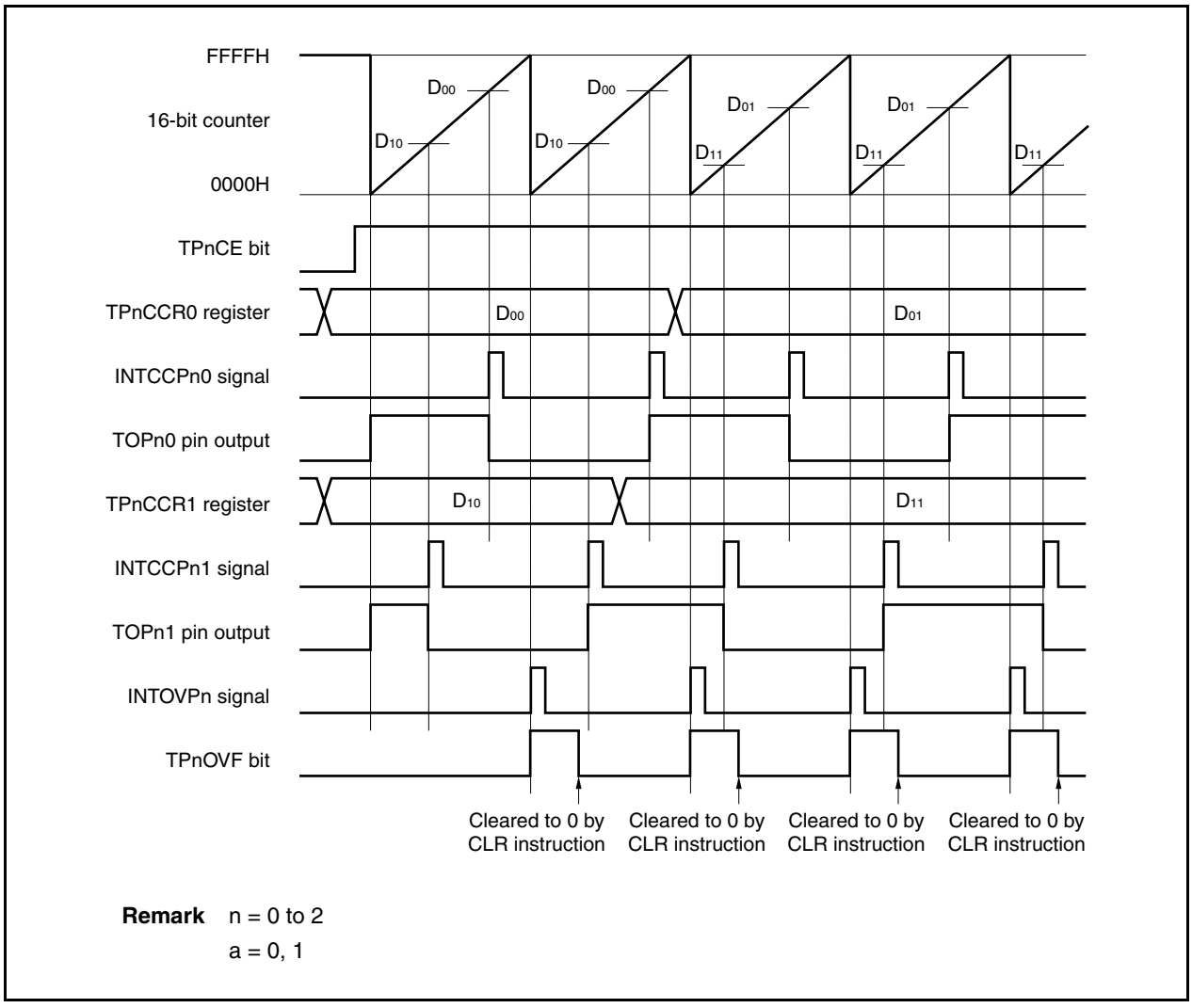
- Compare operation

  When the TPnCE bit is set to 1, 16-bit timer/event counter P starts counting, and the output signals of the TOPn0 and TOPn1 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 (INTCCPna) is generated, and the output signals of the TOPn0 and TOPn1 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 (INTOVPn) 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 8-33. Basic Timing in Free-Running Timer Mode (Compare Function)



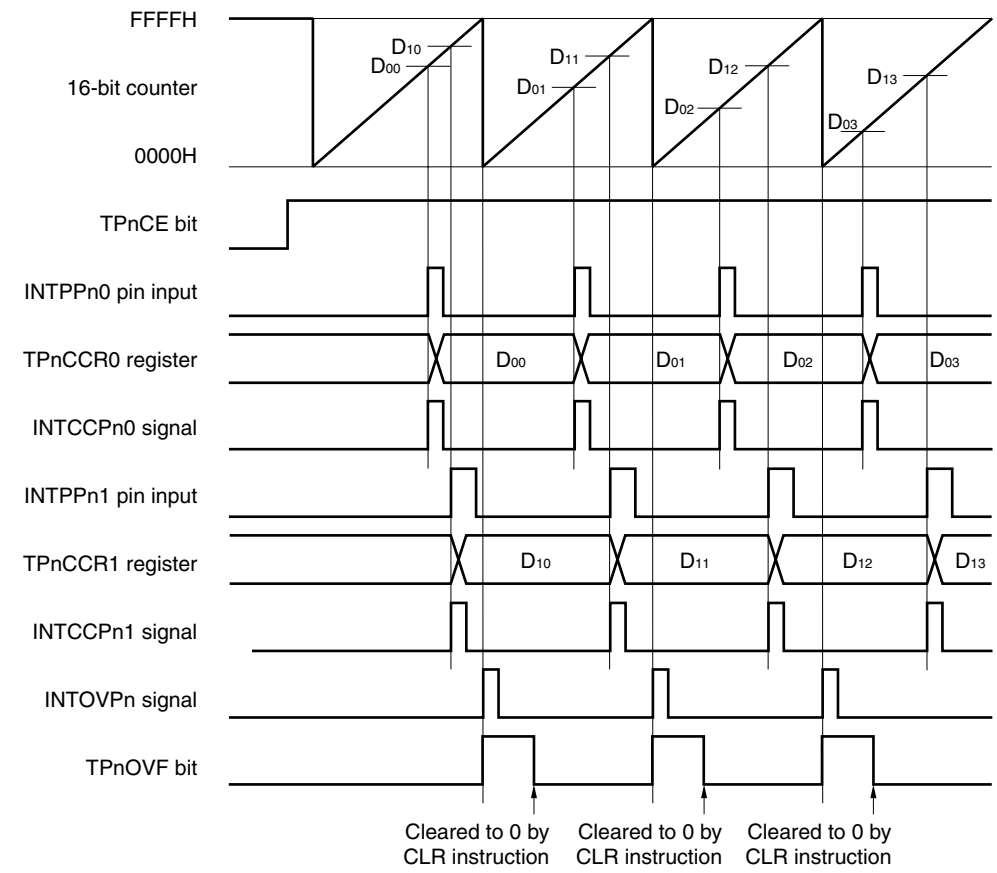
Remark n = 0 to 2
a = 0, 1

- Capture operation

  When the TPnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the INTPPna pin is detected, the count value of the 16-bit counter is stored in the TPnCCRa register, and a capture interrupt request signal (INTCCPna) 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 (INTOVPn) 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.

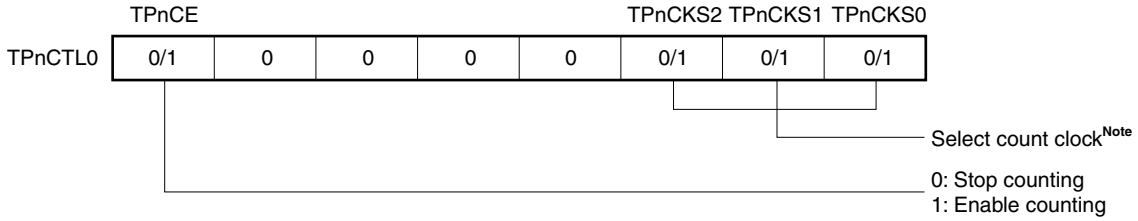
Figure 8-34. Basic Timing in Free-Running Timer Mode (Capture Function)



Remark n = 0 to 2

Figure 8-35. Register Setting in Free-Running Timer Mode (1/2)

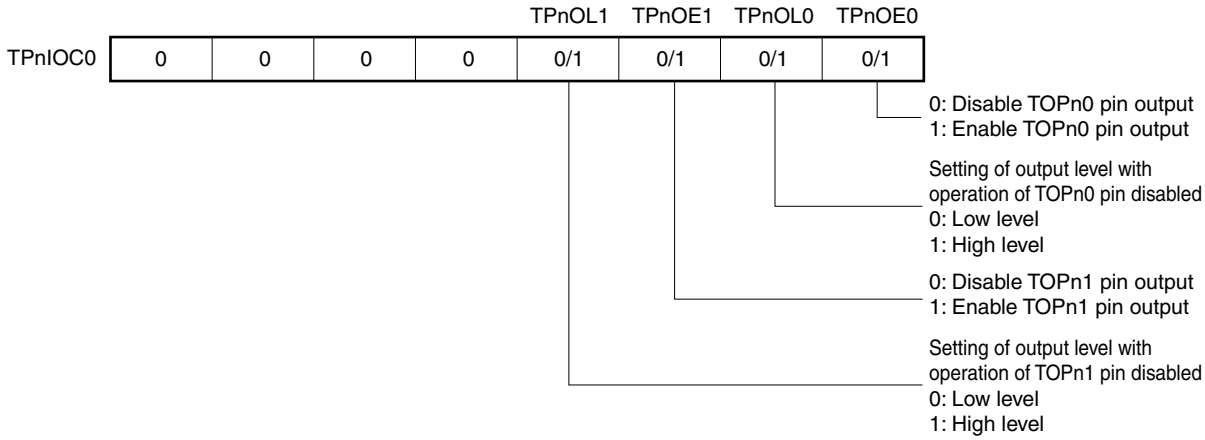
(a) TMPn control register 0 (TPnCTL0)



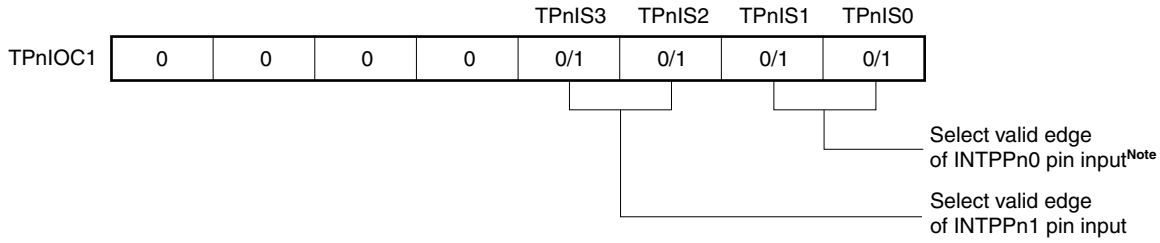
Note The setting is invalid when the TPnCTL1.TPnEEE bit = 1

(b) TMPn control register 1 (TPnCTL1)

(c) TMPn I/O control register 0 (TPnIOC0)



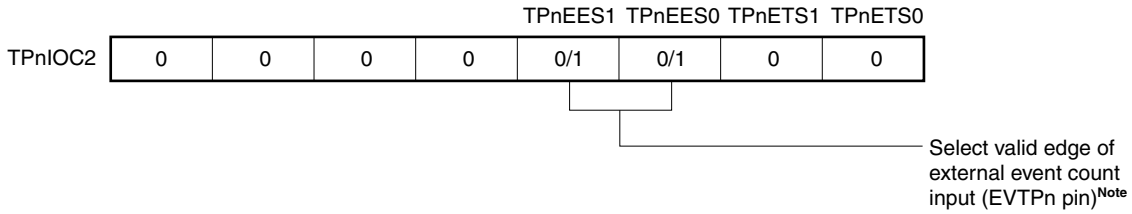
(d) TMPn I/O control register 1 (TPnIOC1)



Note Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

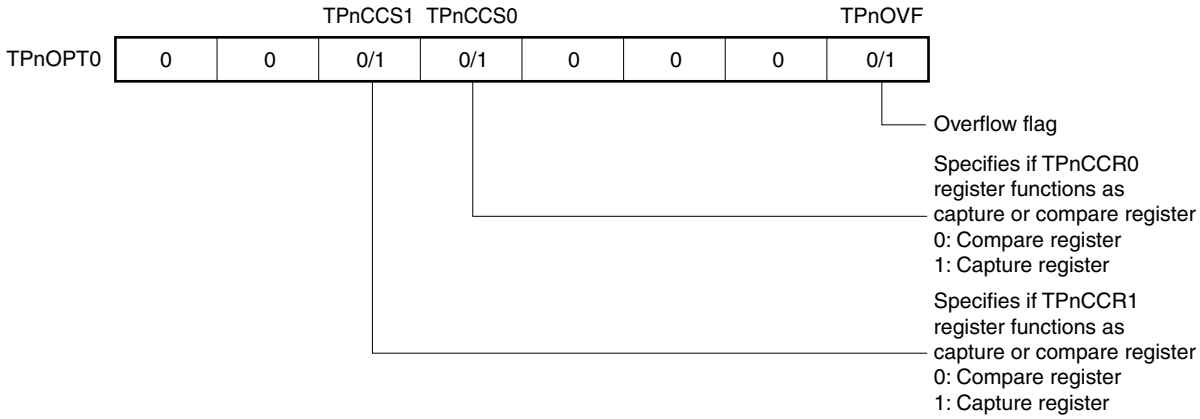
Figure 8-35. Register Setting in Free-Running Timer Mode (2/2)

(e) TMPn I/O control register 2 (TPnIOC2)



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 TPnOPT0.TPnCCSa 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 INTPPna pin is detected.

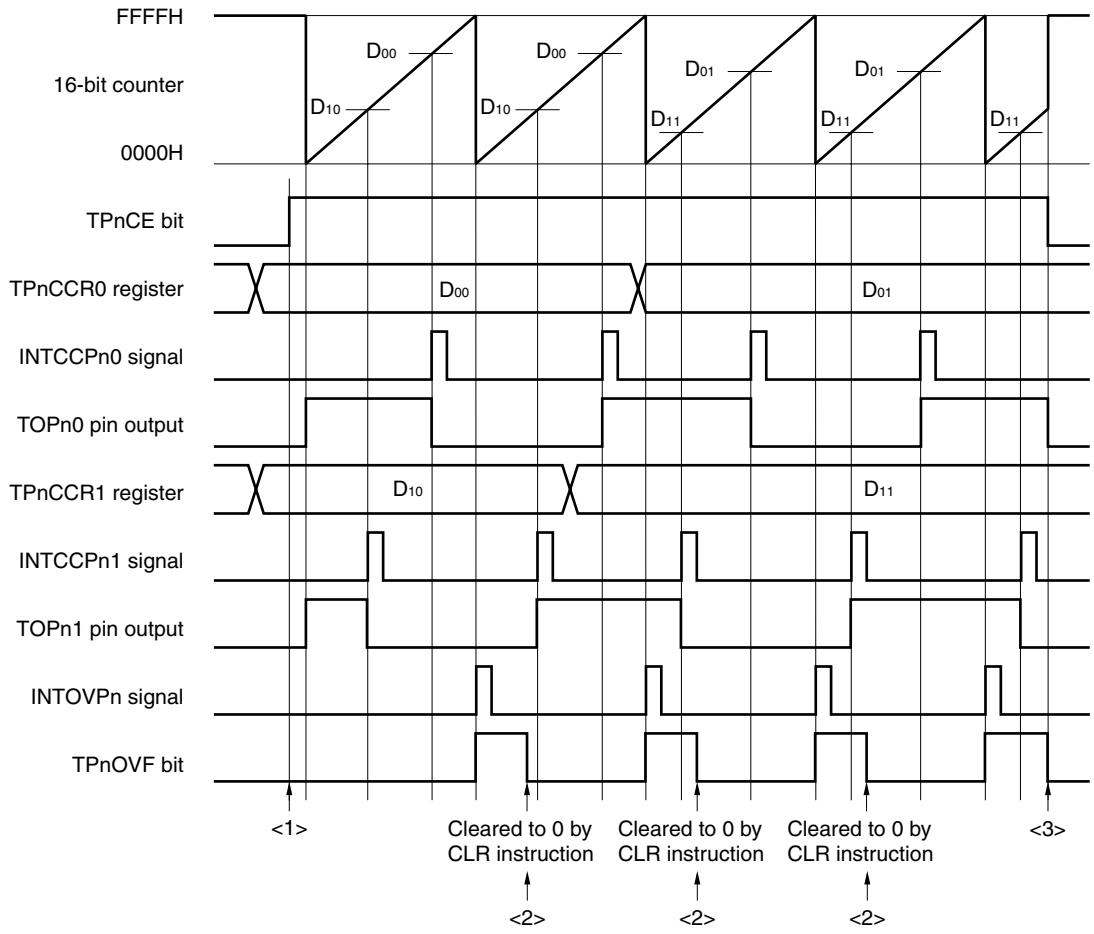
When the registers function as compare registers and when D_a is set to the TPnCCRa register, the INTCCPna signal is generated when the counter reaches (D_a + 1), and the output signals of the TOPn0 and TOPn1 pins are inverted.

Remark n = 0 to 2, a = 0, 1

(1) Operation flow in free-running timer mode

(a) When using capture/compare register as compare register

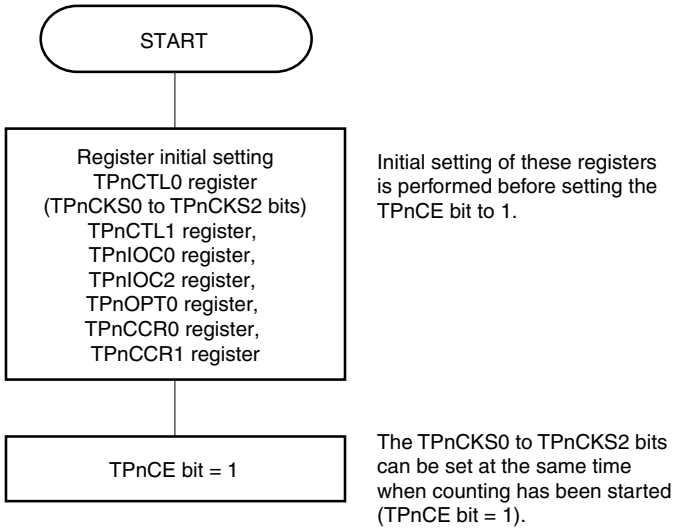
Figure 8-36. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)



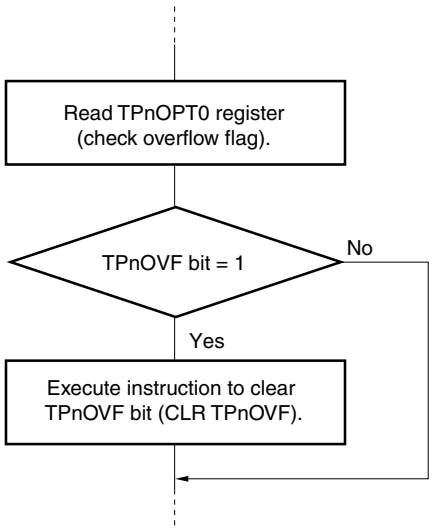
Remark n = 0 to 2

Figure 8-36. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)

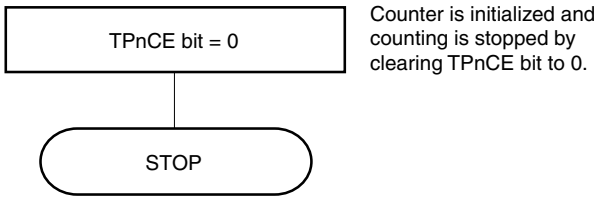
<1> Count operation start flow



<2> Overflow flag clear flow



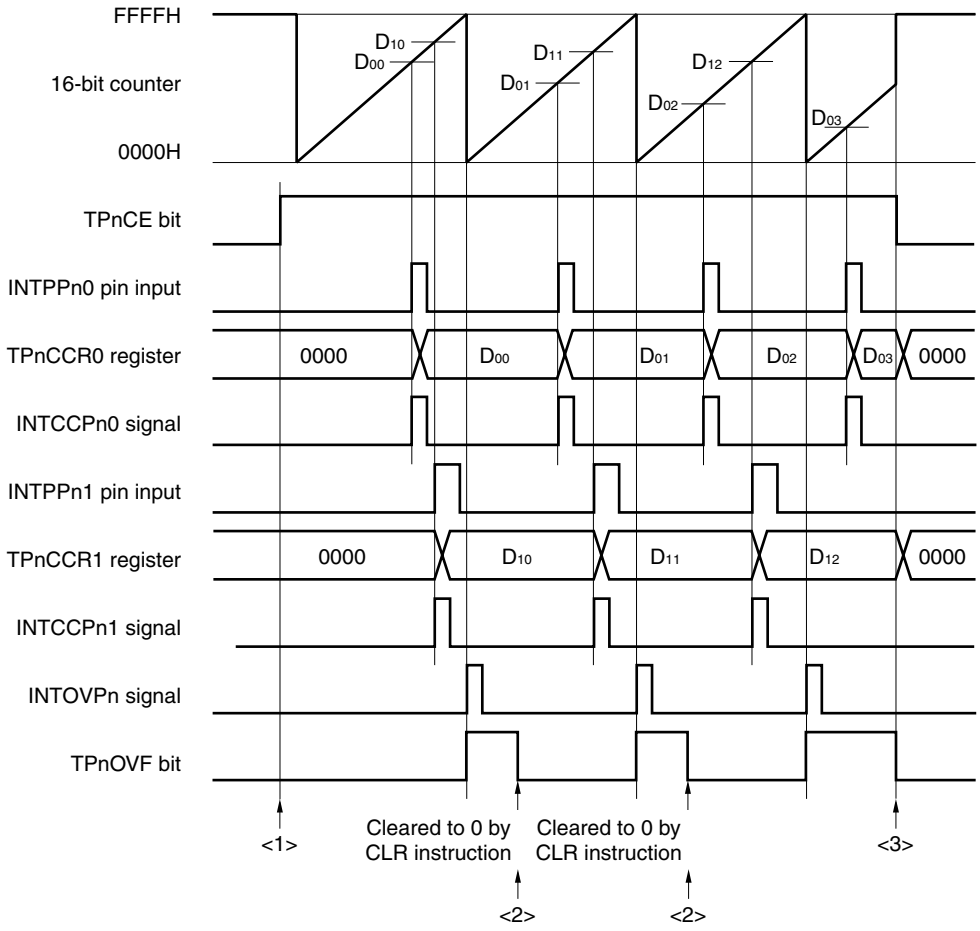
<3> Count operation stop flow



Remark n = 0 to 2

(b) When using capture/compare register as capture register

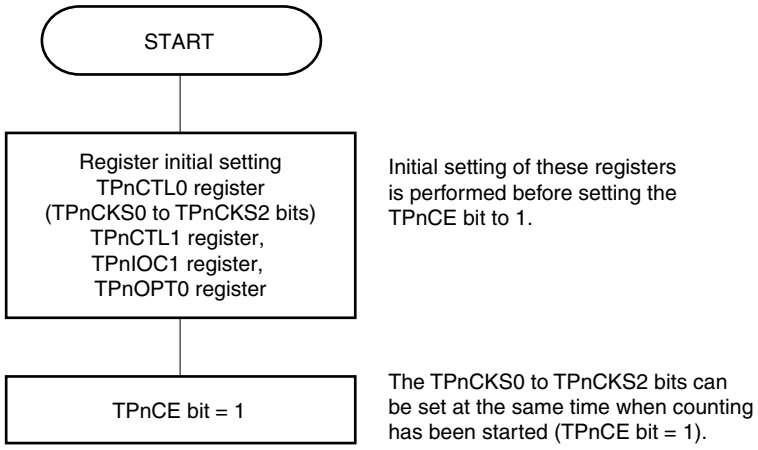
Figure 8-37. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)



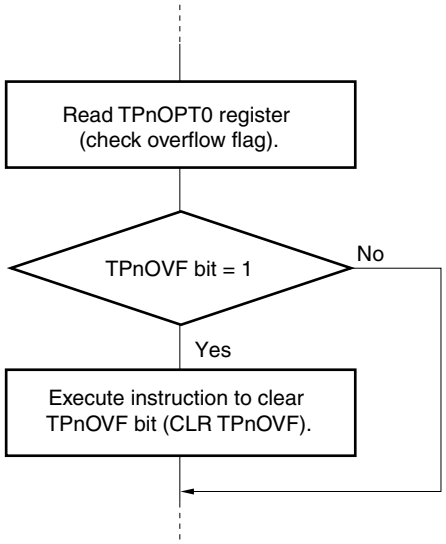
Remark n = 0 to 2

Figure 8-37. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)

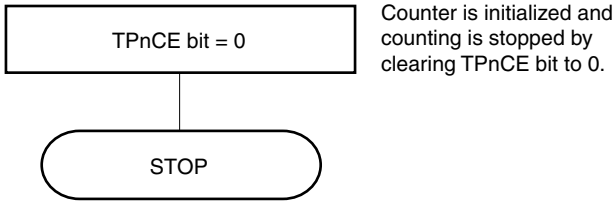
<1> Count operation start flow



<2> Overflow flag clear flow



<3> Count operation stop flow

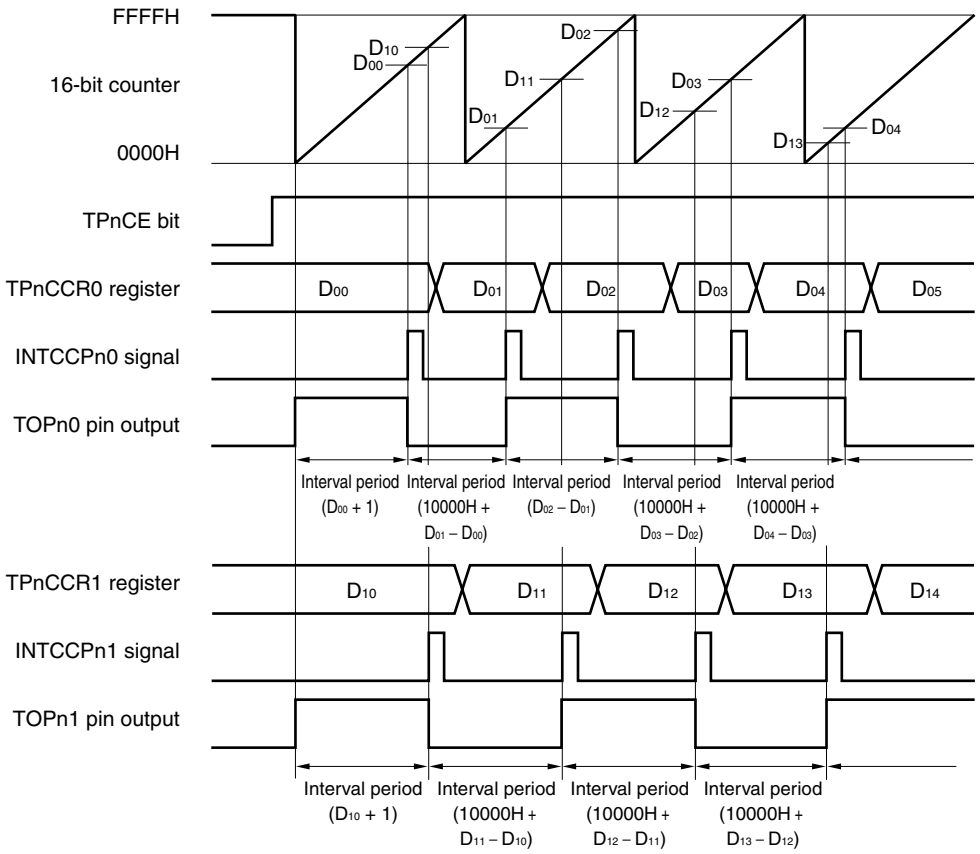


Remark n = 0 to 2

(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 INTCCPna 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 INTCCPna 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

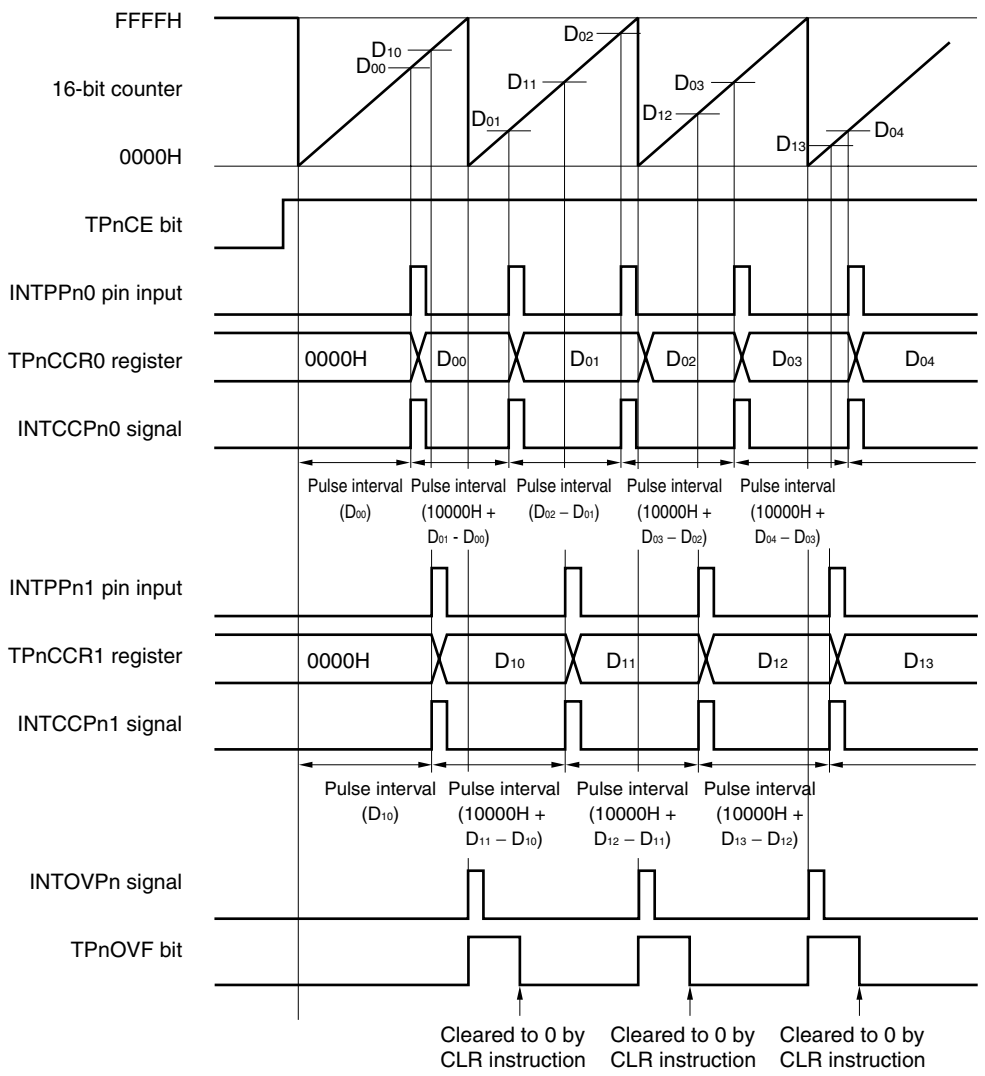
(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

Remark n = 0 to 2

a = 0, 1

(b) Pulse width measurement with capture register

When pulse width measurement is performed with the TPnCCRa register used as a capture register, software processing is necessary for reading the capture register each time the INTCCPna 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.

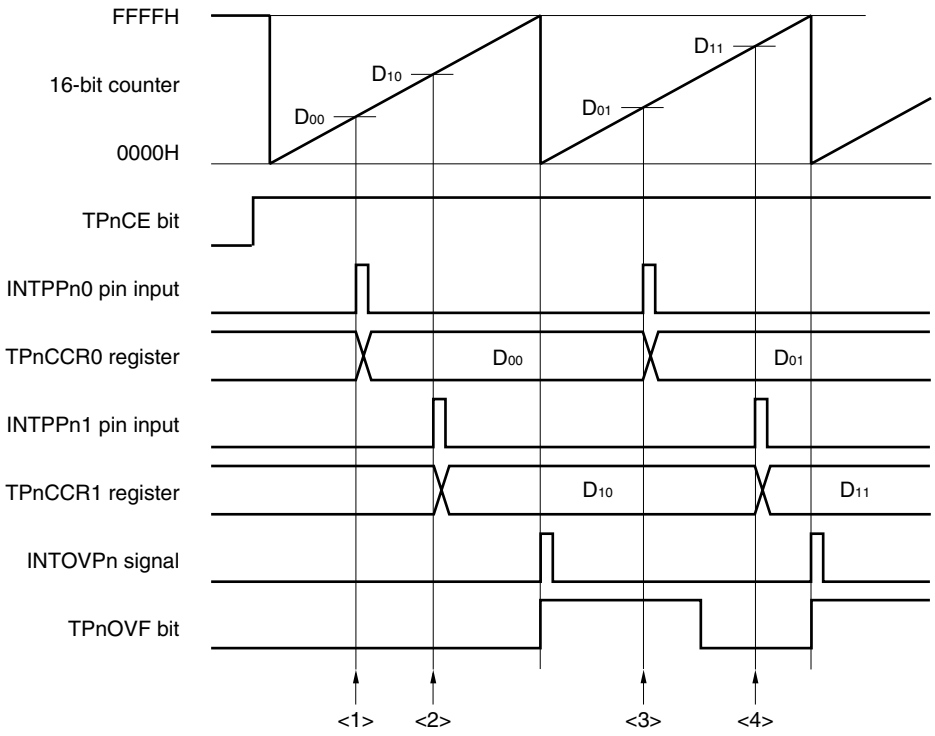
To measure a pulse width, the pulse width can be calculated by reading the value of the TPnCCRa register in synchronization with the INTCCPna signal, and calculating the difference between the read value and the previously read value.

Remark n = 0 to 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.

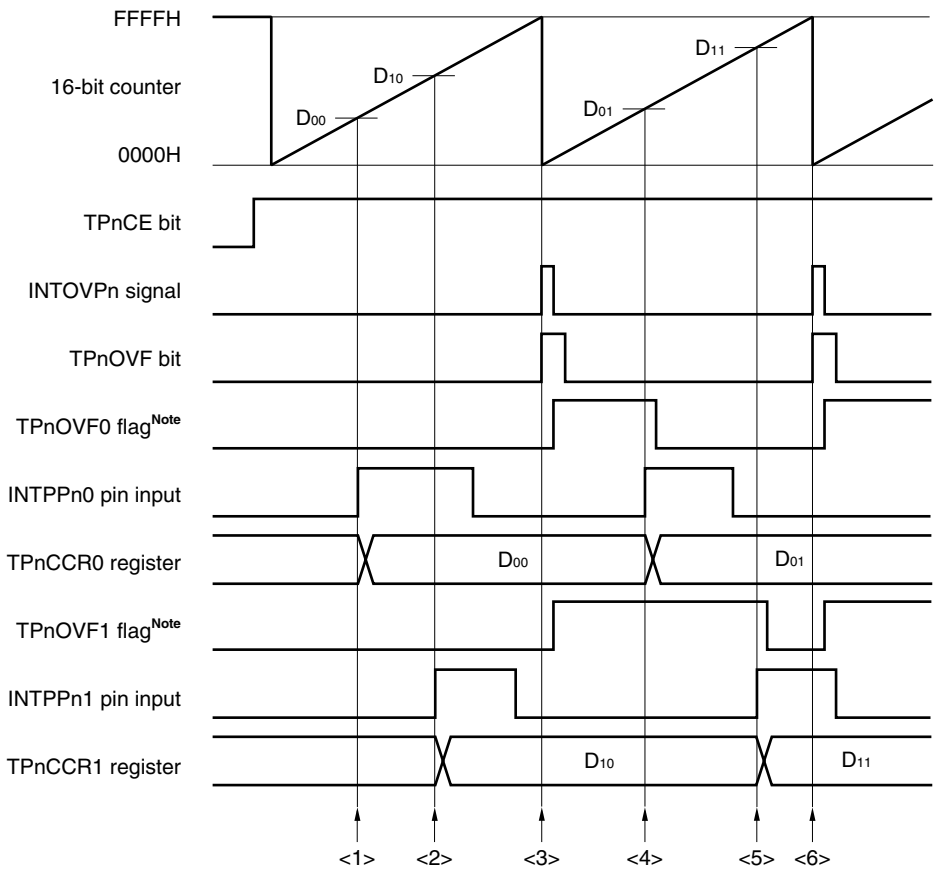
<1> Read the TPnCCR0 register (setting of the default value of the INTPPn0 pin input).
<2> Read the TPnCCR1 register (setting of the default value of the INTPPn1 pin input).
<3> Read the TPnCCR0 register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} − D_{00}).
<4> Read the TPnCCR1 register.
Read the overflow flag. Because the flag is cleared in <3>, 0 is read.
Because the overflow flag is 0, the pulse width can be calculated by (D_{11} − D_{10}) (incorrect).

Remark n = 0 to 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 TPnOVF0 and TPnOVF1 flags are set on the internal RAM by software.

<1> Read the TPnCCR0 register (setting of the default value of the INTPPn0 pin input).

<2> Read the TPnCCR1 register (setting of the default value of the INTPPn1 pin input).

<3> An overflow occurs. Set the TPnOVF0 and TPnOVF1 flags to 1 in the overflow interrupt servicing, and clear the overflow flag to 0.

<4> Read the TPnCCR0 register.
Read the TPnOVF0 flag. If the TPnOVF0 flag is 1, clear it to 0.
Because the TPnOVF0 flag is 1, the pulse width can be calculated by (10000H + D_{01} − D_{00}).

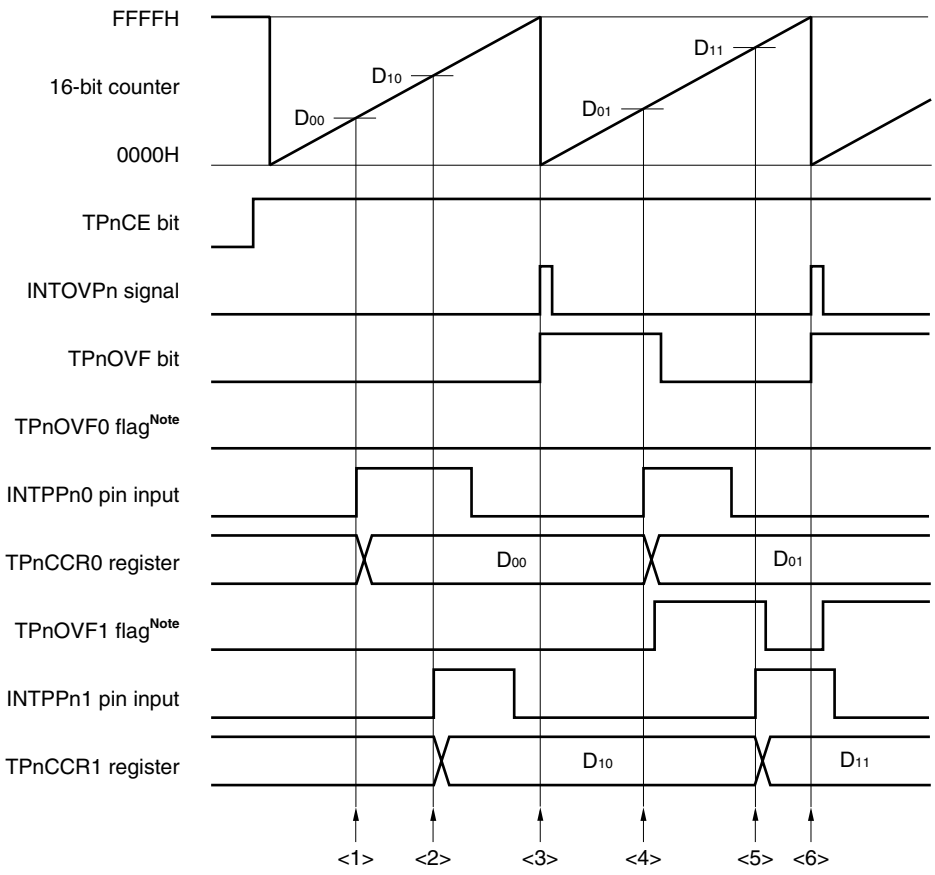
<5> Read the TPnCCR1 register.
Read the TPnOVF1 flag. If the TPnOVF1 flag is 1, clear it to 0 (the TPnOVF0 flag is cleared in <4>, and the TPnOVF1 flag remains 1).
Because the TPnOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} − D_{10}) (correct).

<6> Same as <3>

Remark n = 0 to 2

(2/2)

Example when two capture registers are used (without using overflow interrupt)



Note The TPnOVF0 and TPnOVF1 flags are set on the internal RAM by software.

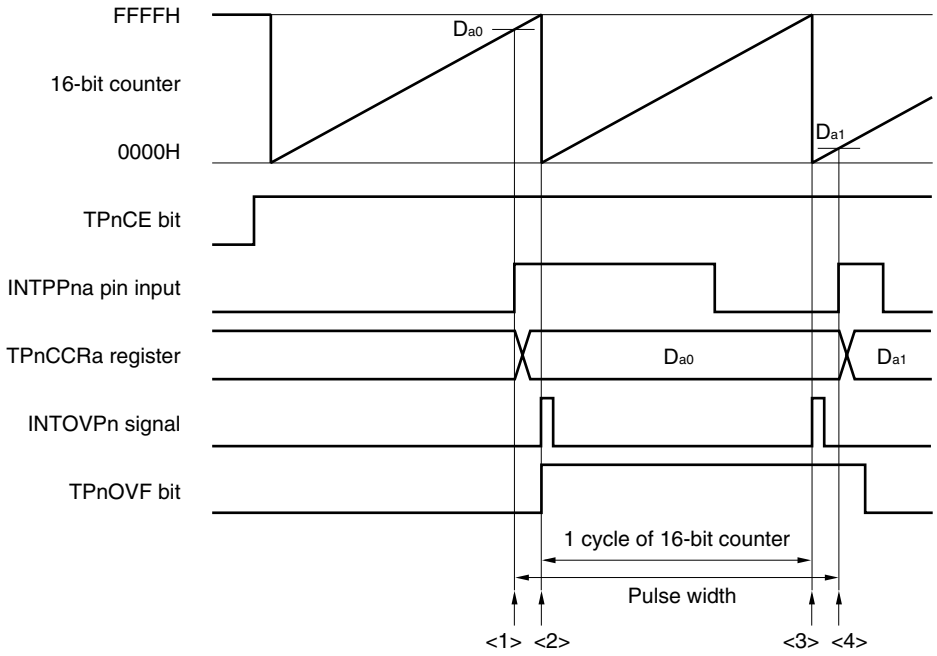
<1> Read the TPnCCR0 register (setting of the default value of the INTPPn0 pin input).
<2> Read the TPnCCR1 register (setting of the default value of the INTPPn1 pin input).
<3> An overflow occurs. Nothing is done by software.
<4> Read the TPnCCR0 register.
Read the overflow flag. If the overflow flag is 1, set only the TPnOVF1 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 TPnCCR1 register.
Read the overflow flag. Because the overflow flag is cleared in <4>, 0 is read.
Read the TPnOVF1 flag. If the TPnOVF1 flag is 1, clear it to 0.
Because the TPnOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} − D_{10}) (correct).
<6> Same as <3>

Remark n = 0 to 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



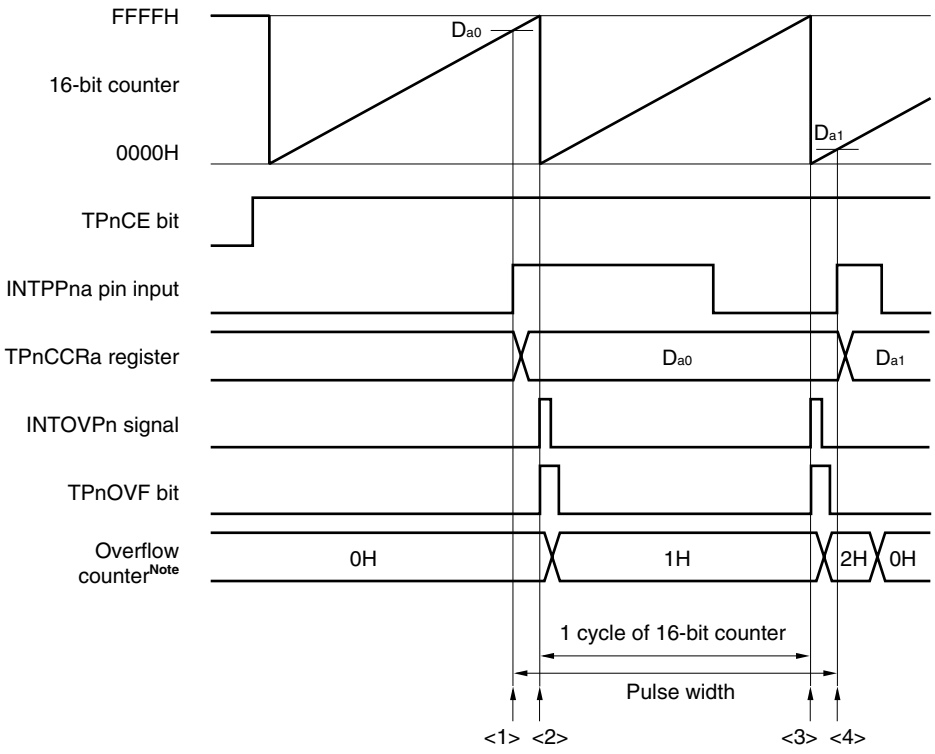
The following problem may occur when long pulse width is measured in the free-running timer mode.

<1> Read the TPnCCRa register (setting of the default value of the INTPPna 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 TPnCCRa register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + $D_{a1} - D_{a0}$) (incorrect).
Actually, the pulse width must be (20000H + $D_{a1} - D_{a0}$) because an overflow occurs twice.

Remark n = 0 to 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 TPnCCRa register (setting of the default value of the INTPPna 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 TPnCCRa register.

Read the overflow counter.

→ When the overflow counter is “N”, the pulse width can be calculated by ($N \times 10000H + D_{a1} - D_{a0}$).

In this example, the pulse width is ($20000H + D_{a1} - D_{a0}$) because an overflow occurs twice.

Clear the overflow counter (0H).

Remark n = 0 to 2

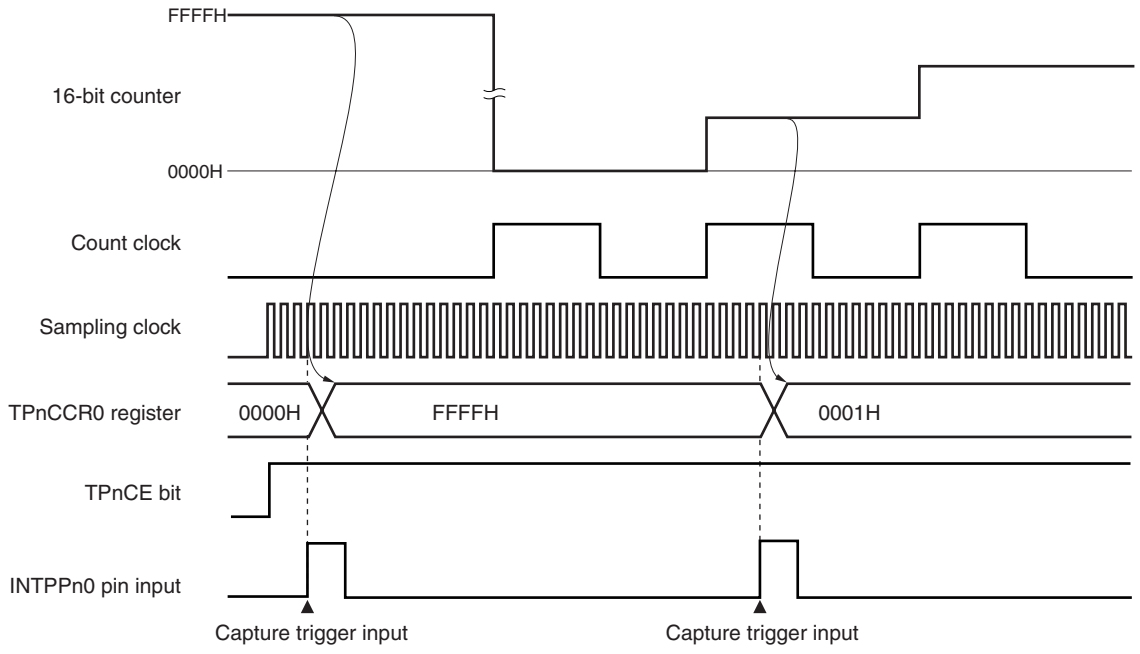
a = 0, 1

(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 TPnCCRa register if the capture trigger is input immediately after the TPnCTL0.TPnCE bit is set to 1.



Remark n = 0 to 2
a = 0, 1

8.6.7 Pulse width measurement mode (TPnMD2 to TPnMD0 bits = 110)

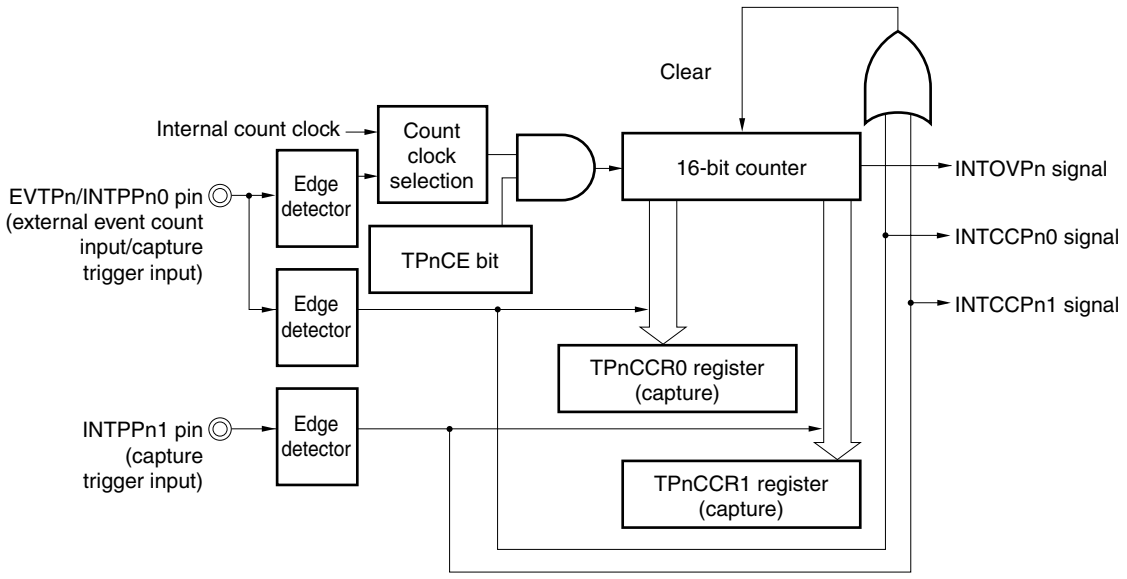
In the pulse width measurement mode, 16-bit timer/event counter P starts counting when the TPnCTL0.TPnCE bit is set to 1. Each time the valid edge input to the INTPPna pin has been detected, the count value of the 16-bit counter is stored in the TPnCCRa register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TPnCCRa register after a capture interrupt request signal (INTCCPna) occurs.

As shown in Figure 8-39, select either the INTPPn0 or INTPPn1 pin as the capture trigger input pin and set the unused pins to “No edge detection” by using the TPnIOC1 register.

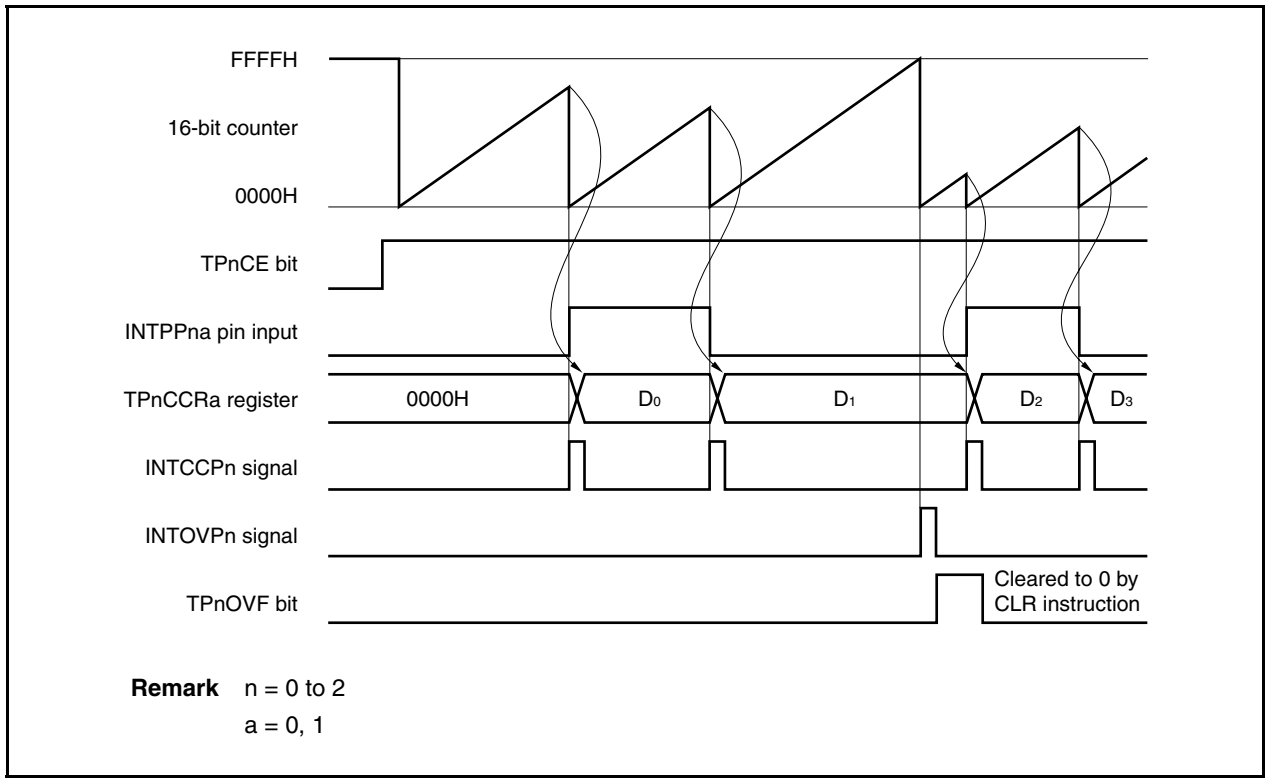
When the external event count input signal (EVTPn pin) is used as the count clock, measure the pulse width of the INTPPn1 pin because the EVTPn pin functions alternately as a capture trigger input signal (INTPPn0 pin). At this time, clear the TPnIOC1.TPnIS1 and TPnIOC1.TPnIS0 bits to 00 (capture trigger input (INTPPn0 pin): No edge detection).

Figure 8-38. Configuration in Pulse Width Measurement Mode



Remark n = 0 to 2
a = 0, 1

Figure 8-39. Basic Timing in Pulse Width Measurement Mode



Remark n = 0 to 2
a = 0, 1

When the TPnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the INTPPna pin is later detected, the count value of the 16-bit counter is stored in the TPnCCRa register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTCCPna) is generated.

The pulse width is calculated as follows.

<R> Pulse width = $D_N \times$ 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 (INTOVPn) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TPnOPT0.TPnOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction via software.

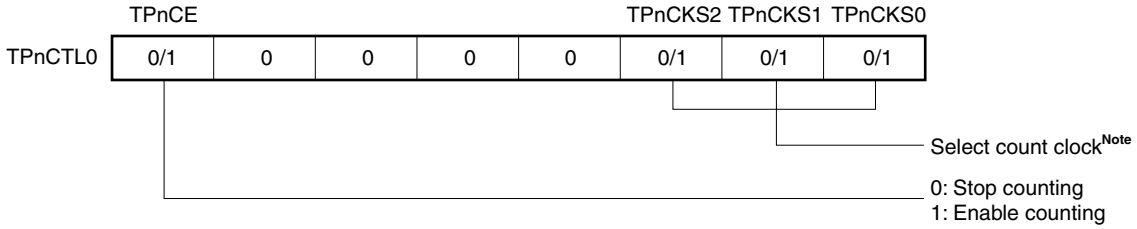
If the overflow flag is set to 1, the pulse width can be calculated as follows.

<R> Pulse width = (10000H $\times$ TPnOVF bit set (1) count + D_N) $\times$ Count clock cycle

Remark n = 0 to 2
a = 0, 1

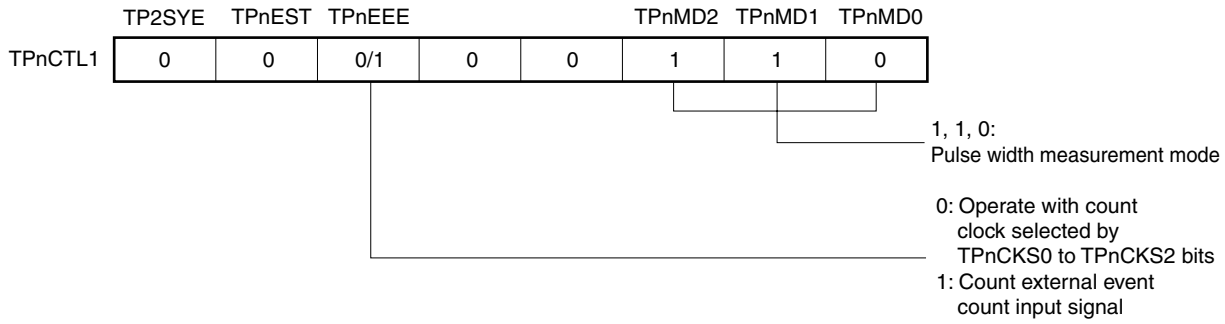
Figure 8-40. Register Setting in Pulse Width Measurement Mode (1/2)

(a) TMPn control register 0 (TPnCTL0)

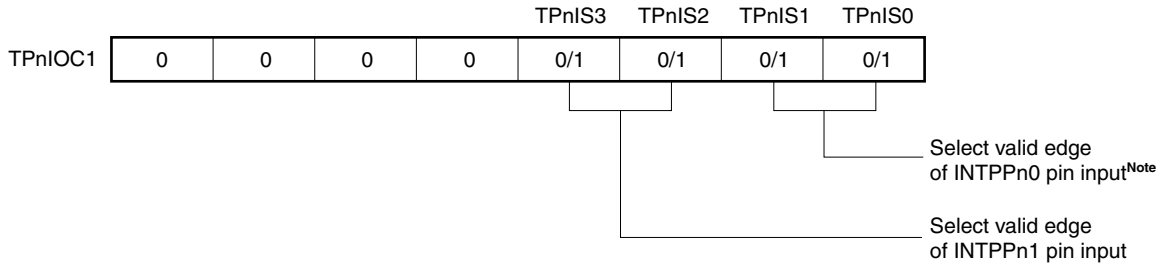


Note Setting is invalid when the TPnCTL1.TPnEEE bit = 1.

(b) TMPn control register 1 (TPnCTL1)

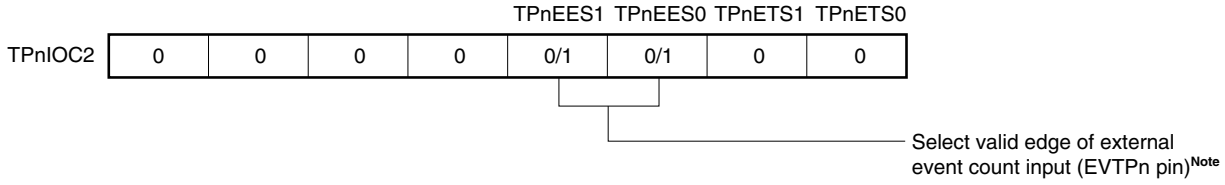


(c) TMPn I/O control register 1 (TPnIOC1)



Note Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

(d) TMPn I/O control register 2 (TPnIOC2)



Note Set the valid edge selection of the unused alternate external input signals to “No edge detection”.

Figure 8-40. Register Setting in Pulse Width Measurement Mode (2/2)

(e) TMPn option register 0 (TPnOPT0)

			TPnCCS1	TPnCCS0				TPnOVF
TPnOPT0	0	0	0	0	0	0	0	0/1

Overflow flag

(f) TMPn counter read buffer register (TPnCNT)

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

(g) TMPn capture/compare registers 0 and 1 (TPnCCR0 and TPnCCR1)

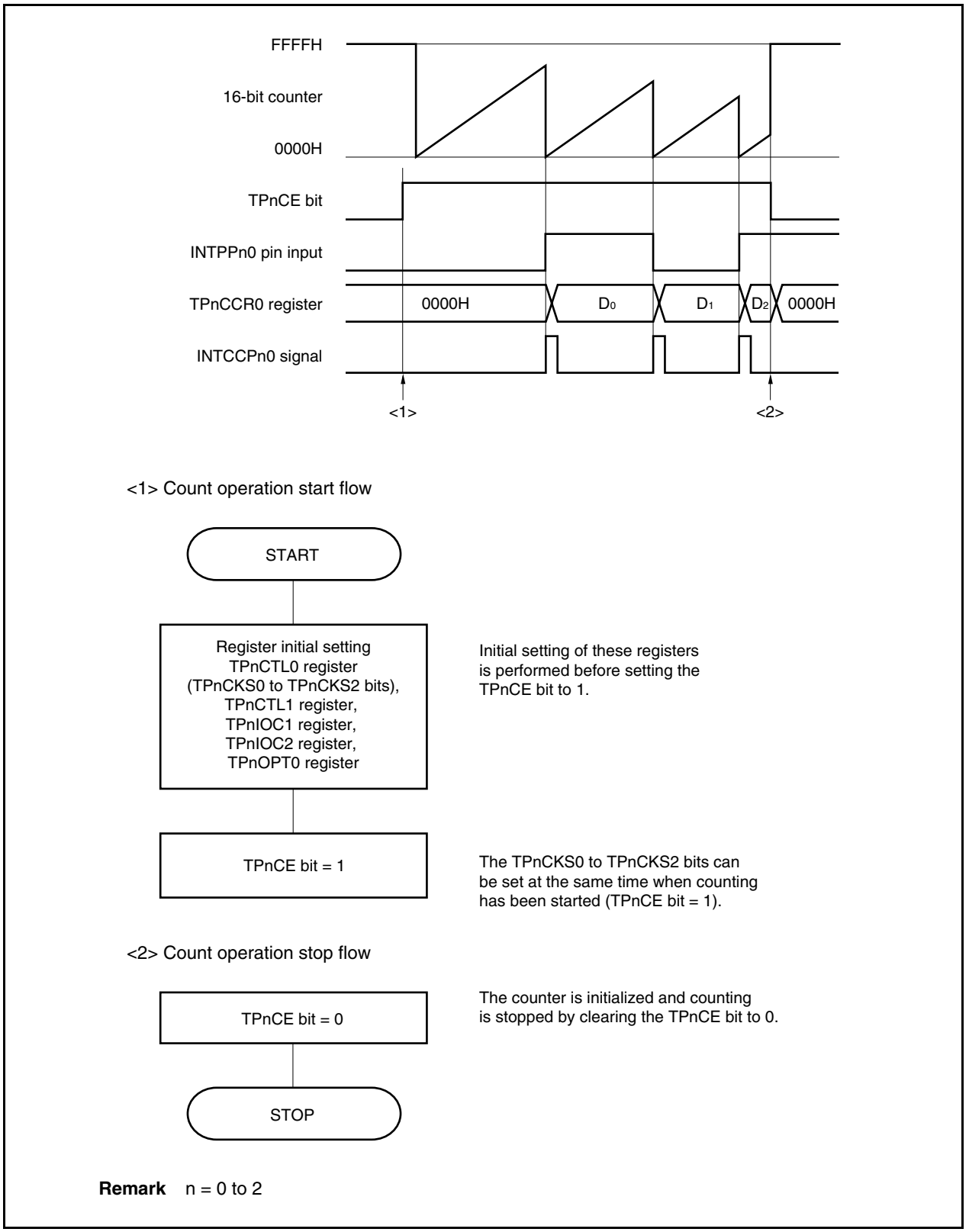
These registers store the count value of the 16-bit counter when the valid edge input to the INTPPn0 and INTPPn1 pins is detected.

Remarks 1. TMPn I/O control register 0 (TPnIOC0) is not used in the pulse width measurement mode.

2. n = 0 to 2

(1) Operation flow in pulse width measurement mode

Figure 8-41. Software Processing Flow in Pulse Width Measurement Mode



Remark n = 0 to 2

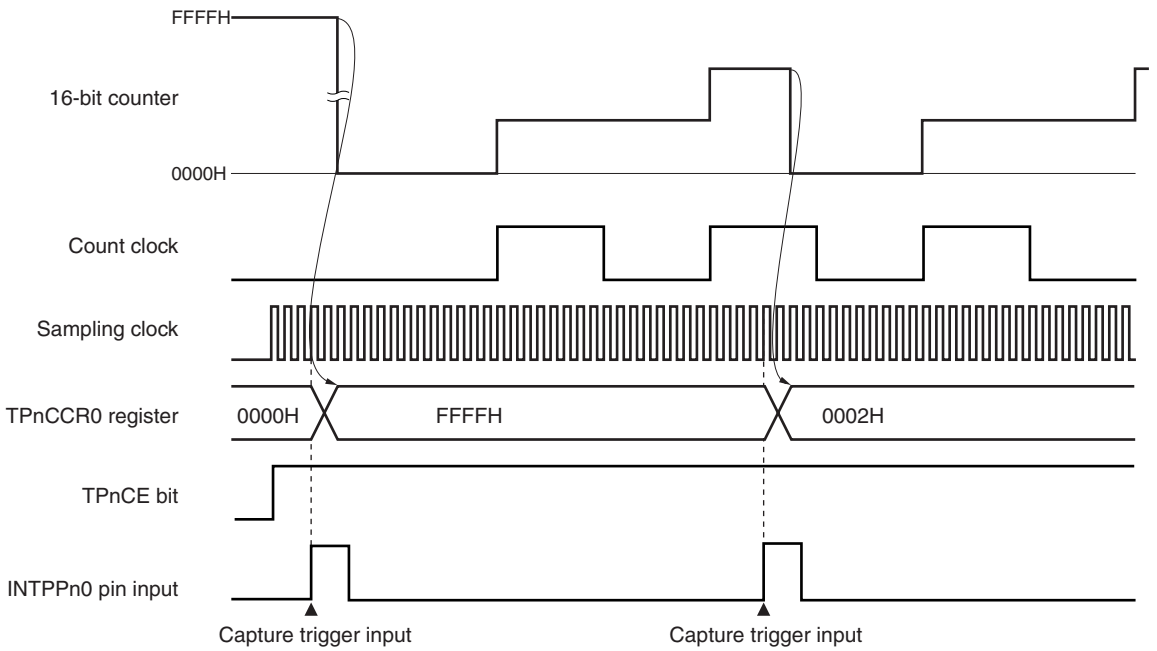
(2) Operation timing in pulse width measurement mode

(a) 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) Notes on capture operation

If a slow clock is selected as the count clock, FFFFH, not 0000H, may be captured to the TPnCCRa register if the capture trigger is input immediately after the TPnCTL0.TPnCE bit has been set to 1.



Remark n = 0 to 2
a = 0, 1

CHAPTER 9 16-BIT TIMER/EVENT COUNTER Q (TMQ)

Timer Q (TMQ) is a 16-bit timer/event counter.
The V850E/MA3 incorporates TMQ0.

9.1 Overview

An outline of TMQ0 is shown below.

- Clock selection: 8 ways
- Capture/trigger input pins: 4
- External event count input pins: 1
- External trigger input pins: 1
- Timer/counters: 1
- Capture/compare registers: 4
- Capture/compare match interrupt request signals: 4
- Timer output pins[Note]: 4

Note This is the number of output pins of TMQ0; it does not include the output pins of TMQOP0. For details of the output pins of TMQOP0, see **CHAPTER 12 MOTOR CONTROL FUNCTION**.

9.2 Functions

TMQ0 has the following functions.

- 6-phase PWM output[Note]
- Interval timer
- External event counter
- External trigger pulse output
- One-shot pulse output
- PWM output
- Free-running timer
- Pulse width measurement

Note This is connected to TMQOP0. For details, see **CHAPTER 12 MOTOR CONTROL FUNCTION**.

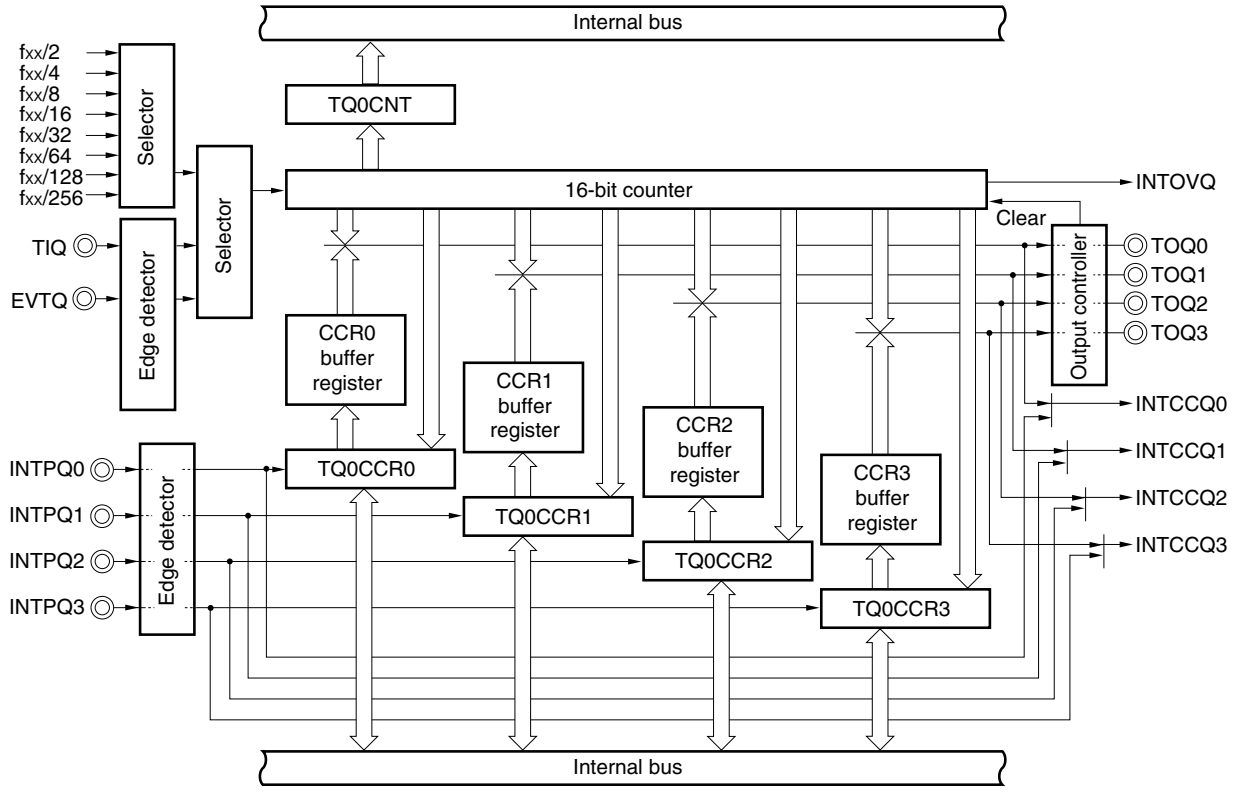
9.3 Configuration

TMQ0 includes the following hardware.

Table 9-1. TMQ0 Configuration

Item	Configuration
Timer register	16-bit counter
Registers	TMQ0 counter read buffer register (TQ0CNT) TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3) CCR0 to CCR3 buffer registers
Timer input	Total of 6 (TIQ, EVTQ, INTPQ0 to INTPQ3 pins)
Timer output	Total of 4 (TOQ0 to TOQ3 pins)
Control registers	TMQ0 control registers 0, 1 (TQ0CTL0, TQ0CTL1) TMQ0 I/O control registers 0 to 2 (TQ0IOC0 to TQ0IOC2) TMQ0 option register 0 (TQ0OPT0)

Figure 9-1. TMQ0 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 TQ0CNT register.

When the TQ0CTL0.TQ0CE bit = 0, the value of the 16-bit counter is FFFFH. If the TQ0CNT register is read at this time, 0000H is read.

Reset input clears the TQ0CE bit to 0.

(2) CCR0 buffer register

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

When the TQ0CCR0 register is used as a compare register, the value written to 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, a compare match interrupt request signal (INTCCQ0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is cleared to 0000H after reset, and the TQ0CCR0 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 TQ0CCR1 register is used as a compare register, the value written to the TQ0CCR1 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 (INTCCQ1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is cleared to 0000H after reset, and the TQ0CCR1 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 TQ0CCR2 register is used as a compare register, the value written to the TQ0CCR2 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 (INTCCQ2) is generated.

The CCR2 buffer register cannot be read or written directly.

The CCR2 buffer register is cleared to 0000H after reset, and the TQ0CCR2 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 TQ0CCR3 register is used as a compare register, the value written to the TQ0CCR3 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 (INTCCQ3) is generated.

The CCR3 buffer register cannot be read or written directly.

The CCR3 buffer register is cleared to 0000H after reset, and the TQ0CCR3 register is cleared to 0000H.

(6) Edge detector

This circuit detects the valid edges input to the TIQ, EVTQ, and INTPQ0 to INTPQ3 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 TOQ0 to TOQ3 pins. The output controller is controlled by the TQ0IOC0 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.

9.4 Registers

(1) TMQ0 control register 0 (TQ0CTL0)

The TQ0CTL0 register is an 8-bit register that controls the operation of TMQ0.

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

Reset input clears this register to 00H.

The same value can always be written to the TQ0CTL0 register by software.

After reset: 00H R/W Address: FFFFF600H

	<7>	6	5	4	3	2	1	0
TQ0CTL0	TQ0CE	0	0	0	0	TQ0CKS2	TQ0CKS1	TQ0CKS0

TQ0CE	TMQ0 operation control
0	TMQ0 operation disabled (TMQ0 reset asynchronously[Note]).
1	TMQ0 operation enabled. TMQ0 operation started.

TQ0CKS2	TQ0CKS1	TQ0CKS0	Internal count clock selection
0	0	0	$f_{XX}/2$
0	0	1	$f_{XX}/4$
0	1	0	$f_{XX}/8$
0	1	1	$f_{XX}/16$
1	0	0	$f_{XX}/32$
1	0	1	$f_{XX}/64$
1	1	0	$f_{XX}/128$
1	1	1	$f_{XX}/256$

<R> **Note** The TPnOPT0.TPnOVF bit and the 16-bit counter are reset simultaneously. Moreover, timer outputs (TOQn0 to TOQn3 pins) are reset to the TQ0IOC0 register set status at the same time as the 16-bit counter.

Cautions 1. Set the TQ0CKS2 to TQ0CKS0 bits when the TQ0CE bit = 0.
When the value of the TQ0CE bit is changed from 0 to 1, the TQ0CKS2 to TQ0CKS0 bits can be set simultaneously.
2. Be sure to set bits 3 to 6 to "0".

Remark f_{XX}: Peripheral clock

(2) TMQ0 control register 1 (TQ0CTL1)

The TQ0CTL1 register is an 8-bit register that controls the operation of TMQ0.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF601H

	7	6	5	4	3	2	1	0
TQ0CTL1	0	TQ0EST	TQ0EEE	0	0	TQ0MD2	TQ0MD1	TQ0MD0

TQ0EST	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	Count clock selection
0	Disable operation with external event count input (EVTQ pin). (Perform counting with the count clock selected by the TQ0CTL0.TQ0CKS0 to TQ0CKS2 bits.)
1	Enable operation with external event count input (EVTQ 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.	

TQ0MD2	TQ0MD1	TQ0MD0	Timer mode selection
0	0	0	Interval timer mode
0	0	1	External event count mode
0	1	0	External trigger pulse output mode
0	1	1	One-shot pulse output mode
1	0	0	PWM output mode
1	0	1	Free-running timer mode
1	1	0	Pulse width measurement mode
1	1	1	6-phase PWM output mode[Note]

Note The 6-phase PWM output mode cannot be used when only TMQ0 is used. For details, see **CHAPTER 12 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 TQ0EEE and TQ0MD2 to TQ0MD0 bits when the TQ0CTL0.TQ0CE bit = 0. (The same value can be written when the TQ0CE bit = 1.) The operation is not guaranteed when rewriting is performed with the TQ0CE bit = 1. If rewriting was mistakenly performed, clear the TQ0CE bit to 0 and then set the bits again.

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

(3) TMQ0 I/O control register 0 (TQ0IOC0)

The TQ0IOC0 register is an 8-bit register that controls the timer output (TOQ0 to TOQ3, TOQT1 to TOQT3 pins).

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF602H

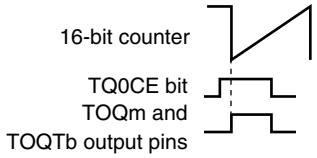
	7	<6>	5	<4>	3	<2>	1	<0>
TQ0IOC0	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0

TQ0OLm	TOQm, TOQTb pin output level setting[Note] (m = 0 to 3, b = 1 to 3)
0	TOQm, TOQTb pin starts output at high level.
1	TOQm, TOQTb pin starts output at low level.

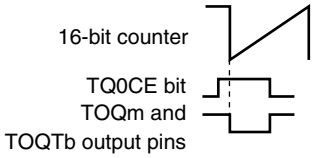
TQ0OEm	TOQm, TOQTb pin output setting (m = 0 to 3, b = 1 to 3)
0	Timer output disabled • When TQ0OLm bit = 0: Low level is output from the TOQm, TOQTb pin • When TQ0OLm bit = 1: High level is output from the TOQm, TOQTb pin
1	Timer output enabled (A pulse is output from the TOQm, TOQTb pin).

Note The output level of the timer output pins (TOQm and TOQTb) specified by the TQ0OLm bit is shown below.

• When TQ0OLm bit = 0



• When TQ0OLm bit = 1



<R> **Cautions 1. If the setting of the TQ0IOC0 register is changed when TOQm and TOQTb 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 TQ0OLm and TQ0OEm 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.

3. If the TQ0OLm bit is manipulated when the TQ0CE and TQ0OEm bits are 0, the output level of the TOQm and TOQTb pins changes.

4. To generate the TOQTb pin output and the A/D conversion start trigger signal of A/D converter in the 6-phase PWM output mode, be sure to set the TOQTb pin output using the TQ0IOC0 register. At this time, be sure to clear the TQ0OL0 bit to 0 and set the TQ0OE0 bit to 1 (b = 1 to 3).

Remark m = 0 to 3

(4) TMQ0 I/O control register 1 (TQ0IOC1)

The TQ0IOC1 register is an 8-bit register that controls the valid edge of the capture trigger input signals (INTPQ0 to INTPQ3 pins).

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF603H

	7	6	5	4	3	2	1	0
TQ0IOC1	TQ0IS7	TQ0IS6	TQ0IS5	TQ0IS4	TQ0IS3	TQ0IS2	TQ0IS1	TQ0IS0

TQ0IS7	TQ0IS6	Capture trigger input signal (INTPQ3 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 (INTPQ2 pin) valid edge detection
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TQ0IS3	TQ0IS2	Capture trigger input signal (INTPQ1 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 (INTPQ0 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 the TQ0CE bit to 0 and then set the bits again.

<R> **2. The TQ0IS7 to TQ0IS0 bits are valid only in the free-running timer mode (only when the TQ0OPT0.TQ0CCSm bit = 1) and the pulse width measurement mode. In all other modes, a capture operation is not possible.**

(5) TMQ0 I/O control register 2 (TQ0IOC2)

The TQ0IOC2 register is an 8-bit register that controls the valid edge of the external event count input signal (TIQ pin) and external trigger input signal (EVTQ pin).

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF604H

	7	6	5	4	3	2	1	0
TQ0IOC2	0	0	0	0	TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0

TQ0EES1	TQ0EES0	External event count input signal (EVTQ 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

TQ0ETS1	TQ0ETS0	External trigger input signal (TIQ 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 TQ0EES1, TQ0EES0, TQ0ETS1, and TQ0ETS0 bits when the TQ0CTL0.TQ0CE bit = 0. (The same value can be written when the TQ0CE bit = 1.) If rewriting was mistakenly performed, clear the TQ0CE bit to 0 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 (TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 bits = 001) has been set.

3. The TQ0ETS1 and TQ0ETS0 bits are valid only in the external trigger pulse output mode or one-shot pulse output mode.

(6) TMQ0 option register 0 (TQ0OPT0)

The TQ0OPT0 register is an 8-bit register used to set the capture/compare operation and detect an overflow.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF605H

	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
TQ0OPT0	TQ0CCS3	TQ0CCS2	TQ0CCS1	TQ0CCS0	0	TQ0CMS Note	TQ0CUF Note	TQ0OVF

<R>

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.	

TQ0OVF	TMQ0 overflow flag
Set (1)	Overflow occurred
Reset (0)	TQ0OVF bit 0 written or TQ0CTL0.TQ0CE bit = 0
• The TQ0OVF bit is set to 1 when the 16-bit counter count value overflows from FFFFH to 0000H in the free-running timer mode or the pulse width measurement mode. • An overflow interrupt request signal (INTOVQ) is generated at the same time that theTQ0OVF bit is set to 1. The INTOVQ signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode. • The TQ0OVF bit is not cleared to 0 even when the TQ0OVF bit or the TQ0OPT0 register are read when the TQ0OVF bit = 1. • Before clearing the TQ0OVF bit to 0 after generation of the INTOVQ signal, be sure to confirm (by reading) that the TQ0OVF bit is set to 1. • The TQ0OVF bit can be both read and written, but the TQ0OVF bit cannot be set to 1 by software. Writing 1 has no influence on the operation of TMQ0.	

Note For details of the TQ0CMS and TQ0CUF bits, see **CHAPTER 12 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 the TQ0CE bit to 0 and then set the bits again.

2. Be sure to set bit 3 to 0.

(7) TMQ0 capture/compare register 0 (TQ0CCR0)

The TQ0CCR0 register can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the 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 TQ0CCR0 register can be read or written during operation.

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

Reset input clears this register to 0000H.

After reset: 0000H R/W Address: FFFFF606H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQ0CCR0																

(a) Function as compare register

The TQ0CCR0 register can be rewritten even when the TQ0CTL0.TQ0CE bit = 1.

The set value of the TQ0CCR0 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 (INTCCQ0) is generated. If TOQ0 pin output is enabled at this time, the output of the TOQ0 pin is inverted.

When the TQ0CCR0 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.

<R> The compare register is not cleared by setting the TQ0CTL0.TQ0CE bit to 0.

(b) Function as capture register

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 (INTPQ0 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 (INTPQ0 pin) is detected.

Even if the capture operation and reading the TQ0CCR0 register conflict, the correct value of the TQ0CCR0 register can be read.

<R> The capture register is cleared by setting 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 9-2. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write[Note]
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write[Note]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	None

Note Writing to the TQ0CCR1 register is the trigger.

Remark For anytime write and batch write, see **9.6 (2) Anytime write and batch write**.

(8) TMQ0 capture/compare register 1 (TQ0CCR1)

The TQ0CCR1 register, which consists of 16 bits, can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the 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 TQ0CCR1 register can be read or written during operation.

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

Reset input clears this register to 0000H.

After reset: 0000H R/W Address: FFFFF608H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQ0CCR1																

(a) Function as compare register

The TQ0CCR1 register can be rewritten even when the TQ0CTL0.TQ0CE bit = 1.

The set value of the TQ0CCR1 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 (INTCCQ1) is generated. If TOQ1 pin output is enabled at this time, the output of the TOQ1 pin is inverted.

<R> The compare register is not cleared by setting the TQ0CTL0.TQ0CE bit to 0.

(b) Function as capture register

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 (INTPQ1 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 (INTPQ1 pin) is detected.

Even if the capture operation and reading the TQ0CCR1 register conflict, the correct value of the TQ0CCR1 register can be read.

<R> The capture register is cleared by setting the TQ0CTL0.TQ0CE bit to 0.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 9-3. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write[Note]
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write[Note]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	None

Note Writing to the TQ0CCR1 register is the trigger.

Remark For anytime write and batch write, see **9.6 (2) Anytime write and batch write**.

(9) TMQ0 capture/compare register 2 (TQ0CCR2)

The TQ0CCR2 register, which consists of 16 bits, can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the 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 TQ0CCR2 register can be read or written during operation.

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

Reset input clears this register to 0000H.

After reset: 0000H R/W Address: FFFFF60AH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQ0CCR2																

(a) Function as compare register

The TQ0CCR2 register can be rewritten even when the TQ0CTL0.TQ0CE bit = 1.

The set value of the TQ0CCR2 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 (INTCCQ2) is generated. If TOQ2 pin output is enabled at this time, the output of the TOQ2 pin is inverted.

<R> The compare register is not cleared by setting the TQ0CTL0.TQ0CE bit to 0.

(b) Function as capture register

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 (INTPQ2 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 (INTPQ2 pin) is detected.

Even if the capture operation and reading the TQ0CCR2 register conflict, the correct value of the TQ0CCR2 register can be read.

<R> The capture register is cleared by setting the TQ0CTL0.TQ0CE bit to 0.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 9-4. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write[Note]
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write[Note]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	None

Note Writing to the TQ0CCR1 register is the trigger.

Remark For anytime write and batch write, see **9.6 (2) Anytime write and batch write**.

(10) TMQ0 capture/compare register 3 (TQ0CCR3)

The TQ0CCR3 register, which consists of 16 bits, can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the 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 TQ0CCR3 register can be read or written during operation.

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

Reset input clears this register to 0000H.

After reset: 0000H R/W Address: FFFFF60CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQ0CCR3																

(a) Function as compare register

The TQ0CCR3 register can be rewritten even when the TQ0CTL0.TQ0CE bit = 1.

The set value of the TQ0CCR3 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 (INTCCQ3) is generated. If TOQ3 pin output is enabled at this time, the output of the TOQ3 pin is inverted.

<R> The compare register is not cleared by setting the TQ0CTL0.TQ0CE bit to 0.

(b) Function as capture register

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 (INTPQ3 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 (INTPQ3 pin) is detected.

Even if the capture operation and reading the TQ0CCR3 register conflict, the correct value of the TQ0CCR3 register can be read.

<R> The capture register is cleared by setting the TQ0CTL0.TQ0CE bit to 0.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 9-5. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write[Note]
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write[Note]
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	None

Note Writing to the TQ0CCR1 register is the trigger.

Remark For anytime write and batch write, see **9.6 (2) Anytime write and batch write**.

(11) TMQ0 counter read buffer register (TQ0CNT)

The TQ0CNT register is a read buffer register that can read the count value of the 16-bit counter.

If this register is read when the TQ0CTL0.TQ0CE 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 TQ0CNT register is cleared to 0000H when the TQ0CE bit = 0. If the TQ0CNT 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 TQ0CNT register is cleared to 0000H after reset, and the TQ0CE bit is cleared to 0.

After reset: 0000H R Address: FFFFF60EH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TQ0CNT																

9.5 Timer Output Operations

The following table shows the operations and output levels of the TOQ0 to TOQ3 pins.

Table 9-6. Timer Output Control in Each Mode

Operation Mode	TOQ0 Pin	TOQ1 Pin	TOQ2 Pin	TOQ3 Pin
Interval timer mode	PWM output			
External event count mode	None			
External trigger pulse output mode	PWM output	External trigger pulse 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			

Table 9-7. Truth Table of TOQ0 to TOQ3 Pins Under Control of Timer Output Control Bits

TQ0IOC0.TQ0OLa Bit	TQ0IOC0.TQ0OEa Bit	TQ0CTL0.TQ0CE Bit	Level of TOQ0a 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 to 3

9.6 Operation

TMQ0 can perform the following functions.

Operation	TQ0CTL1.TQ0EST Bit (Software Trigger Bit)	TIQ Pin (External Trigger Input)	Capture/Compare Register Setting	Compare Register Write
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode	Invalid	Invalid	Compare only	Anytime write
External trigger pulse output mode	Valid	Valid	Compare only	Batch write
One-shot pulse output mode	Valid	Valid	Compare only	Anytime write
PWM output mode	Invalid	Invalid	Compare only	Batch write
Free-running timer mode	Invalid	Invalid	Switching enabled	Anytime write
Pulse width measurement mode	Invalid	Invalid	Capture only	Not applicable

(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.

(a) Counter start operation

The 16-bit counter of TMQ0 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 INTCCQm interrupt signal is not generated (m = 0 to 3).

(c) Overflow operation

The 16-bit counter overflows when the counter counts up from FFFFH to 0000H in the free-running mode or pulse width measurement mode. If the counter overflows, the TQ0OPT0.TQ0OVF bit is set to 1 and an interrupt request signal (INTOVQ) is generated. Note that the INTOVQ signal is not generated under the following conditions.

- Immediately after a count operation has been started
- If the counter value matches the compare value FFFFH and is cleared
- When FFFFH is captured in the pulse width measurement mode and the counter counts up from FFFFH to 0000H

Caution After the overflow interrupt request signal (INTOVQ) has been generated, be sure to check that the overflow flag (TQ0OVF bit) is set to 1.

(d) Counter read operation during counting operation

The value of the 16-bit counter of TMQ0 can be read by using the TQ0CNT register during the count operation. When the TQ0CTL0.TQ0CE bit = 1, the value of the 16-bit counter can be read by reading the TQ0CNT register. When the TQ0CE bit = 0, the 16-bit counter is FFFFH and the TQ0CNT register is 0000H.

(e) Interrupt operation

TMQ0 generates the following five interrupt request signals.

- INTCCQ0 interrupt: This signal functions as a match interrupt request signal of the CCR0 buffer register and as a capture interrupt request signal to the TQ0CCR0 register.
- INTCCQ1 interrupt: This signal functions as a match interrupt request signal of the CCR1 buffer register and as a capture interrupt request signal to the TQ0CCR1 register.
- INTCCQ2 interrupt: This signal functions as a match interrupt request signal of the CCR2 buffer register and as a capture interrupt request signal to the TQ0CCR2 register.
- INTCCQ3 interrupt: This signal functions as a match interrupt request signal of the CCR3 buffer register and as a capture interrupt request signal to the TQ0CCR3 register.
- INTOVQ interrupt: This signal functions as an overflow interrupt request signal.

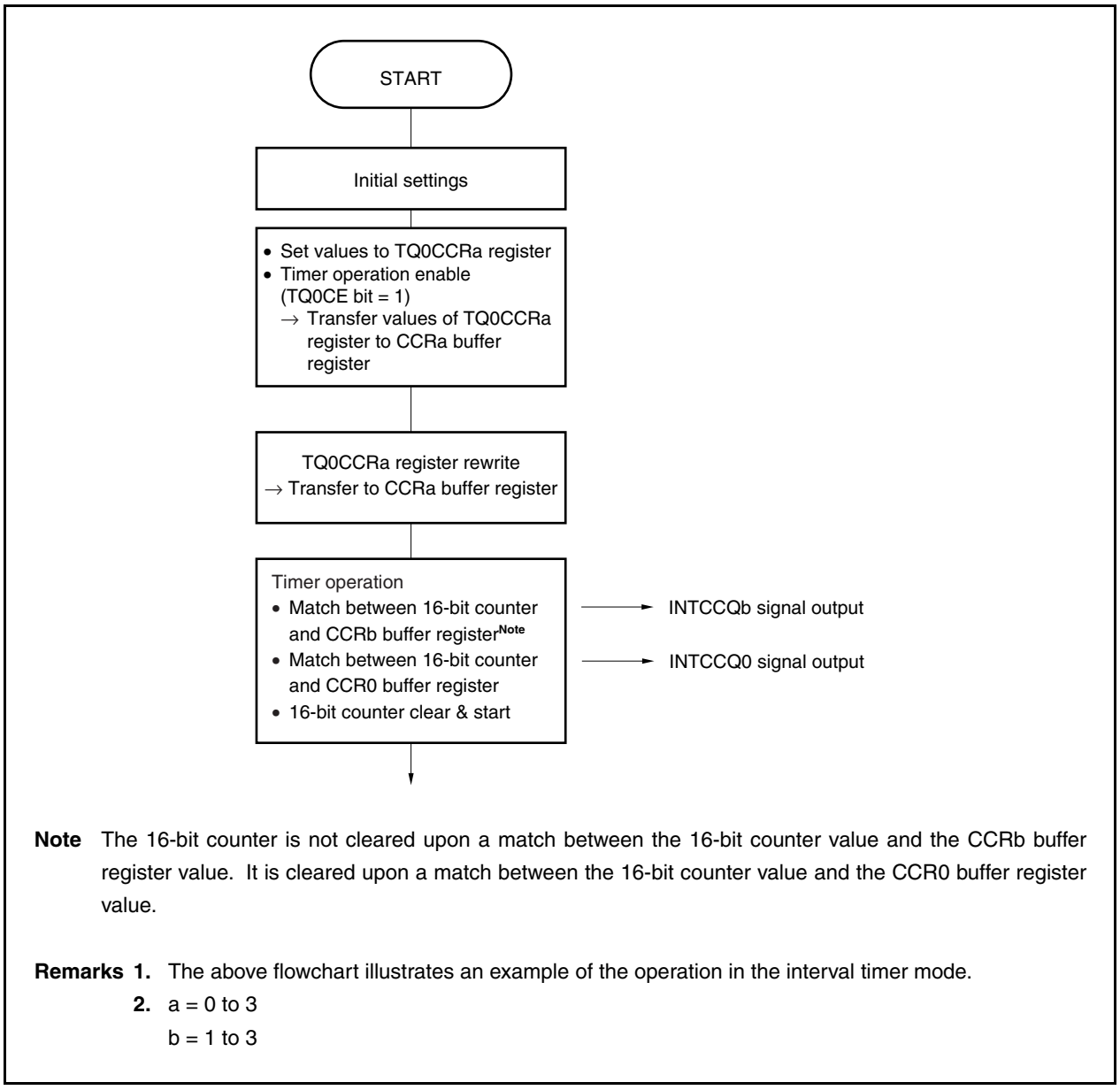
(2) Anytime write and batch write

The TQ0CCR0 to TQ0CCR3 registers can be rewritten in the TMQ0 during timer operation (TQ0CTL0.TQ0CE 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 TQ0CCR0 to TQ0CCR3 registers to the CCR0 to CCR3 buffer registers during the timer operation.

Figure 9-2. Flowchart of Basic Operation for Anytime Write

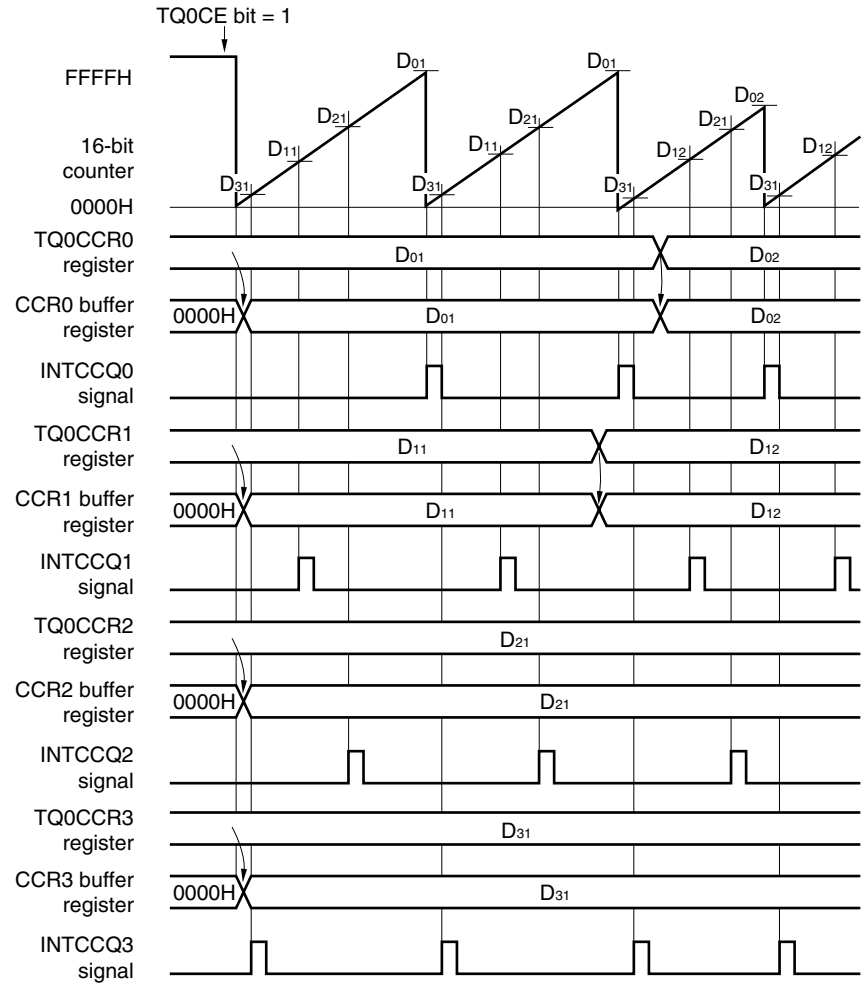


<R> **Note** The 16-bit counter is not cleared upon a match between the 16-bit counter value and the CCRb buffer register value. It is cleared upon a match between the 16-bit counter value and the CCR0 buffer register value.

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

2. a = 0 to 3
b = 1 to 3

Figure 9-3. Timing of Anytime Write



Remarks 1. D_{01}, D_{02}: Setting values of TQ0CCR0 register

D_{11}, D_{12}: Setting values of TQ0CCR1 register

D_{21}: Setting value of TQ0CCR2 register

D_{31}: Setting value of TQ0CCR3 register

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

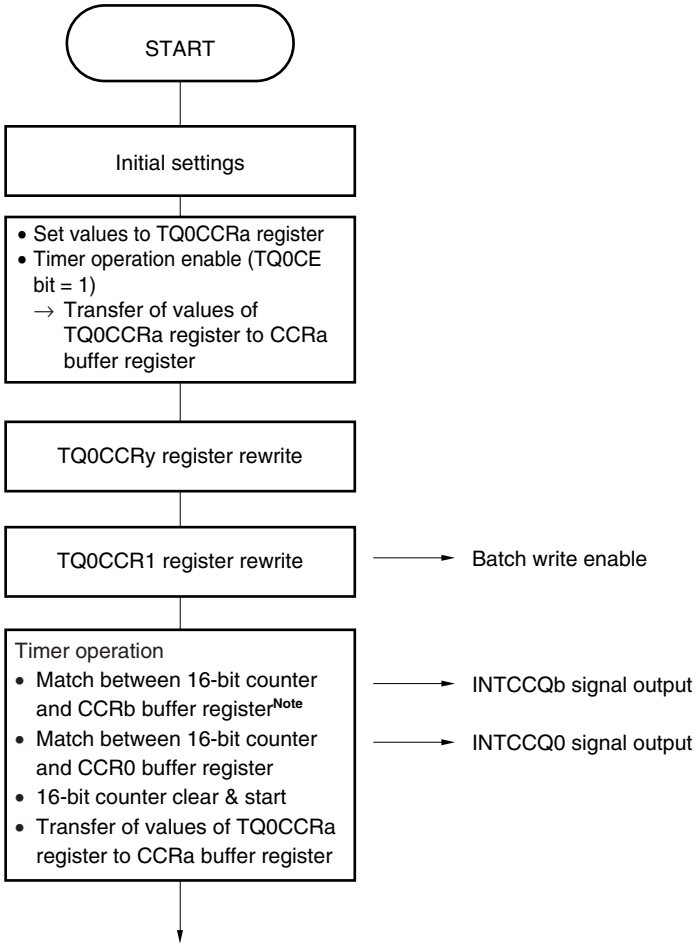
(b) Batch write

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.

Figure 9-4. Flowchart of Basic Operation for Batch Write



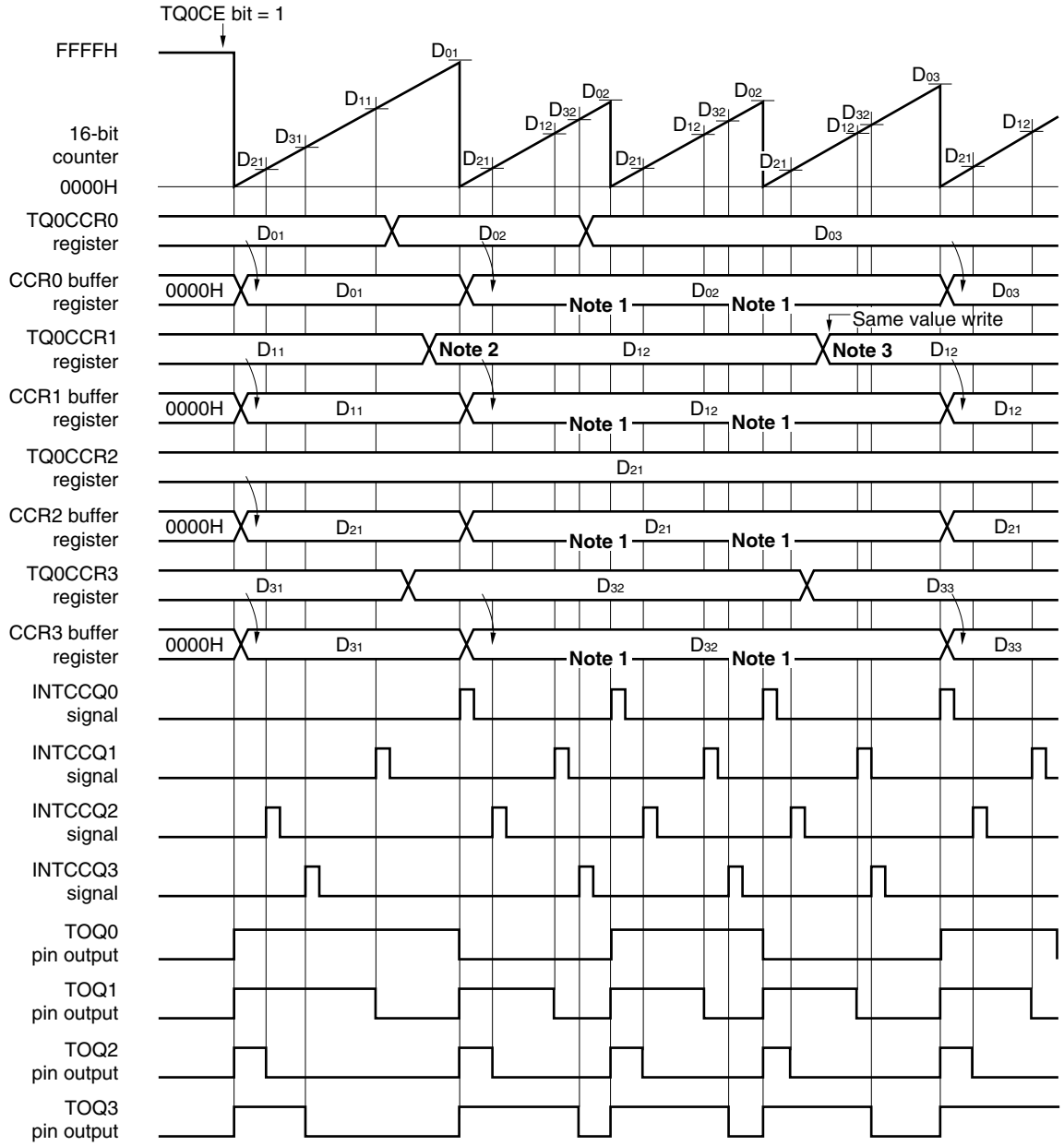
<R> **Note** The 16-bit counter is not cleared upon a match between the 16-bit counter value and the CCRb buffer register value. It is cleared upon a match between the 16-bit counter value and the CCR0 buffer register value.

Caution Writing to the 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 9-5. 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 flowchart illustrates the operation in the PWM output mode as an example.

9.6.1 Interval timer mode (TQ0MD2 to TQ0MD0 bits = 000)

In the interval timer mode, an interrupt request signal (INTCCQ0) is generated at the interval set by the TQ0CCR0 register if the TQ0CTL0.TQ0CE 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 TOQ0 pin.

The TQ0CCR1 to TQ0CCR3 registers are not used in the interval timer mode. However, the set value of the TQ0CCR1 to TQ0CCR3 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 (INTCCQ1 to INTCCQ3) are generated. In addition, a PWM waveform with a duty factor of 50%, which is inverted when the INTCCQ1 to INTCCQ3 signals are generated, can be output from the TOQ1 to TOQ3 pins.

The value of the TQ0CCR1 to TQ0CCR3 registers can be rewritten even while the timer is operating.

Figure 9-6. Interval Timer Configuration

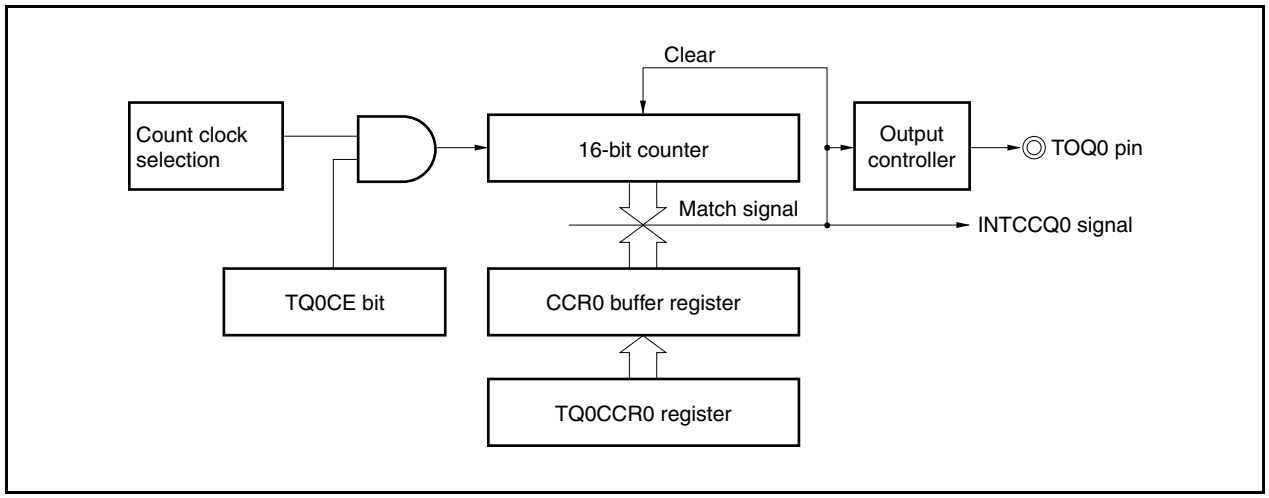
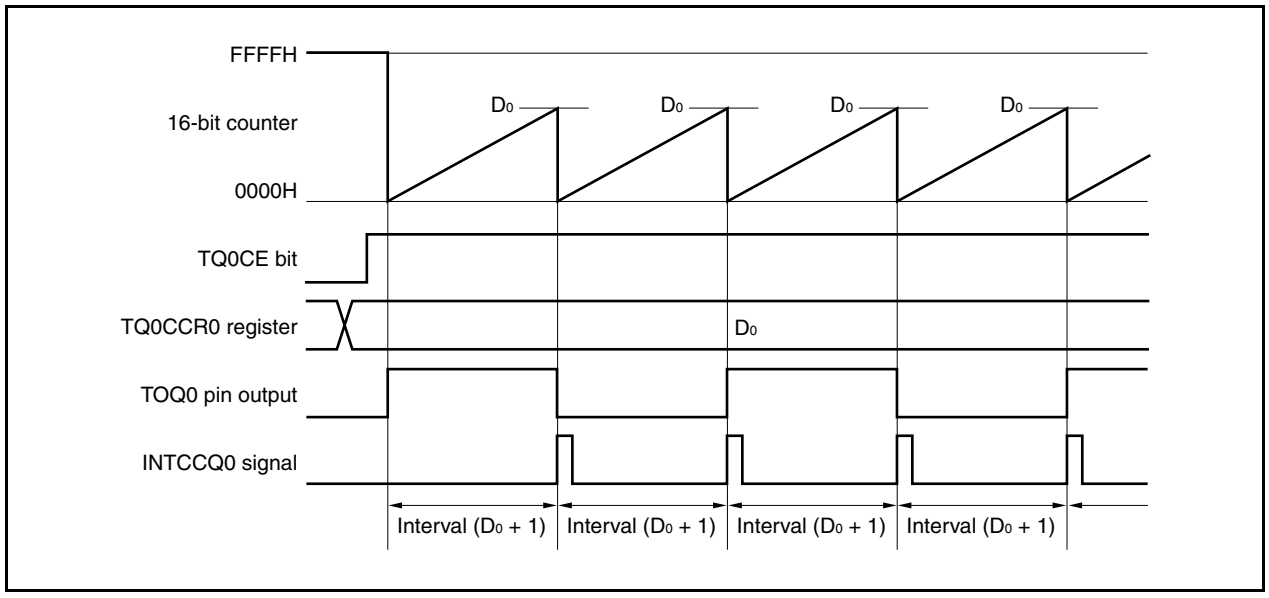


Figure 9-7. Basic Timing of Operation in Interval Timer Mode



When the TQ0CE 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 TOQ0 pin is inverted. 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, the output of the TOQ0 pin is inverted, and a compare match interrupt request signal (INTCCQ0) is generated.

The interval can be calculated by the following expression.

Interval = (Set value of TQ0CCR0 register + 1) × Count clock cycle

Figure 9-8. Register Setting for Interval Timer Mode Operation (1/3)

(a) TMQ0 control register 0 (TQ0CTL0)

	TQ0CE					TQ0CKS2	TQ0CKS1	TQ0CKS0
TQ0CTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock

0: Stop counting
1: Enable counting

(b) TMQ0 control register 1 (TQ0CTL1)

		TQ0EST	TQ0EEE			TQ0MD2	TQ0MD1	TQ0MD0
<R>	TQ0CTL1	0	0	0/1 Note	0	0	0	0	0

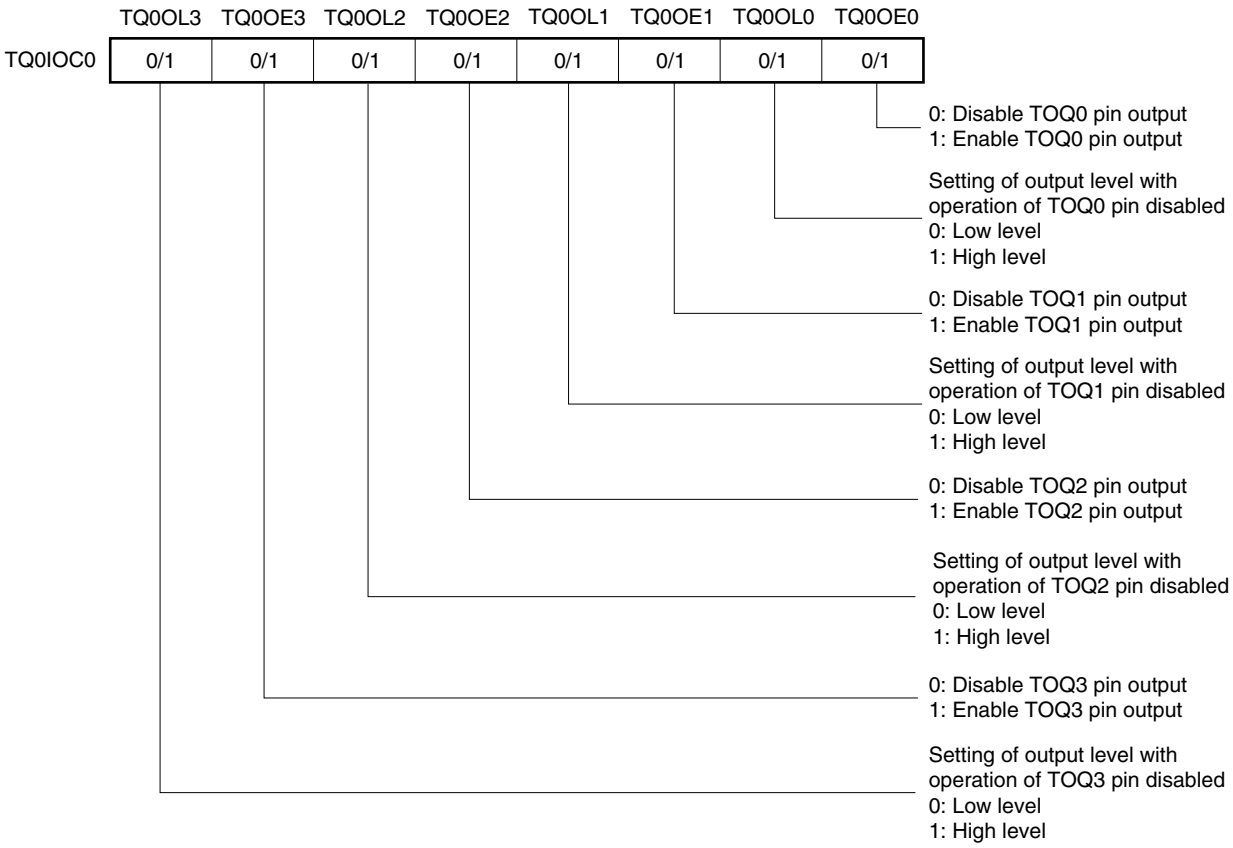
0, 0, 0:
Interval timer mode

0: Operate on count clock selected by TQ0CKS0 to TQ0CKS2 bits
1: Count with external event count input signal

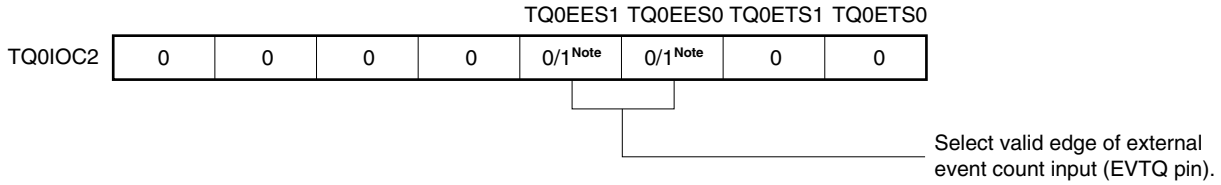
<R> **Note** The TQ0EEE bit can be set to 1 only when timer output (TOQ0, TOQb) is used. However, set the TQ0CCR0 and TQ0CCRb registers to the same value (b = 1 to 3).

Figure 9-8. Register Setting for Interval Timer Mode Operation (2/3)

(c) TMQ0 I/O control register 0 (TQ0IOC0)



<R> **(d) TMQ0 I/O control register 2 (TQ0IOC2)**



Note The TQ0EES1 and TQ0EES0 bits can be set only when timer output (TOQ0 to TOQ3) is used. However, set the TQ0CCR0 to TQ0CCR3 registers to the same value.

(e) TMQ0 counter read buffer register (TQ0CNT)

By reading the TQ0CNT register, the count value of the 16-bit counter can be read.

(f) TMQ0 capture/compare register 0 (TQ0CCR0)

If the TQ0CCR0 register is set to D_0, the interval is as follows.

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

Figure 9-8. Register Setting for Interval Timer Mode Operation (3/3)

(g) TMQ0 capture/compare registers 1 to 3 (TQ0CCR1 to TQ0CCR3)

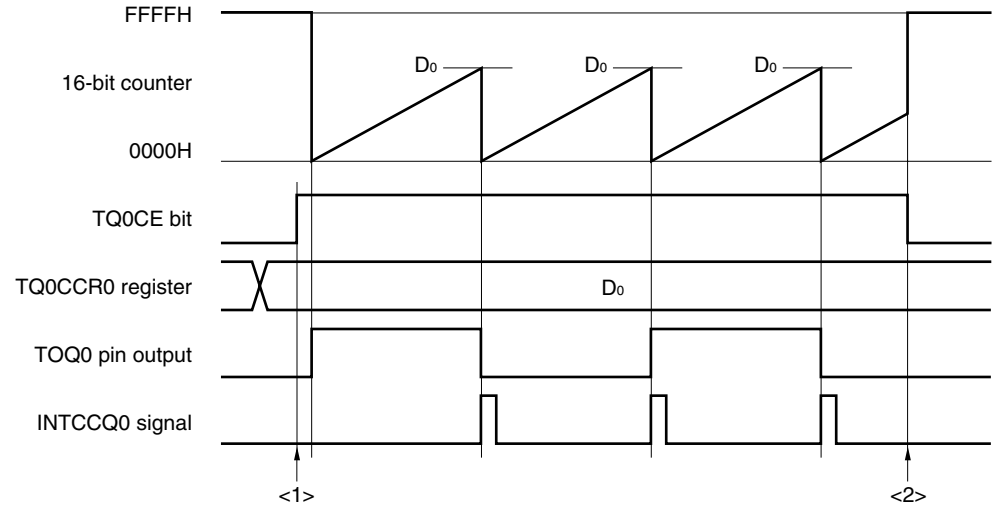
The TQ0CCR1 to TQ0CCR3 registers are not used in the interval timer mode. However, the set value of the TQ0CCR1 to TQ0CCR3 registers are transferred to the CCR1 to CCR3 buffer registers. The compare match interrupt request signals (INTCCQ1 to INTCCQ3) is generated when the count value of the 16-bit counter matches the value of the CCR1 to CCR3 buffer registers.

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 (P01IC1.P01MK1 to P01IC3.P01MK3).

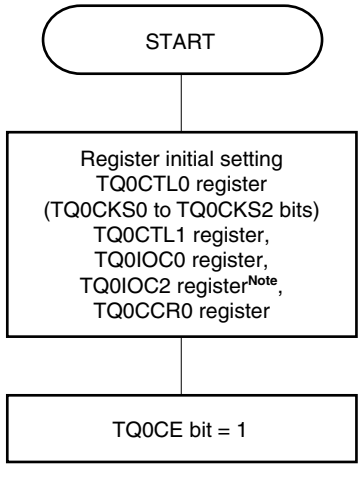
<R> **Remark** TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the interval timer mode.

(1) Interval timer mode operation flow

Figure 9-9. Software Processing Flow in Interval Timer Mode



<1> Count operation start flow

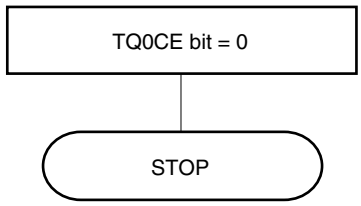


Initial setting of these registers is performed before setting the TQ0CE bit to 1.

The TQ0CKS0 to TQ0CKS2 bits can be set at the same time when counting has been started (TQ0CE bit = 1).

<R> **Note** The TQ0EES1 and TQ0EES0 bits can be set only when timer output (TOQ0, TOQb) is used. However, set the TQ0CCR0 and TQ0CCRb registers to the same value (b = 1 to 3).

<2> Count operation stop flow



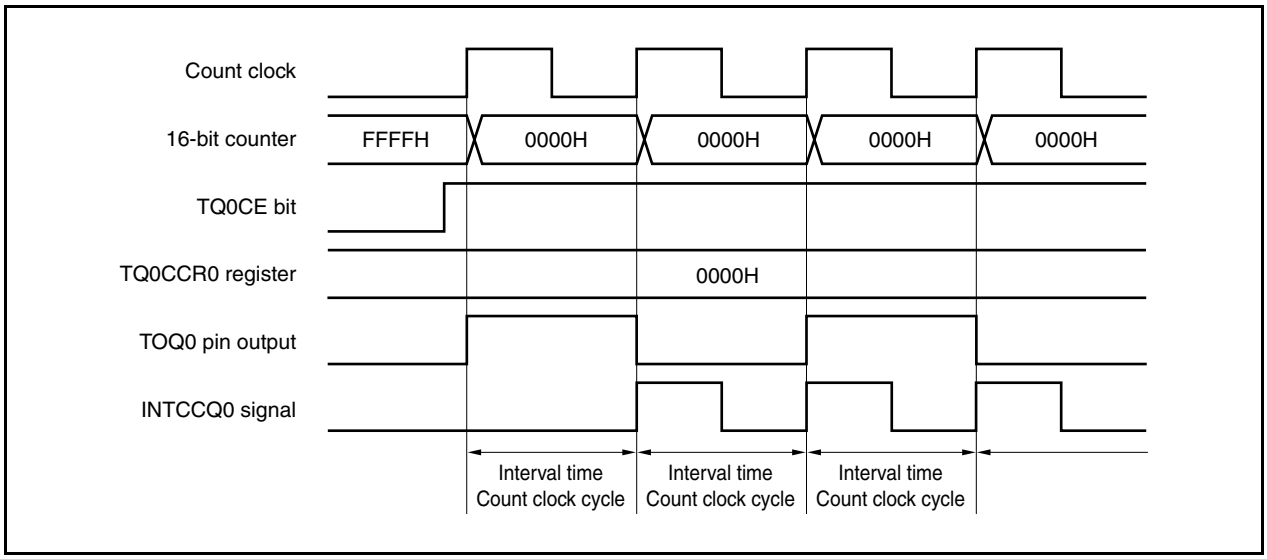
The counter is initialized and counting is stopped by clearing the TQ0CE bit to 0. The output level of the TOQ0 pin is as specified by the TQ0IOC0 register.

(2) Interval timer mode operation timing

(a) Operation if TQ0CCR0 register is set to 0000H

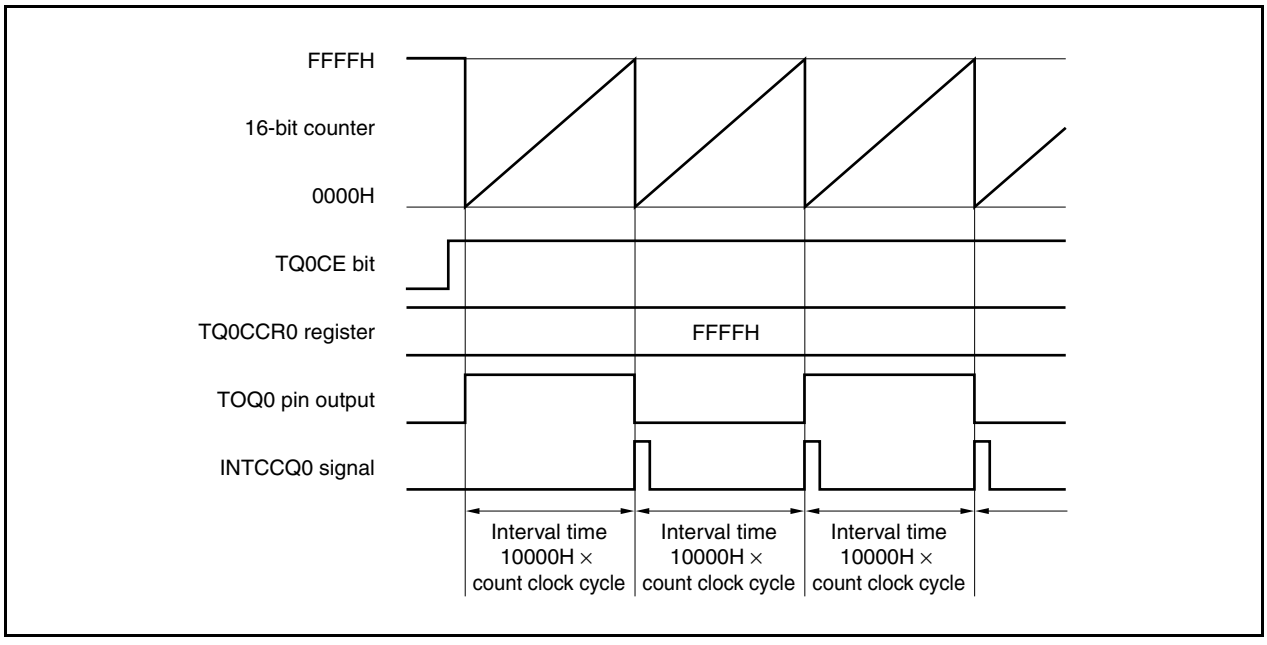
If the TQ0CCR0 register is set to 0000H, the INTCCQ0 signal is generated at each count clock, and the output of the TOQ0 pin is inverted.

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



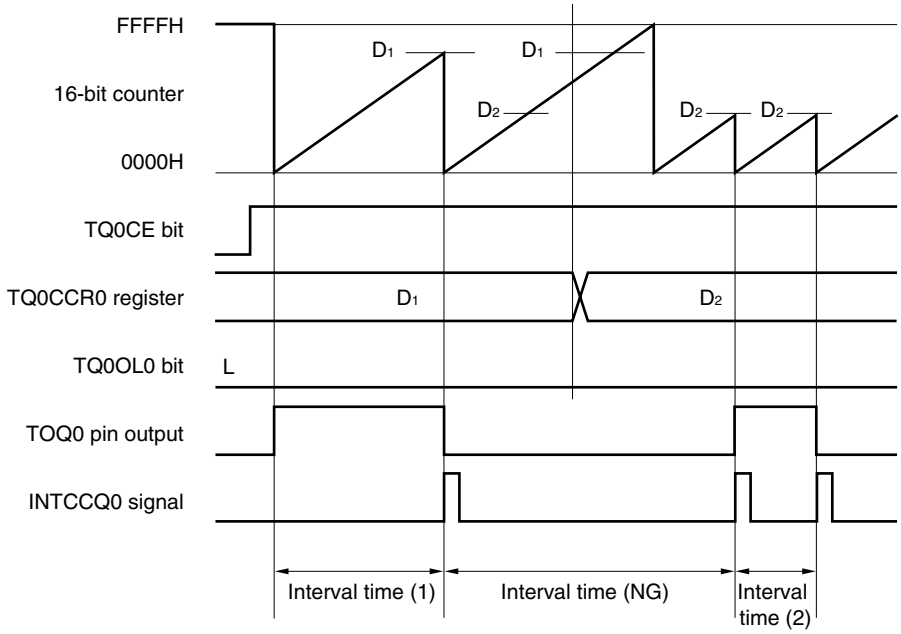
(b) Operation if TQ0CCR0 register is set to FFFFH

If the TQ0CCR0 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 INTCCQ0 signal is generated and the output of the TOQ0 pin is inverted. At this time, an overflow interrupt request signal (INTOVQ) is not generated, nor is the overflow flag (TQ0OPT0.TQ0OVF bit) set to 1.



(c) Notes on rewriting 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.



Remark 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

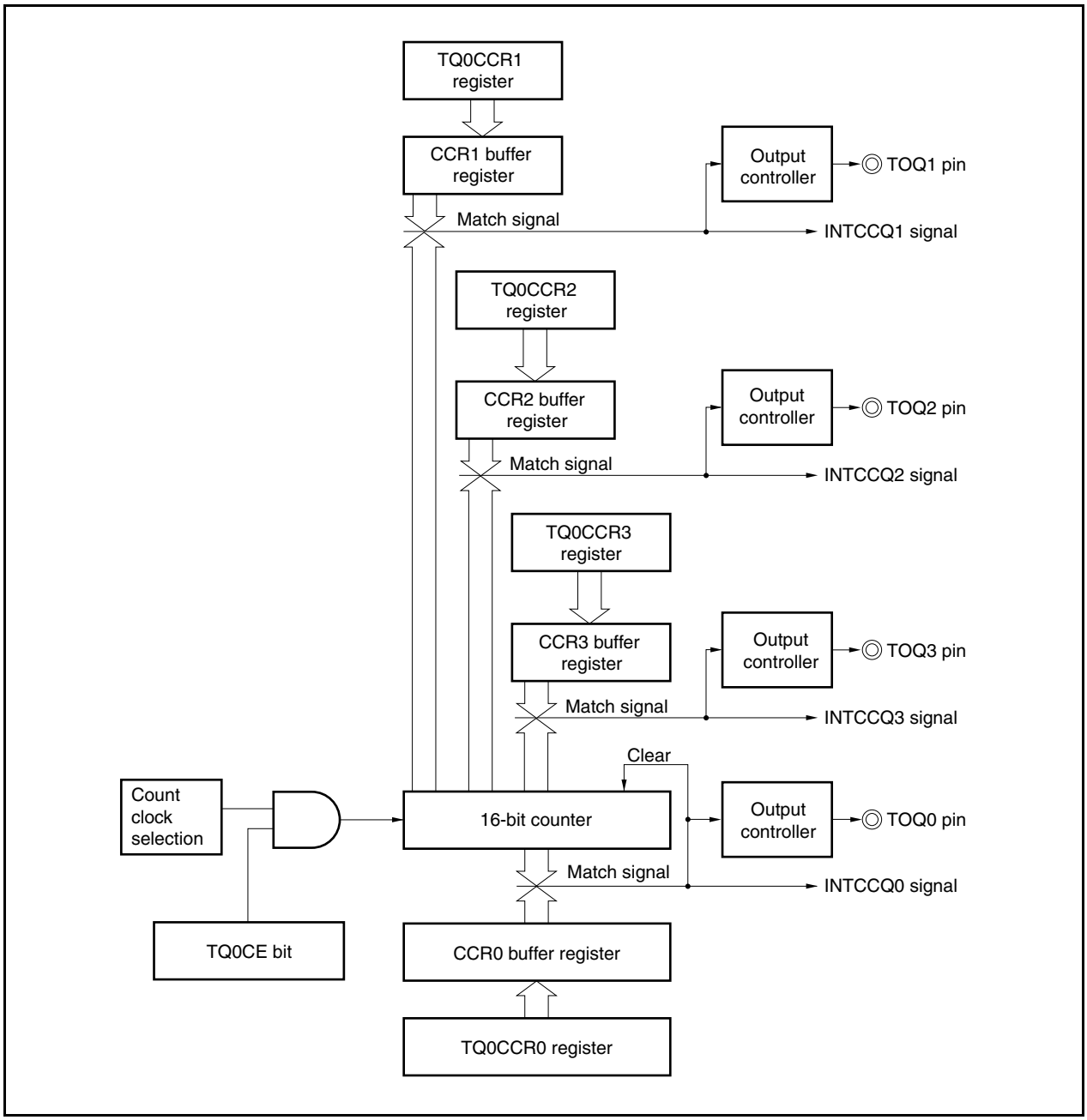
If the value of the TQ0CCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TQ0CCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is 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 INTCCQ0 signal is generated and the output of the TOQ0 pin is inverted.

Therefore, the INTCCQ0 signal may not be generated at the interval time "$(D_1 + 1) \times$ Count clock cycle" or "$(D_2 + 1) \times$ Count clock cycle" originally expected, but may be generated at an interval of "$(10000H + D_2 + 1) \times$ Count clock period".

(d) Operation of TQ0CCR1 to TQ0CCR3 registers

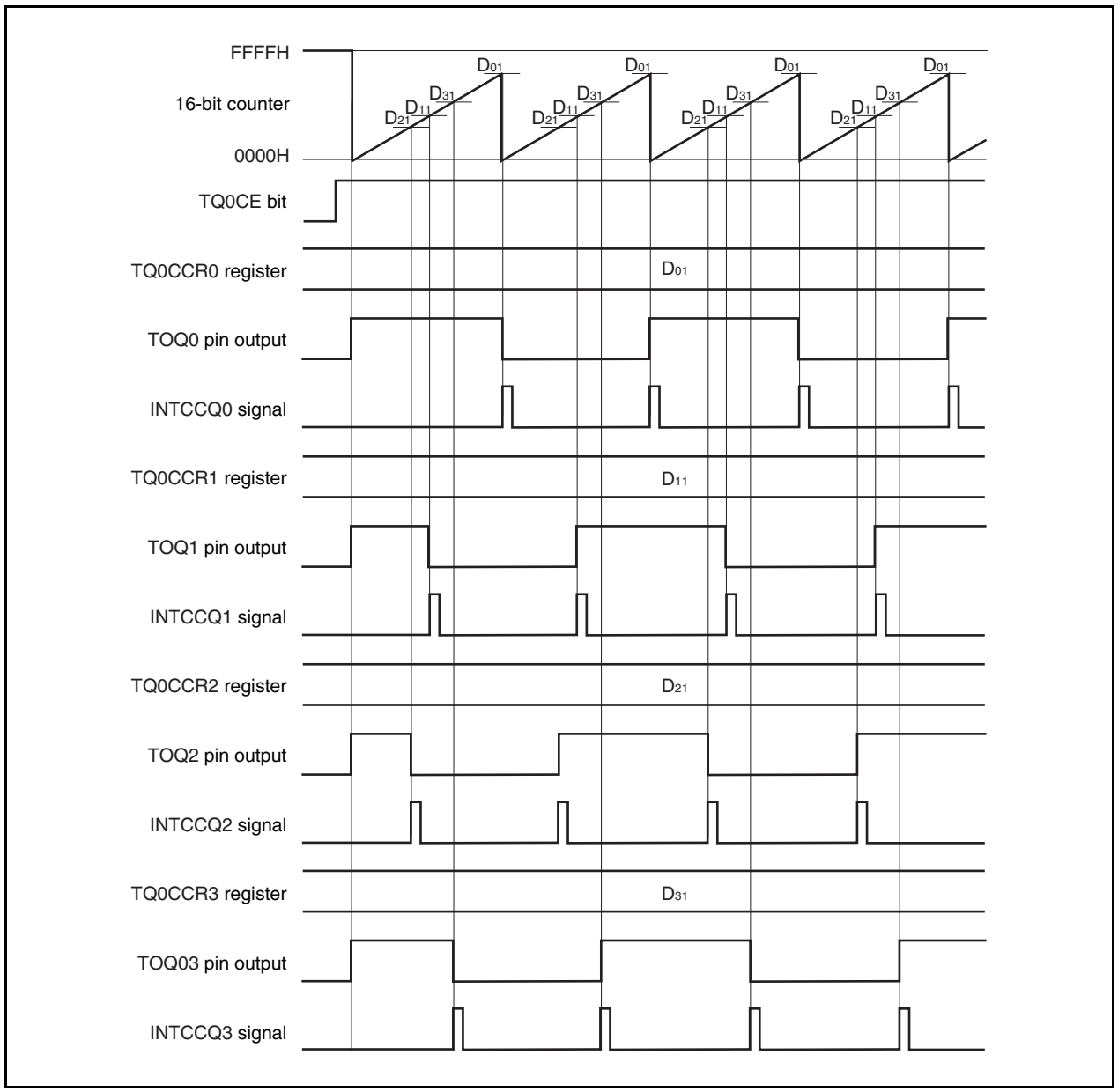
Figure 9-10. Configuration of TQ0CCR1 to TQ0CCR3 Registers



If the set value of the TQ0CCRb register is less than the set value of the TQ0CCR0 register, the INTCCQb signal is generated once per cycle. At the same time, the output of the TOQb pin is inverted.
The TOQb pin outputs a PWM waveform with a duty factor of 50% with the same cycle as that output by the TOQ0 pin.

Remark b = 1 to 3

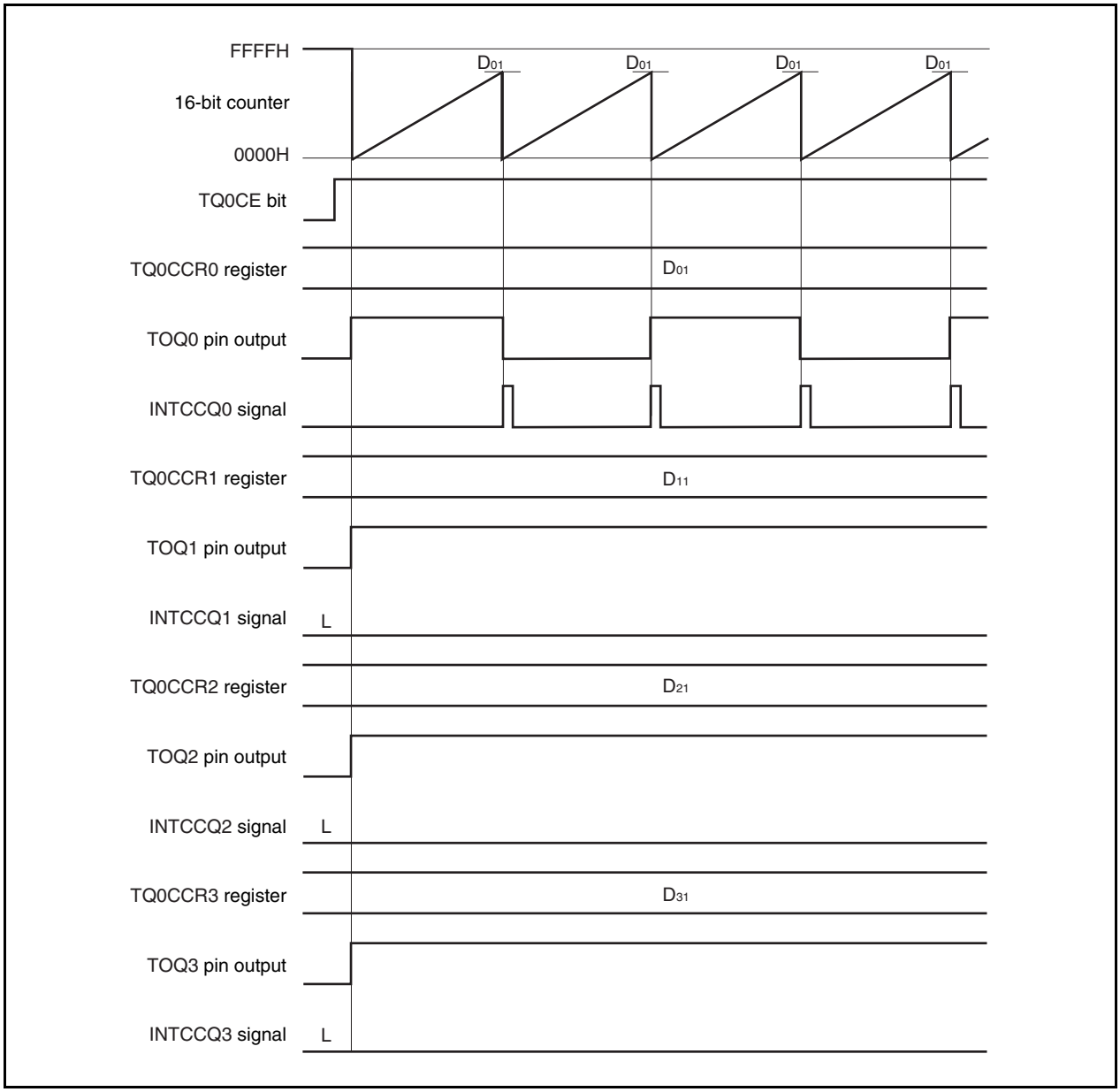
Figure 9-11. Timing Chart When $D_{01} \geq D_{b1}$



If the set value of the TQ0CCRb register is greater than the set value of the TQ0CCR0 register, the count value of the 16-bit counter does not match the value of the TQ0CCRb register. Consequently, the INTCCQb signal is not generated, nor is the output of the TOQb pin changed.
When the TQ0CCRb register is not used, it is recommended to set its value to FFFFH.

Remark b = 1 to 3

Figure 9-12. Timing Chart When $D_{01} < D_{b1}$



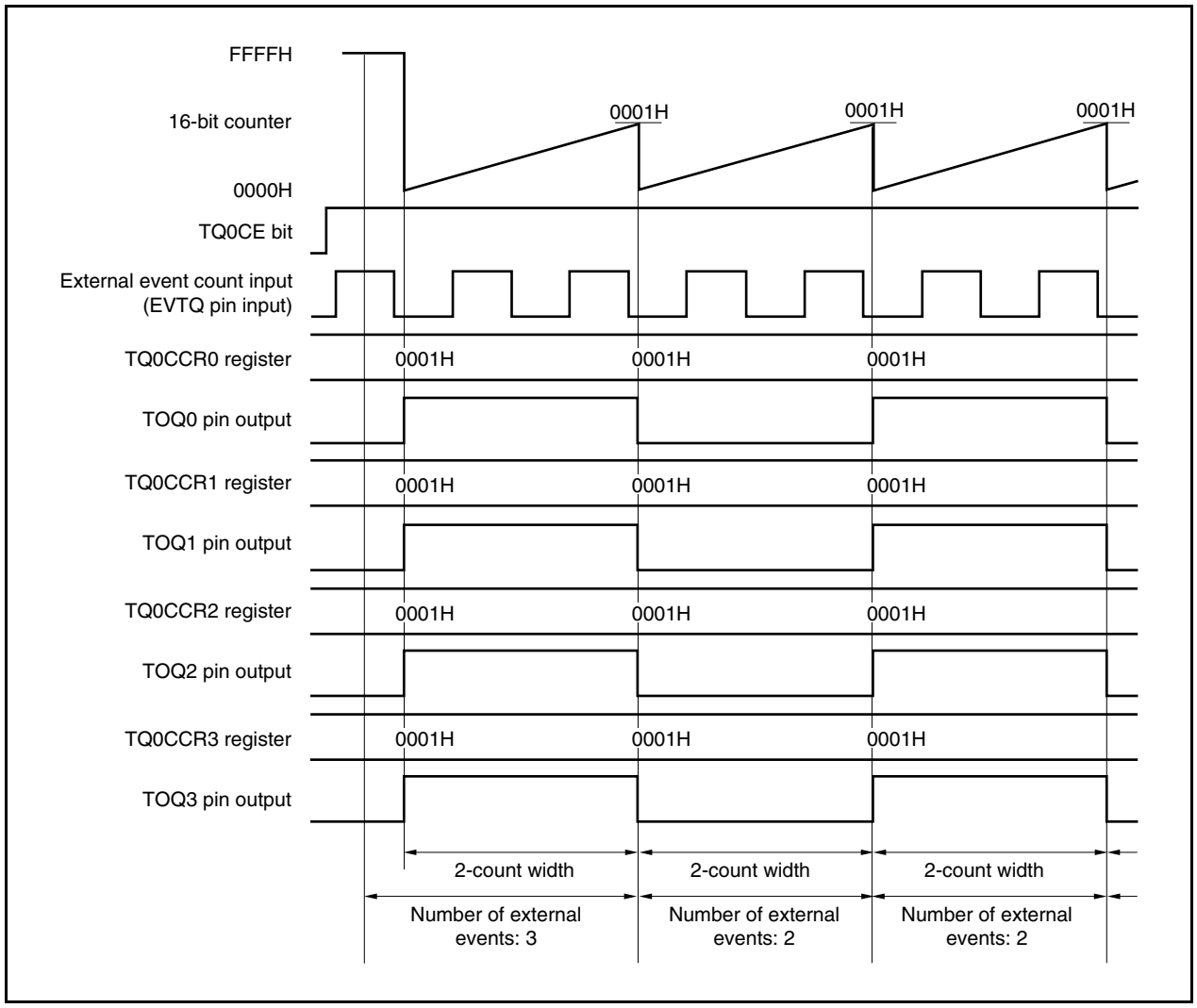
<R> (3) Operation by external event count input (EVTQ)

(a) Operation

To count the 16-bit counter at the valid edge of external event count input (EVTQ) in the interval timer mode, the valid edge of the external event count input is necessary once because the 16-bit counter is cleared from FFFFH to 0000H immediately after the TQ0CE bit is set from 0 to 1.

When 0001H is set to both the TQ0CCR0 and TQ0CCRb registers, the output of the TOQ0 and TOQb pins is inverted each time the 16-bit counter counts twice (b = 1 to 3).

The TQ0CTL1.TQ0EEE bit can be set to 1 in the interval timer mode only when the timer output (TOQ0, TOQb) is used with the external event count input.



9.6.2 External event count mode (TQ0MD2 to TQ0MD0 bits = 001)

In the external event count mode, the valid edge of the external event count input (EVTQ) is counted when the TQ0CTL0.TQ0CE bit is set to 1, and an interrupt request signal (INTCCQ0) is generated each time the specified <R> number of edges set by the TQ0CCR0 register have been counted. The TOQ0 to TOQ3 pins cannot be used. When using the TOQ0 and TOQ3 pins for external event count input, set the TQ0CTL1.TQ0EEE bit to 1 in the interval timer mode (see **9.6.1 (3) Operation by external event count input (EVTQ)**).

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 9-13. Configuration in External Event Count Mode

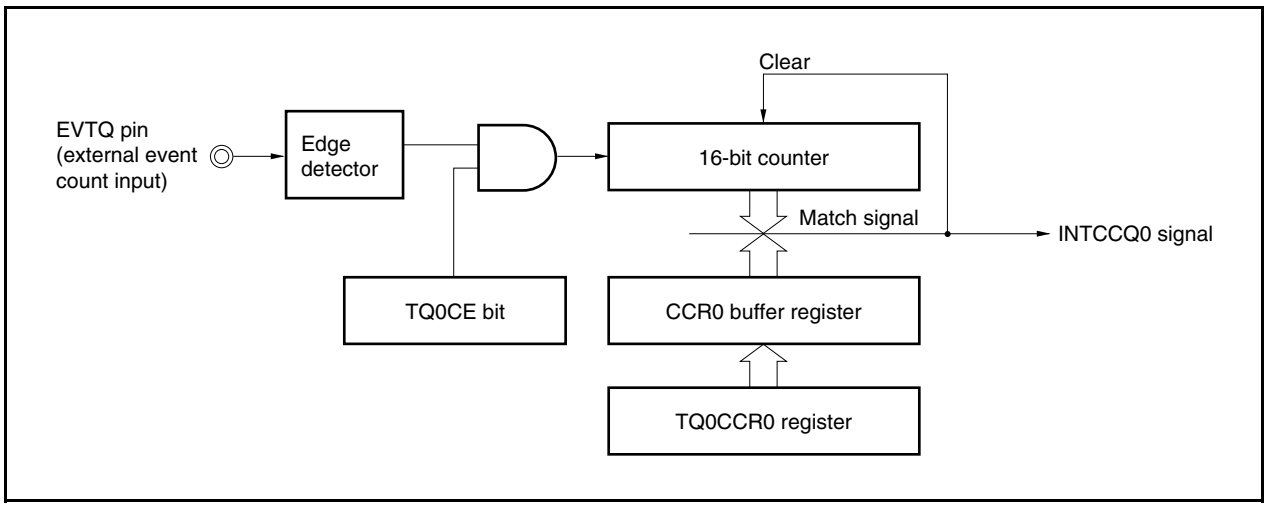
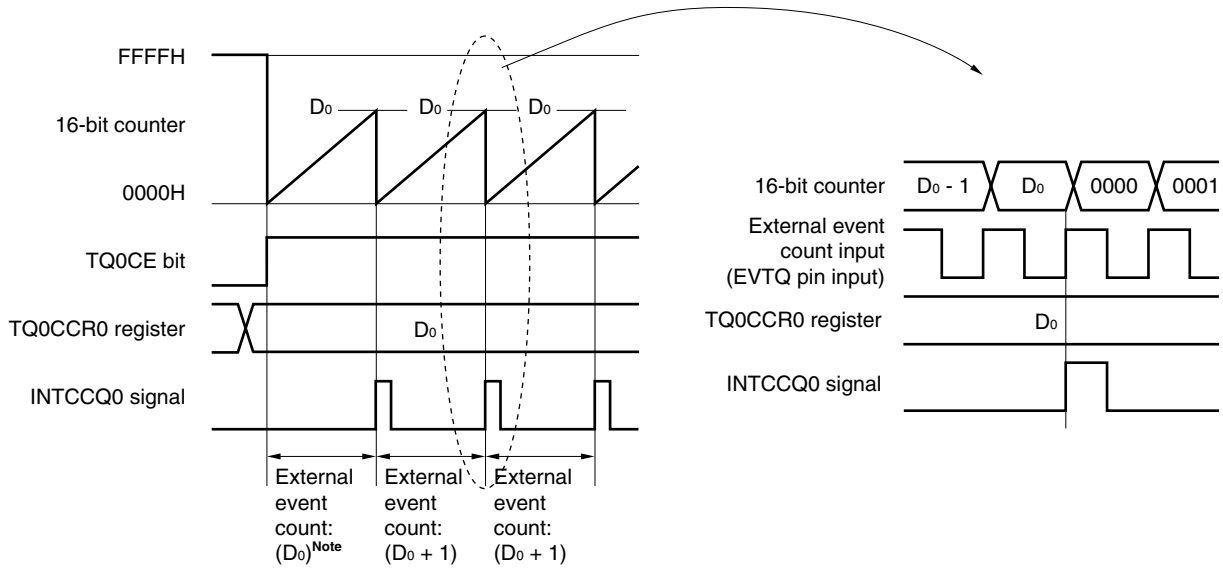


Figure 9-14. Basic Timing in External Event Count Mode



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.

Caution This figure shows the basic timing when the rising edge is specified as the valid edge of the external event count input.

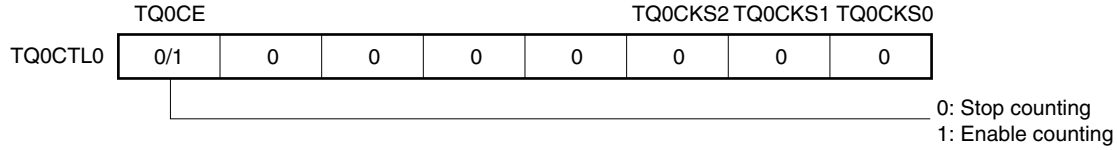
When the TQ0CE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H. The counter counts each time the valid edge of external event count input is detected. Additionally, the set value of the TQ0CCR0 register is transferred to the CCR0 buffer register.

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 (INTCCQ0) is generated.

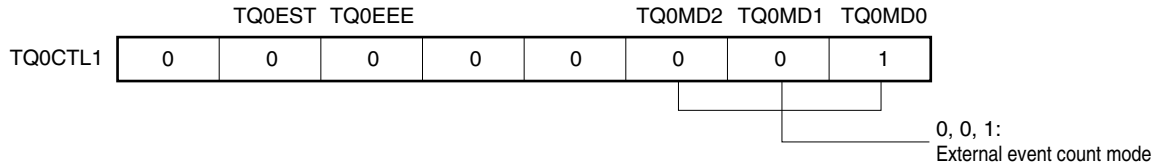
The INTCCQ0 signal is generated for the first time when the valid edge of the external event count input has been detected “value set to TQ0CCR0 register” times. After that, the INTCCQ0 signal is generated each time the valid edge of the external event count has been detected “value set to TQ0CCR0 register + 1” times.

Figure 9-15. Register Setting for Operation in External Event Count Mode

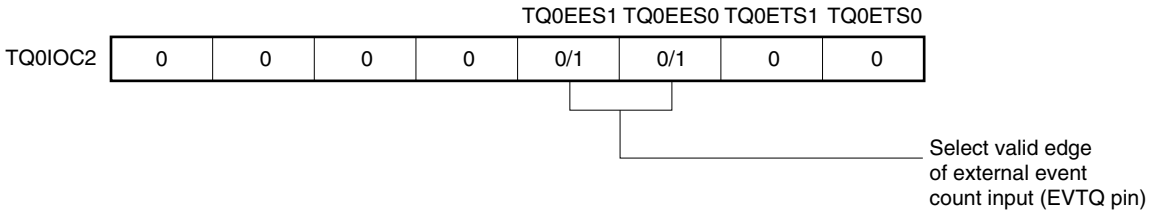
(a) TMQ0 control register 0 (TQ0CTL0)



(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 events has reached (D_0) and the first compare match interrupt request signal (INTCCQ0) is generated. The second compare match interrupt request signal (INTCCQ0) is generated when the number of external events has reached ($D_0 + 1$).

(f) TMQ0 capture/compare registers 1 to 3 (TQ0CCR1 to TQ0CCR3)

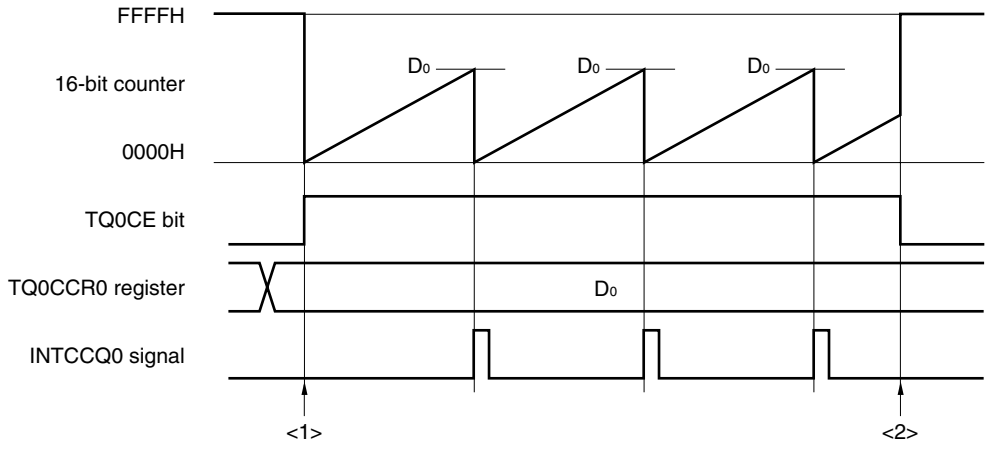
The TQ0CCR1 to TQ0CCR3 registers are not used in the external event count mode. However, the set value of the TQ0CCR1 to TQ0CCR3 registers are transferred to the CCR1 to CCR3 buffer registers. When the count value of the 16-bit counter matches the value of the CCR1 to CCR3 buffer registers, compare match interrupt request signals (INTCCQ1 to INTCCQ3) 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 (P01IC1.P01MK1 to P01IC3.P01MK3).

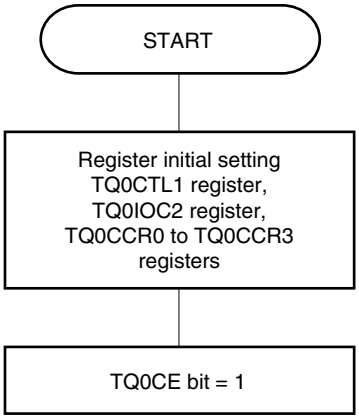
Remark TMQ0 I/O control register 0 (TQ0IOC0), TMQ0 I/O control register 1 (TQ0IOC1), and TMQ0 option register 0 (TQ0OPT0) are not used in the external event count mode.

(1) External event count mode operation flow

Figure 9-16. Software Processing Flow in External Event Count Mode

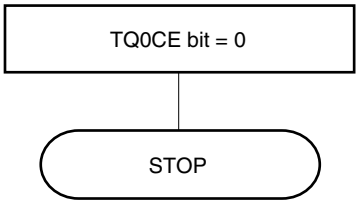


<1> Count operation start flow



Initial setting of these registers is performed before setting the TQ0CE bit to 1.

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TQ0CE bit to 0.

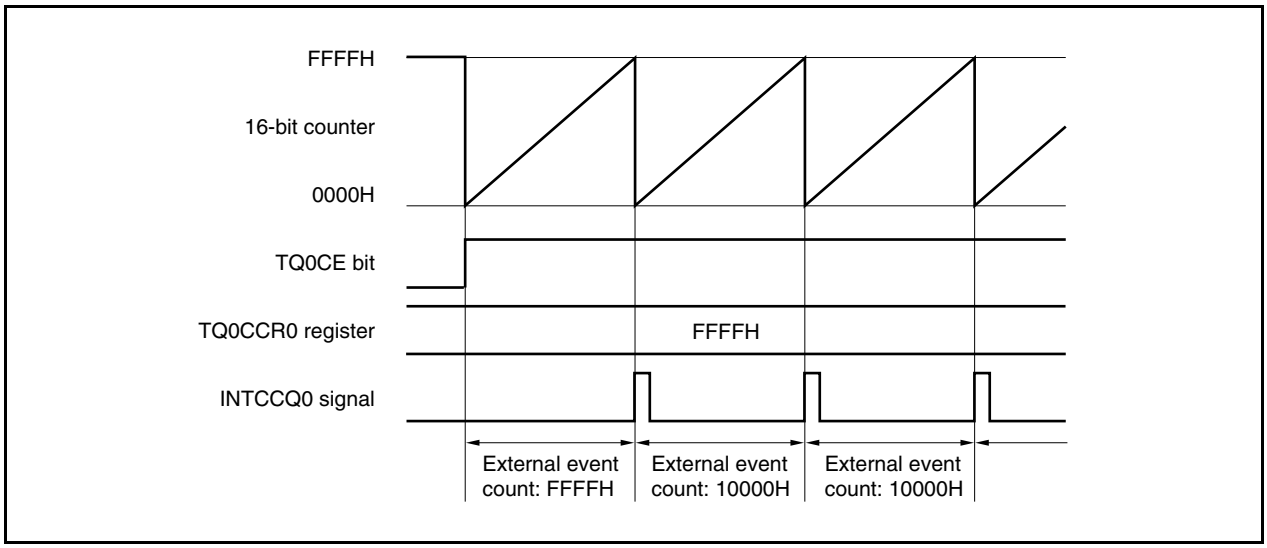
(2) Operation timing in external event count mode

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 (TOQ0 to TOQ3) is disabled. If using timer output (TOQ0, TOQb) with external event count input (EVTQ), set the interval timer mode, and select the operation enabled by the external event count input for the count clock (TQ0CTL1.TQ0EEE bit = 1) (see 9.6.1 (3) Operation by external event count input (EVTQ)) (b = 1 to 3).**

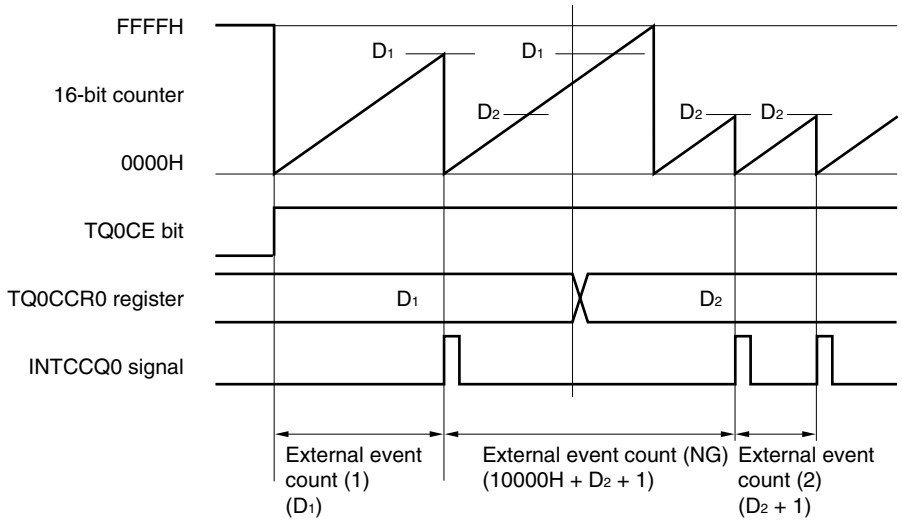
(a) Operation if TQ0CCR0 register is set to FFFFH

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 INTCCQ0 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.



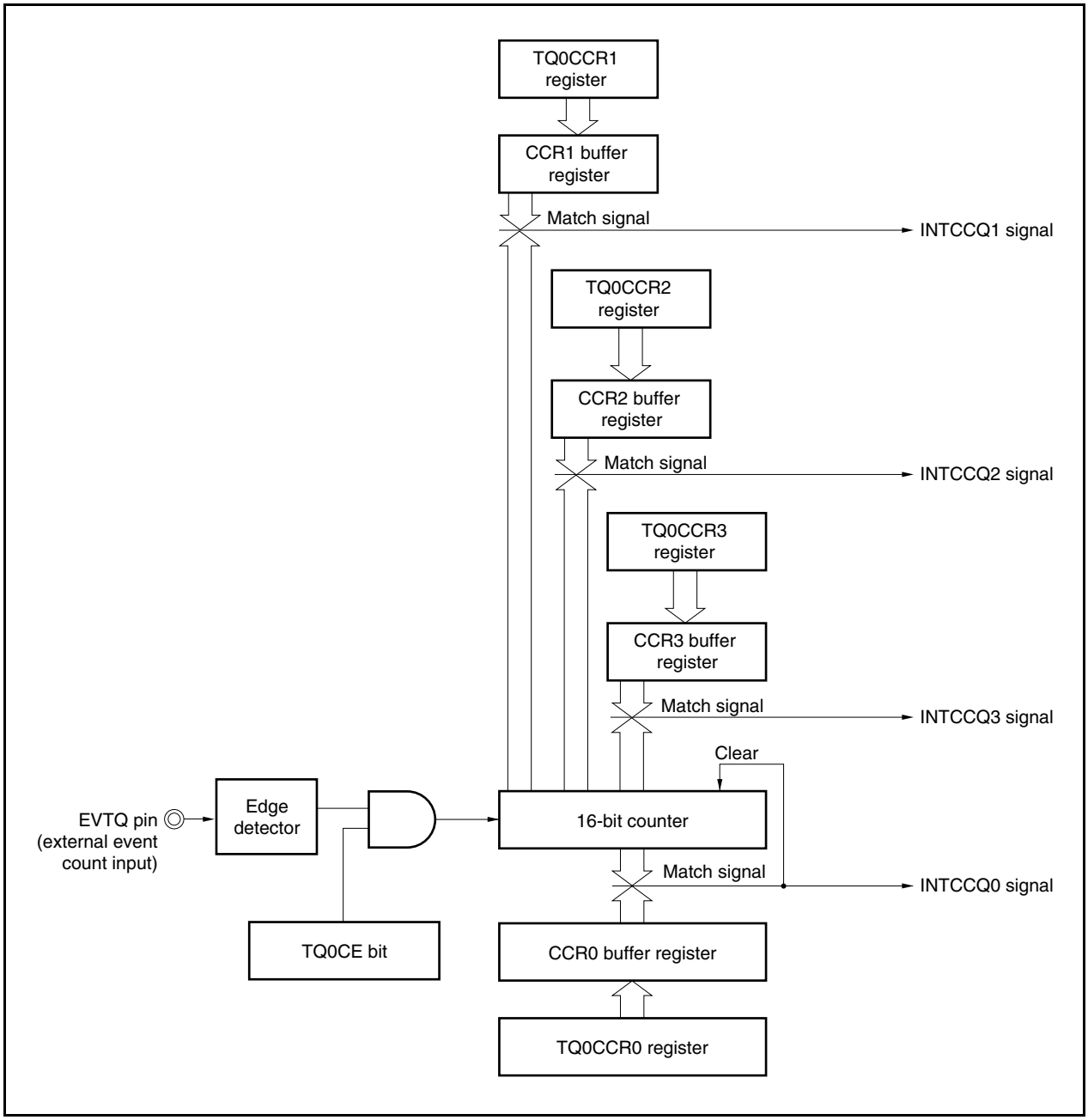
If the value of the TQ0CCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TQ0CCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter 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 INTCCQ0 signal is generated.

Therefore, the INTCCQ0 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

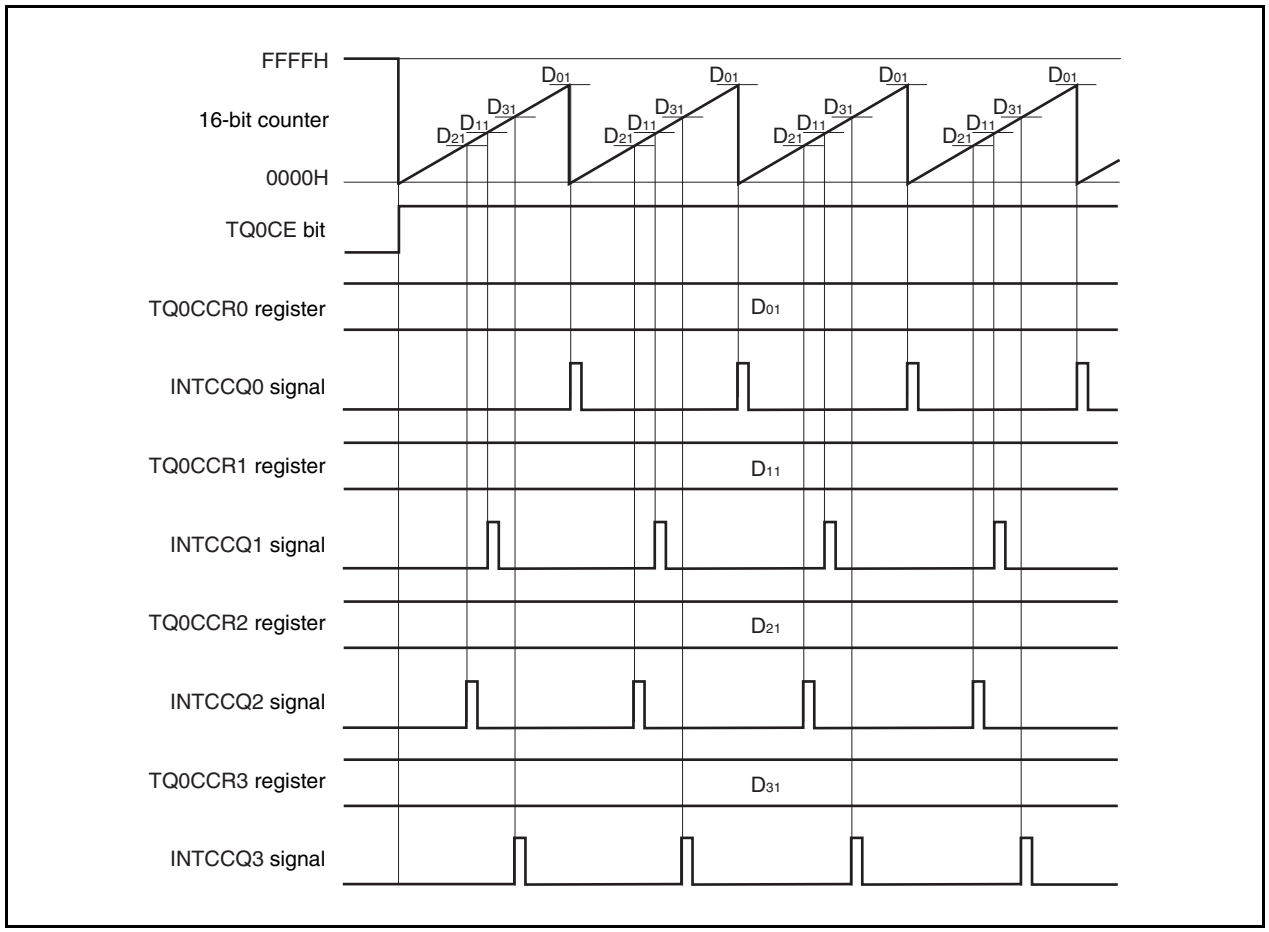
Figure 9-17. Configuration of TQ0CCR1 to TQ0CCR3 Registers



If the set value of the TQ0CCRb register is smaller than the set value of the TQ0CCR0 register, the INTCCQb signal is generated once per cycle.

Remark b = 1 to 3

Figure 9-18. Timing Chart When $D_{01} \geq D_{b1}$

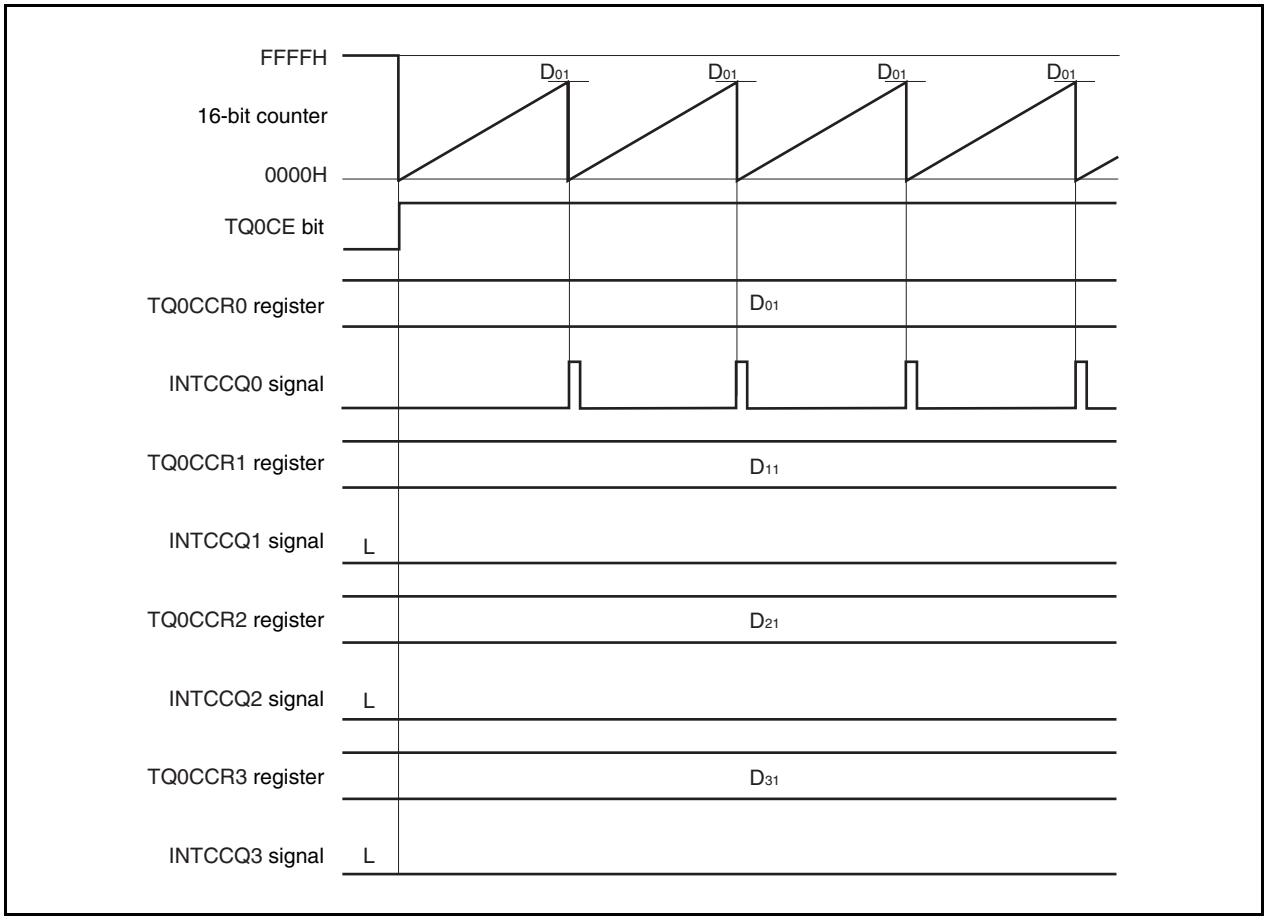


If the set value of the TQ0CCRb register is greater than the set value of the TQ0CCR0 register, the INTCCQb signal is not generated because the count value of the 16-bit counter and the value of the TQ0CCRb register do not match.

When the TQ0CCRb register is not used, it is recommended to set its value to FFFFH.

Remark b = 1 to 3

Figure 9-19. Timing Chart When $D_{01} < D_{b1}$



9.6.3 External trigger pulse output mode (TQ0MD2 to TQ0MD0 bits = 010)

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 the valid edge of an external trigger input signal (TIQ) is detected, 16-bit timer/event counter Q starts counting, and outputs a PWM waveform (up to 3-phase) from the TOQ1 to TOQ3 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 TOQ0 pin.

Pulses can also be output by generating a software trigger instead of using the external trigger.

Figure 9-20. Configuration in External Trigger Pulse Output Mode

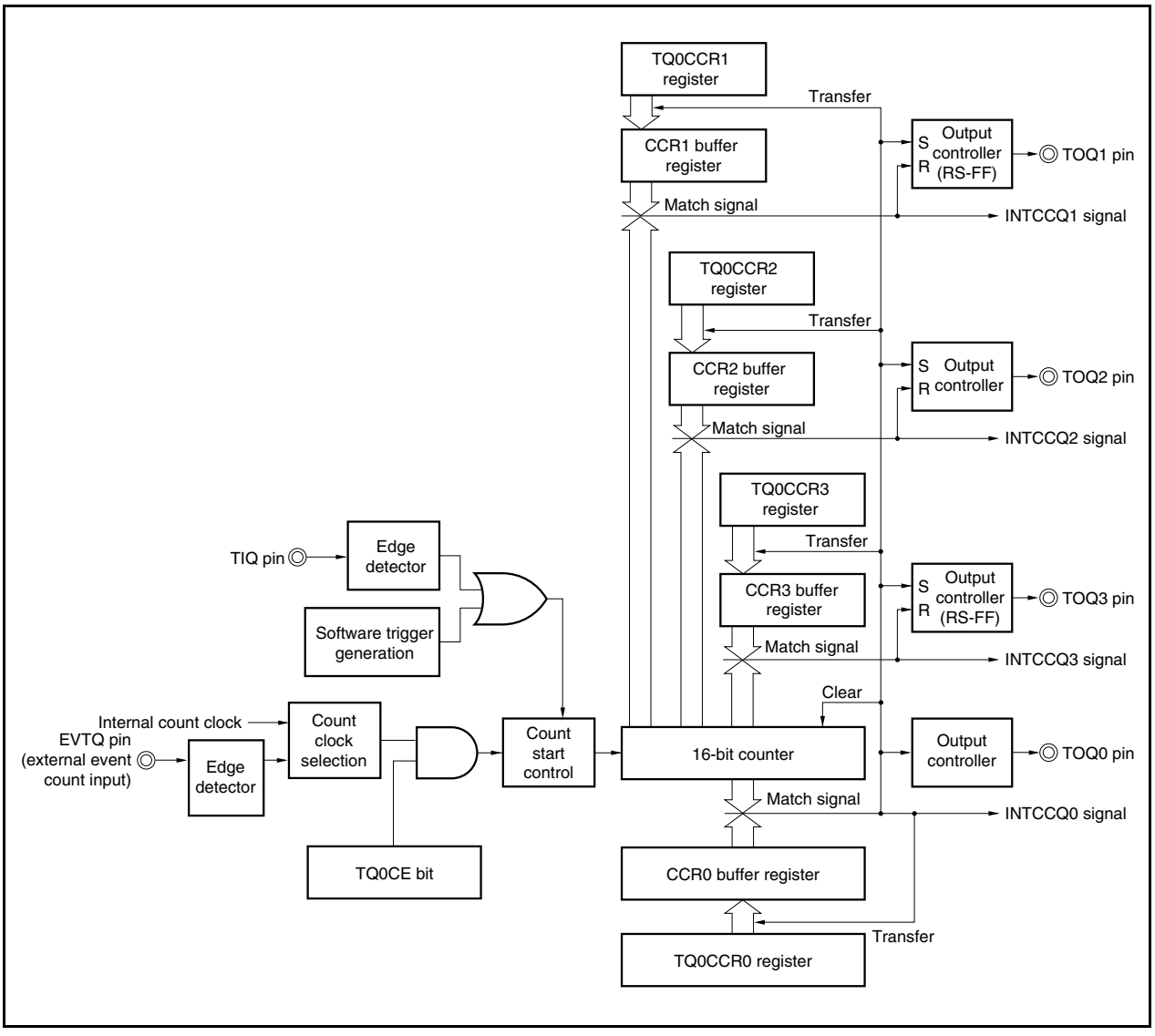
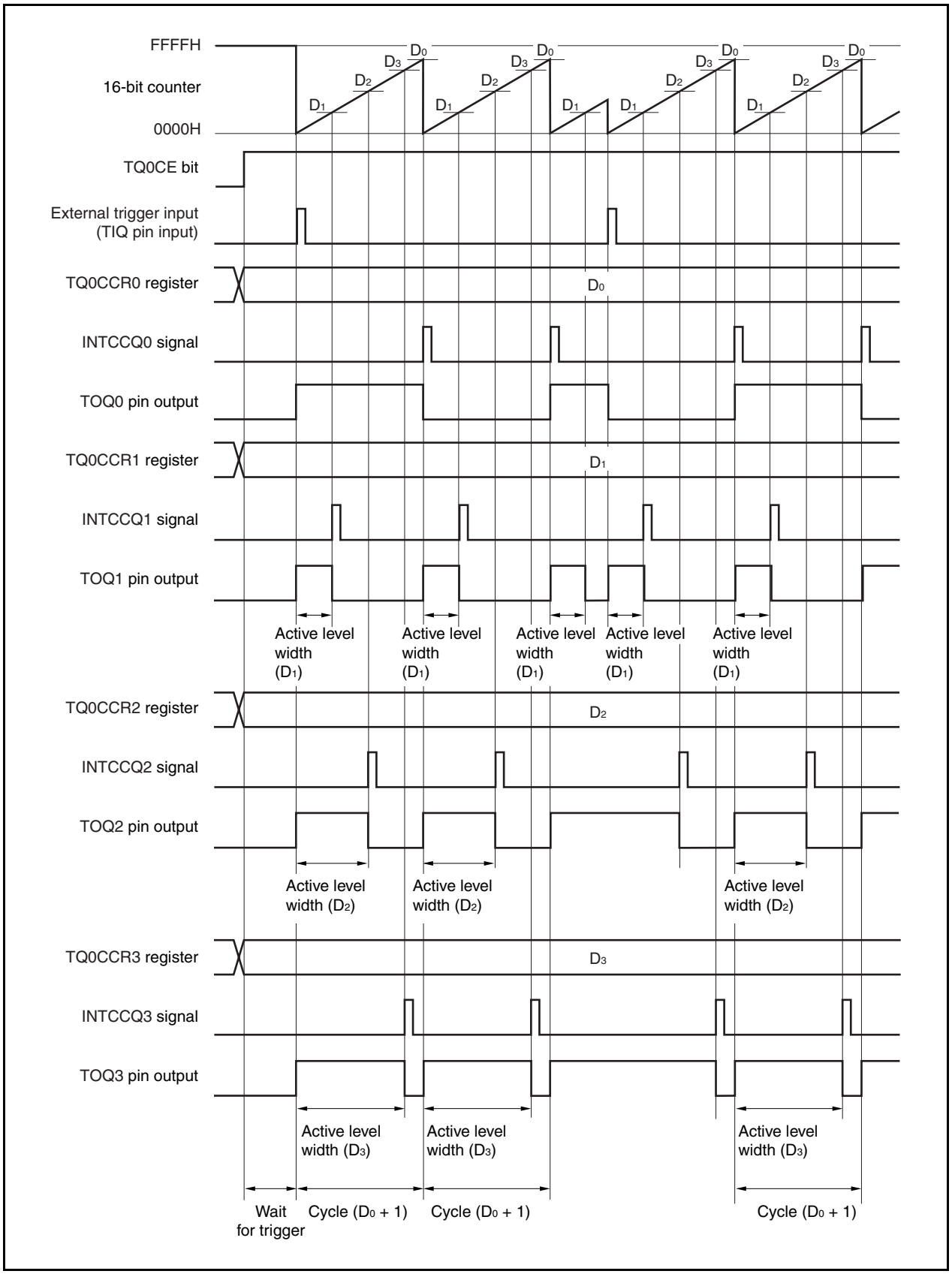


Figure 9-21. Basic Timing in External Trigger Pulse Output Mode

<R>



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 TOQb 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 TOQ0 pin is inverted. The TOQb pin outputs a high-level regardless of the status (high/low) when a trigger occurs.)

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

Active level width = (Set value of 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 request signal INTCCQ0 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 INTCCQb is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.

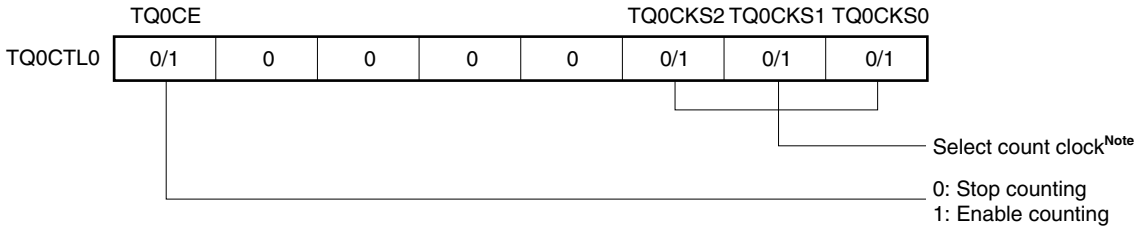
The value set to the TQ0CCRa register is transferred to the CCRa buffer register when the count value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

The valid edge of an external trigger input signal (TIQ), or setting the software trigger (TQ0CTL1.TQ0EST bit) to 1 is used as the trigger.

Remark a = 0 to 3
b = 1 to 3

Figure 9-22. 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 9-22. 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)

<R>

	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 Note

0: Disable TOQ0 pin output
1: Enable TOQ0 pin output

Setting of output level while operation of TOQ0 pin is disabled
0: Low level
1: High level

0: Disable TOQ1 pin output
1: Enable TOQ1 pin output

Setting of output level while operation of TOQ1 pin is disabled
0: Low level
1: High level

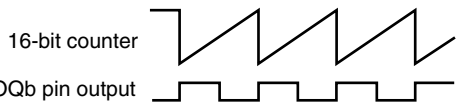
0: Disable TOQ2 pin output
1: Enable TOQ2 pin output

Setting of output level while operation of TOQ2 pin is disabled
0: Low level
1: High level

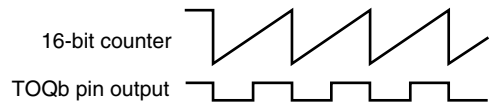
0: Disable TOQ3 pin output
1: Enable TOQ3 pin output

Setting of output level while operation of TOQ3 pin is disabled
0: Low level
1: High level

- When TQ0OLb bit = 0



- When TQ0OLb bit = 1

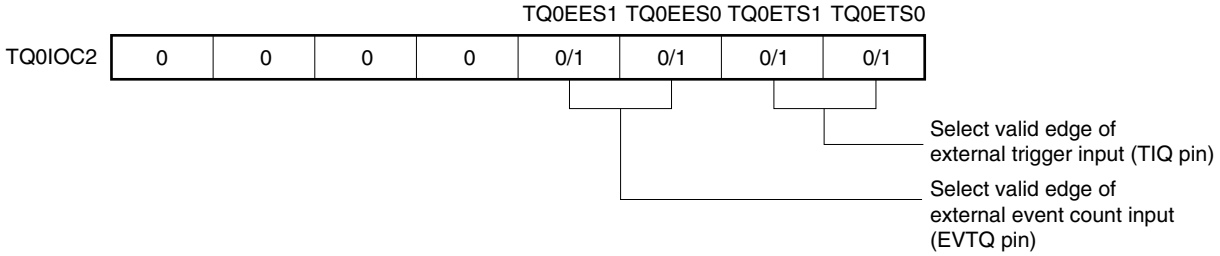


<R>

Note Clear this bit to 0 when the TOQ0 pin is not used in the external trigger pulse output mode.

Figure 9-22. 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.

Cycle = $(D_0 + 1) \times$ Count clock cycle
TOQ1 pin PWM waveform active level width = $D_1 \times$ Count clock cycle
TOQ2 pin PWM waveform active level width = $D_2 \times$ Count clock cycle
TOQ3 pin PWM waveform active level width = $D_3 \times$ Count clock cycle

Remark TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the external trigger pulse output mode.

(1) Operation flow in external trigger pulse output mode

Figure 9-23. Software Processing Flow in External Trigger Pulse Output Mode (1/2)

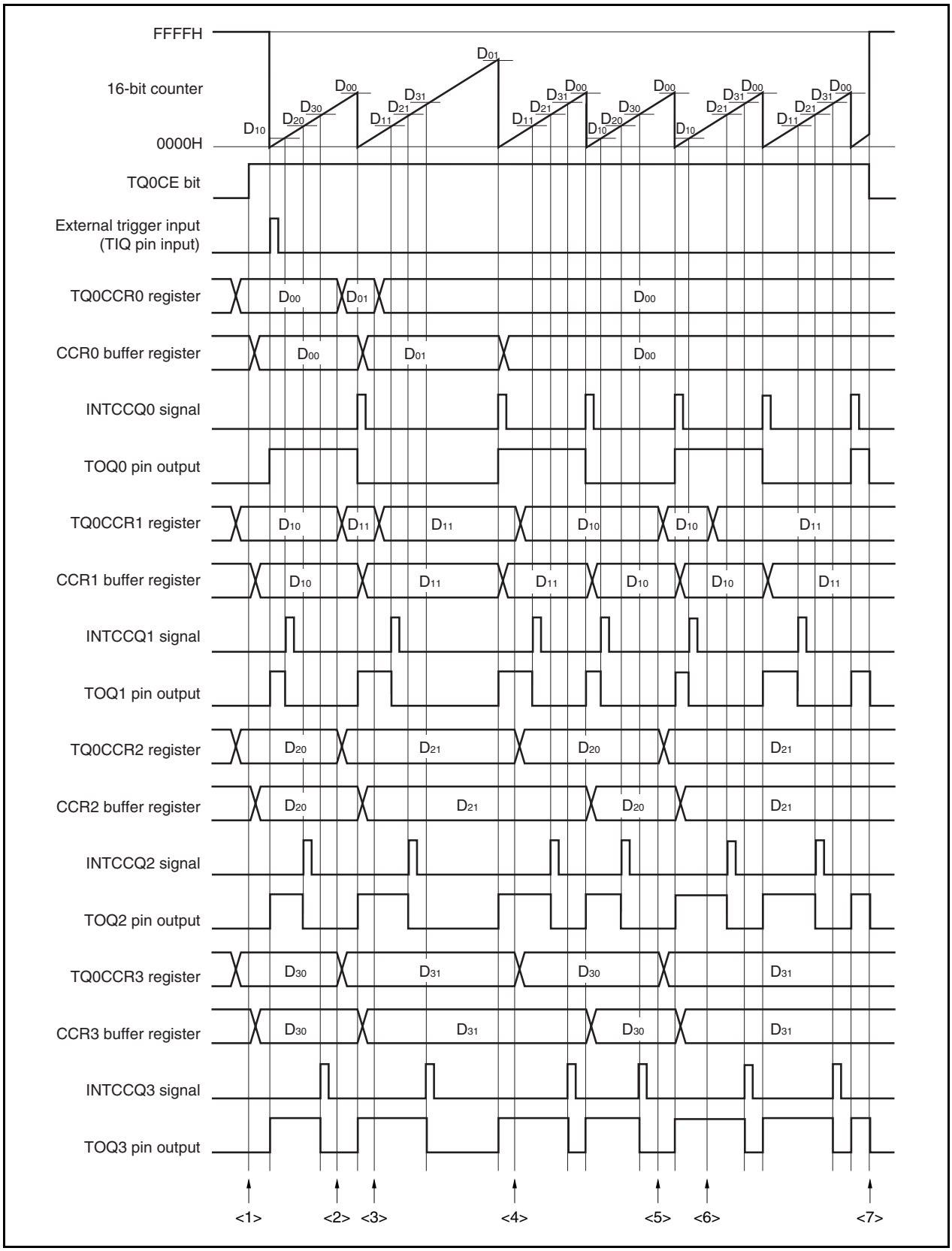
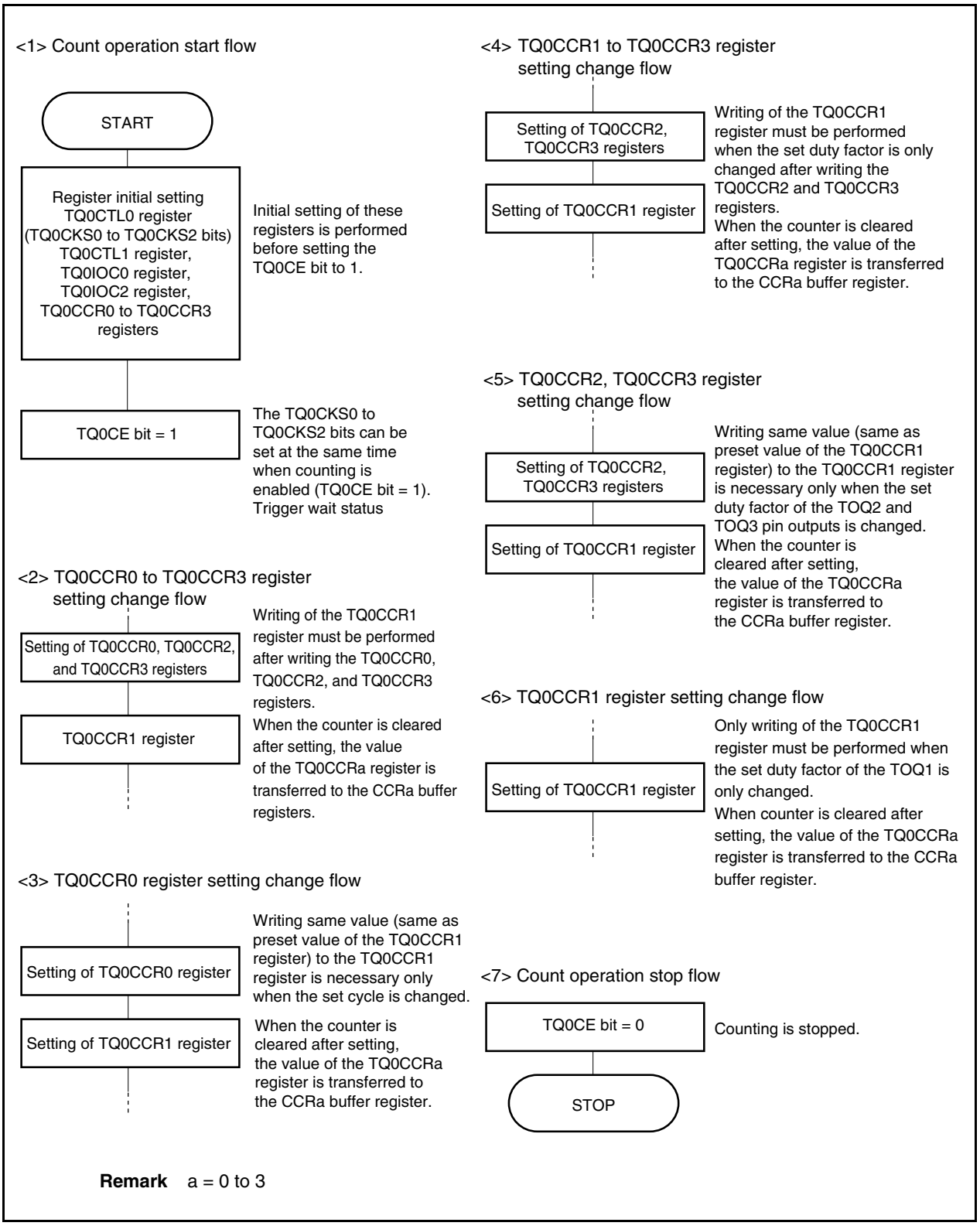


Figure 9-23. Software Processing Flow in External Trigger Pulse Output Mode (2/2)



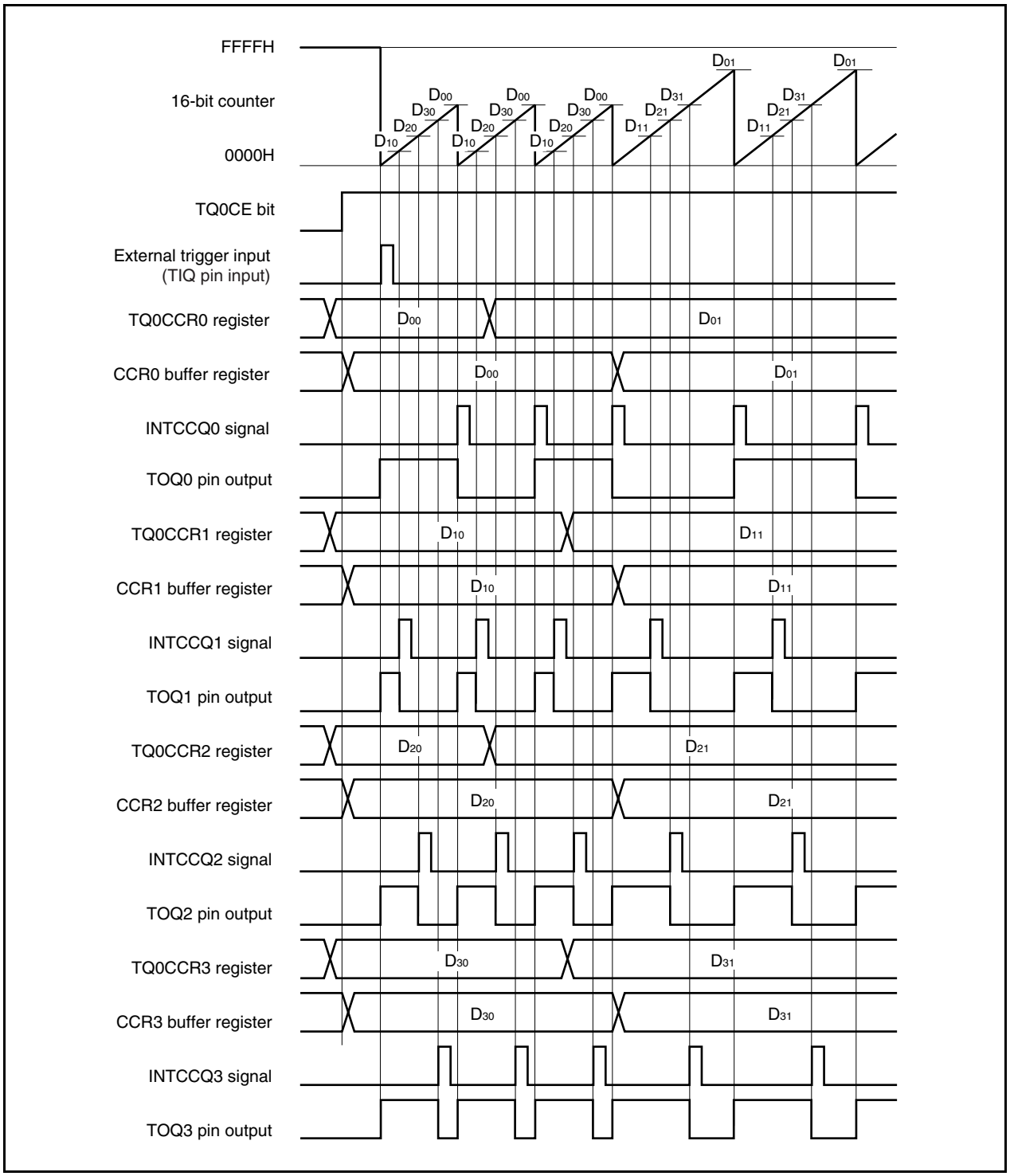
Remark a = 0 to 3

(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 INTCCQ0 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 TOQ1 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 TOQ2 and TOQ3 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.

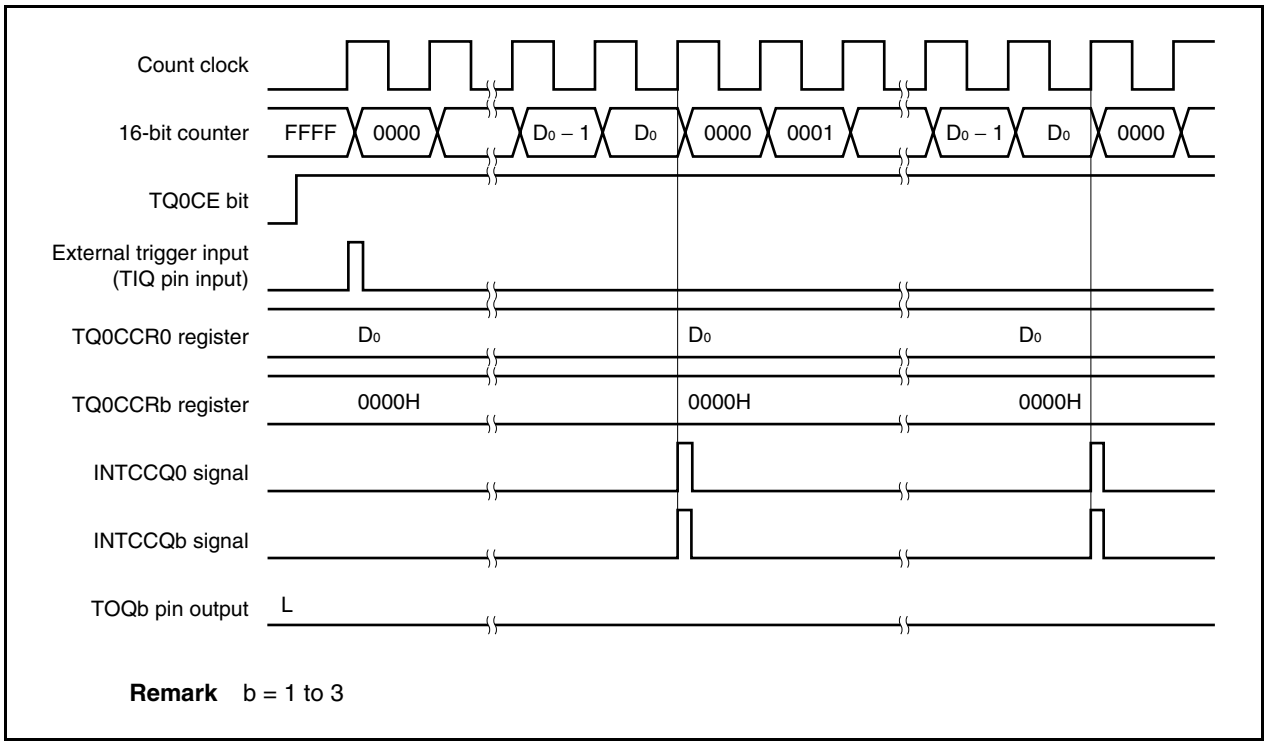
After data is written to the TQ0CCR1 register, the value written to the TQ0CCRa register is transferred to the CCRa buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TQ0CCR0 to TQ0CCR3 registers again after writing the TQ0CCR1 register once, do so after the INTCCQ0 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

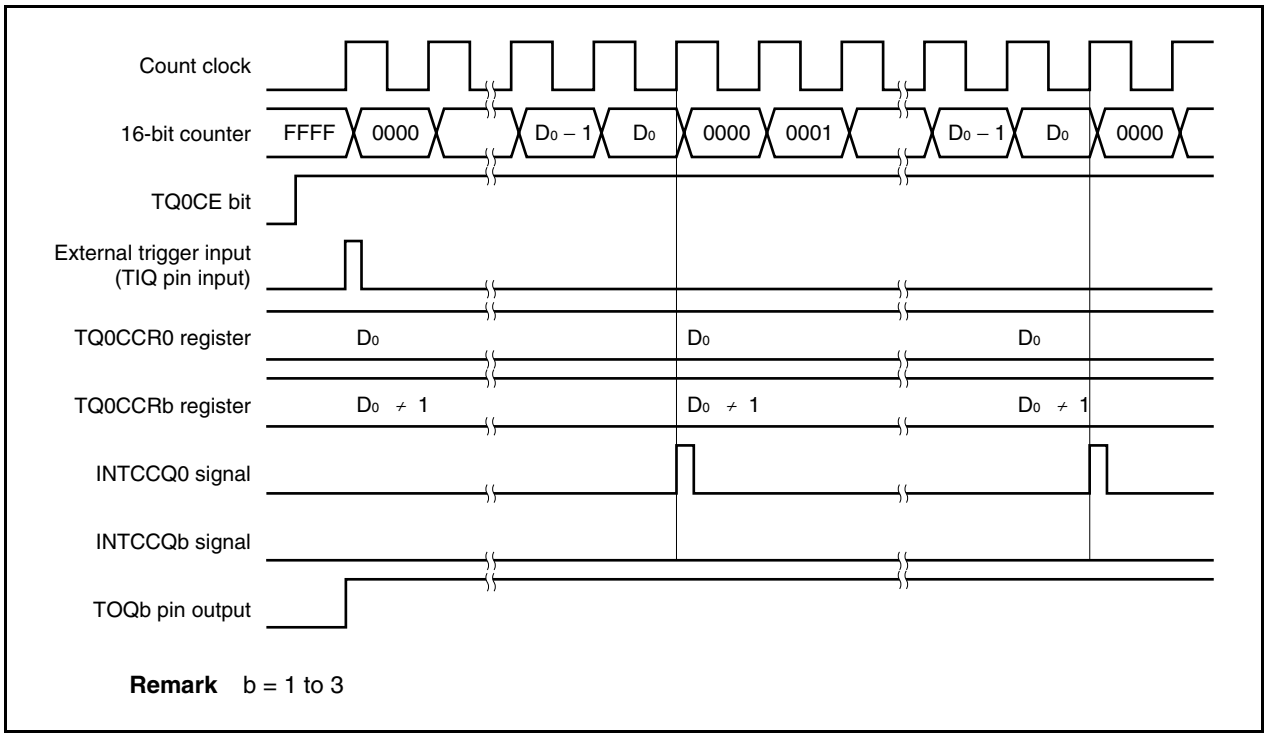
(b) 0%/100% output of PWM waveform

<R> To output a 0% waveform, set the TQ0CCRb register to 0000H. The 16-bit counter is cleared to 0000H and the INTCCQ0 and INTCCQb 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.



Remark b = 1 to 3

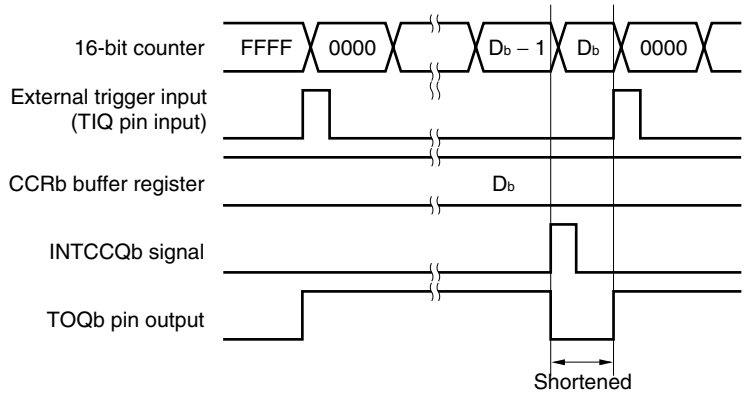
To output a 100% waveform, set a value of (set value of TQ0CCR0 register + 1) to the TQ0CCRb register. If the set value of the TQ0CCR0 register is FFFFH, 100% output cannot be produced.



Remark b = 1 to 3

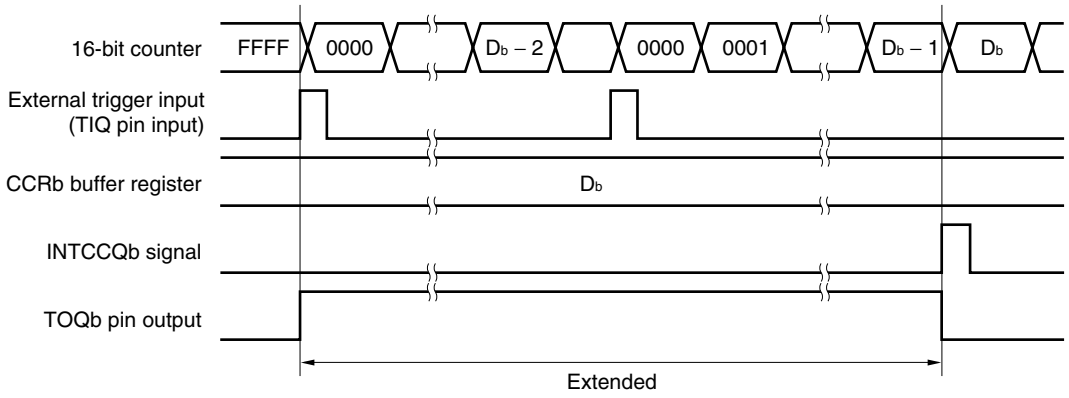
(c) Conflict between trigger detection and match with CCRb buffer register

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



Remark b = 1 to 3

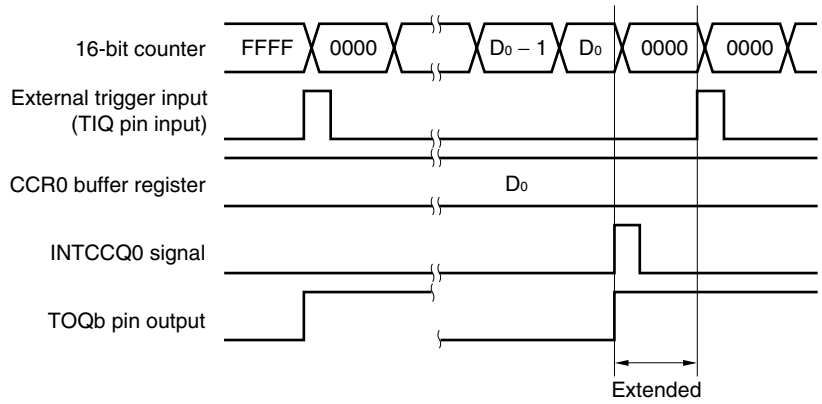
If the trigger is detected immediately before the INTCCQb signal is generated, the INTCCQb signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOQb pin remains active. Consequently, the active period of the PWM waveform is extended.



Remark b = 1 to 3

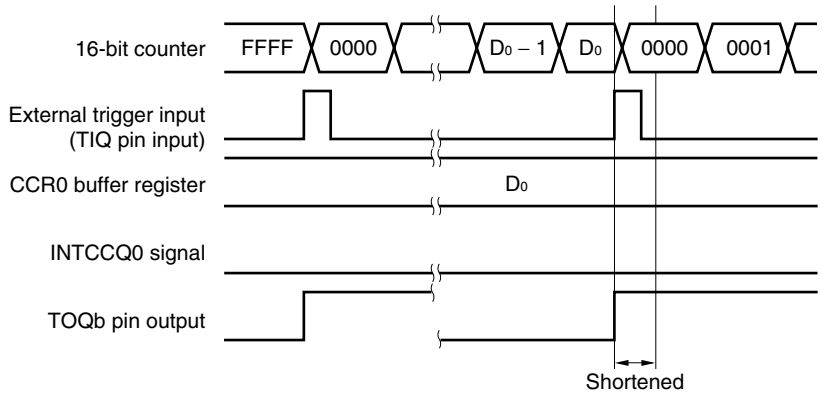
(d) Conflict between trigger detection and match with CCR0 buffer register

If the trigger is detected immediately after the INTCCQ0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOQb pin is extended by time from generation of the INTCCQ0 signal to trigger detection.



Remark b = 1 to 3

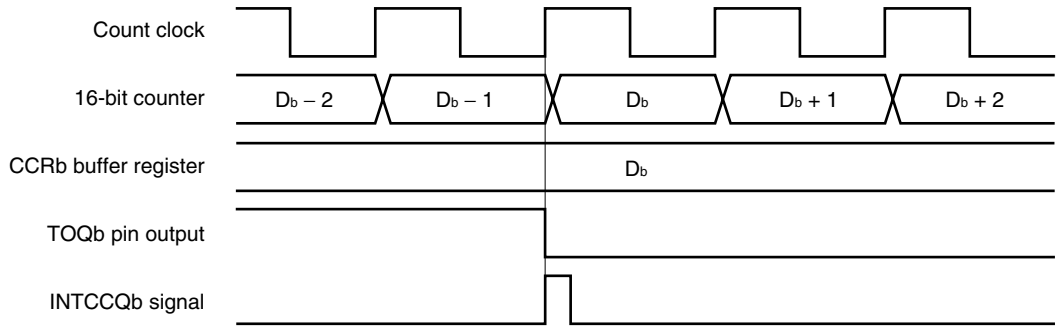
If the trigger is detected immediately before the INTCCQ0 signal is generated, the INTCCQ0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOQb pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark b = 1 to 3

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

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



Remark b = 1 to 3

Usually, the INTCCQb 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 TOQb pin.

9.6.4 One-shot pulse output mode (TQ0MD2 to TQ0MD0 bits = 011)

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 the valid edge of an external trigger input (TIQ) is detected, 16-bit timer/event counter Q starts counting, and outputs a one-shot pulse from the TOQ1 to TOQ3 pins.

Instead of the external trigger, a software trigger can also be generated to output the pulse. The TOQ0 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 9-24. Configuration in One-Shot Pulse Output Mode

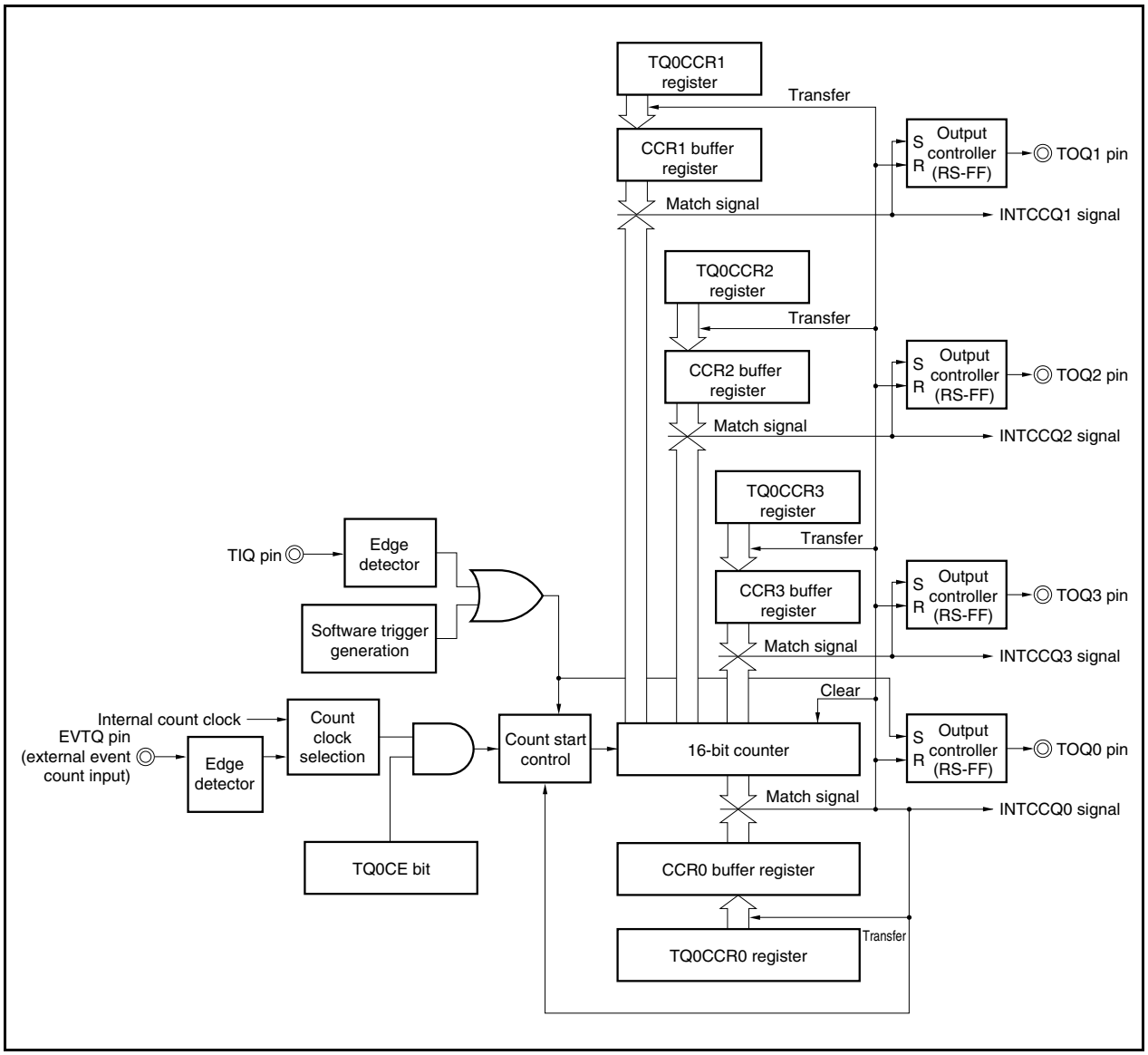
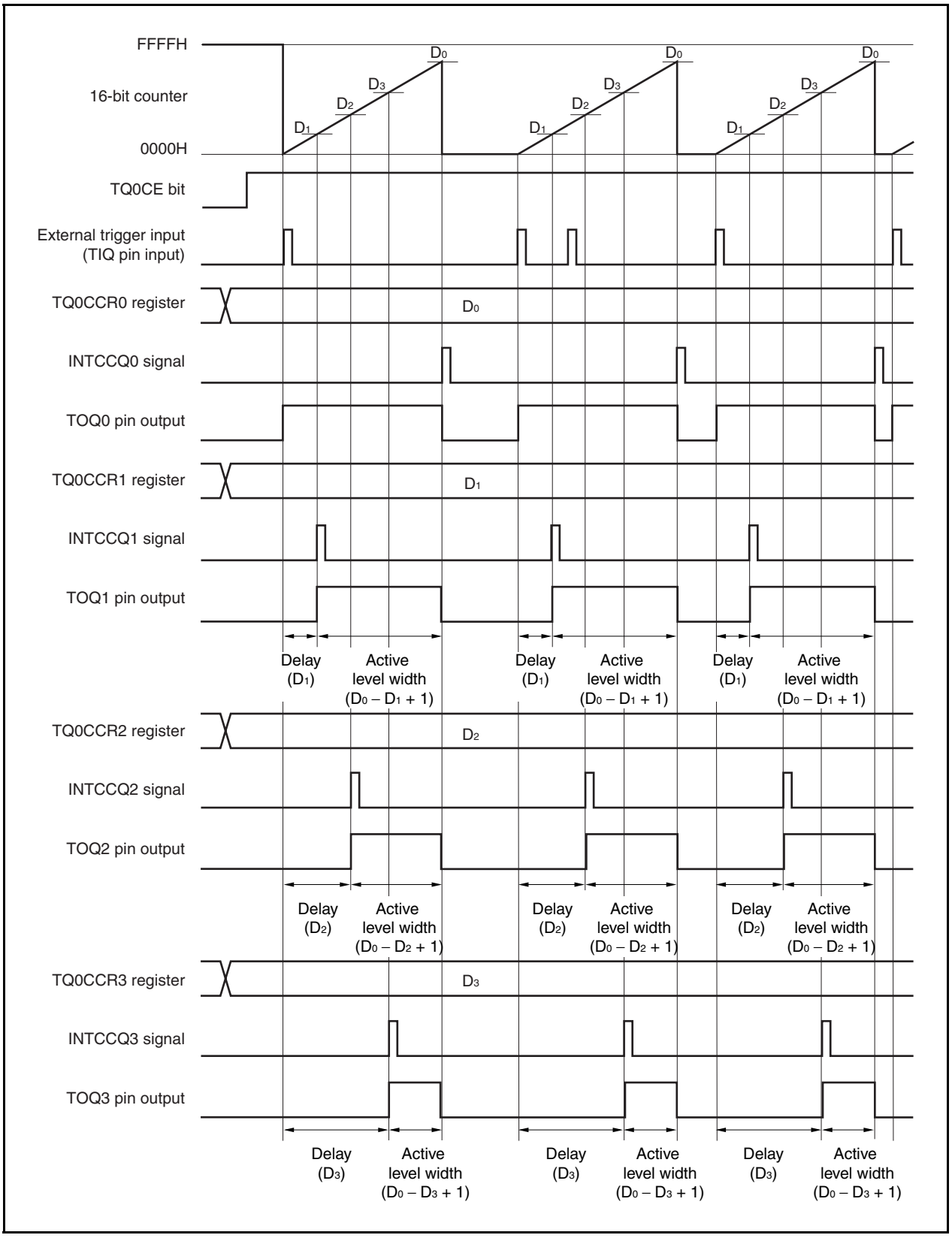


Figure 9-25. Basic Timing in One-Shot Pulse Output Mode



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 TOQb pin. After the one-shot pulse is output, the 16-bit counter is set to 0000H, stops counting, and waits for a trigger. When the trigger is generated again, the 16-bit counter starts counting from 0000H. If a trigger is generated again while the one-shot pulse is being output, it is ignored.

The output delay period and active level width of the one-shot pulse can be calculated as follows.

Output delay period = (Set value of TQ0CCRb register) × Count clock cycle

Active level width = (Set value of TQ0CCR0 register − Set value of TQ0CCRb register + 1) × Count clock cycle

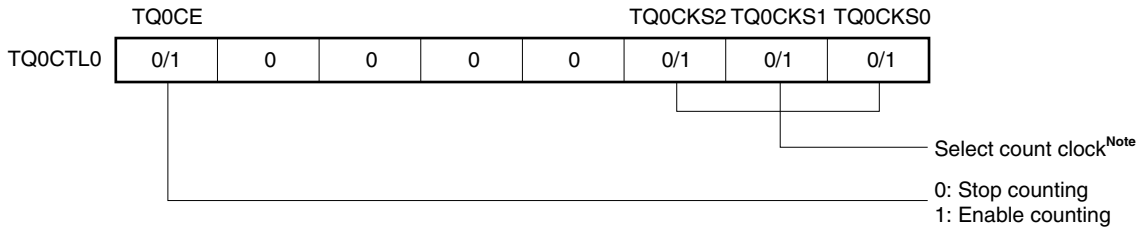
The compare match interrupt request signal INTCCQ0 is generated when the 16-bit counter counts after its count value matches the value of the CCR0 buffer register. The compare match interrupt request signal INTCCQb is generated when the count value of the 16-bit counter matches the value of the CCRb buffer register.

The valid edge of an external trigger input (TIQ pin) or setting the software trigger (TQ0CTL1.TQ0EST bit) to 1 is used as the trigger.

Remark b = 1 to 3

Figure 9-26. Setting of Registers 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)

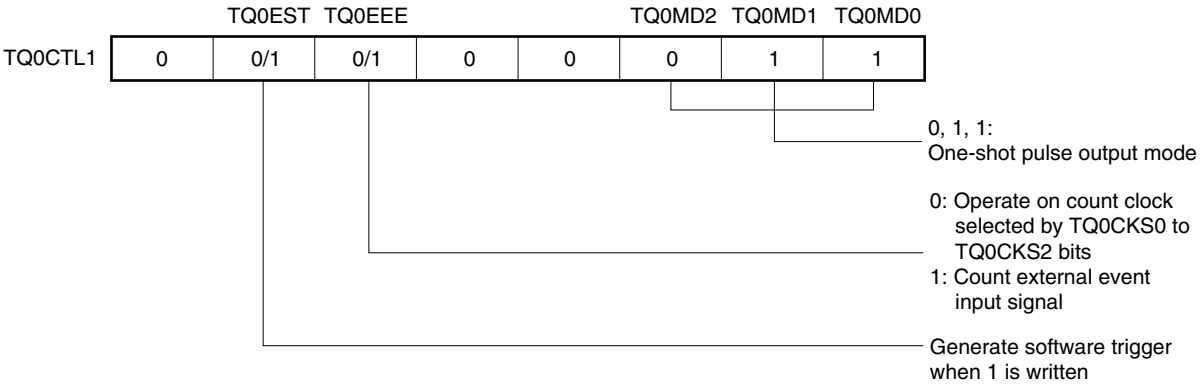


Figure 9-26. Register Setting in One-Shot Pulse Output Mode (2/3)

(c) TMQ0 I/O control register 0 (TQ0IOC0)

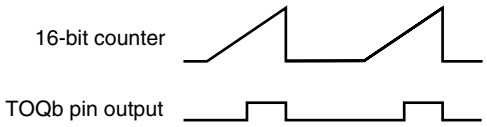
<R>

	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 Note

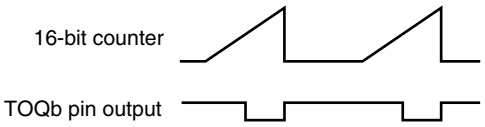
- TQ0OE0: 0: Disable TOQ0 pin output / 1: Enable TOQ0 pin output
- TQ0OL0: Setting of output level while operation of TOQ0 pin is disabled / 0: Low level / 1: High level
- TQ0OE1: 0: Disable TOQ1 pin output / 1: Enable TOQ1 pin output
- TQ0OL1: Setting of output level while operation of TOQ1 pin is disabled / 0: Low level / 1: High level
- TQ0OE2: 0: Disable TOQ2 pin output / 1: Enable TOQ2 pin output
- TQ0OL2: Setting of output level while operation of TOQ2 pin is disabled / 0: Low level / 1: High level
- TQ0OE3: 0: Disable TOQ3 pin output / 1: Enable TOQ3 pin output
- TQ0OL3: Setting of output level while operation of TOQ3 pin is disabled / 0: Low level / 1: High level

<R> **Note** Clear this bit to 0 when the TOQ0 pin is not used in the one-shot pulse output mode.

• When TQ0OLb bit = 0



• When TQ0OLb bit = 1



(d) TMQ0 I/O control register 2 (TQ0IOC2)

					TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0
TQ0IOC2	0	0	0	0	0/1	0/1	0/1	0/1

- TQ0ETS0: Select valid edge of external trigger input (TIQ pin)
- TQ0EES0: Select valid edge of external event count input (EVTQ pin)

(e) TMQ0 counter read buffer register (TQ0CNT)

The value of the 16-bit counter can be read by reading the TQ0CNT register.

Figure 9-26. 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.

Active level width = $(D_b - D_0 + 1) \times$ Count clock cycle

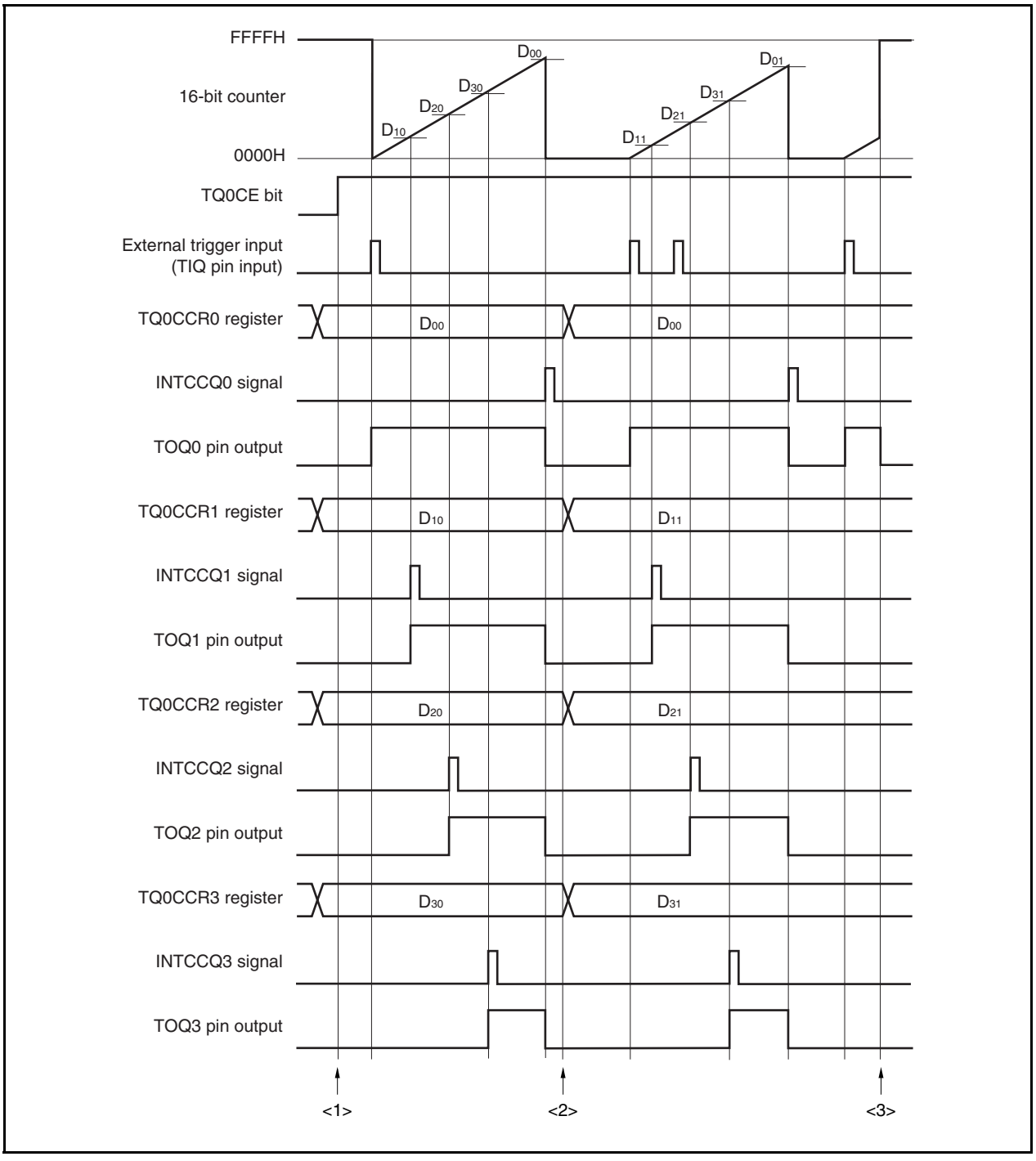
Output delay period = $D_b \times$ Count clock cycle

Remarks 1. TMQ0I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the one-shot pulse output mode.

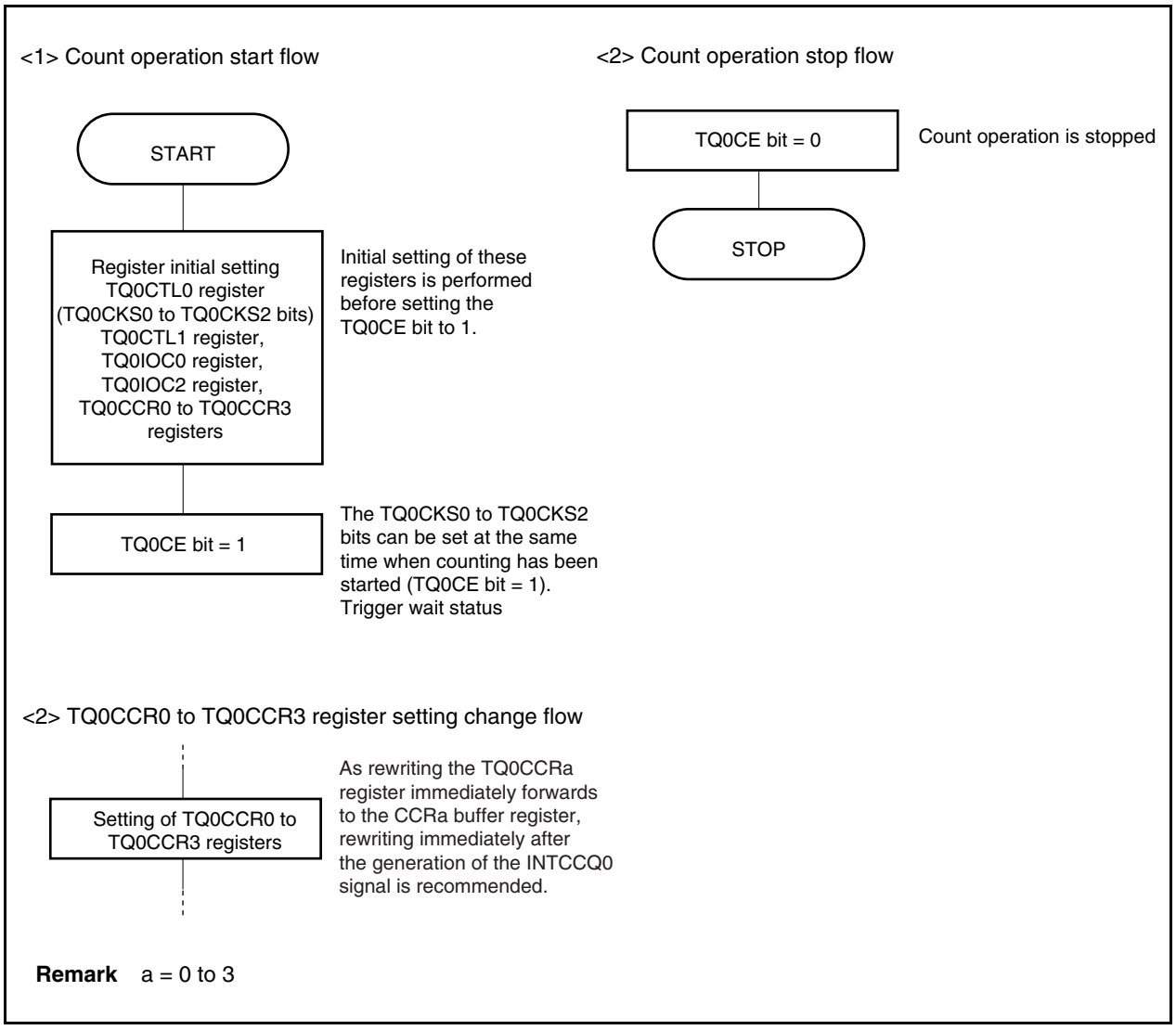
2. b = 1 to 3

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

<R> **Figure 9-27. Software Processing Flow in One-Shot Pulse Output Mode (1/2)**



<R> **Figure 9-27. Software Processing Flow in One-Shot Pulse Output Mode (2/2)**

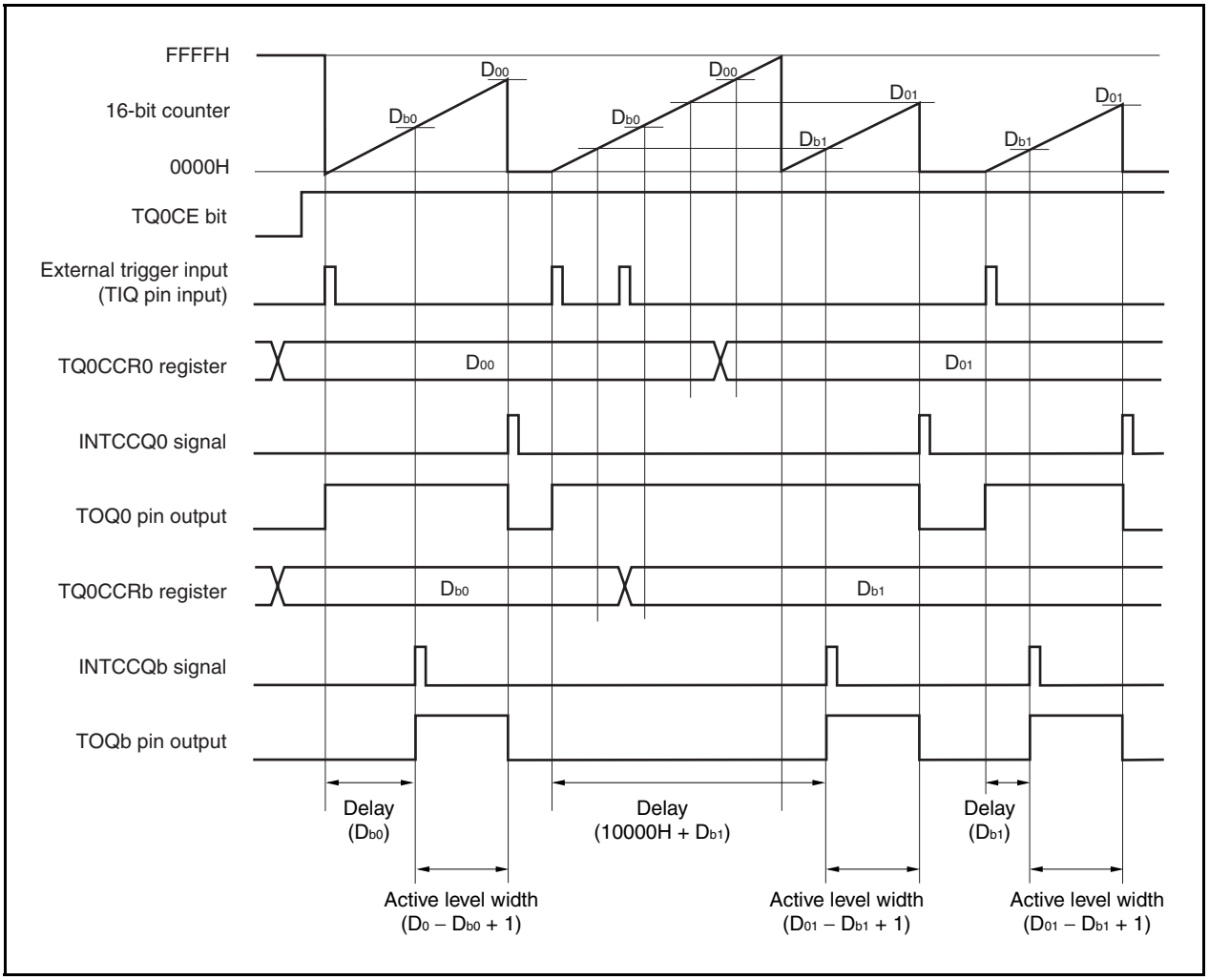


Remark a = 0 to 3

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

(a) Note on rewriting TQ0CCRa register

To change the set value of the TQ0CCRa register to a smaller value, stop counting once, and then change the set value. When the overflow may occur, stop counting once, and then change the set value.



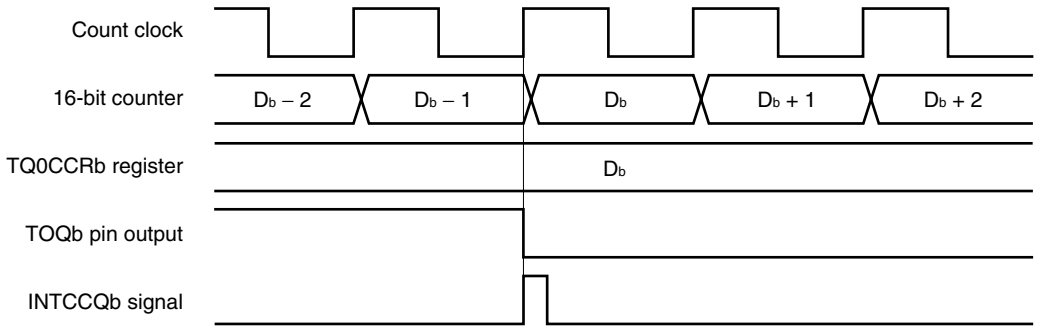
When the TQ0CCR0 register is rewritten from D_{00} to D_{01} and the TQ0CCRb register from D_{b0} to D_{b1} where $D_{00} > D_{01}$ and $D_{b0} > D_{b1}$, if the TQ0CCRb register is rewritten when the count value of the 16-bit counter is greater than D_{b1} and less than D_{b0} and if the TQ0CCR0 register 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 INTCCQb signal and asserts the TOQb pin. When the count value matches D_{01}, the counter generates the INTCCQ0 signal, deasserts the TOQb 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 (INTCCQb)

The generation timing of the INTCCQb signal in the one-shot pulse output mode is different from INTCCQb signals in other mode; the INTCCQb signal is generated when the count value of the 16-bit counter matches the value of the TQ0CCRb register.



Remark b = 1 to 3

Usually, the INTCCQb 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 TOQb pin.

9.6.5 PWM output mode (TQ0MD2 to TQ0MD0 bits = 100)

In the PWM output mode, a PWM waveform is output from the TOQ1 to TOQ3 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 TOQ0 pin.

Figure 9-28. Configuration in PWM Output Mode

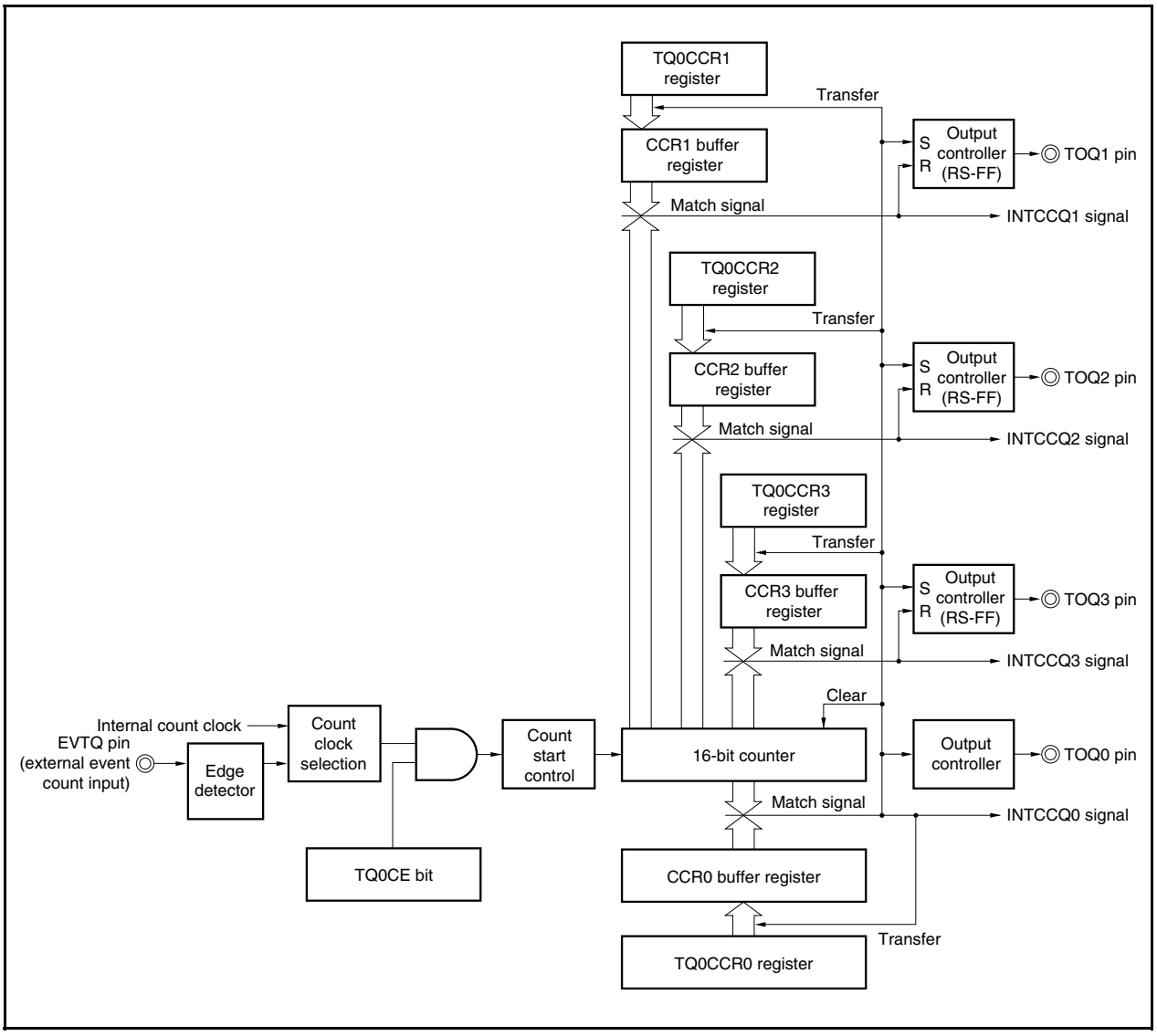
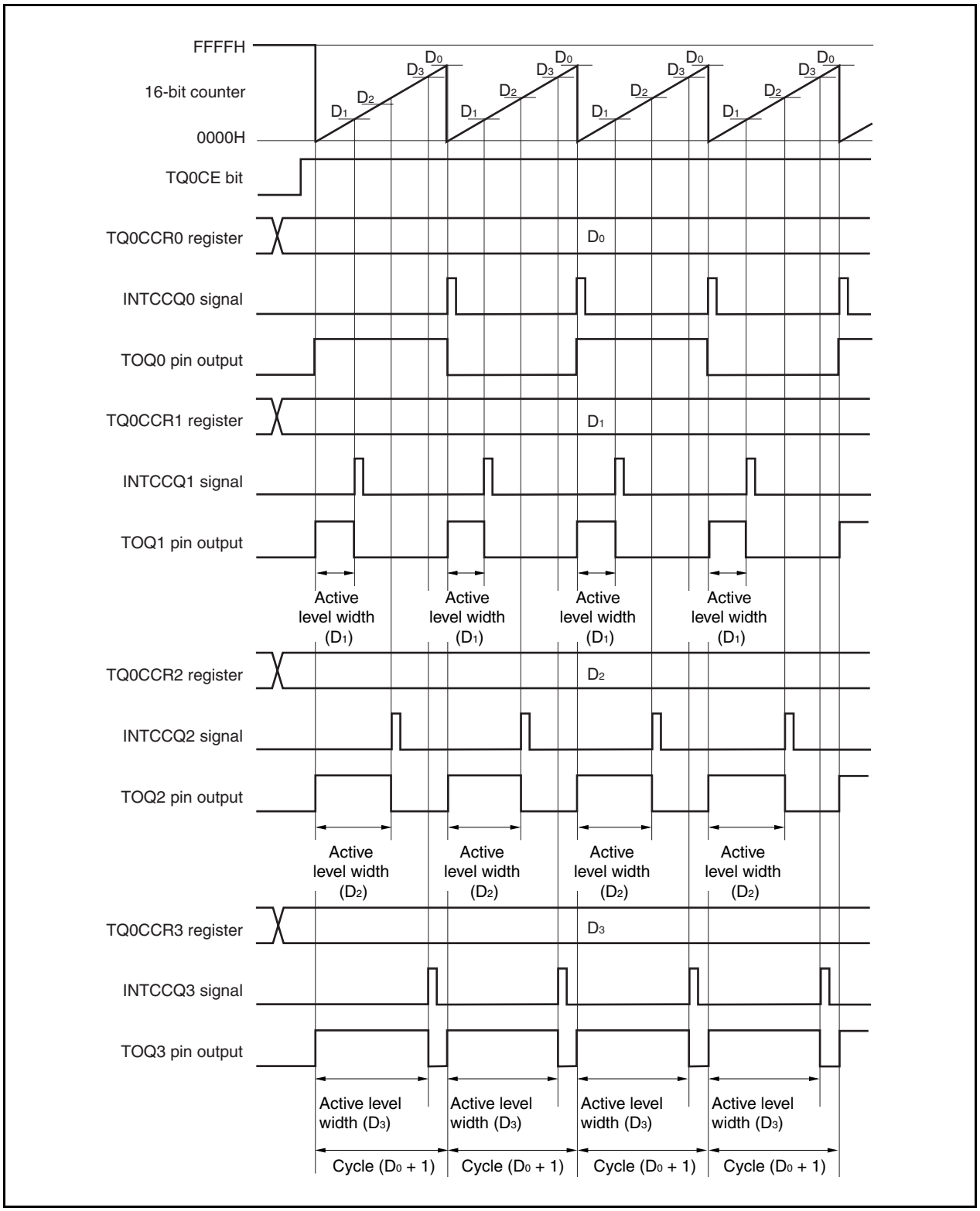


Figure 9-29. Basic Timing in PWM Output Mode



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 TOQb pin.

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)

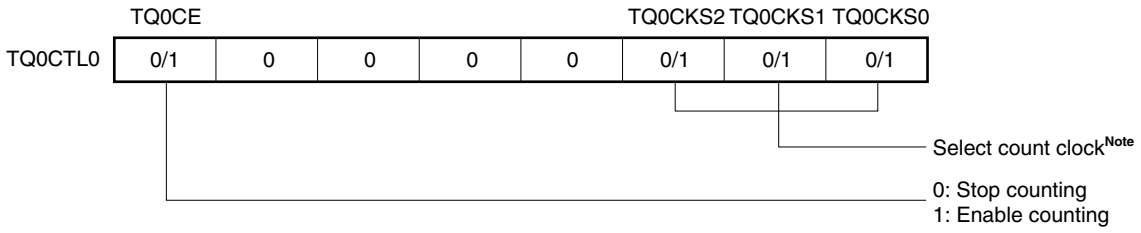
The PWM waveform can be changed by rewriting the TQ0CCRa register while the counter is operating. The newly written value is reflected when the count value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

The compare match interrupt request signal INTCCQ0 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 INTCCQb 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 9-30. Setting of Registers 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)

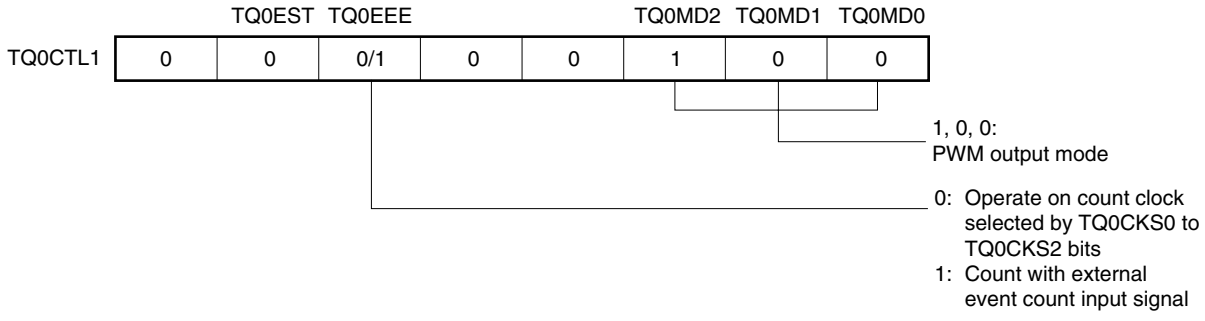
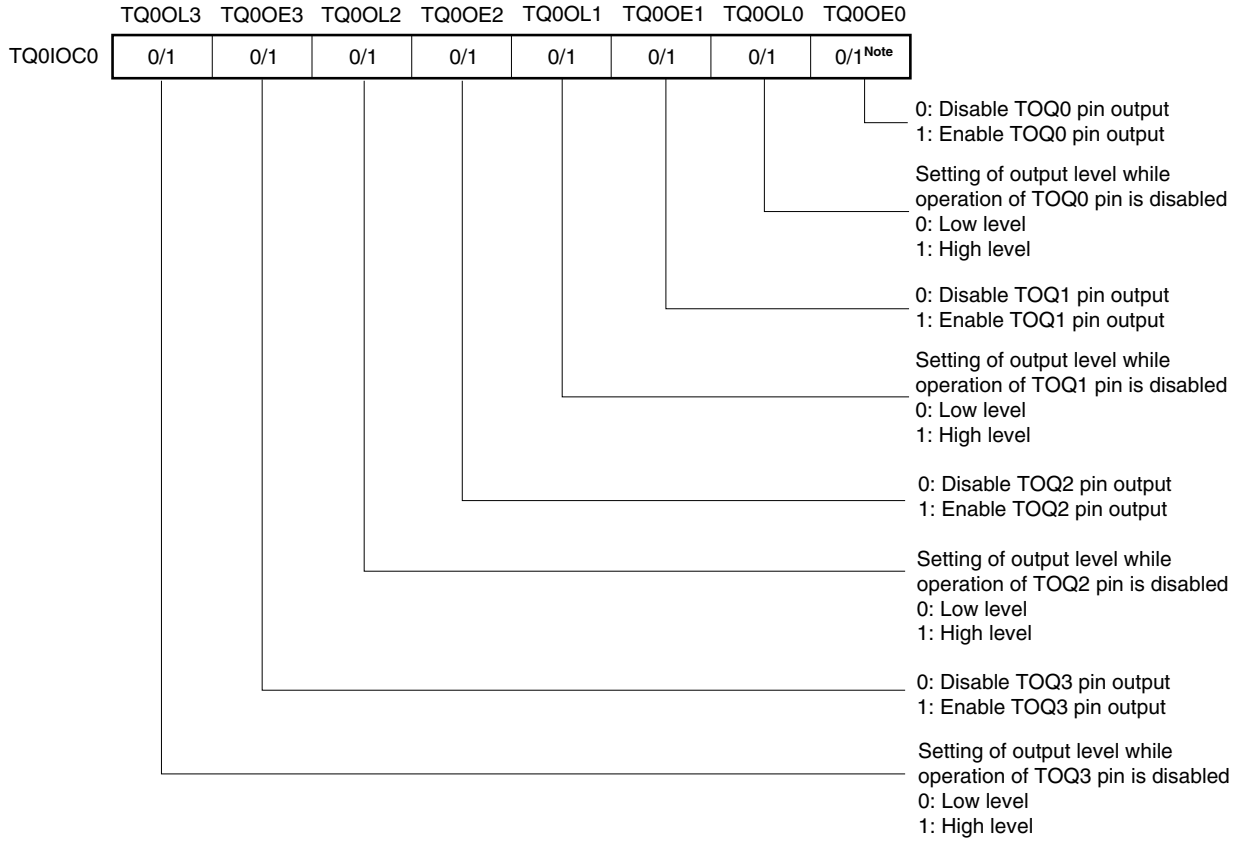


Figure 9-30. Setting of Registers in PWM Output Mode (2/3)

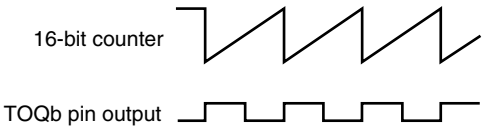
(c) TMQ0 I/O control register 0 (TQ0IOC0)

<R>

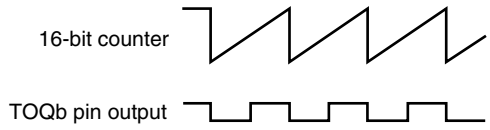


<R> **Note** Clear this bit to 0 when the TOQ0 pin is not used in the PWM output mode.

- When TQ0OLb bit = 0



- When TQ0OLb bit = 1



(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 9-30. 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

Remark TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the PWM output mode.

(1) Operation flow in PWM output mode

Figure 9-31. Software Processing Flow in PWM Output Mode (1/2)

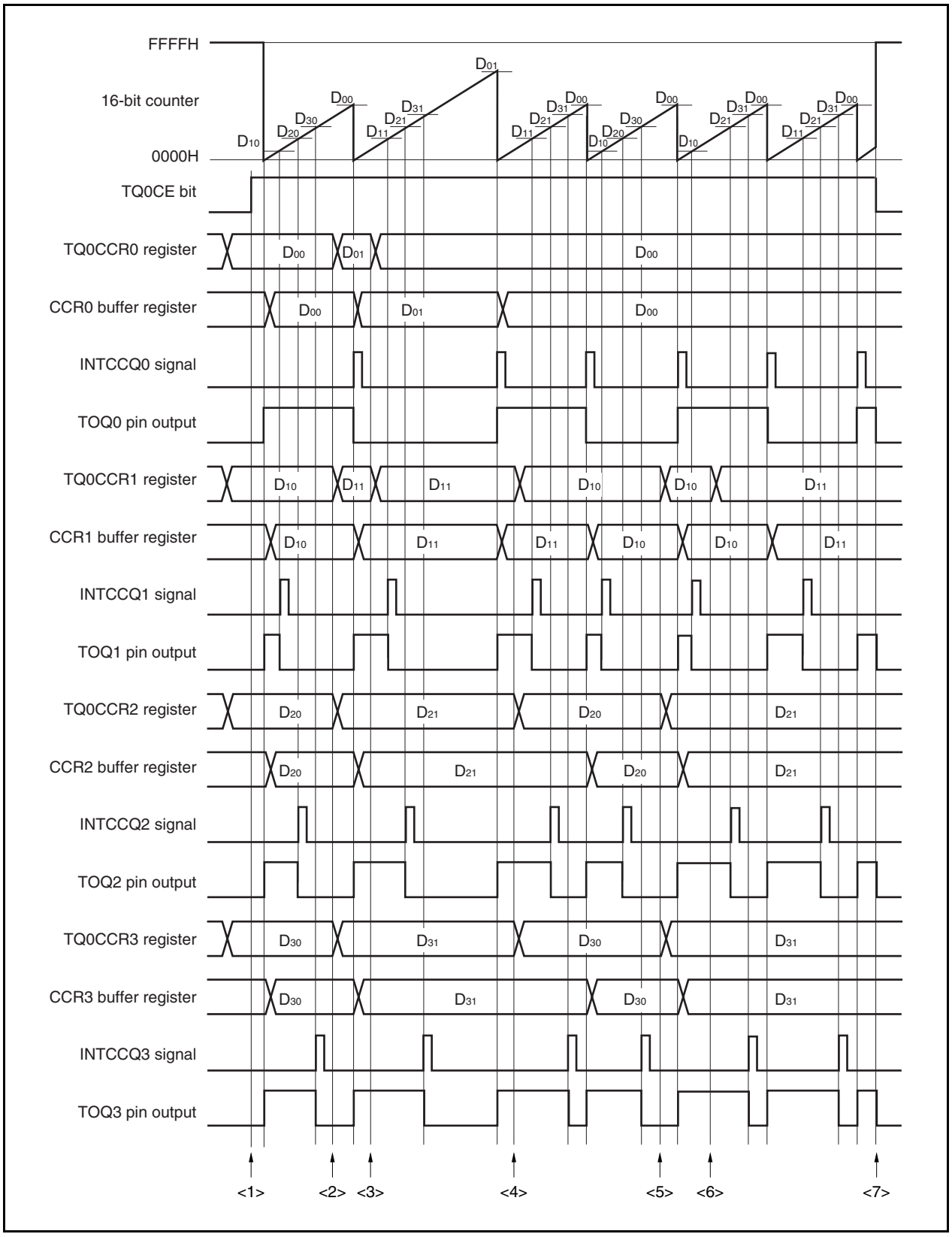
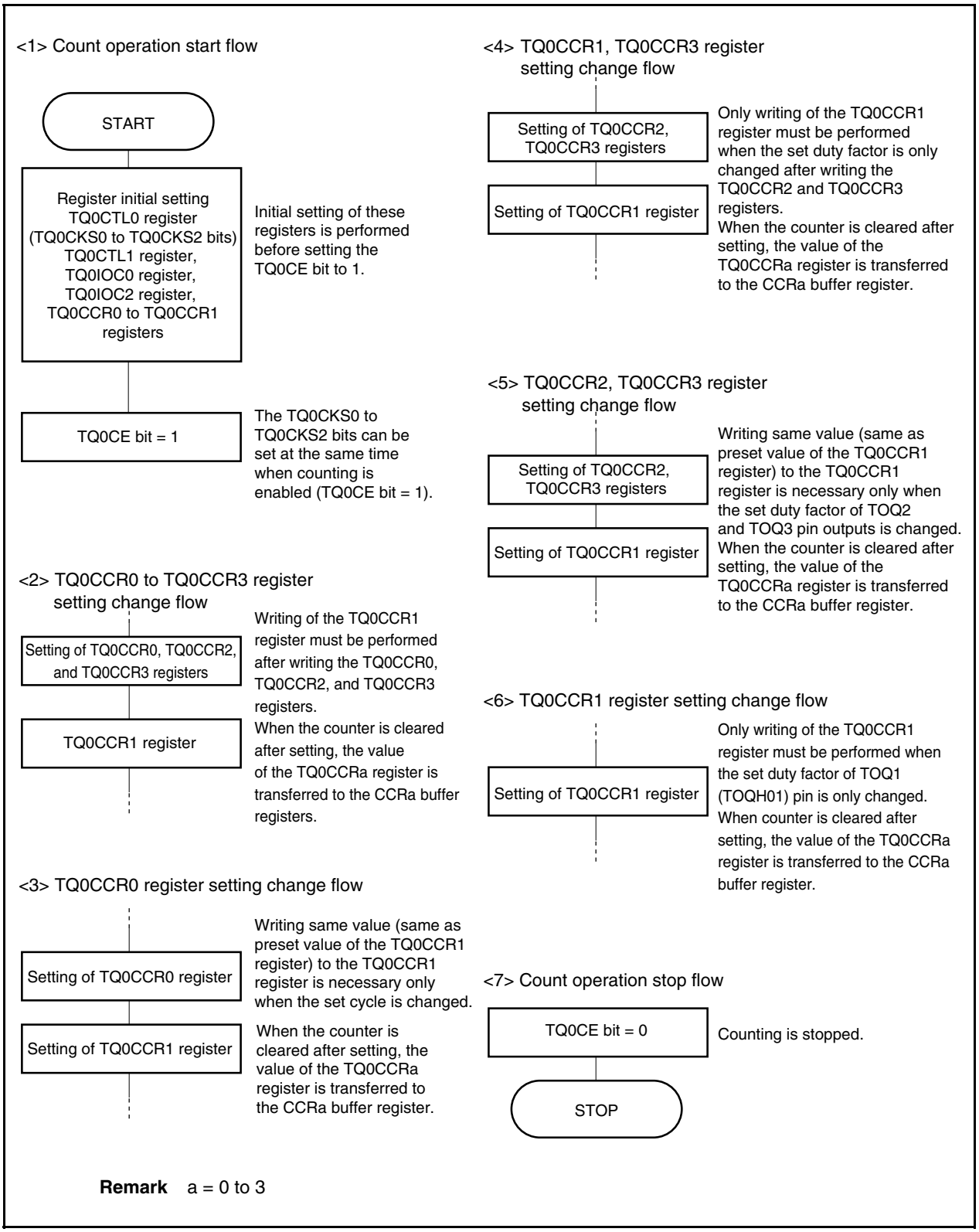


Figure 9-31. Software Processing Flow in PWM Output Mode (2/2)



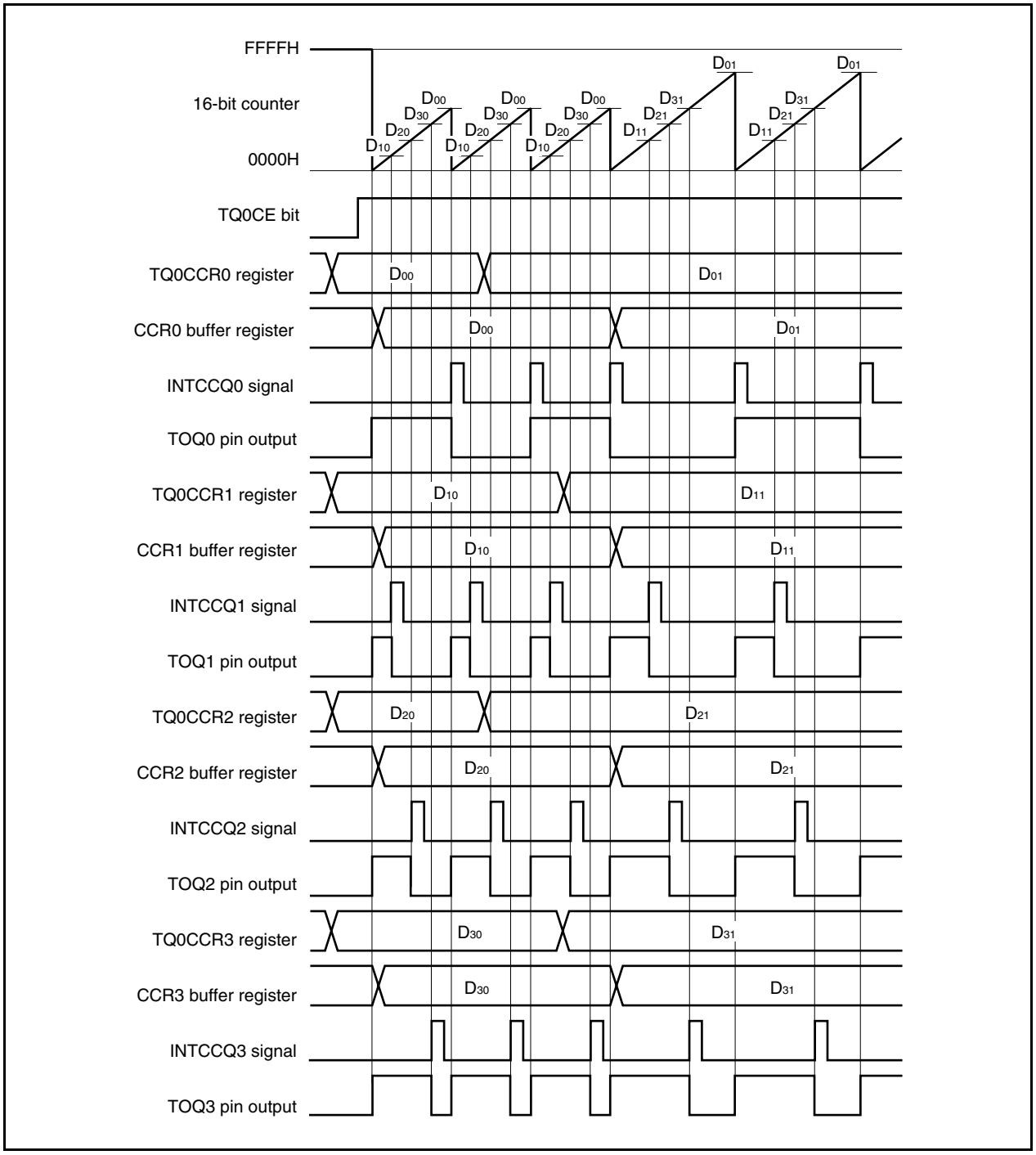
Remark a = 0 to 3

(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 INTCCQ1 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 TOQ1 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 TOQ2 and TOQ3 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.

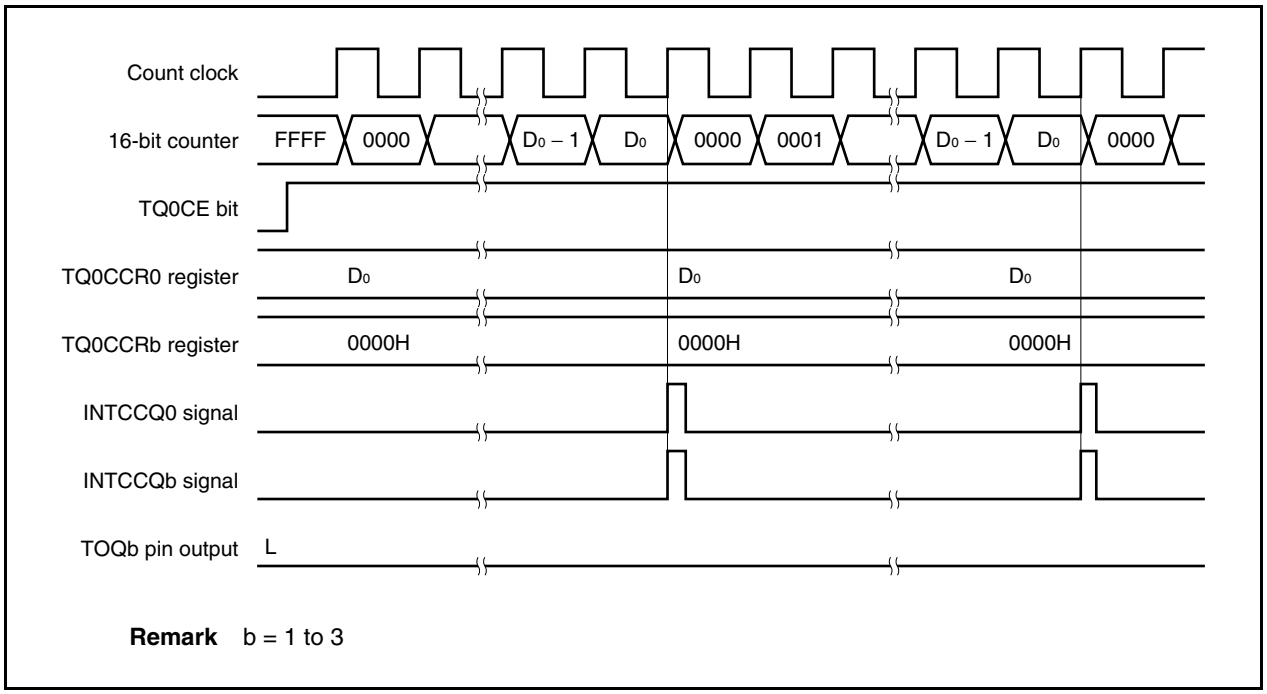
After the TQ0CCR1 register is written, the value written to the TQ0CCRa register is transferred to the CCRa buffer register in synchronization with the timing of clearing the 16-bit counter, and is used as a value to be compared with the value of the 16-bit counter.

To write the TQ0CCR0 to TQ0CCR3 registers again after writing the TQ0CCR1 register once, do so after the INTCCQ0 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

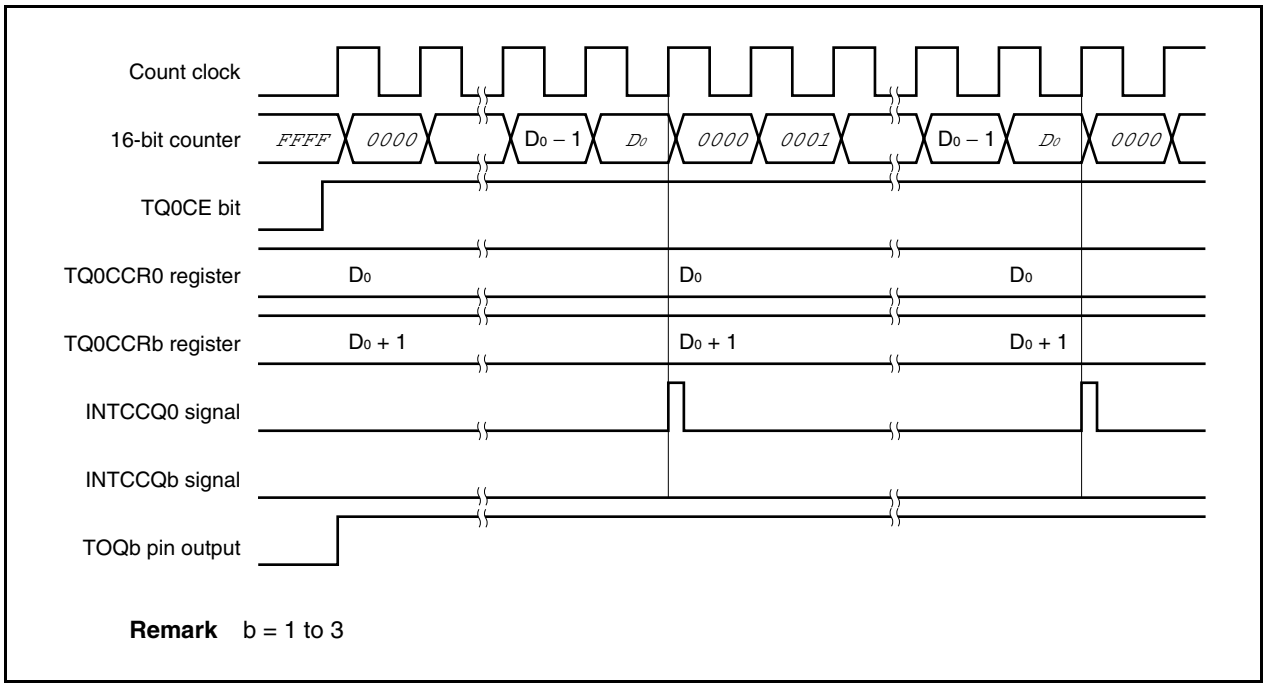
(b) 0%/100% output of PWM waveform

<R> To output a 0% waveform, set the TQ0CCRb register to 0000H. The 16-bit counter is cleared to 0000H and the INTCCQ0 and INTCCQb 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.



Remark b = 1 to 3

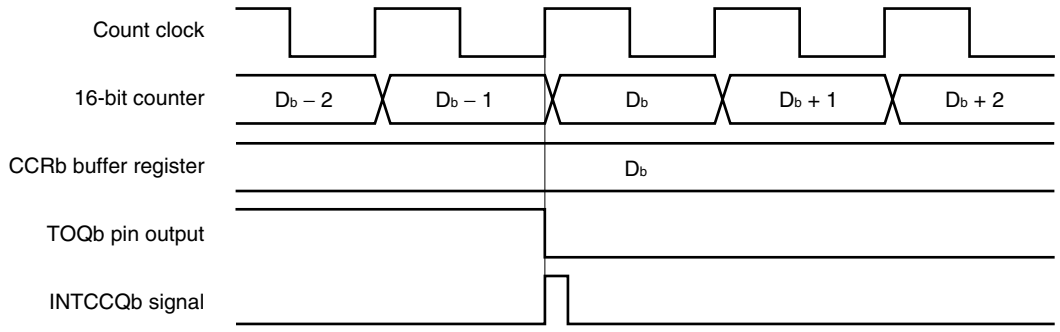
To output a 100% waveform, set a value of (set value of TQ0CCR0 register + 1) to the TQ0CCRb register. If the set value of the TQ0CCR0 register is FFFFH, 100% output cannot be produced.



Remark b = 1 to 3

(c) Generation timing of compare match interrupt request signal (INTCCQb)

The timing of generation of the INTCCQb signal in the PWM output mode differs from the timing of INTCCQb signals in other mode; the INTCCQb signal is generated when the count value of the 16-bit counter matches the value of the TQ0CCRb register.



Remark b = 1 to 3

Usually, the INTCCQb 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 TOQb pin.

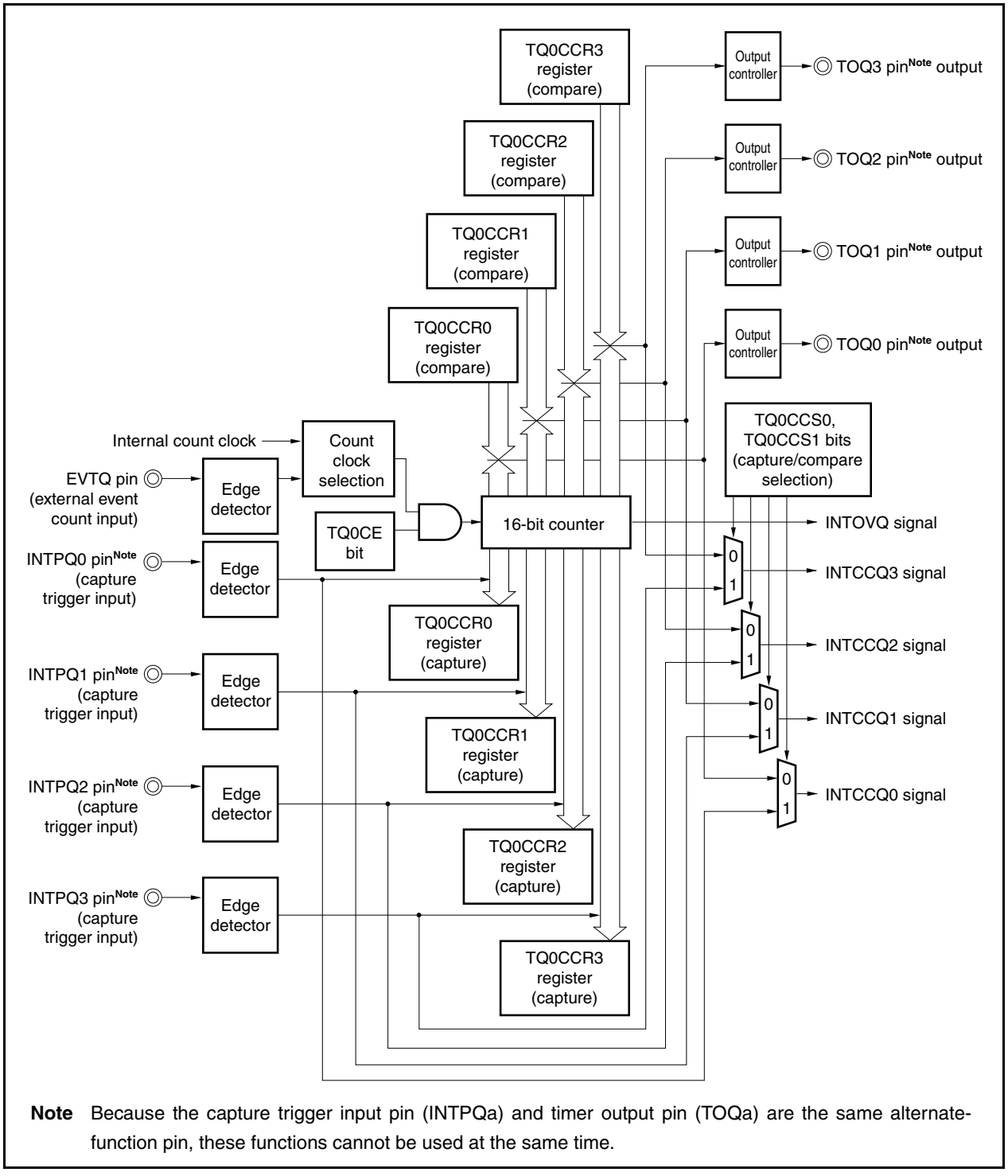
9.6.6 Free-running timer mode (TQ0MD2 to TQ0MD0 bits = 101)

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 TQ0CTL0.TQ0CE 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.

Remark a = 0 to 3

Figure 9-32. Configuration in Free-Running Timer Mode



Note Because the capture trigger input pin (INTPQa) and timer output pin (TOQa) are the same alternate-function pin, these functions cannot be used at the same time.

- Compare operation

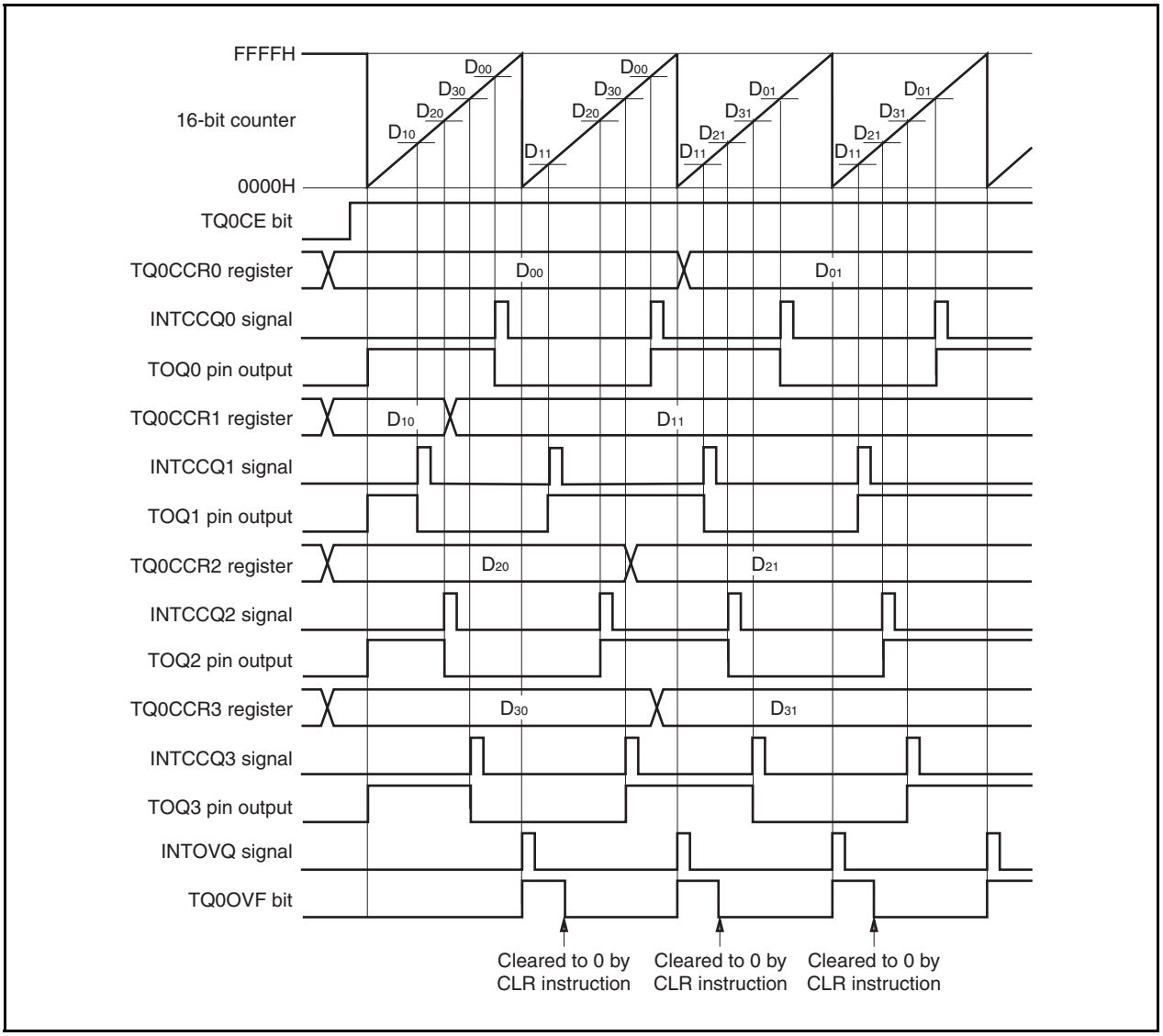
  When the TQ0CE bit is set to 1, 16-bit timer/event counter Q starts counting, and the output signals of the TOQ0 to TOQ3 pins are inverted. When the count value of the 16-bit counter later matches the set value of the TQ0CCRa register, a compare match interrupt request signal (INTCCQa) is generated, and the output signal of the TOQa pin is inverted.

  The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTOVQ) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TQ0OPT0.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.

  The TQ0CCRa register can be rewritten while the counter is operating. If it is rewritten, the new value is reflected at that time, and compared with the count value.

  Remark a = 0 to 3

Figure 9-33. Basic Timing in Free-Running Timer Mode (Compare Function)



- Capture operation

  When the TQ0CE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the INTPQa pin is detected, the count value of the 16-bit counter is stored in the TQ0CCRa register, and a capture interrupt request signal (INTCCQa) 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 (INTOVQ) 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.

Remark a = 0 to 3

Figure 9-34. Basic Timing in Free-Running Timer Mode (Capture Function)

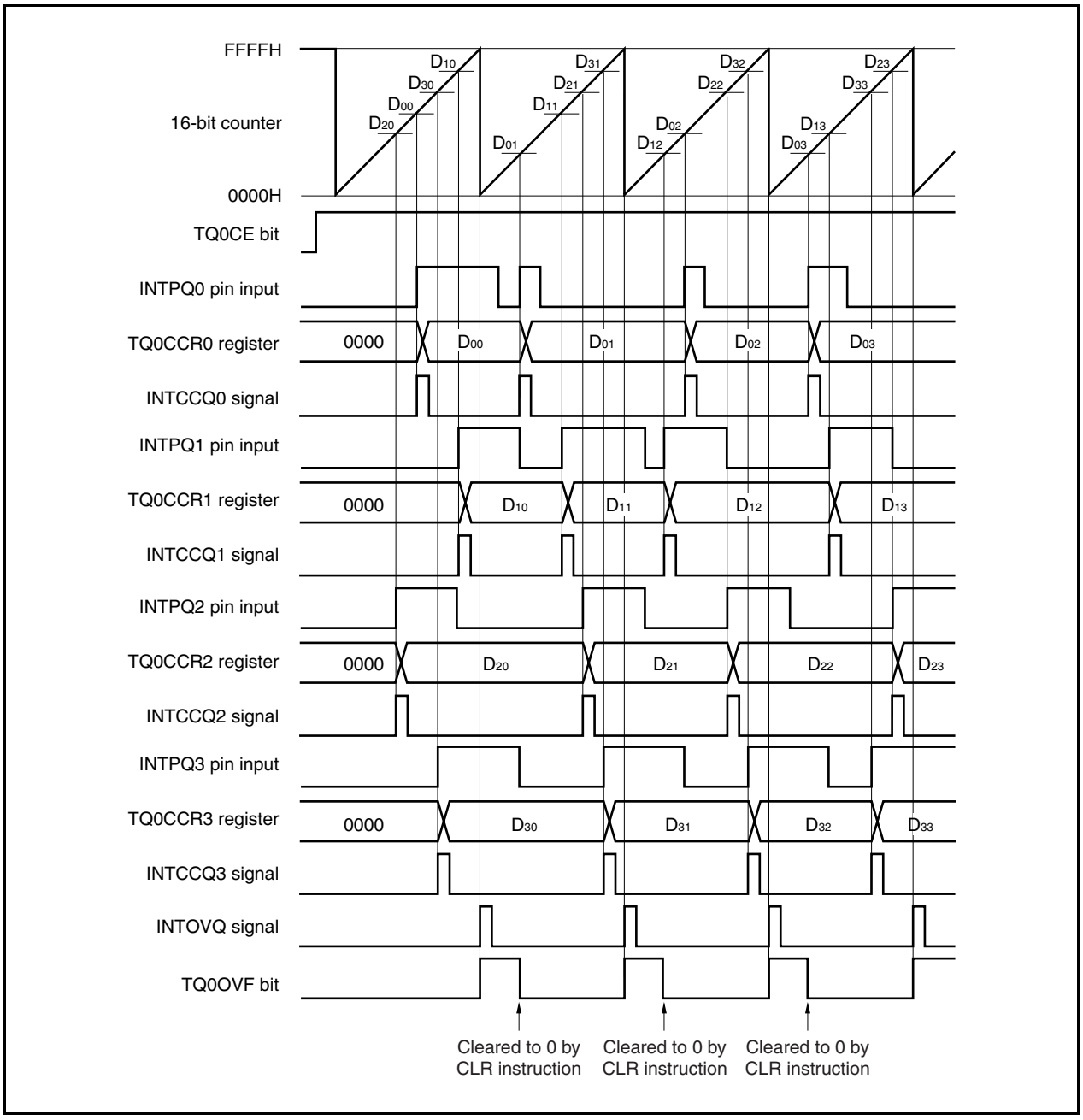
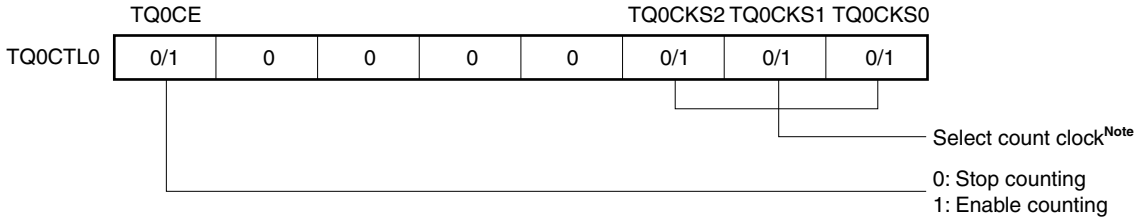


Figure 9-35. Register Setting in Free-Running Timer Mode (1/3)

(a) TMQ0 control register 0 (TQ0CTL0)



Note The setting is invalid when the TQ0CTL1.TQ0EEE bit = 1

(b) TMQ0 control register 1 (TQ0CTL1)

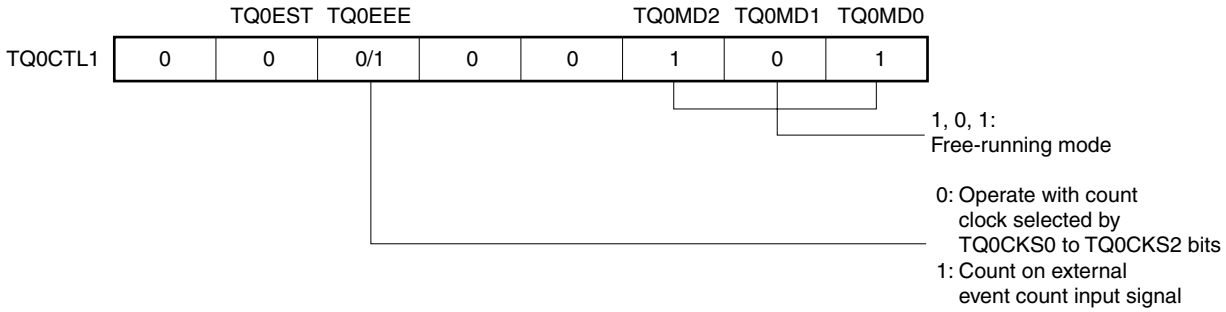
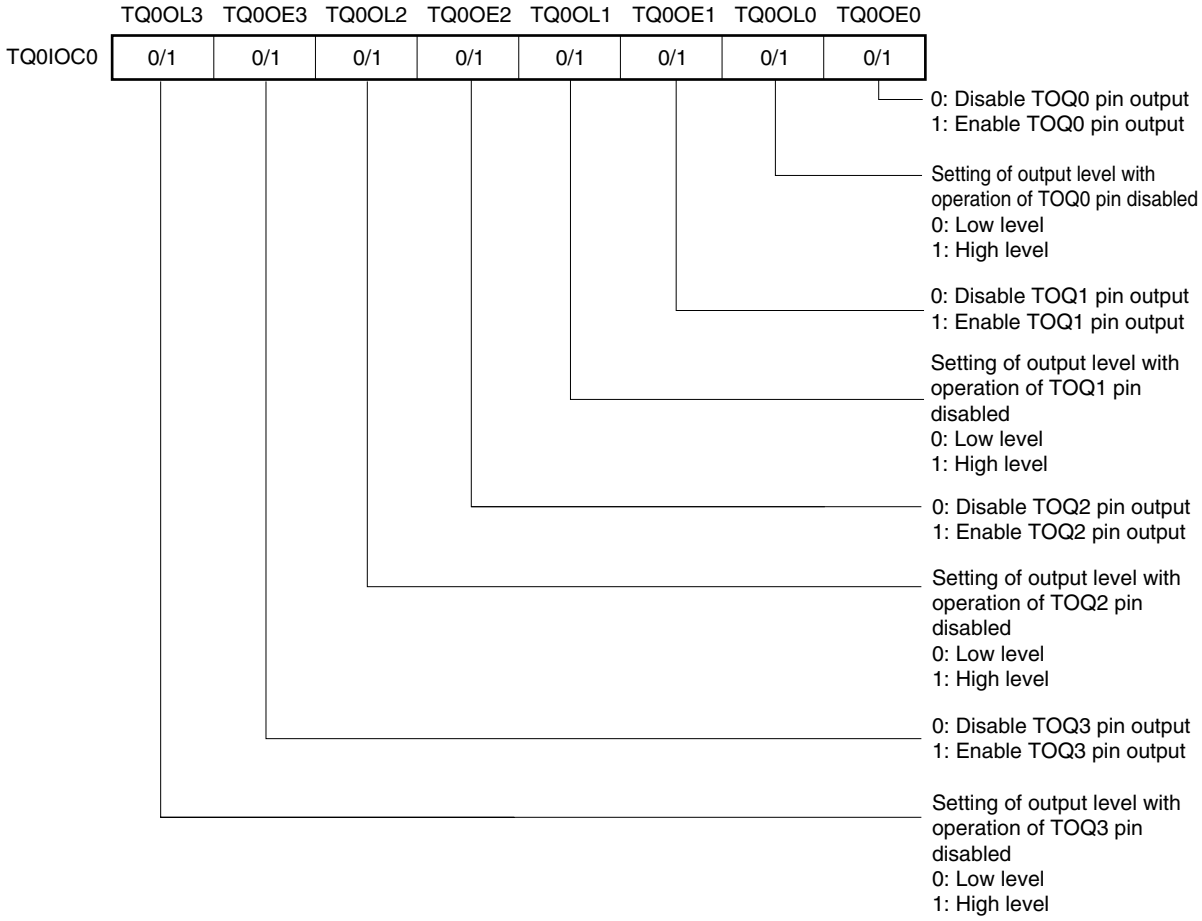


Figure 9-35. Register Setting in Free-Running Timer Mode (2/3)

(c) TMQ0 I/O control register 0 (TQ0IOC0)



(d) TMQ0 I/O control register 1 (TQ0IOC1)

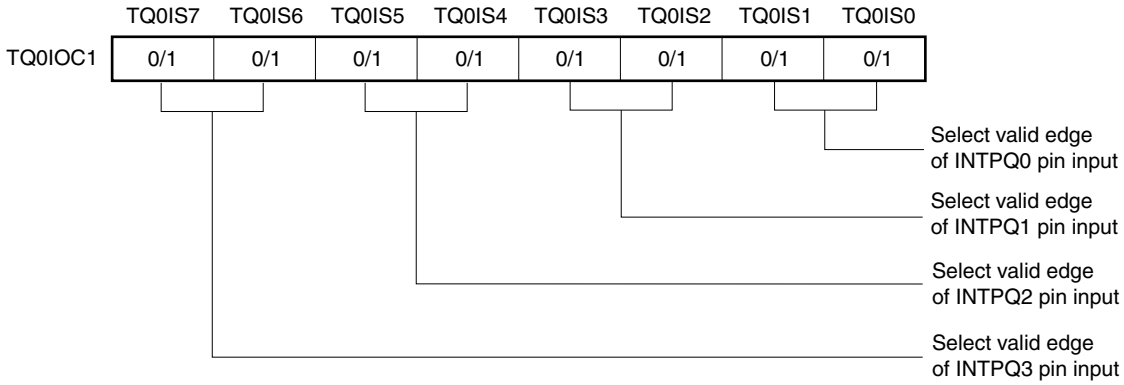
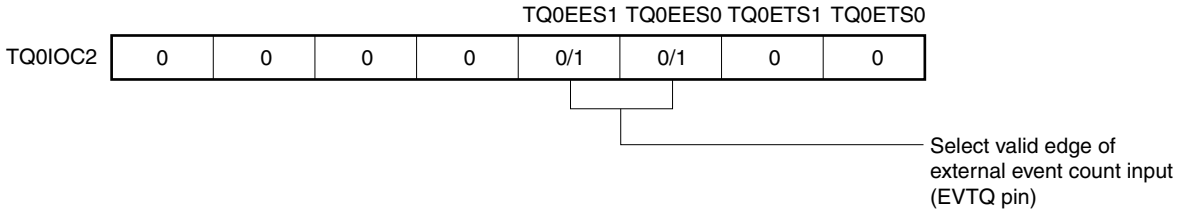
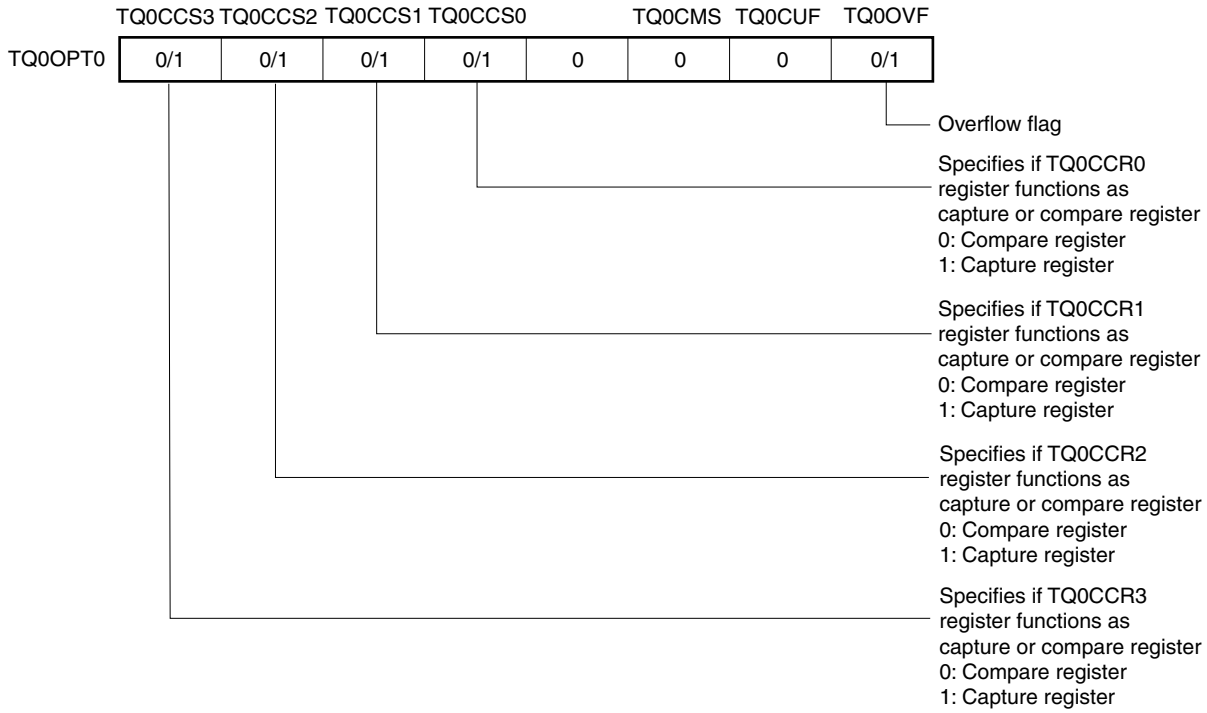


Figure 9-35. Register Setting in Free-Running Timer Mode (3/3)

(e) TMQ0 I/O control register 2 (TQ0IOC2)



(f) TMQ0 option register 0 (TQ0OPT0)



(g) TMQ0 counter read buffer register (TQ0CNT)

The value of the 16-bit counter can be read by reading the TQ0CNT register.

(h) TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3)

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 INTPQa pin is detected.

When the registers function as compare registers and when D_a is set to the TQ0CCRa register, the INTCCQa signal is generated when the counter reaches (D_a + 1), and the output signal of the TOQa pin is inverted.

Remark a = 0 to 3

(1) Operation flow in free-running timer mode

(a) When using capture/compare register as compare register

Figure 9-36. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)

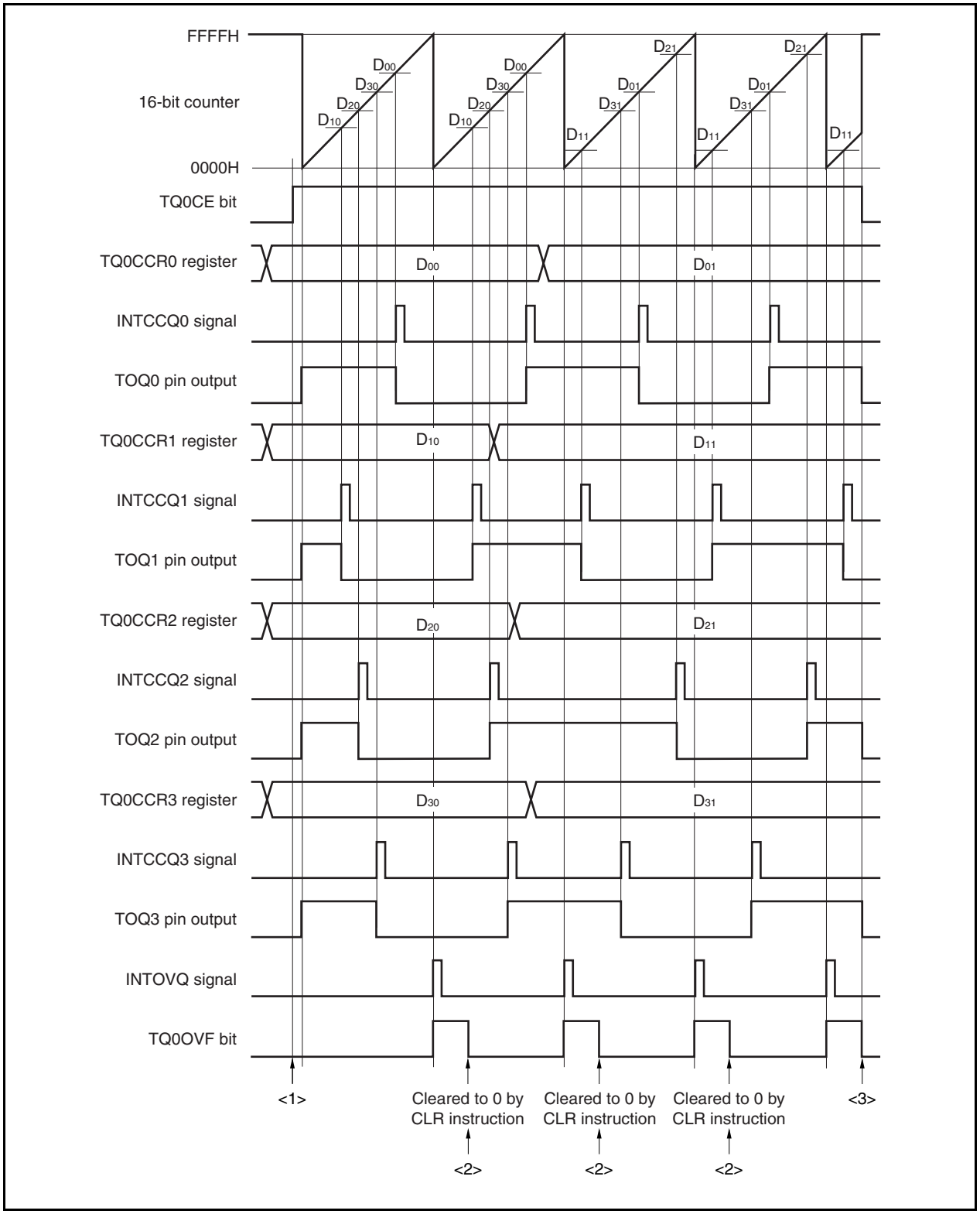
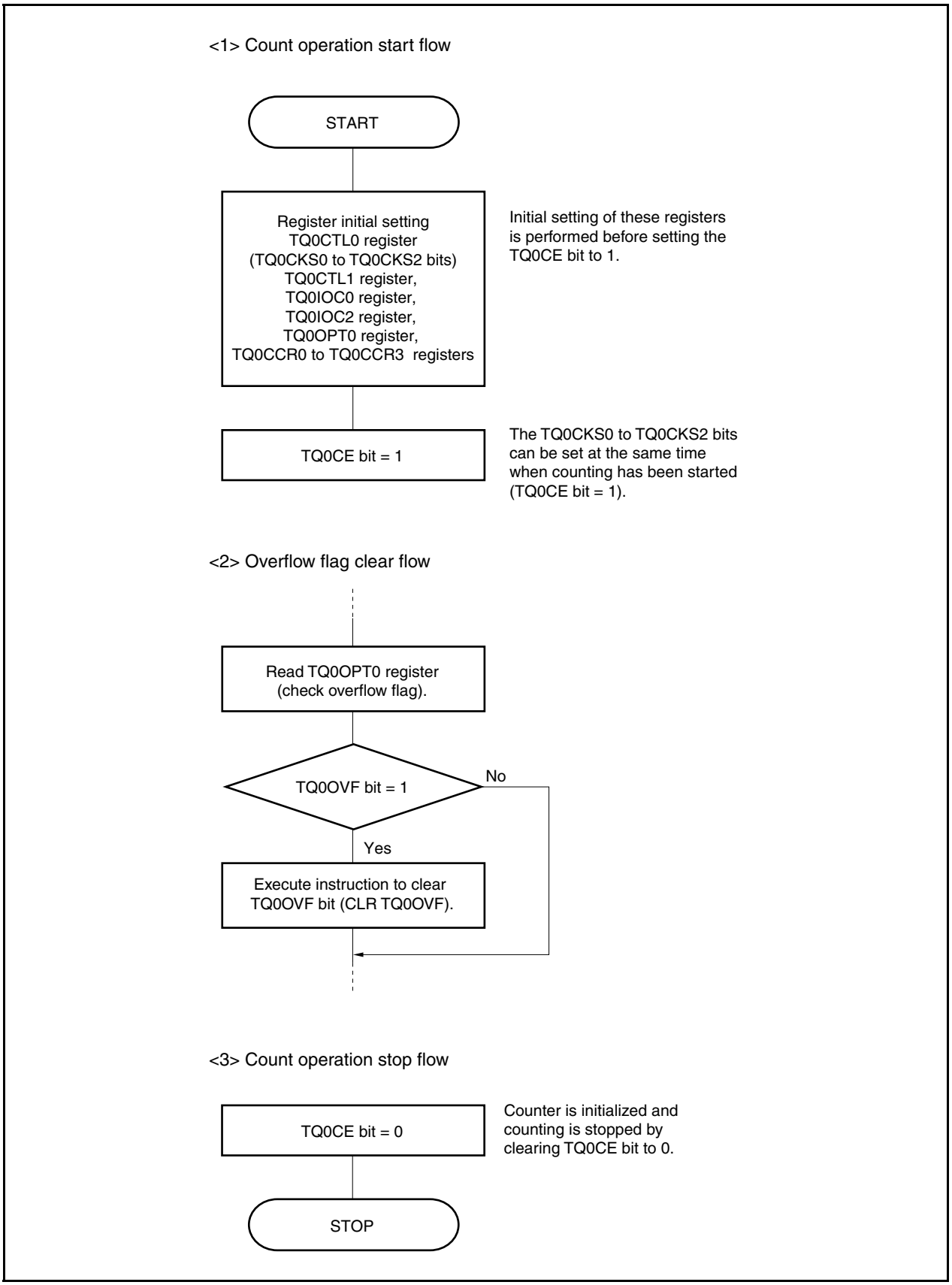


Figure 9-36. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)



(b) When using capture/compare register as capture register

Figure 9-37. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)

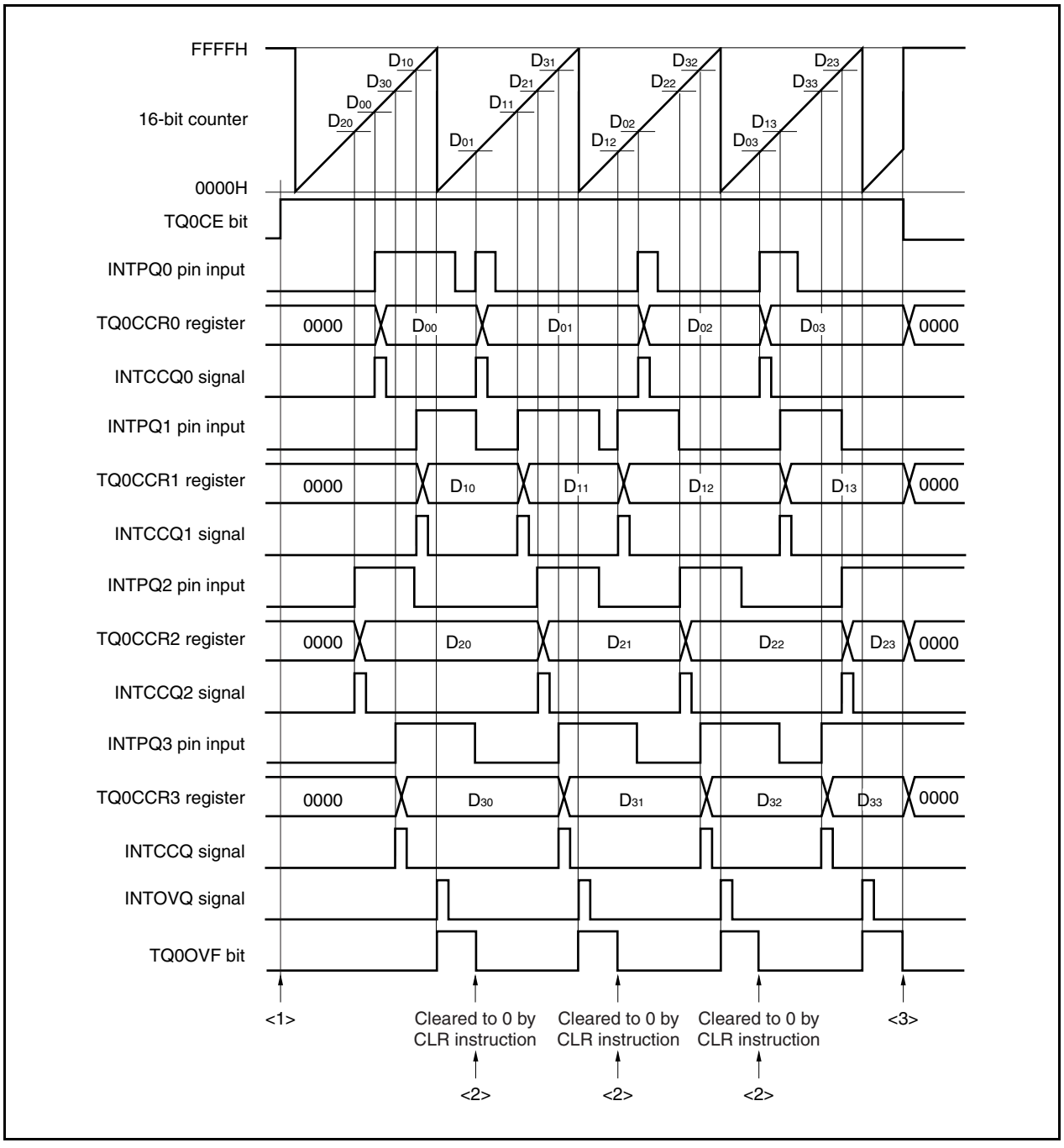
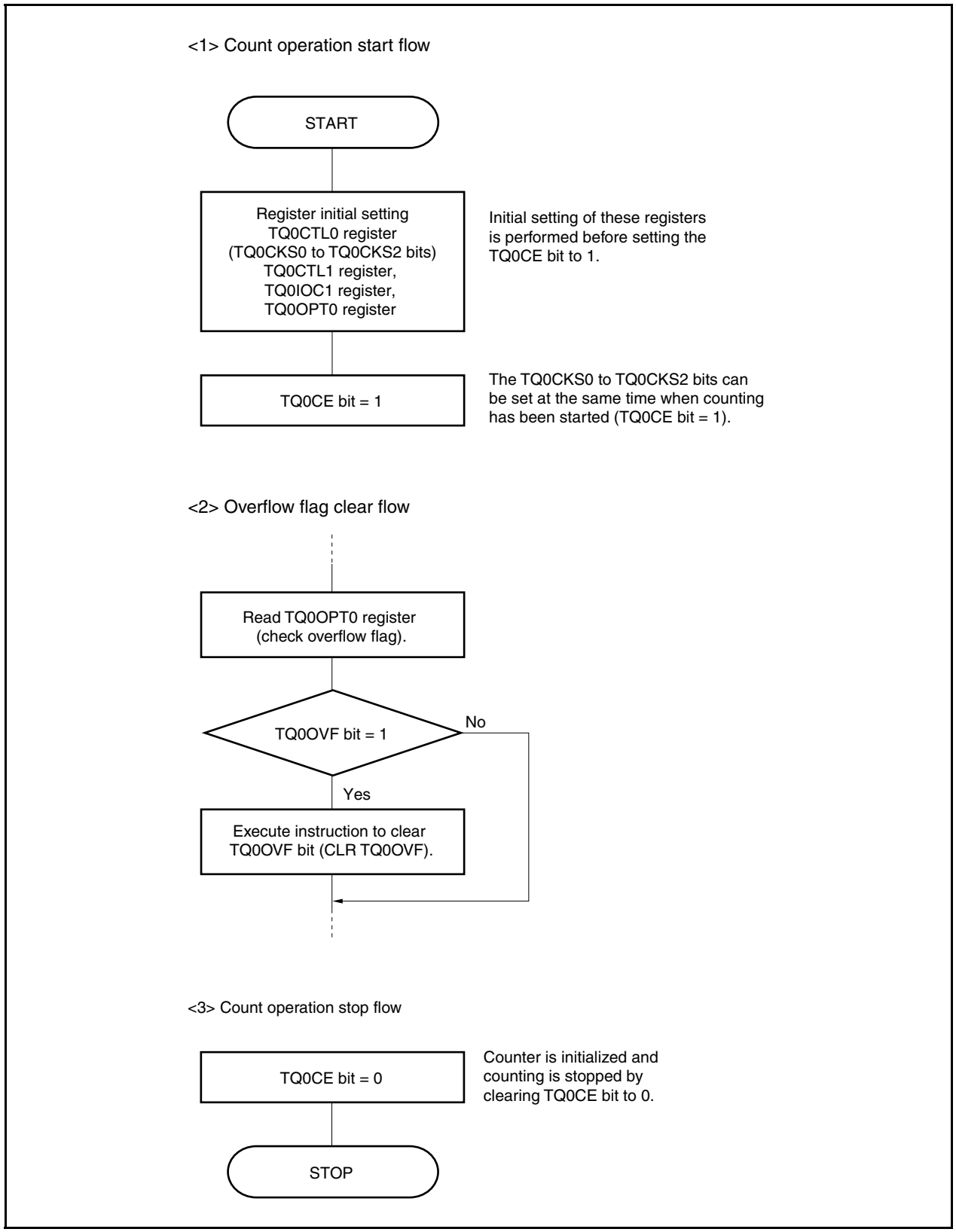


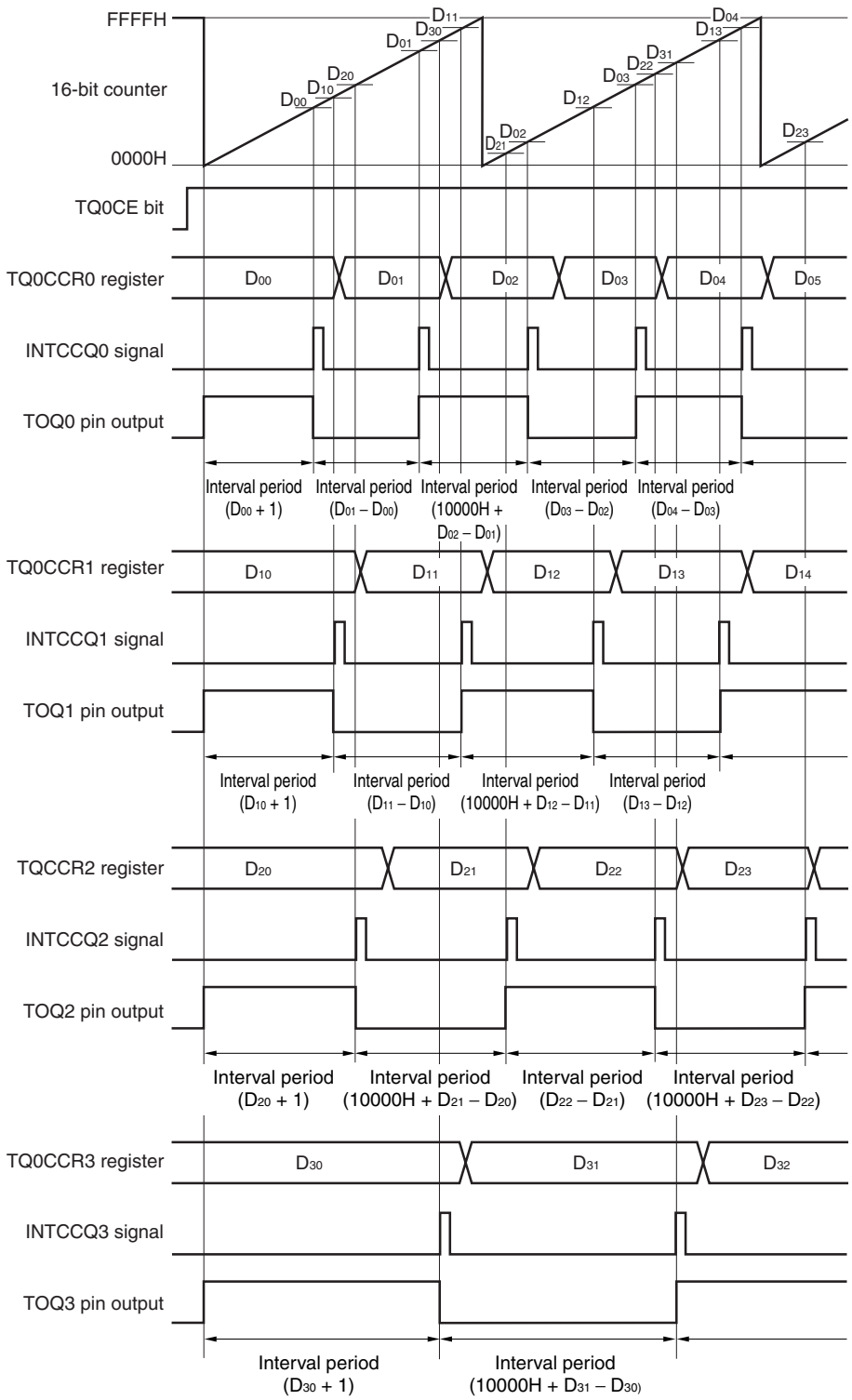
Figure 9-37. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)



(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 TQ0CCRa 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 INTCCQa 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 TQ0CCRa register must be re-set in the interrupt servicing that is executed when the INTCCQa signal is detected.

The set value for re-setting the TQ0CCRa register can be calculated by the following expression, where "D_a" is the interval period.

Compare register default value: $D_a - 1$

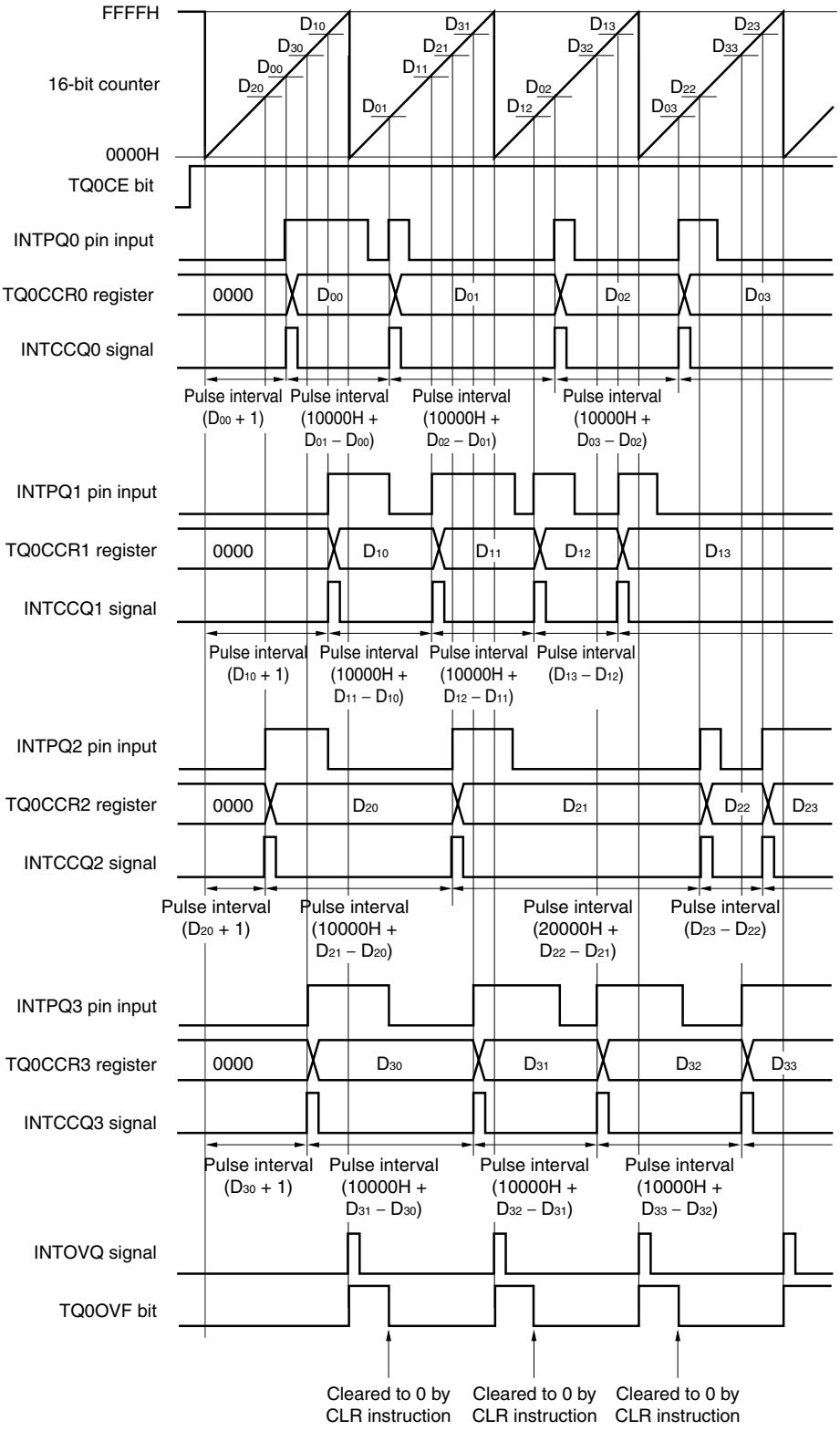
Value set to compare register second and subsequent time: Previous set value + D_a

(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

Remark 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 INTCCQa 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.

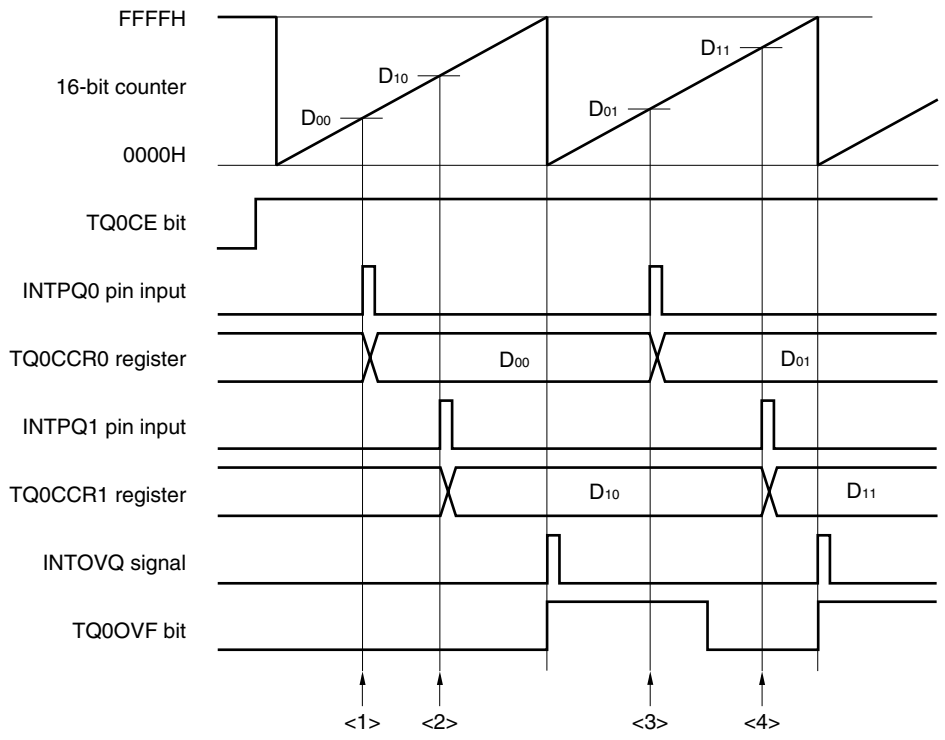
To measure a pulse width, the pulse width can be calculated by reading the value of the TQ0CCRa register in synchronization with the INTCCQa signal, and calculating the difference between the read value and the previously read value.

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



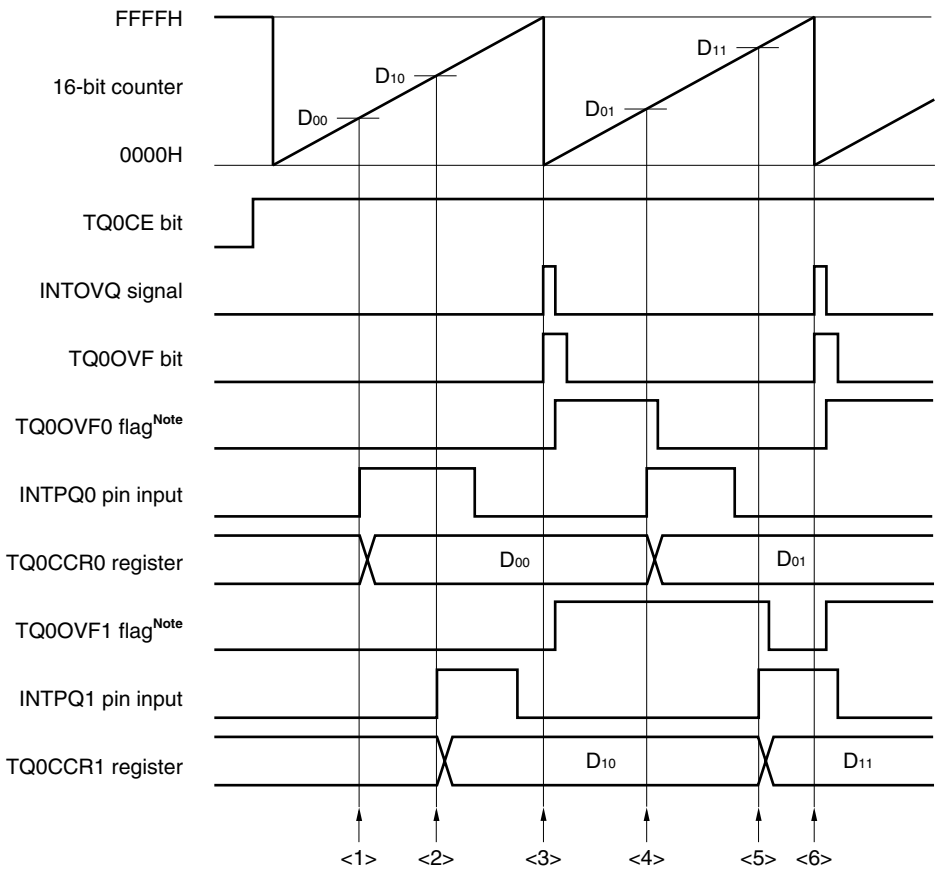
The following problem may occur when two pulse widths are measured in the free-running timer mode.

<1> Read the TQ0CCR0 register (setting of the default value of the INTPQ0 pin input).
<2> Read the TQ0CCR1 register (setting of the default value of the INTPQ1 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 INTPQ0 pin input).

<2> Read the TQ0CCR1 register (setting of the default value of the INTPQ1 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.

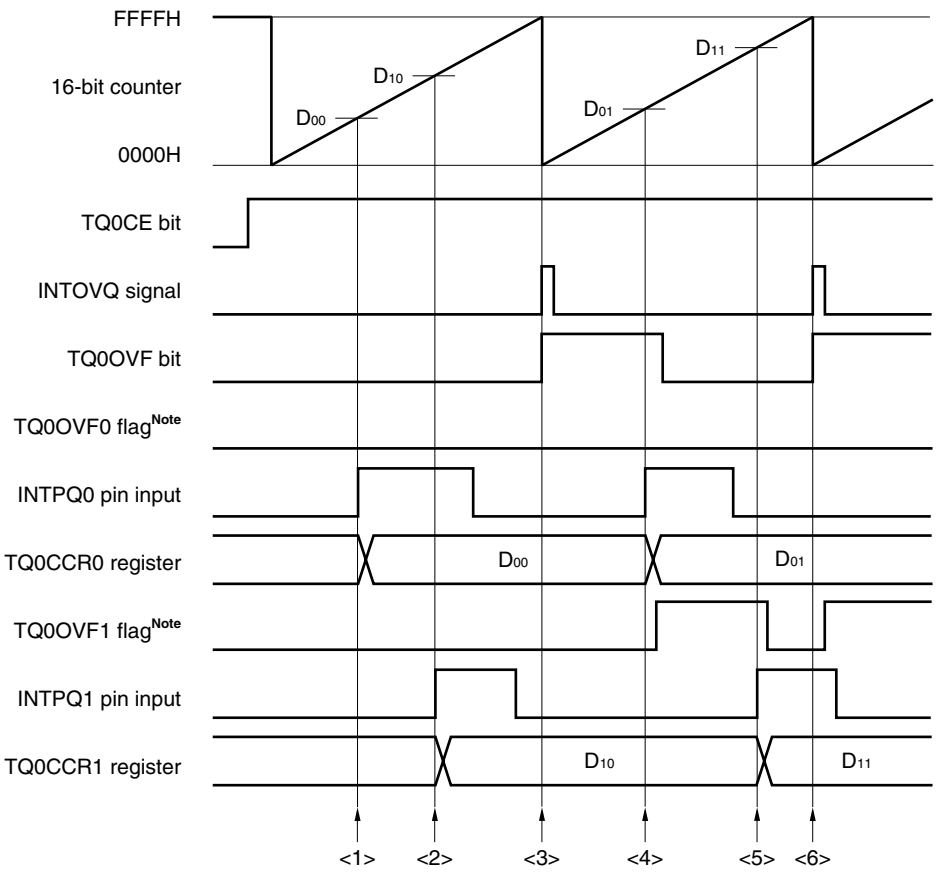
<4> Read the TQ0CCR0 register.
Read the TQ0OVF0 flag. If the TQ0OVF0 flag is 1, clear it to 0.
Because the TQ0OVF0 flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).

<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 INTPQ0 pin input).

<2> Read the TQ0CCR1 register (setting of the default value of the INTPQ1 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.

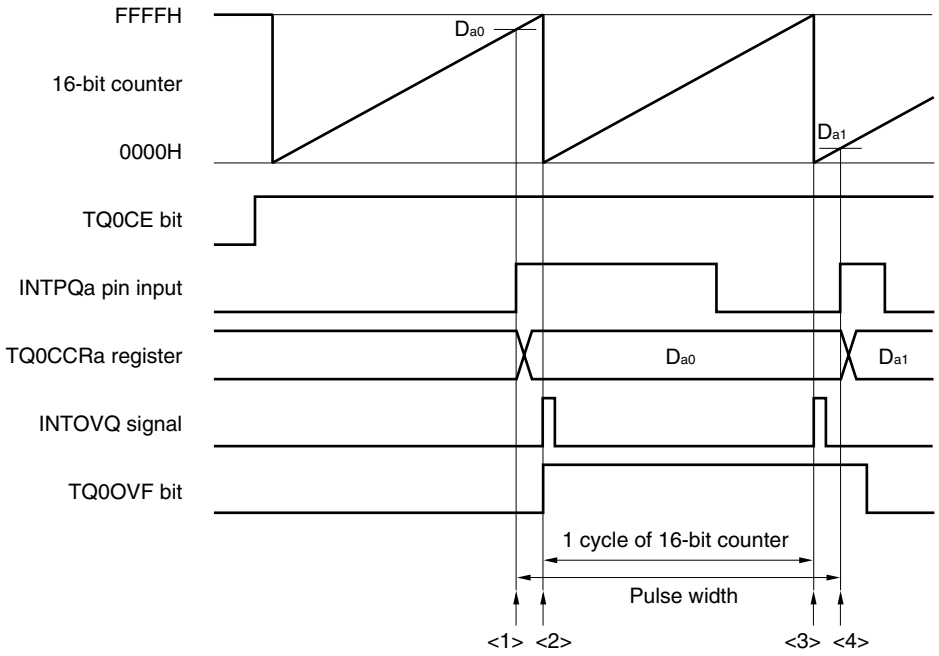
Because the TQ0OVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} − D_{10}) (correct).

<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



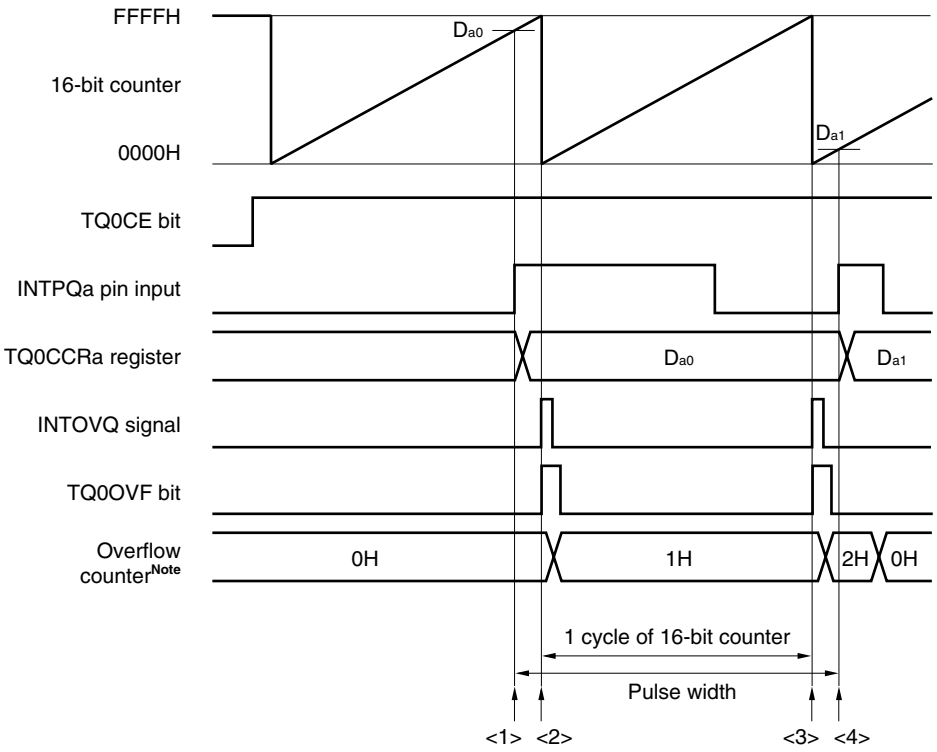
The following problem may occur when a long pulse width in the free-running timer mode.

<1> Read the TQ0CCRa register (setting of the default value of the INTPQa 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 next.

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 INTPQa 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 ($N \times 10000H + D_{a1} - D_{a0}$).

In this example, the pulse width is ($20000H + D_{a1} - D_{a0}$) because an overflow occurs twice.

Clear the overflow counter (0H).

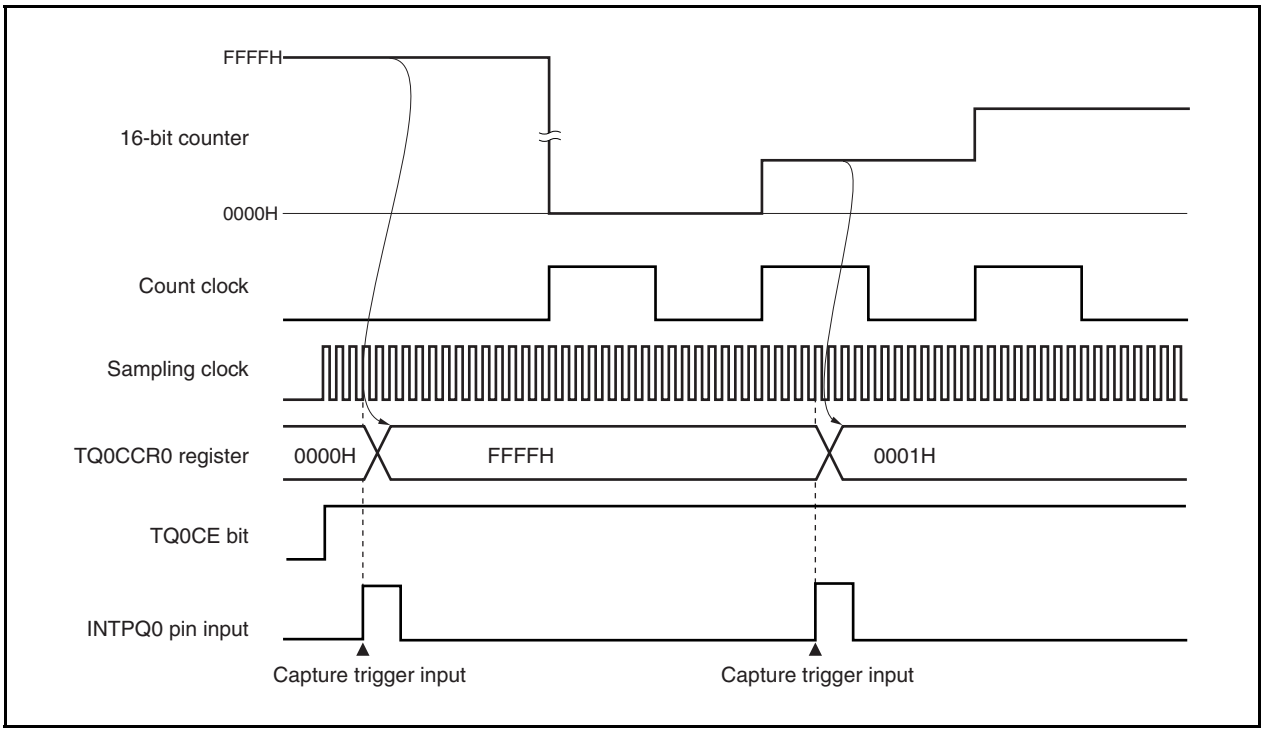
Remark a = 0 to 3

(e) 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 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).



9.6.7 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 INTPQa 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 (INTCCQa) occurs.

As shown in Figure 9-39, select either of the INTPQ0 to INTPQ3 pins as the capture trigger input pin. Specify “No edge detection” by using the TQ0IOC1 register for the unused pins.

Remark a = 0 to 3

Figure 9-38. Configuration in Pulse Width Measurement Mode

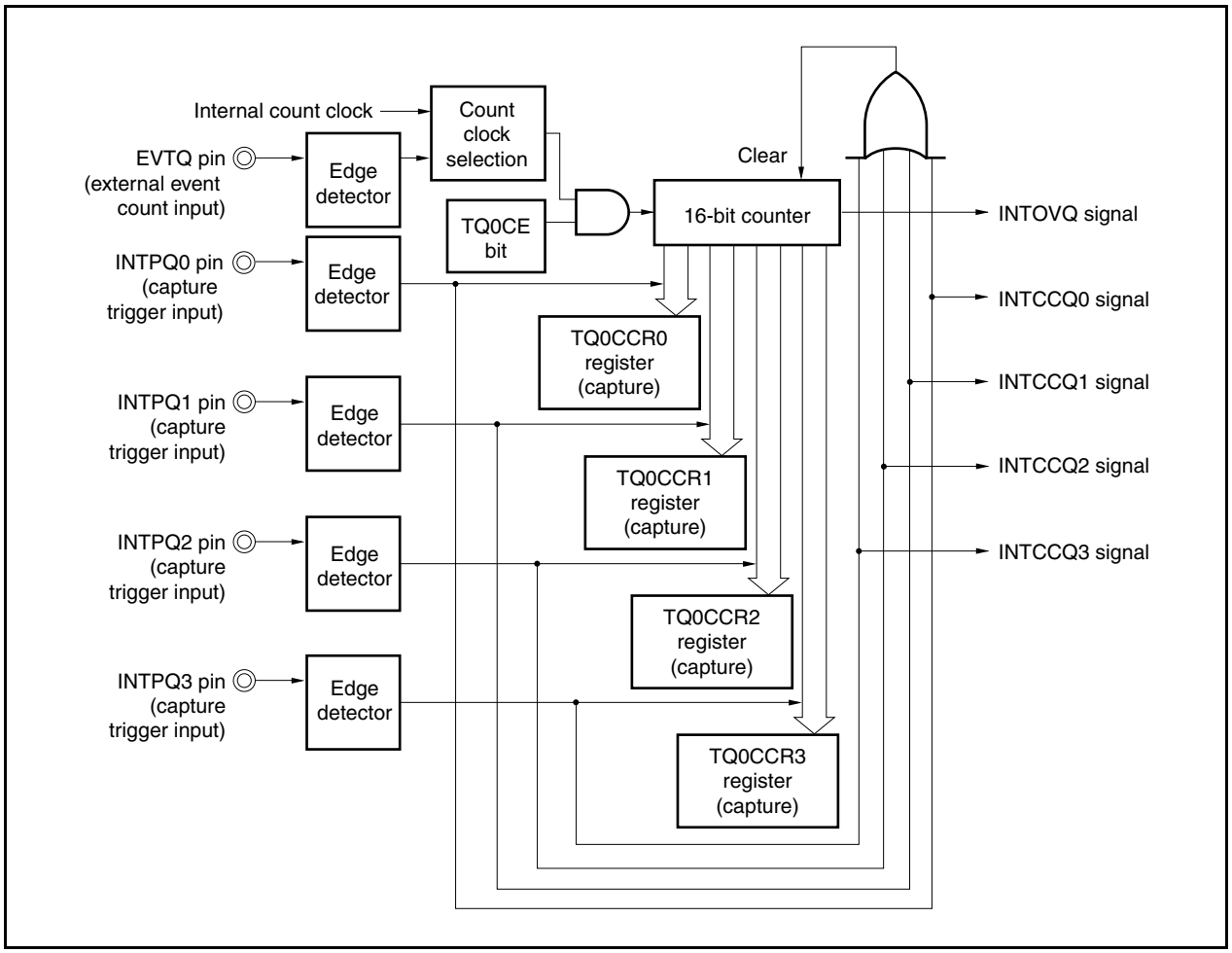
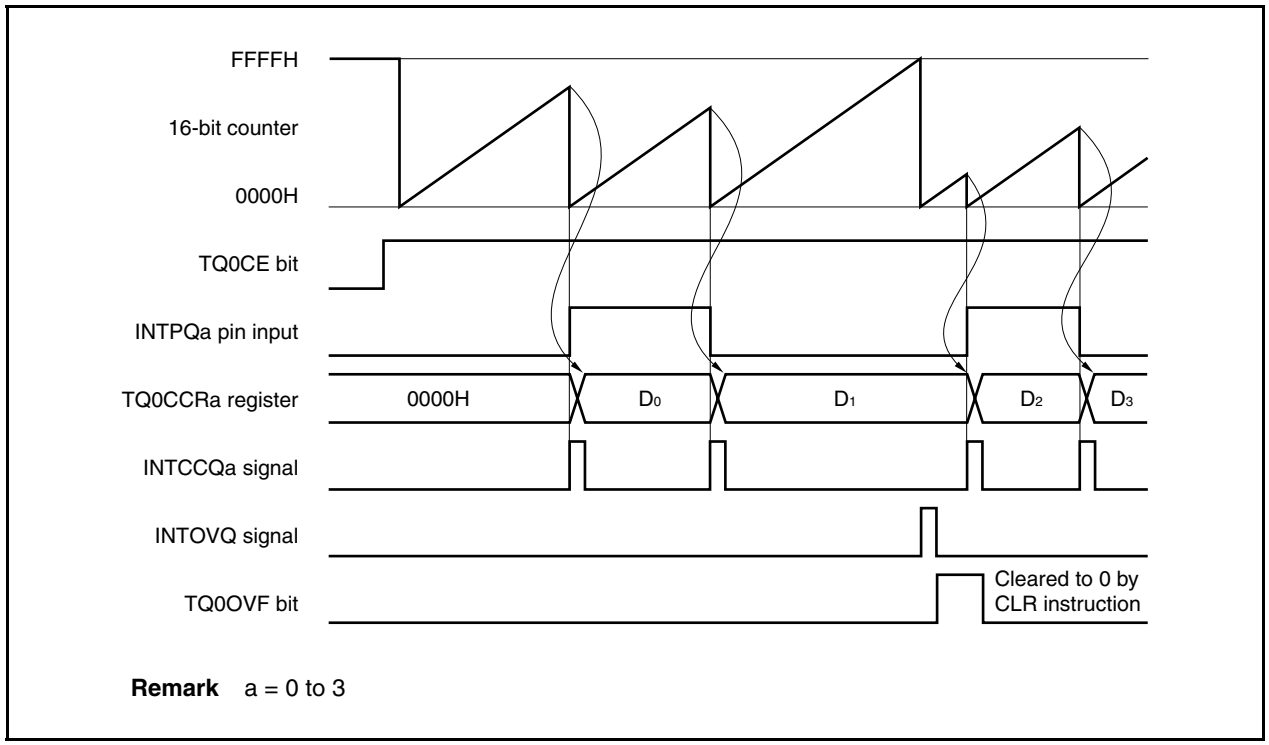


Figure 9-39. Basic Timing in Pulse Width Measurement Mode



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 INTPQa 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 (INTCCQa) is generated.

The pulse width is calculated as follows.

<R> Pulse width = $D_N \times$ 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 (INTOVQ) 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.

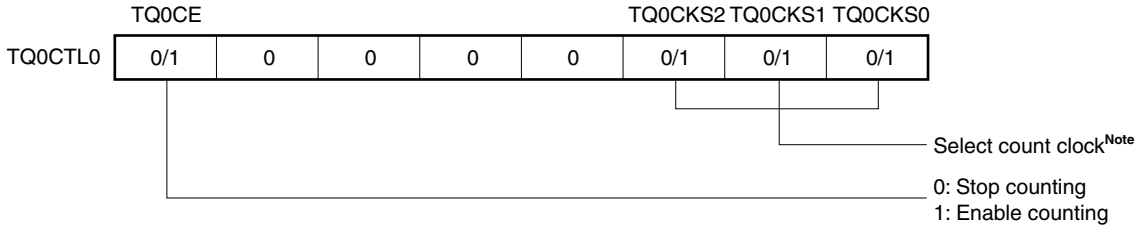
If the overflow flag is set to 1, the pulse width can be calculated as follows.

<R> Pulse width = (10000H $\times$ TQ0OVF bit set (1) count + D_N) $\times$ Count clock cycle

Remark a = 0 to 3

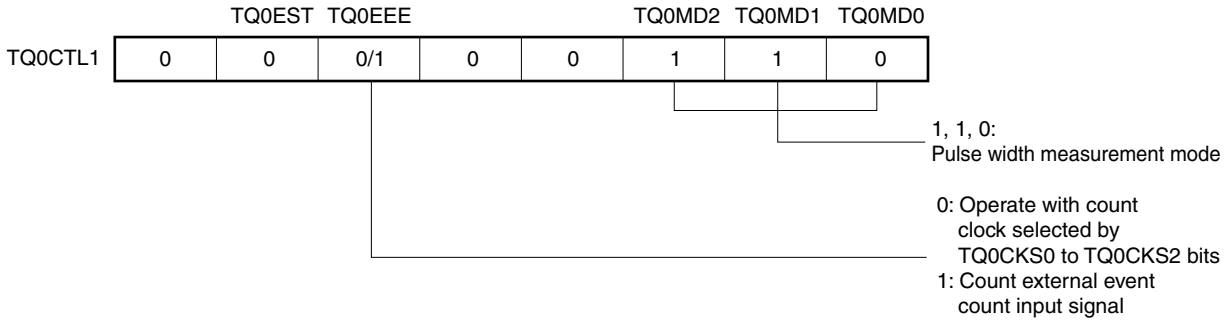
Figure 9-40. Register Setting in Pulse Width Measurement Mode (1/2)

(a) TMQ0 control register 0 (TQ0CTL0)

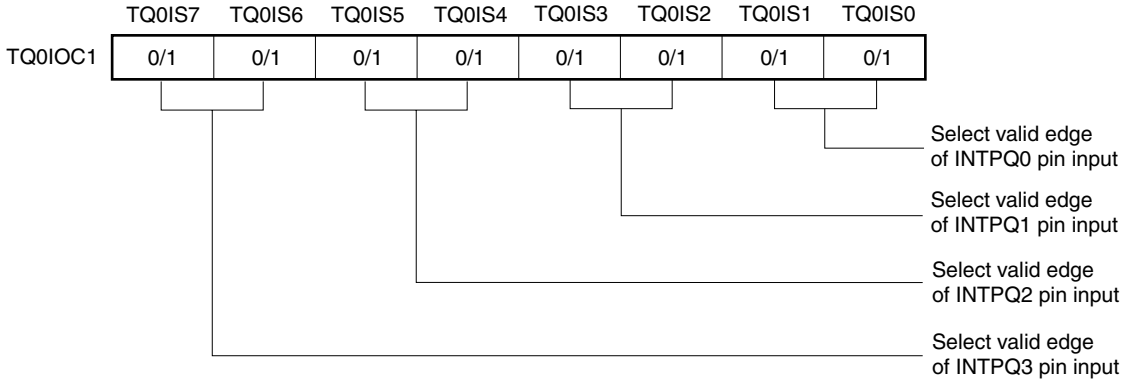


Note Setting is invalid when the TQ0CTL1.TQ0EEE bit = 1.

(b) TMQ0 control register 1 (TQ0CTL1)



(c) TMQ0 I/O control register 1 (TQ0IOC1)



(d) TMQ0 I/O control register 2 (TQ0IOC2)

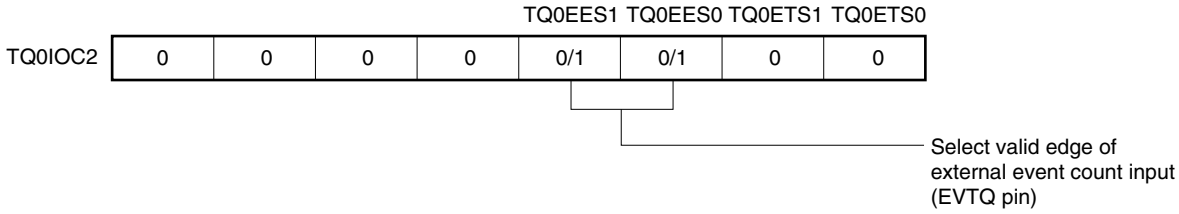


Figure 9-40. Register Setting in Pulse Width Measurement Mode (2/2)

(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)

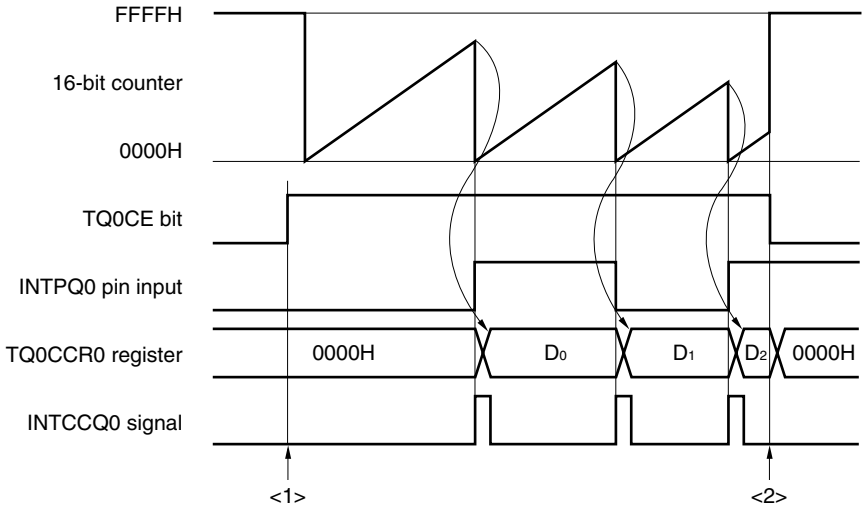
These registers store the count value of the 16-bit counter when the valid edge input to the INTPQa pin is detected.

Remarks 1. TMQ0 I/O control register 0 (TQ0IOC0) is not used in the pulse width measurement mode.

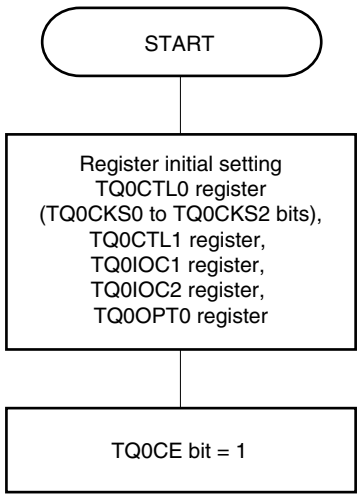
2. a = 0 to 3

(1) Operation flow in pulse width measurement mode

Figure 9-41. Software Processing Flow in Pulse Width Measurement Mode



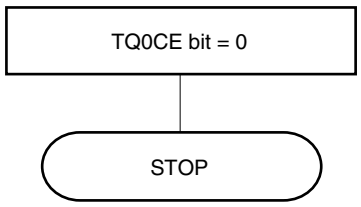
<1> Count operation start flow



Initial setting of these registers is performed before setting the TQ0CE bit to 1.

The TQ0CKS0 to TQ0CKS2 bits can be set at the same time when counting has been started (TQ0CE bit = 1).

<2> Count operation stop flow



The counter is initialized and counting is stopped by clearing the TQ0CE bit to 0.

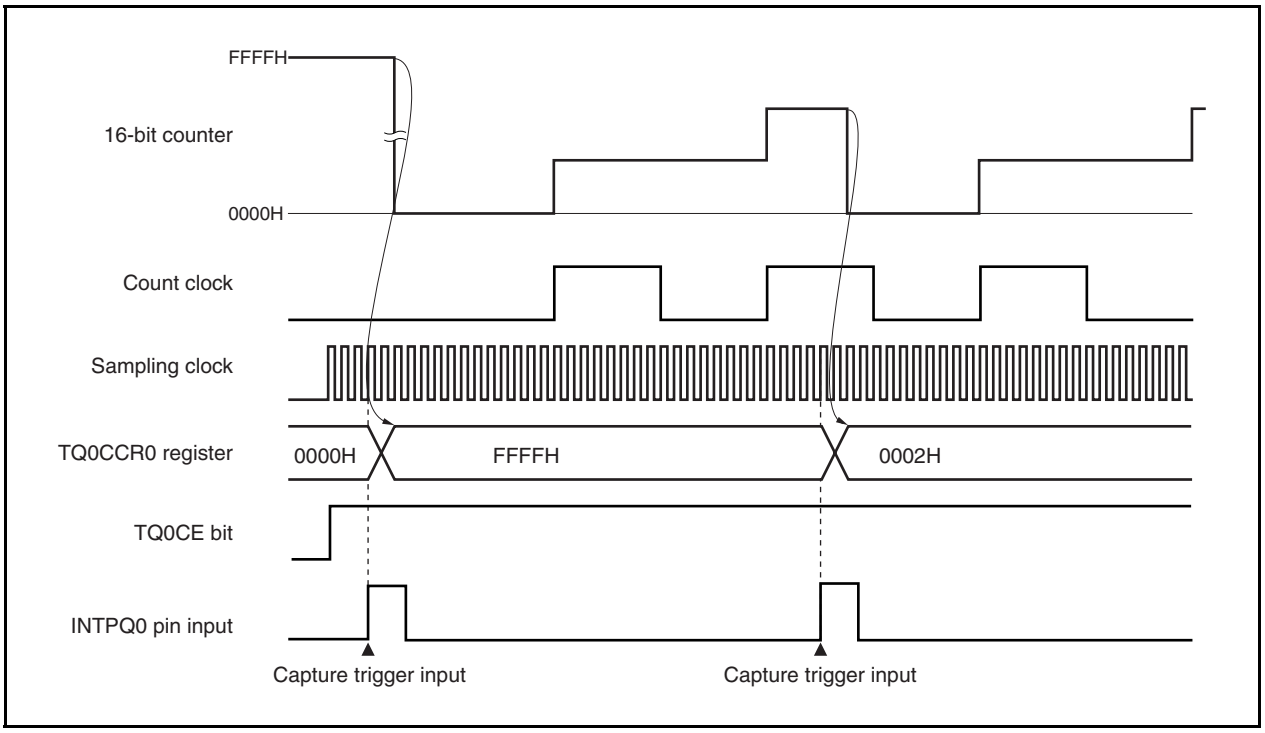
(2) Operation timing in pulse width measurement mode

(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 10 16-BIT INTERVAL TIMER D (TMD)

Timer D (TMD) is a 16-bit interval timer.
The V850E/MA3 incorporates TMD0 to TMD3.

10.1 Features

TMD functions as a 16-bit interval timer.

10.2 Function Overview

- 16-bit interval timer: 4 channels
- Compare registers: 4
- Interrupt request sources: 4
- Count clock selected from division of peripheral clock

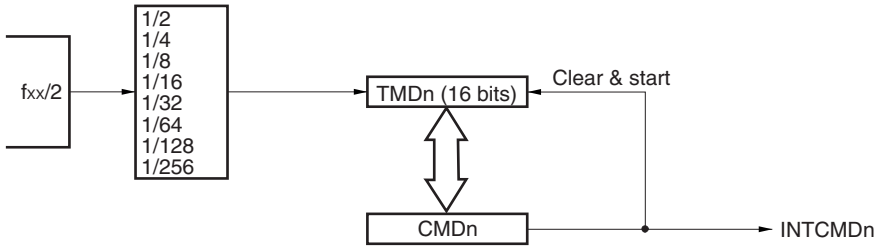
10.3 Configuration

Table 10-1. Timer D Configuration

Timer	Count Clock	Register	Read/Write	Generated Interrupt Signal	Capture Trigger	Timer Output S/R	Other Functions
Timer D	fxx/4, fxx/8, fxx/16, fxx/32, fxx/64, fxx/128, fxx/256, fxx/512	TMD0	Read	–	–	–	–
		CMD0	Read/write	INTCMD0	–	–	–
		TMD1	Read	–	–	–	–
		CMD1	Read/write	INTCMD1	–	–	–
		TMD2	Read	–	–	–	–
		CMD2	Read/write	INTCMD2	–	–	Output trigger in D/A real-time output mode
		TMD3	Read	–	–	–	–
		CMD3	Read/write	INTCMD3	–	–	Output trigger in D/A real-time output mode

Remark fxx: Peripheral clock
S/R: Set/reset

Figure 10-1. Timer D Block Diagram



Remarks 1. n = 0 to 3
2. fxx: Peripheral clock

10.3.1 Timers D0 to D3 (TMD0 to TMD3)

The TMDn register is a 16-bit timer. It is mainly used as an interval timer for software (n = 0 to 3).

Starting and stopping the TMDn register is controlled by the TMCDn.TMDCEn bit (n = 0 to 3).

Division by the prescaler can be selected for the count clock from among fxx/4, fxx/8, fxx/16, fxx/32, fxx/64, fxx/128, fxx/256, and fxx/512 by the TMCDn.CSn0 to TMCDn.CSn2 bits (fxx: Peripheral clock).

The TMDn register is read-only, in 16-bit units.

Reset input clears these registers to 0000H.

After reset: 0000H R Address: TMD0 FFFFF540H, TMD1 FFFFF550H, TMD2 FFFFF560H, TMD3 FFFFF570H

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

The conditions for which the TMDn register becomes 0000H are shown below (n = 0 to 3).

- Reset input
- TMCDn.TMDCAEn bit = 0
- TMCDn.TMDCEn bit = 0
- Match of TMDn register and CMDn register
- Overflow

Cautions 1. If the TMDCAEn bit is cleared (0), a reset is performed asynchronously.

2. If the TMDCEn bit is cleared (0), a reset is performed, in synchronization with the peripheral clock. Similarly, a synchronized reset is performed after a match with the CMDn register and after an overflow.

3. The count clock must not be changed during a timer operation. If it is to be overwritten, it should be overwritten after the TMDCEn bit is cleared (0).

4. Up to 4 peripheral clocks are required after a value is set in the TMDCEn bit until the set value is transferred to internal units. When a count operation begins, the count cycle from 0000H to 0001H differs from subsequent cycles.

5. After a compare match is generated, the timer is cleared at the next count clock. Therefore, if the division ratio is large, the timer value may not be zero even if the timer value is read immediately after a match interrupt is generated.

6. To initialize the status of the TMDn register and start counting again, set the TMDCEn bit to 1 after the lapse of 4 peripheral clocks.

10.3.2 Compare registers D0 to D3 (CMD0 to CMD3)

The CMDn register and the TMDn register count value are compared, and an interrupt request signal (INTCMDn) is generated when a match occurs. The TMDn register is cleared, in synchronization with this match. If the TMCDn.TMDCAEn bit is set to 0, a reset is performed asynchronously, and the registers are initialized (n = 0 to 3).

The CMDn registers are configured using a master/slave configuration. When a CMDn register is written, data is first written to the master register and then the master register data is transferred to the slave register. In a compare operation, the slave register value is compared with the count value of the TMDn register. When a CMDn register is read, data from the master side is read out.

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

Reset input clears these registers to 0000H.

Cautions 1. A write operation to a CMDn register requires 4 peripheral clocks until the value that was set in the CMDn register is transferred to internal units. When writing continuously to the CMDn register, be sure to reserve a time interval of at least 4 peripheral clocks.

2. The CMDn register can be overwritten only once in a single TMDn register cycle (from 0000H until an INTCMDn signal is generated due to a match of the TMDn register and CMDn register). If this cannot be secured by the application, make sure that the CMDn register is not overwritten during timer operation.

3. Note that the INTCMDn signal will be generated after an overflow if a value less than the counter value is written to the CMDn register during TMDn register operation (see Figure 10-2).

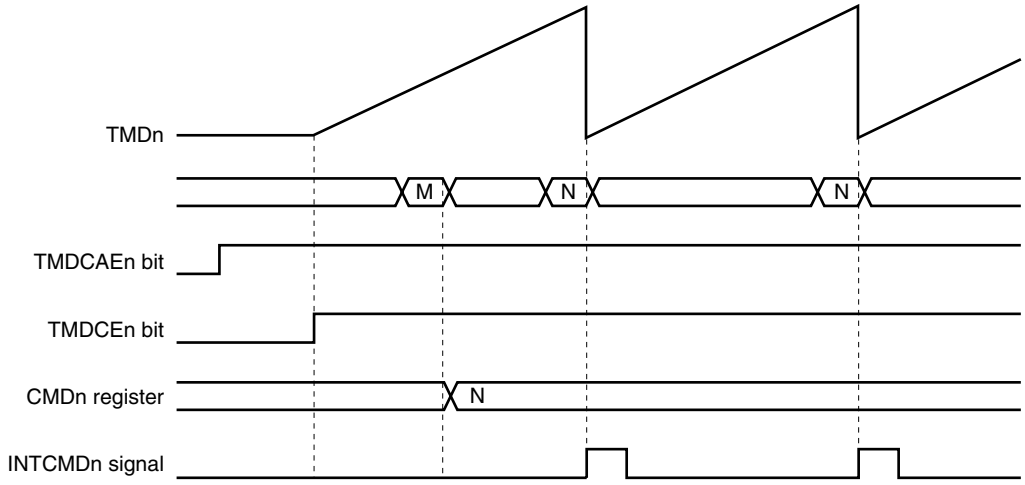
After reset: 0000H R/W Address: CMD0 FFFFF542H, CMD1 FFFFF552H, CMD2 FFFFF562H, CMD3 FFFFF572H

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

(n = 0 to 3)

Figure 10-2. Example of Timing During TMDn Operation

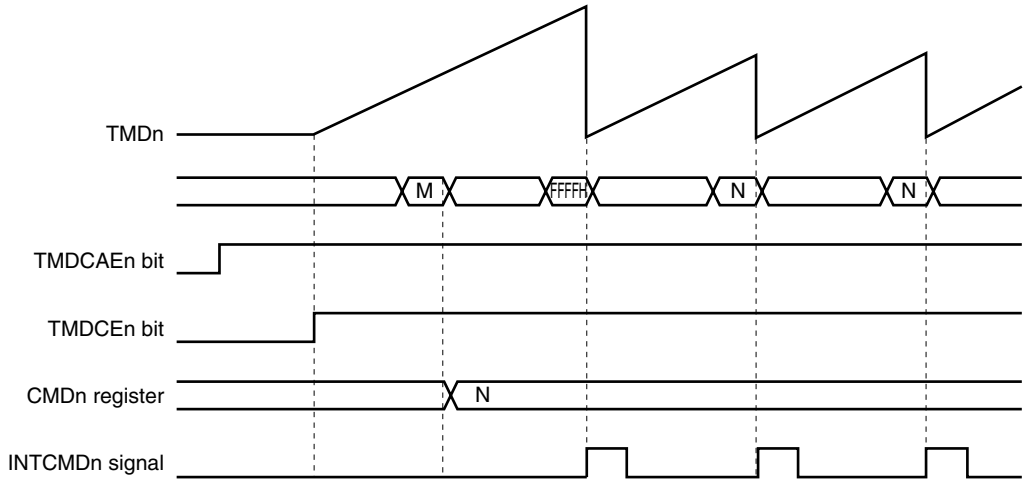
(a) When TMDn < CMDn



Remarks 1. M = TMDn value when overwritten
N = CMDn register value when overwritten
M < N

2. n = 0 to 3

(b) When TMDn > CMDn



Remarks 1. M = TMDn value when overwritten
N = CMDn register value when overwritten
M > N

2. n = 0 to 3

10.4 Control Registers

(1) Timer mode control registers D0 to D3 (TMCD0 to TMCD3)

The TMCDn registers control the operation of timer Dn (n = 0 to 3).

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

Reset input clears these registers to 00H.

Caution TMDCAEn and other bits cannot be set at the same time. The other bits and the registers of the other TMDn units should always be set after the TMDCAEn bit has been set.

After reset: 00H R/W Address: TMCD0 FFFFF544H, TMCD1 FFFFF554H, TMCD2 FFFFF564H, TMCD3 FFFFF574H

TMCDn (n = 0 to 3)

7	6	5	4	3	2	<1>	<0>
0	CSn2	CSn1	CSn0	0	0	TMDCEn	TMDCAEn

CSn2	CSn1	CSn0	Internal count clock selection
0	0	0	$f_{XX}/4$
0	0	1	$f_{XX}/8$
0	1	0	$f_{XX}/16$
0	1	1	$f_{XX}/32$
1	0	0	$f_{XX}/64$
1	0	1	$f_{XX}/128$
1	1	0	$f_{XX}/256$
1	1	1	$f_{XX}/512$

The CSn2 to CSn0 bits must not be changed during timer operation. If they are to be changed, they must be changed after clearing the TMDCEn bit to 0. If these bits are overwritten during timer operation, operation cannot be guaranteed.

TMDCEn	Timer Dn operation control
0	Count disabled (stops at 0000H and does not operate)
1	Counting operation is performed

The TMDCEn bit is not cleared even if a match is detected by the compare operation. To stop the count operation, clear the TMDCEn bit.

TMDCAEn	Count clock control
0	The entire TMDn unit is reset asynchronously. The supply of input clocks to the TMDn unit stops.
1	Clocks are supplied to the TMDn unit

- When the TMDCAEn bit is set to 0, the TMDn unit can be asynchronously reset.
- When the TMDCAEn bit is set to 0, the TMDn unit is in a reset state. Therefore, to operate TMDn, the TMDCAEn bit must be set to 1.
- If the TMDCAEn bit is cleared to 0, all the registers of the TMDn unit are initialized. If TMDCAEn bit is set to 1 again, be sure to set all the registers of the TMDn unit again.

Remark f_{XX}: Peripheral clock

10.5 Operation

10.5.1 Compare operation

Timer Dn can be used for a compare operation in which the value that was set in a compare register (CMDn) is compared with the count value of the TMDn register.

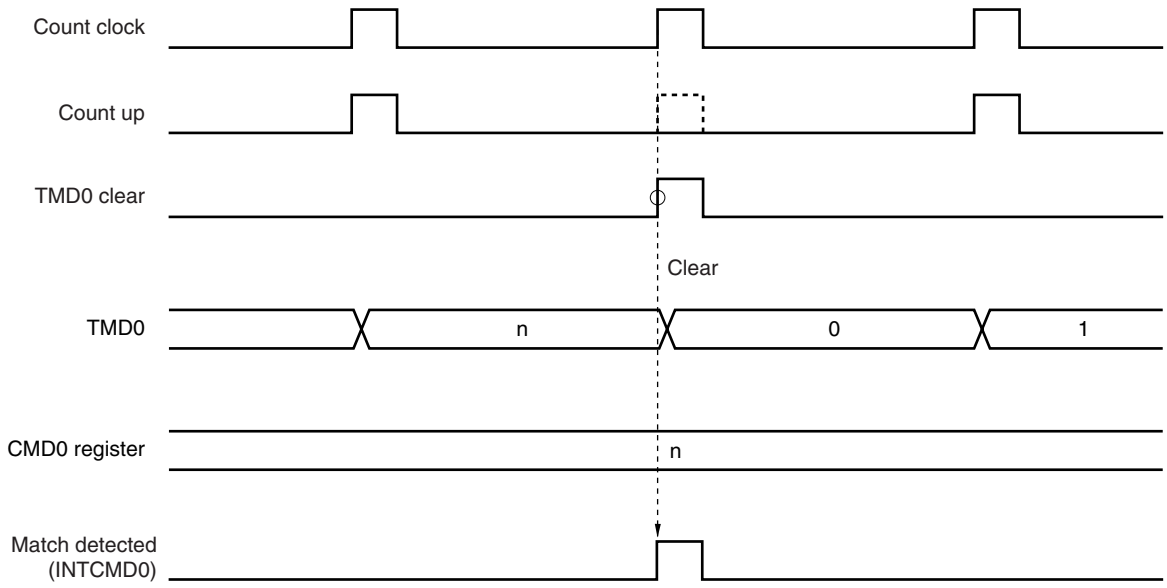
If a match is detected by the compare operation, an interrupt request signal (INTCMDn) is generated. The generation of the interrupt request signal causes the TMDn register to be cleared (0) at the next count timing. This function enables timer D to be used as an interval timer.

The CMDn register can also be set to 0. In this case, when an overflow occurs and the TMDn register becomes 0, a match is detected and the INTCMDn signal is generated. Although the value of the TMDn register is cleared (0) at the next count timing, the INTCMDn signal is not generated by this match.

Remark n = 0 to 3

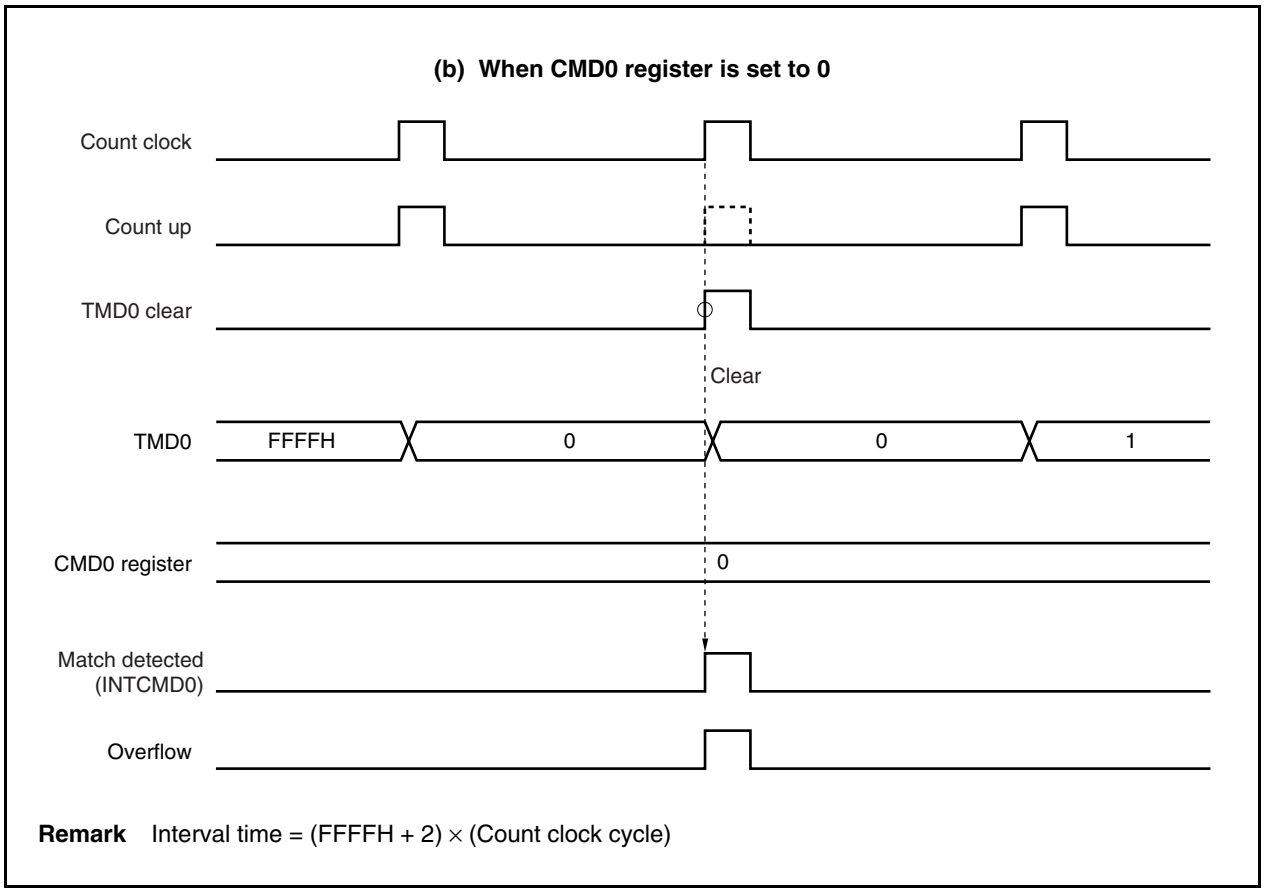
Figure 10-3. TMD0 Compare Operation Example (1/2)

(a) When CMD0 register is set to n (non-zero)



Remark Interval time = (n + 1) × (Count clock cycle)
n = 1 to 65,536 (FFFFH)

Figure 10-3. TMD0 Compare Operation Example (2/2)



Remark Interval time = (FFFFH + 2) × (Count clock cycle)

10.6 Application Examples

(1) Interval timer

This section explains an example in which timer D is used as an interval timer with 16-bit precision.

Interrupt request signals (INTCMDn) are output at equal intervals (see **Figure 10-3 TMD0 Compare Operation Example**). The setup procedure is shown below (n = 0 to 3).

<1> Set (1) the TMCDn.TMDCAEn bit.
<2> Set each register.
- Select the count clock using the TMCDn.CSn0 to TMCDn.CSn2 bits.
- Set the compare value in the CMDn register.

<3> Start counting by setting (1) the TMCDn.TMDCEn bit.
<4> If the TMDn register and CMDn register values match, an INTCMDn signal is generated.
<5> INTCMDn signals are generated thereafter at the same interval.

Remark n = 0 to 3

10.7 Cautions

Various cautions concerning timer D are shown below.

(1) To operate TMDn, first set (1) the TMCDn.TMDCAEn bit.

(2) Up to 4 peripheral clocks are required after a value is set to the TMCDn.TMDCEn bit until the set value is transferred to internal units. When a count operation begins, the count cycle from 0000H to 0001H differs from subsequent cycles.

(3) To initialize the TMDn register status and start counting again, clear (0) the TMDCEn bit and then set (1) the TMDCEn bit after an interval of 4 peripheral clocks has elapsed.

(4) Up to 4 peripheral clocks are required until the value that was set to the CMDn register is transferred to internal units. When writing continuously to the CMDn register, be sure to secure a time interval of at least 4 peripheral clocks.

(5) The CMDn register can be overwritten only once during a timer/counter operation (from 0000H until an INTCMDn signal is generated due to a match of the TMDn register and CMDn register). If this cannot be secured, make sure that the CMDn register is not overwritten during a timer/counter operation.

(6) The count clock must not be changed during a timer operation. If it is to be overwritten, it should be overwritten after the TMDCEn bit is cleared (0). If the count clock is overwritten during a timer operation, operation cannot be guaranteed.

(7) The INTCMDn signal will be generated after an overflow if a value less than the counter value is written to the CMDn register during TMDn register operation.

Remark n = 0 to 3

CHAPTER 11 16-BIT 2-PHASE ENCODER INPUT UP/DOWN COUNTER/GENERAL-PURPOSE TIMER (TMENC1)

Timer ENC1 (TMENC1) is a 16-bit 2-phase encoder input up/down counter general-purpose timer.
The V850E/MA3 incorporates TMENC10.

11.1 Functions

Timer ENC1 (TMENC1) has the following functions.

- General-purpose timer mode (see **11.5.1 Operation in general-purpose timer mode**)
  Free-running timer
  Timer output
- Up/down counter mode (see **11.5.2 Operation in UDC mode**)
  UDC mode A (mode 1, mode 2, mode 3, mode 4)
  UDC mode B (mode 1, mode 2, mode 3, mode 4)

11.2 Features

- 16-bit 2-phase encoder input up/down counter general-purpose timer: 1 channel
- Compare registers: 2
- Capture/compare registers: 2
- Interrupt request sources
  - Capture/compare match interrupt request: 2
  - Compare match interrupt request: 2
- Capture request signal: 2 types
  - The TMENC10 value can be latched using the valid edge of the INTP10 and INTP11 pins corresponding to the capture/compare register as the capture trigger.
- Count clock selectable through division by prescaler (set the frequency of the count clock to 10 MHz or less)
- Timer output function
  In the general-purpose timer mode, 16-bit resolution timer can be output from the TO10 pin.
- Timer clear
  The following timer clear operations are performed according to the mode that is used.
  (a) General-purpose timer mode: Timer clear operation is possible upon occurrence of match with CM100 register set value.
  (b) Up/down counter mode: The timer clear operation can be selected from among the following four conditions.
    (i) Timer clear performed upon occurrence of match with CM100 register set value during TMENC10 count-up operation, and timer clear performed upon occurrence of match with CM101 register set value during TMENC10 count-down operation.
    (ii) Timer clear performed only by external input.
    (iii) Timer clear performed upon occurrence of match between TMENC10 count value and CM100 register set value.
    (iv) Timer clear performed upon occurrence of external input and match between TMENC10 count value and CM100 register set value.
- External pulse output (TO10): 1

11.3 Configuration

The basic configuration is shown below.

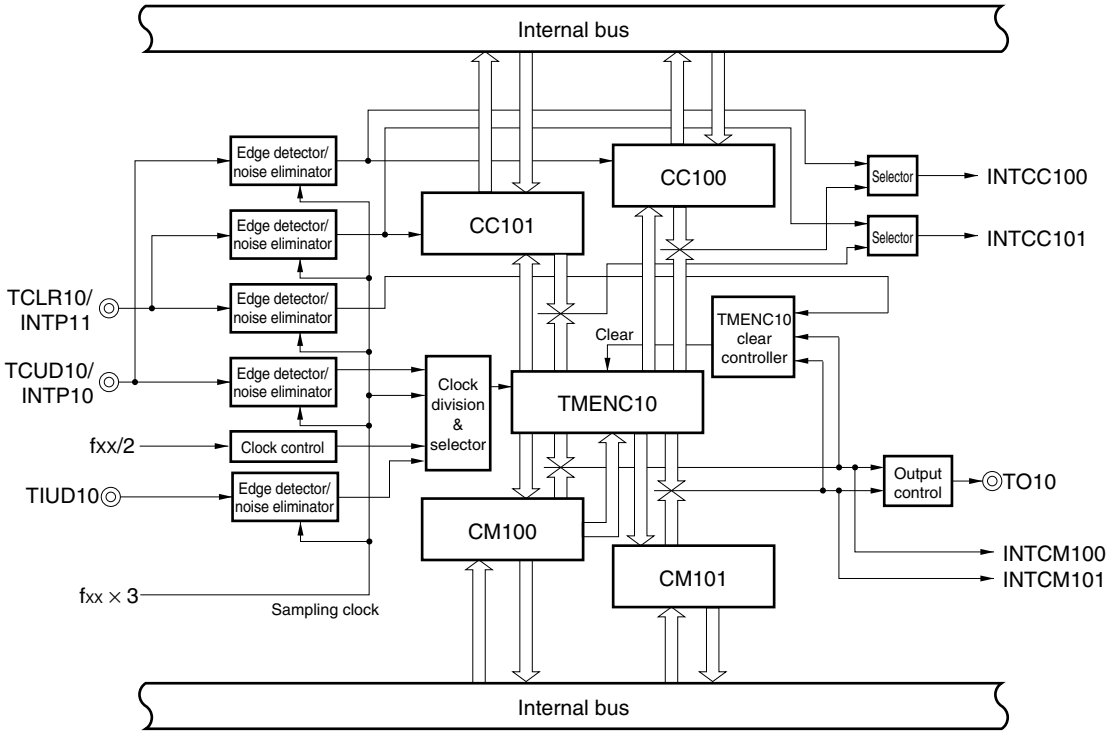
Table 11-1. Timer ENC1 Configuration

Timer	Count Clock	Register	Read/Write	Generated Interrupt Request Signal	Capture Trigger
Timer ENC1	fxx/4, fxx/8, fxx/16, fxx/32, fxx/64, fxx/128, fxx/256	TMENC10	Read/write	–	–
		CM100	Read/write	INTCM100	–
		CM101	Read/write	INTCM101	–
		CC100	Read/write	INTCC100	INTP10
		CC101	Read/write	INTCC101	INTP11

Remark fxx: Peripheral clock

Figure 11-1 shows the block diagram of timer ENC1.

Figure 11-1. Block Diagram of Timer ENC1



Remark fxx: Peripheral clock

(1) Timer ENC10 (TMENC10)

TMENC10 is a general-purpose timer (in general-purpose mode) and 2-phase encoder input up/down counter (in UDC mode).

This timer counts up in the general-purpose operation mode and counts up/down in the UDC mode.

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

Reset input clears this register to 0000H.

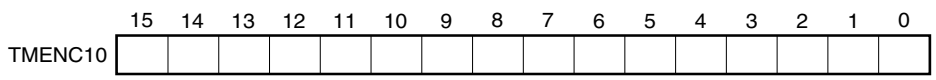
Cautions 1. Writing to TMENC10 is enabled only when the TMC10.CE101 bit is 0 (count operation disabled).

2. Continuous reading of TMENC10 is prohibited. If TMENC10 is continuously read, the second read value may differ from the actual value. If TMENC10 must be read twice, be sure to read another register between the first and the second read operation.

3. Writing the same value to the TMENC10, CC100, and CC101 registers, and the STATUS10 register is prohibited.

Writing the same value to the CCR10, TUM10, TMC10, SESA10, and PRM10 registers, and CM100 and CM101 registers is permitted (writing the same value is guaranteed even during a count operation).

After reset: 0000H R/W Address: FFFFF5C0H



TMENC10 start and stop is controlled by the TMC10.CE101 bit.

The TMENC10 operation consists of the following two modes.

(a) General-purpose timer mode

In the general-purpose timer mode, TMENC10 operates as a 16-bit interval timer, free-running timer, or timer output.

Counting is performed based on the clock selected by software. Division by the prescaler can be selected for the count clock from among fxx/4, fxx/8, fxx/16, fxx/32, fxx/64, fxx/128, or fxx/256, using the PRM10.PRM102 to PRM10.PRM100 bits (fxx: Peripheral clock).

(b) Up/down counter mode (UDC mode)

In the UDC mode, TMENC10 functions as a 16-bit up/down counter that performs counting based on the TCUD10 and TIUD10 input signals. This mode is divided into the UDC A mode and UDC B mode, depending on the condition of clearing TMENC10.

Cautions 1. TCUD10 and INTP10 are alternate-function pins. Therefore, when the TCUD10 pin is used in the UDC mode, the external capture function of the INTP10 pin cannot be used.

2. TCLR10 and INTP11 are alternate-function pins. Therefore, when the TCLR10 input is used in UDC mode A, the external capture function of the INTP11 pin cannot be used.

The conditions for clearing TMENC10 are as follows, according to the operation mode.

Table 11-2. Clear Conditions of Timer ENC1 (TMENC10)

Operation Mode	TUM10 Register		TMC10 Register			TMENC10 Clear
	T1CMD0 Bit	MSEL0 Bit	ENMD10 Bit	CLR101 Bit	CLR100 Bit	
General-purpose timer mode	0	0	0	×	×	Clearing not performed (free-running timer)
			1	×	×	Cleared upon match with CM100 register set value
UDC mode A	1	0	×	0	0	Cleared only by TCLR10 input
			×	0	1	Cleared upon match with CM100 register set value during count-up operation
			×	1	0	Cleared by TCLR10 input or upon match with CM100 register set value during count-up operation
			×	1	1	Clearing not performed
UDC mode B	1	1	×	×	×	Cleared upon match with CM100 register set value during count-up operation or upon match with CM101 register set value during count-down operation
Other than above						Setting prohibited

Remark ×: Indicates that the set value of that bit is ignored.

11.4 Control Registers

(1) Timer unit mode register 10 (TUM10)

The TUM10 register is an 8-bit register used to specify the TMENC10 operation mode or to control the operation of the timer output pin.

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

Reset input clears this register to 00H.

Cautions 1. Changing the value of the TUM10 register during TMENC10 operation (TMC10.CE101 bit = 1) is prohibited.

2. When the T1CMD0 bit = 0 (general-purpose timer mode), setting MSEL0 = 1 (UDC mode B) is prohibited.

After reset: 00H R/W Address: FFFFF5CBH

	7	6	5	4	3	2	1	0
TUM10	T1CMD0	0	0	0	TOE100	ALVT100	0	MSEL0

T1CMD0	TMENC10 operation mode specification
0	General-purpose timer mode (count-up)
1	UDC mode (count-up/-down)

TOE100	Specification of timer output (TO10) enable
0	Timer output disabled
1	Timer output enabled
When T1CMD0 bit = 1 (UDC mode), timer output is not performed regardless of the setting of the TOE100 bit. At this time, timer output is the inverted phase level of the level set by the ALVT100 bit.	

ALVT100	Specification of timer output (TO10) active level
0	Active level is high level
1	Active level is low level
When T1CMD0 bit = 1 (UDC mode), timer output is not performed regardless of the setting of the TOE100 bit. At this time, timer output is the inverted phase level of the level set by the ALVT100 bit.	

MSEL0	Specification of operation in UDC mode (count-up/-down)
0	UDC mode A TMENC10 can be cleared by setting the TMC10.CLR101 and TMC10.CLR100 bits.
1	UDC mode B TMENC10 is cleared in the following cases. • Upon match with CM100 register during TMENC10 count-up operation • Upon match with CM101 register during TMENC10 count-down operation
When UDC mode B is set, the TMC10.ENMD10, TMC10.CLR101, and TMC10.CLR100 bits become invalid.	

(2) Timer control register 10 (TMC10)

The TMC10 register is used to enable/disable TMENC10 operation and to set transfer and timer clear operations.

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

Reset input clears this register to 00H.

Caution Changing the values of the TMC10 register bits other than the CE101 bit during TMENC10 operation (CE101 = 1) is prohibited.

After reset: 00H R/W Address: FFFFF5CCH

	7	<6>	5	4	3	2	1	0
TMC10	0	CE101	0	0	RLEN10	ENMD10	CLR101	CLR100

CE101	TMENC10 operation control
0	Count operation disabled
1	Count operation enabled

RLEN10	Specification of transfer operation from CM100 register to TMENC10
0	Transfer operation disabled
1	Transfer operation enabled

- When RLEN10 = 1, the value set to CM100 register is transferred to TMENC10 upon occurrence of a TMENC10 underflow.
- The RLEN10 bit is valid only in UDC mode A (TUM10.T1CMD0 bit = 1, MSEL0 bit = 0). In the general-purpose timer mode (T1CMD0 bit = 0) and in UDC mode B (T1CMD0 bit = 1, MSEL0 bit =1), a transfer operation is not performed even if the RLEN10 bit is set (1).

ENMD10	Control of TMENC10 clear operation in general-purpose timer mode
0	Clear disabled (free-running mode) Clearing is not performed even when TMENC10 and CM100 register values match.
1	Clear enabled Clearing is performed when TMENC10 and CM100 register values match.

When in the UDC mode (TUM10.T1CMD0 bit) = 1, the ENMD10 bit setting becomes invalid.

CLR101	CLR100	TMENC10 clear source specification
0	0	Cleared only by external input (TCLR10)
0	1	Cleared upon match of TMENC10 count value and CM100 register set value
1	0	Cleared by TCLR10 input or upon match of TMENC10 count value and CM100 register set value
1	1	Not cleared

- Clearing by match of the TMENC10 count value and CM100 register set value is valid only during a TMENC10 count-up operation (TMENC10 is not cleared during a TMENC10 count-down operation).
- When in the general-purpose timer mode (TUM10.T1CMD0 bit = 0), the CLR101 and CLR100 bit settings are invalid.
- When in the UDC mode B (TUM10.MSEL0 bit = 1), the CLR101 and CLR100 bit settings are invalid.
- When clearing by TCLR10 has been enabled by bits CLR101 and CLR100, clearing is performed regardless of whether the value of the CE101 bit is 1 or 0.

(3) Capture/compare control register 10 (CCR10)

The CCR10 register specifies the operation mode of the CC100 and CC101 registers.

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

Reset input clears this register to 00H.

Cautions 1. Overwriting the CCR10 register during TMENC10 operation (TMC10.CE101 bit = 1) is prohibited.

2. TCUD10 and INTP10 are alternate-function pins. Therefore, when the TCUD10 pin is used in the UDC mode, the external capture function of the INTP10 pin cannot be used.

3. TCLR10 and INTP11 are alternate-function pins. Therefore, when the TCLR10 input is used in UDC mode A, the external capture function of the INTP11 pin cannot be used.

After reset: 00H R/W Address: FFFFF5CAH

	7	6	5	4	3	2	1	0
CCR10	0	0	0	0	0	0	CMS01	CMS00

CMS01	CC101 register operation mode specification
0	Operates as capture register
1	Operates as compare register

CMS00	CC100 register operation mode specification
0	Operates as capture register
1	Operates as compare register

(4) Valid edge select register 10 (SESA10)

The SESA10 register is used to specify the valid edge of external interrupt request signals (INTP10, INTP11, TIUD10, TCUD10, TCLR10) from the external pins.

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

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

Reset input clears this register to 00H.

Cautions 1. Changing the values of the SESA10 register bits during TMENC10 operation (TMC10.CE101 = 1) is prohibited.

2. Before setting the trigger mode of the INTP10, INTP11, TIUD10, TCUD10, and TCLR10 pins, set the PMC0 and PMC2 registers. If the PMC0 and PMC2 registers are set after the SESA10 register has been set, an illegal interrupt, incorrect counting, and incorrect clearing may occur, depending on the timing of setting the PMC0 and PMC2 registers.

After reset: 00H R/W Address: FFFFF5CDH

SESA10

7	6	5	4	3	2	1	0
TESUD01	TESUD00	CESUD01	CESUD00	IES101	IES100	IES001	IES000
TIUD10, TCUD10		TCLR10		INTP11		INTP10	

TESUD01	TESUD00	Specification of valid edge of TIUD10 and TCUD10 pins
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

- The setting of the TESUD01 and TESUD00 bits is valid only in UDC mode A and UDC mode B.
- If mode 4 is specified as the operation mode of TMENC10 (specified by the PRM10.PRM102 to PRM10.PRM100 bits), the valid edge specifications for the TIUD10 and TCUD10 pins (TESUD01 and TESUD00 bits) are not valid.

CESUD01	CESUD00	Specification of valid edge of TCLR10 pin
0	0	Falling edge (TMENC10 cleared after edge detection)
0	1	Rising edge (TMENC10 cleared after edge detection)
1	0	Low level (TMENC10 cleared status held)
1	1	High level (TMENC10 cleared status held)

- The setting of the CESUD01 and CESUD00 bits is valid only in UDC mode A.

IES101	IES100	Specification of valid edge of INTP11 pin
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

IES001	IES000	Specification of valid edge of INTP10 pin
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

(5) Prescaler mode register 10 (PRM10)

The PRM10 register is used to perform the following selections.

- Selection of count clock in general-purpose timer mode (TUM10.T1CMD0 bit = 0)
- Selection of count operation mode in UDC mode (TUM10.T1CMD0 = 1)

PRM10 can be read or written in 8-bit units.

Reset input sets this register to 07H.

Cautions 1. Overwriting the PRM10 register during TMENC10 operation (TMC10.CE101 bit = 1) is prohibited.

2. In the UDC mode (the T1CMD0 bit of the TUM10 register = 1), setting the values of the PRM102 bit to 0 is prohibited.

3. When TMENC10 is in mode 4, specification of the valid edge for the TIUD10 and TCUD10 pins is invalid.

After reset: 07H R/W Address: FFFFF5CEH

	7	6	5	4	3	2	1	0
PRM10	0	0	0	0	0	PRM102	PRM101	PRM100

PRM102	PRM101	PRM100	T1CMD0 = 0	T1CMD0 = 1	
			Count clock	Count clock	UDC mode
0	0	0	Setting prohibited	Setting prohibited	
0	0	1	f_{XX}/4		
0	1	0	f_{XX}/8		
0	1	1	f_{XX}/16		
1	0	0	f_{XX}/32	TIUD10	Mode 1
1	0	1	f_{XX}/64		Mode 2
1	1	0	f_{XX}/128		Mode 3
1	1	1	f_{XX}/256		Mode 4

Remark f_{XX}: Peripheral clock

(a) In general-purpose timer mode (TUM10.T1CMD0 bit = 0)

The count clock is specified by the PRM102 to PRM100 bits.

(b) UDC mode (TUM10.T1CMD0 bit = 1)

The TMENC10 count triggers in the UDC mode are as follows.

Operation Mode	TMENC10 Operation
Mode 1	Counts down when TCUD10 = high level Counts up when TCUD10 = low level
Mode 2	Counts up upon detection of valid edge of TIUD10 input Counts down upon detection of valid edge of TCUD10 input
Mode 3	Counts up upon detection of valid edge of TIUD10 input when TCUD10 = high level Counts down upon detection of valid edge of TIUD10 input when TCUD10 = low level
Mode 4	Automatic judgment upon detection of both edges of TIUD10 input and both edges of TCUD10 input

(6) Status register 10 (STATUS10)

The STATUS10 register indicates the operating status of TMENC10.

This register is read-only, in 8-bit or 1-bit units.

Reset input clears this register to 00H.

After reset: 00H R Address: FFFFF5CFH

	7	6	5	4	3	<2>	<1>	<0>
STATUS10	0	0	0	0	0	UDF10	OVF10	UBD10

UDF10	TMENC10 underflow flag
0	No TMENC10 count underflow
1	TMENC10 count underflow
The UDF10 bit is cleared (0) upon completion of a read access to the STATUS10 register from the CPU.	

OVF10	TMENC10 overflow flag
0	No TMENC10 count overflow
1	TMENC10 count overflow
The OVF10 bit is cleared (0) upon completion of a read access to the STATUS10 register from the CPU.	

UBD10	TMENC10 count-up/-down operation status
0	TMENC10 count-up in progress
1	TMENC10 count-down in progress
The state of the UBD10 bit differs according to the mode as follows. • The UBD10 bit is fixed to 0 in the general-purpose timer mode (TUM10.T1CMD0 bit = 0). • The UBD10 bit indicates the TMENC10 count-up/-down status in the UDC mode (TUM10.T1CMD0 bit = 1).	

(7) Compare register 100 (CM100)

The CM100 register is a 16-bit register that always compares its value with the value of TMENC10. When the value of the compare register matches the value of TMENC10, an interrupt request signal is generated. The interrupt request signal generation timing in the various modes is described below.

- In the general-purpose timer mode (TUM10.T1CMD0 bit = 0) and UDC mode A (TUM10.MSEL0 bit = 0), an interrupt request signal (INTCM100) is always generated upon occurrence of a match.
- In UDC mode B (TUM10.MSEL0 bit = 1), an interrupt request signal (INTCM100) is generated only upon occurrence of a match during a count-down operation.

This register can be read or written in 16-bit units.
Reset input clears this register to 0000H.

Caution When the TMC10.CE101 bit is 1, it is prohibited to overwrite the value of the CM100 register.

After reset: 0000H R/W Address: FFFFF5C2H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CM100																

(8) Compare register 101 (CM101)

The CM101 register is a 16-bit register that always compares its value with the value of TMENC10. When the value of the compare register matches the value of TMENC10, an interrupt request signal is generated. The interrupt request signal generation timing in the various modes is described below.

- In the general-purpose timer mode (TUM10.T1CMD0 bit = 0) and UDC mode A (TUM10.MSEL0 bit = 0), an interrupt request signal (INTCM101) is always generated upon occurrence of a match.
- In UDC mode B (TUM10.MSEL0 bit = 1), an interrupt request signal (INTCM101) is generated only upon occurrence of a match during a count-down operation.

This register can be read or written in 16-bit units.
Reset input clears this register to 0000H.

Caution When the TMC10.CE101 bit is 1, it is prohibited to overwrite the value of the CM101 register.

After reset: 0000H R/W Address: FFFFF5C4H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CM101																

(9) Capture/compare register 100 (CC100)

The CC100 register is a 16-bit register. It can be specified as a capture register or as a compare register using the CCR10 register.

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

Reset input clears this register to 0000H.

Cautions 1. When used as a capture register (CCR10.CMS00 bit = 0), write access is prohibited.

2. When used as a compare register (CCR10.CMS00 bit = 1) and while TMENC10 is operating (TMC10.CE101 bit is 1), overwriting the CC100 register values is prohibited.

3. When TMENC10 is stopped (TMC10.CE101 bit = 0), the capture trigger is disabled.

4. When the operation mode is changed from capture register to compare register, set a new compare value.

5. Continuous reading of the CC100 register is prohibited. If the CC100 register is continuously read, the second read value may differ from the actual value. If the CC100 register must be read twice, be sure to read another register between the first and the second read operation.

After reset: 0000H R/W Address: FFFFF5C6H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC100																

(a) When set as a capture register

When the CC100 register is set as a capture register, the valid edge of the corresponding external interrupt request signal (INTP10) is detected as the capture trigger. TMENC10 latches the count value in synchronization with the capture trigger (capture operation). The latched value is held in the capture register until the next capture operation.

The valid edge of external interrupt request signals (rising edge, falling edge, both rising and falling edges) is selected by the SESA10 register.

When the CC100 register is specified as a capture register, interrupts are generated upon detection of the valid edge of the INTP10 signal.

Caution TCUD10 and INTP10 are alternate-function pins. Therefore, when the TCUD10 pin is used in the UDC mode, the external capture function of the INTP10 pin cannot be used.

(b) When set as a compare register

When the CC100 register is set as a compare register, it always compares its own value with the value of TMENC10. If the value of the CC100 register matches the value of the TMENC10, the CC100 register generates an interrupt request signal (INTCC100).

(10) Capture/compare register 101 (CC101)

The CC101 register is a 16-bit register. It can be specified as a capture register or as a compare register using the CCR10 register.

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

Reset input clears this register to 0000H.

Cautions 1. When used as a capture register (CCR10.CMS01 bit = 0), write access is prohibited.

2. When used as a compare register (CCR10.CMS01 bit = 1) and while TMENC10 is operating (TMC10.CE101 bit is 1), overwriting the CC101 register values is prohibited.

3. When TMENC10 is stopped (TMC10.CE101 bit = 0), the capture trigger is disabled.

4. When the operation mode is changed from capture register to compare register, newly set a compare value.

5. Continuous reading of the CC101 register is prohibited. If the CC101 register is continuously read, the second read value may differ from the actual value. If the CC101 register must be read twice, be sure to read another register between the first and the second read operation.

After reset: 0000H R/W Address: FFFFF5C8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC101																

(a) When set as a capture register

When the CC101 register is set as a capture register, the valid edge of the corresponding external interrupt request signal (INTP11) is detected as the capture trigger. TMENC10 latches the count value in synchronization with the capture trigger (capture operation). The latched value is held in the capture register until the next capture operation.

The valid edge of external interrupt request signal (rising edge, falling edge, both rising and falling edges) is selected by the SESA10 register.

When the CC101 register is specified as a capture register, interrupts are generated upon detection of the valid edge of the INTP11 signal.

Caution TCLR10 and INTP11 are alternate-function pins. Therefore, when the TCLR10 input is used in UDC mode A, the external capture function of the INTP11 pin cannot be used.

(b) When set as a compare register

When the CC101 register is set as a compare register, it always compares its own value with the value of TMENC10. If the value of the CC101 register matches the value of the TMENC10, the CC101 register generates an interrupt request signal (INTCC101).

11.5 Operation

11.5.1 Operation in general-purpose timer mode

TMENC10 can perform the following operations in the general-purpose timer mode.

(1) Interval operation (when TMC10.ENMD10 bit = 1)

TMENC10 and the CM100 register always compare their values and the INTCM100 interrupt request signal is generated upon occurrence of a match. TMENC10 is cleared (0000H) at the count clock following the match. Furthermore, when one more count clock is input, TMENC10 counts up to 0001H.

The interval time can be calculated with the following formula.

Interval time = (CM100 register value + 1) × TMENC10 count clock rate

(2) Free-running operation (when TMC10.ENMD10 bit = 0)

TMENC10 fully counts from 0000H to FFFFH, is cleared to 0000H at the next count clock after the STATUS10.OVF10 bit has been set (1), and continues counting.

The free-running cycle can be calculated by the following formula.

Free-running cycle = 65,536 × TMENC10 count clock rate

(3) Compare function

TMENC10 connects two compare register (CM100, CM101) channels and two capture/compare register (CC100, CC101) channels.

When the TMENC10 count value and the set value of one of the compare registers match, a match interrupt request signal (INTCM100, INTCM101, INTCC100[Note], INTCC101[Note]) is output. Particularly in the case of interval operation, TMENC10 is cleared upon generation of the INTCM100 interrupt.

Note This match interrupt request signal is generated when the CC100 and CC101 registers are set to the compare register mode.

(4) Capture function

TMENC10 connects two capture/compare register (CC100, CC101) channels.

When the CC100 and CC101 registers are set to the capture register mode, the value of TMENC10 is captured in synchronization with the corresponding capture trigger signal.

Furthermore, an interrupt request signal (INTCC100, INTCC101) is generated by the valid edge of the INTP10 and INTP11 input signals specified as the capture trigger signals.

Table 11-3. Capture Trigger Signal to 16-Bit Capture Register

Capture Register	Capture Trigger Signal
CC100	INTP10
CC101	INTP11

Remark CC100 and CC101 registers are capture/compare registers. Which of these registers is used is specified by the CCR10 register.

The valid edge of the capture trigger is specified by the SESA10 register. If both the rising edge and the falling edge are selected as the capture triggers, it is possible to measure the input pulse width externally. If a single edge is selected as the capture trigger, the input pulse cycle can be measured.

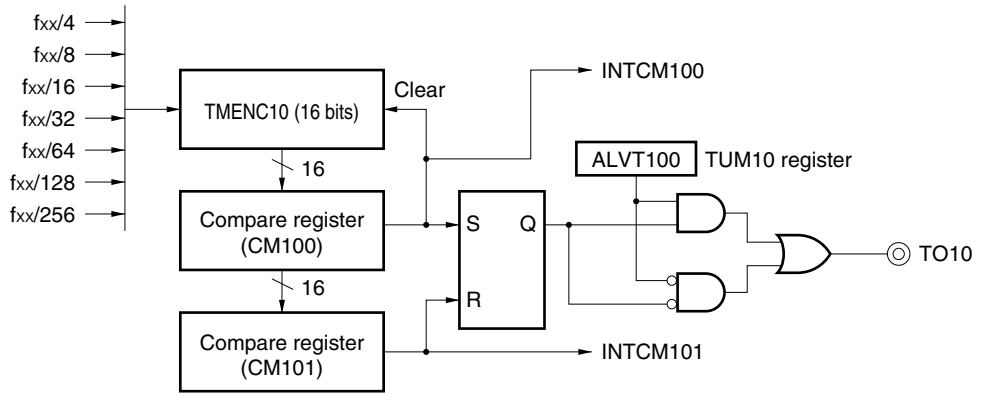
(5) Timer output operation

Timer output operation is performed from the TO10 pin by setting TMENC10 to the general-purpose timer mode (T1CMD0 bit = 0) using the TUM10 register.

During the timer output operation, the cycle and duty (CM100 and CM101 registers) cannot be rewritten.

The resolution is 16 bits, and the count clock can be selected from among seven internal clocks (fxx/4, fxx/8, fxx/16, fxx/32, fxx/64, fxx/128, fxx/256).

Figure 11-2. TMENC10 Block Diagram (During Timer Output Operation)



Caution Be sure to set the count clock of TMENC10 to 10 MHz or lower.

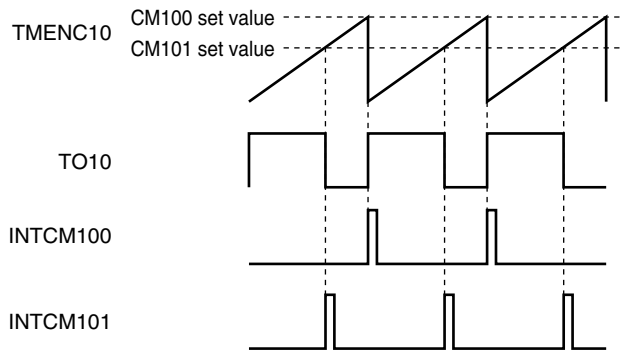
Remark fxx: Peripheral clock

(a) Description of operation

The CM100 register is a compare register used to set the timer output cycle. When the value of this register matches the value of TMENC10, the INTCM100 interrupt request signal is generated. The compare match is saved by hardware, and TMENC10 is cleared at the next count clock after the match.

The CM101 register is a compare register used to set the timer output duty. Set the duty required for the timer cycle.

Figure 11-3. Timer Output Example (When ALVT100 Bit = 0)



Cautions 1. Changing the values of the CM100 and CM101 registers is prohibited during TMENC10 operation (TMC10.CE101 bit = 1).

2. Changing the value of the TUM10.ALVT100 bit is prohibited during TMENC10 operation.

3. Timer output is performed from the second timer cycle after the CE101 bit is set (to 1).

11.5.2 Operation in UDC mode

(1) Overview of operation in UDC mode

The count clock input to TMENC10 in the UDC mode (TUM10.T1CMD0 bit = 1) can only be externally input from the TIUD10 and TCUD10 pins. Count-up/-down judgment in the UDC mode is determined based on the phase difference of the TIUD10 and TCUD10 pin inputs according to the PRM10 register setting (there is a total of four choices).

Table 11-4. List of Count Operations in UDC Mode

PRM10 Register			Operation Mode	TMENC10 Operation
PRM102	PRM101	PRM100		
1	0	0	Mode 1	Counts down when TCUD10 = high level Counts up when TCUD10 = low level
1	0	1	Mode 2	Counts up upon detection of valid edge of TIUD10 input Counts down upon detection of valid edge of TCUD10 input
1	1	0	Mode 3	Counts up upon detection of valid edge of TIUD10 input when TCUD10 = high level Counts down upon detection of valid edge of TIUD10 input when TCUD10 = low level
1	1	1	Mode 4	Automatic judgment upon detection of both edges of TIUD10 input and both edges of TCUD10 input

The UDC mode is further divided into two modes according to the TMENC10 clear conditions (a count operation is performed only with TIUD10 and TCUD10 input in both modes).

- **UDC mode A (TUM10.T1CMD0 bit = 1, TUM10.MSEL0 bit = 0)**

  The TMENC10 clear source can be selected as only external clear input (TCLR10), a match signal between the TMENC10 count value and the CM100 register set value during count-up operation, or the logical sum (OR) of the two signals, using TMC10.CLR101 and TMC10.CLR100 bits. TMENC10 can transfer the value of the CM100 register upon occurrence of a TMENC10 underflow.

- **UDC mode B (TUM10.T1CMD0 bit = 1, TUM10.MSEL0 bit = 1)**

  The status of TMENC10 after a match of the TMENC10 count value and CM100 register set value is as follows.

  <1> In the case of a count-up operation, TMENC10 is cleared (0000H), and the INTCM100 interrupt request signal is generated.

  <2> In the case of a count-down operation, the TMENC10 count value is decremented (–1).

  The status of TMENC10 after a match of the TMENC10 count value and CM101 register set value is as follows.

  <1> In the case of a count-up operation, the TMENC10 count value is incremented (+1).

  <2> In the case of a count-down operation, TMENC10 is cleared (0000H), and the INTCM101 interrupt request signal is generated.

(2) Count-up/-down operation in UDC mode

TMENC10 count-up/-down judgment in the UDC mode is determined based on the phase difference of the TIUD10 and TCUD10 pin inputs according to the PRM10 register setting.

(a) Mode 1 (PRM10.PRM102 bit = 1, PRM10.PRM101 bit = 0, PRM10.PRM100 bit = 0)

In mode 1, the following count operations are performed based on the level of the TCUD10 pin upon detection of the valid edge of the TIUD10 pin.

- TMENC10 count-down operation when TCUD10 pin = high level
- TMENC10 count-up operation when TCUD10 pin = low level

Figure 11-4. Mode 1 (When Rising Edge Is Specified as Valid Edge of TIUD10 Pin)

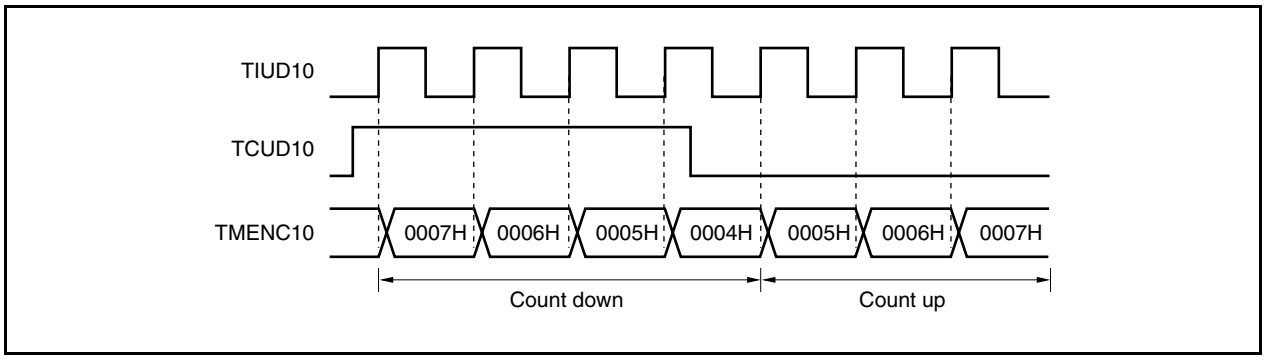
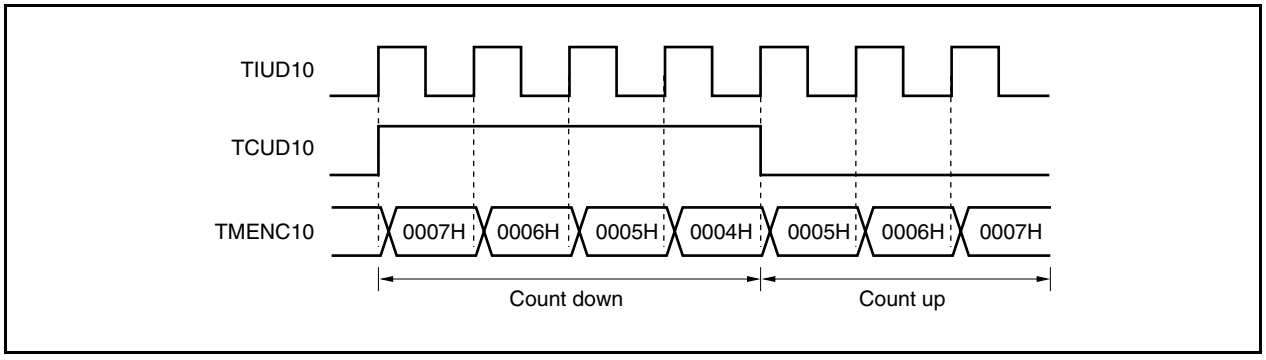


Figure 11-5. Mode 1 (When Rising Edge Is Specified as Valid Edge of TIUD10 Pin): In Case of Simultaneous TCUD10, TCUD10 Pin Edge Timing



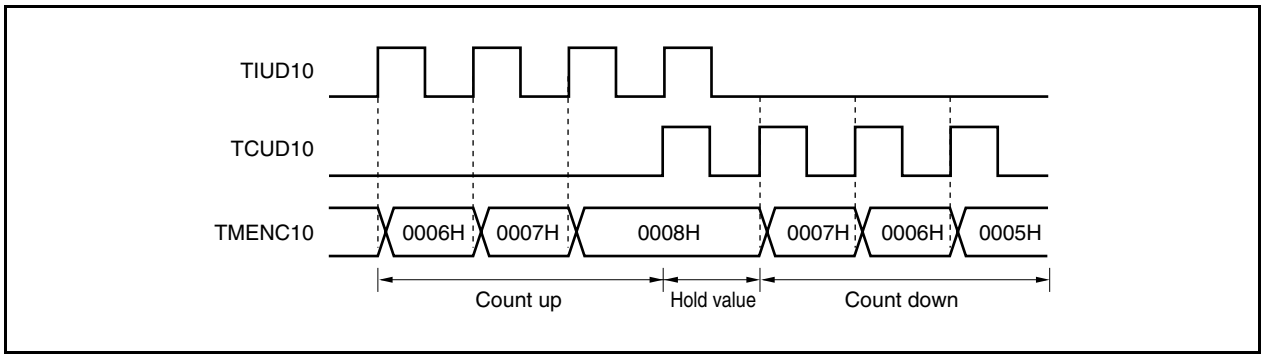
(b) Mode 2 (PRM10.PRM102 bit = 1, PRM10.PRM101 bit = 0, PRM10.PRM100 bit = 1)

The count conditions in mode 2 are as follows.

- TMENC10 count-up upon detection of valid edge of TIUD10 pin
- TMENC10 count-down upon detection of valid edge of TCUD10 pin

Caution If the count clock is simultaneously input to the TIUD10 pin and the TCUD10 pin, a count operation is not performed and the immediately preceding value is held.

Figure 11-6. Mode 2 (When Rising Edge Is Specified as Valid Edge of TIUD10, TCUD10 Pins)



(c) Mode 3 (PRM10.PRM102 = 1, PRM10.PRM101 = 1, PRM10.PRM100 = 0)

In mode 3, when two signals 90 degrees out of phase are input to the TIUD10 and TCUD10 pins, the level of the TCUD10 pin is sampled at the input of the valid edge of the TIUD10 pin (see **Figure 11-7**).

If the TCUD10 pin level sampled at the valid edge input to the TIUD10 pin is low, TMENC10 counts down when the valid edge is input to the TIUD10 pin.

If the TCUD10 pin level sampled at the valid edge input to the TIUD10 pin is high, TMENC10 counts up when the valid edge is input to the TIUD10 pin.

Figure 11-7. Mode 3 (When Rising Edge Is Specified as Valid Edge of TIUD10 Pin)

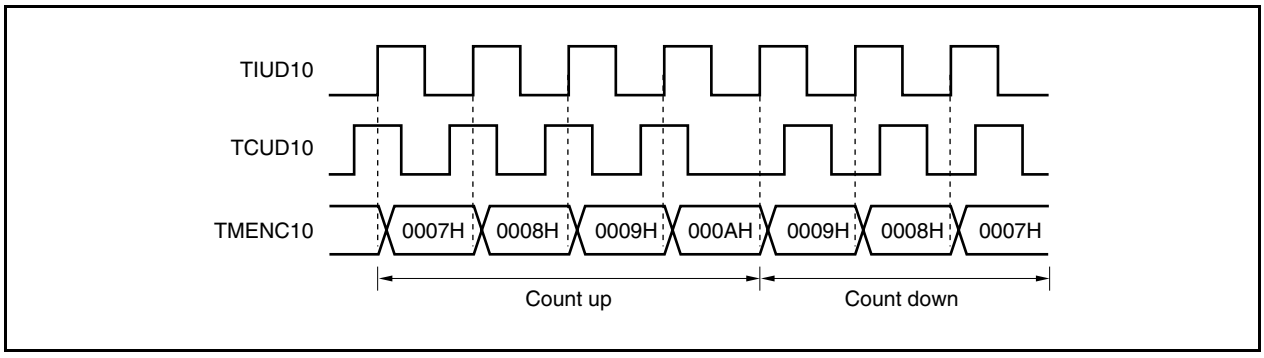
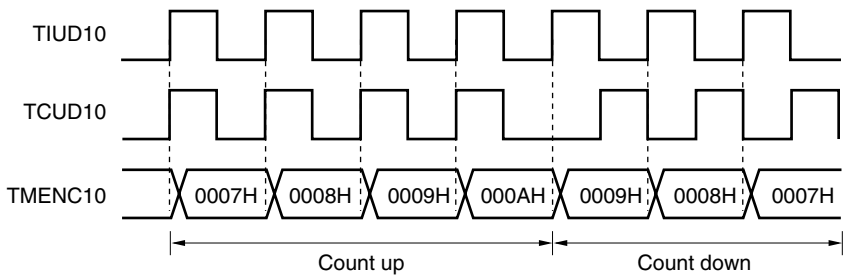


Figure 11-8. Mode 3 (When Rising Edge Is Specified as Valid Edge of TIUD10 Pin): In Case of Simultaneous TIUD10, TCUD10 Pin Edge Timing

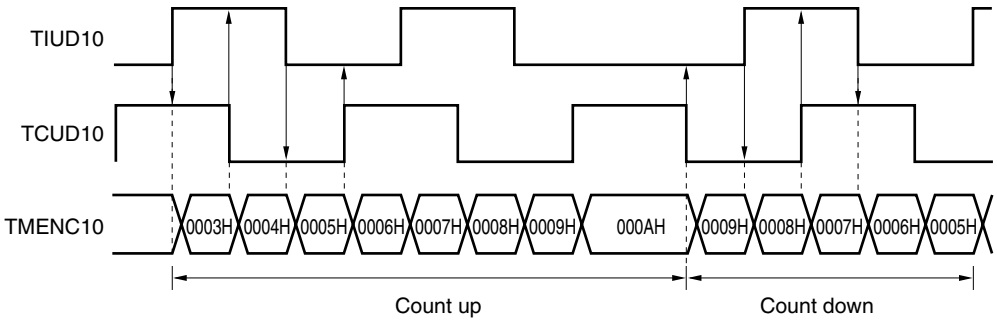


(d) Mode 4 (PRM10.PRM102 = 1, PRM10.PRM101 = 1, PRM10.PRM100 = 1)

In mode 4, when two signals out of phase are input to the TIUD10 and TCUD10 pins, the up/down operation is automatically judged and counting is performed according to the timing shown in **Figure 11-9**.

In mode 4, counting is executed at both the rising and falling edges of the two signals input to the TIUD10 and TCUD10 pins. Therefore, TMENC10 counts four times per cycle of an input signal (×4 count).

Figure 11-9. Mode 4



Cautions 1. When mode 4 is specified as the operation mode of TMENC10, the valid edge specifications for the TIUD10 and TCUD10 pins are not valid.

2. If the TIUD10 pin edge and TCUD10 pin edge are input simultaneously in mode 4, TMENC10 continues the same count operation (up or down) it was performing immediately before the input.

(3) Operation in UDC mode A

(a) Interval operation

The operations at the count clock following a match of the TMENC10 count value and the CM100 register set value are as follows.

- In case of count-up operation: TMENC10 is cleared (0000H) and the INTCM100 interrupt request signal is generated.
- In case count-down operation: The TMENC10 count value is decremented (−1) and the INTCM100 interrupt request signal is generated.

Remark The interval operation can be combined with the transfer operation.

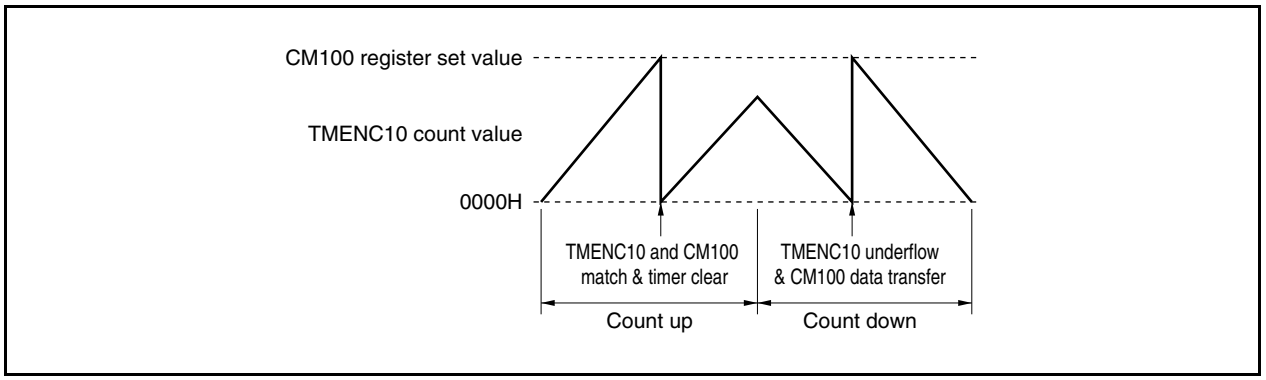
(b) Transfer operation

If TMENC10 = 0000H during down counting when the TMC10.RLEN10 bit = 1, the set value of the CM100 register is transferred to TMENC10 at the next count clock.

Remarks 1. Transfer enable/disable can be set using the TMC10.RLEN10 bit.

2. The transfer operation can be combined with the interval operation.

Figure 11-10. Example of TMENC10 Operation When Interval Operation and Transfer Operation Are Combined



(c) Compare function

TMENC10 connects two compare register (CM100, CM101) channels and two capture/compare register (CC100, CC101) channels.

When the TMENC10 count value and the set value of one of the compare registers match, a match interrupt request signal (INTCM100, INTCM101, INTCC100[Note], INTCC101[Note]) is output.

Note This match interrupt request signal is generated when the CC100 and CC101 registers are set to the compare register mode.

(d) Capture function

TMENC10 connects two capture/compare register (CC100, CC101) channels.

When the CC100 and CC101 registers are set to the capture register mode, the value of TMENC10 is captured in synchronization with the corresponding capture trigger signal.

A capture interrupt request signal (INTCC100, INTCC101) is generated upon detection of the valid edge.

(4) Operation in UDC mode B

(a) Basic operation

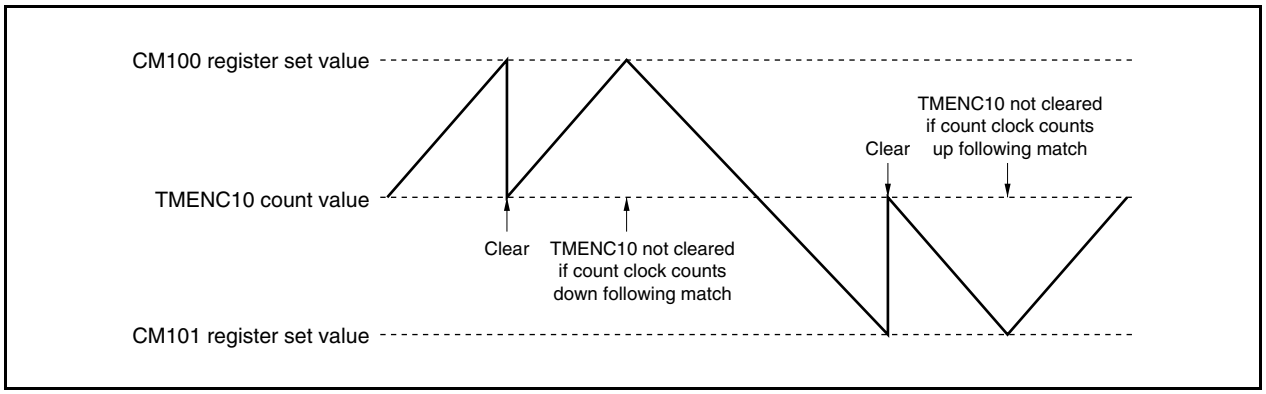
The operations at the next count clock after the count value of TMENC10 and the CM100 register set value match when TMENC10 is in UDC mode B are as follows.

- In case of count-up operation: TMENC10 is cleared (0000H) and the INTCM100 interrupt request signal is generated.
- In case of count-down operation: The TMENC10 count value is decremented (−1).

The operations at the next count clock after the count value of TMENC10 and the CM101 register set value match when TMENC10 is in UDC mode B are as follows.

- In case of count-up operation: The TMENC10 count value is incremented (+1).
- In case of count-down operation: TMENC10 is cleared (0000H) and the INTCM101 interrupt request signal is generated.

Figure 11-11. Example of TMENC10 Operation in UDC Mode



(b) Compare function

TMENC10 connects two compare register (CM100, CM101) channels and two capture/compare register (CC100, CC101) channels.

When the TMENC10 count value and the set value of one of the compare registers match, a match interrupt request signal (INTCM100 (only during count-up operation), INTCM101 (only during count-down operation), INTCC100[Note], INTCC101[Note]) is output.

Note This match interrupt request signal is generated when the CC100 and CC101 registers are set to the compare register mode.

(c) Capture function

TMENC10 connects two capture/compare register (CC100, CC101) channels.

When CC100 and CC101 registers are set to the capture register mode, the value of TMENC10 is captured in synchronization with the corresponding capture trigger signal. A capture interrupt request signal (INTCC100, INTCC101) is generated upon detection of the valid edge.

11.6 Supplementary Description of Internal Operation

11.6.1 Clearing of count value in UDC mode B

When TMENC10 is in UDC mode B, the conditions to clear the count value are as follows.

- In case of TMENC10 count-up operation: TMENC10 count value is cleared upon match with CM100 register
- In case of TMENC10 count-down operation: TMENC10 count value is cleared upon match with CM101 register

Figure 11-12. Clear Operation After Match of CM100 Register Set Value and TMENC10 Count Value

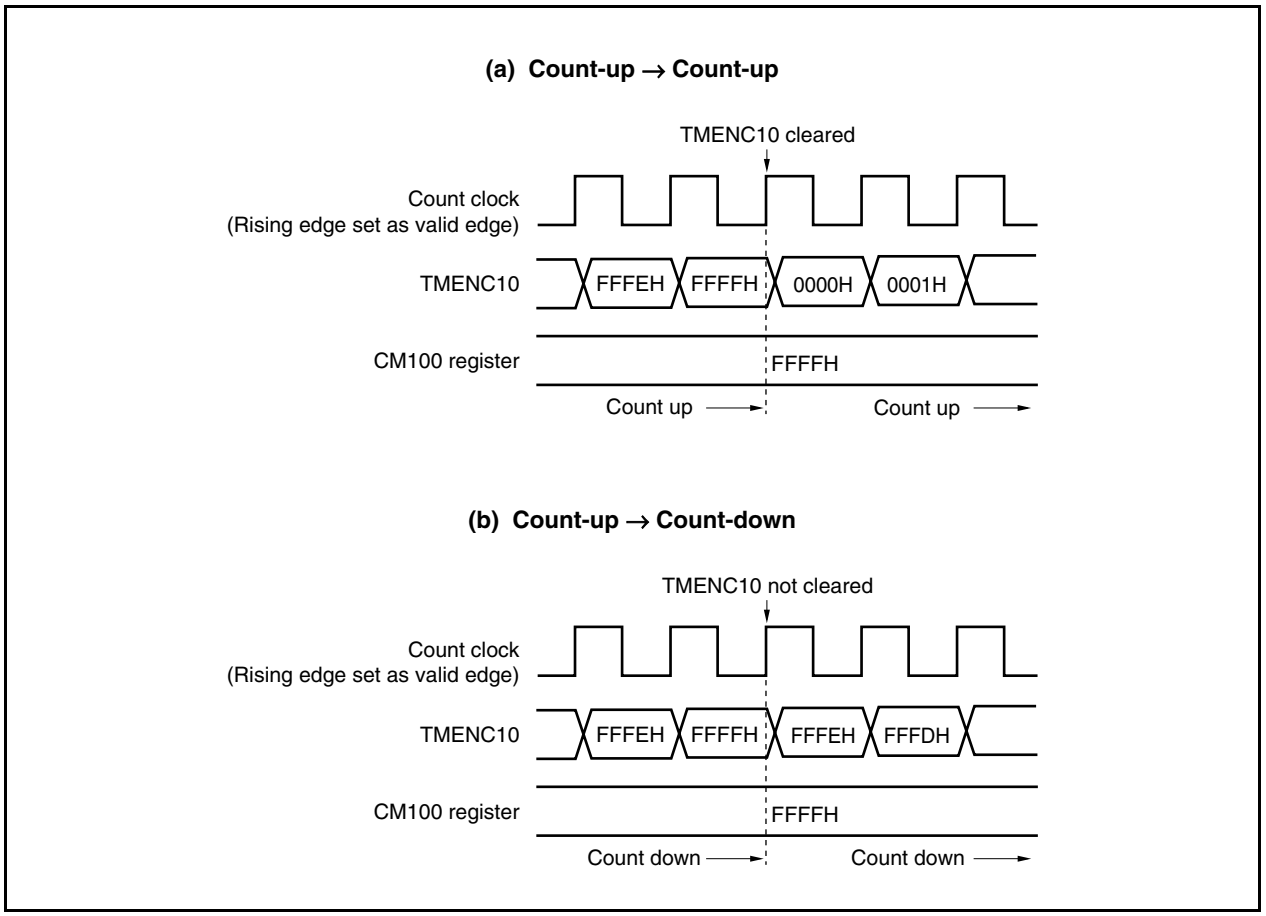
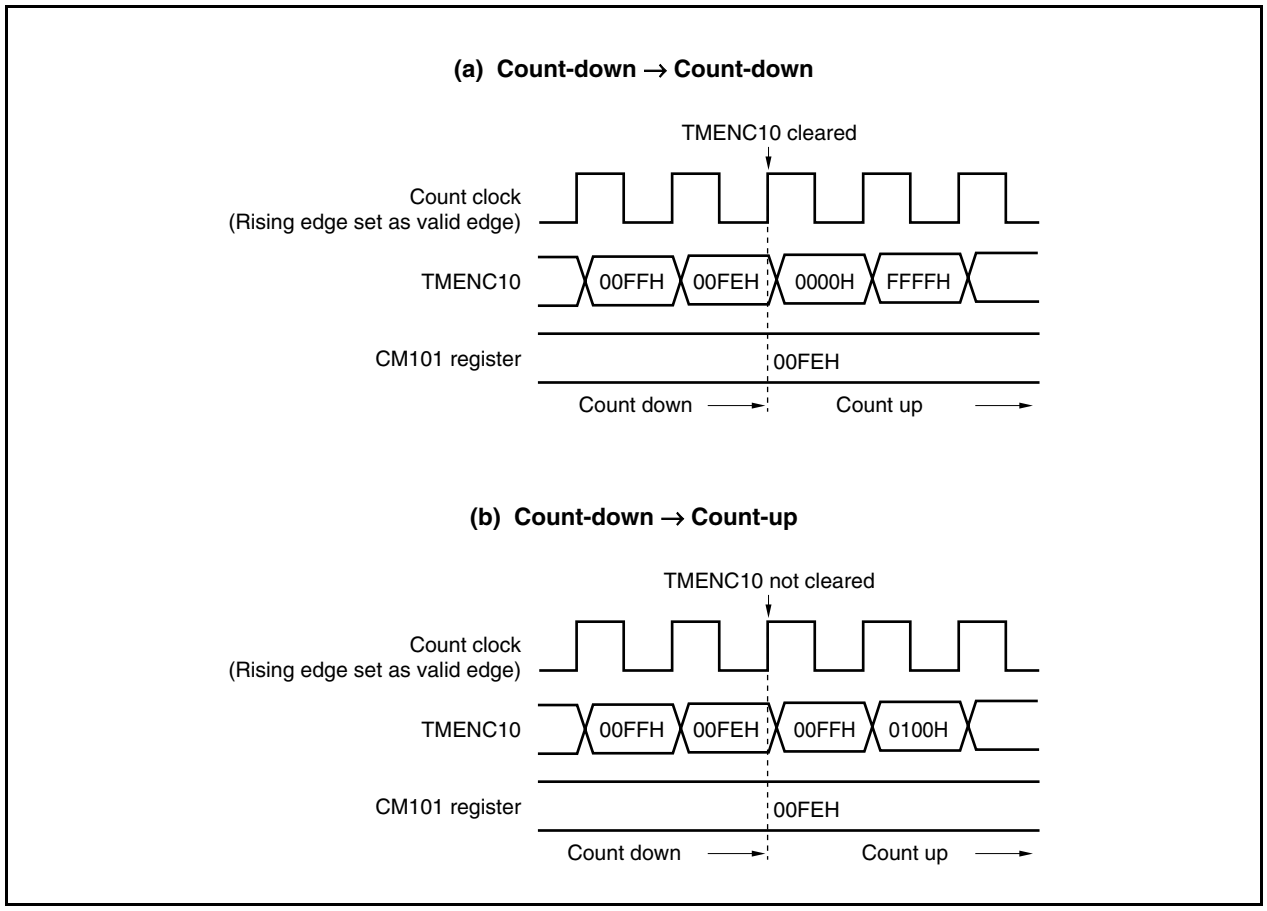


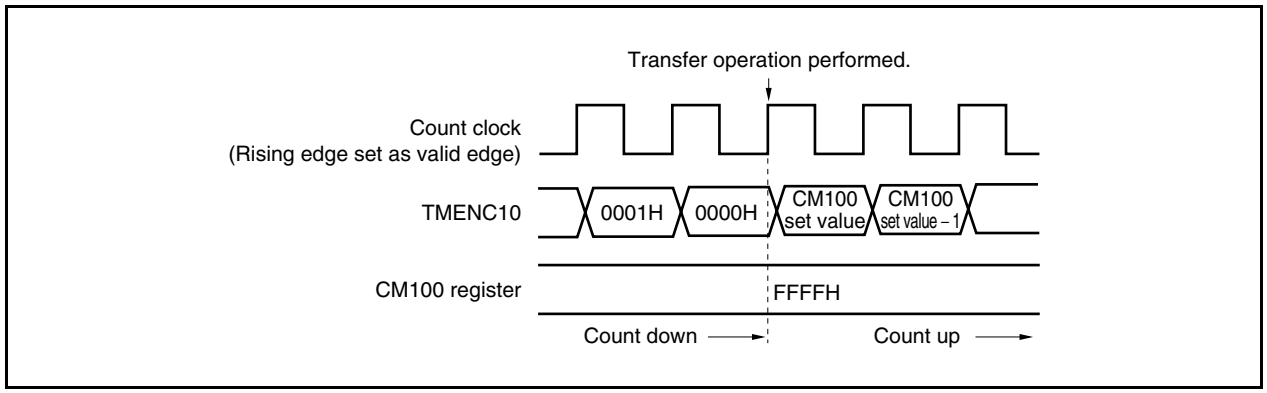
Figure 11-13. Clear Operation After Match of CM101 Register Set Value and TMENC10 Count Value



11.6.2 Transfer operation

If TMENC10 = 0000H during down counting when the TMC10.RLEN10 bit = 1 in UDC mode A, the set value of the CM100 register is transferred to TMENC10 at the next count clock. The transfer operation is not performed during up counting.

Figure 11-14. Internal Operation During Transfer Operation

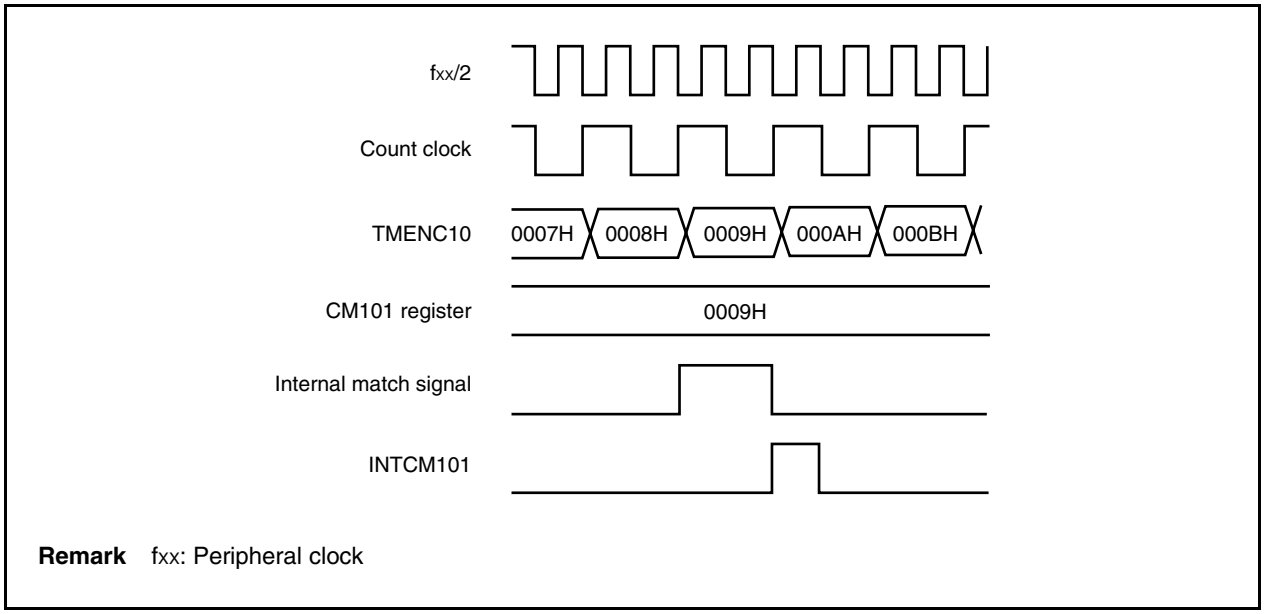


11.6.3 Interrupt request signal output upon compare match

An interrupt request signal is output when the count value of TMENC10 matches the set value of the CM100, CM101, CC100[Note], or CC101[Note] register. The interrupt request timing generation timing is as follows.

Note When the CC100 and CC101 registers are set to the compare register mode.

Figure 11-15. Interrupt Request Signal Output upon Compare Match (CM101 with Operation Mode Set to General-Purpose Timer Mode and Count Clock Set to fxx/4)



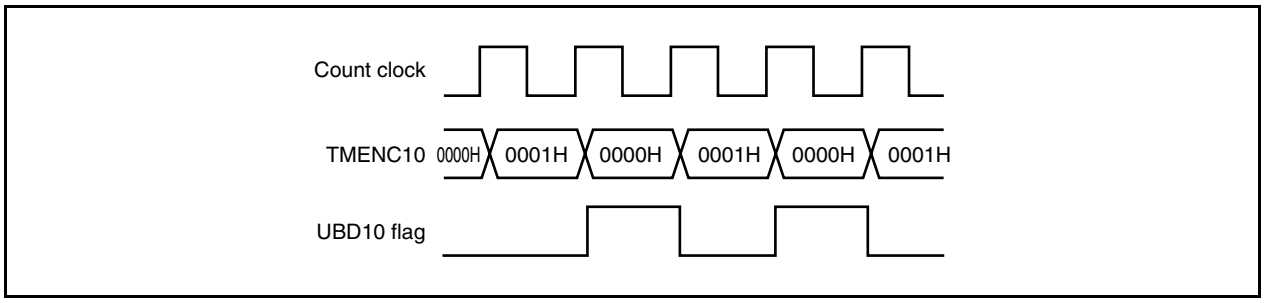
Remark fxx: Peripheral clock

An interrupt request signal such as the one illustrated in Figure 11-15 is output at the next count clock following a match of the TMENC10 count value and the set value of the corresponding compare register.

11.6.4 UBD10 flag (bit 0 of STATUS10 register) operation

In the UDC mode (TUM10.T1CMD0 bit = 1), the UBD10 flag changes as follows during a TMENC10 count-up/-down operation at every internal operation clock.

Figure 11-16. UBD10 Flag Operation



CHAPTER 12 MOTOR CONTROL FUNCTION

12.1 Functional Overview

Timer Q0 (TMQ0) and the TMQ0 option (TMQOP0) can be used as an inverter function that controls a motor. It performs a tuning operation with timer P2 (TMP2) and A/D conversion of the A/D converter can be started when the value of TMQ0 matches the value of TMP2. 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 TMP2)
- 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 TMQ0 operation)
- Interrupt and transfer culling functions
- Dead-time setting function
- A/D trigger timing function of the A/D converter (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 ($\overline{\text{INTP000}}$)

12.2 Configuration

The motor control function consists of the following hardware.

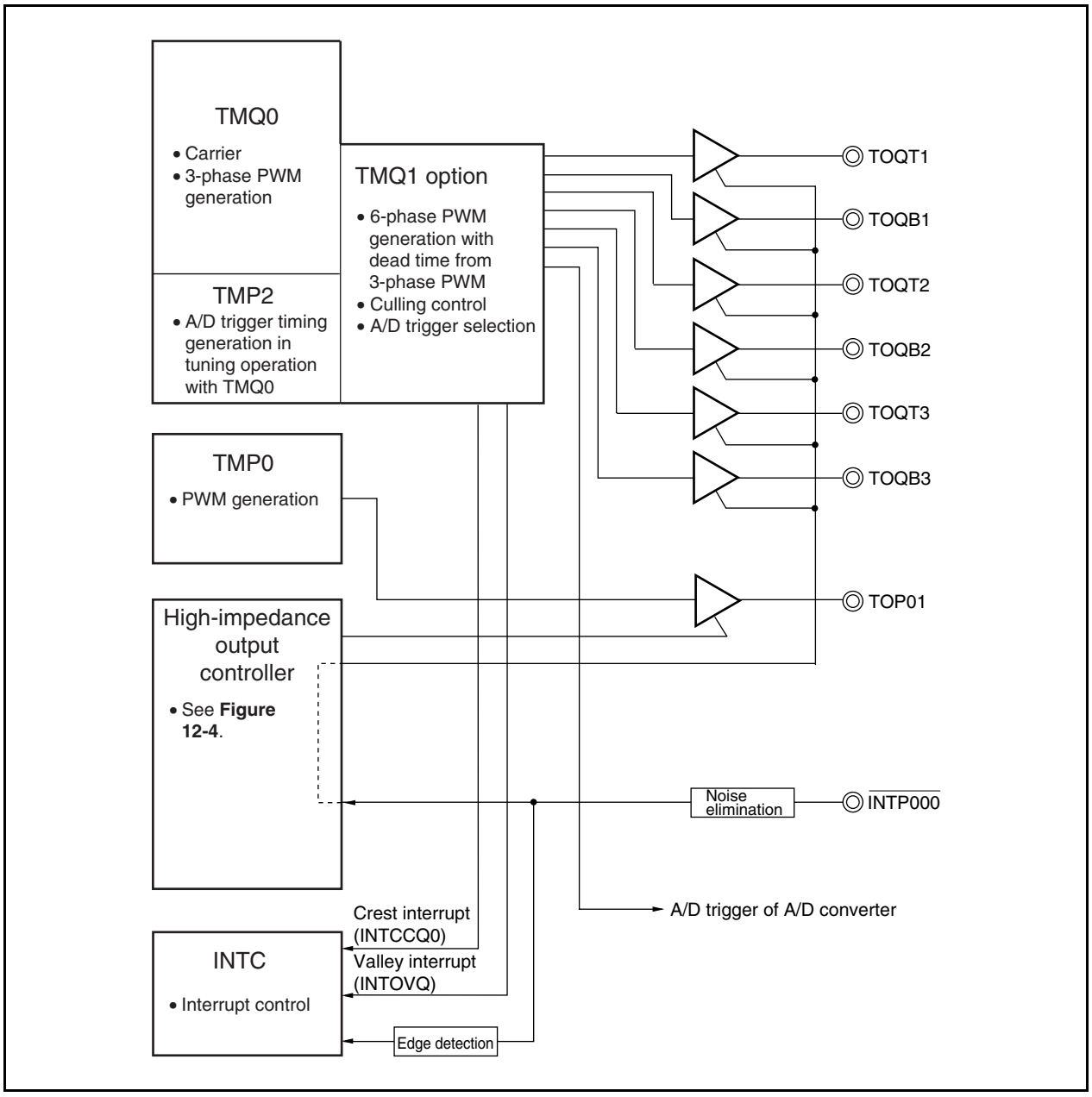
Item	Configuration
Timer register	Dead-time counter m
Compare register	TMQ0 dead-time compare register (TQ0DTC register)
Control registers	TMQ0 option register 0 (TQ0OPT0) TMQ0 option register 1 (TQ0OPT1) TMQ0 option register 2 (TQ0OPT2) TMQ0 I/O control register 3 (TQ0IOC3) High-impedance output control registers 0, 1 (HZA0CTL0, HZA0CTL1)

Remark m = 0 to 3

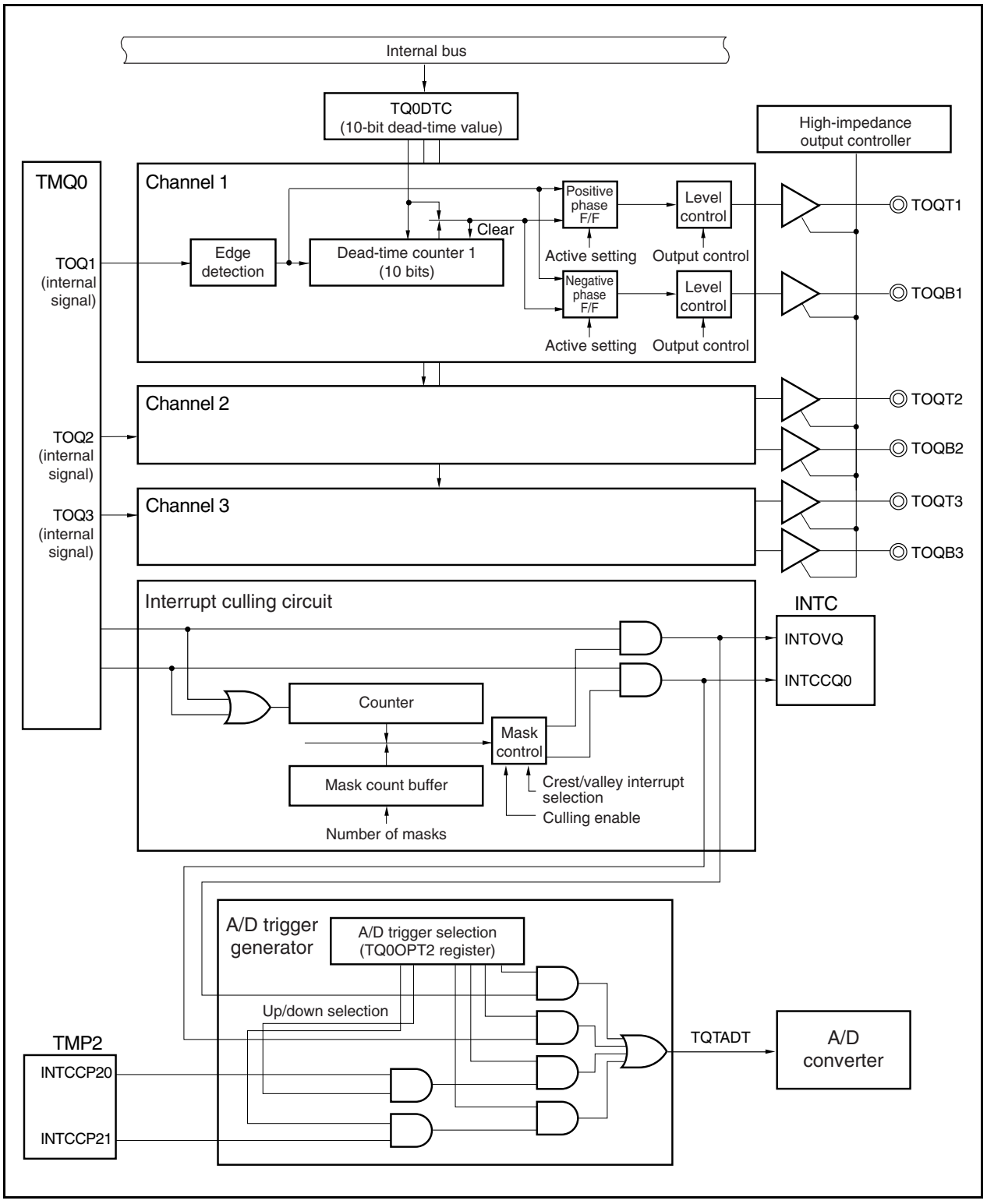
- 6-phase PWM output can be produced with dead time by using the output of TMQ0 (TOQ1, TOQ2, TOQ3)
- The output level of the 6-phase PWM output can be set individually.
- The 16-bit timer/counter of TMQ0 counts up/down triangular waves. When the timer/counter underflows and when a cycle match occurs, an interrupt is generated. Interrupt generation, however, can be suppressed up to 31 times.
- TMP2 can execute counting at the same time as TMQ0 (timer tuning operation function). TMP2 can be set in four ways as it can generate two types of A/D trigger sources (INTCCP20 and INTCCP21), and two types of interrupts: on underflow interrupt (INTOVQ) and cycle match interrupt (INTCCQ0).

<R>

Figure 12-1. Block Diagram of Motor Control



<R> **Figure 12-2. TMQ0 Option**



(1) TMQ0 dead-time compare register (TQ0DTC)

The TQ0DTC register is a 10-bit compare register that specifies a dead-time value.

Rewriting this register is prohibited when the TQ0CTL0.TQ0CE bit = 1.

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

Reset input clears this register to 0000H.

After reset: 0000H R/W Address: FFFFF624H

	15	10	9	0
TQ0DTC	000000		TQ0DTC9 to TQ0DTC0	

(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 TOQm output signal by TMQ0, and are cleared or stopped when their count value matches the value of the TQ0DTC register. The count clock of these counters is the same as that set by the TQ0CTL0.TQ0CKS2 to TQ0CTL0.TQ0CKS0 bits of TMQ0.

Remarks 1. The operation differs when the TQ0OPT2.TQ0DTM bit = 1. For details, see **12.4.2 (4) Automatic dead-time width narrowing function (TQ0OPT2.TQ0DTM bit = 1)**.

2. m = 1 to 3

12.3 Control Registers

(1) TMQ0 option register 0 (TQ0OPT0)

The TQ0OPT0 register is an 8-bit register that controls the timer Q0 option function.

This register can be read or written in 8-bit or 1-bit units. However, the TQ0CUF bit is read-only.

Reset input clears this register to 00H.

Caution The TQ0CMS and TQ0CUF bits can be set only in the 6-phase PWM output mode. Be sure to clear these bits to 0 when TMQ0 is used alone.

After reset: 00H R/W Address: FFFFF605H

	7	6	5	4	3	<2>	<1>	<0>
TQ0OPT0	0	0	0	0	0	TQ0CMS	TQ0CUF	TQ0OVF[Note]

TQ0CMS	Compare register rewrite mode selection
0	Batch rewrite mode (transfer operation)
1	Anytime rewrite mode

- The TQ0CMS bit is valid only when the 6-phase PWM output mode is set (when the TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 bits = 111). Clear the TQ0CMS bit to 0 in any other mode.
- The TQ0CMS bit can be rewritten while the timer is operating (when the TQ0CTL0.TQ0CE bit = 1).
- The following compare registers are rewritten in the batch write mode. TQ0CCR0 to TQ0CCR3, TP2CCR0, TP2CCR1, and TQ0OPT1 registers

TQ0CUF	Count-up/Count-down flag of timer Q0
0	Timer Q0 is counting up.
1	Timer Q0 is counting down.

The TQ0CUF bit is valid only when the 6-phase PWM output mode is set (when the TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 bits = 111).

Note For details of the TQ0OVF bit, see **CHAPTER 9 16-BIT TIMER/EVENT COUNTER Q (TMQ)**.

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

(2) TMQ0 option register 1 (TQ0OPT1)

The TQ0OPT1 register is an 8-bit register that controls the interrupt request signal generated by the timer Q0 option function.

This register can be rewritten when the TQ0CTL0.TQ0CE bit is 1.

Two rewrite modes (batch write mode and anytime write mode) can be selected, depending on the setting of the TQ0OPT0.TQ0CMS bit.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF620H

	<7>	<6>	5	4	3	2	1	0
TQ0OPT1	TQ0ICE	TQ0IOE	0	TQ0ID4	TQ0ID3	TQ0ID2	TQ0ID1	TQ0ID0

TQ0ICE	Crest interrupt (INTCCQ0 signal) enable
0	Do not use INTCCQ0 signal (do not use it as count signal for interrupt culling).
1	Use INTCCQ0 signal (use it as count signal for interrupt culling).

TQ0IOE	Valley interrupt (INTOVQ signal) enable
0	Do not use INTOVQ signal (do not use it as count signal for interrupt culling).
1	Use INTOVQ signal (use it as count signal for interrupt culling).

TQ0ID4	TQ0ID3	TQ0ID2	TQ0ID1	TQ0ID0	Number of times of interrupt
0	0	0	0	0	Not culled (all interrupts are output)
0	0	0	0	1	1 masked (one of two interrupts is output)
0	0	0	1	0	2 masked (one of three interrupts is output)
0	0	0	1	1	3 masked (one of four interrupts is output)
:	:	:	:	:	:
1	1	1	0	0	28 masked (one of 29 interrupts is output)
1	1	1	0	1	29 masked (one of 30 interrupts is output)
1	1	1	1	0	30 masked (one of 31 interrupts is output)
1	1	1	1	1	31 masked (one of 32 interrupts is output)

(3) TMQ0 option register 2 (TQ0OPT2)

The TQ0OPT2 register is an 8-bit register that controls the timer Q0 option function.

This register can be rewritten when the TQ0CTL0.TQ0CE bit is 1. However, rewriting the TQ0DTM bit is prohibited when the TQ0CE bit is 1. The same value can be rewritten.

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

Reset input clears this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF621H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TQ0OPT2	TQ0RDE	TQ0DTM	TQ0ATM3	TQ0ATM2	TQ0AT3	TQ0AT2	TQ0AT1	TQ0AT0

TQ0RDE	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 TQ0OPT1 register.

TQ0DTM	Dead-time counter operation mode selection
0	Dead-time counter counts up normally and, if TOQm output of TMQ0 is at a narrow interval (TOQm output width < dead-time width), the dead-time counter is cleared and counts up again.
1	Dead-time counter counts up normally and, if TOQm output of TMQ0 is at a narrow interval (TOQm output width < dead-time width), the dead-time counter counts down and the dead-time control width is automatically narrowed.
Rewriting the TQ0DTM bit is disabled during timer operation. If it is rewritten by mistake, stop the timer operation by clearing the TQ0CE bit to 0, and re-set the TQ0DTM bit.	

TQ0ATM3	TQ0ATM3 mode selection
0	Output A/D trigger signal (TQTADT) for INTCCP21 interrupt while dead-time counter is counting up.
1	Output A/D trigger signal (TQTADT) for INTCCP21 interrupt while dead-time counter is counting down.

TQ0ATM2	TQ0ATM2 mode selection
0	Output A/D trigger signal (TQTADT) for INTCCP20 interrupt while dead-time counter is counting up.
1	Output A/D trigger signal (TQTADT) for INTCCP20 interrupt while dead-time counter is counting down.

Caution When using interrupt culling (the TQ0OPT1.TQ0ID4 to TQ0OPT1.TQ0ID0 bits are set to other than 00000), be sure to set the TQ0RDE bit to 1.

<R> **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 (TQ0RDE bit = 0), transfer is not performed normally.**

Remark m = 1 to 3

(2/2)

TQ0AT3[Note]	A/D trigger output control 3
0	Disable output of A/D trigger signal (TQTADT) for INTCCP21 interrupt.
1	Enable output of A/D trigger signal (TQTADT) for INTCCP21 interrupt.

TQ0AT2[Note]	A/D trigger output control 2
0	Disable output of A/D trigger signal (TQTADT) for INTCCP20 interrupt.
1	Enable output of A/D trigger signal (TQTADT) for INTCCP20 interrupt.

TQ0AT1[Note]	A/D trigger output control 1
0	Disable output of A/D trigger signal (TQTADT) for INTCCQ0 (crest interrupt).
1	Enable output of A/D trigger signal (TQTADT) for INTCCQ0 (crest interrupt).

TQ0AT0[Note]	A/D trigger output control 0
0	Disable output of A/D trigger signal (TQTADT) for INTOVQ (valley interrupt).
1	Enable output of A/D trigger signal (TQTADT) for INTOVQ (valley interrupt).

Note For the setting of the TQ0AT3 to TQ0AT0 bits, see **CHAPTER 14 A/D CONVERTER**.

(4) TMQ0 I/O control register 3 (TQ0IOC3)

The TQ0IOC3 register is an 8-bit register that controls the output of the timer Q0 option function.

To output from the TOQTm pin, set the TQ0IOC0.TQ0OEm bit to 1 and then set the TQ0IOC3 register.

The TQ0IOC3 register can be rewritten only when the TQ0CTL0.TQ0CE bit is 0.

Rewriting each bit of the TQ0IOC3 register is prohibited when the TQ0CTL0.TQ0CE bit is 1; however the same value can be rewritten to each bit of the TQ0IOC3 register when the TQ0CTL0.TQ0CE bit is 1.

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

Reset input sets this register to A8H.

Caution Set the TQ0IOC3 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 TOQTm pin by the TQ0IOC0 register.

After reset: A8H R/W Address: FFFFF622H

	<7>	<6>	<5>	<4>	<3>	<2>	1	0
TQ0IOC3	TQ0OLB3	TQ0OEB3	TQ0OLB2	TQ0OEB2	TQ0OLB1	TQ0OEB1	0	0

TQ0OLBm	Setting of TOQBm pin output level (m = 1 to 3)
0	Disable inversion of output of TOQBm pin
1	Enable inversion of output of TOQBm pin

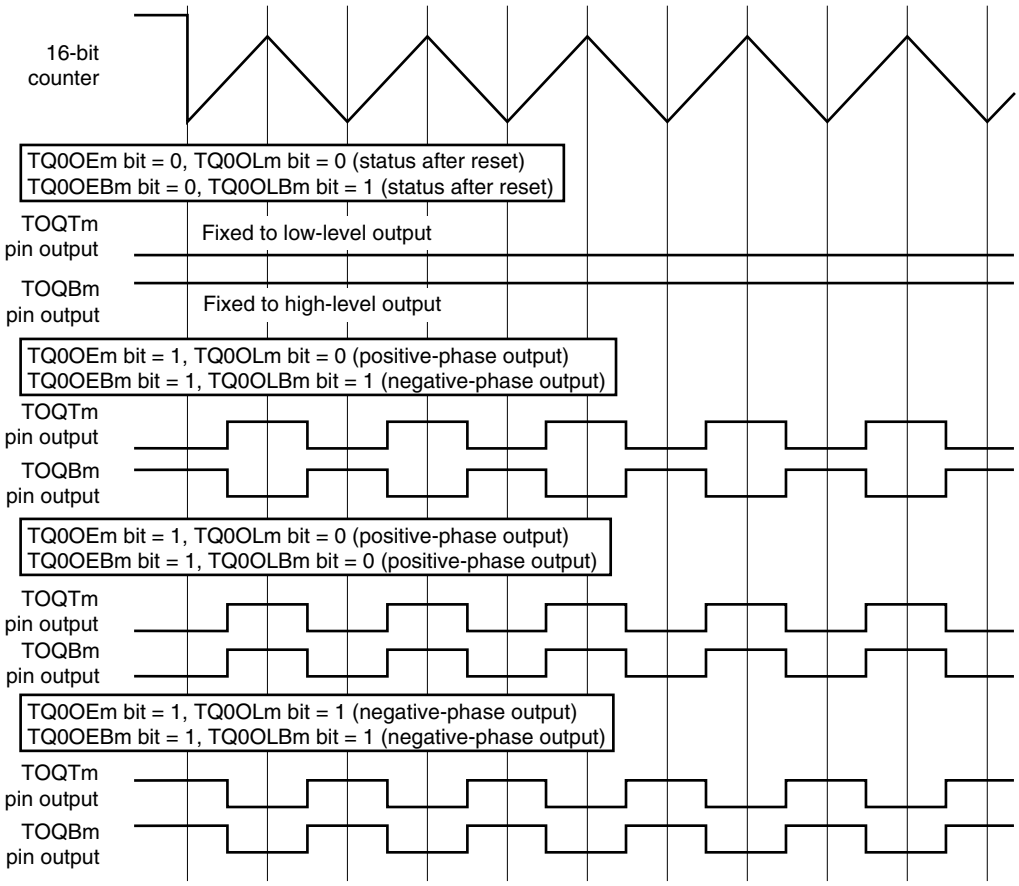
TQ0OEBm	TOQBm pin output (m = 1 to 3)
0	Disable TOQBm pin output. • When TQ0OLBm bit = 0, low level is output from TOQBm pin. • When TQ0OLBm bit = 1, high level is output from TOQBm pin.
1	Enable TOQBm pin output.

(a) Output from TOQTm and TOQBm pins

The TOQTm pin output is controlled by the TQ0IOC0.TQ0OLm and TQ0IOC0.TQ0OEm bits. The TOQBm pin output is controlled by the TQ0IOC3.TQ0OLBm and TQ0IOC3.TQ0OEBm bits.

The timer output with each setting in the 6-phase PWM output mode is shown below.

Figure 12-3. Output Control of TOQTm and TOQBm Pins



Remark m = 1 to 3

Table 12-1. TOQTm Pin Output

TQ0OLm Bit	TQ0OEm Bit	TQ0CE Bit	TOQTm Pin Output
0	0	x	Low-level output
	1	0	Low-level output
		1	TOQTm positive-phase output
1	0	x	High-level output
	1	0	High-level output
		1	TOQTm negative-phase output

Remark m = 1 to 3

Table 12-2. TOQBm Pin Output

TQ0OLBm Bit	TQ0OEBm Bit	TQ0CE Bit	TOQBm Pin Output
0	0	x	Low-level output
	1	0	Low-level output
		1	TOQBm positive-phase output
1	0	x	High-level output
	1	0	High-level output
		1	TOQBm negative-phase output

Remark m = 1 to 3

(5) High-impedance output control registers 00, 01 (HZA0CTLn)

The HZA0CTLn 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 HZA0DCF1 bit is a read-only bit and cannot be written.

16-bit access is not possible.

Reset input clears these registers to 00H.

The same value can be always rewritten to the HZA0CTLn 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 TOP01/P01 is output	$\overline{\text{INTP000}}$	HZA0CTL0
When TOQB1/TOQ0/P10 is output When TOQT1/TOQ1/P11 is output When TOQT2/TOQ2/P12 is output When TOQT3/TOQ3/P13 is output When TOQB2/P14 is output When TOQB3/P15 is output		HZA0CTL1

Caution High-impedance control is affected only when the target pin is set as an output pin (including when it is set in the output mode other than timer output mode).

(1/2)

After reset: 00H R/W Address: HZA0CTL0 FFFFF630H, HZA0CTL1 FFFFF631H

HZA0CTLn (n = 0, 1)

<7>	<6>	5	4	<3>	<2>	1	<0>
HZA0DCEn	HZA0DCMn	HZA0DCNn	HZA0DCPn	HZA0DCTn	HZA0DCCn	0	HZA0DCFn

HZA0DCEn	High-impedance output control
0	Disable high-impedance output control operation. Pins can function as output pins.
1	Enable high-impedance output control operation.

HZA0DCMn	Condition of clearing high-impedance state by HZA0DCCn bit
0	Setting of the HZA0DCCn bit is valid regardless of the $\overline{\text{INTP000}}$ pin input.
1	Setting of the HZA0DCCn bit is invalid while the $\overline{\text{INTP000}}$ pin holds a level detected as abnormal (active level).
Rewrite the HZA0DCMn bit when the HZA0DCEn bit = 0.	

HZA0DCNn	HZA0DCPn	$\overline{\text{INTP000}}$ pin input edge specification
0	0	No valid edge (setting the HZA0DCFn bit by $\overline{\text{INTP000}}$ pin input is prohibited).
0	1	Rising edge of the $\overline{\text{INTP000}}$ pin is valid (abnormality is detected by rising edge input).
1	0	Falling edge of the $\overline{\text{INTP000}}$ pin is valid (abnormality is detected by falling edge input).
1	1	Setting prohibited

- Rewrite the HZA0DCNn and HZA0DCPn bits when the HZA0DCEn bit is 0.
- High-impedance output control is performed when the valid edge is input after the operation is enabled (by setting HZA0DCEn bit to 1). If the $\overline{\text{INTP000}}$ pin is at the active level when the operation is enabled, therefore, high-impedance output control is not performed.

HZA0DCTn	High-impedance output trigger bit
0	No operation
1	Pins are made to go into a high-impedance state by software and the HZA0DCFn bit is set to 1.

- If an edge indicating abnormality is input to the $\overline{\text{INTP000}}$ pin (which is detected according to the setting of the HZA0DCNn and HZA0DCPn bits), the HZA0DCTn bit is invalid even if it is set to 1.
- The HZA0DCTn bit is always 0 when it is read because it is a software-triggered bit.
- The HZA0DCTn bit is invalid even if it is set to 1 when the HZA0DCEn bit = 0.
- Simultaneously setting the HZA0DCTn and HZA0DCCn bits to 1 is prohibited.

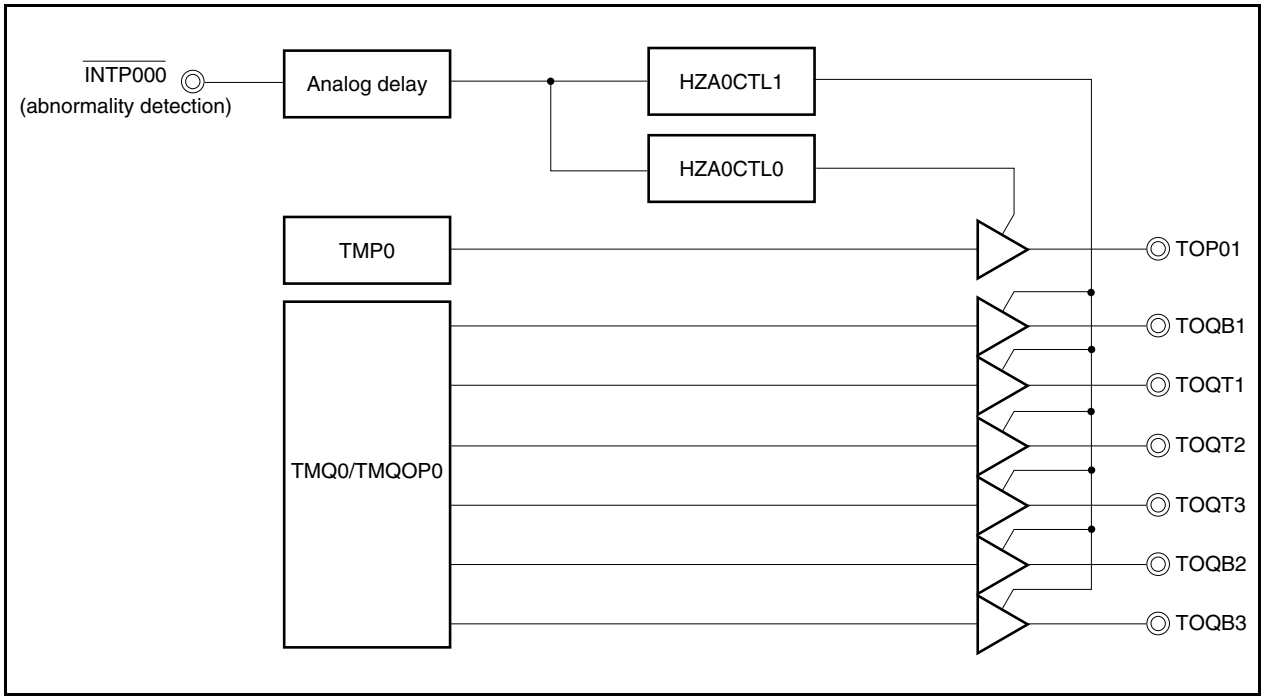
(2/2)

HZA0DCCn	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 HZA0DCFn bit is cleared to 0.

- Pins can function as output pins when the HZA0DCM bit = 0, regardless of the status of the $\overline{\text{INTP000}}$ pin.
- If an edge indicating abnormality is input to the $\overline{\text{INTP000}}$ pin (which is set by the HZA0DCNn and HZA0DCPn bits) when the HZA0DCM bit = 1, the HZA0DCCn bit is invalid even if it is set to 1.
- The HZA0DCCn bit is always 0 when it is read.
- The HZA0DCCn bit is invalid even if it is set to 1 when the HZA0DCEn bit = 0.
- Simultaneously setting the HZA0DCTn and HZA0DCCn bits to 1 is prohibited.

HZA0DCFn	High-impedance output status flag
Clear (0)	Indicates that output of the pin is enabled. • This bit is cleared to 0 when the HZA0DCEn bit = 0. • This bit is cleared to 0 when the HZA0DCCn bit = 1.
Set (1)	Indicates that the pin goes into a high-impedance state. • This bit is set to 1 when the HZA0DCTn bit = 1. • This bit is set to 1 when an edge indicating abnormality is input to the $\overline{\text{INTP000}}$ pin (which is detected according to the setting of the HZA0DCNn and HZA0DCPn bits).

Figure 12-4. High-Impedance Output Controller Configuration



(a) Setting procedure

(i) Setting of high-impedance control operation

<1> Set the HZA0DCMn, HZA0DCNn, and HZA0DCPn bits.

<2> Set the HZA0DCEn bit to 1 (enable high-impedance control).

(ii) Changing setting after enabling high-impedance control operation

<1> Clear the HZA0DCEn bit to 0 (to stop the high-impedance control operation).

<2> Change the setting of the HZA0DCMn, HZA0DCNn, and HZA0DCPn bits.

<3> Set the HZA0DCEn bit to 1 (to enable the high-impedance control operation again).

(iii) Resuming output when pins are in high-impedance state

If the HZA0DCMn bit is 1, set the HZA0DCCn bit to 1 to clear the high-impedance state after the valid edge of the $\overline{\text{INTP000}}$ pin is detected. However, the high-impedance state cannot be cleared unless this bit is set while the input level of the $\overline{\text{INTP000}}$ pin is inactive.

<1> Set the HZA0DCCn bit to 1 (command signal to clear the high-impedance state).

<2> Read the HZA0DCFn bit and check the flag status.

<3> Return to <1> if the HZA0DCFn bit is 1. The input level of the $\overline{\text{INTP000}}$ pin must be checked. The pin can function as an output pin if the HZA0DCFn bit is 0.

(iv) To make the pin to go into a high-impedance state by software

The HZA0DCTn 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 $\overline{\text{INTP000}}$ pin is inactive. The following procedure is an example in which the setting is not dependent upon the setting of the HZA0DCMn bit.

<1> Set the HZA0DCTn bit to 1 (high-impedance output command).

<2> Read the HZA0DCFn bit to check the flag status.

<3> Return to <1> if the HZA0DCFn bit is 0. The input level of the $\overline{\text{INTP000}}$ pin must be checked. The pin is in a high-impedance state if the HZA0DCFn bit is 1.

However, if the $\overline{\text{INTP000}}$ pin is not used with the HZA0DCP1 bit and HZA0DCNn bit cleared to 0, the pin goes into a high-impedance state when the HZA0DCTn bit is set to 1.

12.4 Operation

12.4.1 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 TMQ0 and the TMQ0 option in combination.

The 6-phase PWM output mode is enabled by setting the TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 bits of TMQ0 to “111”.

One 16-bit counter and four 16-bit compare registers of TMQ0 are used to generate a basic 3-phase wave.

The functions of the compare registers are as follows.

TMP2 can perform a tuning operation with TMQ0 to start a conversion trigger source for the A/D converter.

Compare Register	Function	Settable Range
TQ0CCR0 register	Setting of cycle	0002H ≤ m ≤ FFFEH
TQ0CCR1 register	Specifying output width of phase U	0000H ≤ i ≤ m + 1
TQ0CCR2 register	Specifying output width of phase V	0000H ≤ j ≤ m + 1
TQ0CCR3 register	Specifying output width of phase W	0000H ≤ k ≤ m + 1

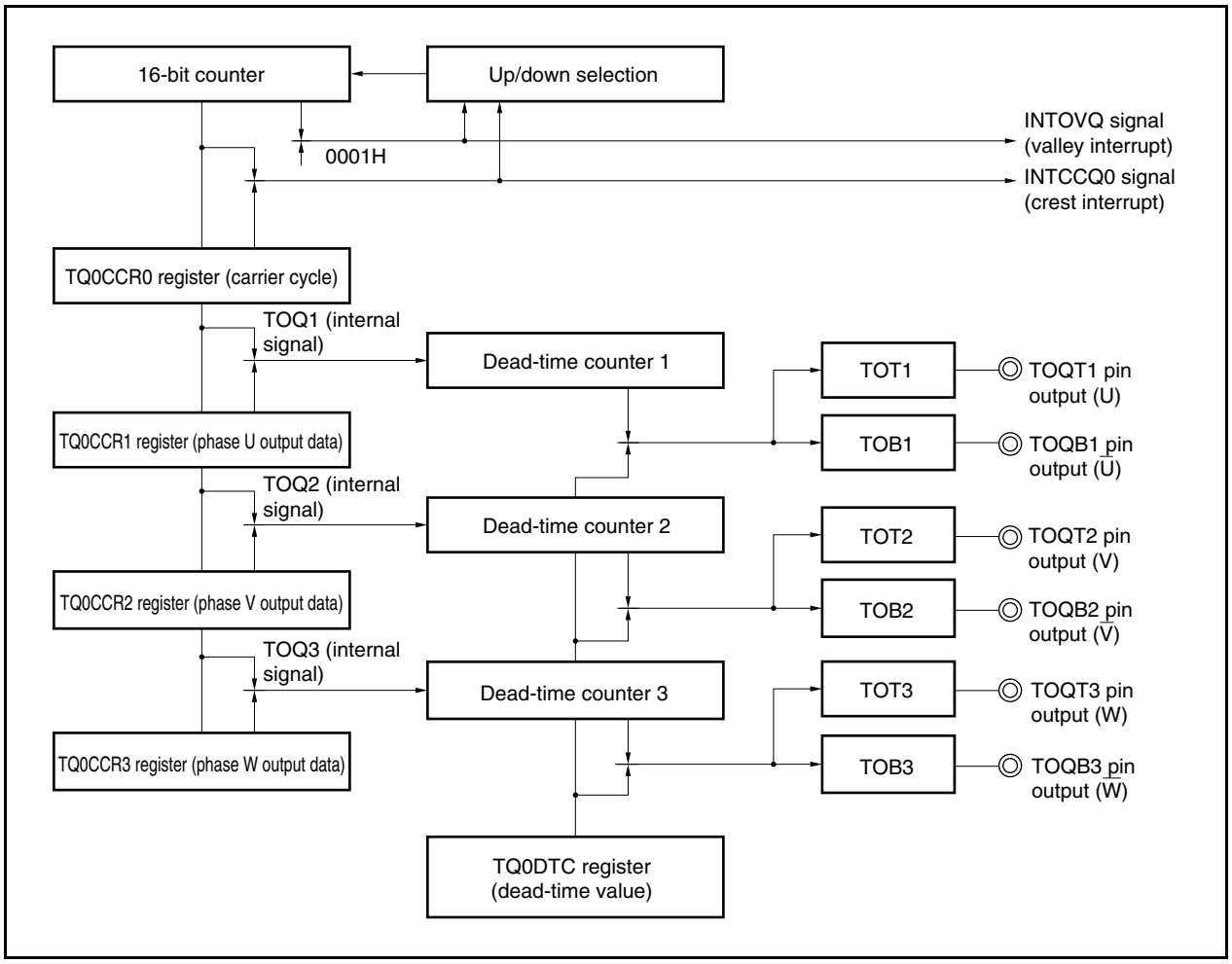
Remark m = Set value of TQ0CCR0 register
i = Set value of TQ0CCR1 register
j = Set value of TQ0CCR2 register
k = Set value of TQ0CCR3 register

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.

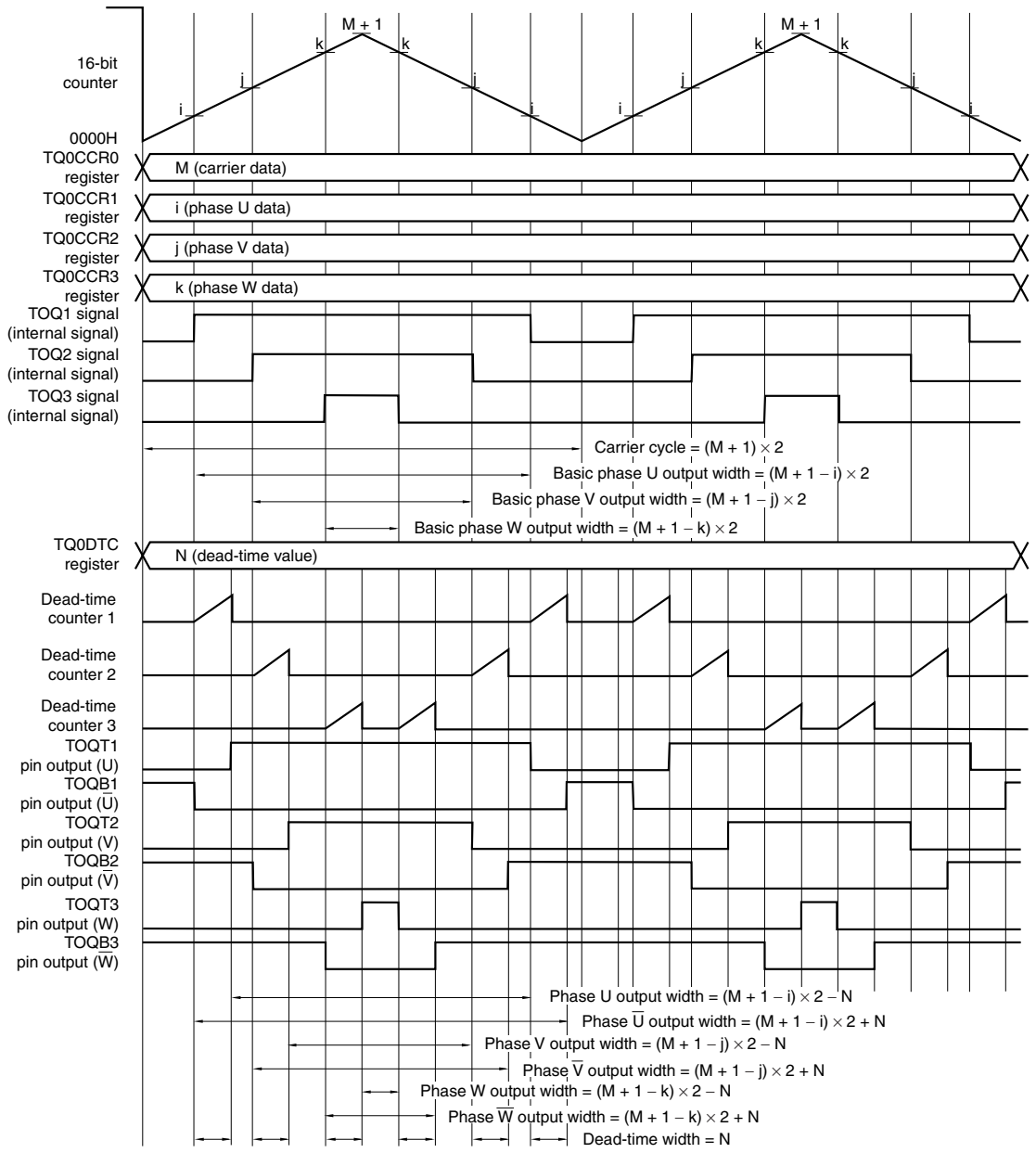
The 16-bit counter for generating the basic 3-phase wave counts up or down. After the operation has been started, this counter counts up. When its count value matches the cycle set to the TQ0CCR0 register, the counter starts counting down. When the count value matches 0001H, the counter counts up again. This means that a value two times higher than the value set to the TQ0CCR0 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 TMQ0 dead-time compare register (TQ0DTC) 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 (TQ0DTC), the same dead-time value is used in the three phases.

<R> **Figure 12-5. Outline of 6-Phase PWM Output Mode**



<R> **Figure 12-6. Timing Chart of 6-Phase PWM Output Mode**



Cautions 1. Set the value "M" of the TQ0CCR0 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 TQ0CCR1, TQ0CCR2, and TQ0CCR3 registers.

3. The output is 100% if "0000H" is set to the TQ0CCR1, TQ0CCR2, and TQ0CCR3 registers. The output is 0% if "M + 1" is set to the TQ0CCR1, TQ0CCR2, and TQ0CCR3 registers.

<R> **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 TQ0CCR1, TQ0CCR2, and TQ0CCR3 registers.**

<R> **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 INTCCQ0 (crest interrupt) signal and INTOVQ (valley interrupt) signal.

The INTCCQ0 and INTOVQ signals can be culled by using the TQ0OPT1 register.

For details of culling interrupts, see **12.4.3 Interrupt culling function**.

- INTCCQ0 (crest interrupt) signal: Interrupt signal indicating match between the value of the 16-bit counter that counts up and the value of the TQ0CCR0 register
- INTOVQ (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 P2	TMP2 capture/compare register 0 (TP2CCR0) TMP2 capture/compare register 1 (TP2CCR1)
Timer Q0	TMQ0 capture/compare register 0 (TQ0CCR0) TMQ0 capture/compare register 1 (TQ0CCR1) TMQ0 capture/compare register 2 (TQ0CCR2) TMQ0 capture/compare register 3 (TQ0CCR3)
Timer Q0 option	TMQ0 option register 1 (TQ0OPT1)

For details of the transfer function of the compare register, see **12.4.4 Operation to rewrite register with transfer function**.

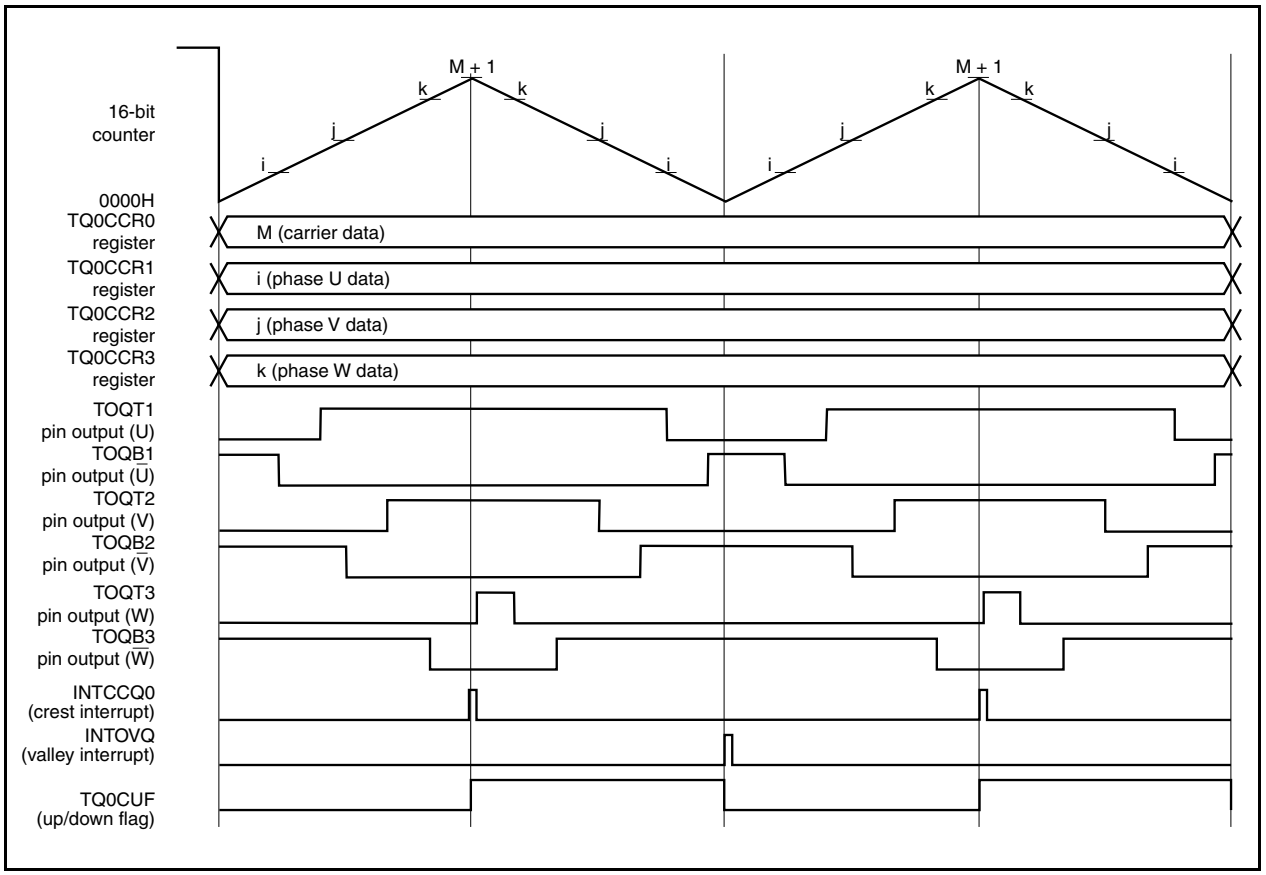
(4) Counting-up/down operation of 16-bit counter

The operation status of the 16-bit counter can be checked by using the TQ0CUF bit of TMQ0 option register 0 (TQ0OPT0).

Status of TQ0CUF Bit	Status of 16-Bit Counter	Range of 16-Bit Counter Value
TQ0CUF bit = 0	Counting up	0000H – m
TQ0CUF bit = 1	Counting down	(m + 1) – 0001H

Remark m = Set value of TQ0CCR0 register

<R> **Figure 12-7. Interrupt and Up/Down Flag**



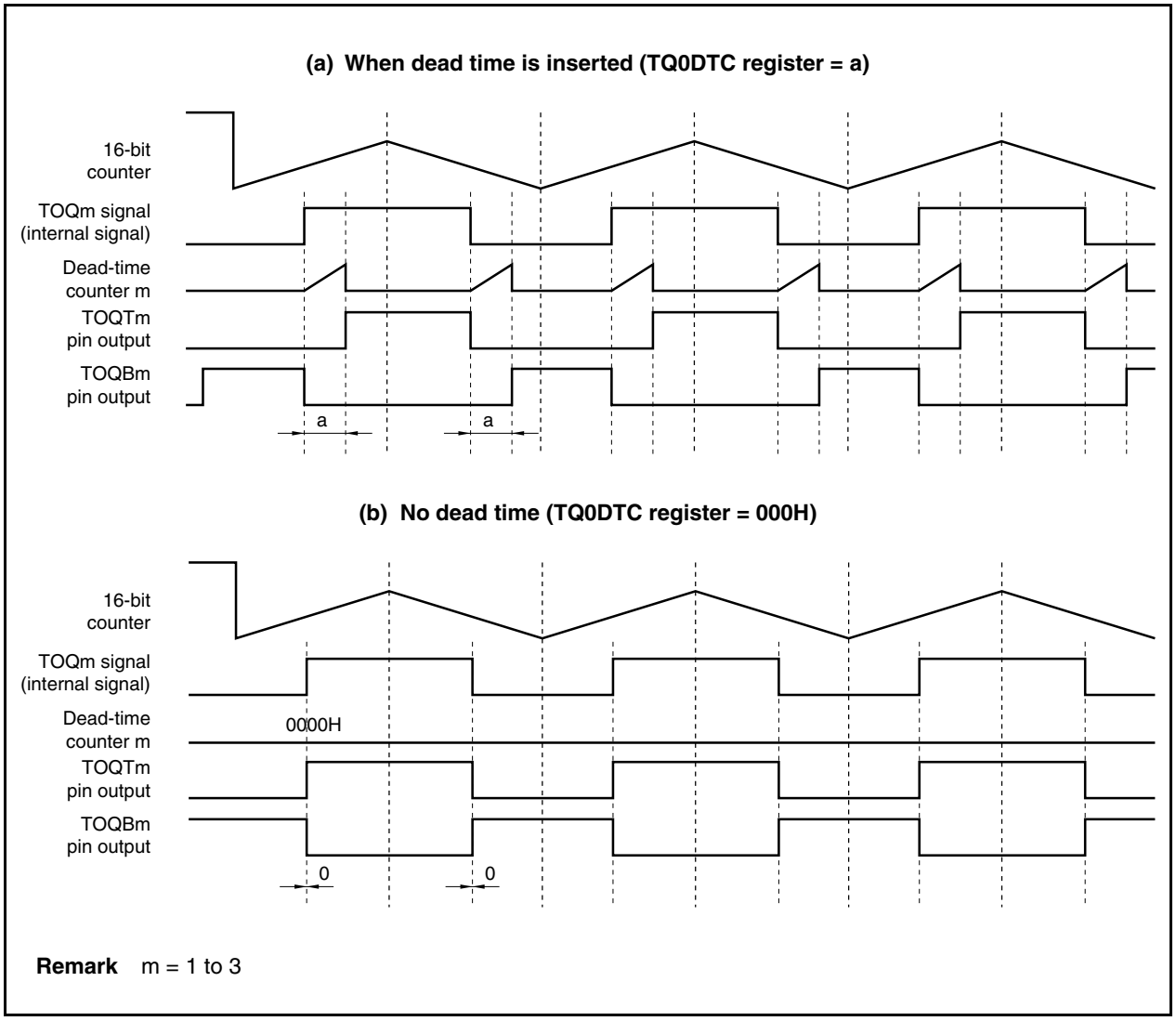
12.4.2 Dead-time control (generation of negative-phase wave signal)

(1) Dead-time control mechanism

In the 6-phase PWM output mode, compare registers 1 to 3 (TQ0CCR1, TQ0CCR2, and TQ0CCR3) are used to set the duty factor, and compare register 0 (TQ0CCR0) 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 (TMQOP0) and their inverted signal with dead-time is created to generate three sets of (six) PWM waves.

The TMQOP0 unit consists of three 10-bit counters (dead-time counters 1 to 3) that operate in synchronization with the count clock of TMQ0, and a TMQ0 dead-time compare register (TQ0DTC) that specifies dead time. If "a" is set to the TQ0DTC register, the dead-time value is "a", and interval "a" is created between a positive-phase wave and a negative-phase wave.

Figure 12-8. PWM Output Wave with Dead Time (1)



Remark m = 1 to 3

(2) PWM output of 0%/100%

The V850E/MA3 is capable of 0% wave output and 100% wave output for PWM output.

A low level is continuously output from TOQTm pin as the 0% wave output. A high level is continuously output from TOQTm pin as the 100% wave output.

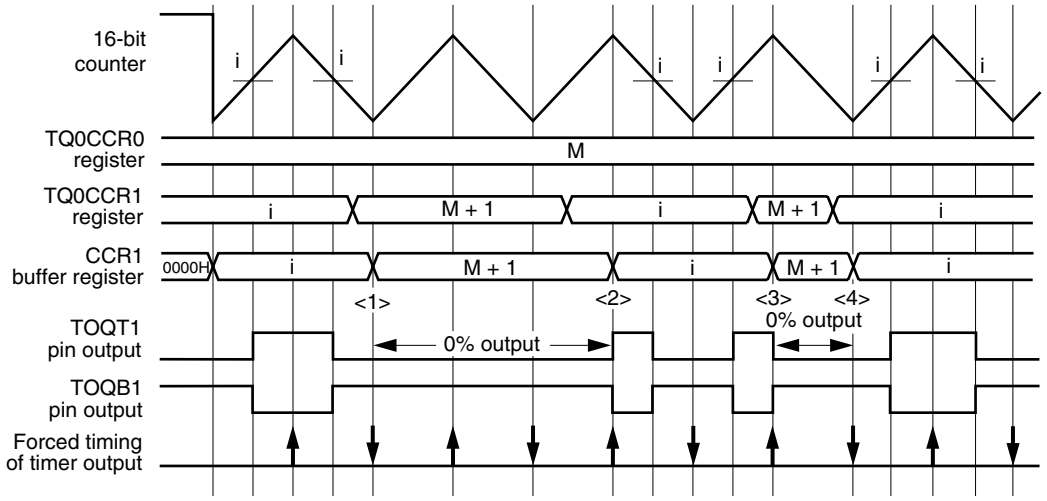
The 0% wave is output by setting the TQ0CCRm register to “M + 1” when the TQ0CCR0 register = M.

The 100% wave is output by setting the TQ0CCRm register to “0000H”.

Rewriting the TQ0CCRm 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 (INTCCQ0) and valley interrupt (INTOVQ).

Remark m = 1 to 3

Figure 12-9. 0% PWM Output Waveform (Without 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.

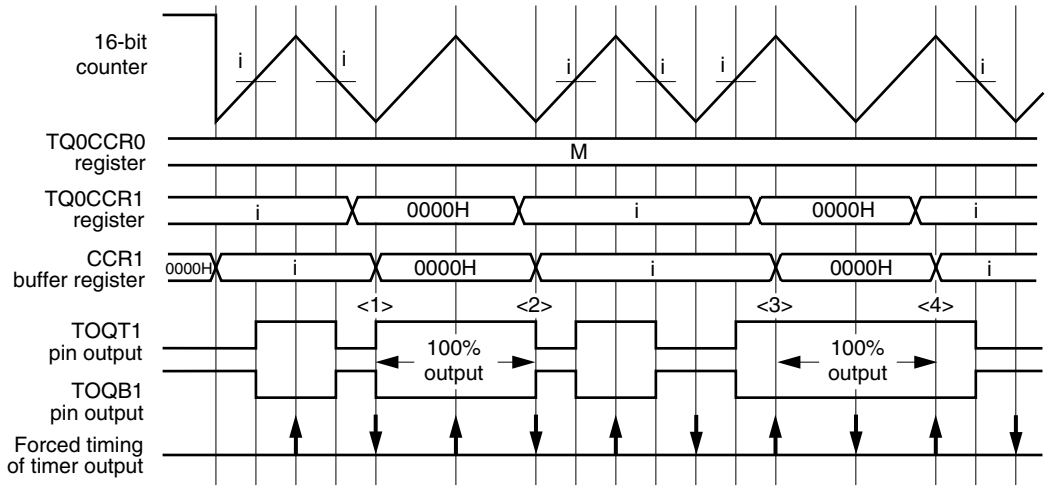
<2> 0% output is canceled by the crest interrupt (without a match with the 16-bit counter).
The crest interrupt forcibly raises the timer output. This cancels the 0% output.

<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 TQ0CCRm register matches the value of the 16-bit counter. As a result, the 0% wave is output.

<4> 0% output is canceled by the valley interrupt (without a match with the 16-bit counter).
The valley interrupt forcibly lowers the timer output. This cancels the 0% output.

Remark ↑ means forced raising and ↓ means forced lowering.

Figure 12-10. 100% PWM Output Waveform (Without 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 TQ0CCRm register matches the value of the 16-bit counter. As a result, the 100% output is produced.

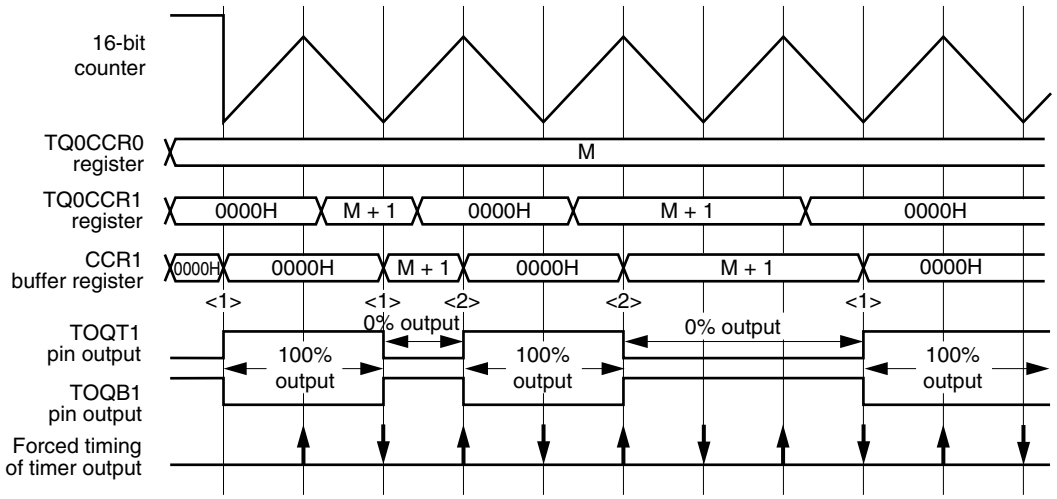
<2> 100% output is canceled by the valley interrupt (without a match with the 16-bit counter).
The valley interrupt forcibly lowers the timer output. This cancels the 100% output.

<3> 100% output is selected by the crest interrupt (without a match with the 16-bit counter).
The crest interrupt forcibly raises the timer output. This produces the 100% output.

<4> 100% output is canceled by the crest interrupt (without a match with the 16-bit counter).
The crest interrupt forcibly raises the timer output. This cancels the 100% output.

Remark ↑ means forced raising and ↓ means forced lowering.

Figure 12-11. PWM Output Waveform from 0% to 100% and from 100% to 0% (Without 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 TQ0CCR0 register value.

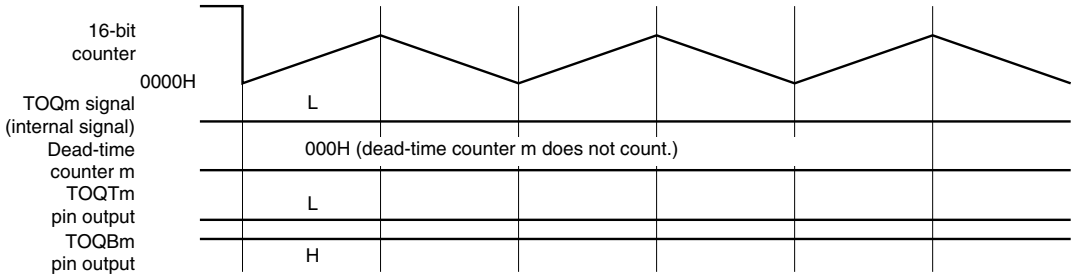
(3) Output waveform in vicinity of 0% and 100% output

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.

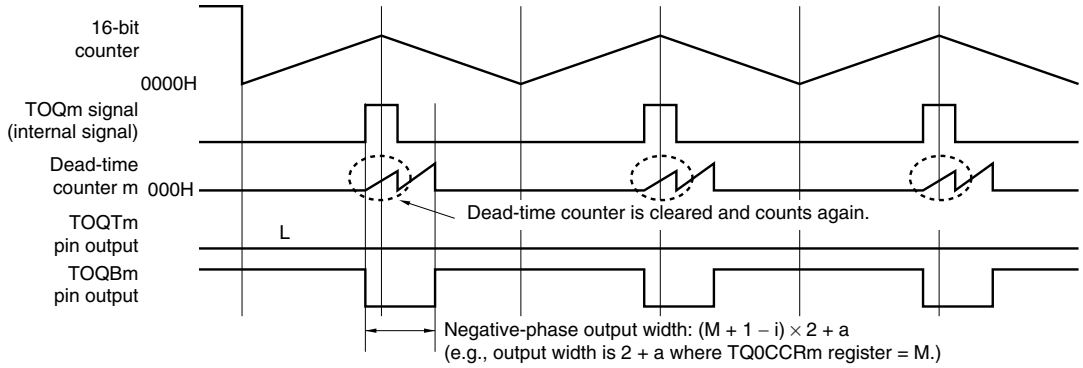
The output waveform of dead-time control in the vicinity of 0% and 100% output is shown below.

Figure 12-12. PWM Output Waveform with Dead Time (2)

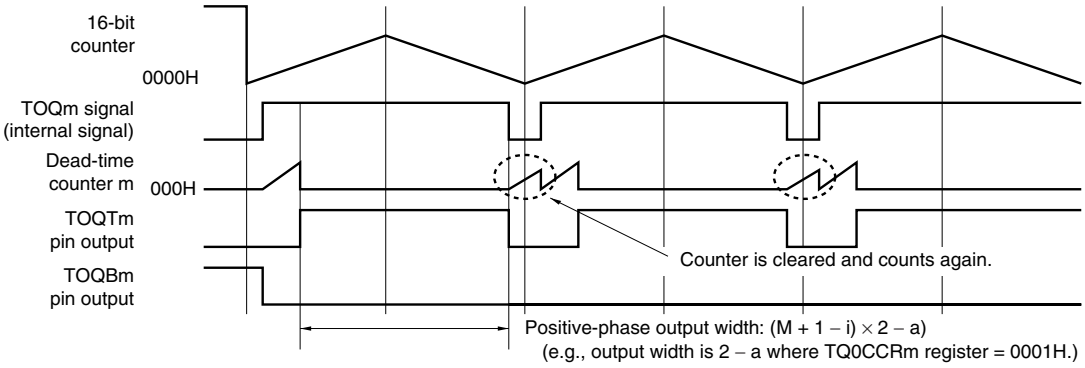
(a) 0% output (TQ0CCRm register = M + 1, TQ0CCR0 register = M, TQ0DTC register = a)



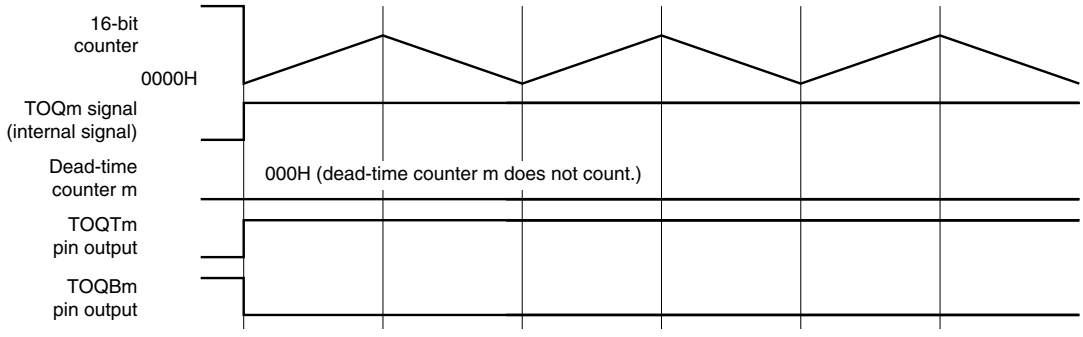
(b) In vicinity of 0% output (TQ0CCRm register = i ≥ M + 1 − a/2, TQ0CCR0 register = M, TQ0DTC register = a)



(c) In vicinity of 100% output (TQ0CCRm register = i ≤ a/2, TQ0CCR0 register = M, TQ0DTC register = a)



(d) 100% output (TQ0CCRm register = 0000H, TQ0CCR0 register = M, TQ0DTC register = a)



Remark m = 1 to 3

(4) Automatic dead-time width narrowing function (TQ0OPT2.TQ0DTM bit = 1)

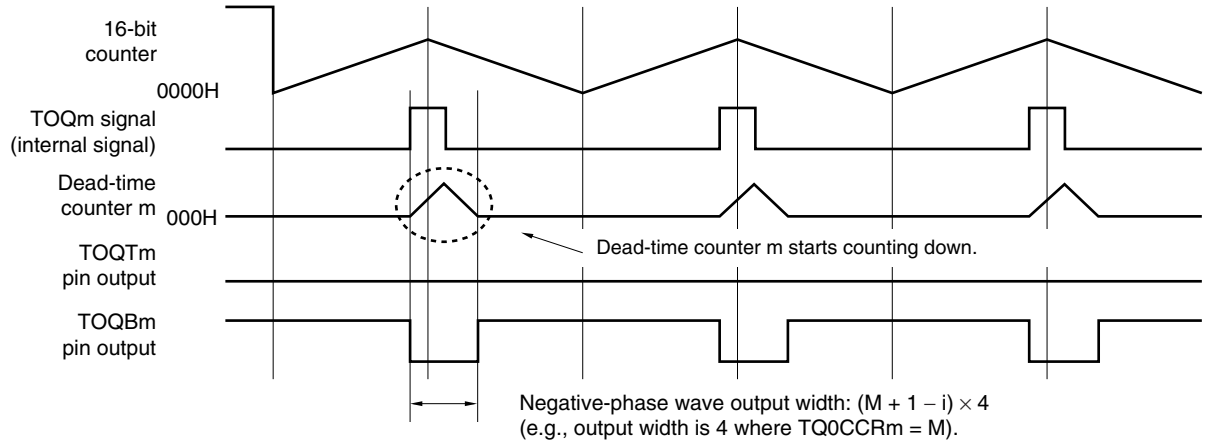
The dead-time width can be automatically narrowed in the vicinity of 0% output or 100% output by setting the TQ0OPT2.TQ0DTM bit to 1.

By setting the TQ0DTM bit to 1, the dead-time counter is not cleared, but starts down counting if the TOQm (internal signal) output of timer Q changes during dead-time counting.

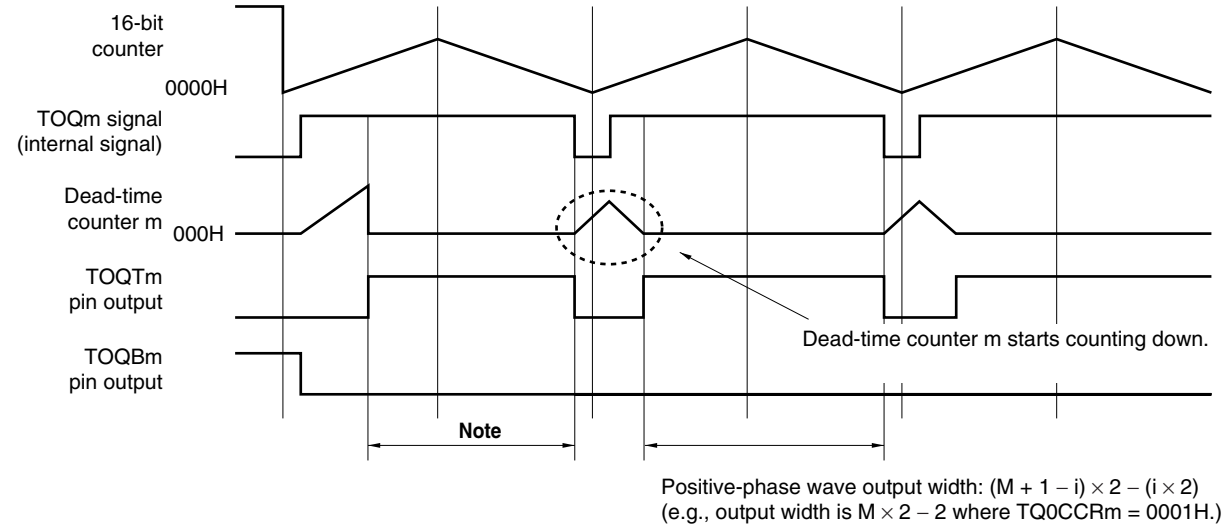
The following timing chart shows the operation of the dead-time counter when the TQ0DTM bit is set to 1.

Figure 12-13. Operation of Dead-Time Counter m (1)

(a) In vicinity of 0% output (TQ0CCRm register = i ≥ M + 1 − a/2, TQ0CCR0 register = M, TQ0DTC register = a)



(b) In vicinity of 100% output (TQ0CCRm register = i ≤ a/2, TQ0CCR0 register = M, TQ0DTC register = a)



Note The output width of the first wave differs from that of the second and subsequent waves immediately after the TQ0CTL0.TQ0CE bit has been set. The first wave is shorter than the second wave because the dead time is fully counted.

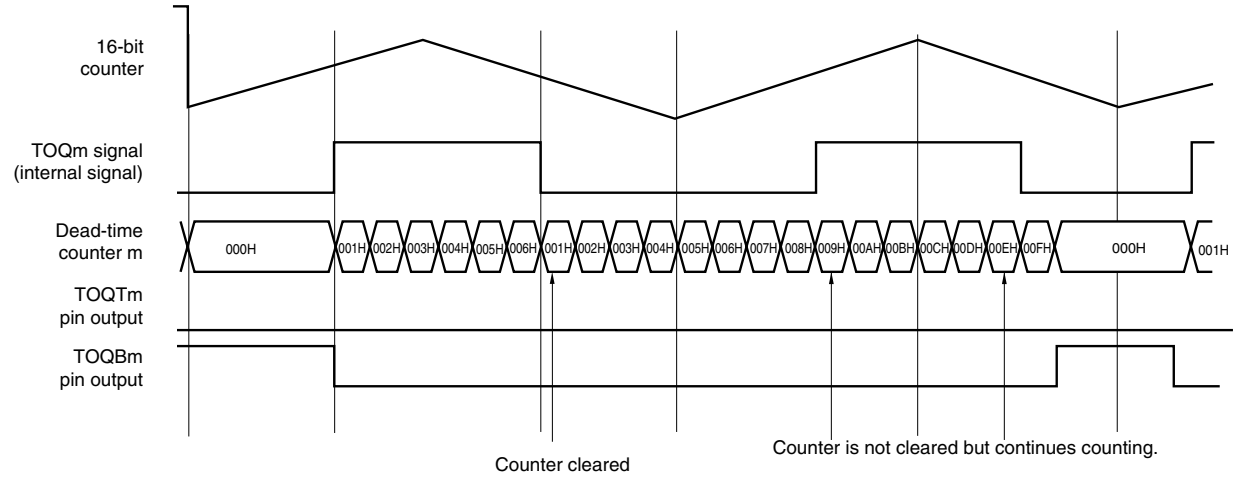
Remark m = 1 to 3

(5) Dead-time control in case of incorrect setting

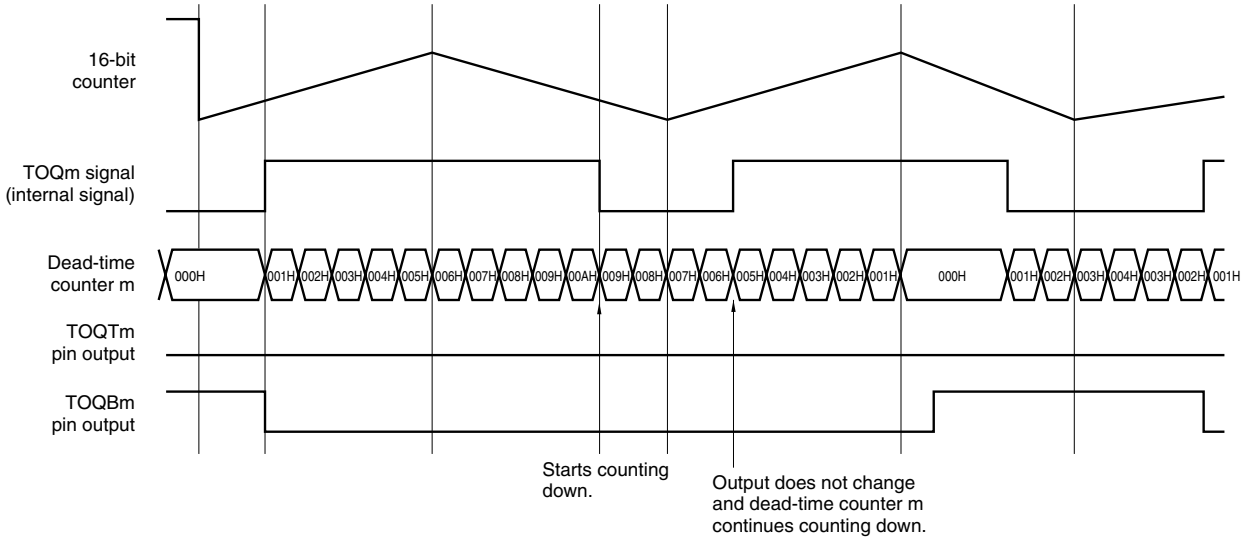
Usually, the TOQm (internal signal) output of TMQ0 changes only once during dead-time counting, only in the vicinity of 0% and 100% output. This section shows an example where the TQ0CCR0 register (carrier cycle) and TQ0DTC register (dead-time value) are incorrectly set. If these registers are incorrectly set, the TOQm (internal signal) output of TMQ0 changes more than once during dead-time counting. The following flowchart shows the 6-phase PWM output wave in this case.

Figure 12-14. Operation of Dead-Time Counter m (2)

(a) When TQ0OPT2.TQ0DTM bit = 0, TQ0CCR0 register = 0006H, TQ0DTC register = 000FH, TQ0CCRm register = 0004H



(b) When TQ0OPT2.TQ0DTM bit = 1, TQ0CCR0 register = 0006H, TQ0DTC register = 000FH, TQ0CCRm register = 0002H



Remark m = 1 to 3

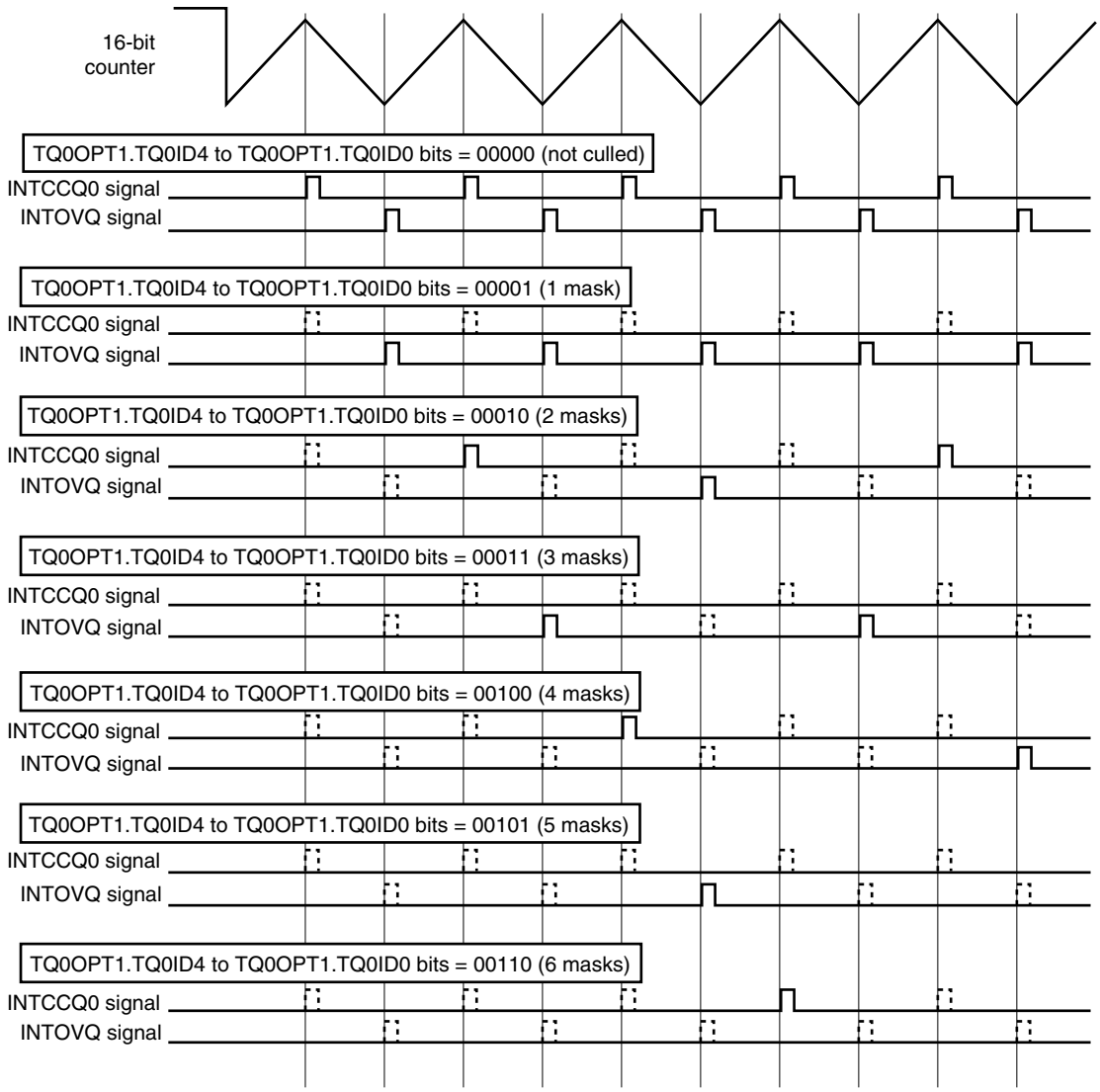
12.4.3 Interrupt culling function

- The interrupts to be culled are INTCCQ0 (crest interrupt) and INTOVQ (valley interrupt).
- The TQ0OPT1.TQ0ICE bit is used to enable output of the INTCCQ0 interrupt and the number of times the interrupt is to be culled.
- The TQ0OPT1.TQ0IOE bit is used to enable output of the INTOVQ interrupt and the number of times the interrupt is to be culled.
- <R> The TQ0OPT1.TQ0ID4 to TQ0OPT1.TQ0ID0 bits are used to specify the number of counts by which a specified interrupt is to be culled.
  The interrupt is culled for the duration of the specified number of counts and is generated at the next interrupt timing.
- The TQ0RDE bit of TQ0OPT2 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 TQ0CCR1 register has been written.
- The TQ0OPT0.TQ0CMS 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 TQ0CMS bit is 0. When the TQ0CMS 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 TQ0CCRm register to the CCRm buffer register in synchronization with interrupt culling timing.

Caution 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).

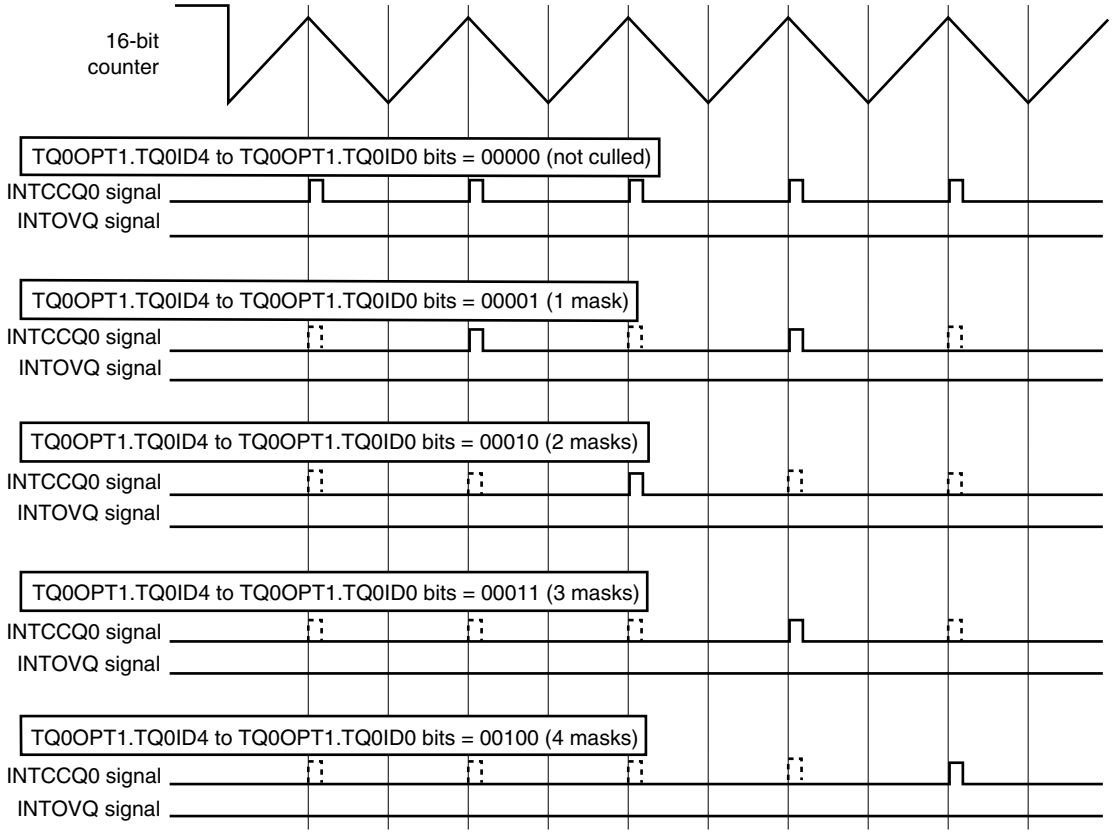
(1) Interrupt culling operation

Figure 12-15. Interrupt Culling Operation When TQ0OPT1.TQ0ICE Bit = 1, TQ0OPT1.TQ0IOE Bit = 1, TQ0OPT2.TQ0RDE Bit = 1 (Crest/Valley Interrupt Output)



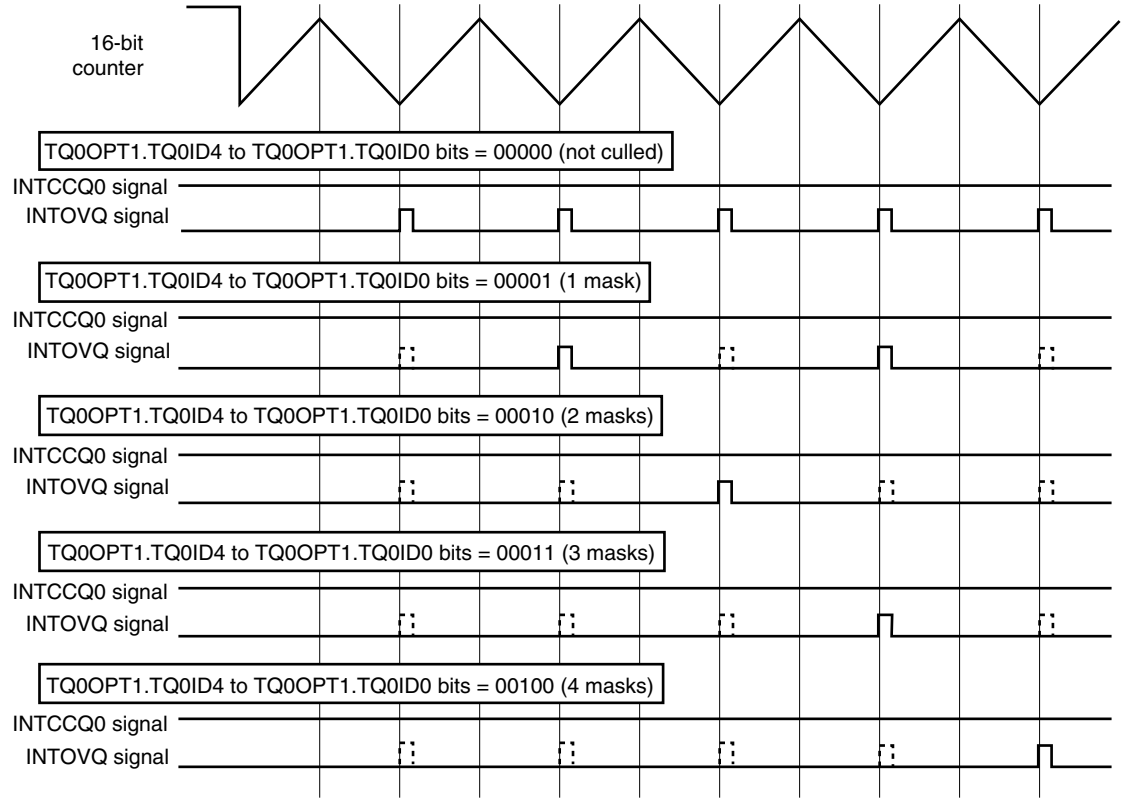
Remark : Culled interrupt

Figure 12-16. Interrupt Culling Operation When TQ0OPT1.TQ0ICE Bit = 1, TQ0OPT1.TQ0IOE Bit = 0, TQ0OPT2.TQ0RDE Bit = 1 (Crest Interrupt Output)



Remark : Culled interrupt

Figure 12-17. Interrupt Culling Operation When TQ0OPT1.TQ0ICE Bit = 0, TQ0OPT1.TQ0IOE Bit = 1, TQ0OPT2.TQ0RDE Bit = 1 (Valley Interrupt Output)



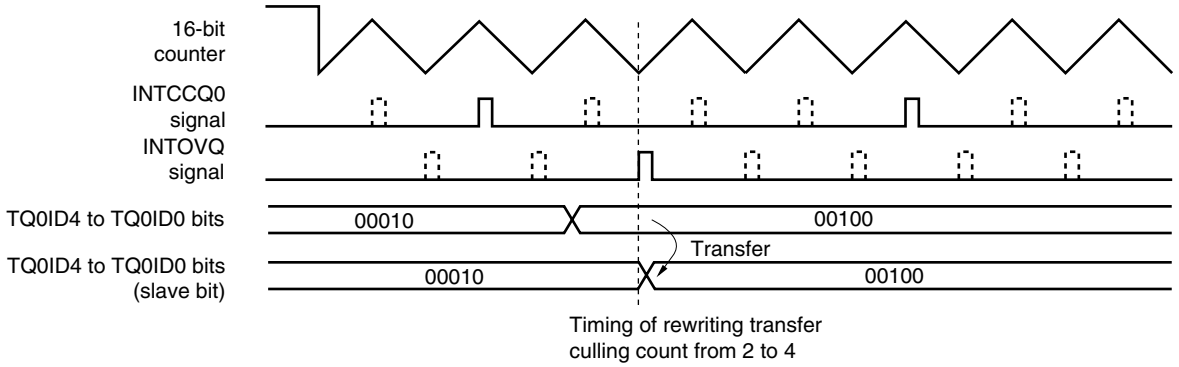
Remark : Culled interrupt

(2) To alternately output crest interrupt (INTCCQ0) and valley interrupt (INTOVQ)

To alternately output the crest and valley interrupts, set both the TQ0OPT1.TQ0ICE and TQ0OPT1.TQ0IOE bits to 1.

Figure 12-18. Crest/Valley Interrupt Output

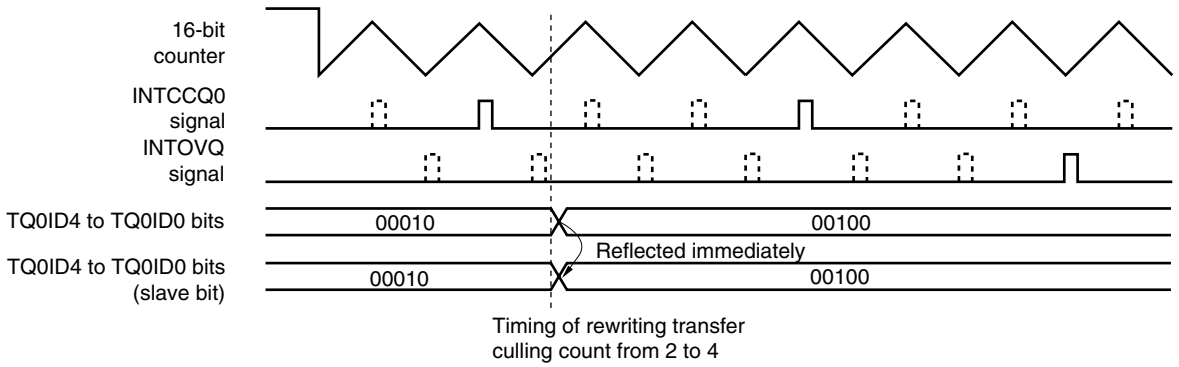
(a) TQ0OPT0.TQ0CMS bit = 0, TQ0OPT2.TQ0RDE bit = 1 (with transfer culling control)



Remarks 1. Transfer is performed at the culled interrupt output timing. The other transfer timing is ignored.

2. : Culled interrupt

(b) TQ0CMS bit = 1, TQ0RDE bit = 0 or 1 (without transfer control)



Remarks 1. Rewriting is reflected immediately. The transfer timing is ignored.

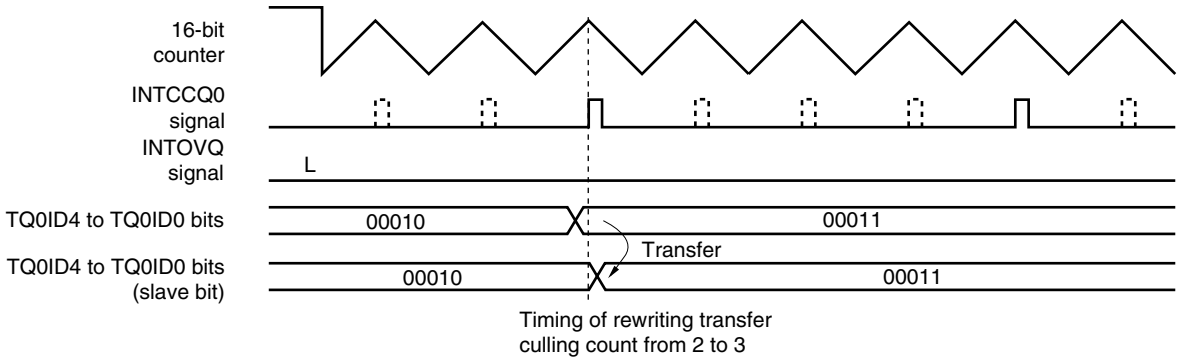
2. : Culled interrupt

(3) To output only crest interrupt (INTCCQ0)

Set the TQ0OPT1.TQ0ICE bit to 1 and clear the TQ0OPT1.TQ0IOE bit to 0.

Figure 12-19. Crest Interrupt Output

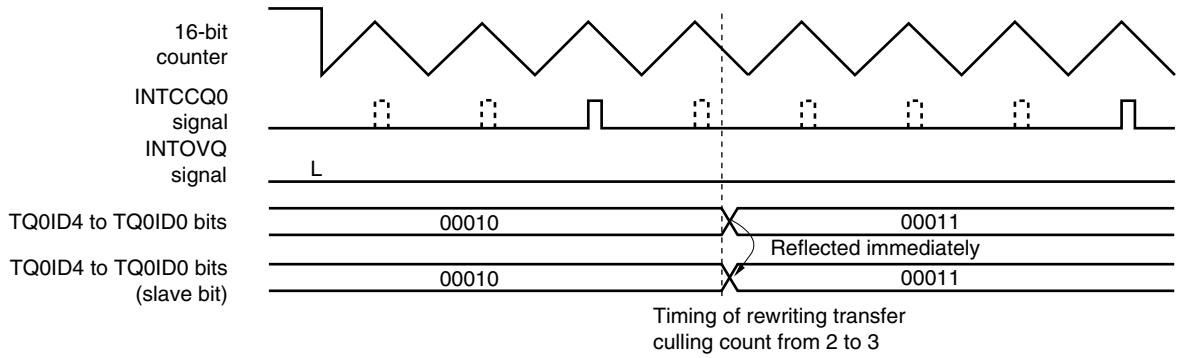
(a) TQ0OPT0.TQ0CMS bit = 0, TQ0OPT2.TQ0RDE bit = 1 (with transfer culling control)



Remarks 1. Transfer is performed at the culled interrupt output timing. The other transfer timing is ignored.

2. : Culled interrupt

(b) TQ0CMS bit = 1, TQ0RDE bit = 0 or 1 (without transfer control)



Remarks 1. Rewriting is reflected immediately. The transfer timing is ignored.

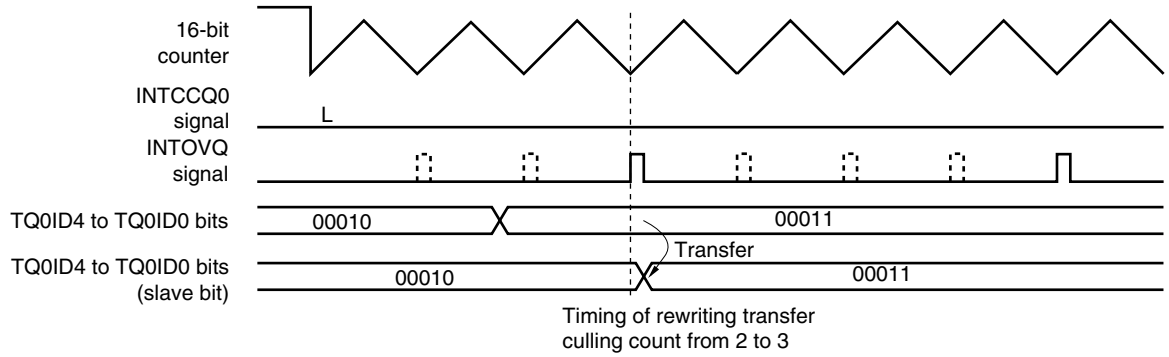
2. : Culled interrupt

(4) To output only valley interrupt (INTOVQ)

Clear the TQ0OPT1.TQ0ICE bit to 0 and set the TQ0IOE bit to 1.

Figure 12-20. Valley Interrupt Output

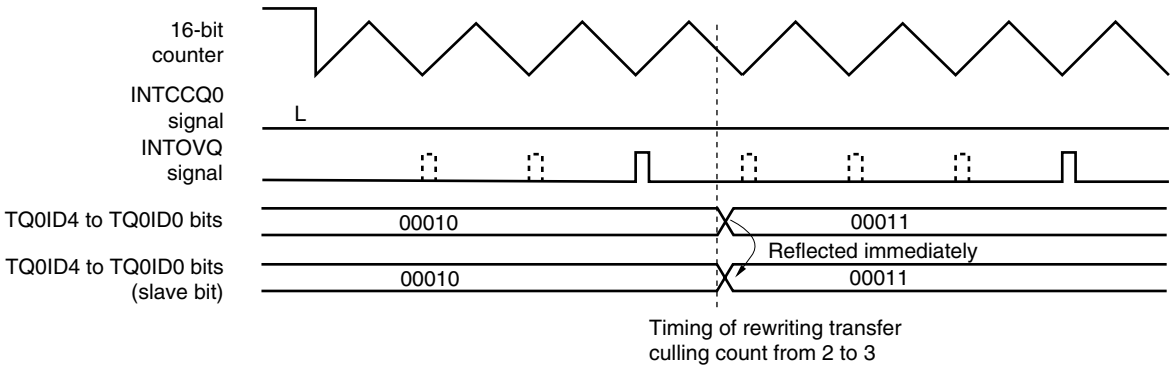
(a) TQ0OPT0.TQ0CMS bit = 0, TQ0OPT2.TQ0RDE bit = 1 (with transfer culling control)



Remarks 1. Transfer is performed at the culled interrupt output timing. The other transfer timing is ignored.

2. : Culled interrupt

(b) TQ0CMS bit = 1, TQ0RDE bit = 0 or 1 (without transfer control)



Remarks 1. Rewriting is reflected immediately. The transfer timing is ignored.

2. : Culled interrupt

12.4.4 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.

- TQ0CCR0: Register that specifies the cycle of the 16-bit counter (TMQ)
- TQ0CCR1: Register that specifies the duty factor of TOQT1 (U) and TOQB1 ($\overline{U}$)
- TQ0CCR2: Register that specifies the duty factor of TOQT2 (V) and TOQB2 ($\overline{V}$)
- TQ0CCR3: Register that specifies the duty factor of TOQT3 (W) and TOQB3 ($\overline{W}$)
- TQ0OPT1: Register that specifies the culling of interrupts
- TP2CCR0: Register that specifies the A/D conversion start trigger generation timing (TMP2 during tuning operation)
- TP2CCR1: Register that specifies the A/D conversion start trigger generation timing (TMP2 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 TQ0OPT0.TQ0CMS bit to 1. The setting of the TQ0OPT2.TQ0RDE 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 TQ0OPT0.TQ0CMS bit to 0, the TQ0OPT1.TQ0ID4 to TQ0OPT1.TQ0ID0 bits to 00000, and the TQ0OPT2.TQ0RDE bit to 0. When data is written to the TQ0CCR1 register, data in the seven registers are transferred to the buffer register all at once at the next transfer timing. Unless the TQ0CCR1 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 TQ0CCR0 register value) and valley (match between the 16-bit counter value and 0001H) regardless of the interrupt.

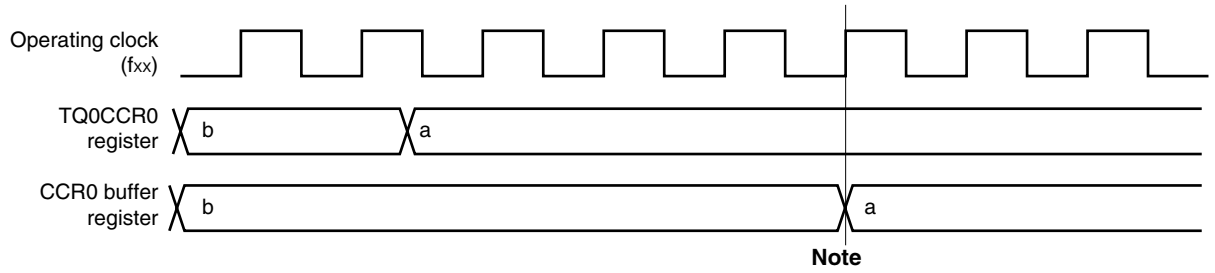
- Intermittent batch rewrite mode (transfer culling mode)
  This mode is set by clearing the TQ0OPT0.TQ0CMS bit to 0 and setting the TQ0OPT2.TQ0RDE bit to 1.
  When data is written to the TQ0CCR1 register, the seven registers are transferred to the buffer register all at once at the next transfer timing. Unless the TQ0CCR1 register is rewritten, the transfer operation is not performed even if the other six registers are rewritten.
  If interrupt culling is specified by the TQ0OPT1 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 TQ0CCR0 register value) or valley interrupt (match between the 16-bit counter value and 0001H).
  For details of the interrupt culling function, see **12.4.3 Interrupt culling function**.

(1) Anytime rewrite mode

This mode is set by setting the TQ0OPT0.TQ0CMS bit to 1. The setting of the TQ0OPT2.TQ0RDE 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 TQ0CCRm register, the rewritten value is not reflected because the next match is ignored after the first match has occurred. If the register is rewritten during up counting, the new register value becomes valid after the counter has started counting down.

Figure 12-21. Timing of Reflecting Rewritten Value

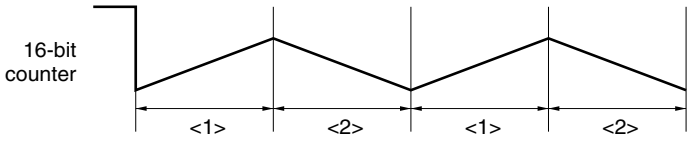


Note After the register (TQ0CCR0, TQ0CCR2, TQ0CCR3, TQ0OPT1, TP2CCR0, or TP2CCR1) has been written, the written value is transferred to an internal buffer register after four clocks of the operating clock. However, the value of only the TQ0CCR1 register is transferred after 5 more clocks.

(a) Rewriting TQ0CCR0 register

Even if the TQ0CCR0 register is rewritten in the anytime rewrite mode, the new value may not be reflected in some cases.

Figure 12-22. Example of Rewriting TQ0CCR0 Register



Rewriting during period <1> (rewriting during up counting)

If the newly rewritten value is greater than the value of the 16-bit counter, there is no problem because it will match the value of the 16-bit counter. If the new value is less than the value of the 16-bit counter, it will not match the value of the counter. As a result, the 16-bit counter overflows and continues counting up from 0000H until it matches the register value again, and the correct PWM waveform is not output.

Rewriting during period <2> (rewriting during down counting)

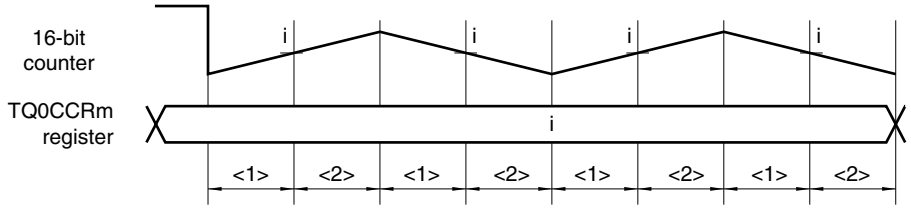
A match with the value of the 16-bit counter is ignored during counting down. Therefore, the rewritten period value is reflected starting from counting up in the next cycle as a match point.

(b) Rewriting TQ0CCRm register

Figure 12-24 shows the timing of rewriting before the value of the 16-bit counter matches the value of the TQ0CCRm register (<1> in Figure 12-23), and Figure 12-25 shows the timing of rewriting after the value of the 16-bit counter matches the value of the TQ0CCRm register (<2> in Figure 12-23).

Figure 12-23. Basic Operation of 16-Bit Counter and TQ0CCRm Register

(a) Basic figure



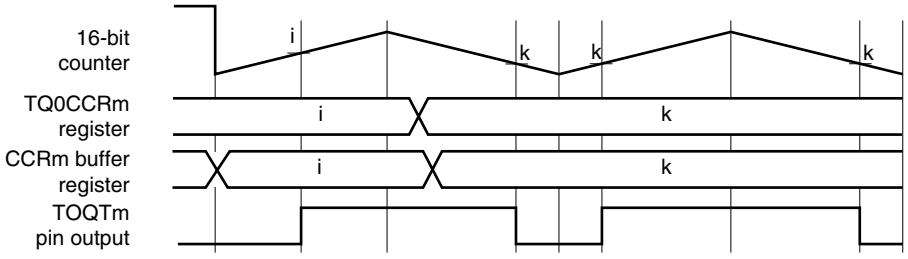
Remarks 1. i = Set value of TQ0CCRm register

2. m = 1 to 3

Figure 12-24. Example of Rewriting TQ0CCR1 to TQ0CCR3 Registers (Rewriting Before Match Occurs)

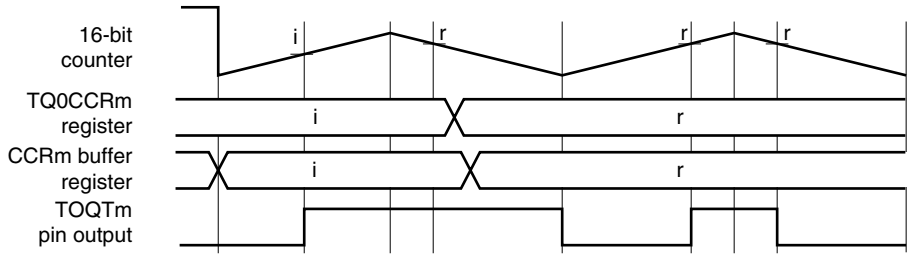
(a)

If the TQ0CCRm 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 TQ0CCRm register, the output waveform is as follows because the register value does not match the counter value.

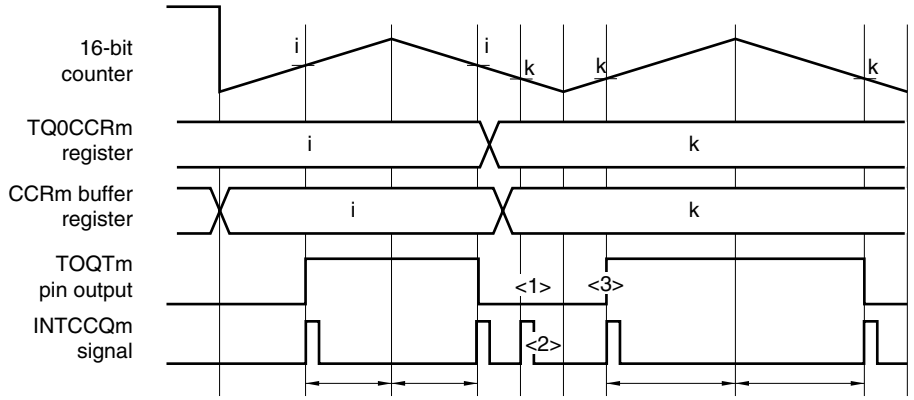


If the register value does not match the counter value, the TOQTm pin output does not change. Even if the value of the 16-bit counter does not match the value of the TQ0CCRm register, the TOQTm pin output always changes to the high level if the crest interrupt occurs and to the low level if the valley interrupt occurs.
This is a function provided for 0% output and 100% output.
For details, see **12.4.2 (2) PWM output of 0%/100%**.

Remarks 1. i, r, k = Set values of TQ0CCRm register

2. m = 1 to 3

Figure 12-25. Example of Rewriting TQ0CCR1 to TQ0CCR3 Registers (Rewriting After Match Occurs)



<1> Matching of the count value of the 16-bit counter and the value of the TQ0CCRm 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 TQ0CCRm register value (INTCCQm) is output.

<3> The next match between the 16-bit counter and TQ0CCRm register is valid after the counter has changed its counting direction to up or down, and the PWM output changes.

If the TQ0CCRm 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 TOQTm 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, k = Set value of TQ0CCRm register

2. m = 1 to 3

(c) Rewriting TQ0OPT1 register

The interrupt culling counter is cleared when the TQ0OPT1 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 TQ0OPT1 register in the intermittent batch rewrite mode (transfer culling mode).

For details of rewriting the TQ0OPT1 register, see **12.4.3 Interrupt culling function**.

(2) Batch rewrite mode (transfer mode)

This mode is set by clearing the TQ0OPT0.TQ0CMS bit to 0, the TQ0OPT1.TQ0ID4 to TQ0OPT1.TQ0ID0 bits to 00000, and the TQ0OPT2.TQ0RDE 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 TQ0CCR1 register, the values set to the TQ0CCR0 to TQ0CCR3, TQ0OPT1, TP2CCR0, and TP2CCR1 registers are transferred all at once to the internal buffer register at the next transfer timing. Therefore, write to the TQ0CCR1 register last. Writing to the register is prohibited after the TQ0CCR1 register has been written and before the transfer timing is generated (until the crest (match between the 16-bit counter value and TQ0CCR0 register value) or the valley (match between the 16-bit counter value and 0001H)). The operation procedure is as follows.

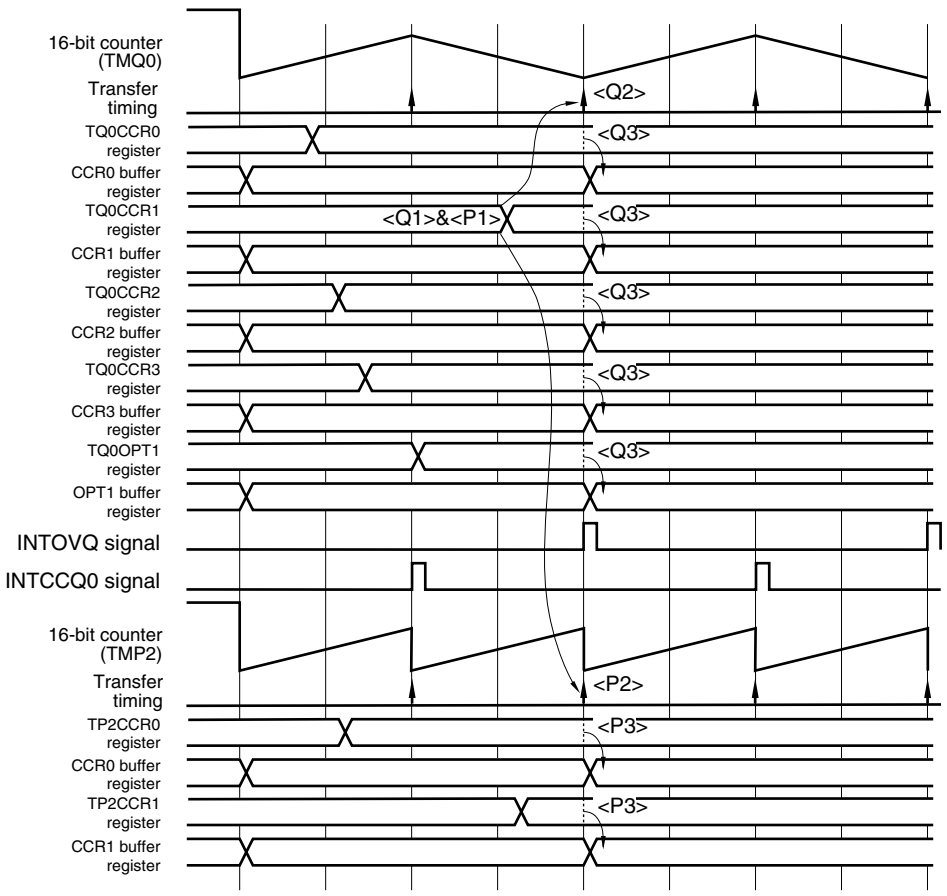
<1> Rewriting the TQ0CCR0, TQ0CCR2, TQ0CCR3, TQ0OPT1, TP2CCR0, and TP2CCR1 registers
Do not rewrite registers that do not have to be rewritten.

<2> Rewriting the TQ0CCR1 register
Rewrite the same value to the register even when it is not necessary to rewrite the TQ0CCR1 register.

<3> Holding the next rewriting pending until the transfer timing is generated
Rewrite the register next time after the INTOVQ or INTCCQ0 interrupt has occurred.

<4> Return to <1>.

Figure 12-26. Basic Operation in Batch Mode



[Operation of TMQ0]

<Q1> Write the TQ0CCR1 register

<Q2> The target timing is the first transfer timing after a write to the TQ0CCR1 register.

<Q3> The values are transferred all at once at the transfer timing.

[Operation of TMP2]

<Q1> Write the TQ2CCR1 register

<Q2> The target timing is the first transfer timing after a write to the TQ2CCR1 register.

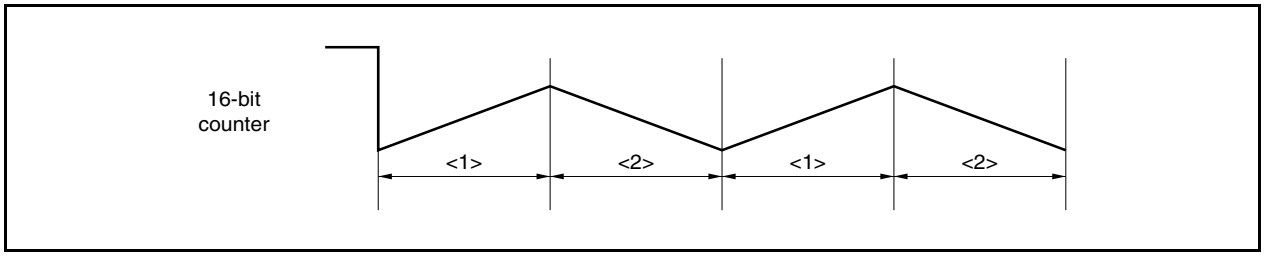
<Q3> The values are transferred all at once at the transfer timing.

(b) Rewriting TQ0CCR0 register

When rewriting the TQ0CCR0 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 TQ0CCR0 register value) or at the valley (match between the 16-bit counter value and 0001H). Usually, it is recommended to rewrite the TQ0CCR0 register while the 16-bit counter is counting down, and transfer the register value at the transfer timing of the crest timing.

Figure 12-28 shows an example of rewriting the TQ0CCR0 register while the 16-bit counter is counting up (during period <1> in Figure 12-27). Figure 12-29 shows an example of rewriting the TQ0CCR0 register while the counter is counting down (during period <2> in Figure 12-27).

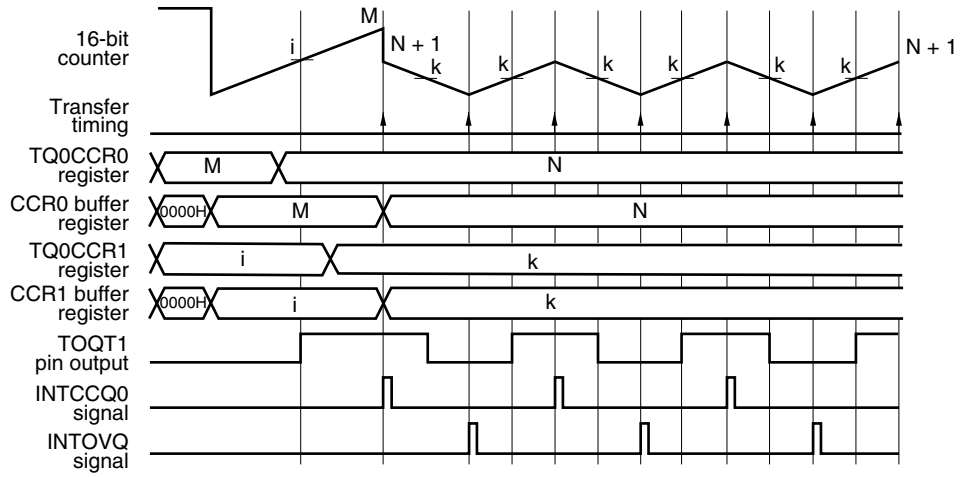
Figure 12-27. Basic Operation of 16-Bit Counter



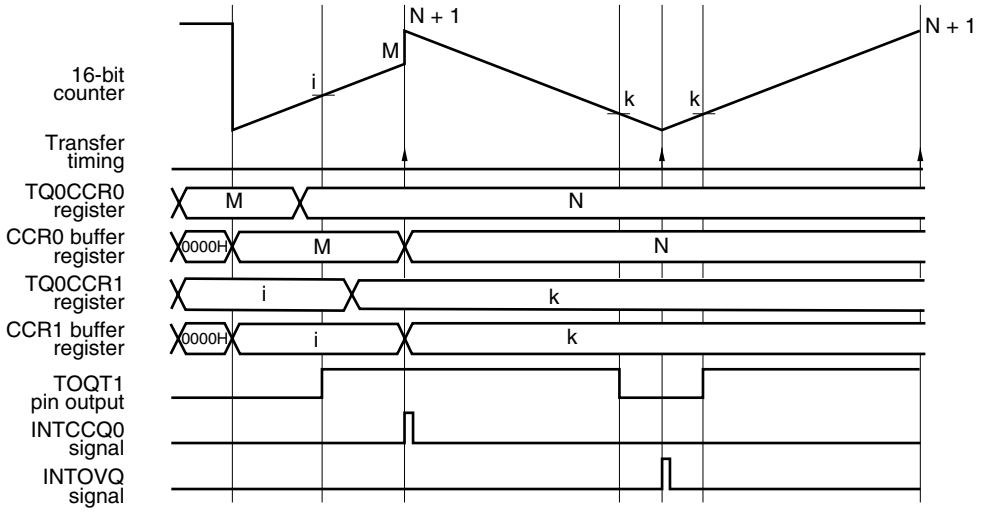
The transfer timing in Figure 12-28 is at the point where the crest timing occurs. While the 16-bit counter is counting down, the cycle changes and an asymmetrical triangular wave is output. Because the cycle changes, rewrite the duty factor (voltage data value).

Figure 12-28. Example of Rewriting TQ0CCR0 Register (During Up Counting)

(a) M > N



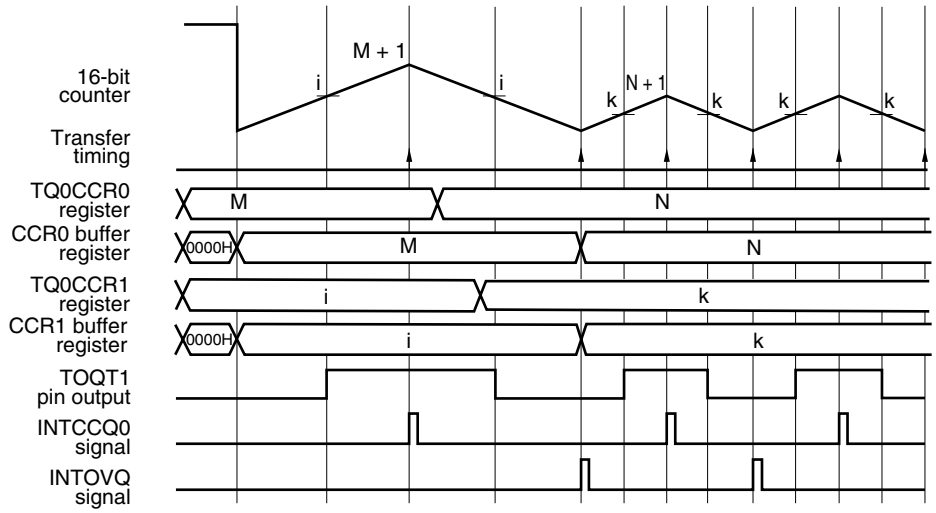
(b) M < N



Remarks 1. If transfer (match between the value of the 16-bit counter and the value of the CCR0 buffer register) occurs in the 6-phase PWM output mode, the value of the TQ0CCR0 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 TQ0CCR0 register value) timing.

2. M: Value of CCR0 buffer register before rewriting
N: Value of CCR0 buffer register after rewriting

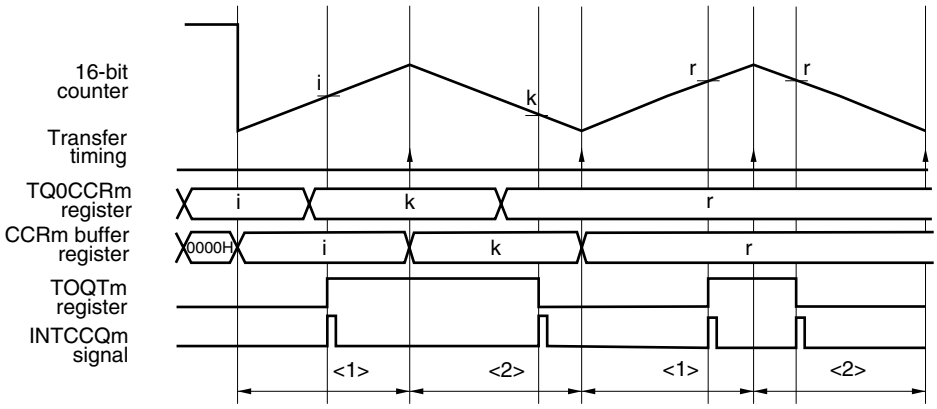
Figure 12-29. Example of Rewriting TQ0CCR0 Register (During Down Counting)



Because the next transfer timing is at the point of the valley (match between the 16-bit counter value and 0001H), the cycle value changes from the next cycle and output of a symmetrical triangular wave is maintained. Because the cycle changes, rewrite the duty value (voltage data value) as required.

(c) Rewriting TQ0CCRm register

Figure 12-30. Example of Rewriting TQ0CCRm Register



Rewriting during period <1> (rewriting during counting up)

Because the TQ0CCRm register value is transferred at the transfer timing of the crest (match between the 16-bit counter value and TQ0CCRm register value), an asymmetrical triangular wave is output.

Rewriting during period <2> (rewriting during counting down)

Because the TQ0CCRm 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 TQ0OPT1 register value

Do not set the TQ0OPT1.TQ0ID4 to TQ0OPT1.TQ0ID0 bits to other than 00000. When using the interrupt culling function, rewrite the TQ0OPT1 register in the intermittent batch rewrite mode (transfer culling mode).

For details of rewriting the TQ0OPT1 register, see **12.4.3 Interrupt culling function**.

(3) Intermittent batch rewrite mode (transfer culling mode)

This mode is set by clearing the TQ0OPT0.TQ0CMS bit to 0 and setting the TQ0OPT2.TQ0RDE bit to 1.

In this mode, the values written to each compare register are transferred to the internal buffer register all at once after the culled transfer timing and compared with the counter value. The transfer timing is the timing at which an interrupt is generated (INTCCQ0, INTOVQ) by interrupt culling.

For details of the interrupt culling function, see **12.4.3 Interrupt culling function**.

(a) Rewriting procedure

If data is written to the TQ0CCR1 register, the TQ0CCR0 to TQ0CCR3, TQ0OPT1, TP2CCR0, and TP2CCR1 registers are transferred all at once to the internal buffer register at the next transfer timing. Therefore, write to the TQ0CCR1 register last. Writing to the register is prohibited after the TQ0CCR1 register has been written until the transfer timing is generated (until the INTOVQ or INTCCQ0 interrupt occurs). The operation procedure is as follows.

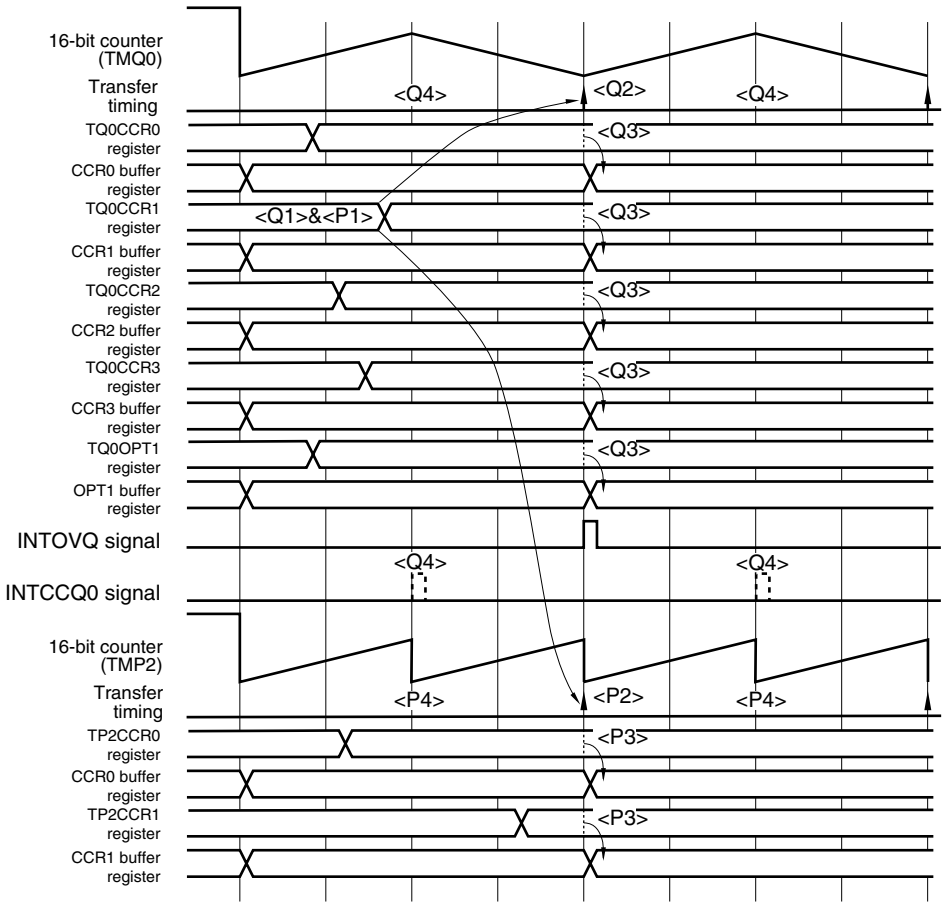
<1> Rewrite the TQ0CCR0, TQ0CCR2, TQ0CCR3, TQ0OPT1, TP2CCR0, and TP2CCR1 registers.
Do not rewrite registers that do not have to be rewritten.

<2> Rewrite the TQ0CCR1 register.
Rewrite the same value to the register even when it is not necessary to rewrite the TQ0CCR1 register.

<3> Hold the next rewriting pending until the transfer timing is generated.
Perform the next rewrite after the INTOVQ or INTCCQ0 interrupt has occurred.

<4> Return to <1>.

Figure 12-31. Basic Operation in Intermittent Batch Rewrite Mode



[TMQ0 operation]

<Q1> Write the TQ0CCR1 register.

<Q2> Rewrite the register at the transfer timing that is generated after the TQ0CCR1 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.

[TMP2 operation]

<P1> Write the TQ2CCR1 register.

<P2> Rewrite the register at the transfer timing that is generated after the TQ2CCR1 register has been rewritten.

<P3> The registers are transferred all at once at the transfer timing.

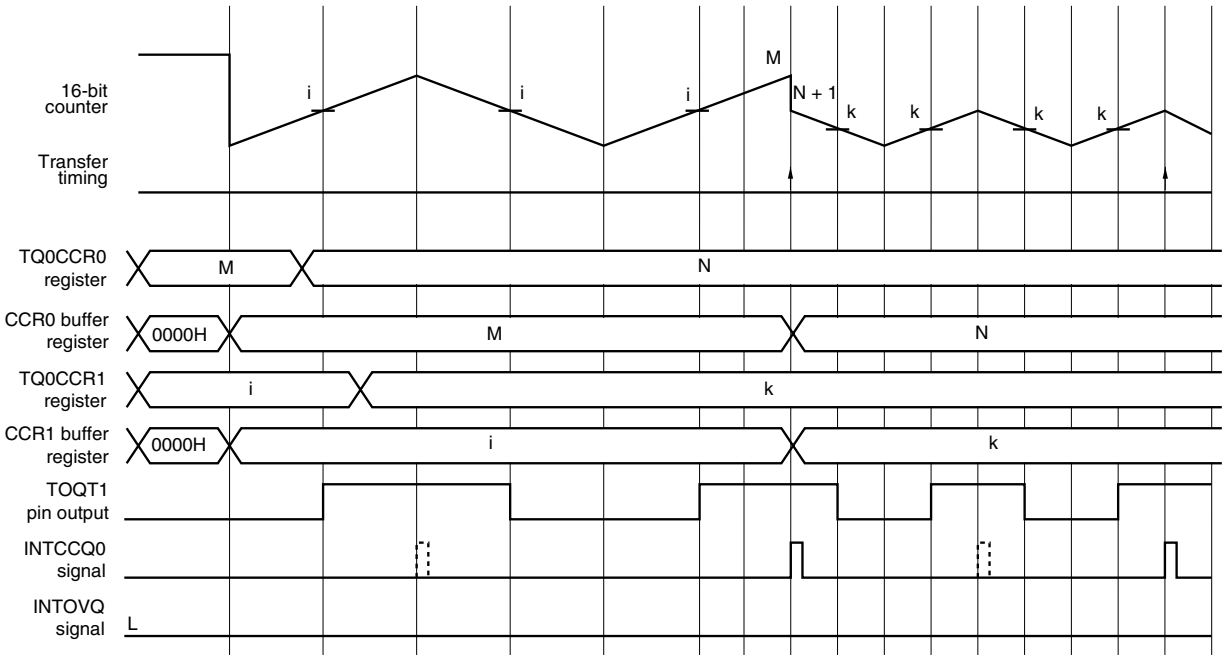
<P4> The transfer timing is also culled as the interrupts are culled.

Remark This is an example of the operation when the TQ0OPT1.TQ0ICE bit = 1, TQ0OPT1.TQ0IOE bit = 1, TQ0OPT1.TQ0ID4 to TQ0OPT1.TQ0ID0 bits = 00001.

(b) Rewriting TQ0CCR0 register

When rewriting the TQ0CCR0 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 12-32. Rewriting TQ0CCR0 Register (When Crest Interrupt Is Set)

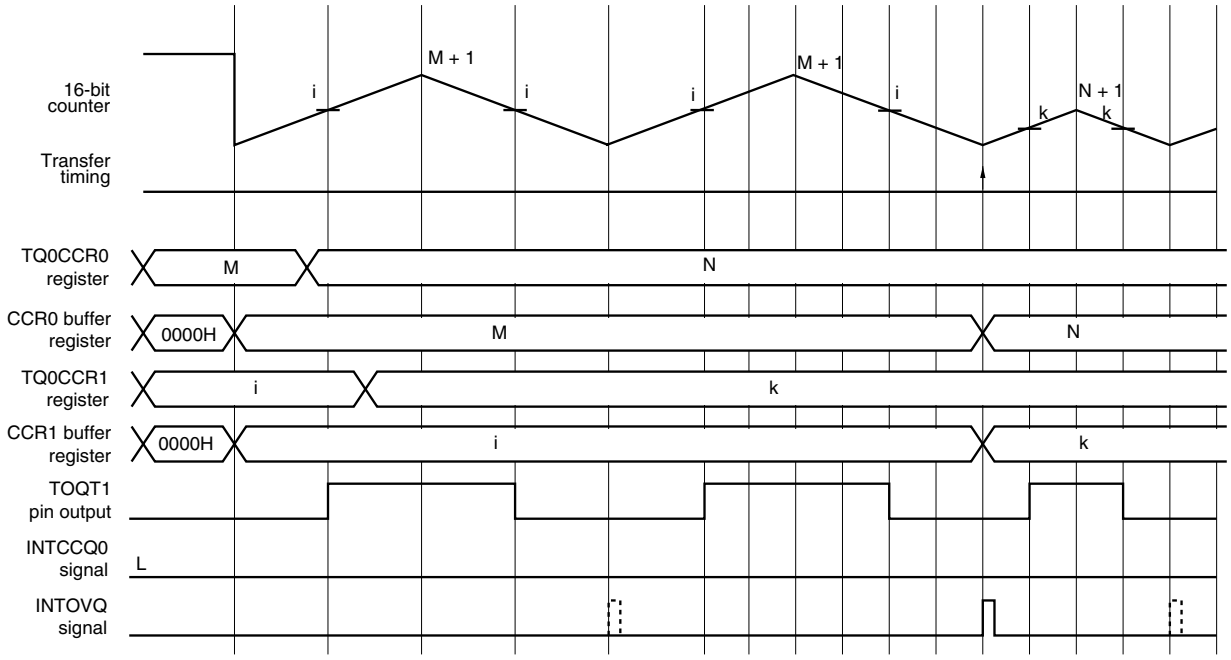


The transfer timing is generated when the crest interrupt occurs, the cycle of up counting and down counting changes, and an asymmetrical triangular wave is output.

Remarks 1. This is an example of the operation when the TQ0OPT1.TQ0ICE bit = 1, TQ0OPT1.TQ0IOE bit = 0, TQ0OPT1.TQ0ID4 to TQ0OPT1.TQ0ID0 bits = 00001.

2. : Culled interrupt

Figure 12-33. Rewriting TQ0CCR0 Register (When Valley Interrupt Is Set)



The transfer timing is generated when the valley interrupt occurs, the cycle of up counting becomes same as cycle of down counting, and a symmetrical triangular wave is output.

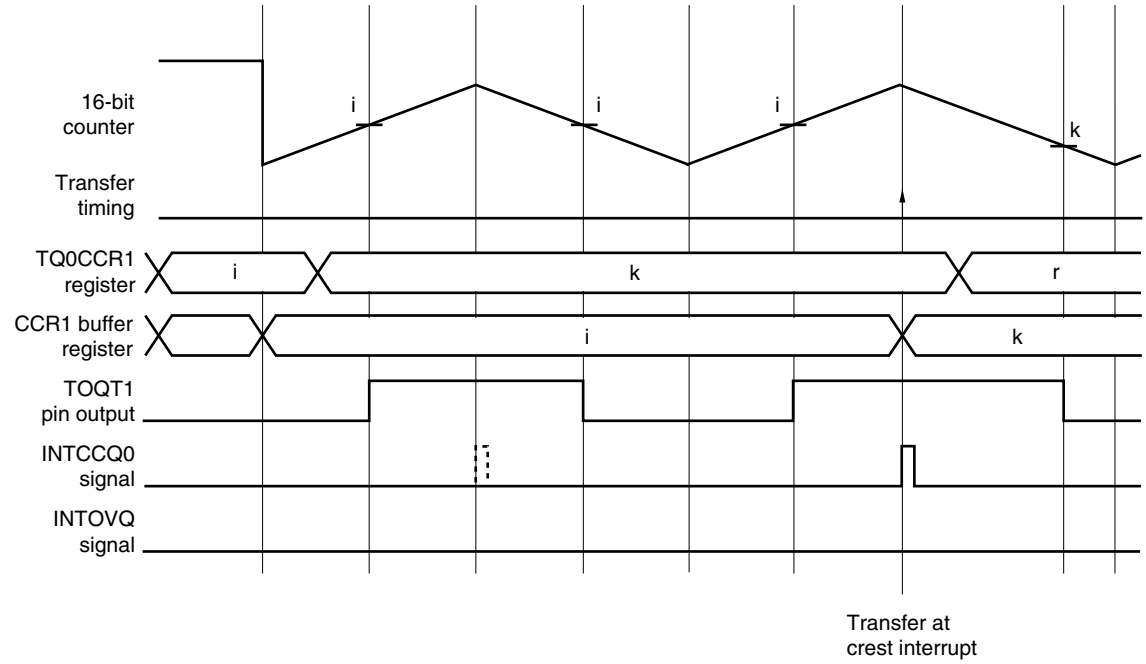
Remarks 1. This is an example of the operation when the TQ0OPT1.TQ0ICE bit = 0, TQ0OPT1.TQ0IOE bit = 1, TQ0OPT1.TQ0ID4 to TQ0OPT1.TQ0ID0 bits = 00001.

2. : Culled interrupt

(c) Rewriting TQ0CCR1 to TQ0CCR3 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.

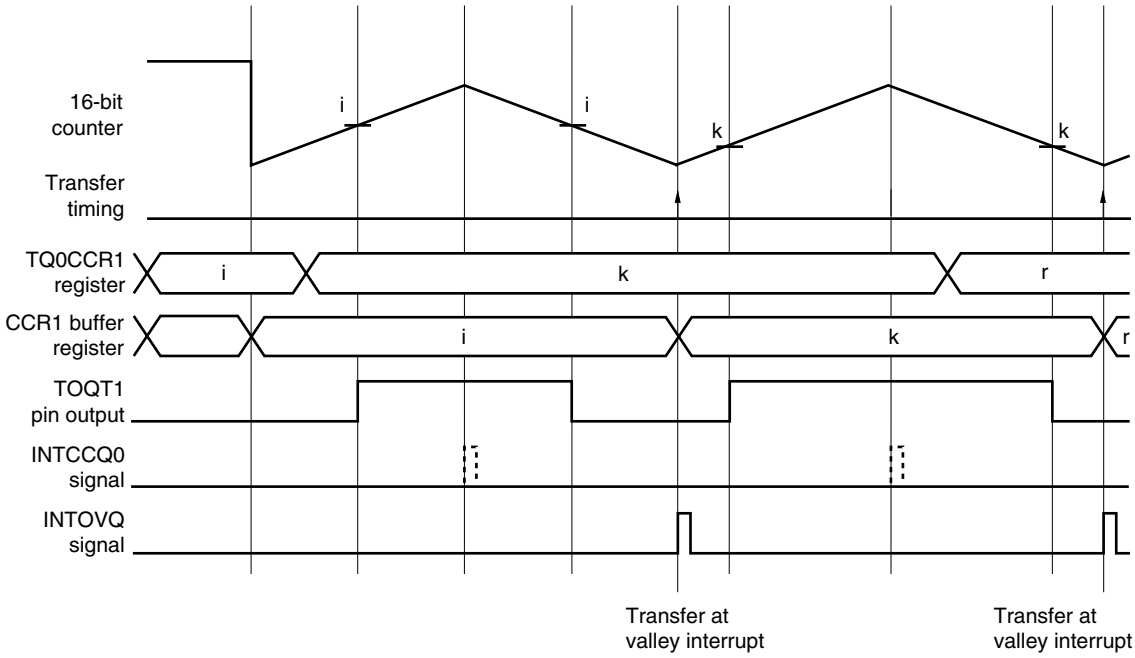
Figure 12-34. Rewriting TQ0CCR1 Register
(TQ0OPT1.TQ0ICE Bit = 1, TQ0OPT1.TQ0IOE Bit = 0, TQ0OPT1.TQ0ID4 to TQ0OPT1.TQ0ID0 Bits = 00001)



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 12-35. Rewriting TQ0CCR1 Register
(TQ0OPT1.TQ0ICE Bit = 1, TQ0OPT1.TQ0IOE Bit = 1, TQ0OPT1.TQ0ID4 to TQ0OPT1.TQ0ID0 Bits = 00001)



Remark : Culled interrupt

(d) Rewriting TQ0OPT1 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 TQ0OPT1 register, see **12.4.3 Interrupt culling function**.

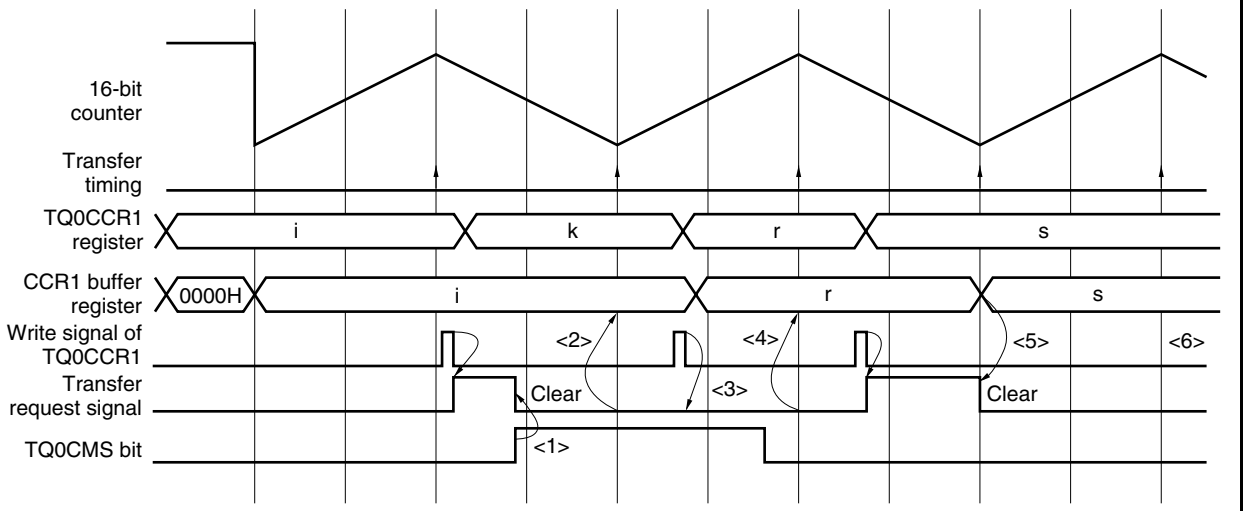
(4) Rewriting TQ0OPT0.TQ0CMS bit

The TQ0CMS bit can select the anytime rewrite mode and batch rewrite mode. This bit can be rewritten during timer operation (when TQ0CTL0.TQ0CE bit = 1). However, the operation and caution illustrated in Figure 12-31 are necessary.

If the TQ0CCR1 register is written when the TQ0CMS 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 TQ0CMS bit is set to 1.

Figure 12-36. Rewriting TQ0CMS Bit



<1> If the TQ0CCR1 register is rewritten when the TQ0CMS bit is 0, the transfer request signal is set.
If the TQ0CMS bit is set to 1 in this status, the transfer request signal is cleared.

<2> The register is not transferred because the TQ0CMS bit is set to 1 and the transfer request signal is cleared.

<3> The transfer request signal is not set even if the TQ0CCR1 register is written when the TQ0CMS bit is 1.

<4> The transfer request signal is not set even if the TQ0CCR1 register is written when the TQ0CMS bit is 1, so even if the TQ0CMS bit is cleared to 0, transfer does not occur at the subsequent transfer timing.

<5> The transfer request signal is set if the TQ0CCR1 register is written when the TQ0CMS 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.

12.4.5 TMP2 tuning operation for A/D conversion start trigger signal output

This section explains the tuning operation of TMP2 and TMQ0 in the 6-phase PWM output mode.

In the 6-phase PWM output mode, the tuning operation is performed with TMQ0 serving as the master and TMP2 as a slave. The conversion start trigger signal of the A/D converter can be set as the A/D conversion start trigger source by the INTCCP20 and INTCCP21 signals of TMP2 and the INTOVQ and INTCCQ0 signals of TMQ0.

(1) Tuning operation starting procedure

The TMP2 and TMQ0 registers should be set using the following procedure to perform the tuning operation.

(a) Setting of TMP2 register (stop the operations of TMQ0 and TMP2 (by clearing the TQ0CTL0.TQ0CE bit and TP2CTL0.TP2CE bit to 0)).

- Set the TP2CTL1 register to 85H (set the tuning operation slave mode and free-running timer mode).
- Clear the TP2OPT0 register to 00H (select the compare register).
- Set an appropriate value to the TP2CCR0 and TP2CCR1 registers (set the default value for comparison for starting the operation).

(b) Setting of TMQ0 register

- Set the TQ0CTL1 register to 07H (master mode and 6-phase PWM output mode).
- Set an appropriate value to the TQ0IOC0 register (set the output mode of TOQT1 to TOQT3).
  However, clear the TQ0OL0 bit to 0 and set the TQ0OE0 bit to 1 (enable positive phase output). Unless this setting is made, the crest interrupt (INTCCQ0) and valley interrupt (INTOVQ) do not occur. Consequently, the conversion start trigger signal of the A/D converter is not correctly generated.
- Clear the TQ0OPT0 register to 00H (select the compare register).
- Set an appropriate value to the TQ0CCR0 to TQ0CCR3 registers (set the default value for comparison for starting the operation).
- <R> Set the TQ0CTL0 register to 0xH (clear the TQ0CE bit to 0 and set the operating clock of TMQ0).
  The operating clock of TMQ0 set by the TQ0CTL0 register is also supplied to TMP2, and the count operation is performed at the same timing. The operating clock of TMP2 set by the TP2CTL0 register is ignored.

(c) Setting of TMQOP0 (TMQ0 option) register

- Set an appropriate value to the TQ0OPT1 and TQ0OPT2 registers.
- Set an appropriate value to the TQ0IOC3 register (set TOQB1 to TOQB3 in the output mode).
- Set an appropriate value to the TQ0DTC 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 TP2CE bit to 1 and set the TQ0CE bit to 1 immediately after that to start the 6-phase PWM output operation

Rewriting the TQ0CTL0, TQ0CTL1, TP0CTL0, and TP0CTL1 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 TQ0CTL0.TQ0CE bit to clear it is permitted. Manipulating (reading/writing) the other TMQ0, TMP2, and TMQ0 option registers is prohibited until the TP2CTL0.TP2CE bit is set to 1 and then the TQ0CE bit is set to 1.

Caution When tuning TMP2 in the 6-phase PWM mode, output of the TOP20 and TOP21 pins is disabled. Clear the TP2IOC0.TP2OE0 and TP2IOC0.TP2OE1 bits to 0.

(2) Tuning operation clearing procedure

To clear the tuning operation and exit the 6-phase PWM output mode, set the TMP2 and TMQ0 registers using the following procedure.

<1> Clear the TQ0CTL0.TQ0CE bit to 0 and stop the timer operation.
<2> Clear the TP2CTL0.TP2CE bit to 0 so that TMP2 can be separated.
<3> Stop the timer output by using the TQ0IOC0 and TP2IOC0 registers.
<4> Clear the TP2CTL1.TP2SYE bit to 0 to clear the tuning operation.

Caution Manipulating (reading/writing) the other TMQ0, TMP2, and TMQ0 option registers is prohibited until the TQ0CE bit is set to 1 and then the TP1CE bit is set to 1.

(3) When not tuning TMP2

When the match interrupt signal of TMP2 is not necessary as the conversion trigger source that starts the A/D converter, TMP2 can be used independently as a separate timer without being tuned. In this case, the match interrupt signal of TMP2 cannot be used as a trigger source to start A/D conversion in the 6-phase PWM output mode. Therefore, fix the TQ0OPT2.TQ0AT2 and TQ0OPT2.TQ0AT3 bits to 00.

The other control bits can be used in the same manner as when TMP2 is tuned.

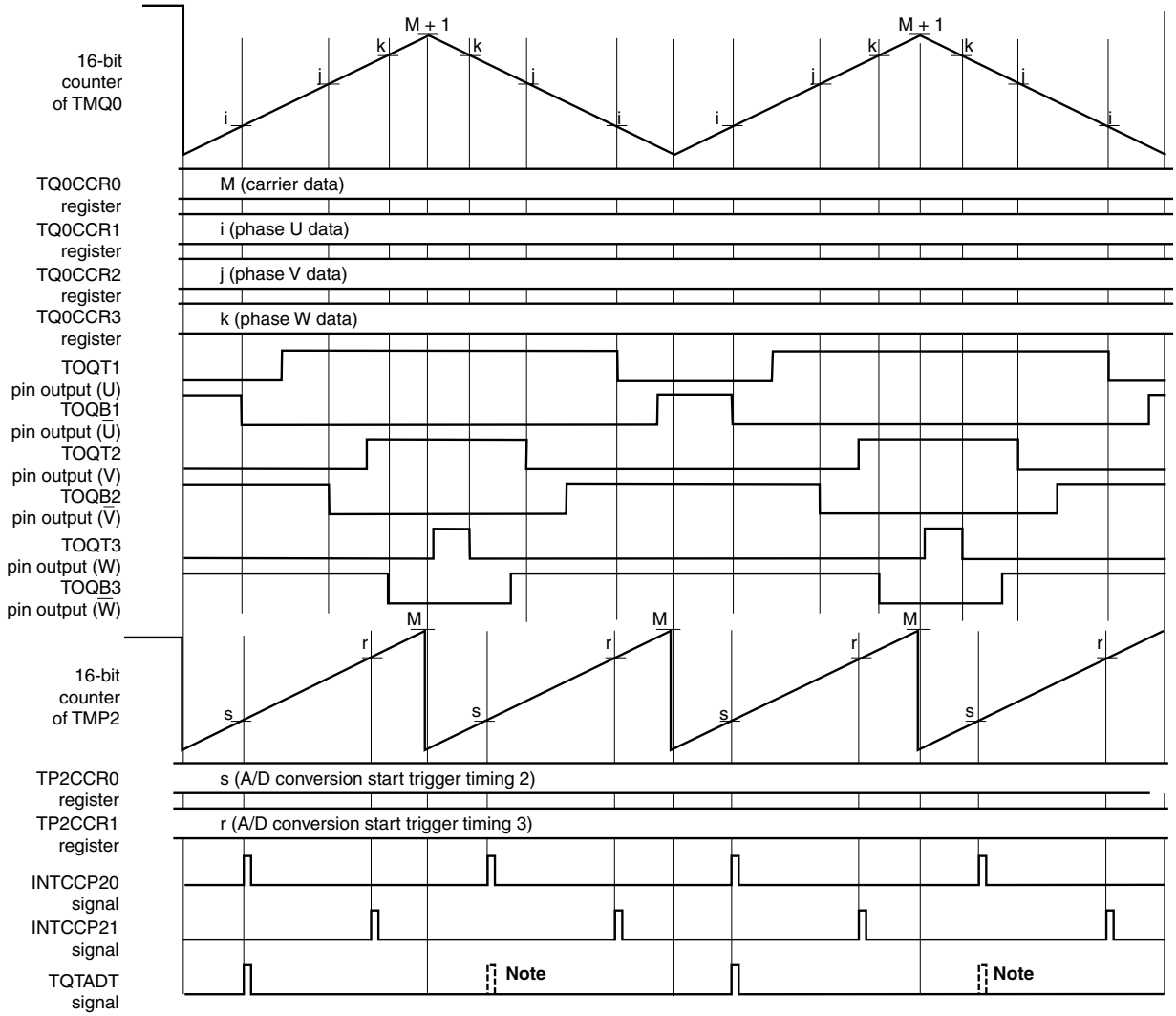
If TMP2 is not tuned, the compare registers (TP2CCR0 and TP2CCR1) of TMP2 are not affected by the setting of the TQ0OPT0.TQ0CMS and TQ0OPT2.TQ0RDE bit. For the initialization procedure when TMP2 is not tuned, see (b) to (e) in **12.4.5 (1) Tuning operation starting procedure**. (a) is not necessary because it is a step used to set TMP2 for the tuning operation.

(4) Basic operation of TMP2 during tuning operation

The 16-bit counter of TMP2 only counts up. The 16-bit counter is cleared by the set cycle value of the TQ0CCR0 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 TMQ0 when it counts up. However, it is not the same when the 16-bit counter of TMP2 counts down.

- When TMQ0 counts up (same value)
  16-bit counter of TMQ0: 0000H → M (up counting)
  16-bit counter of TMP2: 0000H → M (up counting)
- When TMQ0 counts down (not same value)
  16-bit counter of TMQ0: M + 1 → 0001H (down counting)
  16-bit counter of TMP2: 0000H → M (up counting)

Figure 12-37. TMP2 During Tuning Operation



Note The TQTADT signal is masked by the TQ0OPT2.TQ0ATM2 and TQ0OPT2.TQ0ATM3 bits.

12.4.6 A/D conversion start trigger output function

The V850E/MA3 has a function to select four trigger sources (INTOVQ, INTCCQ0, INTCCP20, INTCCP21) to generate the A/D conversion start trigger signal (TQTADT).

The trigger sources are specified by the TQ0OPT2.TQ0AT0 to TQ0OPT2.TQ0AT3 bits.

- TQ0AT0 bit = 1:
  A/D conversion start trigger signal generated when INTOVQ (counter underflow) occurs.
- TQ0AT1 bit = 1:
  A/D conversion start trigger signal generated when INTCCQ0 (cycle match) occurs.
- TQ0AT2 bit = 1:
  A/D conversion start trigger signal generated when INTCCP20 (match of TP2CCR0 register of TMP2 during tuning operation) occurs.
- TQ0AT3 bit = 1:
  A/D conversion start trigger signal generated when INTCCP21 (match of TP2CCR1 register of TMP2 during tuning operation) occurs.

The A/D conversion start trigger signals selected by the TQ0AT0 to TQ0AT3 bits are ORed and output. Therefore, two or more trigger sources can be specified at the same time.

The INTOVQ and INTCCQ0 signals selected by the TQ0AT0 and TQ0AT1 bits are culled interrupt signals.

Therefore, these signals are output after the interrupts have been culled and, unless interrupt output is enabled (TQ0OPT1.TQ0ICE, TQ0OPT1.TQ0IOE bits), the A/D conversion start trigger signal is not output.

The trigger sources (INTCCP20 and INTCCP21) from TMP2 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 TQ0AT2 and TQ0AT3 bits.

- TQ0ATM2 bit: Correspond to the TQ0AT2 bit and control INTCCP20 (match interrupt signal) of TMP2.
  - TQ0ATM2 bit = 0: The A/D conversion start trigger signal is output when the 16-bit counter counts up (TQ0OPT0.TQ0CUF bit = 0), and the A/D conversion start trigger signal is not output when the 16-bit counter counts down (TQ0OPT0.TQ0CUF bit = 1).
  - TQ0ATM2 bit = 1: The A/D conversion start trigger signal is output when the 16-bit counter counts up (TQ0OPT0.TQ0CUF bit = 1), and the A/D conversion start trigger signal is not output when the 16-bit counter counts down (TQ0OPT0.TQ0CUF bit = 0).

- TQ0ATM3 bit: Correspond to the TQ0AT3 bit and control INTCCP21 (match interrupt signal) of TMP2.
  - TQ0ATM3 bit = 0: The A/D conversion start trigger signal is output when the 16-bit counter counts up (TQ0OPT0.TQ0CUF bit = 0), and the A/D conversion start trigger signal is not output when the 16-bit counter counts down (TQ0OPT0.TQ0CUF bit = 1).
  - TQ0ATM3 bit = 1: The A/D conversion start trigger signal is output when the 16-bit counter counts up (TQ0OPT0.TQ0CUF bit = 1), and the A/D conversion start trigger signal is not output when the 16-bit counter counts down (TQ0OPT0.TQ0CUF bit = 0).

The TQ0ATM3, TQ0ATM2, and TQ0AT3 to TQ0AT0 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 signal. 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 TQ0AT2 and TQ0AT3 bits can be used only when TMP2 is performing a tuning operation as the slave timer of TMQ0. If TMQ0 and TMP2 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 TMQ0 signal output is internally used to identify whether the 16-bit counter is counting up or down. Therefore, enable TOQ0 pin output by clearing the TQ0IOC0.TQ0OL0 bit to 0 and setting the TQ0IOC0.TQ0OE0 bit to 1.

Figure 12-38. Example of A/D Conversion Start Trigger (TQTADT) Signal Output (TQ0OPT1.TQ0ICE Bit = 1, TQ0OPT1.TQ0IOE Bit = 1, TQ0OPT1.TQ0ID4 to TQ0OPT1.TQ0ID0 Bits = 00000: Without Interrupt Culling)

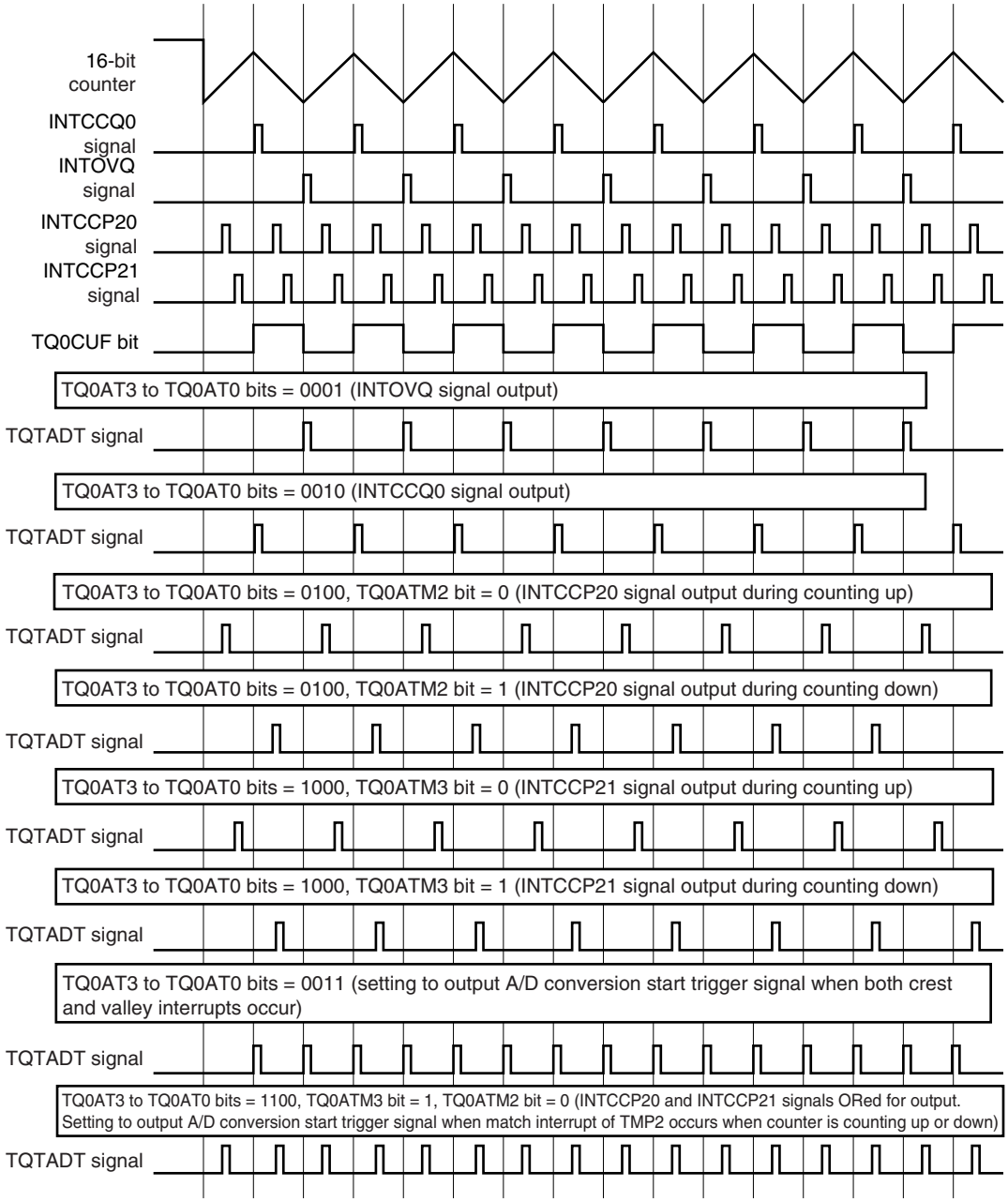
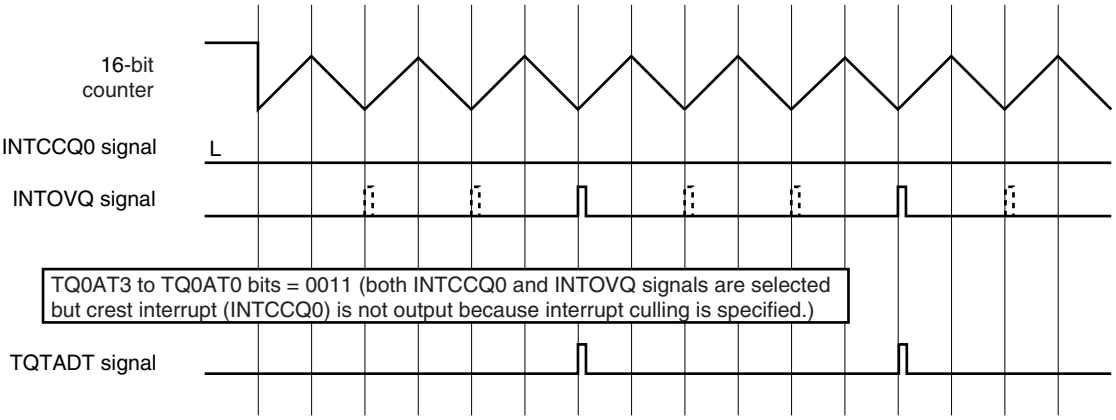
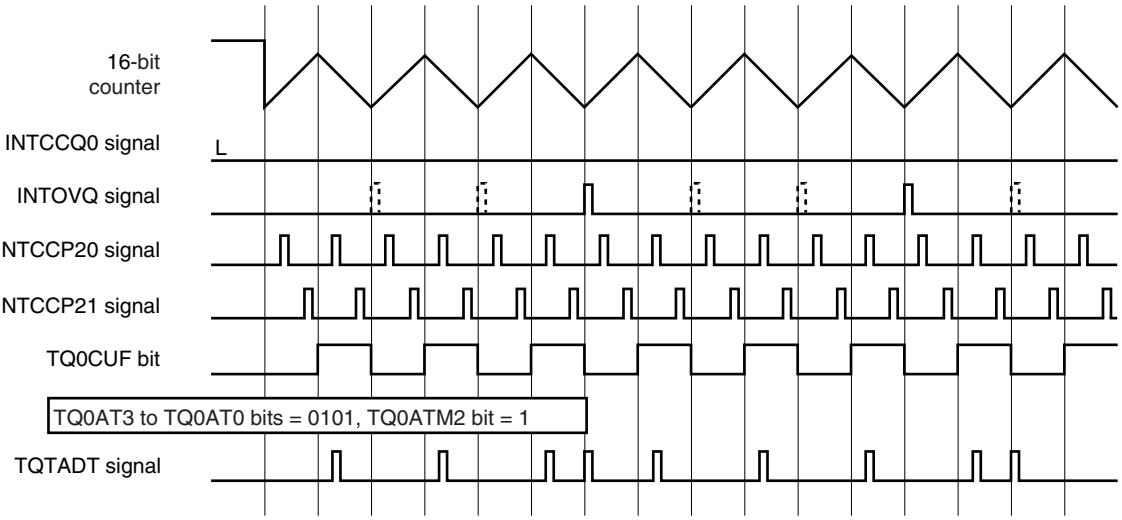


Figure 12-39. Example of A/D Conversion Start Trigger (TQTADT) Signal Output (TQ0OPT1.TQ0ICE Bit = 0, TQ0OPT1.TQ0IOE Bit = 1, TQ0OPT1.TQ0ID4 to TQ0OPT1.TQ0ID0 Bits = 00010: With Interrupt Culling) (1)



Remark : Culled interrupt

Figure 12-40. Example of A/D Conversion Start Trigger (TQTADT) Signal Output (TQ0OPT1.TQ0ICE Bit = 0, TQ0OPT1.TQ0IOE Bit = 1, TQ0OPT1.TQ0ID4 to TQ0OPT1.TQ0ID0 Bits = 00010: With Interrupt Culling) (2)



Caution The INTCCQ0 signal is culled but the INTCCP20 signal is not.

Remark : Culled interrupt

(1) Operation under boundary condition (operation when 16-bit counter matches INTCCP20 signal)

Table 12-3. Operation When TQ0CCR0 Register = M, TQ0AT2 Bit = 1, TQ0ATM2 Bit = 0 (Up Counting Period Selected)

Value of TP2CCR0 Register	Value of 16-Bit Counter of TMQ0	Value of 16-Bit Counter of TMP2	Status of 16-Bit Counter of TMQ0	TQTADT Signal Output by INTCCP20 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 12-4. Operation When TQ0CCR0 Register = M, TQ0AT2 Bit = 1, TQ0ATM2 Bit = 1 (Down Counting Period Selected)

Value of TP2CCR0 Register	Value of 16-Bit Counter of TMQ0	Value of 16-Bit Counter of TMP2	Status of 16-Bit Counter of TMQ0	TQTADT Signal Output by INTCCP20 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 TP2CCRm register enables setting of "0" to "M" when the TQ0CCR0 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 TMP2 is cleared by "M". Therefore, the TQTADT signal is not output.

CHAPTER 13 WATCHDOG TIMER FUNCTIONS

13.1 Functions

The watchdog timer has the following operation modes.

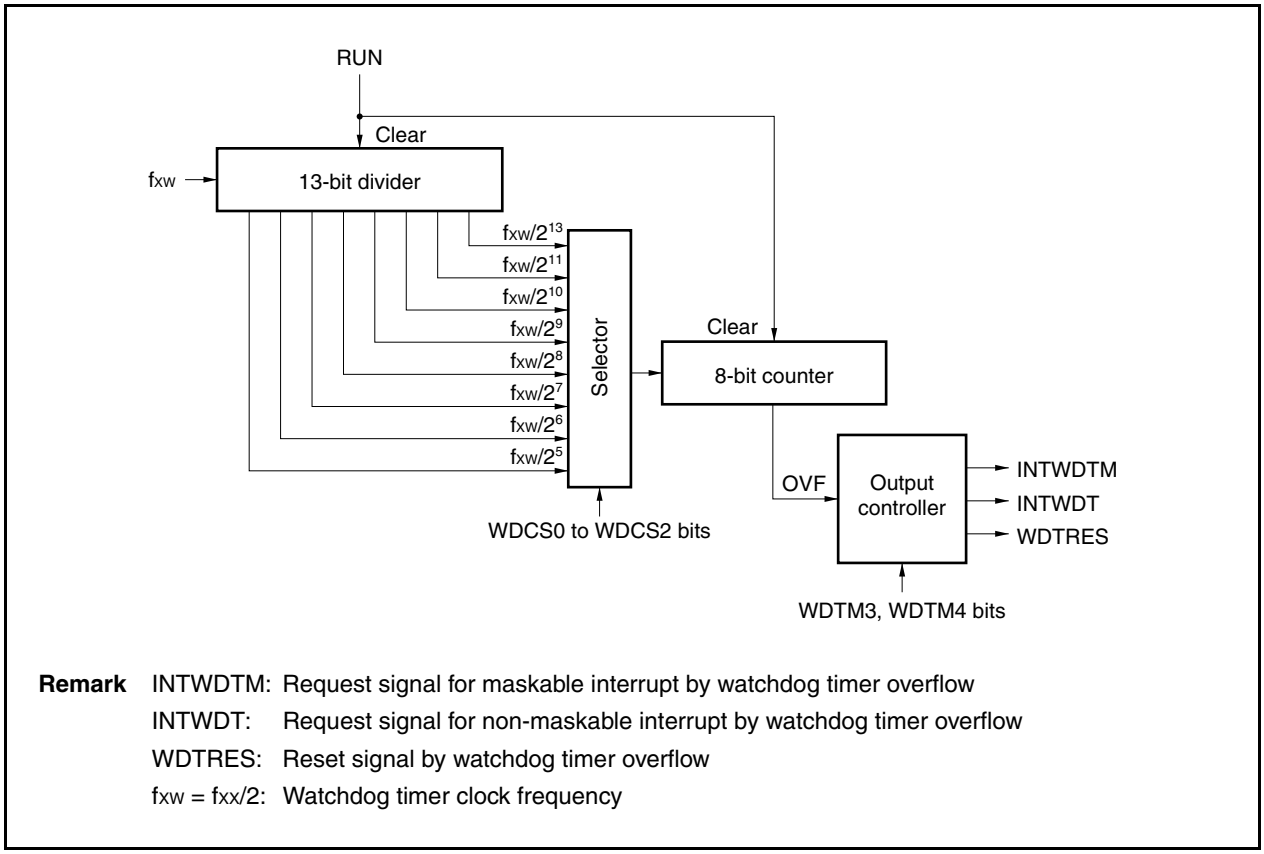
- Watchdog timer
- Interval timer

The following functions are realized from the above-listed operation modes.

- Generation of non-maskable interrupt request signal (INTWDT) upon overflow of watchdog timer
- Generation of system reset signal (WDTRES) upon overflow of watchdog timer
- Generation of maskable interrupt request signal (INTWDTM) upon overflow of interval timer

Remark Select whether to use the watchdog timer in the watchdog timer mode or the interval timer mode using the WDTM register.

Figure 13-1. Block Diagram of Watchdog Timer



Remark INTWDTM: Request signal for maskable interrupt by watchdog timer overflow
INTWDT: Request signal for non-maskable interrupt by watchdog timer overflow
WDTRES: Reset signal by watchdog timer overflow
$f_{XW} = f_{XX}/2$: Watchdog timer clock frequency

13.2 Configuration

The watchdog timer consists of the following hardware.

Table 13-1. Configuration of Watchdog Timer

Item	Configuration
Control registers	Watchdog timer clock select register (WDCS) Watchdog timer mode register (WDTM) Watchdog timer reset status register (WDRES)

13.3 Control Registers

The registers that control the watchdog timer are as follows.

- Watchdog timer clock select register (WDCS)
- Watchdog timer mode register (WDTM)
- Watchdog timer reset status register (WDRES)

(1) Watchdog timer clock select register (WDCS)

The WDCS register sets the overflow time of the watchdog timer and the interval timer.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF6C1H

	7	6	5	4	3	2	1	0
WDCS	0	0	0	0	0	WDCS2	WDCS1	WDCS0

WDCS2	WDCS1	WDCS0	Overflow time of watchdog timer/interval timer		
				f_{XX}	
				80 MHz	60 MHz
0	0	0	$2^{14}/f_{XX}$	204.8 μs	273.1 μs
0	0	1	$2^{15}/f_{XX}$	409.6 μs	546.1 μs
0	1	0	$2^{16}/f_{XX}$	819.2 μs	1.092 ms
0	1	1	$2^{17}/f_{XX}$	1.638 ms	2.185 ms
1	0	0	$2^{18}/f_{XX}$	3.277 ms	4.369 ms
1	0	1	$2^{19}/f_{XX}$	6.554 ms	8.738 ms
1	1	0	$2^{20}/f_{XX}$	13.11 ms	17.48 ms
1	1	1	$2^{22}/f_{XX}$	52.43 ms	69.91 ms

Remark $f_{XW} = f_{XX}/2$: Watchdog timer clock frequency

(2) Watchdog timer mode register (WDTM)

The WDTM register sets the watchdog timer operation mode and enables/disables count operations.

This register is a special register that can be written only in a specific sequence (see **3.4.9 Special registers**).

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF6C2H

	<7>	6	5	4	3	2	1	0
WDTM	RUN	0	0	WDTM4	WDTM3	0	0	0

RUN	Selection of operation mode of watchdog timer[Note 1]
0	Stop counting
1	Clear counter and start counting

WDTM4	WDTM3	Selection of operation mode of watchdog timer[Note 2]
0	0	Interval timer mode
0	1	(Upon overflow, maskable interrupt INTWDTM is generated.)
1	0	Watchdog timer mode 1 (Upon overflow, non-maskable interrupt INTWDT is generated.)
1	1	Watchdog timer mode 2 (Upon overflow, reset operation WDTRES is started.)

Notes 1. Once the RUN bit is set (to 1), it cannot be cleared (to 0) by software.
Therefore, when counting is started, it cannot be stopped except by reset input.

2. Once the RUN bit is set to 1, the WDTM3 and WDTM4 bits cannot be cleared (to 0) by software and can be cleared only by reset.

(3) Watchdog timer reset status register (WDRES)

When the V850E/MA3 has been reset, the WDRES register is used to check whether it has been reset by the watchdog timer (WDTRES) or not.

The WDRES register is a special register. The status of WDTRES is shown below.

This register is set only in 8-bit units when it is written, and in 8-bit or 1-bit units when it is read.

To write the WDRES register, a specific sequence using the PRCMD register as the command register is required. If the register is written in an illegal sequence, the written data is invalid and the protect error bit (SYS.PRERR bit) is set to 1, and the write operation is not performed.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF82AH

	7	6	5	4	3	2	1	<0>
WDRES	0	0	0	0	0	0	0	WRESF

WRESF	WDTRES detection flag
0	WDTRES not generated
1	WDTRES generated
Setting (1) condition:	Reset signal generation due to watchdog timer (WDT) overflows
Clearing (0) condition:	Writing 0 by instruction or $\overline{\text{RESET}}$ pin input Only 0 can be written to the WRESF bit.

Caution Before writing 0 to the WRESF bit, check if the WRESF bit is 1 (read) to avoid a conflict with the flag setting.

Remark The WRESF bit can be read/written but it can only cleared by writing 0 and cannot be operated by writing 1.

13.4 Operation

13.4.1 Operation as watchdog timer

The watchdog timer operation to detect a program loop is selected by the WDTM.WDTM4 bit to 1.

The count clock (program loop detection time interval) of the watchdog timer can be selected using the WDCS.WDCS0 to WDCS.WDCS2 bits. The count operation is started by setting the WDTM.RUN bit to 1. When, after the count operation is started, the RUN bit is again set to 1 within the set program loop detection time interval, the watchdog timer is cleared and the count operation starts again.

If the program loop detection time is exceeded without the RUN bit being set to 1, a reset (WDTRES) or a non-maskable interrupt request signal (INTWDT) is generated, depending on the value of the WDTM.WDTM3 bit.

The count operation of the watchdog timer stops in the software STOP mode and IDLE mode. Therefore, set the RUN bit to 1 before the software STOP mode or IDLE mode is entered in order to clear the watchdog timer.

Table 13-2. Program Loop Detection Time of Watchdog Timer

Clock	Program Loop Detection Time	
	80 MHz	60 MHz
$2^{14}/f_{XX}$	204.8 μs	273.1 μs
$2^{15}/f_{XX}$	409.6 μs	546.1 μs
$2^{16}/f_{XX}$	819.2 μs	1.092 ms
$2^{17}/f_{XX}$	1.638 ms	2.185 ms
$2^{18}/f_{XX}$	3.277 ms	4.369 ms
$2^{19}/f_{XX}$	6.554 ms	8.738 ms
$2^{20}/f_{XX}$	13.11 ms	17.48 ms
$2^{22}/f_{XX}$	52.43 ms	69.91 ms

Remark $f_{XW} = f_{XX}/2$: Watchdog timer clock frequency

13.4.2 Operation as interval timer

The watchdog timer can be made to operate as an interval timer that repeatedly generates interrupts using the count value set in advance as the interval, by setting the WDTM.WDTM4 bit to 0.

When the watchdog timer operates as an interval timer, the interrupt mask flag (WDTMK) and priority specification flags (WDTPR0 to WDTPR2) of the WDTIC register are valid and maskable interrupt request signals (INTWDTM) can be generated. The default priority of the INTWDTM signal is set to the highest level among the maskable interrupt request signals.

The interval timer continues to operate in the HALT mode, but it stops operating in the software STOP mode and the IDLE mode.

Caution Once the WDTM4 bit is set to 1 (thereby selecting the watchdog timer mode) and the RUN bit is set (1), the interval timer mode is not changed unless a reset is performed.

Table 13-3. Interval Time of Interval Timer

Clock	Interval Time	
	80 MHz	60 MHz
2^{14}/fxx	204.8 μs	273.1 μs
2^{15}/fxx	409.6 μs	546.1 μs
2^{16}/fxx	819.2 μs	1.092 ms
2^{17}/fxx	1.638 ms	2.185 ms
2^{18}/fxx	3.277 ms	4.369 ms
2^{19}/fxx	6.554 ms	8.738 ms
2^{20}/fxx	13.11 ms	17.48 ms
2^{22}/fxx	52.43 ms	69.91 ms

Remark fxw = fxx/2: Watchdog timer clock frequency

CHAPTER 14 A/D CONVERTER

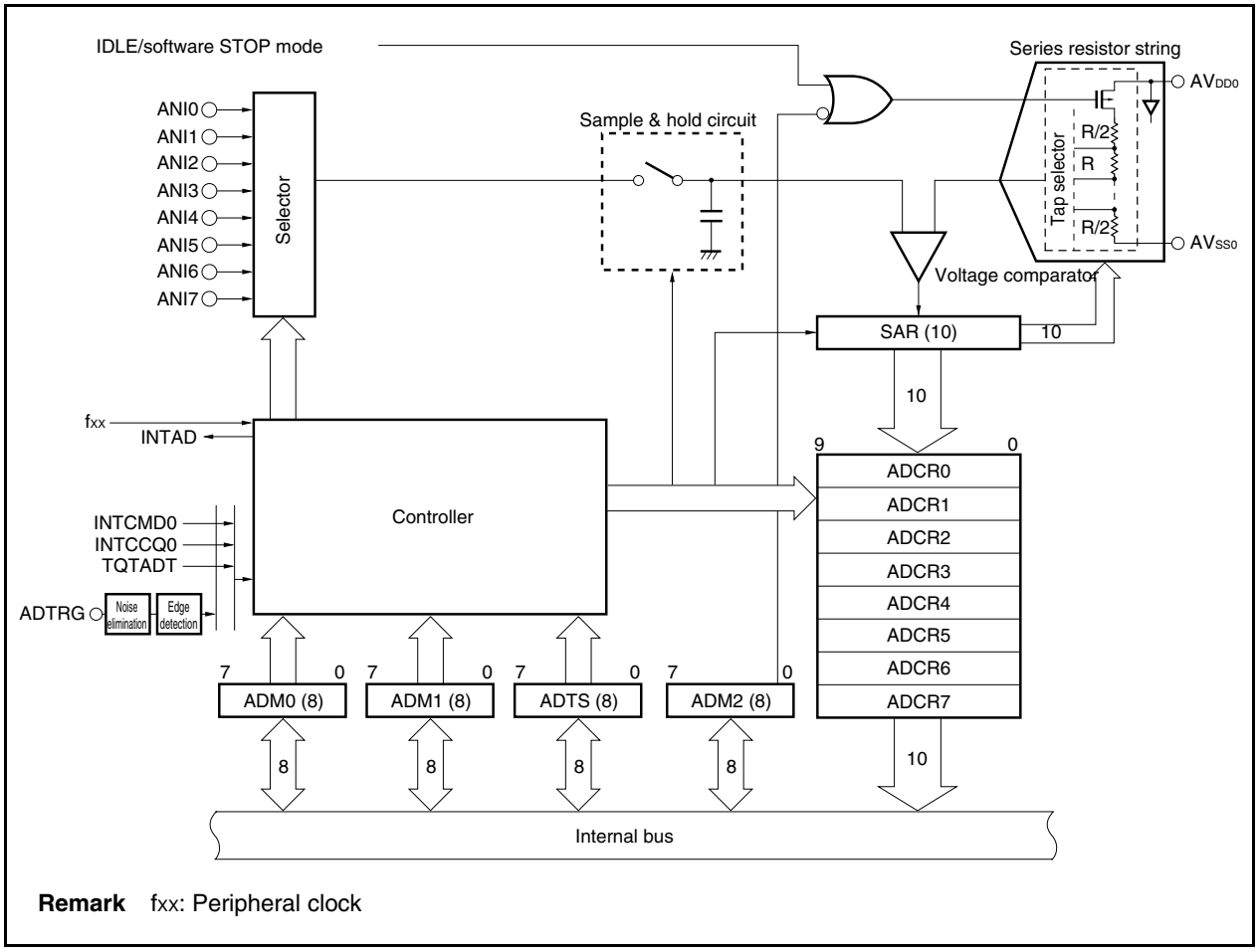
14.1 Features

- Analog input: 8 channels
- 10-bit A/D converter
- On-chip A/D conversion result registers (ADCR0 to ADCR7)
  10 bits × 8
- A/D conversion trigger mode
  Software trigger mode
  Timer trigger mode
  External trigger mode
- A/D conversion operation mode
  Select mode
  Scan mode
- Buffer mode
  1-buffer mode
  4-buffer mode
- Successive approximation method
- Operable when the internal system clock (fxx) is 10 MHz or higher

14.2 Configuration

The block diagram is shown below.

Figure 14-1. Block Diagram of A/D Converter



Remark fXX: Peripheral clock

Cautions 1. If there is noise at the analog input pins (ANI0 to ANI7) or at the A/D converter power supply voltage pin (AV_{DD0}), that noise may generate an illegal conversion result.
Software processing will be needed to avoid a negative effect on the system from this illegal conversion result.
An example of this software processing is shown below.

- **Take the average result of a number of A/D conversions and use that as the A/D conversion result.**
- **Execute a number of A/D conversions consecutively and use those results, omitting any exceptional results that may have been obtained.**
- **If an A/D conversion result that is judged to have generated a system malfunction is obtained, be sure to recheck the system malfunction before performing malfunction processing.**

2. Do not apply a voltage outside the AV_{SS0} to AV_{DD0} range to the pins that are used as A/D converter input pins.
3. The A/D converter is operable when the peripheral clock (f_{XX}) is 10 MHz or higher.

The A/D converter consists of the following hardware.

Table 14-1. Configuration of A/D Converter

Item	Configuration
Analog input	ANI0 to ANI7 (total of eight channels)
Registers	Successive approximation register (SAR) A/D conversion result registers 0 to 7 (ADCR0 to ADCR7) A/D conversion result registers 0H to 7H (ADCR0H to ADCR7H)
Control registers	A/D converter mode register 0 (ADM0) A/D converter mode register 1 (ADM1) A/D converter mode register 2 (ADM2) A/D trigger select register (ADTS)

(1) Selector

The input circuit selects the analog input pins (ANI0 to ANI7) according to the mode set by the ADM0 and ADM1 registers and sends the input to the sample & hold circuit.

(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

The voltage comparator compares the analog input voltage with the output voltage of the series resistor string voltage tap.

(4) Series resistor string

The series resistor string is used to generate voltages to match the analog input voltage.

The series resistor string is connected between the power supply voltage pin for the A/D converter (AV_{DD0}) and the ground pin for the A/D converter (AV_{SS0}). To make 1,024 equivalent voltage steps between these 2 pins, it is configured from 1,023 equivalent resistors and 2 resistors with 1/2 of the resistance value.

The voltage tap of the series resistor string is selected by a tap selector controlled by the successive approximation register (SAR).

(5) Successive approximation register (SAR)

The SAR is a 10-bit register that sets series resistor string voltage tap data, whose values 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) (A/D conversion completed), the contents of the SAR (conversion results) are held in A/D conversion result register n (ADCRn) (n = 0 to 7). When all the specified A/D conversion operations have been completed, an A/D conversion end interrupt request signal (INTAD) is generated.

(6) A/D conversion result register n (ADCRn)

The ADCRn register is a 10-bit register that holds the A/D conversion results (n = 0 to 7). Each time A/D conversion is completed, the conversion results are loaded from the successive approximation register (SAR) and stored in the lower 10 bits of the ADCRn register. The higher 6 bits of this register is always 0.

(7) A/D converter mode register 0 (ADM0)

This register is used to select the analog input pin, specify the operation mode, and control the conversion operation.

(8) A/D converter mode register 1 (ADM1)

This register is used to set the conversion operation time of the analog input to be A/D converted and specify the trigger mode.

(9) A/D converter mode register 2 (ADM2)

This register is used to reset the A/D converter and control the clocks.

(10) A/D trigger select register (ADTS)

This register is used to specify the timer trigger input.

(11) Controller

The controller selects the analog input pin, generates the sample & hold circuit operation timing, and controls the conversion trigger according to the mode set by the ADM0 to ADM2 and ADTS registers.

(12) ANI0 to ANI7 pins

These are 8-channel analog input pins for the A/D converter. They input the analog signals to be A/D converted.

Caution Make sure that the voltages input to ANI0 to ANI7 do not exceed the rated values. If a voltage higher than or equal to AV_{DD0} or lower than or equal to AV_{SS0} (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.

(13) AV_{DD0} pin

This alternates with the pin for inputting the positive power supply and reference voltage of the A/D converter. It converts signals input to the ANIn pin to digital signals based on the voltage applied between AV_{DD0} and AV_{SS0}.

Always make the potential at this pin the same as that at the EV_{DD} pin even when the A/D converter is not used. Do not stop supplying power to the AV_{DD0} pin even in the standby status.

(14) AV_{SS0} pin

This is the ground pin of the A/D converter. Always make the potential at this pin the same as that at the EV_{SS} pin.

14.3 Control Registers

The A/D converter is controlled by the following registers.

- A/D converter mode registers 0 to 2 (ADM0 to ADM2)
- A/D trigger select register (ADTS)

The following registers are also used.

- A/D conversion result registers 0 to 7 (ADCR0 to ADCR7)
- A/D conversion result registers 0H to 7H (ADCR0H to ADCR7H)

(1) A/D converter mode register 0 (ADM0)

The ADM0 register is an 8-bit register that selects the analog input pin, specifies the operation mode, and executes conversion operations.

This register can be read or written in 8-bit or 1-bit units. However, when data is written to the ADM0 register during an A/D conversion operation, the conversion operation is initialized and conversion is executed from the beginning. Bit 6 is read-only and writing executed is ignored.

Reset input clears this register to 00H.

Cautions 1. When the ADCE bit is 1 in the timer/external trigger mode, the trigger signal standby state is set. To clear the ADCE bit, write 0, clear (0) the ADM2.ADCAE bit, or reset the V850E/MA3.

In the software trigger mode, the conversion trigger is set by writing 1 to the ADCE bit. After the operation, when the mode is changed to the timer/external trigger mode without clearing the ADCE bit, the trigger input standby state is set immediately after the register value is changed.

2. It takes 7 to 9 clocks until the ADCS bit is set to 1 from when the ADCE bit was set to 1 in the software trigger mode.

After reset: 00H R/W Address: FFFFF200H

	<7>	<6>	5	4	3	2	1	0
ADM0	ADCE	ADCS	BS	MS	0	ANIS2	ANIS1	ANIS0

ADCE	A/D conversion control
0	Conversion disabled
1	Conversion enabled

ADCS	A/D converter status
0	A/D conversion stopped
1	A/D conversion operating

BS	Buffer mode specification
0	1-buffer mode
1	4-buffer mode

MS	Operation mode specification
0	Scan mode
1	Select mode

Caution For ANIS2 to ANIS0 bits, see Table 14-2 Specification of Analog Input Pin.

Table 14-2. Specification of Analog Input Pin

ANIS2	ANIS1	ANIS0	Select Mode		Scan Mode	
			Software Trigger Mode	Timer/External Trigger Mode[Note]	Software Trigger Mode	Timer/External Trigger Mode
0	0	0	ANI0	ANI0	ANI0	1
0	0	1	ANI1	ANI1	ANI0, ANI1	2
0	1	0	ANI2	ANI2	ANI0 to ANI2	3
0	1	1	ANI3	ANI3	ANI0 to ANI3	4
1	0	0	ANI4	Setting prohibited	ANI0 to ANI4	4 + ANI4
1	0	1	ANI5	Setting prohibited	ANI0 to ANI5	4 + ANI4, ANI5
1	1	0	ANI6	Setting prohibited	ANI0 to ANI6	4 + ANI4 to ANI6
1	1	1	ANI7	Setting prohibited	ANI0 to ANI7	4 + ANI4 to ANI7

Note The analog input pins that can be specified in the timer/external trigger mode of the select mode are the ANI0 to ANI3 pins. If the ANIS2 bit is set to 1, therefore, the A/D conversion operation is not performed.

(2) A/D converter mode register 1 (ADM1)

The ADM1 register is an 8-bit register that specifies the conversion operation time and trigger mode.

This register can be read or written in 8-bit units. However, when data is written to the ADM1 register during an A/D conversion operation, the conversion operation is initialized and conversion is executed from the beginning.

Reset input sets this register to 07H.

Cautions 1. It takes the following number of clocks from trigger input to the start of A/D conversion.

In software trigger mode: 9 to 11 clocks
In timer/external trigger mode: 5 to 7 clocks

2. Input the trigger at an interval longer than the minimum trigger interval indicated in 14.8.5 A/D conversion time.

After reset: 07H R/W Address: FFFFF201H

	7	6	5	4	3	2	1	0
ADM1	0	TRG2	TRG1	0	0	FR2	FR1	FR0

TRG2	TRG1	Trigger mode
0	0	Software trigger mode
0	1	Timer trigger mode[Note 1]
1	0	Setting prohibited
1	1	External trigger mode[Note 2]

Notes 1. The trigger is specified by the ADTS register in the timer trigger mode.

2. The valid edge of the trigger (ADTRG pin) is specified by the INTR3.INTR37 and INTF3.INTF37 bits in the external trigger mode (see **20.4.2 (5) External interrupt rising edge specification register 3 (INTR3), external interrupt falling edge specification register 3 (INTF3)**). Do not set the INTR37 and INTF37 bits to detect a level (low level detection).

Cautions 1. For FR2 to FR0 bits, see Table 14-3 Conversion Operation Time.

2. Be sure to set bits 3, 4, and 7 to "0".

Table 14-3. Conversion Operation Time

FR2	FR1	FR0	Number of Conversion Clocks	Conversion Operation Time[Note]			
				f_{XX} = 80 MHz	f_{XX} = 66.7 MHz	f_{XX} = 50 MHz	f_{XX} = 10 MHz
0	0	0	52	Setting prohibited	Setting prohibited	Setting prohibited	5.20 μs
0	0	1	104	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0	1	0	156	Setting prohibited	2.34 μs	3.12 μs	Setting prohibited
0	1	1	208	2.60 μs	3.12 μs	4.16 μs	Setting prohibited
1	0	0	260	3.25 μs	3.90 μs	5.20 μs	Setting prohibited
1	0	1	312	3.90 μs	4.68 μs	Setting prohibited	Setting prohibited
1	1	0	364	4.55 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	1	416	5.20 μs	Setting prohibited	Setting prohibited	Setting prohibited

Note Figures under the conversion operation time parameter are target values. Set the conversion operation time in the range of 2.34 to 5.20 μs.

The A/D converter cannot be used if f_{XX} < 10 MHz.

Remark f_{XX}: Internal system clock

(3) A/D converter mode register 2 (ADM2)

The ADM2 register is an 8-bit register that controls the reset and clock of the A/D converter.

This register can be read or written in 8-bit or 1-bit units. However, bit 1 is read-only.

Reset input sets this register to 02H.

Cautions 1. When the ADCAE bit is cleared to 0, the ADM0.ADCE and ADM0.ADCS bits are automatically cleared to 0 (the ADCE bit cannot be set if the ADCAE bit is 0).
The other registers are not initialized.

2. The A/D converter enters the reset state after reset release. When operating the A/D converter, be sure to set the ADCAE bit to 1 after setting the ADM0 and ADM1 registers.

After reset: 02H R/W Address: FFFFF202H

	7	6	5	4	3	2	<1>	<0>
ADM2	0	0	0	0	0	0	ADNCS	ADCAE

ADNCS	A/D conversion invalid status
0	A/D conversion enabled
1	A/D conversion disabled

- This bit is set when the A/D converter is not used, in the IDLE/software STOP mode, after the IDLE/software STOP mode has been released, and within the oscillation stabilization time[Note] after the ADCAE bit has been set to 1.
- In the software trigger mode, starting the conversion is postponed even if the ADM0.ADCE bit is set to 1.
- In the timer/external trigger mode, the trigger is ignored even if it is input within the stabilization time.

ADCAE	A/D converter operation control
0	Clock supply to the A/D converter is stopped; the A/D converter is in the reset state
1	The clock is supplied to the A/D converter; A/D converter operation is enabled

Note Time is required to stabilize the analog circuit after supplying the clock to the A/D converter is started. The V850E/MA3 uses hardware to ensure the lapse of the stabilization time. For the stabilization time, see **14.8.6 Stabilization time**.

(4) A/D conversion result registers 0 to 7, 0H to 7H (ADCR0 to ADCR7, ADCR0H to ADCR7H)

The ADCRn register is a 10-bit register holding the A/D conversion results. There are eight 10-bit registers. These registers are read-only, in 16-bit or 8-bit units. To read the A/D conversion result in 16-bit units, specify the ADCRn register, and to read the higher 8 bits of the result, specify the ADCRnH register (n = 0 to 7).

When reading the 10-bit data of the A/D conversion results from the ADCRn register during 16-bit access, only the lower 10 bits are valid and the higher 6 bits are always 0.

Reset input clears the ADCRn register to 0000H, and the ADCRnH register to 00H.

After reset: 0000H R Address: ADCR0 FFFFF210H, ADCR1 FFFFF212H, ADCR2 FFFFF214H, ADCR3 FFFFF216H, ADCR4 FFFFF218H, ADCR5 FFFFF21AH, ADCR6 FFFFF21CH, ADCR7 FFFFF21EH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCRn (n = 0 to 7)	0	0	0	0	0	0	AD n9	AD n8	AD n7	AD n6	AD n5	AD n4	AD n3	AD n2	AD n1	AD n0

After reset: 00H R Address: ADCR0H FFFFF220H, ADCR1H FFFFF221H, ADCR2H FFFFF222H, ADCR3H FFFFF223H, ADCR4H FFFFF224H, ADCR5H FFFFF225H, ADCR6H FFFFF226H, ADCR7H FFFFF227H

	7	6	5	4	3	2	1	0
ADCRnH (n = 0 to 7)	ADn9	ADn8	ADn7	ADn6	ADn5	ADn4	ADn3	ADn2

The correspondence between each analog input pin and the ADCRn register (except in the 4-buffer mode) is shown below.

Analog Input Pin	ADCRn Register
ANI0	ADCR0, ADCR0H
ANI1	ADCR1, ADCR1H
ANI2	ADCR2, ADCR2H
ANI3	ADCR3, ADCR3H
ANI4	ADCR4, ADCR4H
ANI5	ADCR5, ADCR5H
ANI6	ADCR6, ADCR6H
ANI7	ADCR7, ADCR7H

The relationship between the analog voltage input to the analog input pins (ANI0 to ANI7) and the A/D conversion result (of A/D conversion result register n (ADCRn)) is as follows:

$$ADCR = INT\left(\frac{V_{IN}}{AV_{DD0}} \times 1{,}024 + 0.5\right)$$

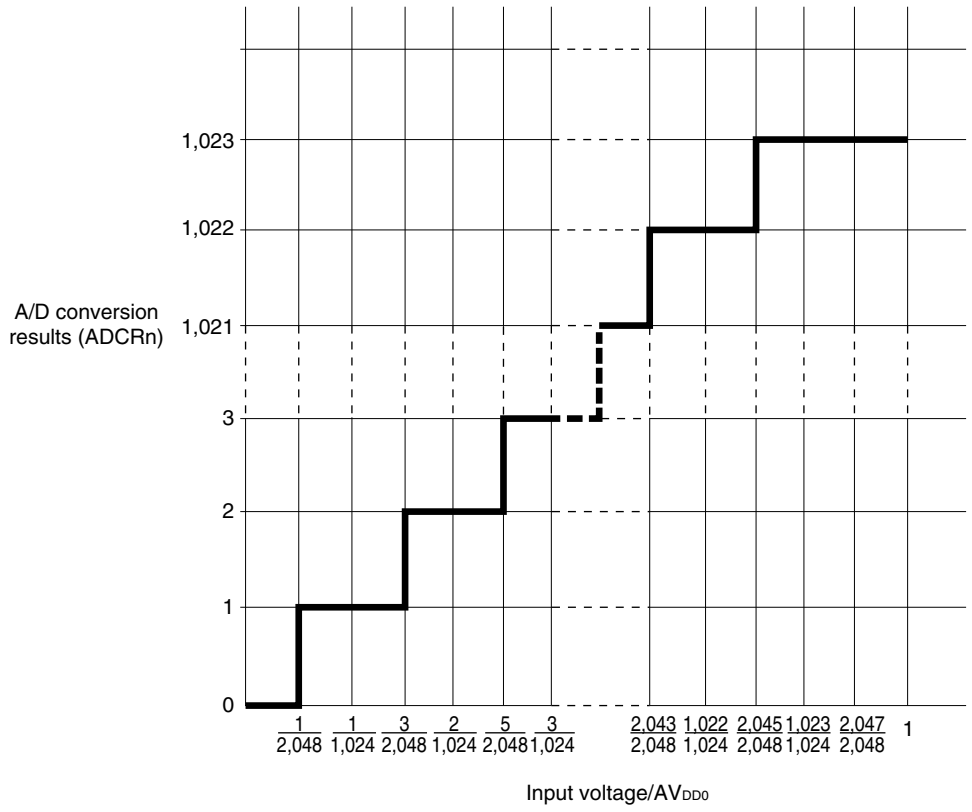
or,

$$(ADCR - 0.5) \times \frac{AV_{DD0}}{1{,}024} \leq V_{IN} < (ADCR + 0.5) \times \frac{AV_{DD0}}{1{,}024}$$

INT( ): Function that returns the integer of the value in ( )
V_{IN}: Analog input voltage
AV_{DD0}: AV_{DD0} pin voltage
ADCR: Value of A/D conversion result register n (ADCRn)

The relationship between the analog input voltage and the A/D conversion results is shown below.

Figure 14-2. Relationship Between Analog Input Voltage and A/D Conversion Results



Remark n = 0 to 7

(5) A/D trigger select register (ADTS)

The ADTS register selects the timer trigger input.

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

Reset input sets this register to 01H.

After reset: 01H R/W Address: FFFFF288H

	7	6	5	4	3	<2>	<1>	<0>
ADTS	0	0	0	0	0	TMS2	TMS1	TMS0

TMS2	Timer trigger signal (INTCMD0) enable
0	Do not select INTCMD0 signal as timer trigger input.
1	Select INTCMD0 signal as timer trigger input.

TMS1	Timer trigger signal (INTCCQ0) enable
0	Do not select INTCCQ0 signal as timer trigger input.
1	Select INTCCQ0 signal as timer trigger input.

TMS0	Timer trigger signal (TQTADT[Note]) enable
0	Do not select TQTADT signal[Note] as timer trigger input.
1	Select TQTADT signal[Note] as timer trigger input.

Note The TQTADT signal is set by the TQ0OPT2.TQ0AT0 to TQ0OPT2.TQ0AT3 bits.

Cautions 1. Do not write to the ADTS register during an A/D conversion operation.

2. Set the TMS0 bit to 1 when selecting the interrupt request signal (TQTADT) of the motor control function as the timer trigger input. The following interrupt request signals of the motor control function can be selected as the timer trigger input (two or more interrupt signals can be selected).

- **Timer Q0 (in 6-phase PWM output mode (including when the tuning operation with timer P2 is not performed))**
  - **TQ0AT0 bit = 1: INTOVQ signal (match interrupt of the 16-bit counter value and 0001H during down-counting)**
  - **TQ0AT1 bit = 1: INTCCQ0 signal (match interrupt of the 16-bit counter value and the TQ0CCR0 register value during up-counting)**
- **Timer P2 (only when the tuning operation with timer Q0 is performed in the 6-phase PWM output mode)**
  - **TQ0AT2 bit = 1: INTCCP20 signal (when the value of timer P2 during the tuning operation matches the set value of TP2CCR0 register value)**
  - **TQ0AT3 bit = 1: INTCCP21 signal (when the value of timer P2 during the tuning operation matches the set value of TP2CCR1 register value)**

14.4 Operation

14.4.1 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 ADM0 and ADM1 registers[Note 1]. The setting of the stabilization time is determined by the specification of the ADM1.FR0 to ADM1.FR2 bits.

(2) Set the ADM2.ADCAE bit to 1[Note 2]. When the ADCAE bit is set from 0 to 1, counting the stabilization time is started.

(3) In the software trigger mode, setting the ADM0.ADCE bit to 1 starts A/D conversion after the lapse of the stabilization wait time. If the ADCE bit is set to 1 in the timer/external trigger mode, the A/D converter ignores the trigger during the stabilization wait time and waits for the trigger after the lapse of the stabilization wait time[Note 3].

(4) The voltage generated from the voltage tap of the series resistor string is compared with the analog input voltage by the comparator.

(5) 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 the ADCRn register and the conversion result is stored in this register. When A/D conversion has been completed the specified number of times, an A/D conversion end interrupt request signal (INTAD) is generated (n = 0 to 7).

Notes 1. If the ADM0 to ADM2 registers are written during A/D conversion, the conversion result is not stored in the ADCRn register and the conversion operation is performed from the beginning again.
If the ADCAE bit is cleared to 0, clock supply to the A/D converter is stopped, the current A/D conversion operation is initialized, and the ADCE bit is cleared to 0. The ADM0.ADCS bit register is 0 when it is read.

2. If the ADM1 register is set during the stabilization wait time after the ADCAE bit has been set to 1, the stabilization wait time is reset, and stabilization wait time is generated according to the values of the FR0 to FR2 bits.

3. In the timer/external trigger mode, if the ADCE bit is set to 1, the mode changes to the trigger standby state. The A/D conversion operation is started by the trigger signal, and the trigger standby state is returned to when the A/D conversion operation ends.

14.4.2 Operation mode and trigger mode

Various conversion operations can be specified for the A/D converter by specifying the operation mode and trigger mode. The operation mode and trigger mode are set by ADM0 and ADM1 registers.

The following shows the relationship between the operation mode and trigger mode.

Trigger Mode	Operation Mode		Setting Value		Analog Input Pin
			ADM0	ADM1	
Software trigger	Select	1 buffer	xx010xxxB	000x0xxxB	ANI0 to ANI7
		4 buffers	xx110xxxB	000x0xxxB	
	Scan		xxx00xxxB	000x0xxxB	
Timer trigger	Select	1 buffer	xx010xxxB	00100xxxB	ANI0 to ANI3
		4 buffers	xx110xxxB	00100xxxB	
	Scan		xxx00xxxB	00100xxxB	ANI0 to ANI7
External trigger	Select	1 buffer	xx010xxxB	01100xxxB	ANI0 to ANI3
		4 buffers	xx110xxxB	01100xxxB	
	Scan		xxx00xxxB	01100xxxB	ANI0 to ANI7

(1) Trigger mode

There are three types of trigger modes that serve as the start timing of A/D conversion operation: software trigger mode, timer trigger mode, and external trigger mode.

(a) Software trigger mode

In this mode, the analog input pin selected by the ADM0.ANIS0 to ADM0.ANIS2 bits from the ANI0 to ANI7 pins is used for A/D conversion. The conversion is started when the ADM0.ADCE bit is set to 1.

(b) Timer trigger mode

In this mode, the analog input pin selected by the ANIS0 to ANIS2 bits from the ANI0 to ANI7 pins is used for A/D conversion. The conversion is started by one of the timers (timer Q0, timer D0, or the motor control function). The timer interrupt request signal selected by the ADTS register can be selected as a trigger of the A/D conversion operation.

The ANI0 to ANI3 pins can be specified in the select mode and scan mode, but the ANI4 to ANI7 pins cannot be specified in the select mode. In the scan mode, input from the ANI3 pin is converted and conversion of the ANI4 to ANI7 pins is immediately started when the fourth trigger is generated.

When a timer trigger is generated, the analog input conversion timing is generated.

(c) External trigger mode

In this mode, the analog input pin selected by the ANIS0 to ANIS2 bits from the ANI0 to ANI7 pins is used for A/D conversion. The ADTRG pin is used to start the A/D conversion.

The ANI0 to ANI3 pins can be specified in the select mode and scan mode, but the ANI4 to ANI7 pins cannot be specified in the select mode. In the scan mode, input from the ANI3 pin is converted and conversion of the ANI4 to ANI7 pins is immediately started when the fourth trigger is generated.

(2) Operation mode

There are two operation modes that set the ANI0 to ANI7 pins: select mode and scan mode. The select mode has sub-modes that consist of 1-buffer mode and 4-buffer mode. These modes are set by the ADM0 register.

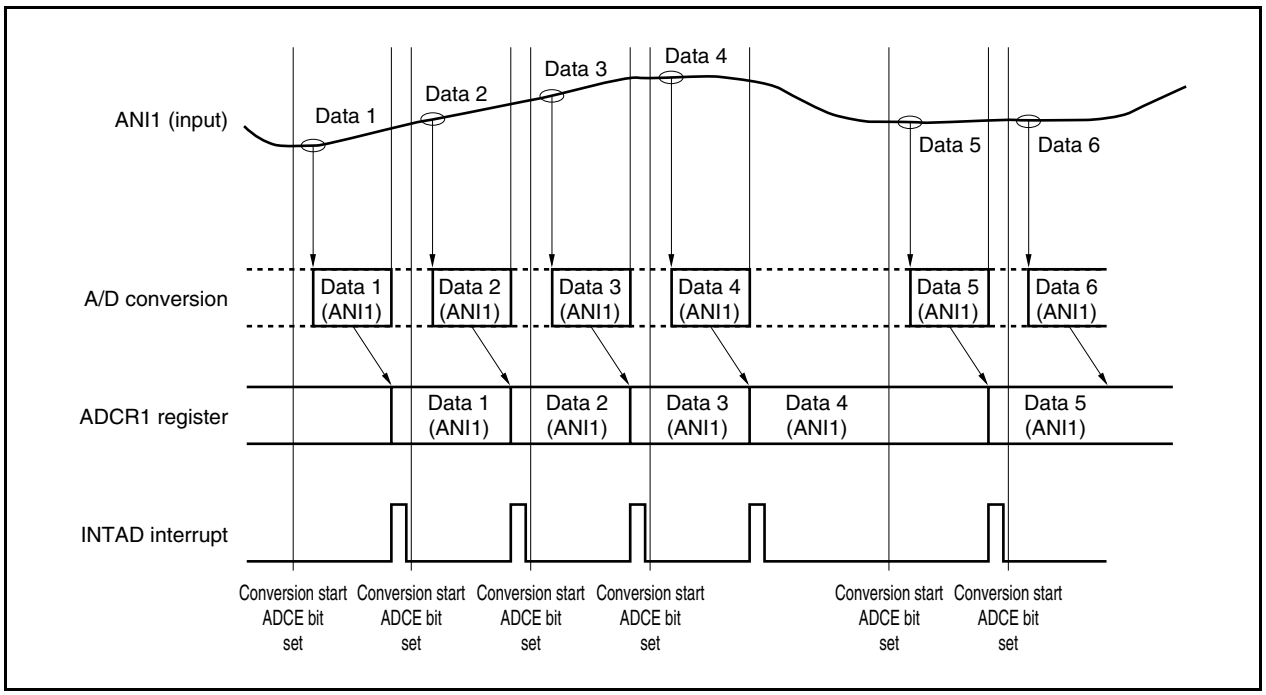
(a) Select mode

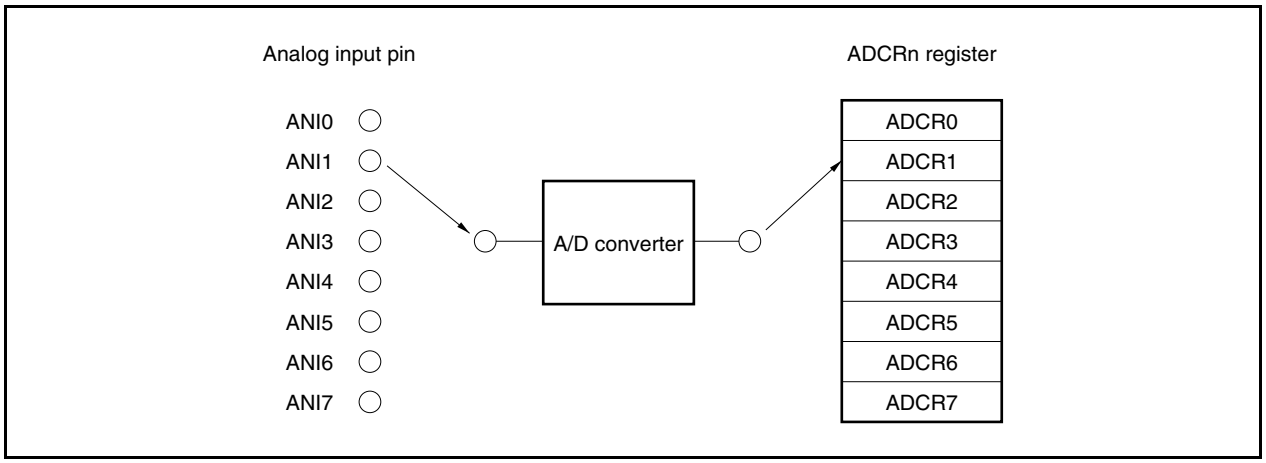
In this mode, one analog input pin voltage specified by the ADM0 register is A/D converted. The conversion results are stored in the ADCRn register corresponding to the analog input pin (ANIn). For this mode, the 1-buffer mode and 4-buffer mode are provided for storing the A/D conversion results (software trigger mode: n = 0 to 7, timer/external trigger mode: n = 0 to 3).

- **1-buffer mode**

  In this mode, one analog input pin voltage specified by the ADM0 register is A/D converted. The conversion results are stored in the ADCRn register corresponding to the analog input pin (ANIn). The ANIn pin and ADCRn register correspond one to one, and an A/D conversion end interrupt request signal (INTAD) is generated each time one A/D conversion ends.

Figure 14-3. Select Mode Operation Timing: 1-Buffer Mode (ANI1)

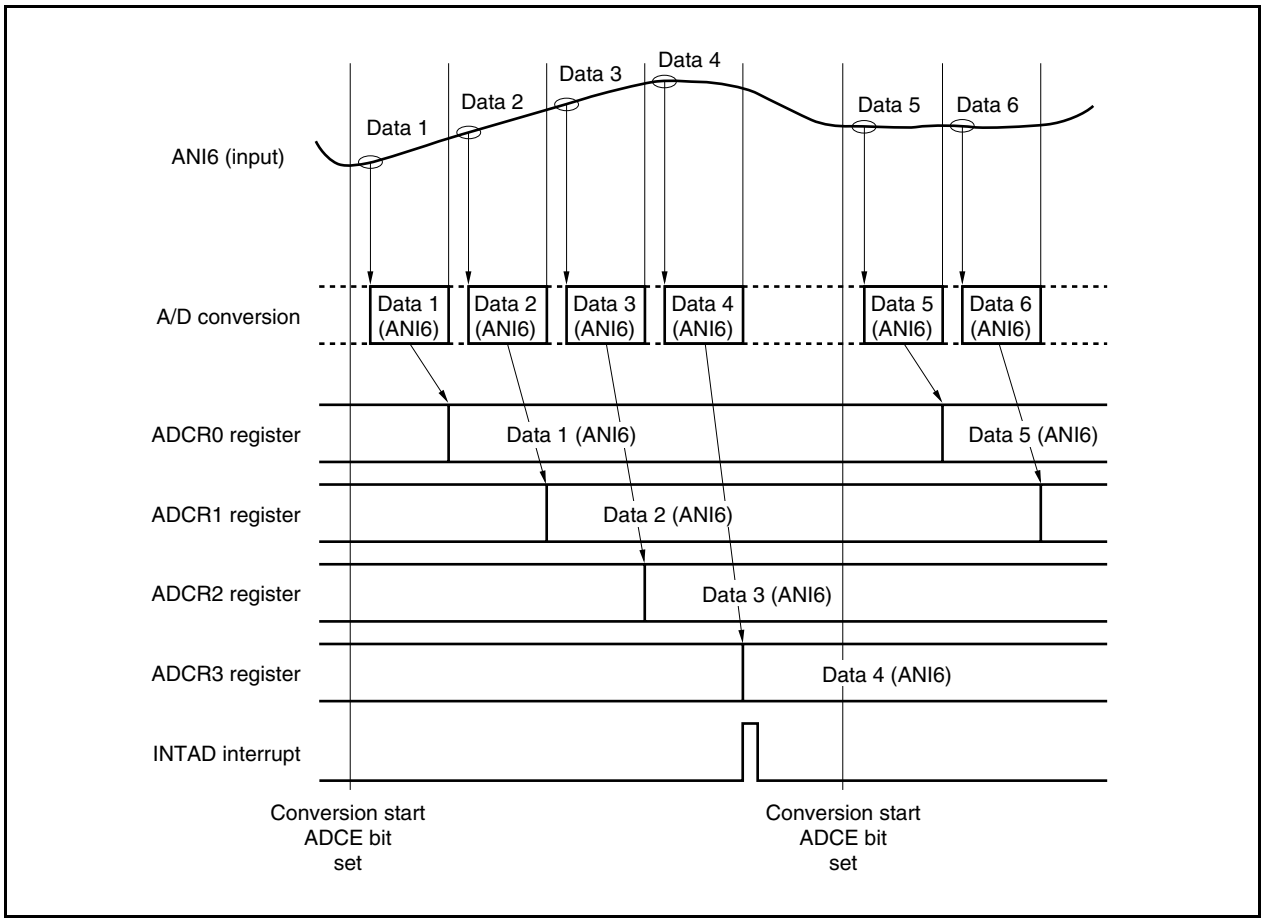


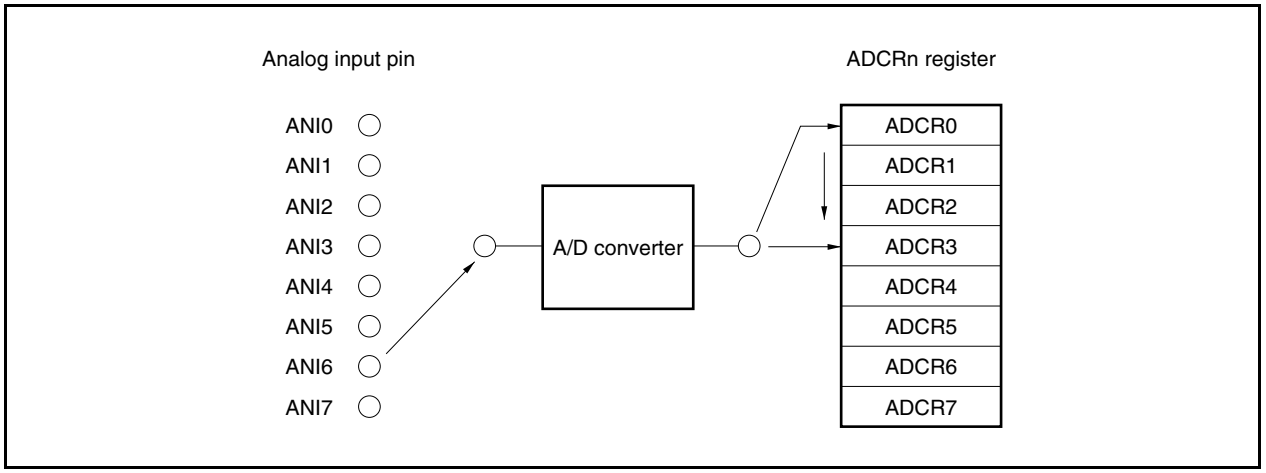


- **4-buffer mode**

  In this mode, one analog input pin voltage is A/D converted four times and the results are stored in the ADCR0 to ADCR3 registers. The A/D conversion end interrupt request signal (INTAD) is generated when the four A/D conversions end.

Figure 14-4. Select Mode Operation Timing: 4-Buffer Mode (ANI6)

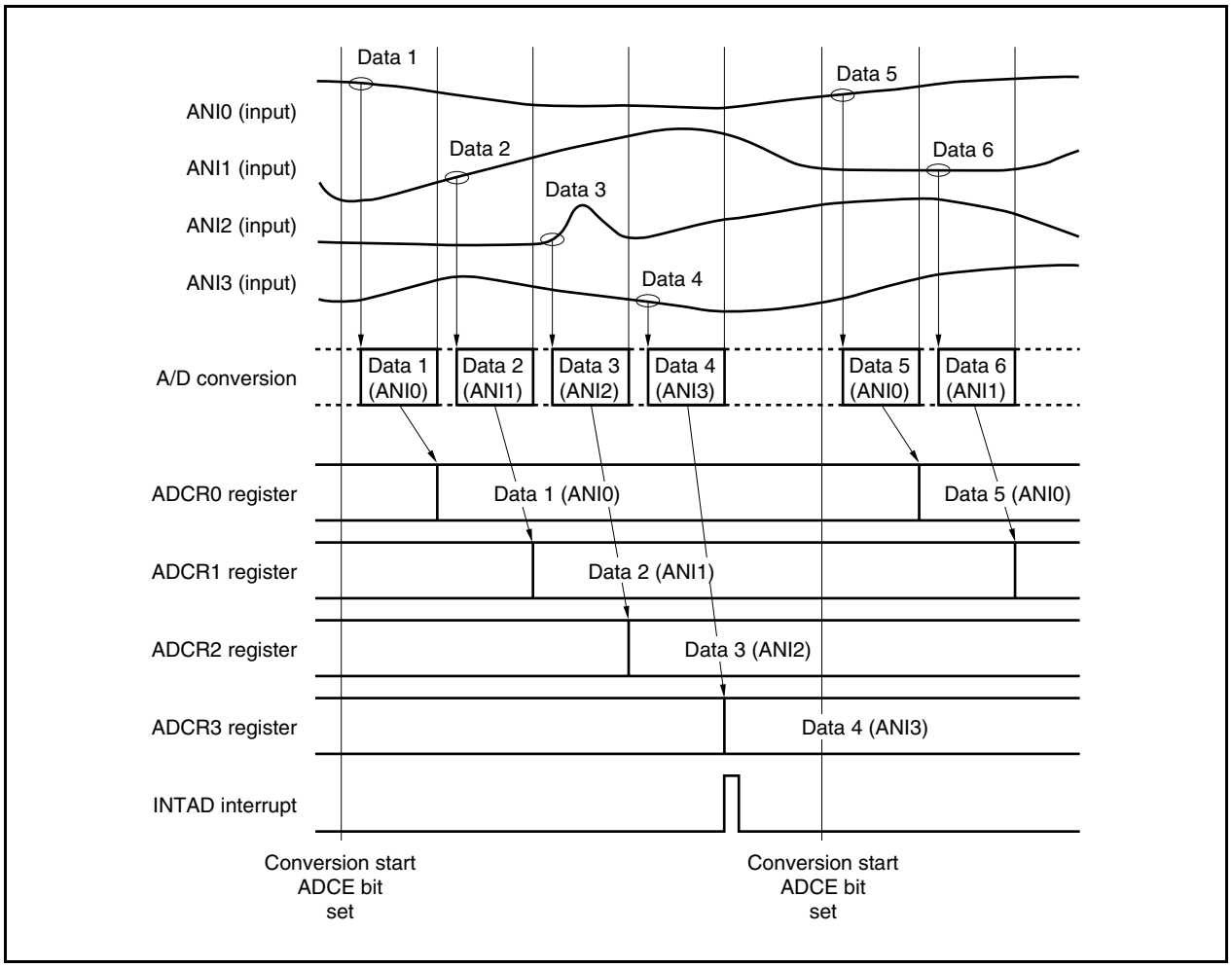


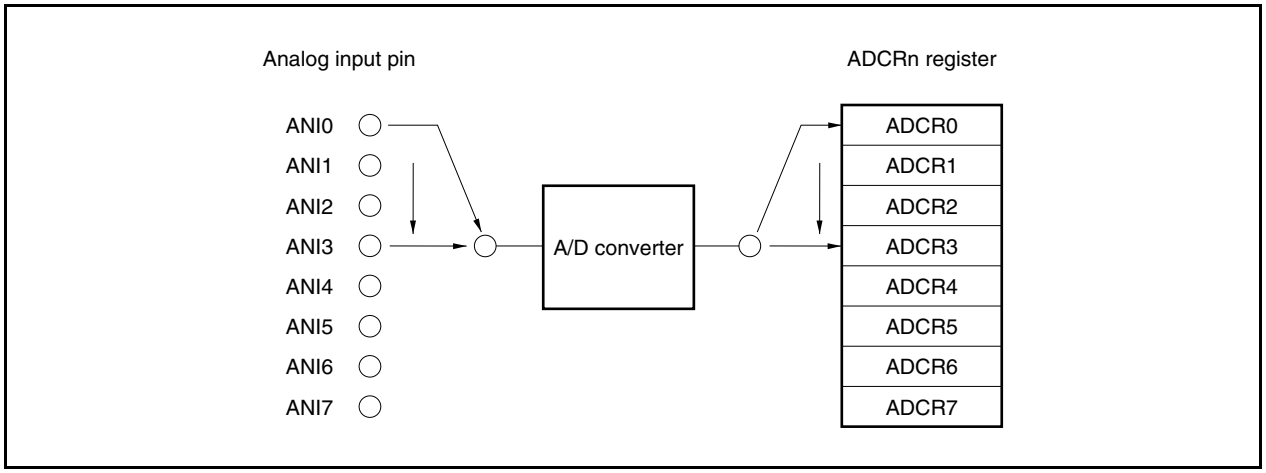


(b) Scan mode

In this mode, the analog input pins (ANIn) specified by the ADM0 register are selected sequentially from the ANI0 pin, and A/D conversion is executed. The A/D conversion results are stored in the ADCRn register corresponding to the analog input pin (ANIn) (n = 0 to 7). When the conversion of the specified analog input pin ends, the A/D conversion end interrupt request signal (INTAD) is generated.

Figure 14-5. Scan Mode Operation Timing: 4-Channel Scan (ANI0 to ANI3)





14.5 Operation in Software Trigger Mode

When the ADM0.ADCE bit is set (1), A/D conversion is started.

14.5.1 Select mode operation

In this mode, the analog input pin voltage specified by the ADM0 register is A/D converted. The conversion results are stored in A/D conversion result register n (ADCRn) corresponding to the analog input. In the select mode, the 1-buffer mode and 4-buffer mode are supported according to the method of storing the A/D conversion results (n = 0 to 7).

(1) 1-buffer mode (software trigger select: 1 buffer)

In this mode, one analog input pin voltage is A/D converted once. The conversion results are stored in one ADCRn register. The analog input pin (ANIn) and ADCRn register correspond one to one.

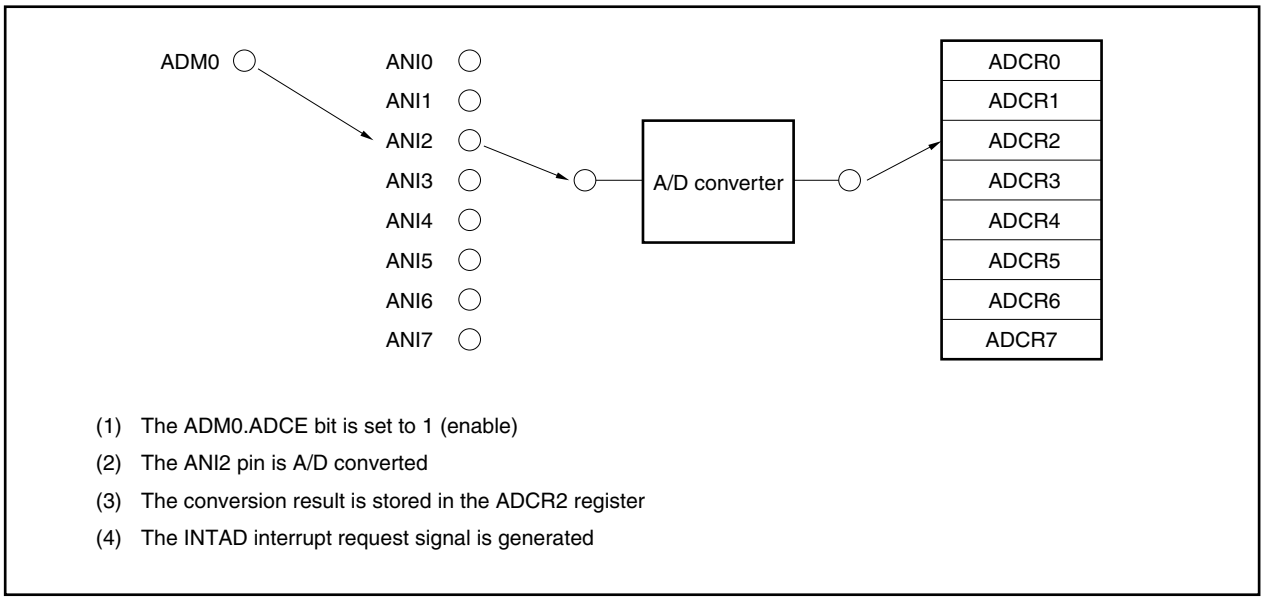
Each time an A/D conversion is executed, an A/D conversion end interrupt request signal (INTAD) is generated and A/D conversion ends.

Analog Input Pin	A/D Conversion Result Register
ANIn	ADCRn

If the ADM0.ADCE bit is set (1), A/D conversion can be restarted.

This mode is most appropriate for applications in which the results of each first-time A/D conversion are read.

Figure 14-6. Example of 1-Buffer Mode Operation (Software Trigger Select: 1 Buffer)



(1) The ADM0.ADCE bit is set to 1 (enable)
(2) The ANI2 pin is A/D converted
(3) The conversion result is stored in the ADCR2 register
(4) The INTAD interrupt request signal is generated

(2) 4-buffer mode (software trigger select: 4 buffers)

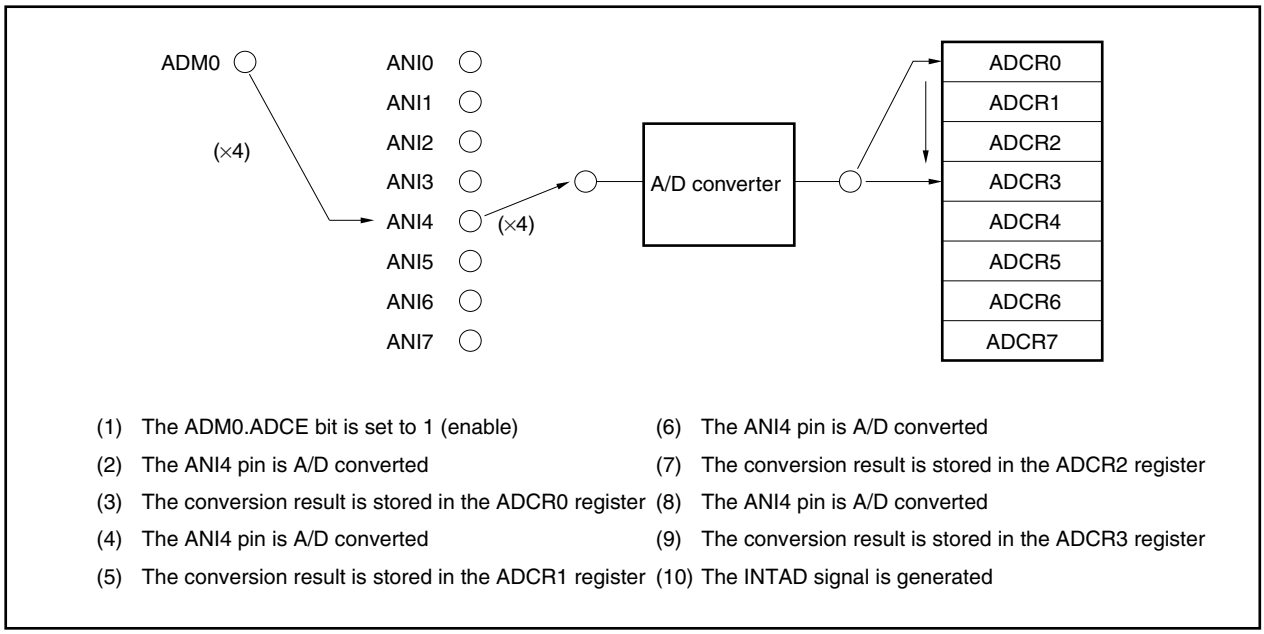
In this mode, one analog input pin voltage is A/D converted four times and the results are stored in the ADCR0 to ADCR3 registers. When the 4th A/D conversion ends, an A/D conversion end interrupt request signal (INTAD) is generated and the A/D conversion is stopped.

Analog Input Pin	A/D Conversion Result Register
ANIn	ADCR0
ANIn	ADCR1
ANIn	ADCR2
ANIn	ADCR3

If the ADM0.ADCE 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.

Figure 14-7. Example of 4-Buffer Mode Operation (Software Trigger Select: 4 Buffers)



(1) The ADM0.ADCE bit is set to 1 (enable)
(2) The ANI4 pin is A/D converted
(3) The conversion result is stored in the ADCR0 register
(4) The ANI4 pin is A/D converted
(5) The conversion result is stored in the ADCR1 register
(6) The ANI4 pin is A/D converted
(7) The conversion result is stored in the ADCR2 register
(8) The ANI4 pin is A/D converted
(9) The conversion result is stored in the ADCR3 register
(10) The INTAD signal is generated

14.5.2 Scan mode operations

In this mode, the analog input pins (ANIn) specified by the ADM0 register are selected sequentially from the ANI0 pin, and A/D conversion is executed. The A/D conversion results are stored in the ADCRn register corresponding to the analog input pin (n = 0 to 7).

When conversion of all the specified analog input pin ends, the A/D conversion end interrupt request signal (INTAD) is generated, and A/D conversion is stopped.

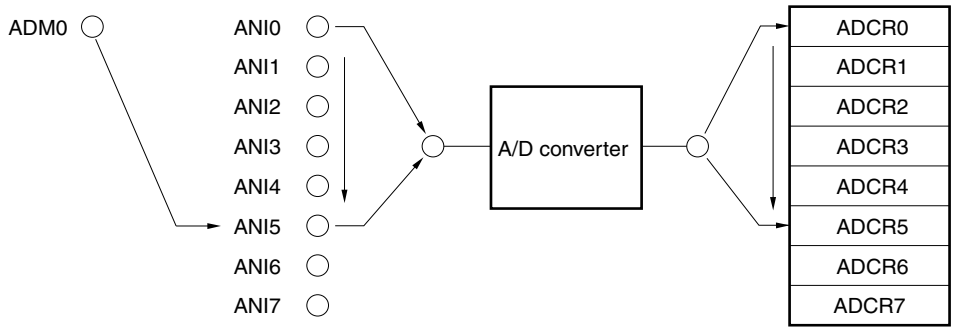
Analog Input Pin	A/D Conversion Result Register
ANI0	ADCR0
⋮	⋮
ANIn[Note]	ADCRn

Note Set by the ADM0.ANIS0 to ADM0.ANIS2 bits.

If the ADM0.ADCE bit is set (1), A/D conversion can be restarted.

This mode is most appropriate for applications in which multiple analog inputs are constantly monitored.

Figure 14-8. Example of Scan Mode Operation (Software Trigger Scan)



(1) The ADM0.ADCE bit is set to 1 (enable)
(2) The ANI0 pin is A/D converted
(3) The conversion result is stored in the ADCR0 register
(4) The ANI1 pin is A/D converted
(5) The conversion result is stored in the ADCR1 register
(6) The ANI2 pin is A/D converted
(7) The conversion result is stored in the ADCR2 register
(8) The ANI3 pin is A/D converted
(9) The conversion result is stored in the ADCR3 register
(10) The ANI4 pin is A/D converted
(11) The conversion result is stored in the ADCR4 register
(12) The ANI5 pin is A/D converted
(13) The conversion result is stored in the ADCR5 register
(14) The INTAD signal is generated

14.6 Operation in Timer Trigger Mode

Conversion timing for up to four-channel analog inputs (ANI0 to ANI3) can be set for the A/D converter using the interrupt request signal of the timer selected by the ADTS register.

Timer Q0, timer D0, or the motor control function can be used for the timer to specify the A/D conversion trigger.

The following two modes are provided according to the value set in the timer.

(1) One-shot mode

To use the one-shot mode, set the ADTS.TMS1 bit to 1, and set timer Q0 in the one-shot pulse mode.

When the 16-bit counter value matches the set value of the TQ0CCR0 register, an interrupt request signal (INTCCQ0) is generated and the 16-bit counter is cleared to 0000H and stopped. After that, because timer Q0 does not generate the INTCCQ0 signal, the A/D trigger signal is not generated and the A/D converter enters the conversion standby status.

When timer Q0 is restarted, the A/D trigger signal is generated.

The one-shot mode is used if the A/D conversion cycle is longer than the timer Q0 cycle.

(2) Loop mode

The loop mode is used as follows.

- When timer Q0 is used in the interval timer mode (when the ADTS.TMS1 bit is set to 1)
- Crest interrupt/valley interrupt of motor control function
- When timer P2 that operates in tune with the motor control function is used in the free-running mode
- Timer D0

In this mode, the timer repeatedly outputs an interrupt request signal and A/D conversion is also performed repeatedly.

Depending on the setting of the TQ0OPT2.TQ0AT0 to TQ0OPT2.TQ0AT3 bits of the motor control function, two or more timer interrupt request signals can be used as a trigger of A/D conversion. At this time, the generation interval of the timer interrupt request signal must not be shorter than the minimum trigger interval shown in **Table 14-5 Conversion Time in Timer Trigger Mode and External Trigger Mode**. The interrupt request signal is ignored even if it is generated at interval shorter than the minimum trigger interval.

14.6.1 Select mode operation

In this mode, the analog input pin voltage specified by the ADM0 register is A/D converted. The conversion results are stored in ADCRn register. In the select mode, the 1-buffer mode and 4-buffer mode are provided according to the method of storing the A/D conversion results (n = 0 to 3).

(1) 1-buffer mode operation (timer trigger select: 1 buffer)

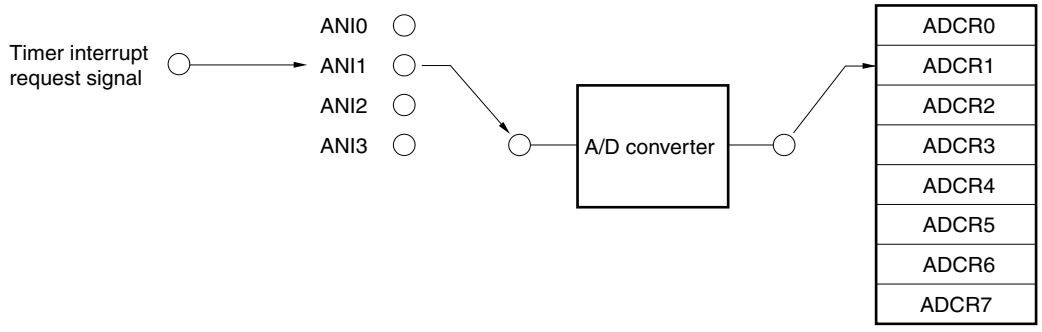
In this mode, one analog input pin voltage is A/D converted once using the signal from the timer as a trigger, and the results are stored in one ADCRn register. An A/D conversion end interrupt request signal (INTAD) is generated for each A/D conversion and A/D conversion is stopped (n = 0 to 3).

Trigger	Analog Input Pin	A/D Conversion Result Register
Timer trigger signal	ANIn	ADCRn

In one-shot mode, A/D conversion stops after one conversion. To restart A/D conversion, set (1) the TQ0CTL0.TQ0CE bit to restart timer Q0.

When set to the loop mode, unless the ADM0.ADCE bit is cleared (0) or the ADM2.ADCAE bit is cleared (0), A/D conversion is repeated each time a timer interrupt request signal is generated.

Figure 14-9. Example of 1-Trigger Mode Operation (Timer Trigger Select: 1 Buffer)



(1) The ADM0.ADCE bit is set to 1 (enable)
(2) The timer interrupt request signal is generated
(3) The ANI1 pin is A/D converted
(4) The conversion result is stored in the ADCR1 register
(5) The INTAD signal is generated

(2) 4-buffer mode operation (timer trigger select: 4 buffers)

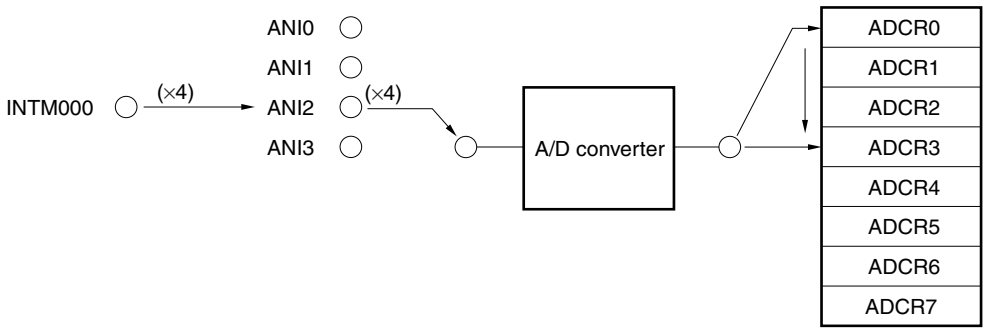
In this mode, one analog input pin voltage is A/D converted four times using the timer interrupt request signal as a trigger, and the results are stored in the ADCR0 to ADCR3 registers. The A/D conversion end interrupt request signal (INTAD) is generated when the four A/D conversions end and A/D conversion is stopped.

This mode is suitable for applications in which the average of the A/D conversion results is calculated.

Trigger	Analog Input Pin	A/D Conversion Result Register
Timer interrupt request signal	ANIn	ADCR0
Timer interrupt request signal	ANIn	ADCR1
Timer interrupt request signal	ANIn	ADCR2
Timer interrupt request signal	ANIn	ADCR3

If the one-shot mode is set and the timer interrupt request signal (INTCCQ0) is generated less than four times, the INTAD signal is not generated and the standby state is set.

Figure 14-10. Example of 1-Trigger Mode Operation (Timer Trigger Select: 4 Buffers)



(1) The ADM0.ADCE bit is set to 1 (enable)
(2) The timer interrupt request signal is generated
(3) The ANI2 pin is A/D converted
(4) The conversion result is stored in the ADCR0 register
(5) The timer interrupt request signal is generated
(6) The ANI2 pin is A/D converted
(7) The conversion result is stored in the ADCR1 register
(8) The timer interrupt request signal is generated
(9) The ANI2 pin is A/D converted
(10) The conversion result is stored in the ADCR2 register
(11) The timer interrupt request signal is generated
(12) The ANI2 pin is A/D converted
(13) The conversion result is stored in the ADCR3 register
(14) The INTAD signal is generated

14.6.2 Scan mode operation

In this mode, the analog input pins (ANIn) specified by the ADM0 register are selected sequentially from the ANI0 pin using the timer interrupt request signal as a trigger and A/D conversion is performed. The A/D conversion results are stored in the ADCRn register corresponding to the analog input pin (n = 0 to 7).

In the conversion operation, first the lower 4 analog input pin channels (ANI0 to ANI3) are A/D converted the specified number of times. If the lower 4 analog input pin channels (ANI0 to ANI3) are set by the ADM0 register so that they are scanned, and when the set number of A/D conversions ends, the A/D conversion end interrupt request signal (INTAD) is generated and A/D conversion is stopped.

When the higher 4 channels (ANI4 to ANI7) of the analog input pins are set by the ADM0 register so that they are scanned, after the conversion of the lower 4 channels has ended, the mode is shifted to the software trigger mode, and the remaining A/D conversions are executed.

The conversion results are stored in the ADCRn register corresponding to the analog input pin.

When conversion of all the specified analog input pins has ended, the INTAD signal is generated and A/D conversion is stopped (n = 0 to 7).

Trigger	Analog Input Pin	A/D Conversion Result Register
Timer interrupt request signal	ANI0	ADCR0
Timer interrupt request signal	ANI1	ADCR1
Timer interrupt request signal	ANI2	ADCR2
Timer interrupt request signal	ANI3	ADCR3
(Software trigger mode)	ANI4	ADCR4
	ANI5	ADCR5
	ANI6	ADCR6
	ANI7	ADCR7

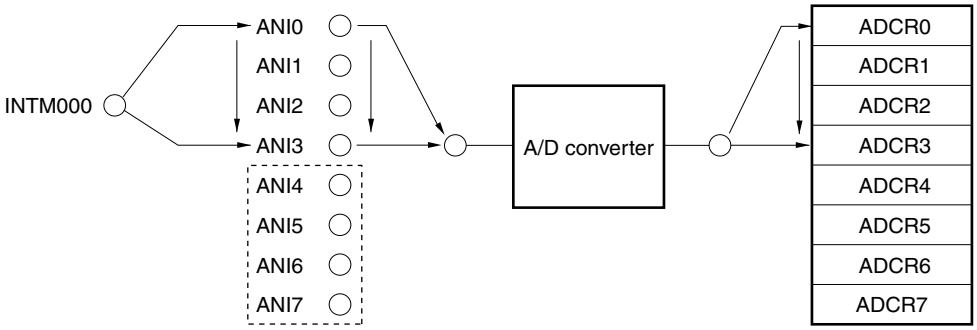
When the timer interrupt request signal is generated while the ADM0.ADCE bit is 1, A/D conversion is restarted.

In one-shot mode, and when less than the specified number of timer interrupt request signals are generated, if the ADCE bit is set to 1, the INTAD signal is not generated and the standby state is set.

The timer interrupt request signal is ignored even if it is generated when the software trigger mode is set. This mode is suitable for applications in which two or more analog input pins are constantly monitored.

Figure 14-11. Example of Timer Trigger Scan Operation

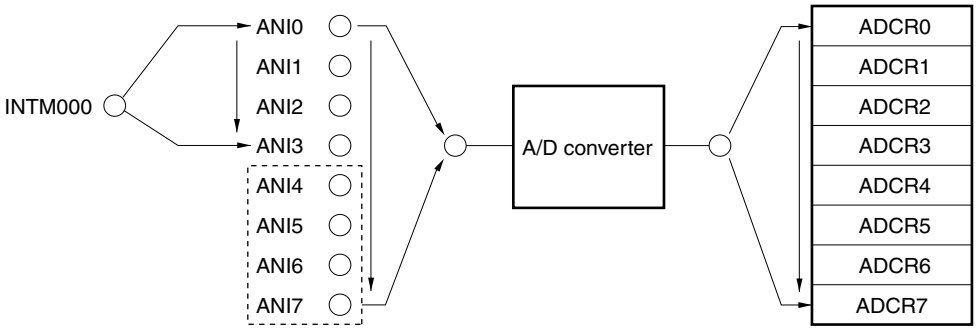
(a) Setting when scanning ANI0 to ANI3 pins



(1) The ADM0.ADCE bit is set to 1 (enable)
(2) The timer interrupt request signal is generated
(3) The ANI0 pin is A/D converted
(4) The conversion result is stored in the ADCR0 register
(5) The timer interrupt request signal is generated
(6) The ANI1 pin is A/D converted
(7) The conversion result is stored in the ADCR1 register
(8) The timer interrupt request signal is generated
(9) The ANI2 pin is A/D converted
(10) The conversion result is stored in the ADCR2 register
(11) The timer interrupt request signal is generated
(12) The ANI3 pin is A/D converted
(13) The conversion result is stored in the ADCR3 register
(14) The INTAD signal is generated

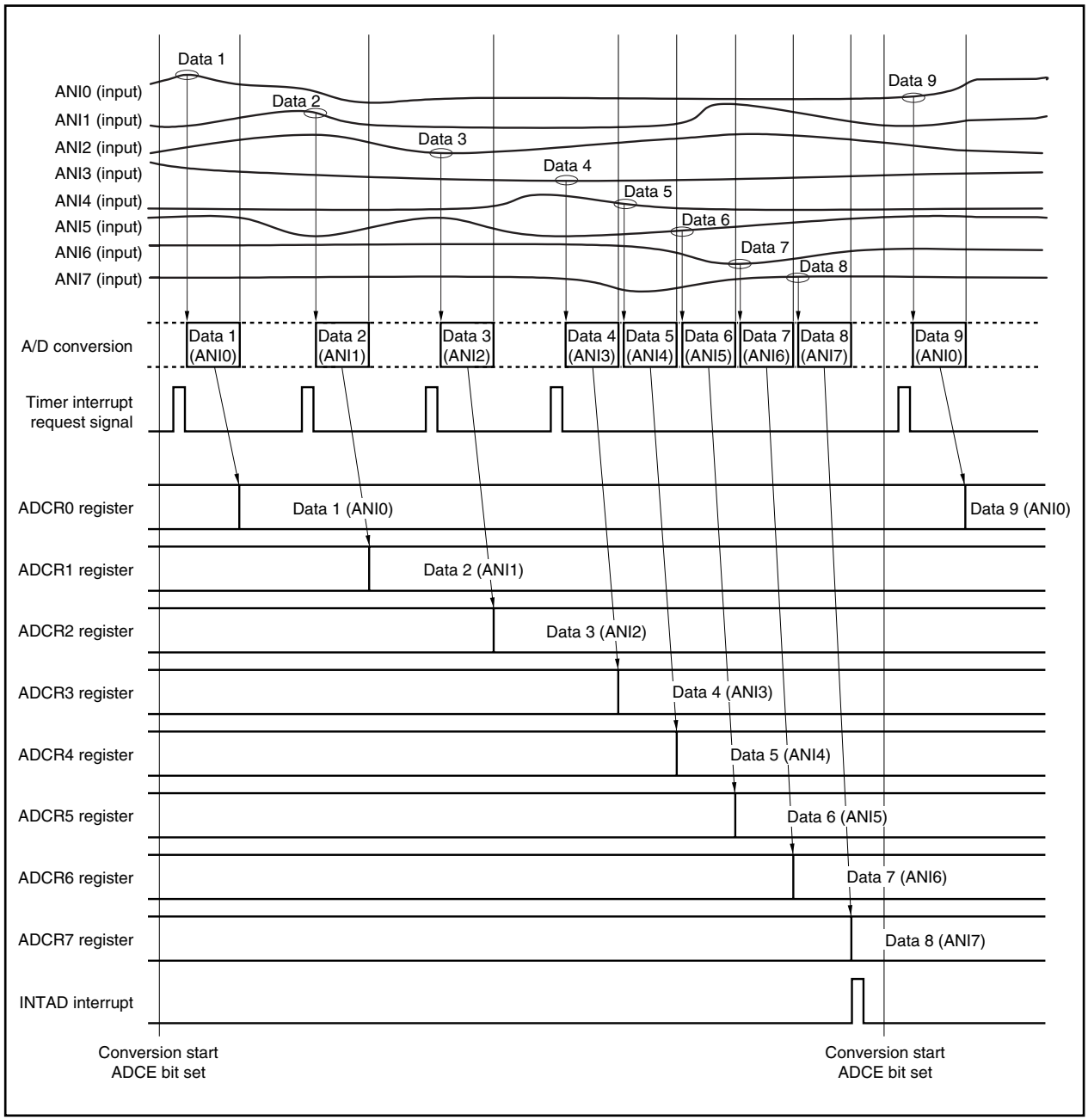
Caution The timer interrupt request signal cannot be used as a trigger for the analog input pins enclosed in the broken lines. When a setting is made to scan the ANI0 to ANI7 pins, the ANI4 to ANI7 pins are converted in software trigger mode (see (b) below).

(b) Setting when scanning ANI0 to ANI7 pins



(1) to (13) Same as (a)
(14) The ANI4 pin is A/D converted
(15) The conversion result is stored in the ADCR4 register
(16) The ANI5 pin is A/D converted
(17) The conversion result is stored in the ADCR5 register
(18) The ANI6 pin is A/D converted
(19) The conversion result is stored in the ADCR6 register
(20) The ANI7 pin is A/D converted
(21) The conversion result is stored in the ADCR7 register
(22) The INTAD signal is generated

Figure 14-12. Timer Trigger Scan Operation Timing: 8-Channel Scan (ANI0 to ANI7)



14.7 Operation in External Trigger Mode

In the external trigger mode, the analog input pins (ANI0 to ANI7) are A/D converted at the ADTRG pin input timing.

The ADTRG pin has an alternate function as the P37/$\overline{\text{INTP137}}$ pin. To set the external trigger mode, set the PMC3.PMC37 bit to 1 and the ADM1.TRG2 and ADM1.TRG1 bits of to 11.

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 using the INTR3.INTR37 bit and INTF3.INTF37 bit (see **20.4.2 (5) External interrupt rising edge specification register 3 (INTR3), external interrupt falling edge specification register 3 (INTF3)**). Do not specify level detection when using the ADTRG pin.

14.7.1 Select mode operations

In this mode, one analog input pin (ANI0 to ANI3) voltage specified by the ADM0 register is A/D converted. The conversion results are stored in the ADCRn register corresponding to the analog input. There are two select modes: 1-buffer mode and 4-buffer mode, according to the method of storing the A/D conversion results (n = 0 to 3).

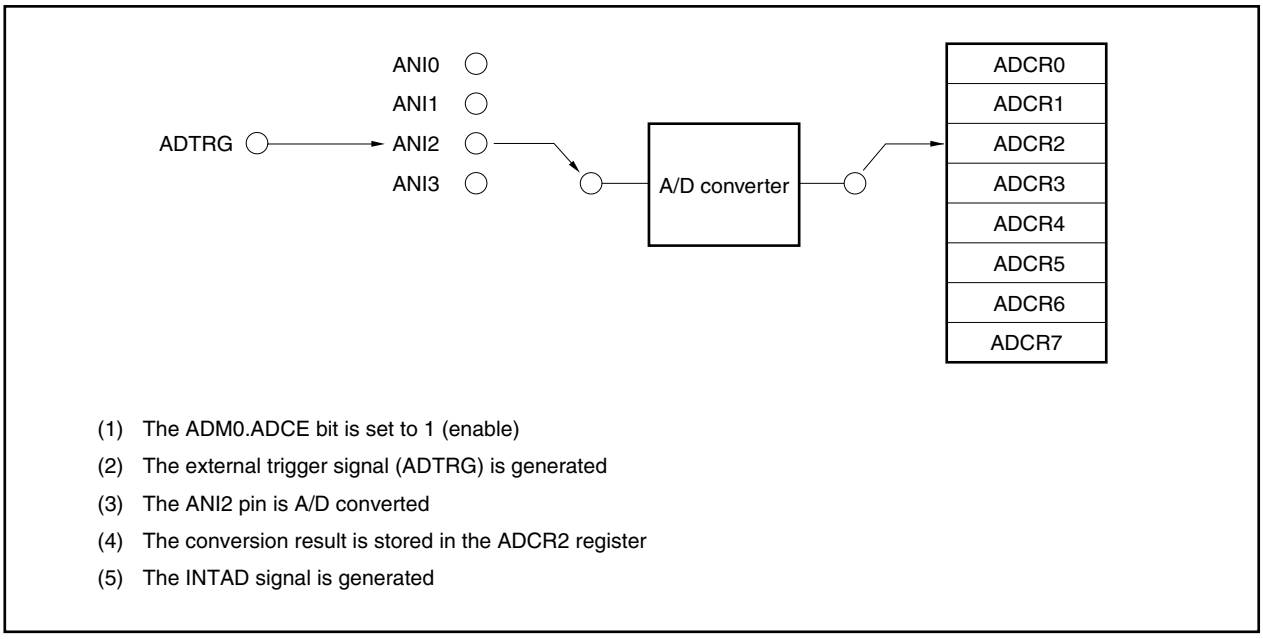
(1) 1-buffer mode (external trigger select: 1-buffer)

In this mode, one analog input pin voltage is A/D converted using the ADTRG signal as a trigger. The conversion results are stored in one ADCRn register. The analog input pin (ANIn) and the ADCRn register correspond one to one. The A/D conversion end interrupt request signal (INTAD) is generated for each A/D conversion, and A/D conversion is stopped.

Trigger	Analog Input Pin	A/D Conversion Result Register
ADTRG signal	ANIn	ADCRn

While the ADM0.ADCE bit is 1, A/D conversion is repeated every time a trigger is input from the ADTRG pin. This mode is most appropriate for applications in which the results are read after each A/D conversion.

Figure 14-13. Example of 1-Buffer Mode Operation (External Trigger Select: 1 Buffer)



(1) The ADM0.ADCE bit is set to 1 (enable)
(2) The external trigger signal (ADTRG) is generated
(3) The ANI2 pin is A/D converted
(4) The conversion result is stored in the ADCR2 register
(5) The INTAD signal is generated

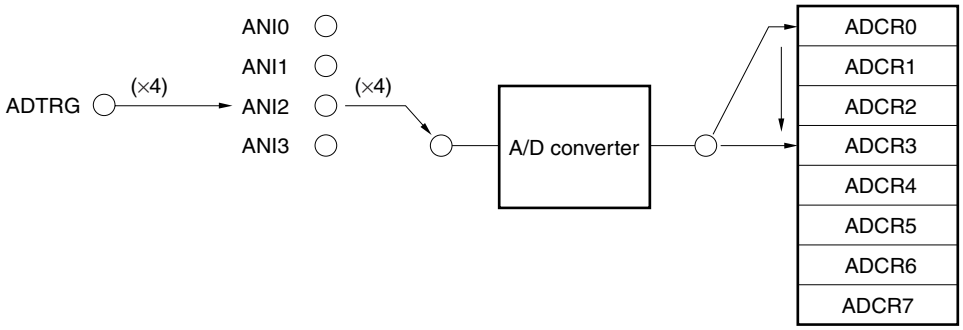
(2) 4-buffer mode (external trigger select: 4 buffers)

In this mode, one analog input pin voltage is A/D converted four times using the ADTRG signal as a trigger and the results are stored in the ADCR0 to ADCR3 registers. The A/D conversion end interrupt request signal (INTAD) is generated and A/D conversion is stopped after the 4th A/D conversion.

Trigger	Analog Input Pin	A/D Conversion Result Register
ADTRG signal	ANIn	ADCR0
ADTRG signal	ANIn	ADCR1
ADTRG signal	ANIn	ADCR2
ADTRG signal	ANIn	ADCR3

While the ADM0.ADCE bit is 1, A/D conversion is repeated every time a trigger is input from the ADTRG pin. This mode is suitable for applications in which the average of the A/D conversion results is calculated.

Figure 14-14. Example of 4-Buffer Mode Operation (External Trigger Select: 4 Buffers)



(1) The ADM0.ADCE bit is set to 1 (enable)
(2) The external trigger signal (ADTRG) is generated
(3) The ANI2 pin is A/D converted
(4) The conversion result is stored in the ADCR0 register
(5) The ADTRG signal is generated
(6) The ANI2 pin is A/D converted
(7) The conversion result is stored in the ADCR1 register
(8) The ADTRG signal is generated
(9) The ANI2 pin is A/D converted
(10) The conversion result is stored in the ADCR2 register
(11) The ADTRG signal is generated
(12) The ANI2 pin is A/D converted
(13) The conversion result is stored in the ADCR3 register
(14) The INTAD signal is generated

14.7.2 Scan mode operation

In this mode, the analog input pins specified by A/D converter mode register 0 (ADM0) are selected sequentially from the ANI0 pin using the ADTRG signal as a trigger, and A/D converted. The A/D conversion results are stored in A/D conversion result register n (ADCRn) corresponding to the analog input pin (n = 0 to 7).

When the lower 4 analog input pin channels (ANI0 to ANI3) are set by the ADM0 register so that they are scanned, the A/D conversion end interrupt request signal (INTAD) is generated when the specified number of A/D conversions have ended, and A/D conversion is stopped.

When the higher 4 analog input pin channels (ANI4 to ANI7) are set by the ADM0 register so that they are scanned, after the conversion of the lower 4 channels is ended, the mode is shifted to the software trigger mode, and the remaining A/D conversions are executed. The conversion results are stored in the ADCRn register corresponding to the analog input pin (n = 0 to 7).

Trigger	Analog Input Pin	A/D Conversion Result Register
ADTRG signal	ANI0	ADCR0
ADTRG signal	ANI1	ADCR1
ADTRG signal	ANI2	ADCR2
ADTRG signal	ANI3	ADCR3
(Software trigger mode)	ANI4	ADCR4
	ANI5	ADCR5
	ANI6	ADCR6
	ANI7	ADCR7

When conversion of all the specified analog input pins has ended, the INTAD signal is generated and A/D conversion is stopped.

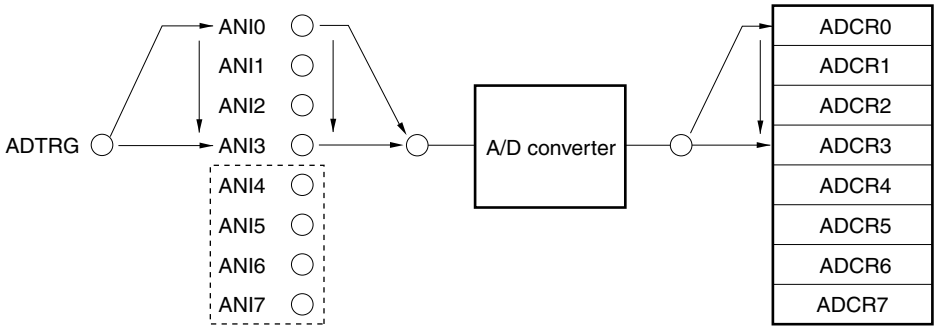
When a trigger is input to the ADTRG pin while the ADM0.ADCE bit is 1, A/D conversion is started again.

The next trigger is ignored even if it is input when the software trigger mode is set.

This is most suitable for applications in which multiple analog inputs are constantly monitored.

Figure 14-15. Example of Scan Mode Operation (External Trigger Scan)

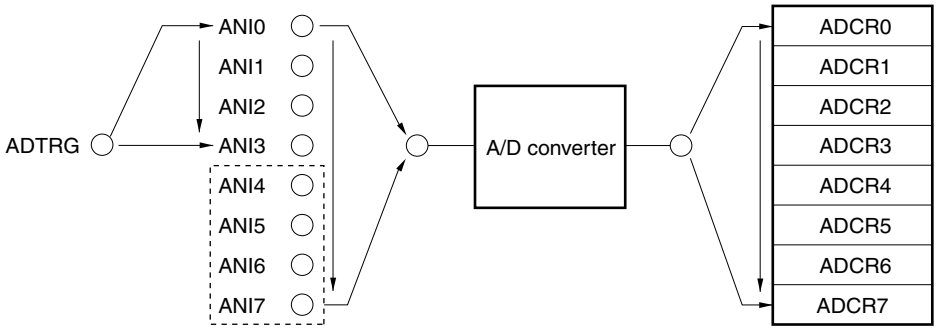
(a) Setting when scanning ANI0 to ANI3 pins



(1) The ADM0.ADCE bit is set to 1 (enable)
(2) The external trigger signal (ADTRG) is generated
(3) The ANI0 pin is A/D converted
(4) The conversion result is stored in the ADCR0 register
(5) The ADTRG signal is generated
(6) The ANI1 pin is A/D converted
(7) The conversion result is stored in the ADCR1 register
(8) The ADTRG signal is generated
(9) The ANI2 pin is A/D converted
(10) The conversion result is stored in the ADCR2 register
(11) The ADTRG signal is generated
(12) The ANI3 pin is A/D converted
(13) The conversion result is stored in the ADCR3 register
(14) The INTAD signal is generated

Caution The ADTRG pin cannot be used as a trigger for the analog input pins enclosed in the broken lines. When a setting is made to scan the ANI0 to ANI7 pins, the ANI4 to ANI7 pins are converted in software trigger mode (see (b) below).

(b) Setting when scanning ANI0 to ANI7 pins



(1) to (13) Same as (a)
(14) The ANI4 pin is A/D converted
(15) The conversion result is stored in the ADCR4 register
(16) The ANI5 pin is A/D converted
(17) The conversion result is stored in the ADCR5 register
(18) The ANI6 pin is A/D converted
(19) The conversion result is stored in the ADCR6 register
(20) The ANI7 pin is A/D converted
(21) The conversion result is stored in the ADCR7 register
(22) The INTAD signal is generated

14.8 Notes on Operation

14.8.1 Stopping conversion operation

When the ADM0.ADCE bit is set to 0 during a conversion operation, the conversion operation stops and the conversion results are not stored in the ADCRn register (n = 0 to 7).

14.8.2 Timer/external trigger interval

For the interval (input time interval) of the trigger in the timer/external trigger mode, see **Table 14-5 Conversion Time in Timer Trigger Mode and External Trigger Mode**.

(1) When interval = 0

When several triggers are input simultaneously, the analog input pin with the smaller ANIn pin number (n) is converted. The other trigger signals input simultaneously are ignored, and the number of trigger inputs is not counted. Note, therefore, that the generation of an interrupt request signal (INTAD) and saving of the result to the ADCRn register are abnormalities (n = 0 to 7).

(2) When 0 < interval < A/D conversion operation time

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 are not stored in the ADCRn register, and the number of trigger inputs is not counted. Note, therefore, that the generation of the INTAD signal and saving of the result to the ADCRn register are abnormalities (n = 0 to 7).

(3) When interval ≥ A/D conversion operation time

The number of trigger inputs is counted, the INTAD signal is generated, and the value at the end of conversion is correctly saved in the ADCRn register. Design so that the interval is equal or greater than the A/D conversion operation time.

14.8.3 Operation in standby mode

(1) HALT mode

In this mode, A/D conversion continues.

(2) IDLE mode, software STOP mode

As clock supply to the A/D converter is stopped, no conversion operations are performed.

When these modes are released by NMI input or the maskable interrupt request signal input pin[Note], the ADM0 to ADM2 and ADCRn registers hold the value. However, when the IDLE or software STOP mode is set during a conversion operation, the conversion operation is suspended. At this time, if the mode released by NMI input or the maskable interrupt request signal input pin[Note], the conversion operation resumes. At this time, the interrupt request signal (INTAD) may be generated, but the conversion result written to the ADCRn register will become undefined (x = 0 to 3, n = 0 to 7).

Note $\overline{\text{INTP000}}$, $\overline{\text{INTP001}}$, $\overline{\text{INTP004}}$, $\overline{\text{INTP005}}$, $\overline{\text{INTP010}}$ to $\overline{\text{INTP013}}$, $\overline{\text{INTP021}}$, $\overline{\text{INTP022}}$, $\overline{\text{INTP050}}$, $\overline{\text{INTP051}}$, $\overline{\text{INTP106}}$, $\overline{\text{INTP107}}$, $\overline{\text{INTP114}}$, $\overline{\text{INTP115}}$, $\overline{\text{INTP124}}$ to $\overline{\text{INTP126}}$, $\overline{\text{INTP130}}$ to $\overline{\text{INTP134}}$, $\overline{\text{INTP137}}$ pins

14.8.4 Timer interrupt request signal in timer trigger mode

The timer interrupt request signal 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 match interrupts for the CPU, disable interrupts using the mask bits of the interrupt control register.

14.8.5 A/D conversion time

(1) Conversion time in software trigger mode

The table below shows the conversion time in the software trigger mode.

Table 14-4. Conversion Time in Software Trigger Mode

FR2	FR1	FR0	Number of Conversion Clocks	Number of Initialization Clocks	Minimum Trigger Interval[Note]
0	0	0	52	11	63
0	0	1	104	22	126
0	1	0	156	33	189
0	1	1	208	44	252
1	0	0	260	55	315
1	0	1	312	66	378
1	1	0	364	77	441
1	1	1	416	88	504

Note Target

(2) Conversion time in timer trigger mode and external trigger mode

The table below shows the conversion time in the timer trigger mode and external trigger mode.

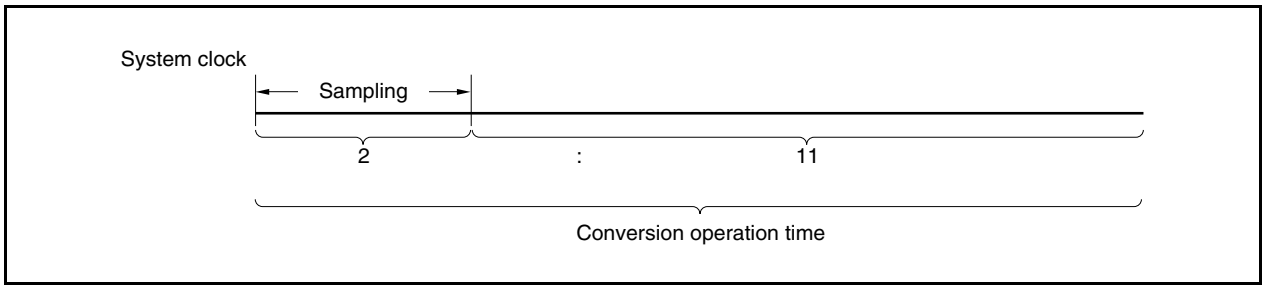
In the external trigger mode, analog delay time (80 ns (TYP.)) required to eliminate noise must be also taken into consideration in addition to the A/D conversion time shown below.

Table 14-5. Conversion Time in Timer Trigger Mode and External Trigger Mode

FR2	FR1	FR0	Number of Conversion Clocks	Number of Initialization Clocks	Minimum Trigger Interval[Note]
0	0	0	52	9	61
0	0	1	104	18	122
0	1	0	156	27	183
0	1	1	208	36	244
1	0	0	260	45	305
1	0	1	312	54	366
1	1	0	364	63	427
1	1	1	416	72	488

Note Target

Figure 14-16. Outline of A/D Conversion Operation Time



14.8.6 Stabilization time

Immediately after clock supply to the A/D converter was started by setting the ADM2.ADCAE bit to 1 and when the IDLE/software STOP mode has been released, time during which the analog circuit stabilizes must be secured.

In order that A/D conversion is correctly executed when it is started, the stabilization time of the analog circuit as well as A/D conversion time are necessary. The V850E/MA3 uses hardware to ensure the lapse of the stabilization time. The stabilization time is as shown in **Table 14-6 Stabilization Time**. In the software trigger mode, the trigger is kept waiting until the stabilization time elapses. In the timer trigger mode and external trigger mode, the trigger is ignored even if it is input within the stabilization time, and the converter waits for input of a new trigger.

Table 14-6. Stabilization Time

FR2	FR1	FR0	Number of Stabilization Time Clocks
0	0	0	31
0	0	1	60
0	1	0	89
0	1	1	118
1	0	0	147
1	0	1	176
1	1	0	205
1	1	1	234

<R>

14.8.7 Variation of A/D conversion results

The results of the A/D conversion may vary depending on the fluctuation of the supply voltage, or may be affected by noise. To reduce the variation, take counteractive measures with the program such as averaging the A/D conversion results.

14.9 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range). %FSR indicates the ratio of analog input voltage that can be converted as a percentage, and is always represented by the following formula regardless of the resolution.

1%FSR = (Max. value of analog input voltage that can be converted − Min. value of analog input voltage that can be converted)/100
= $(AV_{DD0} - 0)/100$
= $AV_{DD0}/100$

1LSB is as follows when the resolution is 10 bits.

$1LSB = 1/2^{10} = 1/1024$
= 0.098%FSR

Accuracy has no relation to resolution, but is determined by overall error.

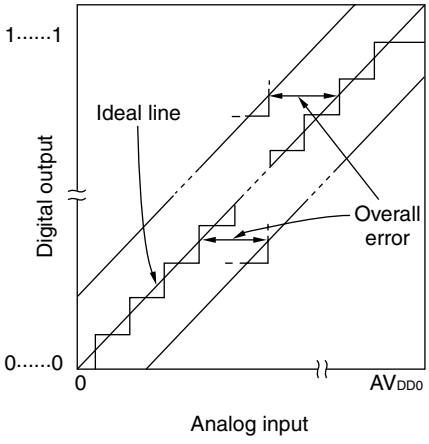
(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, linearity error and errors that are combinations of these express the overall error.

Note that the quantization error is not included in the overall error in the characteristics table.

Figure 14-17. Overall Error

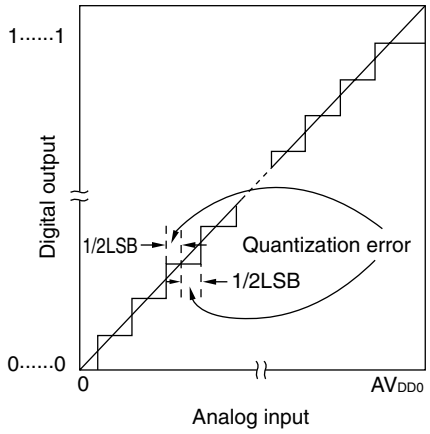


(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

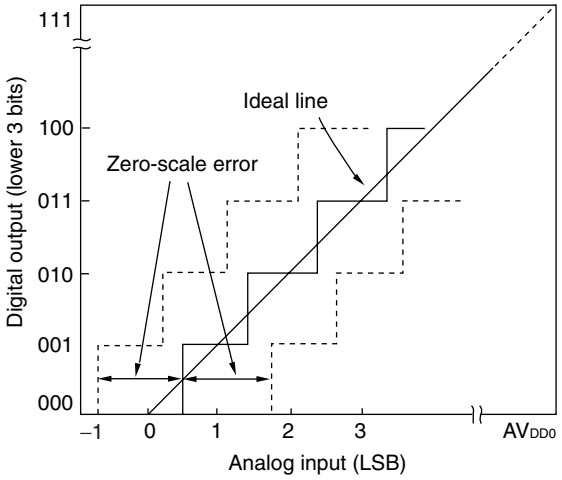
Figure 14-18. Quantization Error



(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.

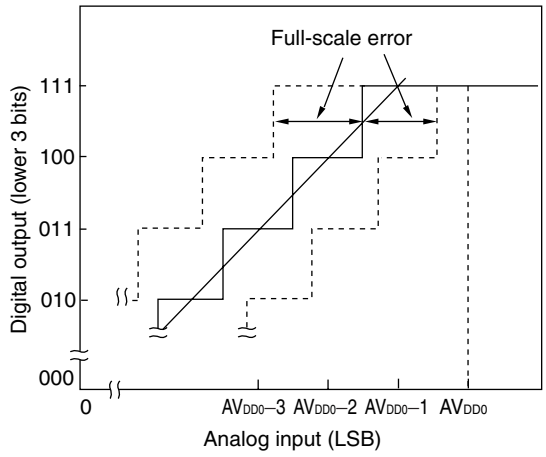
Figure 14-19. Zero-Scale Error



(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 1……110 to 1……111.

Figure 14-20. Full-Scale Error

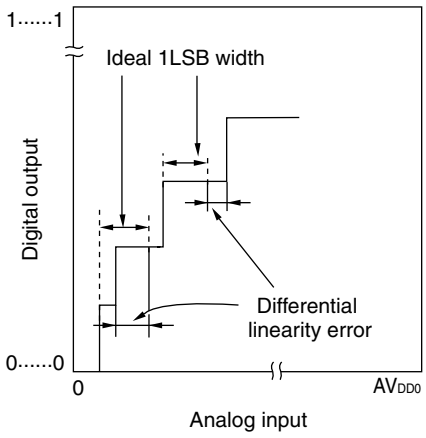


(6) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

<R> 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_{SS0} to AV_{DD0}. When the input voltage is increased or decreased, or when two or more channels are used, see **14.9 (2) Overall error**.

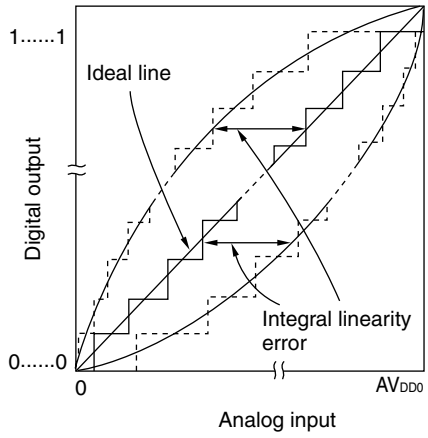
Figure 14-21. Differential Linearity Error



(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 14-22. Integral Linearity Error



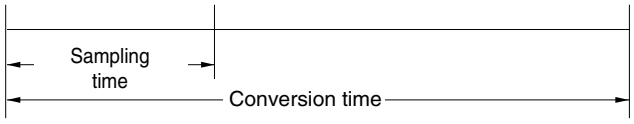
(8) Conversion time

This expresses the time from when a trigger is generated to the time when the digital output was obtained.
The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.

Figure 14-23. Sampling Time



CHAPTER 15 D/A CONVERTER

15.1 Functions

The D/A converter has the following functions.

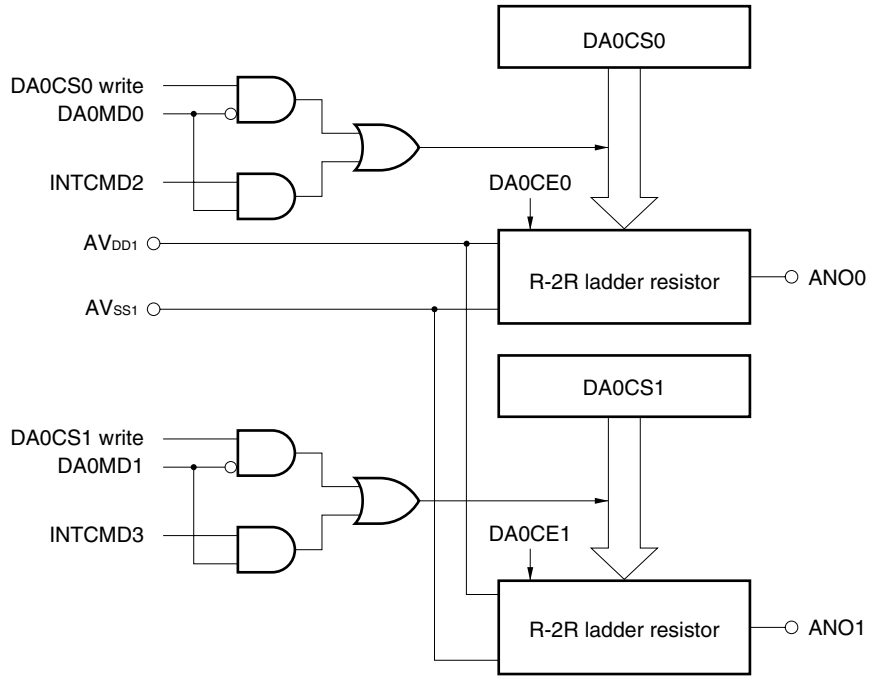
- 8-bit resolution × 2 channels (DAC0, DAC1)
- R-2R ladder method
- Settling time: 3 μs (max.)
- Analog output voltage: AV_{DD1} × m/256 (m = 0 to 255; value set to DA0CSn register)
- Operation modes: Normal mode, real-time output mode

Remark n = 0, 1

15.2 Configuration

The D/A converter configuration is shown below.

Figure 15-1. Block Diagram of D/A Converter



Caution Do not turn off power to the AV_{DD1} pin even in the standby mode.

The D/A converter consists of the following hardware.

Table 15-1. Configuration of D/A Converter

Item	Configuration
Control registers	D/A converter mode register (DA0M) D/A conversion value setting registers 0 and 1 (DA0CS0 and DA0CS1)

15.3 Control Registers

The registers that control the D/A converter are as follows.

- D/A converter mode register (DA0M)
- D/A conversion value setting registers 0 and 1 (DA0CS0 and DA0CS1)

(1) D/A converter mode register (DA0M)

The DA0M register controls the operation of the D/A converter.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF2C2H

	7	6	5	4	3	2	1	0
DA0M	0	0	DA0CE1	DA0CE0	0	0	DA0MD1	DA0MD0

DA0CEn	D/A converter operation enable/disable control (n = 0, 1)
0	Operation disabled
1	Operation enabled

DA0MDn	Selection of D/A converter operation mode (n = 0, 1)
0	Normal mode
1	Real-time output mode[Note]

Note The output trigger in the real-time output mode (DA0MDn bit = 1) is as follows.

- When n = 0: INTCMD2 signal (see **CHAPTER 10 16-BIT INTERVAL TIMER D (TMD)**)
- When n = 1: INTCMD3 signal (see **CHAPTER 10 16-BIT INTERVAL TIMER D (TMD)**)

Caution The output goes into a high-impedance state when D/A conversion is stopped (DA0CEn bit = 0).

(2) D/A conversion value setting registers 0, 1 (DA0CS0, DA0CS1)

The DA0CSn register sets the analog voltage value output to the ANOn pin.

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

Reset input clears these registers to 00H.

After reset: 00H R/W Address: DA0CS0 FFFFF2C0H, DA0CS1 FFFFF2C1H

	7	6	5	4	3	2	1	0
DA0CSn (n = 0, 1)	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

Cautions 1. In the real-time output mode (DA0M.DA0MDn bit = 1), set the DA0CSn register before the INTCMD2/INTCMD3 signal is generated. D/A conversion starts when the INTCMD2/ INTCMD3 signal is generated.

2. Set the DA0M.DAnCE1 bit to 1 after setting the DA0CSn register.

15.4 Operation

15.4.1 Operation in normal mode

D/A conversion is performed using a write operation to the DA0CSn register as the trigger.

The setting method is described below.

<1> Set the DA0M.DA0MDn bit to 0 (normal mode).

<2> Set the analog voltage value to be output to the ANOn pin to the DA0CSn register.
Steps <1> and <2> above constitute the initial settings.

<3> Set the DA0M.DA0CEn bit to 1 (D/A conversion enable).
D/A conversion starts when this setting is performed.

<4> To perform subsequent D/A conversions, write to the DA0CSn register.
The previous D/A conversion result is held until the next D/A conversion is performed.

Remark n = 0, 1

15.4.2 Operation in real-time output mode

D/A conversion is performed using the interrupt request signals (INTCMD2 and INTCMD3) of timer D2 and timer D3 (TMD2 and TMD3) as triggers.

The setting method is described below.

<1> Set the DA0M.DA0MDn bit to 1 (real-time output mode).

<2> Set the analog voltage value to be output to the ANOn pin to the DA0CSn register.

<3> Set the DA0M.DA0CEn bit to 1 (D/A conversion enable).
Steps <1> to <3> above constitute the initial settings.

<4> Operate timer D2 and timer D3 (TMD2 and TMD3).

<5> D/A conversion starts when the INTCMD2 and INTCMD3 signals are generated.

<6> The INTCMD2 and INTCMD3 signals are generated when subsequent D/A conversions are performed.
Before performing the next D/A conversion (generation of INTCMD2 and INTCMD3 signals), set the analog voltage value to be output to the ANOn pin to the DA0CSn register.

15.4.3 Cautions

Observe the following cautions when using the D/A converter of the V850E/MA3.

(1) Do not change the set value of the DA0CSn register while the trigger signal is being issued in the real-time output mode.

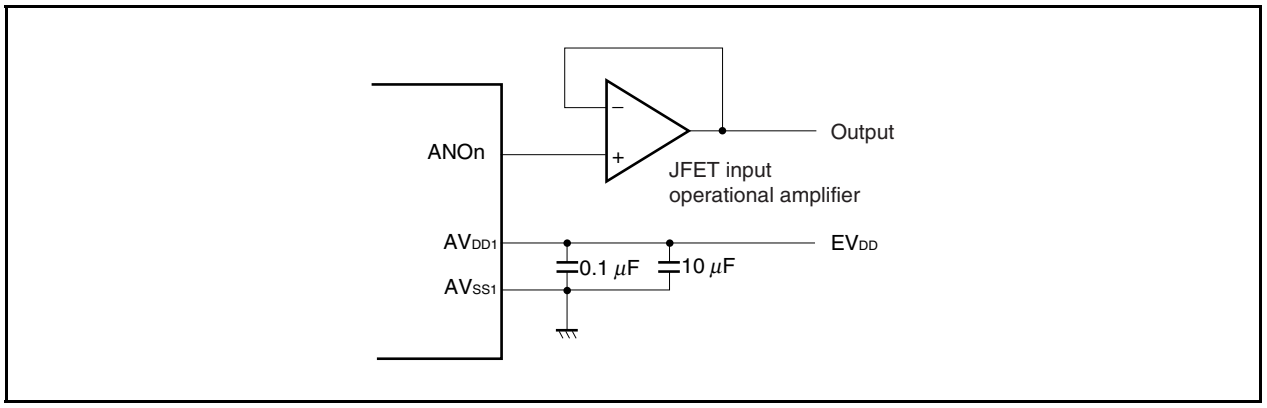
(2) Before changing the operation mode, be sure to clear DA0M.DA0CEn bit to 0.

(3) When using the P80/ANO0 and P81/ANO1 pins as port pins, make sure that their input level changes as little as possible.

(4) Apply power to AV_{DD1} at the same timing as EV_{DD}.

(5) No current can be output from the ANOn pin (n = 0, 1) because the output impedance of the D/A converter is high. When connecting a resistor of 2 MΩ or less, insert a JFET input operational amplifier between the resistor and the ANOn pin.

Figure 15-2. External Pin Connection Example



<R> (6) The D/A converter holds the pin status in the IDLE mode or software STOP mode. To reduce the power consumption, clear the DA0M.DA0CEn bit to 0. The ANOn pin goes into a high-impedance state when the DA0CEn bit = 0.

16.1 Mode Switching Between UARTA and Other Serial Interface

16.1.1 Mode switching between UARTA0 and CSIB0, UARTA1 and CSIB1, and UARTA2 and CSIB2

In the V850E/MA3, UARTA0 and CSIB0, UARTA1 and CSIB1, and UARTA2 and CSIB2 function alternately, and these pins cannot be used at the same time. To use UARTA0/CSIB0, and UARTA1/CSIB1, the PMC4 and PFC4 registers must be set in advance. To use UARTA2/CSIB2, the PMC3, PFC3, and PFCE3 registers must be set in advance.

Caution The operations related to transmission and reception of UARTA0/CSIB0, UARTA1/CSIB1, and UARTA2/CSIB2 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 16-1. Mode Switch Settings of UARTA0/CSIB0, UARTA1/CSIB1

After reset: 00H R/W Address: FFFFF448H

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

After reset: 00H R/W Address: FFFFF468H

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

PMC45	PFC45	Specification of alternate function of P45 pin
0	×	I/O port
1	0	SCK1 I/O
1	1	ASCK1 input

PMC44	PFC44	Specification of alternate function of P44 pin
0	×	I/O port
1	0	SI1 input
1	1	RXD1 input

PMC43	PFC43	Specification of alternate function of P43 pin
0	×	I/O port
1	0	SO1 output
1	1	TXD1 output

PMC42	PFC42	Specification of alternate function of P42 pin
0	×	I/O port
1	0	SCK0 I/O
1	1	ASCK0 input

PMC41	PFC41	Specification of alternate function of P41 pin
0	×	I/O port
1	0	SI0 input
1	1	RXD0 input

PMC40	PFC40	Specification of alternate function of P40 pin
0	×	I/O port
1	0	SO0 output
1	1	TXD0 output

Remark x = don't care

Figure 16-2. UARTA2/CSIB2 Mode Switch Settings

After reset: 00H R/W Address: FFFFF446H

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

After reset: 00H R/W Address: FFFFF466H

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

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	0	0	0	PFCE34	PFCE33	PFCE32	PFCE31	PFCE30

PMC34	PFCE34	PFC34	Specification of alternate function of P34 pin
0	×	×	I/O port
1	0	1	RXD3 input
1	1	0	SCL[Note] I/O

PMC33	PFCE33	PFC33	Specification of alternate function of P33 pin
0	×	×	I/O port
1	0	1	TXD3 output
1	1	0	SDA[Note] I/O

PMC32	PFCE32	PFC32	Specification of alternate function of P32 pin
0	×	×	I/O port
1	0	1	ASCK2 input
1	1	0	SCK2 I/O

PMC31	PFCE31	PFC31	Specification of alternate function of P31 pin
0	×	×	I/O port
1	0	1	RXD2 input
1	1	0	SI2 input

PMC30	PFCE30	PFC30	Specification of alternate function of P30 pin
0	×	×	I/O port
1	0	1	TXD2 output
1	1	0	SO2 output

Note I^2C bus versions (Y products) only
When using the SDA and SCL pins, the pins function as dummy open-drain output pins (P-ch side is always off).

Remark x = don't care

16.1.2 UARTA3/I²C mode switching

In I²C bus versions (Y products) of the V850E/MA3, UARTA3 and I²C function alternately, and these pins cannot be used at the same time. To switch between UARTA3 and I²C, the PMC3, PFC4, and PFCE3 registers must be set in advance.

Caution The operations related to transmission and reception of UARTA3/I²C are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 16-3. UARTA3/I²C Mode Switch Settings

After reset: 00H R/W Address: FFFFF446H

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

After reset: 00H R/W Address: FFFFF466H

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

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	0	0	0	PFCE34	PFCE33	PFCE32	PFCE31	PFCE30

PMC34	PFCE34	PFC34	Specification of alternate function of P34 pin
0	×	×	I/O port
1	0	1	RXD3 input
1	1	0	SCL[Note] I/O

PMC33	PFCE33	PFC33	Specification of alternate function of P33 pin
0	×	×	I/O port
1	0	1	TXD3 output
1	1	0	SDA[Note] I/O

Note I²C bus versions (Y products) only
When using the SDA and SCL pins, the pins function as dummy open-drain output pins (P-ch side is always off).

Remark x = don't care

16.2 Features

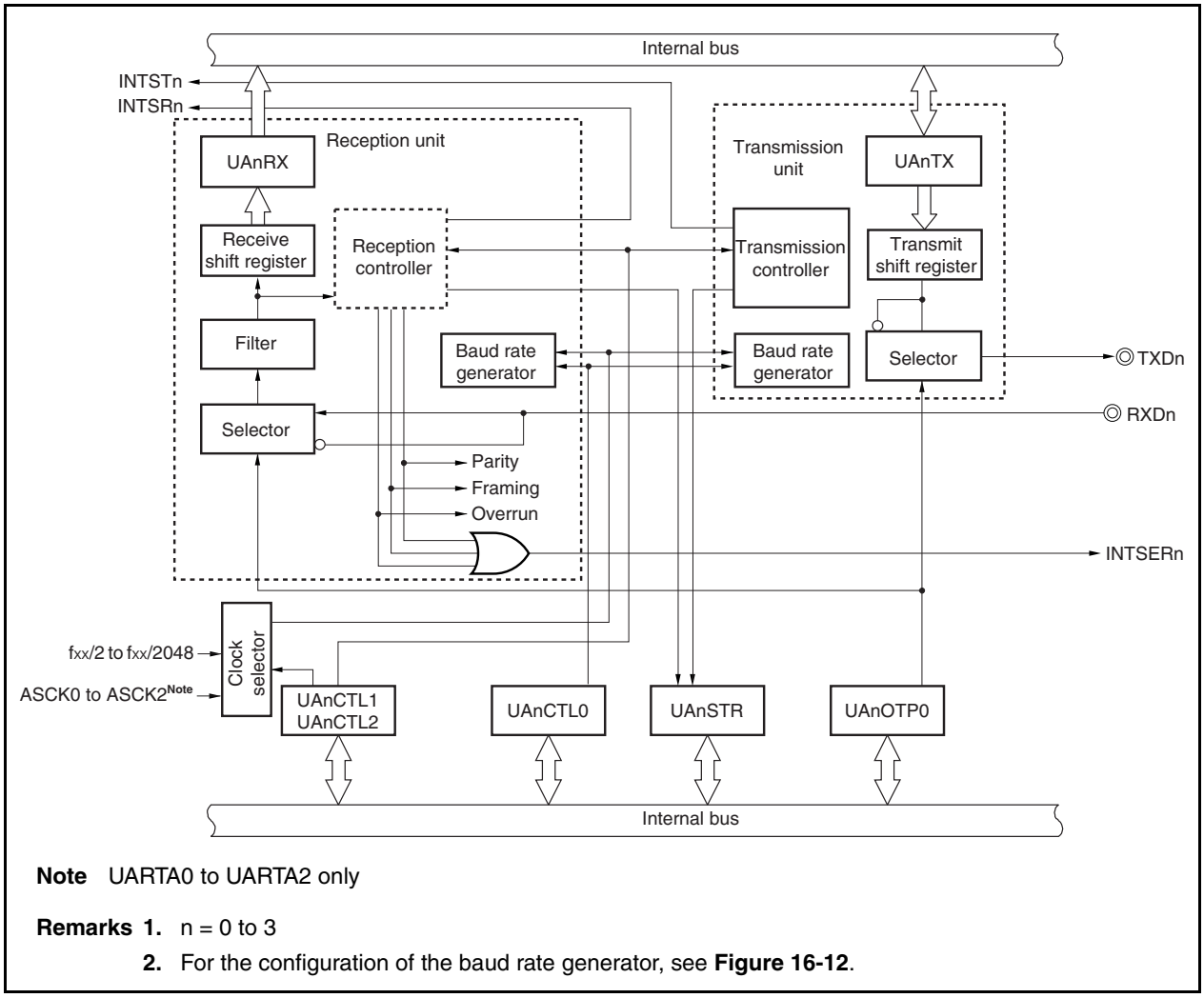
- Transfer rate: 300 bps to 5 Mbps (using peripheral clock (fxx) of 80 MHz and dedicated baud rate generator)
- Full-duplex communication: Internal UARTA receive data register n (UAnRX)
  Internal UARTA transmit data register n (UAnTX)
- 2-pin configuration: TXDn: Transmit data output pin
  RXDn: Receive data input pin
- Reception error output function
  - Parity error
  - Framing error
  - Overrun error
- Interrupt sources: 3
  - Reception error interrupt (INTSERn): This interrupt is generated by ORing the three types of reception errors
  - Reception end interrupt (INTSRn): 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 (INTSTn): This interrupt occurs upon transfer of transmit data from the UAnTX register to the shift register in the transmission enabled status.
- Character length: 7, 8 bits
- Parity function: Odd, even, 0, none
- Transmission stop bit: 1, 2 bits
- On-chip dedicated baud rate generator
- MSB-/LSB-first transfer selectable
- Transmit/receive data inverted input/output possible

Remark n = 0 to 3

16.3 Configuration

The block diagram of the UARTAn is shown below.

Figure 16-4. Block Diagram of UARTAn



Note UARTA0 to UARTA2 only

Remarks 1. n = 0 to 3

2. For the configuration of the baud rate generator, see **Figure 16-12**.

UARTAn consists of the following hardware units.

Table 16-1. Configuration of UARTAn

Item	Configuration
Registers	UARTAn control register 0 (UAnCTL0) UARTAn control register 1 (UAnCTL1) UARTAn control register 2 (UAnCTL2) UARTAn option control register 0 (UAnOPT0) UARTAn status register (UAnSTR) UARTAn receive shift register UARTAn receive data register (UAnRX) UARTAn transmit shift register UARTAn transmit data register (UAnTX)

(1) UARTAn control register 0 (UAnCTL0)

The UAnCTL0 register is an 8-bit register used to specify the UARTAn operation.

(2) UARTAn control register 1 (UAnCTL1)

The UAnCTL1 register is an 8-bit register used to select the base clock (f_{UCLK}) for the UARTAn.

(3) UARTAn control register 2 (UAnCTL2)

The UAnCTL2 register is an 8-bit register used to control the baud rate for the UARTAn.

(4) UARTAn option control register 0 (UAnOPT0)

The UAnOPT0 register is an 8-bit register used to control serial transfer for the UARTAn.

(5) UARTAn status register (UAnSTR)

The UAnSTR register consists of flags indicating the error contents when a reception error occurs. Each one of the reception error flags is set (to 1) upon occurrence of a reception error and is cleared (to 0) by reading the UAnSTR register.

(6) UARTAn receive shift register

This is a shift register used to convert the serial data input to the RXDn 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 (INTSRn) 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 TXDn 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 (INTSTn) is generated.

16.4 Control Registers

(1) UARTAn control register 0 (UAnCTL0)

The UAnCTL0 register is an 8-bit register that controls the UARTAn serial transfer operation.

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

Reset input sets this register to 10H.

(1/2)

After reset: 10H R/W Address: UA0CTL0 FFFFFA00H, UA1CTL0 FFFFFA10H
UA2CTL0 FFFFFA20H, UA3CTL0 FFFFFA30H

	<7>	<6>	<5>	<4>	3	2	1	0
UAnCTL0	UAnPWR	UAnTXE	UAnRXE	UAnDIR	UAnPS1	UAnPS0	UAnCL	UAnSL

(n = 0 to 3)

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 **16.7 (1) (a) Base clock**).
<R> • 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.
<R> • 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.

(2/2)

UAnRXE	Reception operation enable
0	Disable reception operation
1	Enable reception operation

- To start reception, set the UAnPWR bit to 1 and then set the UAnRXE bit to 1.
- To initialize the reception unit, clear the UAnRXE bit to 0, wait for two cycles of the base clock, and then set the UAnRXE bit to 1 again. Otherwise, initialization may not be executed (for the base clock, see **16.7 (1) (a) Base clock**).
- When the operation is enabled (UAnPWR bit = 1), the reception operation is enabled after two or more cycles of the base clock (f_{UCLK}) have elapsed since UAnRXE = 1.
- 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.

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 **16.6.6 Parity types and operations**.

(2) UARTAn control register 1 (UAnCTL1)

For details, see **16.7 (2) UARTAn control register 1 (UAnCTL1)**.

(3) UARTAn control register 2 (UAnCTL2)

For details, see **16.7 (3) UARTAn control register 2 (UAnCTL2)**.

(4) UARTAn option control register 0 (UAnOPT0)

The UAnOPT0 register is an 8-bit register that controls the serial transfer operation of the UARTAn register.

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

Reset input sets this register to 14H.

After reset: 14H R/W Address: UA0OPT0 FFFFFA03H, UA1OPT0 FFFFFA13H
UA2OPT0 FFFFFA23H, UA3OPT0 FFFFFA33H

	7	6	5	4	3	2	1	0
UAnOPT0	0	0	0	1	0	1	UAnTDL	UAnRDL

(n = 0 to 3)

UAnTDL	Transmit data level bit
0	Normal output of transfer data
1	Inverted output of transfer data

- The output level of the TXDn 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 RXDn pin can be inverted using the UAnRDL bit.
- This register can be set when the UAnPWR bit = 0 or the UAnCTL0.UAnRXE bit = 0.
- When the UAnRDL bit is set to 1 (inverted input of receive data), reception must be enabled (UAnCTL0.UAnRXE bit = 1) after setting the data reception pin to the UART reception pin (RXDn) 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.

<R>

Caution Be sure to clear bits 3 and 5 to 7 to "0", and set bits 4 and 2 to "1". Operation with other settings is not guaranteed.

<R> (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 = 0
UAnTSF bit	• UAnCTL0.UAnTXE = 0
UAnPE, UAnFE, UAnOVE bits	• 0 write • UAnCTL0.UAnRXE = 0

After reset: 00H R/W Address: UA0STR FFFFFA04H, UA1STR FFFFFA14H, UA2STR FFFFFA24H, UA3STR FFFFFA34H

UAnSTR (n = 0 to 3)

<7>	6	5	4	3	<2>	<1>	<0>
UAnTSF	0	0	0	0	UAnPE	UAnFE	UAnOVE

UAnTSF	Transfer status flag
0	• When the UAnPWR bit = 0 or the UAnTXE bit = 0 has been set. • When, following transfer end, there was no next data transfer from UAnTX register
1	Write to UAnTX register
The UAnTSF bit is always 1 when performing continuous transmission. When initializing the transmission unit, check that the UAnTSF bit = 0 before performing initialization. The transmit data is not guaranteed when initialization is performed while the UAnTSF bit = 1.	

UAnPE	Parity error flag
0	• When the UAnPWR bit = 0 or the UAnRXE bit = 0 has been set. • When 0 has been written
1	When parity of data and parity bit do not match during reception.
• The operation of the UAnPE bit is controlled by the settings of the UAnCTL0.UAnPS1 and UAnCTL0.UAnPS0 bits. • The UAnPE bit can be read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. When 1 is written to this bit, the value is retained.	

UAnFE	Framing error flag
0	• When the UAnPWR bit = 0 or the UAnRXE bit = 0 has been set • When 0 has been written
1	When no stop bit is detected during reception
• Only the first bit of the receive data stop bits is checked, regardless of the value of the UAnCTL0.UAnSL bit. • The UAnFE bit can be both read and written, but it can only be cleared by writing 0 to it, and it cannot be set by writing 1 to it. When 1 is written to this bit, the value is retained.	

UAnOVE	Overrun error flag
0	• When the UAnPWR bit = 0 or the UAnRXE bit = 0 has been set. • When 0 has been written
1	When receive data has been set to the UAnRX register and the next receive operation is ended before that receive data has been read.
• When an overrun error occurs, the data is discarded without the next receive data being written to the UAnRX register. • The UAnOVE bit can be both read and written, but it can only be cleared by writing 0 to it. 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 (INTSRn) 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 (the 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 input, the UAnRX register can be set to FFH by clearing the UAnCTL0.UAnPWR bit to 0.

After reset: FFH R Address: UA0RX FFFFFA06H, UA1RX FFFFFA16H, UA2RX FFFFFA26H, UA3RX FFFFFA36H

	7	6	5	4	3	2	1	0
UAnRX								

(n = 0 to 3)

(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 (INTSTn) is generated.

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

Reset input sets this register to FFH.

After reset: FFH R/W Address: UA0TX FFFFFA07H, UA1TX FFFFFA17H, UA2TX FFFFFA27H, UA3TX FFFFFA37H

	7	6	5	4	3	2	1	0
UAnTX								

(n = 0 to 3)

16.5 Interrupt Request Signals

The following three interrupt request signals are generated from UARTAn.

- Reception error interrupt request signal (INTSERn)
- Reception end interrupt request signal (INTSRn)
- Transmission enable interrupt request signal (INTSTn)

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 16-2. Interrupts and Their Default Priorities

Interrupt	Priority
Reception error	High
Reception end	⇕
Transmission enable	Low

(1) Reception error interrupt request signal (INTSERn)

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 (INTSRn)

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 (INTSTn)

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.

16.6 Operation

16.6.1 Data format

Full-duplex serial data reception and transmission is performed.

As shown in Figure 16-5, one data frame of transmit/receive data consists of a start bit, character bits, parity bit, and stop bit(s).

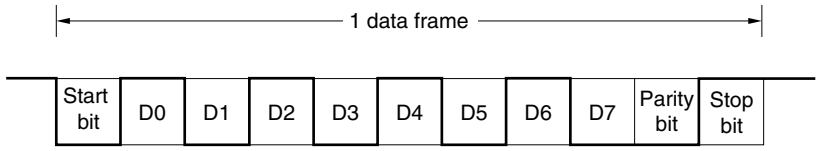
Specification of the character bit length within 1 data frame, parity selection, specification of the stop bit length, and specification of MSB/LSB-first transfer are performed using the UAnCTL0 register.

Moreover, control of UARTAn output/inverted output for the TXDn bit is performed using the UAnOPT0.UAnTDL bit.

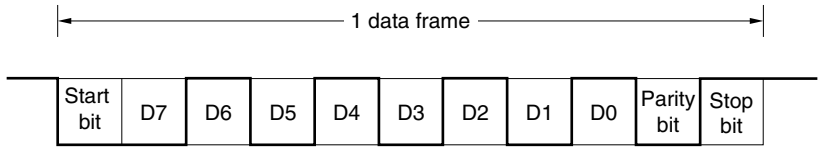
- Start bit.....1 bit
- Character bits.....7 bits/8 bits
- Parity bit .....Even parity/odd parity/0 parity/no parity
- Stop bit .....1 bit/2 bits

Figure 16-5. UARTA Transmit/Receive Data Format

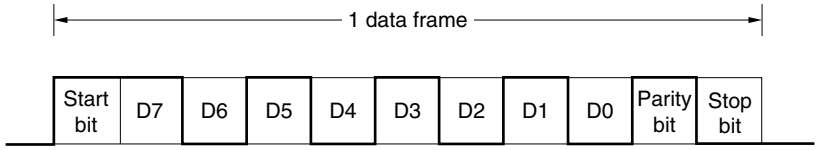
(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H



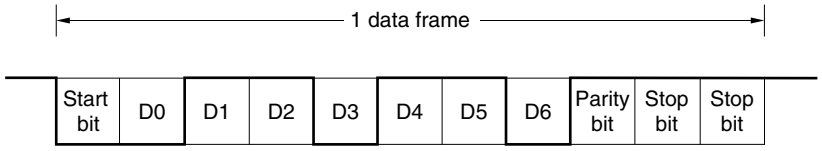
(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H



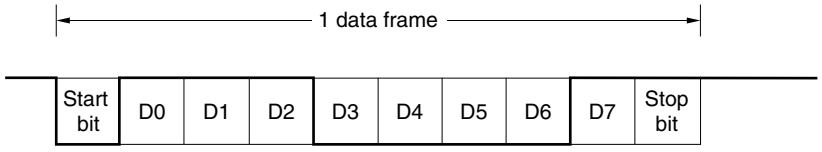
(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H, TXDn inversion



(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



16.6.2 UART transmission

First set the base clock of UARTAn with the UAnCTL1 register and the baud rate clock with the UAnCTL2 register. Set the output level of transmit data with the UAnOPT0 register. Also set a transfer direction, parity, data character length, and stop bit length with the UAnCTL0 register.

A high level is output to the TXDn 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.

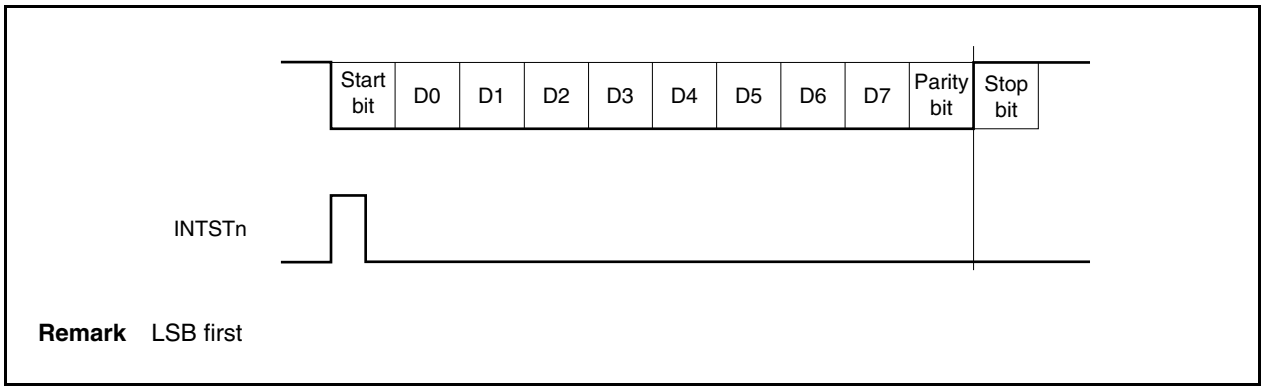
The data in the UAnTX register is transferred to the UARTAn transmit shift register upon the start of the transmit operation.

A transmission enable interrupt request signal (INTSTn) 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 TXDn pin.

Write of the next transmit data to the UAnTX register is enabled by generating the INTSTn signal.

Remark n = 0 to 3

Figure 16-6. UART Transmission



Remark LSB first

16.6.3 Continuous transmission procedure

UARTAn can write the next transmit data to the UAnTX register when the UARTAn transmit shift register starts the shift operation. The transmit timing of the UARTAn transmit shift register can be judged from the transmission enable interrupt request signal (INTSTn).

An efficient communication rate is realized by writing the data to be transmitted next to the UAnTX register during transfer.

Caution During continuous transmission execution, perform initialization after checking that the UAnSTR.UAnTSF bit is 0. The transmit data cannot be guaranteed when initialization is performed while the UAnTSF bit is 1.

Remark n = 0 to 3

Figure 16-7. Continuous Transmission Processing Flow

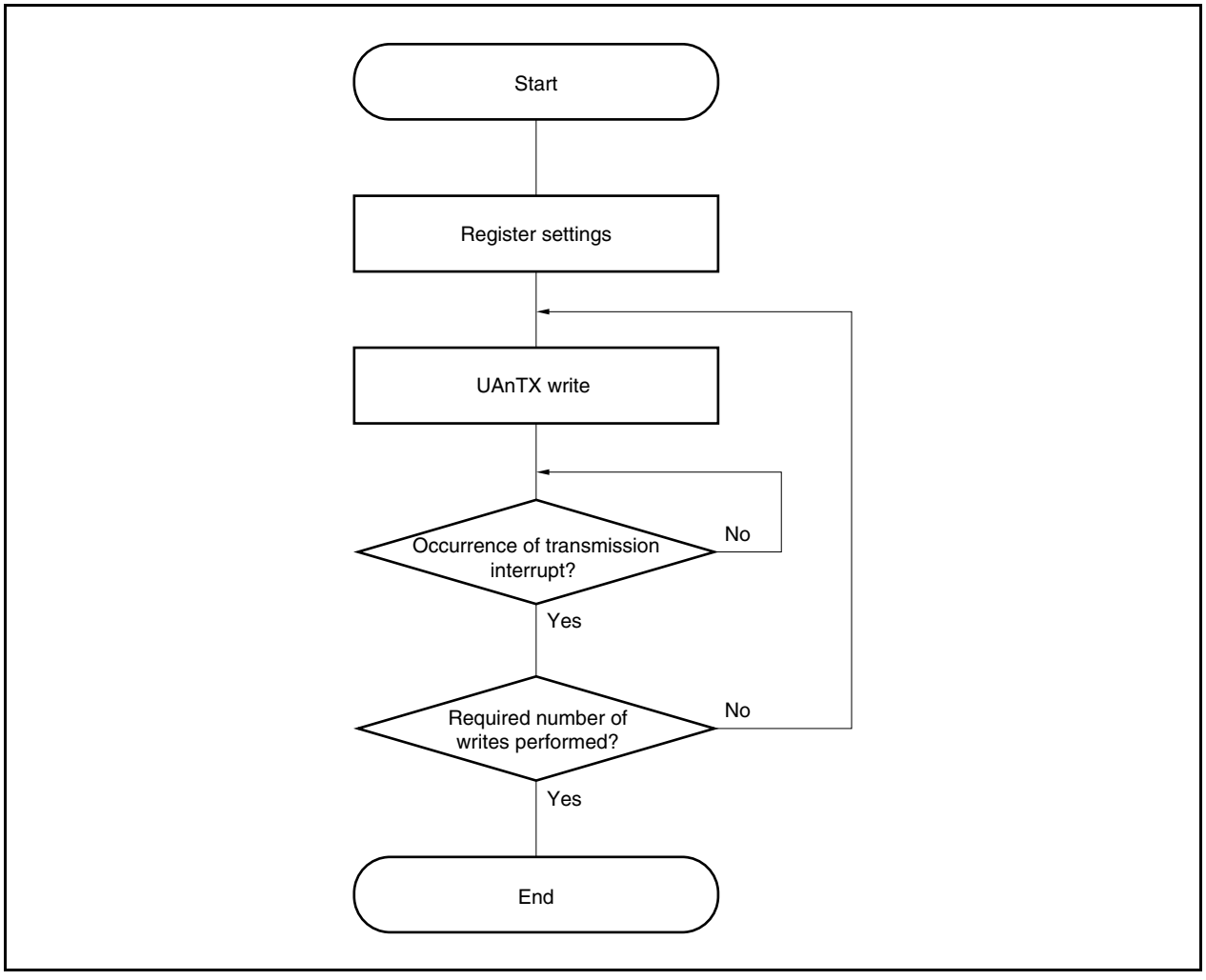
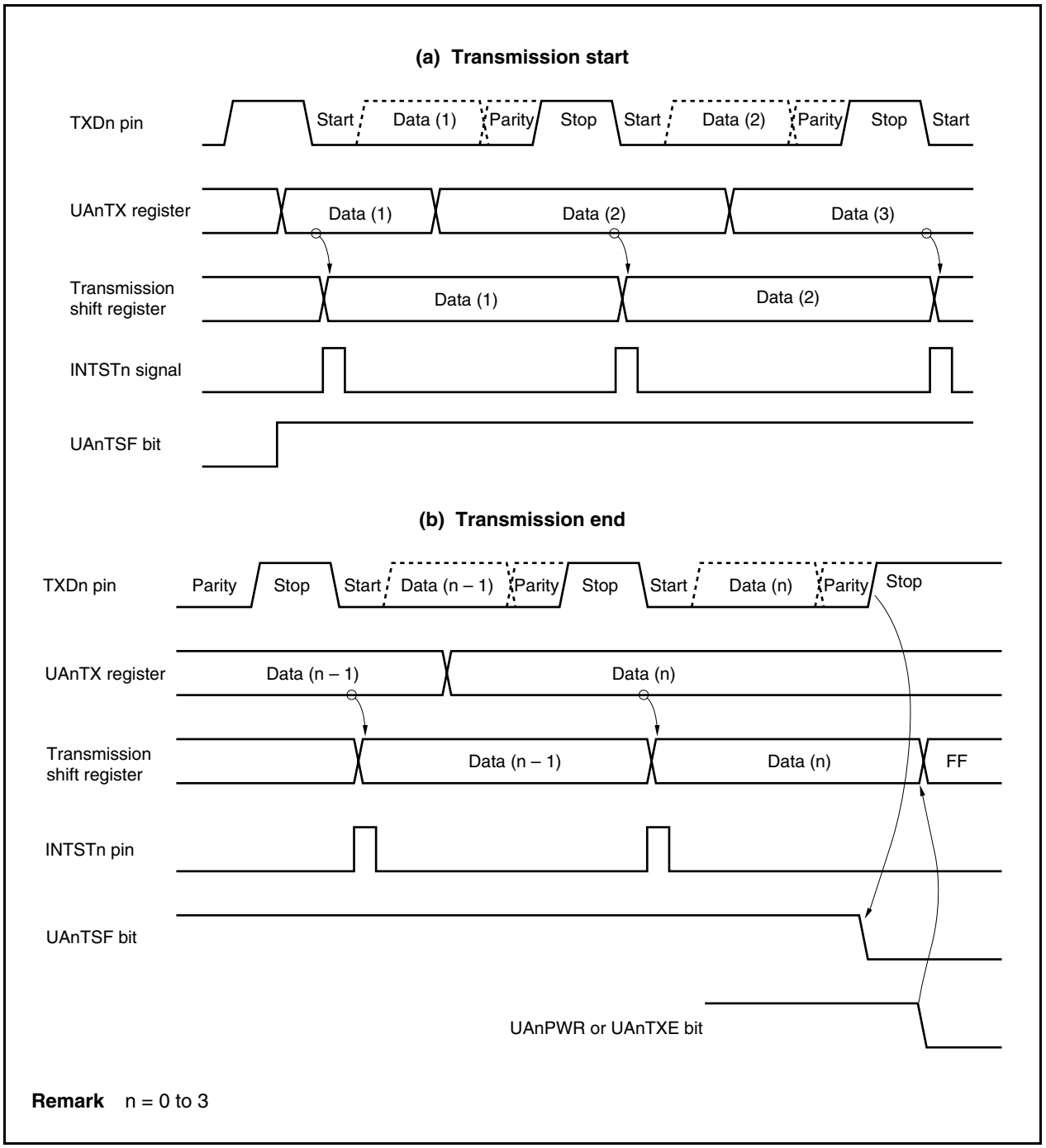


Figure 16-8. Continuous Transmission Operation Timing



Remark n = 0 to 3

16.6.4 UART reception

First set the base clock of UARTAn with the UAnCTL1 register and the baud rate clock with the UAnCTL2 register. Set the input level of receive data with the UAnOPT0 register. Also set a transfer direction, parity, data character length, and stop bit length with the UAnCTL0 register.

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 RXDn pin is monitored and start bit detection is performed.

Start bit detection is performed using a two-step detection routine.

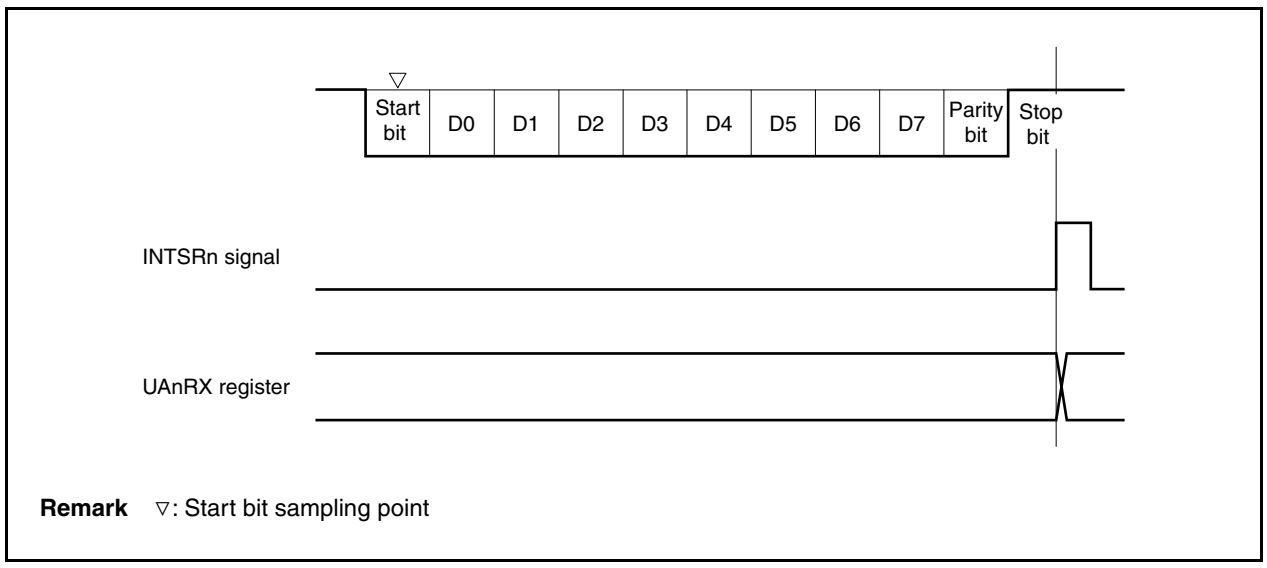
First the rising edge of the RXDn pin is detected and sampling is started at the falling edge. The start bit is recognized if the RXDn pin is low level at the start bit sampling point. After a start bit has been recognized, the receive operation starts, and serial data is saved to the UARTAn receive shift register according to the set baud rate.

When the reception end interrupt request signal (INTSRn) 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, <R> reception continues until the reception position of the first stop bit, and INTSERn is output following reception end.

Remark n = 0 to 3

Figure 16-9. UART Reception



Remark ▽: Start bit sampling point

Cautions 1. Be sure to read the UAnRX register even when a reception error occurs. If the UAnRX register is not read, an overrun error occurs during reception of the next data, and reception errors continue occurring indefinitely.

2. The operation during reception is performed assuming that there is only one stop bit. A second stop bit is ignored.

3. When reception is completed, read the UAnRX register after the reception end interrupt request signal (INTSRn) has been generated, and clear the UAnPWR or UAnRXE bit to 0. If the UAnPWR or UAnRXE bit is cleared to 0 before the INTSRn signal is generated, the read value of the UAnRX register cannot be guaranteed.

4. If receive end processing (INTSRn signal generation) of UARTAn and the UAnPWR bit = 0 or UAnRXE bit = 0 conflict, the INTSRn signal may be generated in spite of these being no data stored in the UAnRX register.

To end reception without waiting INTSRn signal generation, be sure to set (1) the interrupt mask flag (SRICn.SRMKn), clear (0) the interrupt request flag (SRICn.SRIFn) in that order, and then clear the UAnPWR bit or UAnRXE bit to 0.

16.6.5 Reception errors

Errors during a receive operation are of three types: parity errors, framing errors, and overrun errors. Data reception result error flags are set in the UAnSTR register and a reception error interrupt request signal (INTSERn) 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.

<R> **Caution The reception end interrupt request signal (INTSRn) and reception error interrupt request signal (INTSERn) are not generated simultaneously. The INTSRn signal is generated when a reception ends normally. The INTSERn signal is generated and the INTSRn signal is not generated when a reception error occurs.**

Remark n = 0 to 3

- 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

16.6.6 Parity types and operations

The parity bit is used to detect bit errors in the communication data. Normally the same parity is used on the transmission side and the reception side.

In the case of even parity and odd parity, it is possible to detect odd-count bit errors. In the case of 0 parity and no parity, errors cannot be detected.

(a) Even parity

(i) During transmission

The number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so as to be an even number. The parity bit values are as follows.

- Odd number of bits whose value is “1” among transmit data: 1
- Even number of bits whose value is “1” among transmit data: 0

(ii) During reception

The number of bits whose value is “1” among the reception data, including the parity bit, is counted, and if it is an odd number, a parity error is output.

(b) Odd parity

(i) During transmission

Opposite to even parity, the number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so that it is an odd number. The parity bit values are as follows.

- Odd number of bits whose value is “1” among transmit data: 0
- Even number of bits whose value is “1” among transmit data: 1

(ii) During reception

The number of bits whose value is “1” among the receive data, including the parity bit, is counted, and if it is an even number, a parity error is output.

(c) 0 parity

During transmission, the parity bit is always made 0, regardless of the transmit data.

During reception, parity bit check is not performed. Therefore, no parity error occurs, regardless of whether the parity bit is 0 or 1.

(d) No parity

No parity bit is added to the transmit data.

Reception is performed assuming that there is no parity bit. No parity error occurs since there is no parity bit.

16.6.7 Receive data noise filter

This filter samples the RXDn 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 16-11**). See **16.7 (1) (a) Base clock** regarding the base clock.

Moreover, since the circuit is as shown in Figure 16-10, the processing that goes on within the receive operation is delayed by 3 clocks in relation to the external signal status.

Remark n = 0 to 3

Figure 16-10. Noise Filter Circuit

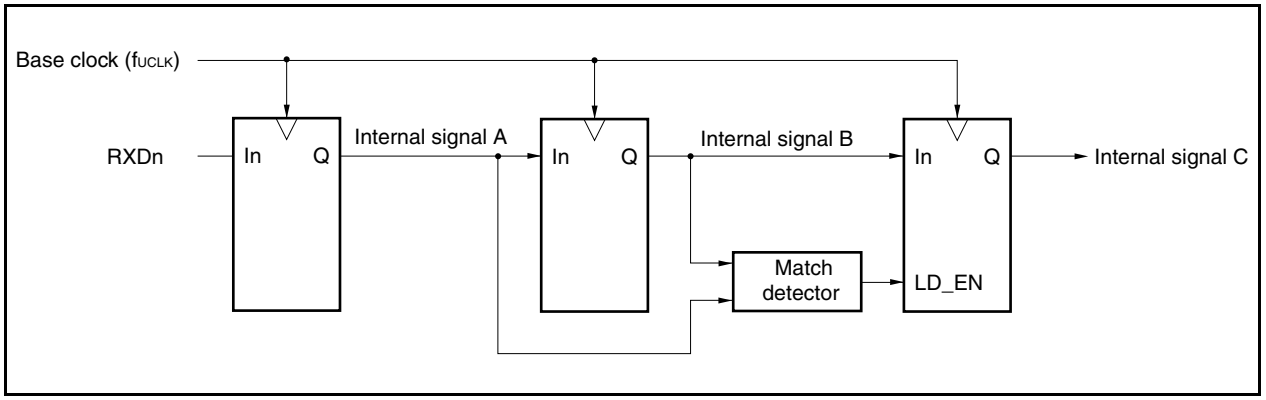
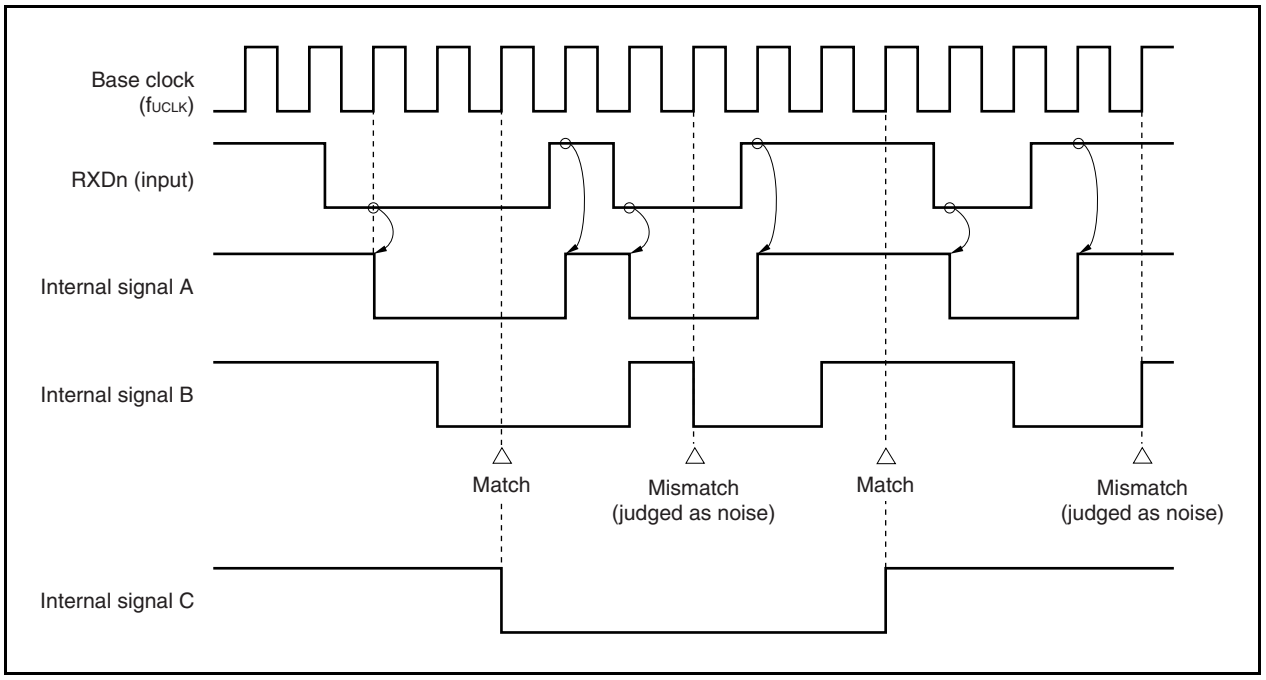


Figure 16-11. Timing of RXDn Signal Judged as Noise



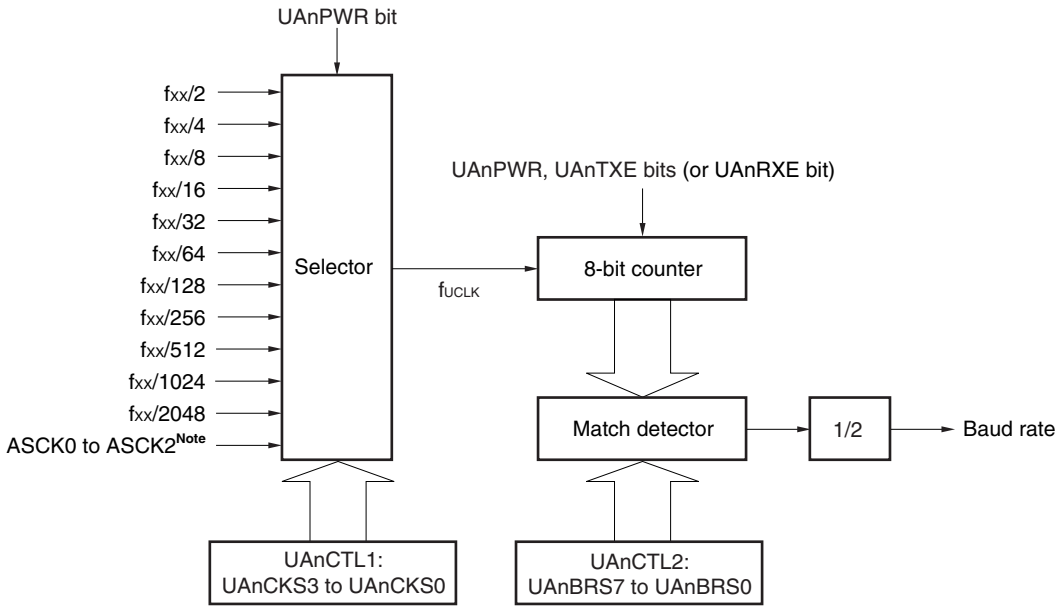
16.7 Dedicated Baud Rate Generator

The dedicated baud rate generator consists of a source clock selector block and an 8-bit programmable counter, and generates a serial clock during transmission and reception with UARTAn. Regarding the serial clock, a dedicated baud rate generator output can be selected for each channel.

There is an 8-bit counter for transmission and another one for reception.

(1) Baud rate generator configuration

Figure 16-12. Configuration of Baud Rate Generator



Note Valid only for the UARTA0 to UARTA2 only. Setting UARTA3 is prohibited.

Caution If the CPU clock (f_{CPU}) is slower than f_{UCLK}, UARTAn cannot be used.

Remarks 1. n = 0 to 3

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. 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 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 input clears this register to 00H.

Caution Clear the UAnCTL0.UAnPWR bit to 0 before rewriting the UAnCTL1 register.

After reset: 00H R/W Address: UA0CTL1 FFFFFA01H, UA1CTL1 FFFFFA11H
UA2CTL1 FFFFFA21H, UA3CTL1 FFFFFA31H

	7	6	5	4	3	2	1	0
UAnCTL1 (n = 0 to 3)	0	0	0	0	UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0

UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0	Base clock (f_{UCLK}) selection
0	0	0	0	fxx/2
0	0	0	1	fxx/4
0	0	1	0	fxx/8
0	0	1	1	fxx/16
0	1	0	0	fxx/32
0	1	0	1	fxx/64
0	1	1	0	fxx/128
0	1	1	1	fxx/256
1	0	0	0	fxx/512
1	0	0	1	fxx/1,024
1	0	1	0	fxx/2,048
1	0	1	1	External clock[Note] (ASCK0 to ASCK2 pins)
Other than above				Setting prohibited

Note Valid only for the UARTA0 to UARTA2 only. Setting UARTA3 is 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 input sets this register to FFH.

Caution Clear the UAnCTL0.UAnPWR bit to 0 or clear the UAnTXE and UAnRXE bits to 00 before rewriting the UAnCTL2 register.

After reset: FFH R/W Address: UA0CTL2 FFFFFA02H, UA1CTL2 FFFFFA12H
UA2CTL2 FFFFFA22H, UA3CTL2 FFFFFA32H

UAnCTL2 (n = 0 to 3)

7	6	5	4	3	2	1	0
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 = 80 MHz = 80,000,000 Hz

Setting value of UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits = 0001B (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 16-3. Baud Rate Generator Setting Data

Baud Rate	f_{XX} = 80 MHz			f_{XX} = 66 MHz			f_{XX} = 50 MHz		
(bps)	f_{UCLK}	k	ERR (%)	f_{UCLK}	k	ERR (%)	f_{UCLK}	k	ERR (%)
300	f_{XX}/1,024	130	0.16	f_{XX}/512	215	–0.07	f_{XX}/512	163	–0.15
600	f_{XX}/512	130	0.16	f_{XX}/256	215	–0.07	f_{XX}/256	163	–0.15
1,200	f_{XX}/256	130	0.16	f_{XX}/128	215	–0.07	f_{XX}/128	163	–0.15
2,400	f_{XX}/128	130	0.16	f_{XX}/64	215	–0.07	f_{XX}/64	163	–0.15
4,800	f_{XX}/64	130	0.16	f_{XX}/32	215	–0.07	f_{XX}/32	163	–0.15
9,600	f_{XX}/32	130	0.16	f_{XX}/16	215	–0.07	f_{XX}/16	163	–0.15
19,200	f_{XX}/16	130	0.16	f_{XX}/8	215	–0.07	f_{XX}/8	163	–0.15
31,250	f_{XX}/8	160	0.00	f_{XX}/8	132	0.00	f_{XX}/4	200	0.00
38,400	f_{XX}/8	130	0.16	f_{XX}/4	215	–0.07	f_{XX}/4	163	–0.15
76,800	f_{XX}/4	130	0.16	f_{XX}/2	215	–0.07	f_{XX}/2	163	–0.15
153,600	f_{XX}/2	130	0.16	f_{XX}/2	107	0.39	f_{XX}/2	81	0.47
250,000	f_{XX}/2	80	0.00	f_{XX}/2	66	0.00	f_{XX}/2	50	0.00
312,500	f_{XX}/2	64	0.00	f_{XX}/2	53	–0.377	f_{XX}/2	40	0.00
2,000,000	f_{XX}/2	10	0.00	–	–	–	–	–	–
5,000,000	f_{XX}/2	4	0.00	–	–	–	–	–	–

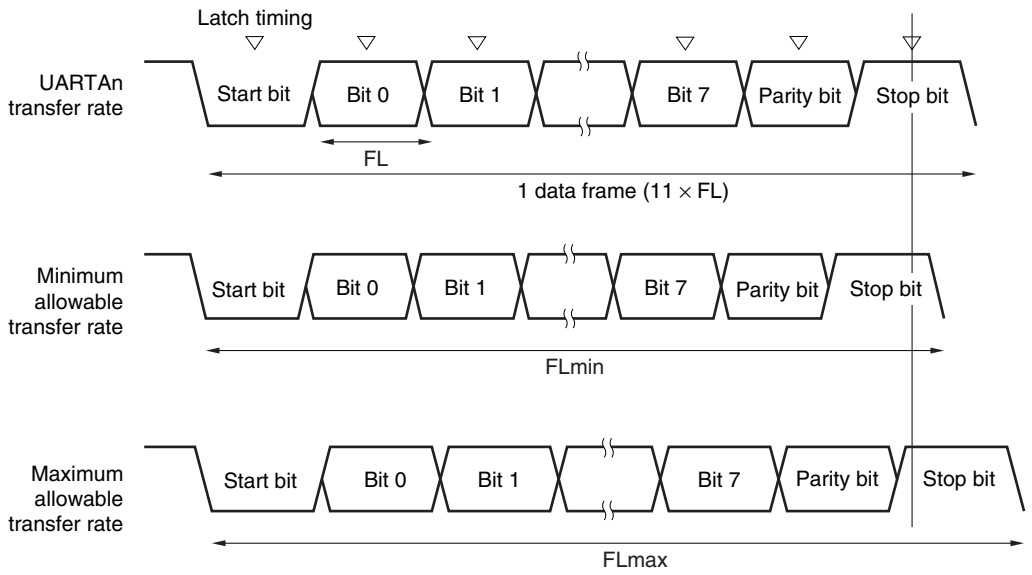
Remark f_{XX}: Peripheral clock frequency
f_{UCLK}: Frequency of base clock selected by UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits
k: Setting value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (n = 0 to 3)
ERR: Baud rate error (%)

(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 16-13. Allowable Baud Rate Range During Reception



Remark n = 0 to 3

As shown in Figure 16-13, the receive data latch timing is determined by the counter set using the UAnCTL2 register following start bit detection. The transmit data can be normally received if up to the last data (stop bit) can be received in time for this latch timing.

When this is applied to 11-bit reception, the following is the theoretical result.

$FL = (Brate)^{-1}$

Brate: UARTAn baud rate (n = 0 to 3)

k: Setting value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (n = 0 to 3)

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.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2} Brate$$

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.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k - 2} Brate$$

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 16-4. 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%

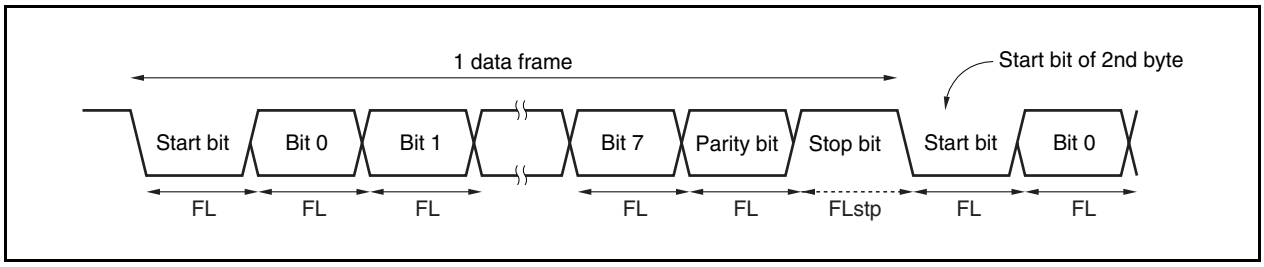
Remarks 1. The reception accuracy depends on the bit count in 1 frame, the input clock frequency, and the division ratio (k). The higher the input clock frequency and the larger the division ratio (k), the higher the accuracy.

2. k: Setting value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits (n = 0 to 3)

(8) Baud 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 16-14. Transfer Rate During Continuous Transfer



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.

$$\text{Transfer rate} = 11 \times FL + (2/f_{UCLK})$$

16.8 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 TXDn pin output also holds and outputs the value it had immediately before the clock supply was stopped. However, the operation is not guaranteed after the clock supply is resumed. Therefore, after the clock supply is resumed, the circuits should be initialized by setting the UAnCTL0.UAnPWR, UAnCTL0.UAnRXE, and UAnCTL0.UAnTXE bits to 000.

Remark n = 0 to 3

CHAPTER 17 CLOCKED SERIAL INTERFACE B (CSIB)

17.1 Mode Switching Between CSIB and Other Serial Interface

17.1.1 Mode switching between UARTA0 and CSIB0, UARTA1 and CSIB1, and UARTA2 and CSIB2

In the V850E/MA3, UARTA0 and CSIB0, UARTA1 and CSIB1, and UARTA2 and CSIB2 function alternately, and these pins cannot be used at the same time. To switch between UARTA0 and CSIB0, and UARTA1 and CSIB1, the PMC4 and PFC4 registers must be set in advance. To switch between UARTA2 and CSIB2, the PMC3, PFC3, and PFCE3 registers must be set in advance.

Caution The operations related to transmission and reception of UARTA0/CSIBn, UARTA1/CSIB1, and UARTA2/CSIB2 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

17.2 Features

- Transfer rate: 10 Mbps (using internal clock)
- Master mode and slave mode selectable
- 8-bit to 16-bit transfer, 3-wire serial interface
- Interrupt request signals (INTCSIERn, INTCSIRn, INTCSITn)
- 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 SOn: Serial data output
  SIn: Serial data input
  SCKn: Serial clock output

  Transmission mode, reception mode, and transmission/reception mode specifiable

Remark n = 0 to 2

Figure 17-1. Mode Switch Settings of UARTA0/CSIB0, UARTA1/CSIB1

After reset: 00H R/W Address: FFFFF448H

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

After reset: 00H R/W Address: FFFFF468H

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

PMC45	PFC45	Specification of alternate function of P45 pin
0	×	I/O port
1	0	SCK1 I/O
1	1	ASCK1 input

PMC44	PFC44	Specification of alternate function of P44 pin
0	×	I/O port
1	0	SI1 input
1	1	RXD1 input

PMC43	PFC43	Specification of alternate function of P43 pin
0	×	I/O port
1	0	SO1 output
1	1	TXD1 output

PMC42	PFC42	Specification of alternate function of P42 pin
0	×	I/O port
1	0	SCK0 I/O
1	1	ASCK0 input

PMC41	PFC41	Specification of alternate function of P41 pin
0	×	I/O port
1	0	SI0 input
1	1	RXD0 input

PMC40	PFC40	Specification of alternate function of P40 pin
0	×	I/O port
1	0	SO0 output
1	1	TXD0 output

Remark x = don't care

Figure 17-2. UARTA2/CSIB2 Mode Switch Settings

After reset: 00H R/W Address: FFFFF446H

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

After reset: 00H R/W Address: FFFFF466H

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

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	0	0	0	PFCE34	PFCE33	PFCE32	PFCE31	PFCE30

PMC34	PFCE34	PFC34	Specification of alternate function of P34 pin
0	×	×	I/O port
1	0	1	RXD3 input
1	1	0	SCL[Note] I/O

PMC33	PFCE33	PFC33	Specification of alternate function of P33 pin
0	×	×	I/O port
1	0	1	TXD3 output
1	1	0	SDA[Note] I/O

PMC32	PFCE32	PFC32	Specification of alternate function of P32 pin
0	×	×	I/O port
1	0	1	ASCK2 input
1	1	0	SCK2 I/O

PMC31	PFCE31	PFC31	Specification of alternate function of P31 pin
0	×	×	I/O port
1	0	1	RXD2 input
1	1	0	SI2 input

PMC30	PFCE30	PFC30	Specification of alternate function of P30 pin
0	×	×	I/O port
1	0	1	TXD2 output
1	1	0	SO2 output

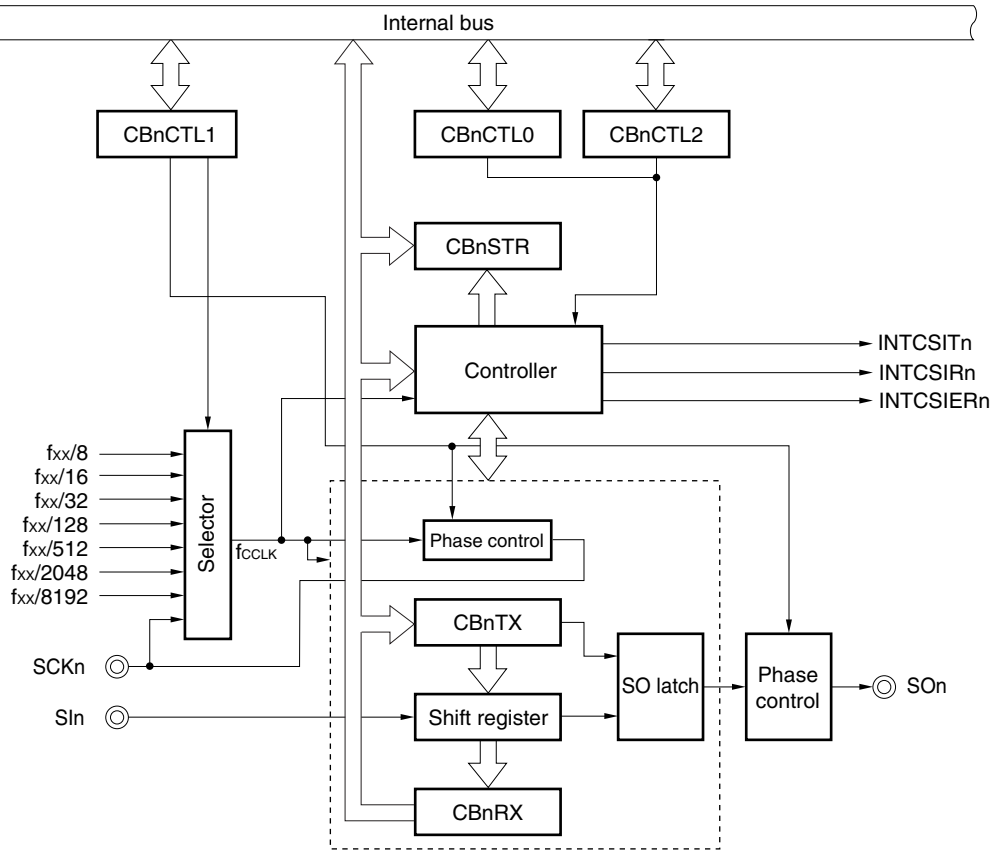
Note I^2C bus versions (Y products) only
When using the SDA and SCL pins, the pins function as dummy open-drain output pins (P-ch side is always off).

Remark x = don't care

17.3 Configuration

The following shows the block diagram of CSIBn.

Figure 17-3. Block Diagram of CSIBn



Caution CSIBn cannot be used if the CPU clock (f_{CPU}) is slower than f_{CCLK}.

Remark f_{CCLK}: Communication clock (10 MHz (max.))

CSIBn includes the following hardware.

Table 17-1. Configuration of CSIBn

Item	Configuration
Registers	CSIBn receive data register (CBnRX)
	CSIBn transmit data register (CBnTX)
Control registers	CSIBn control register 0 (CBnCTL0)
	CSIBn control register 1 (CBnCTL1)
	CSIBn control register 2 (CBnCTL2)
	CSIBn status register (CBnSTR)

(1) CSIBn receive data register (CBnRX)

The CBnRX register is a 16-bit buffer register that holds receive data.

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

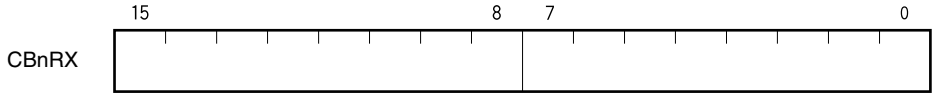
The receive operation is started by reading the CBnRX register in the reception enabled status.

If the transfer data length is 8 bits, the lower 8 bits of this register are read-only in 8-bit units as the CBnRXL register.

Reset input clears this register to 0000H.

In addition to reset input, the CBnRX register can be initialized by clearing (to 0) the CBnCTL0.CBnPWR bit.

After reset: 0000H R Address: CB0RX FFFFFD04H, CB0RXL FFFFFD04H,
CB1RX FFFFFD14H, CB1RXL FFFFFD14H,
CB2RX FFFFFD24H, CB2RXL FFFFFD24H



(2) CSIBn transmit data register (CBnTX)

The CBnTX register is a 16-bit buffer register used to write the CSIBn transfer data.

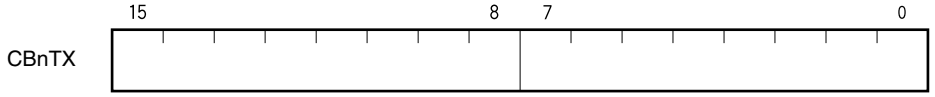
This register can be read or written in 16-bit units.

The transmit operation is started by writing data to the CBnTX register in the transmission enabled status.

If the transfer data length is 8 bits, the lower 8 bits of this register are read-only in 8-bit units as the CBnTXL register.

Reset input clears this register to 0000H.

After reset: 0000H R/W Address: CB0TX FFFFFD06H, CB0TXL FFFFFD06H,
CB1TX FFFFFD16H, CB1TXL FFFFFD16H,
CB2TX FFFFFD26H, CB2TXL FFFFFD26H



Remark The communication start conditions are shown below.

Transmission mode (CBnTXE bit = 1, CBnRXE bit = 0):	Write to CBnTX register
Transmission/reception mode (CBnTXE bit = 1, CBnRXE bit = 1):	Write to CBnTX register
Reception mode (CBnTXE bit = 0, CBnRXE bit = 1):	Read from CBnRX register

17.4 Control Registers

The following registers are used to control CSIBn.

- CSIBn control register 0 (CBnCTL0)
- CSIBn control register 1 (CBnCTL1)
- CSIBn control register 2 (CBnCTL2)
- CSIBn status register (CBnSTR)

(1) CSIBn control register 0 (CBnCTL0)

CBnCTL0 is a register that controls the CSIBn serial transfer operation.

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

Reset input sets this register to 01H.

(1/2)

After reset: 01H R/W Address: CB0CTL0 FFFFFD00H, CB1CTL0 FFFFFD10H, CB2CTL0 FFFFFD20H

CBnCTL0 (n = 0 to 2)

<7>	<6>	<5>	<4>	3	2	1	<0>
CBnPWR	CBnTXE[Note]	CBnRXE[Note]	CBnDIR[Note]	0	0	CBnTMS[Note]	CBnSCE

CBnPWR	Specification of CSIBn operation disable/enable
0	Disable CSIBn operation and reset the CBnSTR register
1	Enable CSIBn operation
• The CBnPWR bit controls the CSIBn operation and resets the internal circuit.	

CBnTXE[Note]	Specification of transmit operation disable/enable
0	Disable transmit operation
1	Enable transmit operation
• The SOn output is low level when the CBnTXE bit is 0.	

CBnRXE[Note]	Specification of receive operation disable/enable
0	Disable receive operation
1	Enable receive operation
• When the CBnRXE bit is cleared to 0, no reception end interrupt is output even when the prescribed data is transferred in order to disable the receive operation, and the receive data (CBnRX register) is not updated.	

Note These bits can only be rewritten when the CBnPWR bit = 0. However, CBnPWR bit = 1 can also be set at the same time as rewriting these bits.

Caution Be sure to clear bits 3 and 2 to 0.

CBnDIR[Note 1]	Specification of transfer direction mode (MSB/LSB)
0	MSB first
1	LSB first

CBnTMS[Note 1]	Transfer mode specification
0	Single transfer mode
1	Continuous transfer mode

<R> • When using single transmission or transmission/reception mode with communication type 2 or 4 (CBnCTL1.CBnDAP bit = 1), write the transfer data to the CBnTX register after checking that the CBnSTR.CBnTSF bit is 0.

<R> • When using DMA, use the continuous transfer mode.

CBnSCE	Specification of start transfer disable/enable
0	Communication start trigger invalid
1	Communication start trigger valid

- In master mode
  This bit enables or disables the communication start trigger.
  (a) In single reception mode
  Clear the CBnSCE bit to 0 before reading the receive data (CBnRX register)[Note 2].
  (b) In continuous reception mode
  Clear the CBnSCE bit to 0 one communication clock before reception of the last data is ended[Note 3].
- In slave mode
  This bit enables or disables the communication start trigger.
  (a) In single reception mode, or continuous reception mode
  Set the CBnSCE bit to 1[Note 4].

<R> • In single transmission or transmission/reception mode, or continuous transmission mode, or transmission/reception mode
The function of the CBnSCE bit is invalid. It is recommended to set this bit to 1.

Notes 1. These bits can only be rewritten when the CBnPWR bit = 0. However, the CBnPWR can be set to 1 at the same time as these bits are rewritten.

2. If the CBnSCE bit is read while it is 1, the next communication operation is started.

3. The CBnSCE bit is not cleared to 0 one communication clock before the end of the last data reception, the next communication operation is automatically started.

<R> To start communication operation again after reading the last data, set the CBnSCE bit to 1 and perform a dummy read of the CBnRX register.

<R> **4.** To start the reception, a dummy read is necessary.

<R> **(a) How to use CBnSCE bit**

(i) In single reception mode

<1> When the reception of the last data is completed with INTCSIRn interrupt servicing, clear the CBnSCE bit to 0, and then read the CBnRX register.

<2> When the reception is disabled after the reception of the last data has been completed, check that the CBnSTR.CBnTSF bit is 0, and then clear the CBnPWR and CBnRXE bits to 0. To continue reception, set the CBnSCE bit to 1 and start the next receive operation by performing a dummy read of the CBnRX register.

(ii) In continuous reception mode

<1> Clear the CBnSCE bit to 0 during reception of the last data with INTCSIRn interrupt servicing by the reception before the last reception, and then read the CBnRX register.

<2> After receiving the INTCSIRn signal of the last reception, read the last data from the CBnRX register.

<3> When the reception is disabled after the reception of the last data has been completed, check that the CBnSTR.CBnTSF bit is 0, and then clear the CBnPWR and CBnRXE bits to 0. To continue reception, set the CBnSCE bit to 1 and start the next receive operation by performing a dummy read of the CBnRX register.

Caution In continuous reception mode, the serial clock is not stopped until the reception executed when the CBnSCE bit is cleared to 0 is completed after the reception is started by a dummy read.

(2) CSIBn control register 1 (CBnCTL1)

CBnCTL1 is an 8-bit register that controls the CSIBn serial transfer operation.

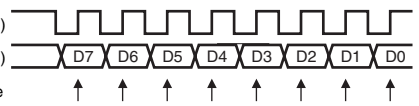
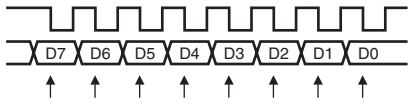
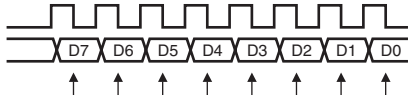
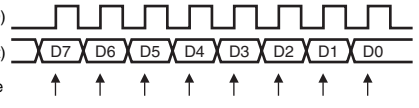
This register can be read or written in 8-bit or 1-bit units.

Reset input clears this register to 00H.

Caution The CBnCTL1 register can be rewritten only when the CBnTL0.CBnPWR bit.

After reset: 00H R/W Address: CB0CTL1 FFFFFD01H, CB1CTL1 FFFFFD11H, CB2CTL1 FFFFFD21H

	7	6	5	4	3	2	1	0
CBnCTL1 (n = 0 to 2)	0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0

	CBnCKP	CBnDAP	Specification of data transmission/ reception timing in relation to SCKn
Communication type 1	0	0	
Communication type 2	0	1	
Communication type 3	1	0	
Communication type 4	1	1	

CBnCKS2	CBnCKS1	CBnCKS0	Communication clock (f_{CCLK})	Mode
0	0	0	$f_{XX}/8$	Master mode
0	0	1	$f_{XX}/16$	Master mode
0	1	0	$f_{XX}/32$	Master mode
0	1	1	$f_{XX}/128$	Master mode
1	0	0	$f_{XX}/512$	Master mode
1	0	1	$f_{XX}/2048$	Master mode
1	1	0	$f_{XX}/8192$	Master mode
1	1	1	External clock (SCKn)	Slave mode

Caution Set f_{CCLK} to 10 MHz or lower.

(3) CSIBn control register 2 (CBnCTL2)

CBnCTL2 is an 8-bit register that controls the number of CSIBn serial transfer bits.

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

Reset input clears this register to 00H.

Caution The CBnCTL2 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0 or when both the CBnTXE and CBnRXE bits = 0.

After reset: 00H R/W Address: CB0CTL2 FFFFFD02H, CB1CTL2 FFFFFD12H, CB2CTL2 FFFFFD22H

CBnCTL2 (n = 0 to 2)

7	6	5	4	3	2	1	0
0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0

CBnCL3	CBnCL2	CBnCL1	CBnCL0	Serial register bit length
0	0	0	0	8 bits
0	0	0	1	9 bits
0	0	1	0	10 bits
0	0	1	1	11 bits
0	1	0	0	12 bits
0	1	0	1	13 bits
0	1	1	0	14 bits
0	1	1	1	15 bits
1	×	×	×	16 bits

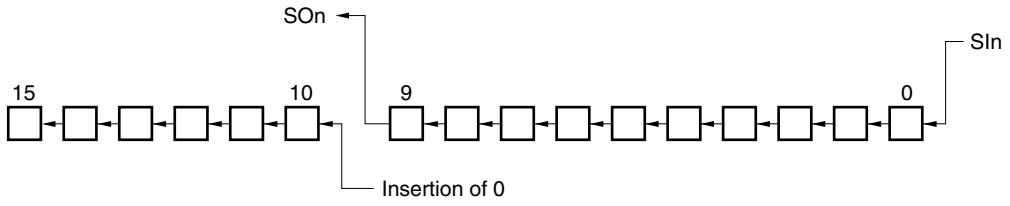
Remark If the number of transfer bits is other than 8 or 16, prepare and use data stuffed from the LSB of the CBnTX and CBnRX registers.

(a) Transfer data length change function

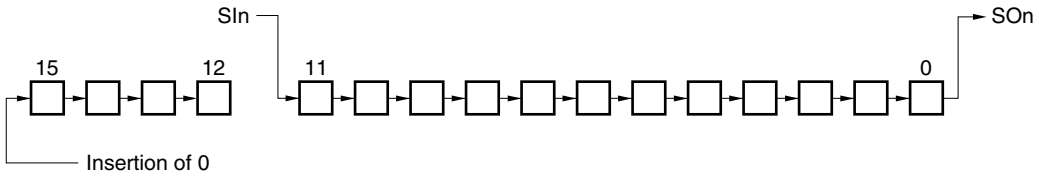
The CSIBn transfer data length can be set in 1-bit units between 8 and 16 bits using the CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits.

When the transfer bit length is set to a value other than 16 bits, set the data to the CBnTX or CBnRX register starting from the LSB, regardless of whether the transfer start bit is the MSB or LSB. Any data can be set for the higher bits that are not used, but the receive data becomes 0 following serial transfer.

(i) Transfer bit length = 10 bits, MSB first



(ii) Transfer bit length = 12 bits, LSB first



(4) CSIBn status register (CBnSTR)

CBnSTR is an 8-bit register that displays the CSIBn status.

This register can be read or written in 8-bit or 1-bit units, but the CBnTSF flag is read-only.

Reset input clears this register to 00H.

In addition to reset input, the CBnSTR register can be initialized by clearing (0) the CBnCTL0.CBnPWR bit.

After reset: 00H R/W Address: CB0STR FFFFFD03H, CB1STR FFFFFD13H, CB2STR FFFFFD23H

CBnSTR (n = 0 to 2)

<7>	6	5	4	3	2	1	<0>
CBnTSF	0	0	0	0	0	0	CBnOVE

CBnTSF	Communication status flag
0	Communication stopped
1	Communicating

- During transmission, this register is set when data is prepared in the CBnTX register, and during reception, it is set when a dummy read of the CBnRX register is performed.
  When transfer ends, this flag is cleared to 0 at the last edge of the clock.

CBnOVE	Overrun error flag
0	No overrun
1	Overrun

- An overrun error occurs when the next reception starts without performing a CPU read of the value of the CBnRX register, upon end of the receive operation.
  The CBnOVE flag displays the overrun error occurrence status in this case.
- The CBnOVE flag is cleared by writing 0 to it. It cannot be set even by writing 1 to it.

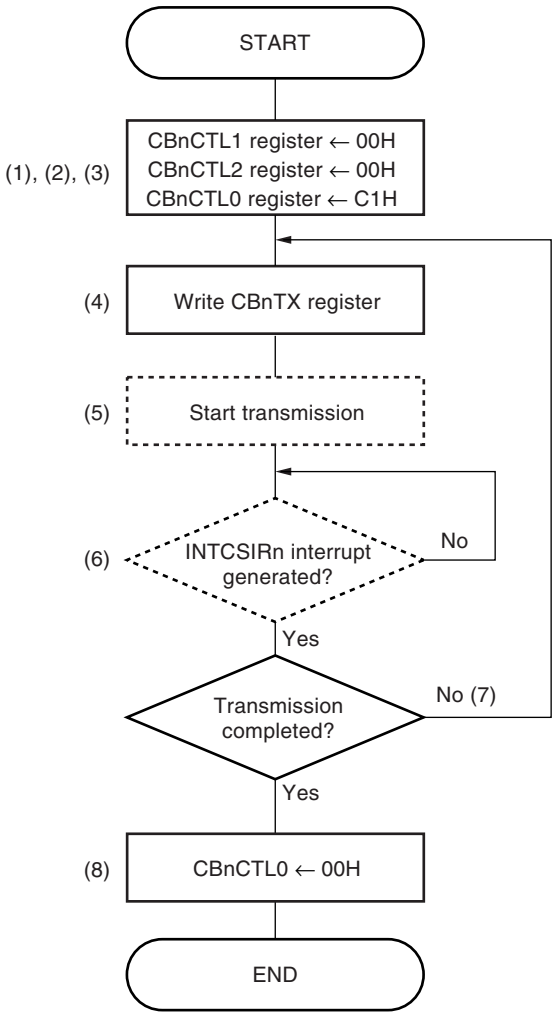
<R> **Caution In single transfer mode, writing to the CBnTX register with the CBnTSF bit set to 1 is ignored. This has no influence on the operation during transfer.**
For example, if the next data is written to the CBnTX register when DMA is started by generating the INTCSIRn signal, the written data is not transferred because the CBnTSF bit is set to 1.
Use the continuous transfer mode, not the single transfer mode, for such applications.

17.5 Operation

<R> **17.5.1 Single transfer mode (master mode, transmission mode)**

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/8 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

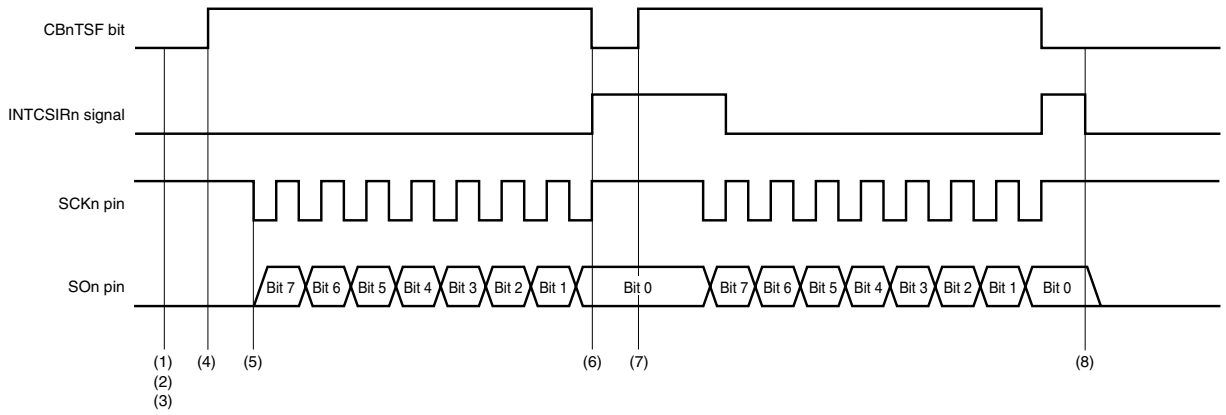


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0, 1

(2) Operation timing



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/8, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write C1H to the CBnCTL0 register, and select the transmission mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and transmission is started.

(5) When transmission is started, output the serial clock to the SCKn pin, and output the transmit data from the SOn pin in synchronization with the serial clock.

(6) When transmission of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output and transmit data output, generate the reception completion interrupt request signal (INTCSIRn) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

(7) To continue transmission, start the next transmission by writing the transmit data to the CBnTX register again after the INTCSIRn signal is generated.

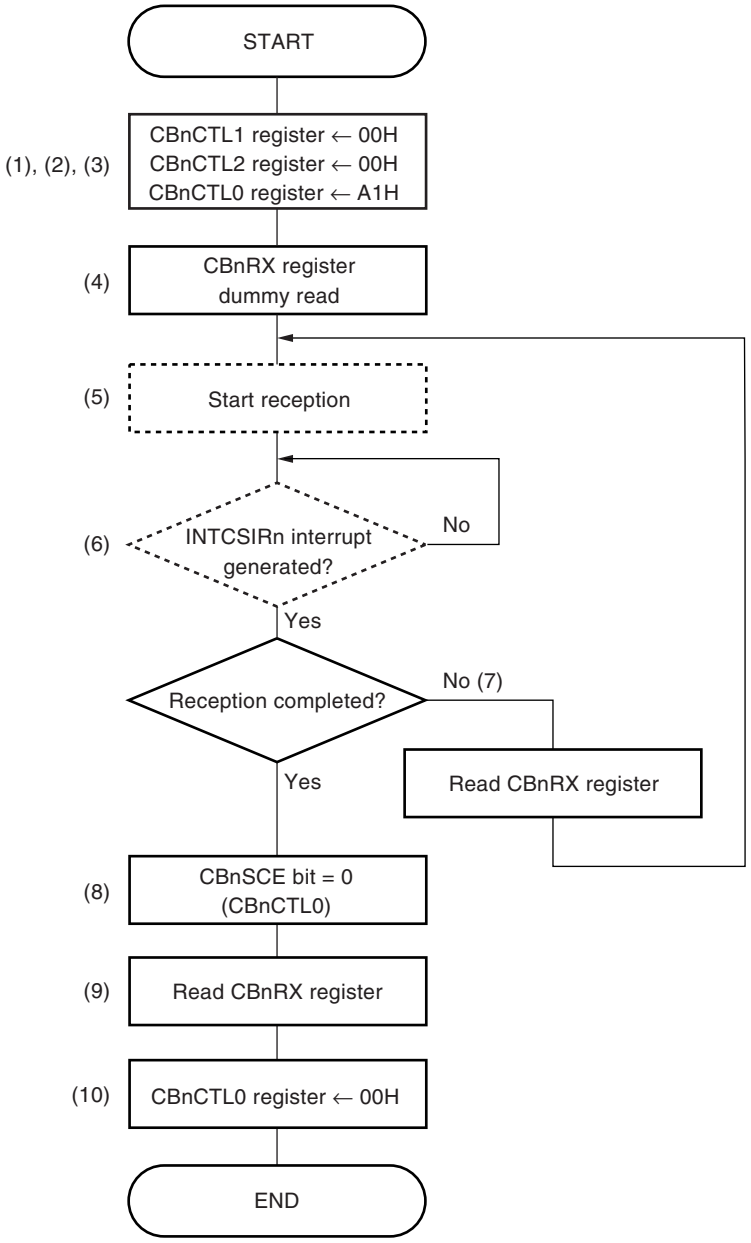
(8) To end transmission, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnTXE bit = 0.

Remark n = 0, 1

<R> **17.5.2 Single transfer mode (master mode, reception mode)**

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/8 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

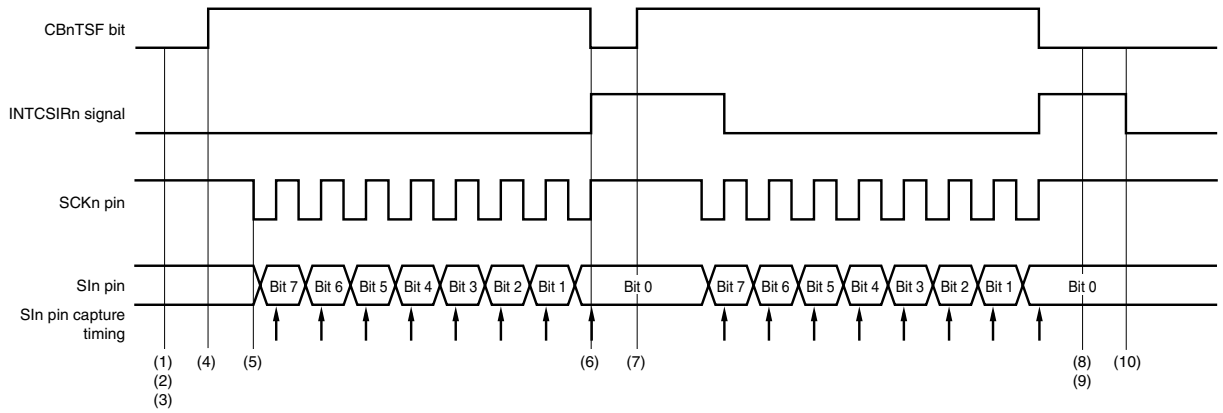


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0, 1

(2) Operation timing



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/8, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write A1H to the CBnCTL0 register, and select the reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by performing a dummy read of the CBnRX register, and reception is started.

(5) When reception is started, output the serial clock to the SCKn pin, and capture the receive data of the SIn pin in synchronization with the serial clock.

(6) When reception of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output and data capturing, generate the reception completion interrupt request signal (INTCSIRn) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

(7) To continue reception, read the CBnRX register with the CBnCTL0.CBnSCE bit = 1 remained after the INTCSIRn signal is generated.

(8) To read the CBnRX register without starting the next reception, write the CBnSCE bit = 0.

(9) Read the CBnRX register.

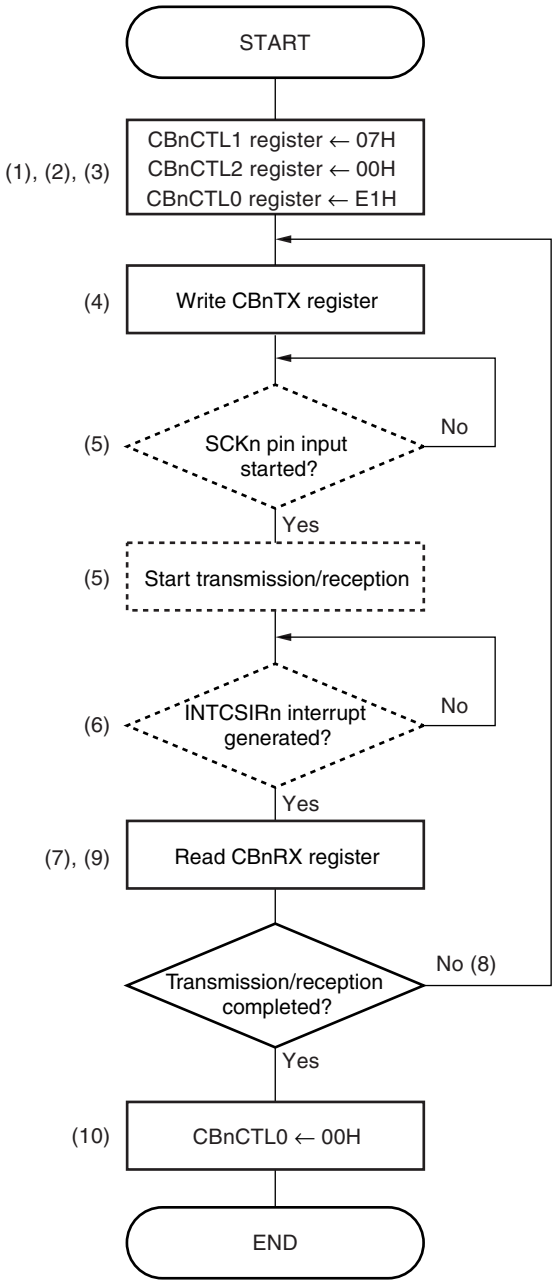
(10) To end reception, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnRXE bit = 0.

Remark n = 0, 1

<R> **17.5.3 Single transfer mode (master mode, transmission/reception mode)**

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/8 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

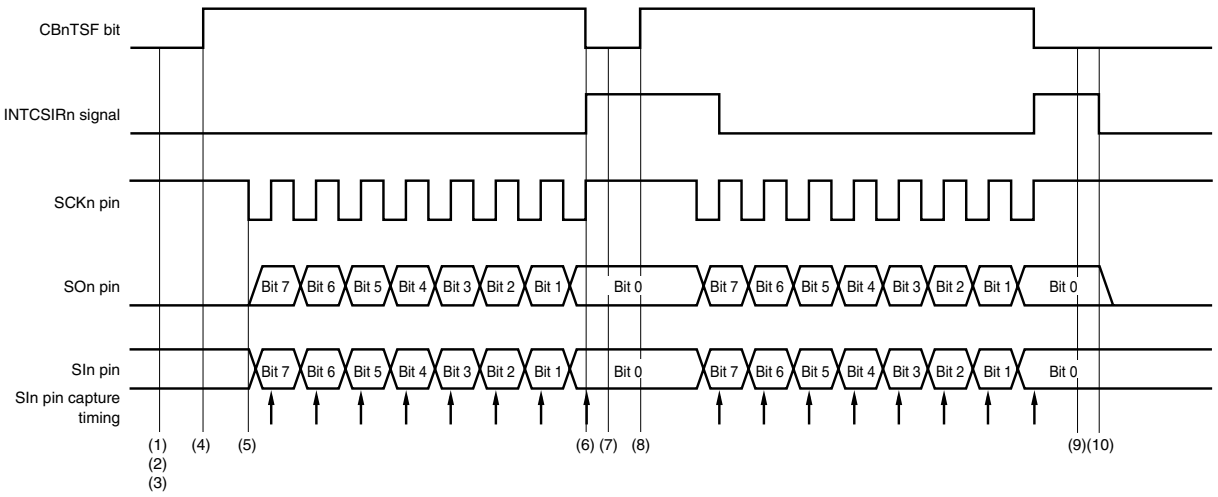


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0, 1

(2) Operation timing



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/8, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write E1H to the CBnCTL0 register, and select the transmission/reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and transmission/reception is started.

(5) When transmission/reception is started, output the serial clock to the SCKn pin, output the transmit data to the SOn pin in synchronization with the serial clock, and capture the receive data of the SIn pin.

(6) When transmission/reception of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output, transmit data output, and data capturing, generate the reception completion interrupt request signal (INTCSIRn) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

(7) Read the CBnRX register.

(8) To continue transmission/reception, write the transmit data to the CBnTX register again.

(9) Read the CBnRX register.

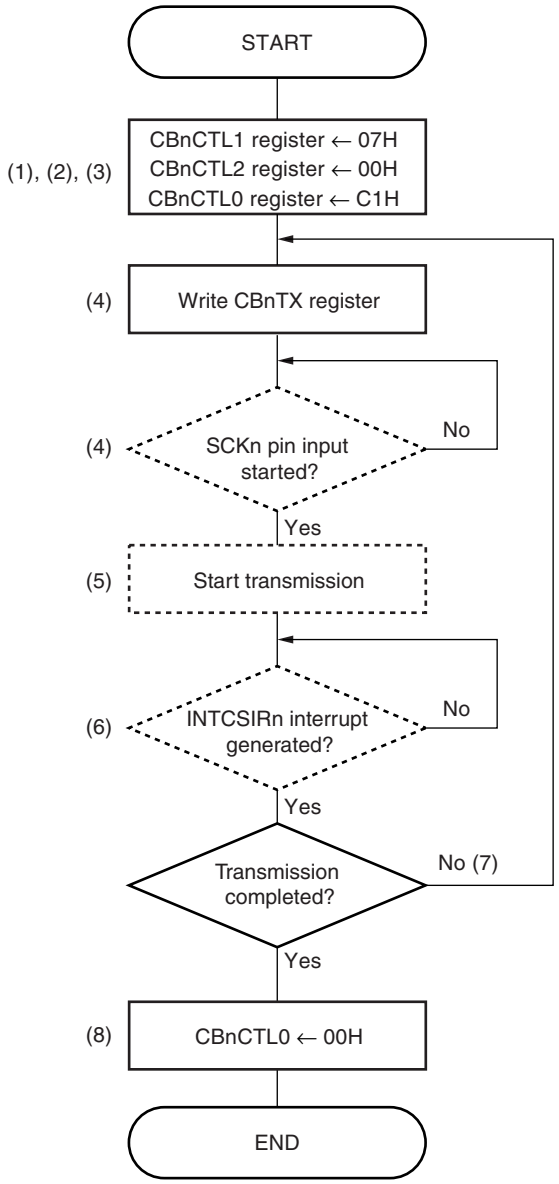
(10) To end transmission/reception, write the CBnCTL0.CBnPWR bit = 0, the CBnCTL0.CBnTXE bit = 0, and the CBnCTL0.CBnRXE bit = 0.

Remark n = 0, 1

<R> **17.5.4 Single transfer mode (slave mode, transmission mode)**

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = external clock (SCKn) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

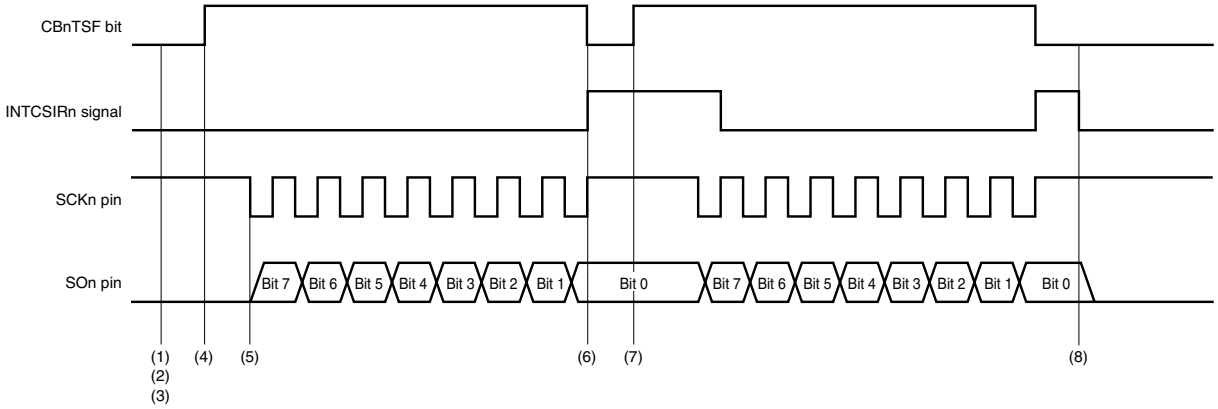


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0, 1

(2) Operation timing



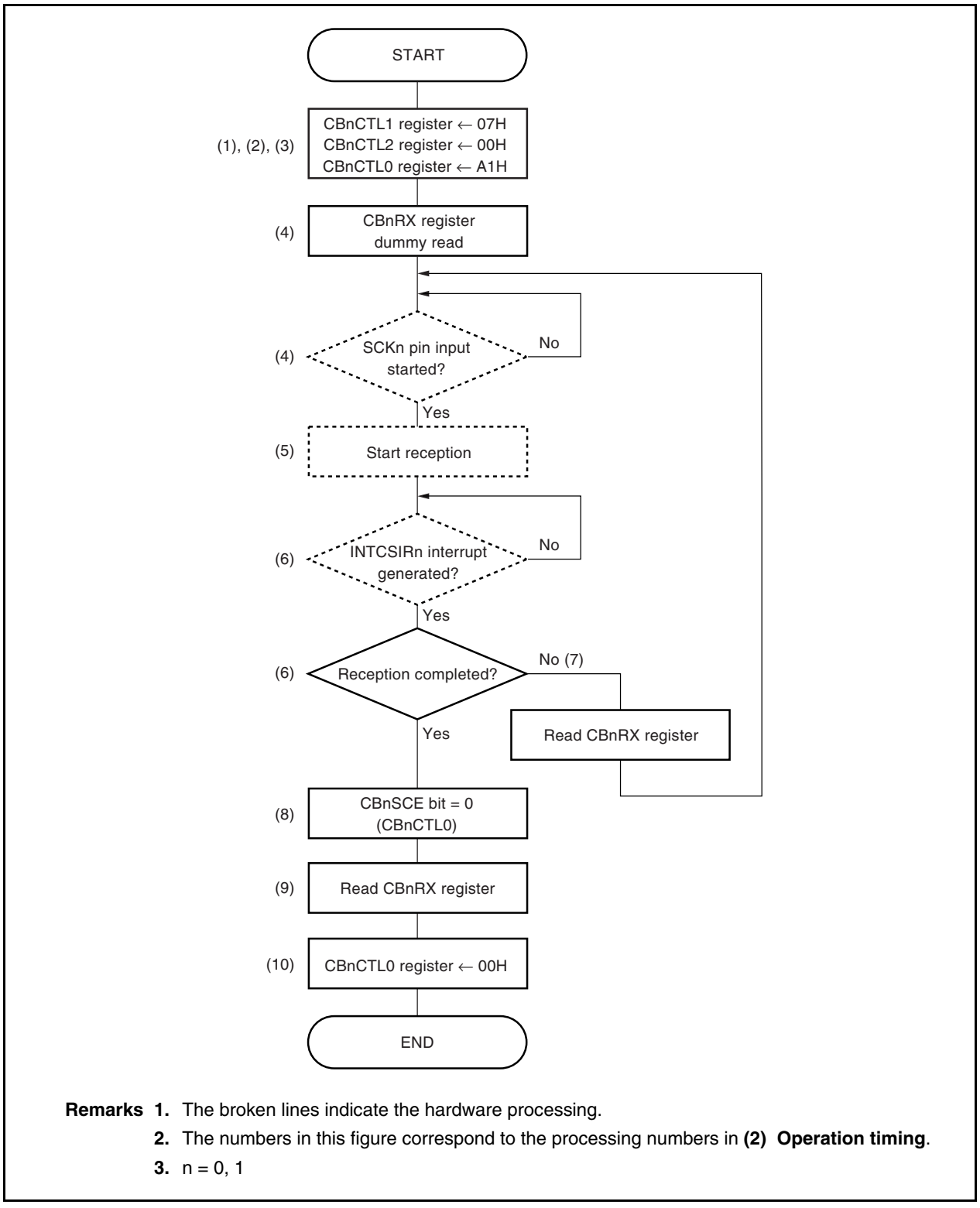
(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock (SCKn), and slave mode.
(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.
(3) Write C1H to the CBnCTL0 register, and select the transmission mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).
(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and the device waits for a serial clock input.
(5) When a serial clock is input, output the transmit data from the SOn pin in synchronization with the serial clock.
(6) When transmission of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output and transmit data output, generate the reception completion interrupt request signal (INTCSIRn) at the last edge of the serial clock, and clear the CBnTSF bit to 0.
(7) To continue transmission, write the transmit data to the CBnTX register again after the INTCSIRn signal is generated, and wait for a serial clock input.
(8) To end transmission, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnTXE bit = 0.

Remark n = 0, 1

<R> **17.5.5 Single transfer mode (slave mode, reception mode)**

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = external clock (SCKn) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

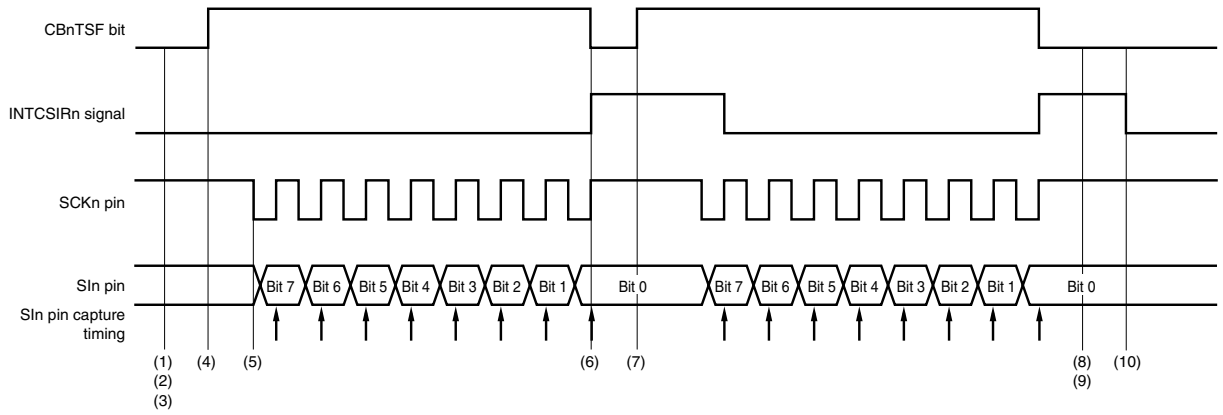


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0, 1

(2) Operation timing



(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock (SCKn), and slave mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write A1H to the CBnCTL0 register, and select the reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by performing a dummy read of the CBnRX register, and the device waits for a serial clock input.

(5) When a serial clock is input, capture the receive data of the SIn pin in synchronization with the serial clock.

(6) When reception of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output and data capturing, generate the reception completion interrupt request signal (INTCSIRn) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

(7) To continue reception, read the CBnRX register with the CBnCTL0.CBnSCE bit = 1 remained after the INTCSIRn signal is generated, and wait for a serial clock input.

(8) To end reception, write the CBnSCE bit = 0.

(9) Read the CBnRX register.

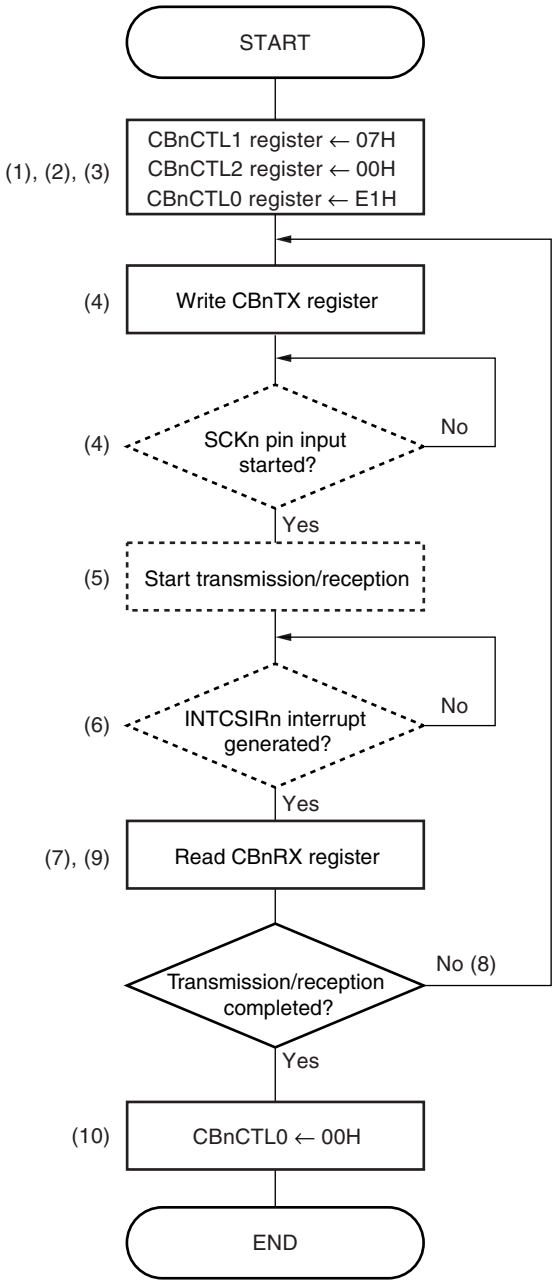
(10) To end reception, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnRXE bit = 0.

Remark n = 0, 1

<R> **17.5.6 Single transfer mode (slave mode, transmission/reception mode)**

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = external clock (SCKn) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

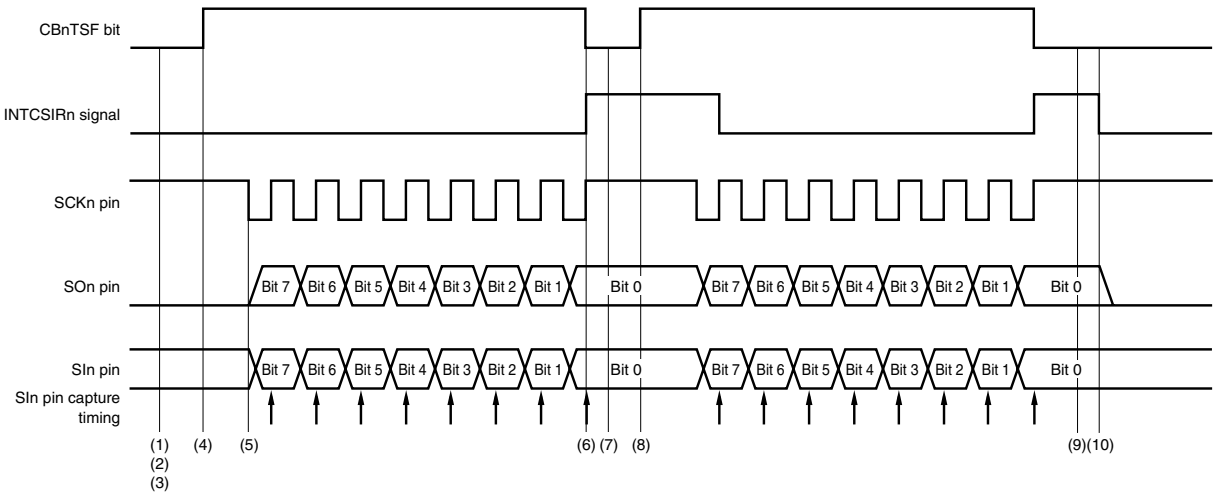


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0, 1

(2) Operation timing



(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock (SCKn), and slave mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write E1H to the CBnCTL0 register, and select the transmission/reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data to the SOn pin in synchronization with the serial clock, and capture the receive data of the SIn pin.

(6) When transmission/reception of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output, transmit data output, and data capturing, generate the reception completion interrupt request signal (INTCSIRn) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

(7) Read the CBnRX register.

(8) To continue transmission/reception, write the transmit data to the CBnTX register again, and wait for a serial clock input.

(9) Read the CBnRX register.

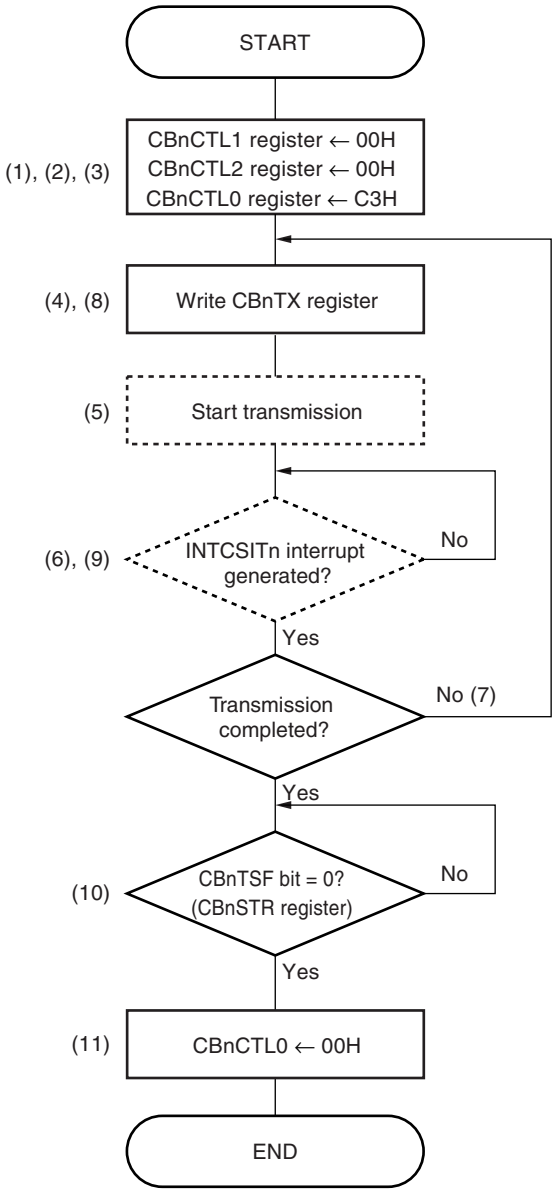
(10) To end transmission/reception, write the CBnCTL0.CBnPWR bit = 0, the CBnCTL0.CBnTXE bit = 0, and the CBnCTL0.CBnRXE bit = 0.

Remark n = 0, 1

<R> **17.5.7 Continuous transfer mode (master mode, transmission mode)**

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/8 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

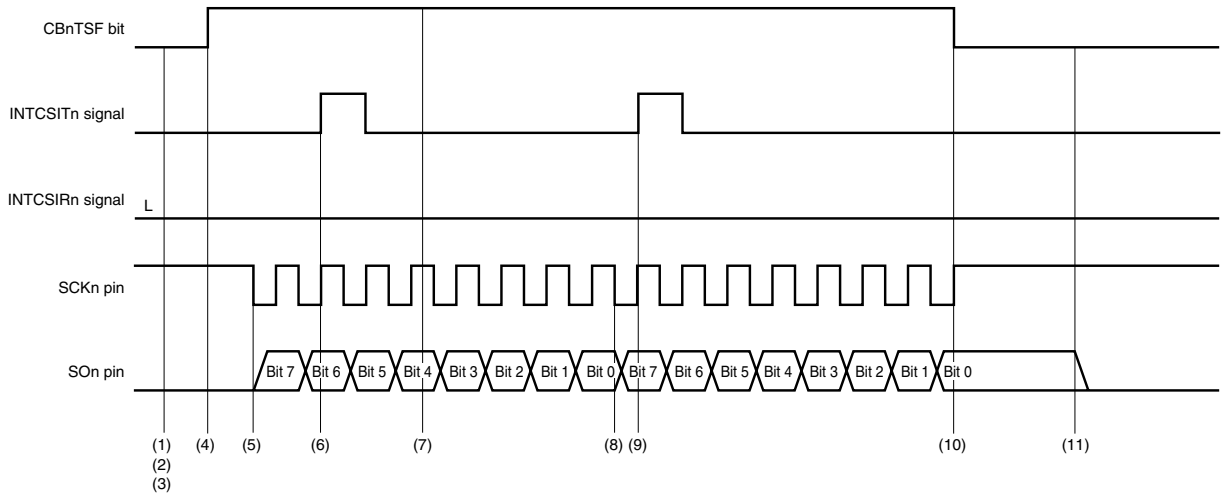


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0, 1

(2) Operation timing



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/8, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write C3H to the CBnCTL0 register, and select the transmission mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and transmission is started.

(5) When transmission is started, output the serial clock to the SCKn pin, and output the transmit data from the SOn pin in synchronization with the serial clock.

(6) When transfer of the transmit data from the CBnTX register to the shift register is completed and writing to the CBnTX register is enabled, the transmission enable interrupt request signal (INTCSITn) is generated.

(7) To continue transmission, write the transmit data to the CBnTX register again after the INTCSITn signal is generated.

(8) When a new transmit data is written to the CBnTX register before communication completion, the next communication is started following communication completion.

(9) The transfer of the transmit data from the CBnTX register to the shift register is completed and the INTCSITn signal is generated. To end continuous transmission with the current transmission, do not write to the CBnTX register.

(10) When the next transmit data is not written to the CBnTX register before transfer completion, stop the serial clock output to the SCKn pin after transfer completion, and clear the CBnTSF bit to 0.

(11) To release the transmission enable status, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnTXE bit = 0 after checking that the CBnTSF bit = 0.

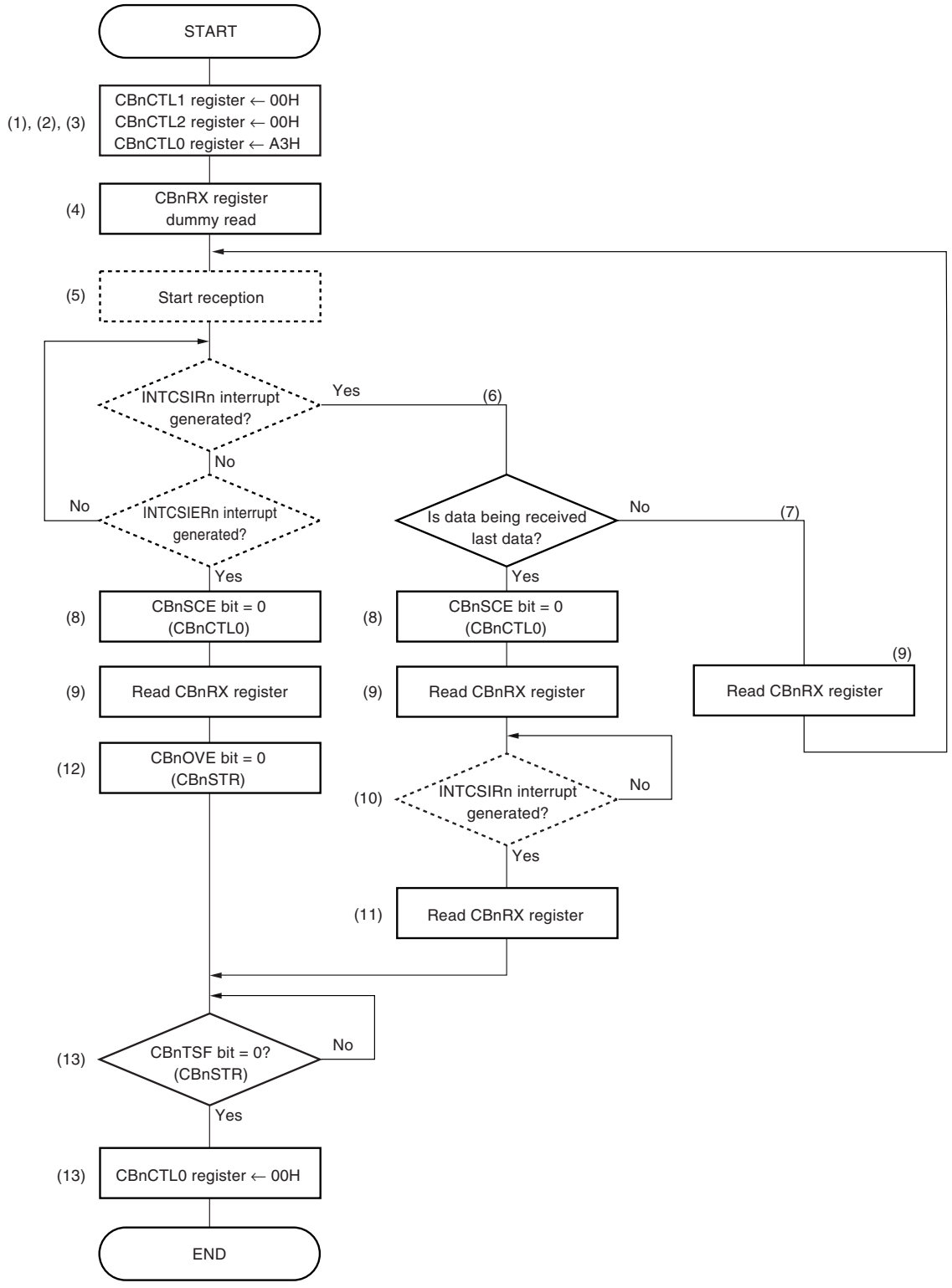
Caution In continuous transmission mode, the reception completion interrupt request signal (INTCSIRn) is not generated.

Remark n = 0, 1

<R> **17.5.8 Continuous transfer mode (master mode, reception mode)**

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/8 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

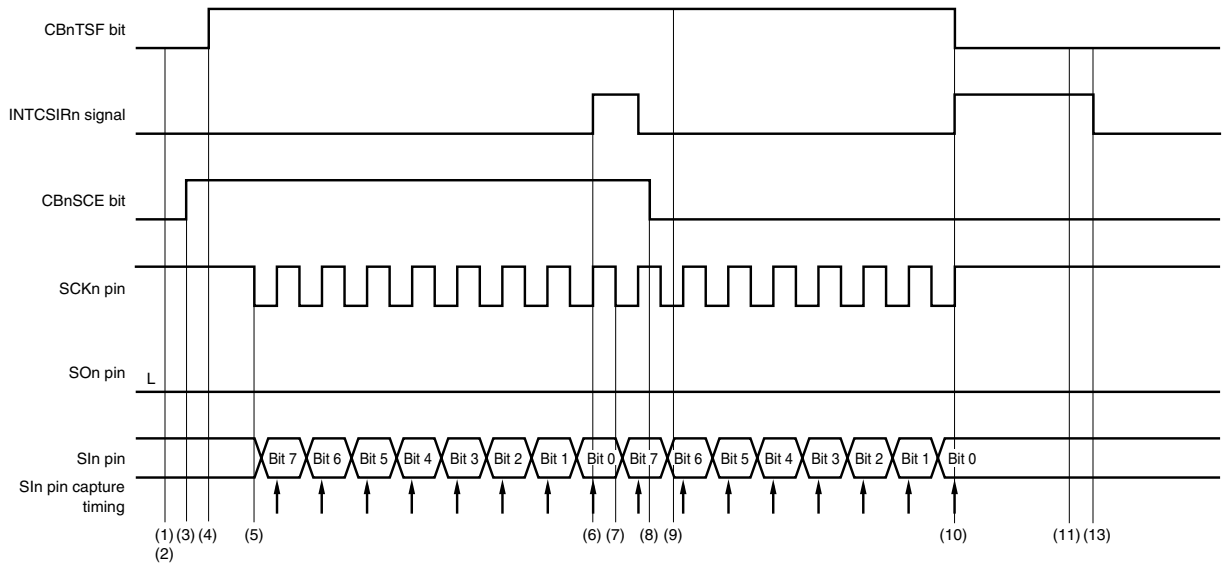


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0, 1

(2) Operation timing



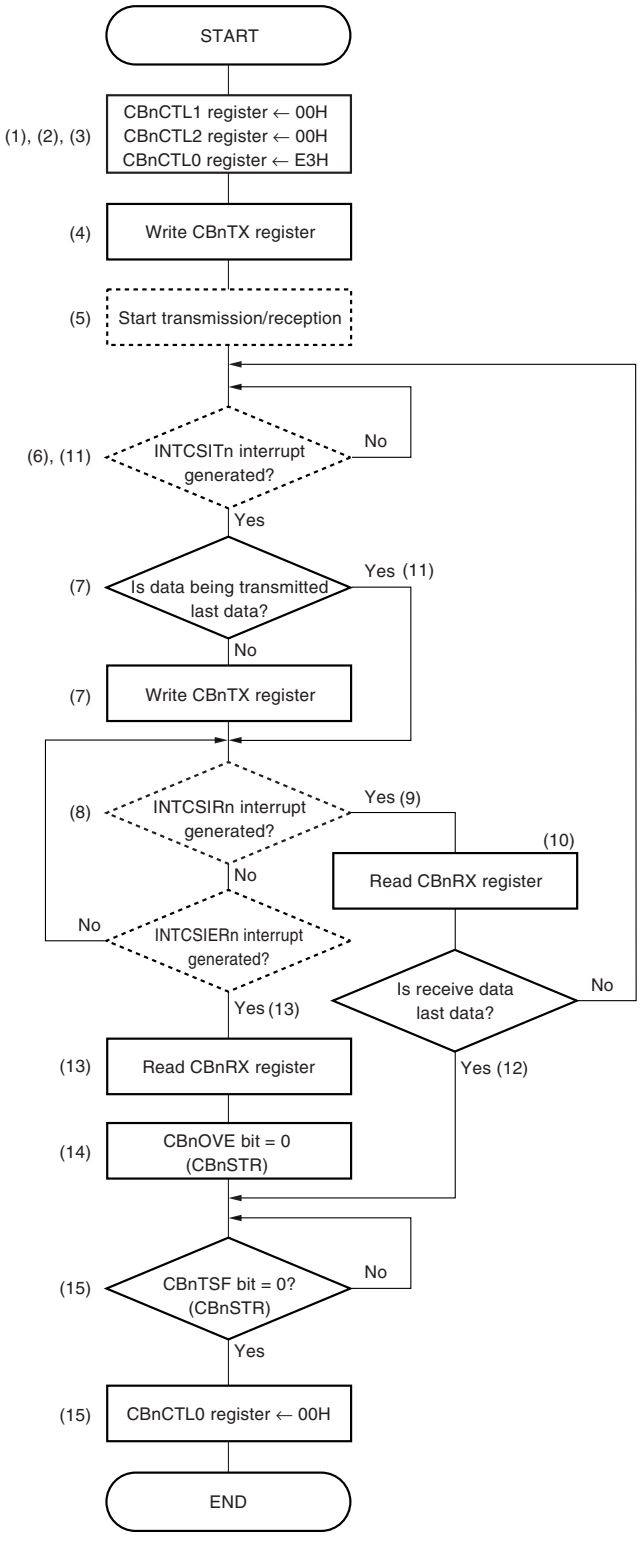
(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/8, and master mode.
(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.
(3) Write A3H to the CBnCTL0 register, and select the reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).
(4) The CBnSTR.CBnTSF bit is set to 1 by performing a dummy read of the CBnRX register, and reception is started.
(5) When reception is started, output the serial clock to the SCKn pin, and capture the receive data of the SIn pin in synchronization with the serial clock.
(6) When reception is completed, the reception completion interrupt request signal (INTCSIRn) is generated, and reading of the CBnRX register is enabled.
(7) When the CBnCTL0.CBnSCE bit = 1 upon communication completion, the next communication is started following communication completion.
(8) To end continuous reception with the current reception, write the CBnSCE bit = 0.
(9) Read the CBnRX register.
(10) When reception is completed, the INTCSIRn signal is generated, and reading of the CBnRX register is enabled. When the CBnSCE bit = 0 is set before communication completion, stop the serial clock output to the SCKn pin, and clear the CBnTSF bit to 0, to end the receive operation.
(11) Read the CBnRX register.
(12) If an overrun error occurs, write the CBnSTR.CBnOVE bit = 0, and clear the error flag.
(13) To release the reception enable status, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnRXE bit = 0 after checking that the CBnTSF bit = 0.

Remark n = 0, 1

<R> **17.5.9 Continuous transfer mode (master mode, transmission/reception mode)**

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = f_{XX}/8 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow



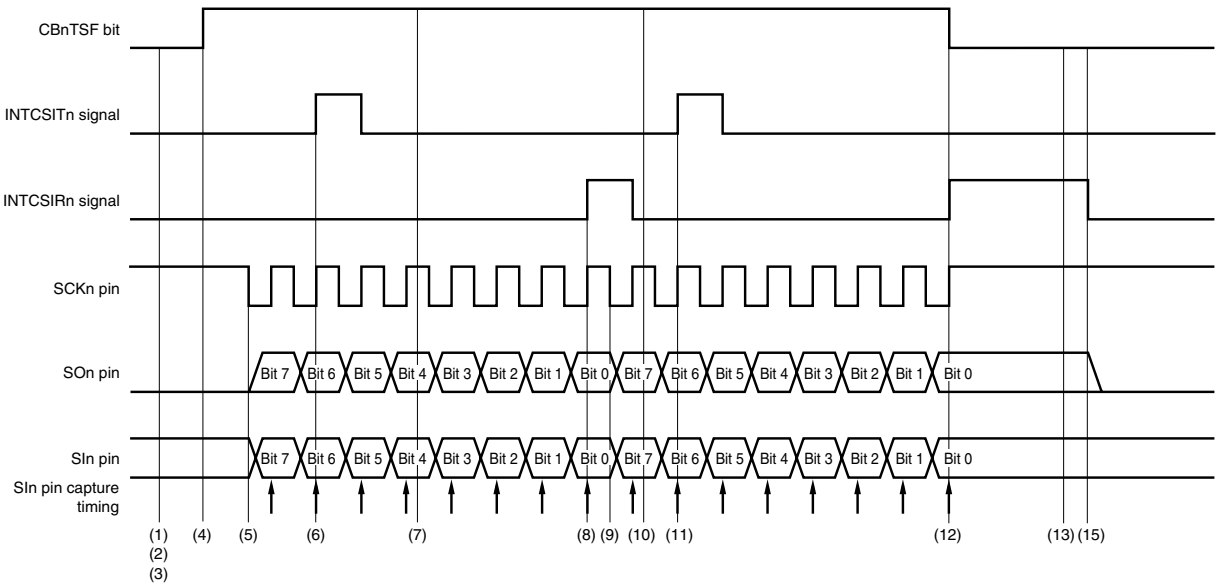
Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0, 1

(2) Operation timing

(1/2)



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/8, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write E3H to the CBnCTL0 register, and select the transmission/reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and transmission/reception is started.

(5) When transmission/reception is started, output the serial clock to the SCKn pin, output the transmit data to the SOn pin in synchronization with the serial clock, and capture the receive data of the SIn pin.

(6) When transfer of the transmit data from the CBnTX register to the shift register is completed and writing to the CBnTX register is enabled, the transmission enable interrupt request signal (INTCSITn) is generated.

(7) To continue transmission/reception, write the transmit data to the CBnTX register again after the INTCSITn signal is generated.

(8) When one transmission/reception is completed, the reception completion interrupt request signal (INTCSIRn) is generated, and reading of the CBnRX register is enabled.

(9) When a new transmit data is written to the CBnTX register before communication completion, the next communication is started following communication completion.

(10) Read the CBnRX register.

Remark n = 0, 1

(2/2)

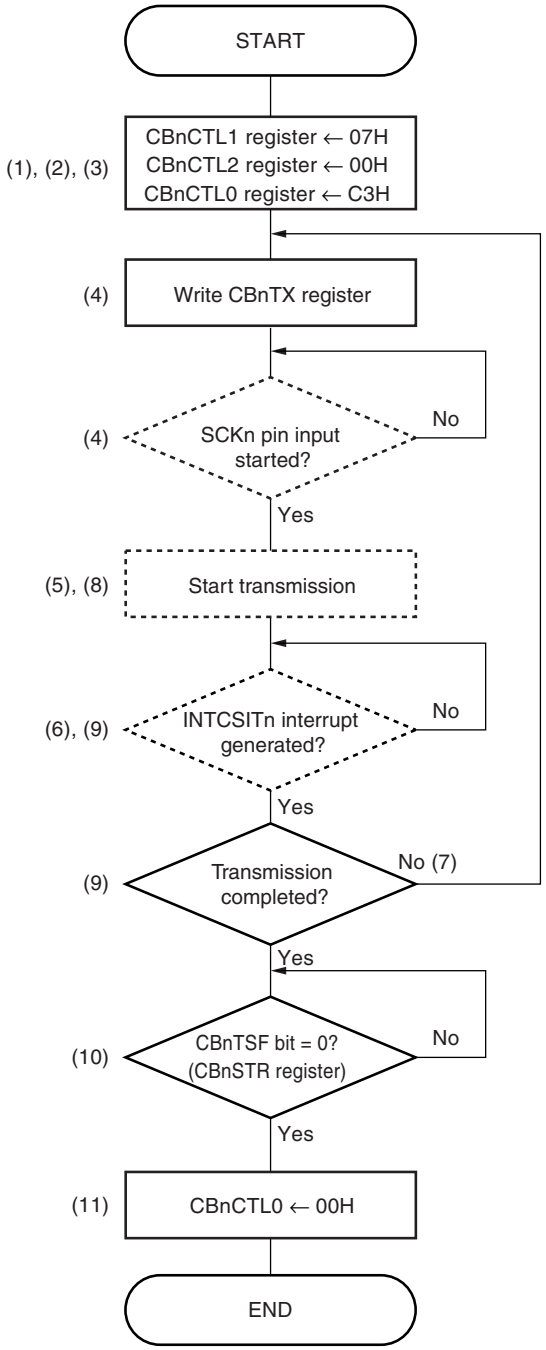
(11) The transfer of the transmit data from the CBnTX register to the shift register is completed and the INTCSITn signal is generated. To end continuous transmission/reception with the current transmission/reception, do not write to the CBnTX register.
(12) When the next transmit data is not written to the CBnTX register before transfer completion, stop the serial clock output to the SCKn pin after transfer completion, and clear the CBnTSF bit to 0.
(13) When the reception error interrupt request signal (INTCSIERn) is generated, read the CBnRX register.
(14) If an overrun error occurs, write the CBnSTR.CBnOVE bit = 0, and clear the error flag.
(15) To release the transmission/reception enable status, write the CBnCTL0.CBnPWR bit = 0, the CBnCTL0.CBnTXE bit = 0, and the CBnCTL0.CBnRXE bit = 0 after checking that the CBnTSF bit = 0.

Remark n = 0, 1

<R> 17.5.10 Continuous transfer mode (slave mode, transmission mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = external clock (SCKn) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

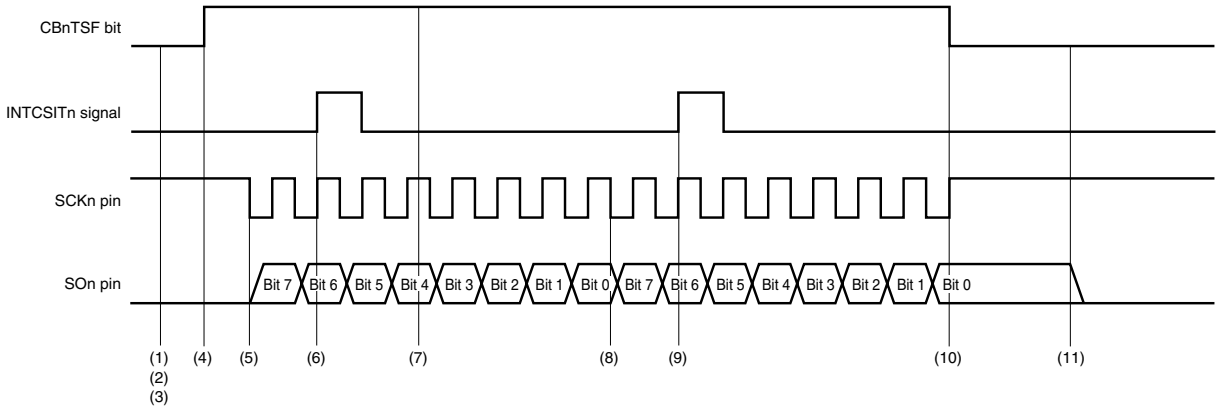


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0, 1

(2) Operation timing



(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock (SCKn), and slave mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write C3H to the CBnCTL0 register, and select the transmission mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data from the SOn pin in synchronization with the serial clock.

(6) When transfer of the transmit data from the CBnTX register to the shift register is completed and writing to the CBnTX register is enabled, the transmission enable interrupt request signal (INTCSITn) is generated.

(7) To continue transmission, write the transmit data to the CBnTX register again after the INTCSITn signal is generated.

(8) When a serial clock is input following completion of the transmission of the transfer data length set with the CBnCTL2 register, continuous transmission is started.

(9) When transfer of the transmit data from the CBnTX register to the shift register is completed and writing to the CBnTX register is enabled, the INTCSITn signal is generated. To end continuous transmission with the current transmission, do not write to the CBnTX register.

(10) When the clock of the transfer data length set with the CBnCTL2 register is input without writing to the CBnTX register, clear the CBnTSF bit to 0 to end transmission.

(11) To release the transmission enable status, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnTXE bit = 0 after checking that the CBnTSF bit = 0.

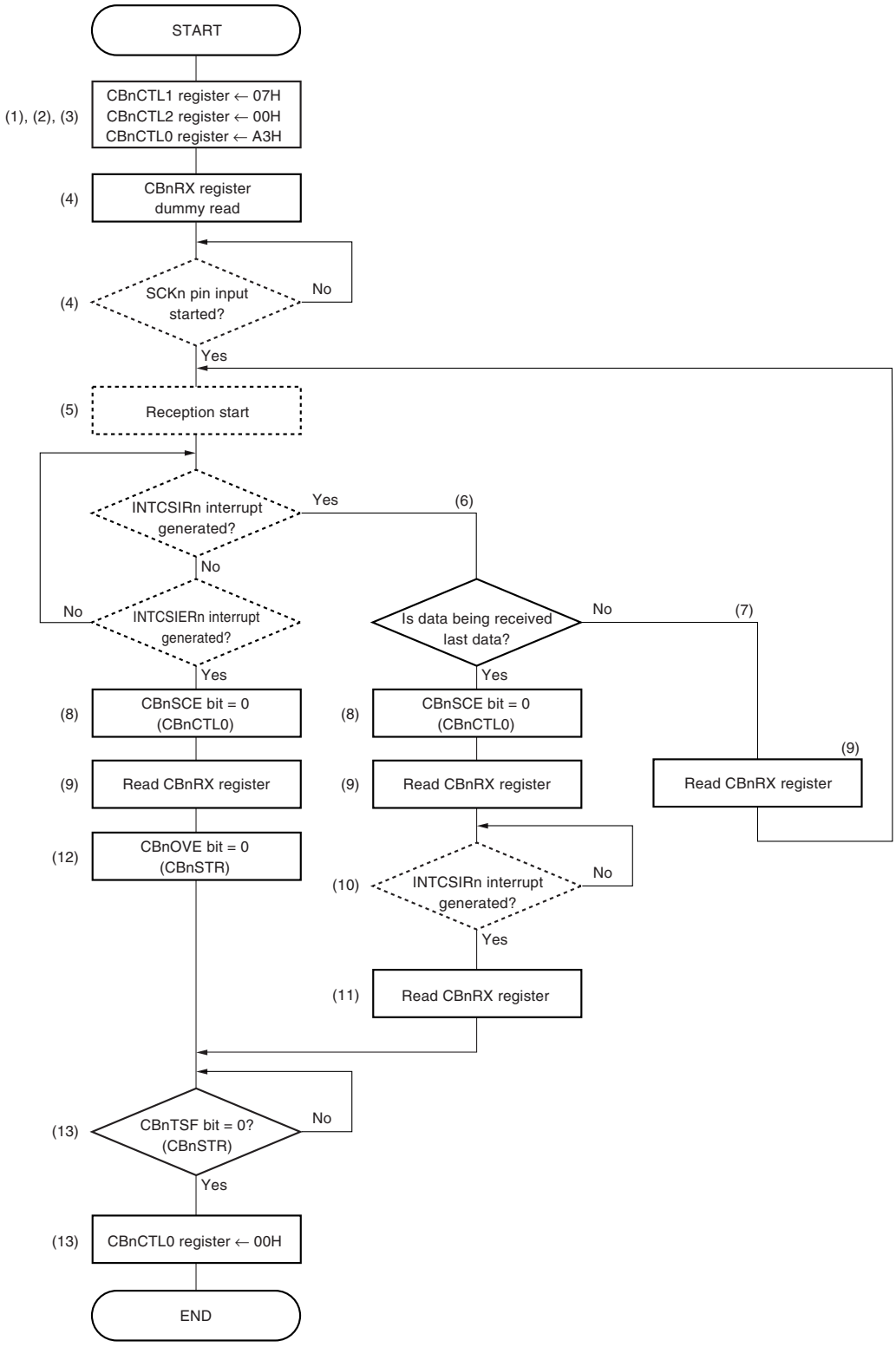
Caution In continuous transmission mode, the reception completion interrupt request signal (INTCSIRn) is not generated.

Remark n = 0, 1

<R> 17.5.11 Continuous transfer mode (slave mode, reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = external clock (SCKn) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow

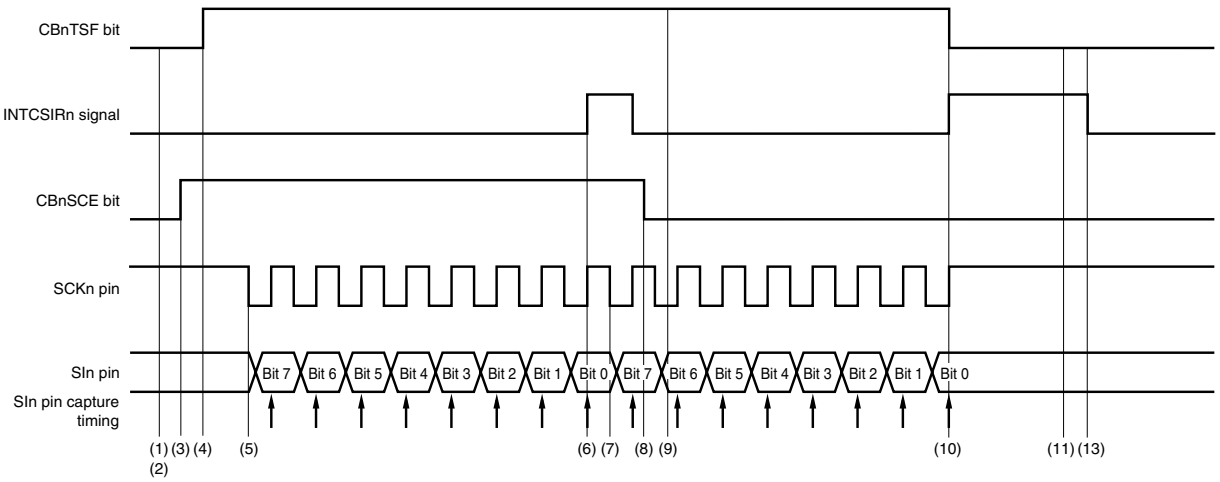


Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0, 1

(2) Operation timing



(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock (SCKn), and slave mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write A3H to the CBnCTL0 register, and select the reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by performing a dummy read of the CBnRX register, and the device waits for a serial clock input.

(5) When a serial clock is input, capture the receive data of the SIn pin in synchronization with the serial clock.

(6) When reception is completed, the reception completion interrupt request signal (INTCSIRn) is generated, and reading of the CBnRX register is enabled.

(7) When a serial clock is input in the CBnCTL0.CBnSCE bit = 1 status, continuous reception is started.

(8) To end continuous reception with the current reception, write the CBnSCE bit = 0.

(9) Read the CBnRX register.

(10) When reception is completed, the INTCSIRn signal is generated, and reading of the CBnRX register is enabled. When the CBnSCE bit = 0 is set before communication completion, clear the CBnTSF bit to 0 to end the receive operation.

(11) Read the CBnRX register.

(12) If an overrun error occurs, write the CBnSTR.CBnOVE bit = 0, and clear the error flag.

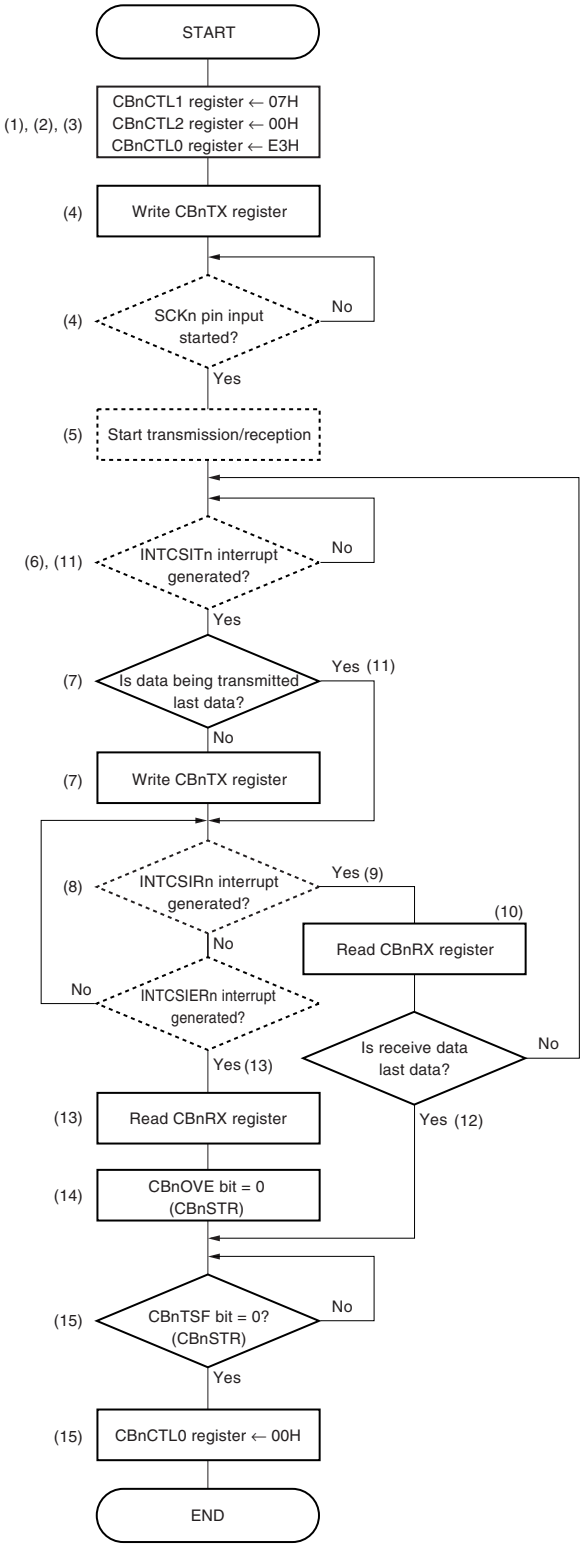
(13) To release the reception enable status, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnRXE bit = 0 after checking that the CBnTSF bit = 0.

Remark n = 0, 1

<R> **17.5.12 Continuous transfer mode (slave mode, transmission/reception mode)**

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = external clock (SCKn) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

(1) Operation flow



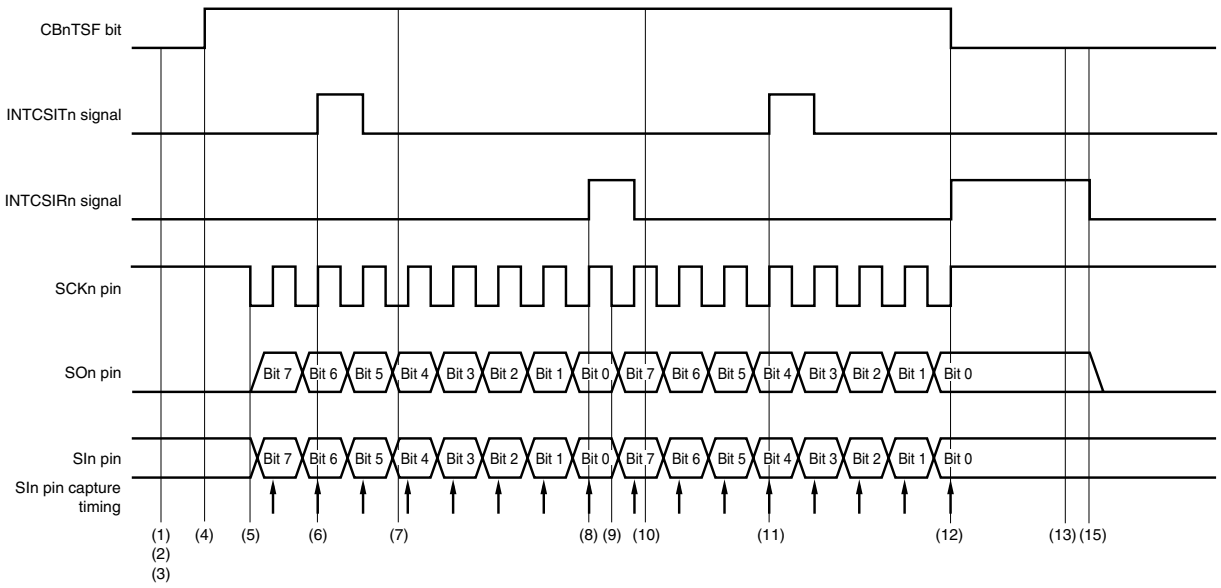
Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

3. n = 0, 1

(2) Operation timing

(1/2)



(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock (SCKn), and slave mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write E3H to the CBnCTL0 register, and select the transmission/reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data to the SOn pin in synchronization with the serial clock, and capture the receive data of the SIn pin.

(6) When transfer of the transmit data from the CBnTX register to the shift register is completed and writing to the CBnTX register is enabled, the transmission enable interrupt request signal (INTCSITn) is generated.

(7) To continue transmission, write the transmit data to the CBnTX register again after the INTCSITn signal is generated.

(8) When reception of the transfer data length set with the CBnCTL2 register is completed, the reception completion interrupt request signal (INTCSIRn) is generated, and reading of the CBnRX register is enabled.

(9) When a serial clock is input continuously, continuous transmission/reception is started.

(10) Read the CBnRX register.

(11) When transfer of the transmit data from the CBnTX register to the shift register is completed and writing to the CBnTX register is enabled, the INTCSITn signal is generated. To end continuous transmission/reception with the current transmission/reception, do not write to the CBnTX register.

Remark n = 0, 1

(2/2)

(12) When the clock of the transfer data length set with the CBnCTL2 register is input without writing to the CBnTX register, the INTCSIRn signal is generated. Clear the CBnTSF bit to 0 to end transmission/reception.
(13) When the reception error interrupt request signal (INTCSIERn) is generated, read the CBnRX register.
(14) If an overrun error occurs, write the CBnSTR.CBnOVE bit = 0, and clear the error flag.
(15) To release the transmission/reception enable status, write the CBnCTL0.CBnPWR bit = 0, the CBnCTL0.CBnTXE bit = 0, and the CBnCTL0.CBnRXE bit = 0 after checking that the CBnTSF bit = 0.

Remark n = 0, 1

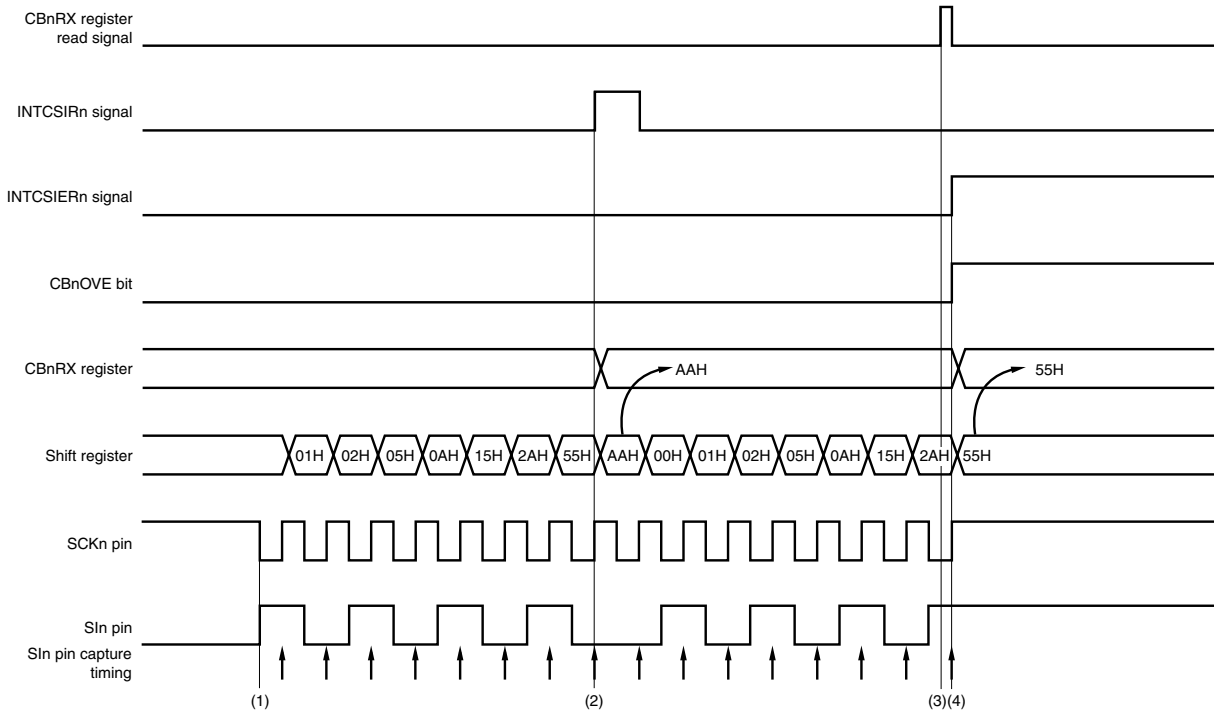
<R> **17.5.13 Reception error**

When transfer is performed with reception enabled (CBnCTL0.CBnRXE bit = 1) in the continuous transfer mode, the reception error interrupt request signal (INTCSIERn) is generated when the next receive operation is completed before the CBnRX register is read after the reception completion interrupt request signal (INTCSIRn) is generated, and the overrun error flag (CBnSTR.CBnOVE) is set to 1.

Even if an overrun error has occurred, the previous receive data is lost since the CBnRX register is updated. Even if a reception error has occurred, the INTCSIERn signal is generated again upon the next reception completion if the CBnRX register is not read.

To avoid an overrun error, complete reading the CBnRX register until one half clock before sampling the last bit of the next receive data from the INTCSIRn signal generation.

(1) Operation timing

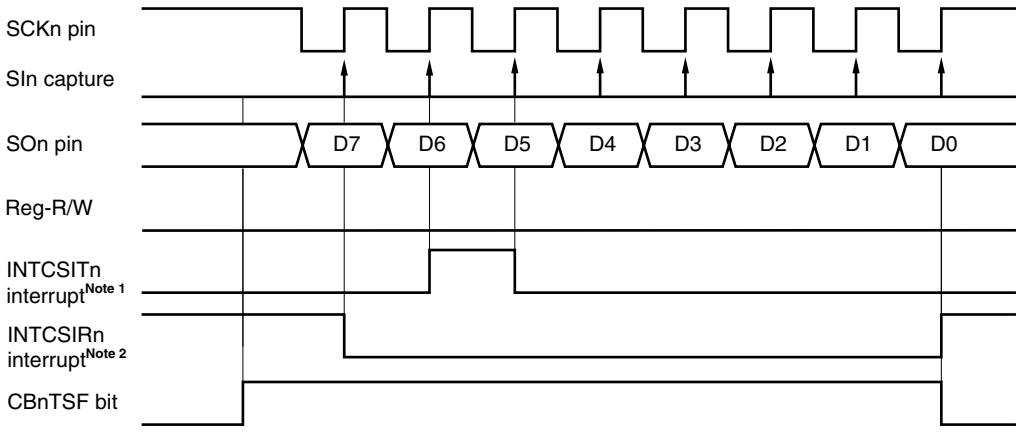


(1) Start continuous transfer.
(2) Completion of the first transfer
(3) The CBnRX register cannot be read until one half clock before the completion of the second transfer.
(4) An overrun error occurs, and the reception error interrupt request signal (INTCSIERn) is generated. The receive data is overwritten.

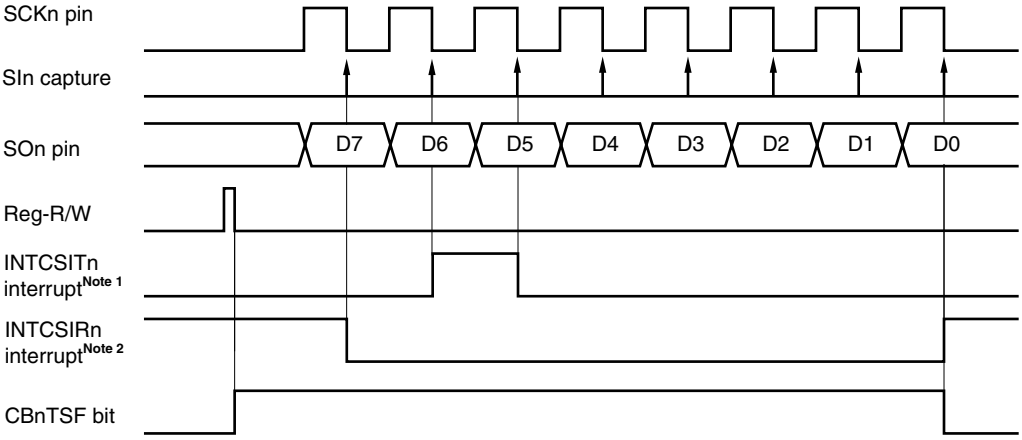
Remark n = 0, 1

17.5.14 Clock timing

(i) Communication type 1 (CBnCKP and CBnDAP bits = 00)



(ii) Communication type 3 (CBnCKP and CBnDAP bits = 10)



Notes 1. The INTCSITn interrupt is set when the data written to the CBnTX register is transferred to the data shift register in the continuous transmission or continuous transmission/reception mode. In the single transmission or single transmission/reception mode, the INTCSITn interrupt request signal is not generated, but the INTCSIRn interrupt request signal is generated upon end of communication.

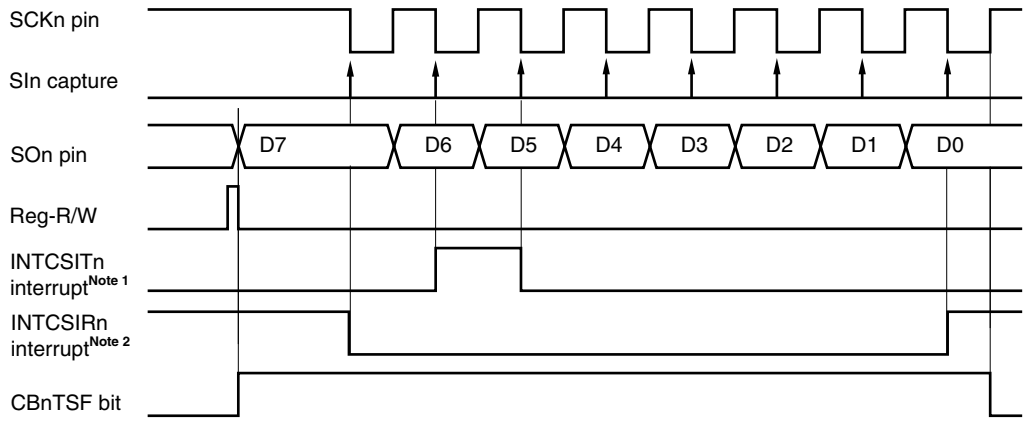
2. The INTCSIRn interrupt occurs if reception is correctly ended and receive data is ready in the CBnRX register while reception is enabled. In the single mode, the INTCSIRn interrupt request signal is generated even in the transmission mode, upon end of communication.

<R> **Caution In single transfer mode, writing to the CBnTX register with the CBnTSF bit set to 1 is ignored. This has no influence on the operation during transfer.**

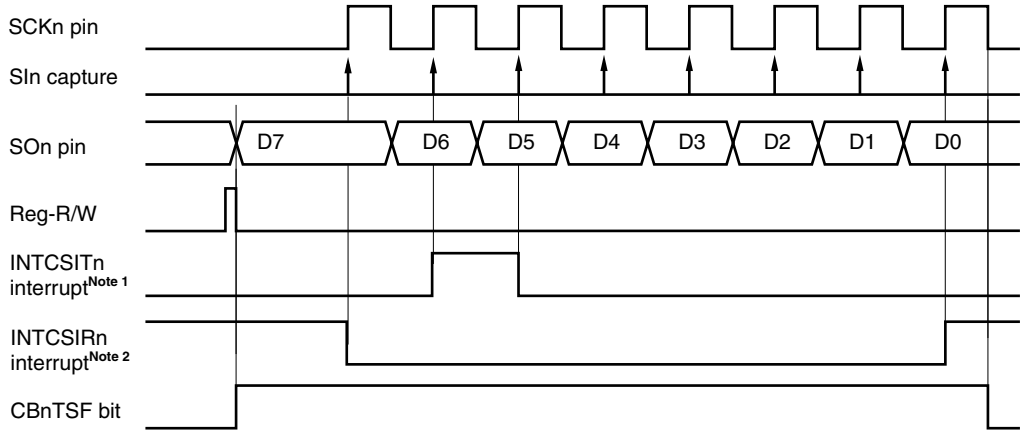
For example, if the next data is written to the CBnTX register when DMA is started by generating the INTCSIRn signal, the written data is not transferred because the CBnTSF bit is set to 1.

Use the continuous transfer mode, not the single transfer mode, for such applications.

(iii) Communication type 2 (CBnCKP and CBnDAP bits = 01)



(iv) Communication type 4 (CBnCKP and CBnDAP bits = 11)



Notes 1. The INTCSITn interrupt is set when the data written to the CBnTX register is transferred to the data shift register in the continuous transmission or continuous transmission/reception modes. In the single transmission or single transmission/reception modes, the INTCSITn interrupt request signal is not generated, but the INTCSIRn interrupt request signal is generated upon end of communication.

2. The INTCSIRn interrupt occurs if reception is correctly ended and receive data is ready in the CBnRX register while reception is enabled. In the single mode, the INTCSIRn interrupt request signal is generated even in the transmission mode, upon end of communication.

<R> **Caution In single transfer mode, writing to the CBnTX register with the CBnTSF bit set to 1 is ignored. This has no influence on the operation during transfer.**

For example, if the next data is written to the CBnTX register when DMA is started by generating the INTCSIRn signal, the written data is not transferred because the CBnTSF bit is set to 1.

Use the continuous transfer mode, not the single transfer mode, for such applications.

17.6 Output Pins

(1) SCKn pin

When CSIBn operation is disabled (CBnCTL0.CBnPWR bit = 0), the SCKn pin output status is as follows.

CBnCKP	CBnCKS2	CBnCKS1	CBnCKS0	SCKn 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 SCKn pin changes if any of the CBnCTL1.CBnCKP and CBnCKS2 to CBnCKS0 bits is rewritten.

(2) SOn pin

When CSIBn operation is disabled (CBnPWR bit = 0), the SOn pin output status is as follows.

CBnTXE	CBnDAP	CBnDIR	SOn Pin Output
0	×	×	Fixed to low level
1	0	×	SOn latch value (low level)
	1	0	CBnTX0 value (MSB)
		1	CBnTX0 value (LSB)

Remarks 1. The SOn pin output changes when any one of the CBnCTL0.CBnTXE, CBnCTL0.CBnDIR bits, and CBnCTL1.CBnDAP bit is rewritten.

2. ×: Don't care

CHAPTER 18 I²C BUS

To use the I²C bus function, set the P33/SDA and P34/SCL pins to the SDA and SCL pins. Then dummy open drain output (P-ch side is always off) is automatically set.

In the V850E/MA3, one channel of I²C bus is provided.
The products with an on-chip I²C bus are shown below.

<R> μPD703131AY, 703132AY, 703133AY, 703134AY, 703136AY, 70F3134AY

18.1 UARTA3/I²C Mode Switching

In the V850E/MA3, UARTA3 and I²C function alternately, and these pins cannot be used at the same time. To switch between UARTA3 and I²C, the PMC3, PFC4, and PFCE3 registers must be set in advance.

Caution The operations related to transmission and reception of UARTA3/I²C are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 18-1. UARTA3/I²C Mode Switch Settings

After reset: 00H R/W Address: FFFFF446H

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

After reset: 00H R/W Address: FFFFF466H

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

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3	0	0	0	PFCE34	PFCE33	PFCE32	PFCE31	PFCE30

PMC34	PFCE34	PFC34	Specification of alternate function of P34 pin
0	×	×	I/O port
1	0	1	RXD3 input
1	1	0	SCL[Note] I/O

PMC33	PFCE33	PFC33	Specification of alternate function of P33 pin
0	×	×	I/O port
1	0	1	TXD3 output
1	1	0	SDA[Note] I/O

Note I²C bus versions (Y products) only
When using the SDA and SCL pins, the pins function as dummy open-drain output pins (P-ch side is always off).

Remark x = don't care

18.2 Features

The I²C has the following two modes.

- Operation stop mode
- I²C (Inter IC) bus mode (multimaster supported)

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

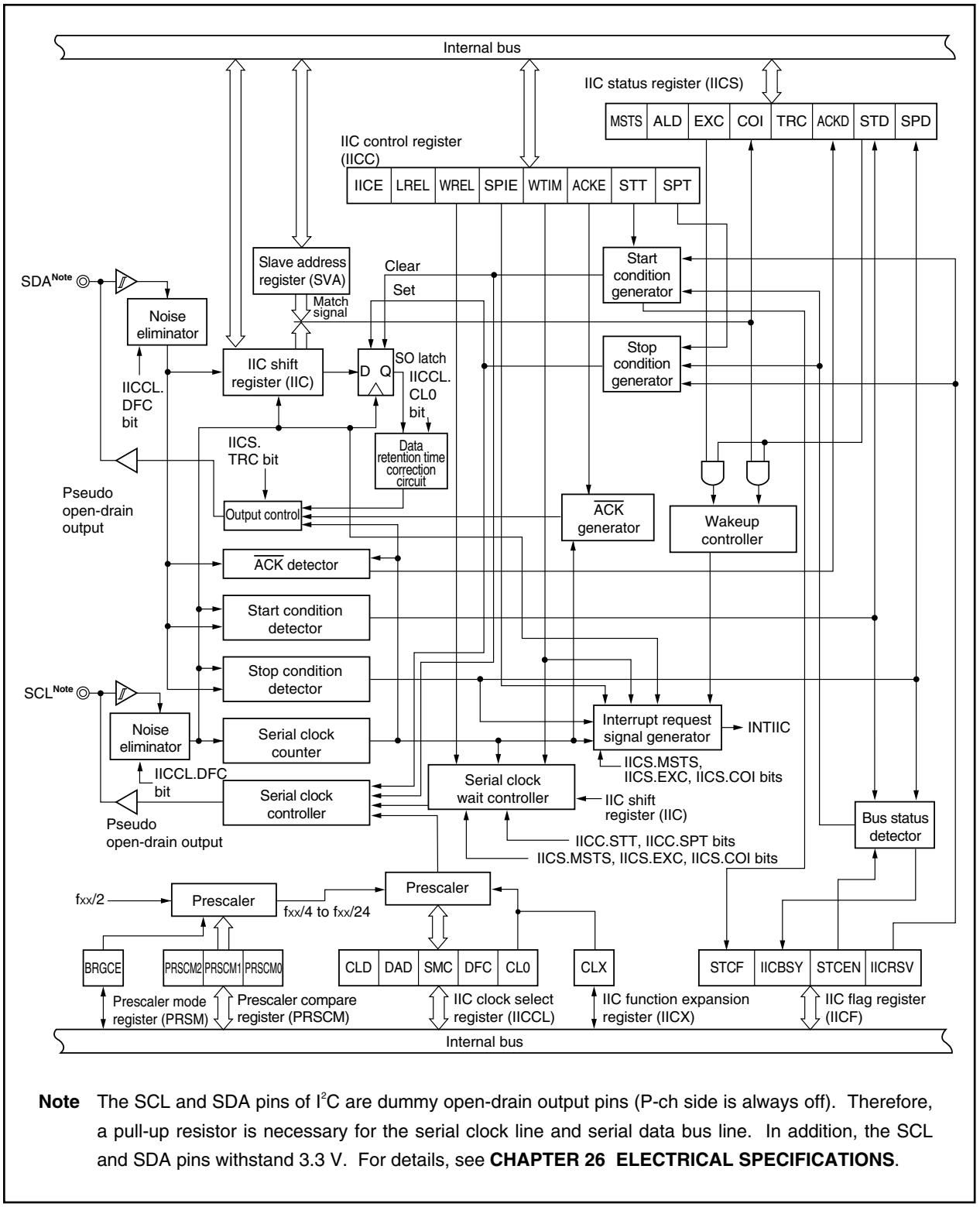
(2) I²C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCL) line and a serial data bus (SDA) line.

This mode complies with the I²C bus format and the master device can generate “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received statuses and data by hardware. This function can simplify the part of application program that controls the I²C bus.

Since the SCL and SDA pins are used for dummy open drain outputs (P-ch side is always off), I²C requires pull-up resistors for the serial clock line and the serial data bus line. The SCL and SDA pins withstand 3.3 V. For details, see **CHAPTER 26 ELECTRICAL SPECIFICATIONS**.

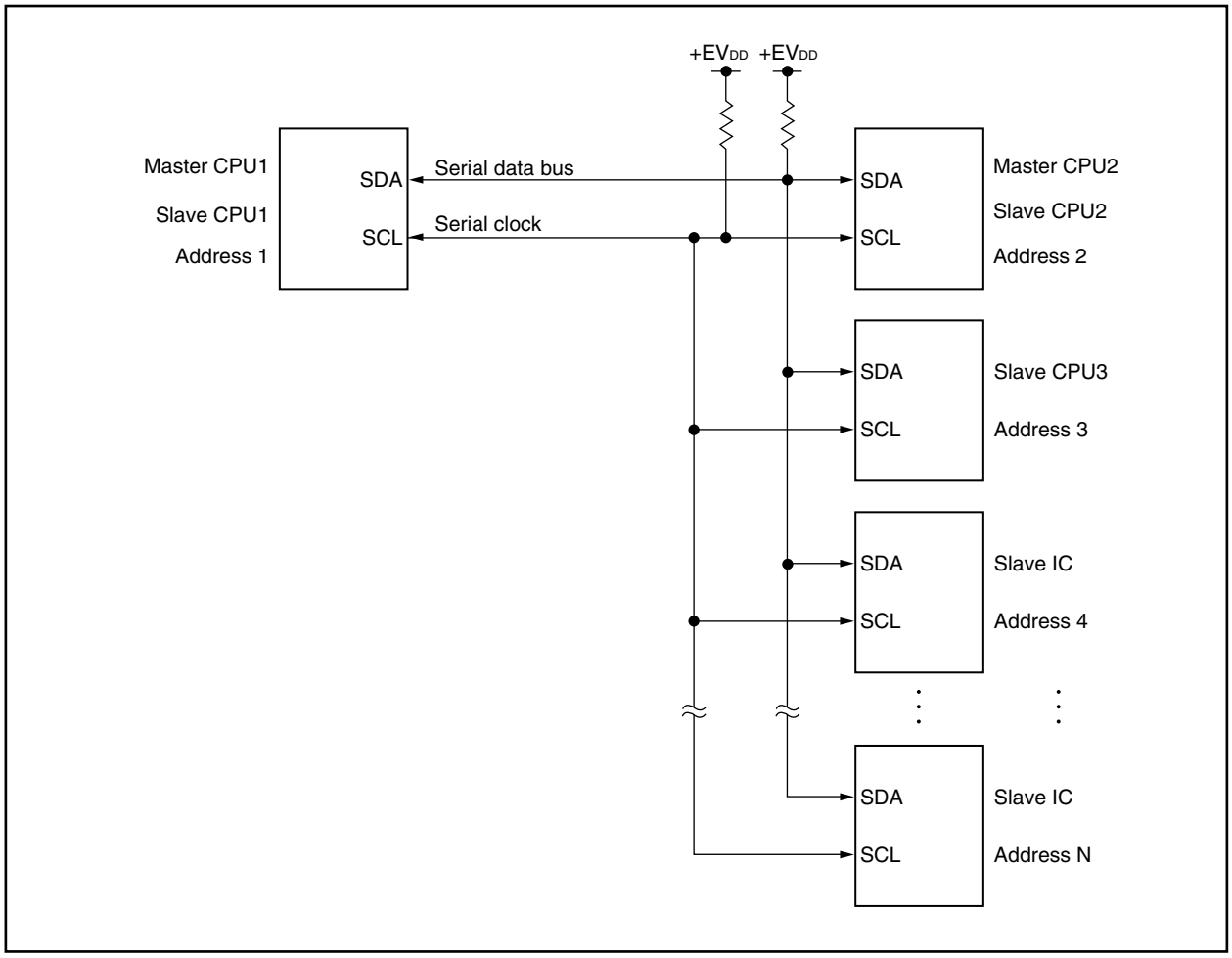
<R> **Figure 18-2. I²C Block Diagram**



Note The SCL and SDA pins of I²C are dummy open-drain output pins (P-ch side is always off). Therefore, a pull-up resistor is necessary for the serial clock line and serial data bus line. In addition, the SCL and SDA pins withstand 3.3 V. For details, see **CHAPTER 26 ELECTRICAL SPECIFICATIONS**.

A serial bus configuration example is shown below.

Figure 18-3. Serial Bus Configuration Example Using I²C Bus



18.3 Configuration

I²C includes the following hardware.

Table 18-1. Configuration of I²C

Item	Configuration
Registers	IIC shift register (IIC) Slave address register (SVA)
Control registers	IIC control register (IICC) IIC status register (IICS) IIC flag register (IICCF) IIC clock selection register (IICCL) IIC function expansion register (IICX) Prescaler mode register (PRSM) Prescaler compare register (PRSCM)

(1) IIC shift register (IIC)

The IIC register is used to convert 8-bit serial data to 8-bit parallel data and to convert 8-bit parallel data to 8-bit serial data. The IIC register can be used for both transmission and reception.

Write and read operations to the IIC register are used to control the actual transmit and receive operations.

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

After reset, IIC is cleared to 00H.

(2) Slave address register (SVA)

The SVA register sets local addresses when in slave mode.

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

After reset, SVA0 and SVA1 are cleared to 00H.

(3) SO latch

The SO latch is used to retain the SDA pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request signal (INTIIC) when the address received by this register matches the address value set to the SVA register or when an extension code is received.

(5) Prescaler

This selects the sampling clock to be used.

(6) Serial clock counter

This counter counts the serial clocks that are output and the serial clocks that are input during transmit/receive operations and is used to verify that 8-bit data was sent or received.

(7) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIIC).

An I²C interrupt is generated following either of two triggers.

- Falling of the eighth or ninth clock of the serial clock (set by IICC.WTIM bit)
- Interrupt request generated when a stop condition is detected (set by IICC.SPIE bit)

(8) Serial clock controller

In master mode, this circuit generates the clock output via the SCL pin from a sampling clock.

(9) Serial clock wait controller

This circuit controls the wait timing.

(10) ACK generator, stop condition detector, start condition detector, and ACK detector

These circuits are used to generate and detect various statuses.

(11) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

(12) Start condition generator

This circuit generates a start condition when the IICC.STT bit is set.

However, in the communication reservation disabled status (IICF.IICRSV bit = 1), when the bus is not released (IICF.IICBSY bit = 1), start condition requests are ignored and the IICF.STCF bit is set to 1.

<R> **(13) Stop condition generator**

A stop condition is generated when the IICC.SPT bit is set.

(14) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the IICF.STCEN bit.

18.4 Registers

I²C is controlled by the following registers.

- IIC control register (IICC)
- IIC status register (IICS)
- IIC flag register (IICF)
- IIC clock selection register (IICCL)
- IIC function expansion register (IICX)
- Prescaler mode register (PRSM)
- Prescaler compare register (PRSCM)

The following registers are also used.

- IIC shift register (IIC)
- Slave address register (SVA)

Remark For the alternate-function pin settings, see **Table 4-19 Using Alternate Function of Port Pins**.

(1) IIC control register (IICC)

The IICC register is used to enable/stop I²C operations, set wait timing, and set other I²C operations.

<R> The IICC register can be read or written in 8-bit or 1-bit units. However, set the SPIE, WTIM, and ACKE bits when the IICE bit is 0 or during the wait period. When setting the IICE bit from “0” to “1”, these bits can also be set at the same time.

Reset input clears this register.

(1/4)

After reset: 00H R/W Address: FFFFFD82H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICC	IICE	LREL	WREL	SPIE	WTIM	ACKE	STT	SPT

IICE	I²C operation enable/disable specification
0	Stop operation. Reset the IICS register[Note 1]. Stop internal operation.
1	Enable operation.
Be sure to set this register when the SCL and SDA lines are high level.	
Condition for clearing (IICE bit = 0)	Condition for setting (IICE bit = 1)
• Cleared by instruction • Reset	• Set by instruction

LREL[Note 2]	Exit from communications
0	Normal operation
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCL and SDA lines are set to high impedance. The STT, SPT, IICS.MSTS, IICS.EXC, IICS.COI, IICS.TRC, IICS.ACKD, and IICS.STD bits are cleared to 0.
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • After a stop condition is detected, restart is in master mode. • An address match or extension code reception occurs after the start condition.	
Condition for clearing (LREL bit = 0)	Condition for setting (LREL bit = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WREL[Note]	Wait cancellation control
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared to 0 after wait is canceled.
Condition for clearing (WREL bit = 0)	Condition for setting (WREL bit = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Notes 1. The IICS register, and the IICF.STCF, IICF.IICBSY, IICCL.CLD, and IICCL.DAD bits are reset.

2. This flag's signal is invalid when the IICE bit = 0.

<R> **Caution If the I²C operation is enabled (IICE bit = 1) when the SCL line is high level and the SDA line is low level, the start condition is detected immediately. After enabling I²C operation (IICE bit = 1), set the LREL bit to 1 with a bit manipulation instruction after the wait time shown in Table 18-2.**

(2/4)

SPIE[Note]	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable

Condition for clearing (SPIE bit = 0)	Condition for setting (SPIE bit = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIM[Note]	Control of wait and interrupt request generation
0	Interrupt request is generated at the eighth clock's falling edge. Master mode: After output of eight clocks, clock output is set to low level and wait is set. Slave mode: After input of eight clocks, the clock is set to low level and wait is set for master device.
1	Interrupt request is generated at the ninth clock's falling edge. Master mode: After output of nine clocks, clock output is set to low level and wait is set. Slave mode: After input of nine clocks, the clock is set to low level and wait is set for master device.
An interrupt is generated at the falling of the 9th clock during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. When in master mode, a wait is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a wait is inserted at the falling edge of the ninth clock after $\overline{ACK}$ is issued. However, when the slave device has received an extension code, a wait is inserted at the falling edge of the eighth clock.	

Condition for clearing (WTIM bit = 0)	Condition for setting (WTIM bit = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKE[Note]	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDA line is set to low level.
<R> The ACKE bit setting is invalid for address reception by the slave device. In this case, $\overline{ACK}$ is generated when the addresses match. However, the ACKE bit setting is valid for reception of the extension code. Set the ACKE bit in the system that receives the extension code.	

Condition for clearing (ACKE bit = 0)	Condition for setting (ACKE bit = 1)
• Cleared by instruction • Reset	• Set by instruction

Note This flag's signal is invalid when the IICE bit = 0.

(3/4)

STT	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in STOP mode): Generate a start condition (for starting as master). The SDA line is changed from high level to low level while the SCL line is high level and then the start condition is generated. Next, after the rated amount of time has elapsed, the SCL line is changed to low level (in wait state). When a third party is communicating • When communication reservation function is enabled (IICF.IICRSV bit = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. • When communication reservation function is disabled (IICRSV bit = 1) The IICF.STCF bit is set to 1 to clear the STT bit which is set to 1. In the wait state (when master device): Generates a restart condition after releasing the wait.

Cautions concerning set timing

For master reception: Cannot be set to 1 during transfer. Can be set to 1 only when the ACKE bit has been cleared to 0 and slave has been notified of final reception.

For master transmission: A start condition cannot be generated normally during the acknowledgment. Set to 1 during the wait period that follows output of the ninth clock.

- Cannot be set to 1 at the same time as the SPT bit.
- <R> When the STT bit is set to 1, setting the STT bit to 1 again is disabled until the setting is cleared to 0.

<R> Condition for clearing (STT bit = 0)	<R> Condition for setting (STT bit = 1)
• When the STT bit is set to 1 in the communication reservation disabled status • Cleared by loss in arbitration • Cleared because start condition is generated by master device • When the LREL bit = 1 (exit from communications) • When the IICE bit = 0 (operation stop) • Reset	• Set by instruction

Remark The STT bit is 0 if it is read after data setting.

(4/4)

SPT	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master device's transfer). After the SDA line goes to low level, either set the SCL line to high level or wait until the SCL pin goes to high level. Next, after the rated amount of time has elapsed, the SDA line is changed from low level to high level and a stop condition is generated.

Cautions concerning setting timing

For master reception: Cannot be set to 1 during transfer. Can be set to 1 only when the ACKE bit has been cleared to 0 and during the wait period after slave has been notified of final reception.

For master transmission: A stop condition cannot be generated normally during the acknowledgment period. Set to 1 during the wait period that follows output of the ninth clock.

- Cannot be set to 1 at the same time as the STT bit.
- The SPT bit can be set to 1 only when in master mode[Note].
- When the WTIM bit has been cleared to 0, if the SPT bit is set to 1 during the wait period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock. The WTIM bit should be set from 0 to 1 during the wait period following output of eight clocks, and the SPT bit should be set to 1 during the wait period that follows output of the ninth clock.
- When the SPT bit is set to 1, setting the SPT bit to 1 again is disabled until the setting is cleared to 0.

<R>

Condition for clearing (SPT bit = 0)	Condition for setting (SPT bit = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • When the LREL bit = 1 (exit from communications) • When the IICE bit = 0 (operation stop) • Reset	• Set by instruction

Note Set the SPT bit to 1 only in master mode. However, the SPT bit must be set to 1 and a stop condition generated before the first stop condition is detected following the switch to operation enable status. For details, see **18.15 Cautions**.

Caution When the IICS.TRC bit is set to 1, the WREL bit is set to 1 during the ninth clock and wait is canceled, after which the TRC bit is cleared to 0 and the SDA line is set to high impedance.

Remark The SPT bit is 0 if it is read after data setting.

Make sure by using software that the wait time shown in Table 18-2 elapses. The wait time can be set by using the IICCL.SMC and IICCL.CL0 bits, IICX.CLX bit, and PRSCM.PRSCM2 to PRSCM.PRSCM0 bits in combination.

<R>

Table 18-2. Wait Periods

IICX	IICCL		PRSCM Register	Division Clock	Wait Period
Bit 0	Bit 3	Bit 0			
CLX	SMC	CL0			
0	0	0	01H	f_{XX}/4	12 clocks
0	0	0	02H	f_{XX}/8	24 clocks
0	0	0	03H	f_{XX}/12	36 clocks
0	0	0	04H	f_{XX}/16	48 clocks
0	0	0	05H	f_{XX}/20	60 clocks
0	0	0	06H	f_{XX}/24	72 clocks
0	0	1	01H	f_{XX}/4	12 clocks
0	0	1	02H	f_{XX}/8	24 clocks
0	0	1	03H	f_{XX}/12	36 clocks
0	0	1	04H	f_{XX}/16	48 clocks
0	1	x	01H	f_{XX}/4	12 clocks
0	1	x	02H	f_{XX}/8	24 clocks
0	1	x	03H	f_{XX}/12	36 clocks
0	1	x	04H	f_{XX}/16	48 clocks
0	1	x	05H	f_{XX}/20	60 clocks
1	1	x	01H	f_{XX}/4	12 clocks
1	1	x	02H	f_{XX}/8	24 clocks
1	1	x	03H	f_{XX}/12	36 clocks
1	1	x	04H	f_{XX}/16	48 clocks
1	1	x	05H	f_{XX}/20	60 clocks
Other than above				Setting prohibited	

Remark Don't care

(2) IIC status register (IICS)

The IICS register indicates the status of the I²C bus.

<R> The IICS register is read-only, in 8-bit or 1-bit units. However, the IICS register can be read only when the IICC.STT bit is 1 or during the wait period.

After reset, IICS is cleared to 00H.

(1/3)

After reset: 00H R Address: FFFFFD86H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS	MSTS	ALD	EXC	COI	TRC	ACKD	STD	SPD

MSTS	Master device status
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS bit = 0)	Condition for setting (MSTS bit = 1)
• When a stop condition is detected • When the ALD bit = 1 (arbitration loss) • Cleared by the IICC.LREL bit = 1 (exit from communications) • When the IICC.IICE bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. The MSTS bit is cleared to 0.

Condition for clearing (ALD bit = 0)	Condition for setting (ALD bit = 1)
• Automatically cleared after the IICS register is read[Note] • When the IICE bit changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

EXC	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC bit = 0)	Condition for setting (EXC bit = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by the LREL bit = 1 (exit from communications) • When the IICE bit changes from 1 to 0 (operation stop) • Reset	• When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

Note This bit is also cleared when a bit manipulation instruction is executed for another bit in the IICS register.

(2/3)

COI	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI bit = 0)	Condition for setting (COI bit = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by the LREL bit = 1 (exit from communications) • When the IICE bit changes from 1 to 0 • Reset	• When the received address matches the local address (SVA register) (set at the rising edge of the eighth clock).

TRC	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDA line is set for high impedance.
1	Transmit status. The value in the SO latch is enabled for output to the SDA line (valid starting at the rising edge of the first byte's ninth clock).

Condition for clearing (TRC bit = 0)	Condition for setting (TRC bit = 1)
• When a stop condition is detected • Cleared by the LREL bit = 1 (exit from communications) • When the IICE bit changes from 1 to 0 (operation stop) • Cleared by the IICC.WREL bit = 1[Note] (wait release) • When the ALD bit changes from 0 to 1 (arbitration loss) • Reset Master • When "1" is output to the first byte's LSB (transfer direction specification bit) Slave • When a start condition is detected When not used for communication	Master • When a start condition is generated • When "0" is output to the first byte's LSB (transfer direction specification bit) Slave • When "1" is input in the first byte's LSB (transfer direction specification bit)

<R>

ACKD	Detection of $\overline{\text{ACK}}$
0	Acknowledgment was not detected.
1	Acknowledgment was detected.

Condition for clearing (ACKD bit = 0)	Condition for setting (ACKD bit = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by the LREL bit = 1 (exit from communications) • When the IICE bit changes from 1 to 0 (operation stop) • Reset	• After the SDA pin is set to low level at the rising edge of the SCL pin's ninth clock

Note The IICS.TRC bit is cleared to 0 and the SDA line become high impedance when the IICC.WREL bit is set to 1 and wait state is released at the ninth clock with the TRC bit = 1.

(3/3)

STD	Detection of start condition
0	Start condition was not detected.
1	Start condition was detected. This indicates that the address transfer period is in effect

Condition for clearing (STD bit = 0)	Condition for setting (STD bit = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by the LREL bit = 1 (exit from communications) • When the IICE bit changes from 1 to 0 (operation stop) • Reset	When a start condition is detected

SPD	Detection of stop condition
0	Stop condition was not detected.
1	Stop condition was detected. The master device's communication is terminated and the bus is released.

Condition for clearing (SPD bit = 0)	Condition for setting (SPD bit = 1)
• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition • When the IICE bit changes from 1 to 0 (operation stop) • Reset	When a stop condition is detected

(3) IIC flag register (IICF)

IICF is register that sets the operation mode of I²C and indicate the status of the I²C bus.

These registers can be read or written in 8-bit or 1-bit units. However, the STCF and IICBSY bits are read-only.

The IICRSV bit can be used to enable/disable the communication reservation function (see **18.14 Communication Reservation**).

The STCEN bit can be used to set the initial value of the IICBSY bit (see **18.15 Cautions**).

The IICRSV and STCEN bits can be written only when the operation of I²C is disabled (IICC.IICE bit = 0). When operation is enabled, the IICF register can be read.

After reset, IICF is cleared to 00H.

After reset: 00H R/W[Note] Address: FFFFFD8AH

	<7>	<6>	5	4	3	2	<1>	<0>
IICF	STCF	IICBSY	0	0	0	0	STCEN	IICRSV

STCF	IICC.STT clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear STT flag

Condition for clearing (STCF bit = 0)	Condition for setting (STCF bit = 1)
• Clearing by setting the IICC.STT bit = 1 • When the IICE bit = 0 • Reset	• Generating start condition unsuccessful and the STT bit cleared to 0 when communication reservation is disabled (IICRSV bit = 1).

<R>

IICBSY	I²C bus status flag
0	Bus release status (default communication status when STCEN bit = 1)
1	Bus communication status (default communication status when STCEN bit = 0)

Condition for clearing (IICBSY bit = 0)	Condition for setting (IICBSY bit = 1)
• Detection of stop condition • When the IICE bit = 0 • Reset	• Detection of start condition • Setting of the IICC.IICE bit when the STCEN bit = 0

<R>

STCEN	Initial start enable trigger
0	After operation is enabled (IICE bit = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICE bit = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCEN bit = 0)	Condition for setting (STCEN bit = 1)
• Detection of start condition • Reset	• Setting by instruction

IICRSV	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSV bit = 0)	Condition for setting (IICRSV bit = 1)
• Clearing by instruction • Reset	• Setting by instruction

Note Bits 6 and 7 are read-only bits.

Cautions 1. Write to the STCEN bit only when the operation is stopped (IICE bit = 0).

2. As the bus release status (IICBSY bit = 0) is recognized regardless of the actual bus status when the STCEN bit = 1, when generating the first start condition (STT bit = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

3. Write to the IICRSV bit only when the operation is stopped (IICE bit = 0).

(4) IIC clock selection register (IICCL)

The IICCL register is used to set the transfer clock for the I²C bus.

The IICCL register can be read or written in 8-bit or 1-bit units. However, the CLD and DAD bits are read-only.

The SMC and CL0 bits are set using the IICX.CLX bit in combination with the PRSCM.PRSCM2 to PRSCM.PRSCM0 bits (see **18.4 (8) I²C transfer clock setting method**).

<R> Set the IICCL register when the IICC.IICE bit = 0.

Reset input clears this register to 00H.

After reset: 00H R/W[Note] Address: FFFFFD84H

	7	6	<5>	<4>	3	2	1	0
IICCL	0	0	CLD	DAD	SMC	DFC	0	CL0

CLD	Detection of SCL pin level (valid only when IICC.IICE bit = 1)
0	The SCL pin was detected at low level.
1	The SCL pin was detected at high level.

Condition for clearing (CLD bit = 0)	Condition for setting (CLD bit = 1)
• When the SCL pin is at low level • When the IICE bit = 0 (operation stop) • Reset	• When the SCL pin is at high level

DAD	Detection of SDA pin level (valid only when IICE bit = 1)
0	The SDA pin was detected at low level.
1	The SDA pin was detected at high level.

Condition for clearing (DAD bit = 0)	Condition for setting (DAD bit = 1)
• When the SDA pin is at low level • When IICE bit = 0 (operation stop) • Reset	• When the SDA pin is at high level

SMC	Operation mode switching
0	Operates in standard mode.
1	Operates in high-speed mode.

DFC	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Digital filter can be used only in high-speed mode. In high-speed mode, the transfer clock does not vary regardless of DFC bit set/clear. The digital filter is used for noise elimination in high-speed mode.	

Note Bits 4 and 5 are read-only bits.

<R> **Caution Be sure to clear bits 1, 6, and 7 to "0".**

(5) IIC function expansion register (IICX)

These registers set the function expansion of I²C (valid only in high-speed mode).

These registers can be read or written in 8-bit or 1-bit units. The CLX bit is set by using PRSCM.PRSCM2 to PRSCM.PRSCM0 bits in combination with the IICCL.SMC and IICCL.CL0 bits (see **18.4 (8) I²C transfer clock setting method**).

<R> Set the IICX register when the IICC.IICE bit = 0.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFFD85H

	7	6	5	4	3	2	1	<0>
IICX	0	0	0	0	0	0	0	CLX

(6) Prescaler mode register (PRSM)

The PRSM register controls the generation of the baud rate signal for I²C.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFFD90H

	7	6	5	<4>	3	2	1	0
PRSM	0	0	0	BRGCE	0	0	0	0

BRGCE	Baud rate output
0	Fixed to 0
1	Operates

Caution Be sure to clear bits 0 to 3 and 5 to 7 to "0".

(7) Prescaler compare register (PRSCM)

The PRSCM register controls the I²C division clock.

This register can be read or written in 8-bit units. The PRSCM2 to PRSCM0 bits are set by using the IICCL.SMC and IICCL.CL0 bits in combination with the IICX.CLX bit (see **18.4 (8) I²C transfer clock setting method**).

<R> Set the PRSCM register when the IICC.IICE bit = 0.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFFD91H

	7	6	5	4	3	2	1	0
PRSCM	0	0	0	0	0	PRSCM2	PRSCM1	PRSCM0

Cautions 1. Do not rewrite the PRSCM register during transmission.

2. Set the PRSCM register before setting the PRSM.BRGCE bit to 1.

<R> **3. Be sure to clear bits 3 to 7 to "0".**

(8) I²C transfer clock setting method

The I²C transfer clock frequency (f_{SCL}) is calculated using the following expression.

$$f_{SCL} = 1/(m \times T + t_R + t_F)$$

m = 48, 96, 144, 176, 192, 240, 288, 344, 352, 384, 480, 528, 688, 704, 880, 1,032, 1,056, 1,376 (see **Table 18-3 Selection Clock Setting**)

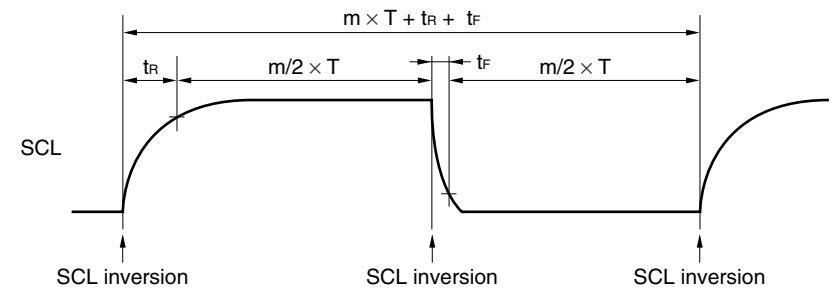
T: $1/f_{XX}$

t_R: SCL rise time

t_F: SCL fall time

For example, the I²C transfer clock frequency (f_{SCL}) when f_{XX} = 80 MHz, m = 880, t_R = 200 ns, and t_F = 50 ns is calculated using following expression.

$$f_{SCL} = 1/(880 \times 12.5 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 88.9 \text{ kHz}$$



The selection clock is set using a combination of the IICCL.SMC and IICCL.CL0 bits, the IICX.CLX bit, and the PRSCM.PRSCM2 to PRSCM.PRSCM0 bits.

Table 18-3. Selection Clock Setting

IICX	IICCL		PRSCM Register	Division Clock	Transfer Clock (f_{XX}/m)	Settable Peripheral Clock Frequency (f_{XX}) Range	Operation Mode
Bit 0	Bit 3	Bit 0					
CLX	SMC	CL0					
0	0	0	01H	f_{XX}/4	f_{XX}/176	10.00 to 16.76 MHz	Normal mode (SMC bit = 0)
0	0	0	02H	f_{XX}/8	f_{XX}/352	20.00 to 33.52 MHz	
0	0	0	03H	f_{XX}/12	f_{XX}/528	24.00 to 50.28 MHz	
0	0	0	04H	f_{XX}/16	f_{XX}/704	32.00 to 67.04 MHz	
0	0	0	05H	f_{XX}/20	f_{XX}/880	40.00 to 80.00 MHz	
0	0	0	06H	f_{XX}/24	f_{XX}/1,056	48.00 to 80.00 MHz	
0	0	1	01H	f_{XX}/4	f_{XX}/344	16.76 to 33.52 MHz	
0	0	1	02H	f_{XX}/8	f_{XX}/688	33.52 to 67.04 MHz	
0	0	1	03H	f_{XX}/12	f_{XX}/1,032	50.28 to 80.00 MHz	
0	0	1	04H	f_{XX}/16	f_{XX}/1,376	67.04 to 80.00 MHz	
0	1	x	01H	f_{XX}/4	f_{XX}/96	16.00 to 33.52 MHz	High-speed mode (SMC bit = 1)
0	1	x	02H	f_{XX}/8	f_{XX}/192	32.00 to 67.04 MHz	
0	1	x	03H	f_{XX}/12	f_{XX}/288	48.00 to 80.00 MHz	
0	1	x	04H	f_{XX}/16	f_{XX}/384	64.00 to 80.00 MHz	
0	1	x	05H	f_{XX}/20	f_{XX}/480	80.00 to 80.00 MHz	
1	1	x	01H	f_{XX}/4	f_{XX}/48	16.00 to 16.76 MHz	
1	1	x	02H	f_{XX}/8	f_{XX}/96	32.00 to 33.52 MHz	
1	1	x	03H	f_{XX}/12	f_{XX}/144	48.00 to 50.28 MHz	
1	1	x	04H	f_{XX}/16	f_{XX}/192	64.00 to 67.04 MHz	
1	1	x	05H	f_{XX}/20	f_{XX}/240	80.00 to 80.00 MHz	
Other than above					Setting prohibited		

Remark x: Don’t care

(9) IIC shift register (IIC)

The IIC register is used for serial transmission/reception (shift operations) that is synchronized with the serial clock.

The IIC register can be read or written in 8-bit units, but data should not be written to the IIC register during a data transfer.

<R> Access (read/write) the IIC register only during the wait period. Accessing this register in communication states other than the wait period is prohibited. However, for the master device, the IIC register can be written once only after the transmission trigger bit (IICC.STT bit) has been set to 1.

When the IIC register is written during wait, the wait is cancelled and data transfer is started.

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFFD80H

	7	6	5	4	3	2	1	0
IIC								

(10) Slave address register (SVA)

<R> The SVA register holds the I²C bus's slave addresses. However, rewriting this register is prohibited when the IICS.STD bit = 1 (start condition detection).

The SVA register can be read or written in 8-bit units, but bit 0 should be fixed as 0.

After reset, SVA is cleared to 00H.

After reset: 00H R/W Address: FFFFFD83H

	7	6	5	4	3	2	1	0
SVA								0

18.5 Functions

18.5.1 Pin configuration

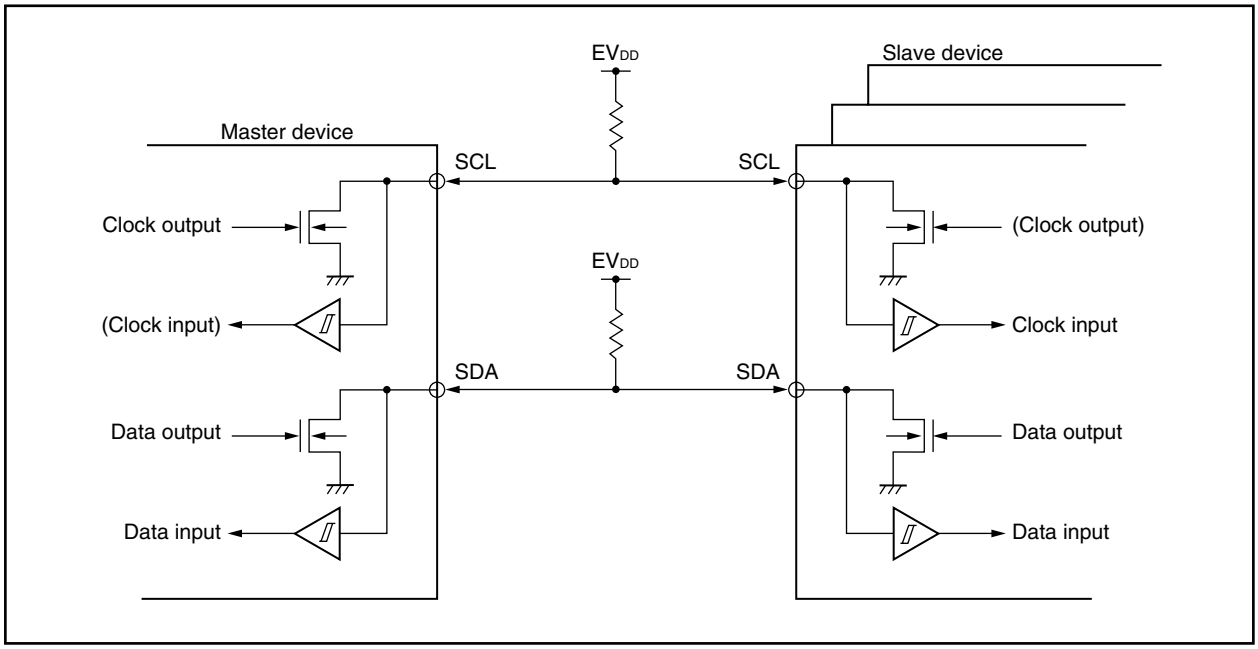
The serial clock pin (SCL) and serial data bus pin (SDA) are configured as follows.

SCL .....This pin is used for serial clock input and output.
This pin is a dummy open-drain output (P-ch side is always off) for both master and slave devices. Input is Schmitt input.

SDA .....This pin is used for serial data input and output.
This pin is a dummy open-drain output (P-ch side is always off) for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are dummy open-drain outputs, an external pull-up resistor is required.

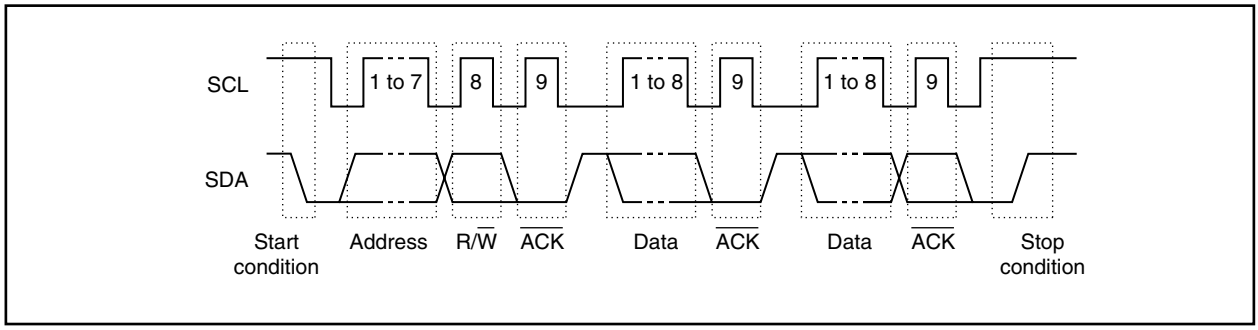
Figure 18-4. Pin Configuration Diagram



18.6 I²C Bus Definitions and Control Methods

The following section describes the I²C bus's serial data communication format and the statuses generated by the I²C bus. The transfer timing for the "start condition", "address", "transfer direction specification", "data", and "stop condition" generated via the I²C bus's serial data bus is shown below.

Figure 18-5. I²C Bus's Serial Data Transfer Timing



The master device generates the start condition, slave address, and stop condition.

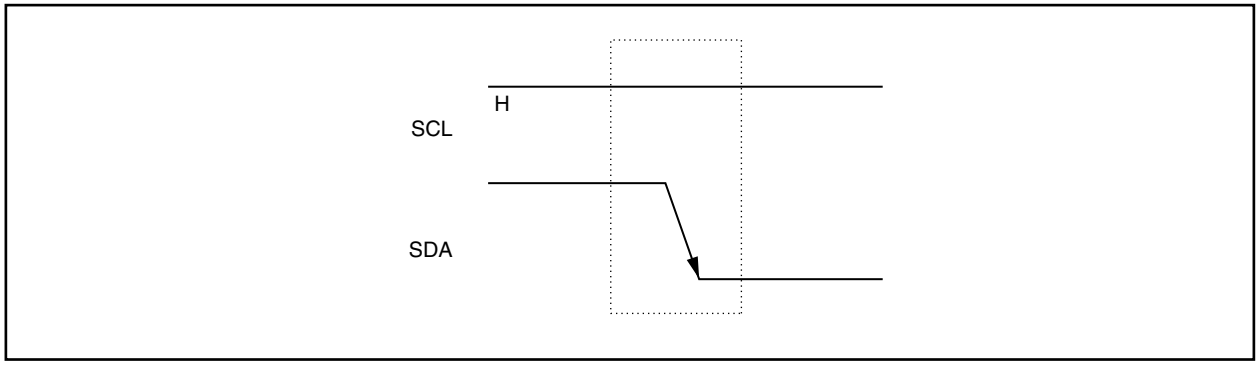
The $\overline{ACK}$ can be generated by either the master or slave device (normally, it is generated by the device that receives 8-bit data).

The serial clock (SCL) is continuously output by the master device. However, in the slave device, the SCL's low-level period can be extended and a wait can be inserted.

18.6.1 Start condition

A start condition is met when the SCL pin is at high level and the SDA pin changes from high level to low level. The start conditions for the SCL pin and SDA pin are signals that the master device generates to the slave device when starting a serial transfer. Start conditions can be detected when the device is used as a slave.

Figure 18-6. Start Conditions



A start condition is generated when the IICC.STT bit is set to 1 after a stop condition has been detected (IICS.SPD bit = 1). When a start condition is detected, IICS.STD bit is set to 1.

Caution When the IICC.IICE bit of the V850E/MA3 is set to 1 during communication between other devices, a start condition may be detected depending on the communication line state. Be sure to set the IICE bit to 1 when the SCL and SDA lines are high level.

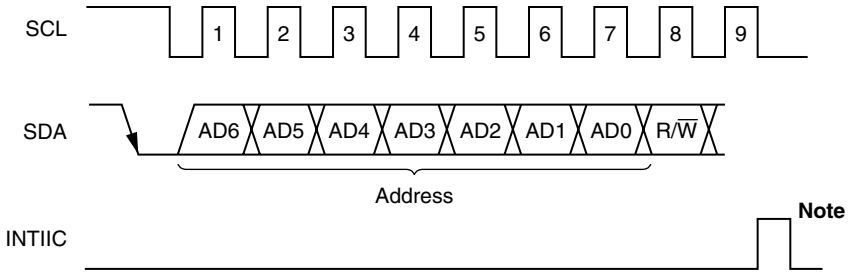
18.6.2 Addresses

The 7 bits of data that follow the start condition are defined as an address.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data value stored in the SVA register. If the address data matches the SVA register value, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 18-7. Address



Note The interrupt request signal (INTIIC) is generated if a local address or extension code is received during slave device operation.

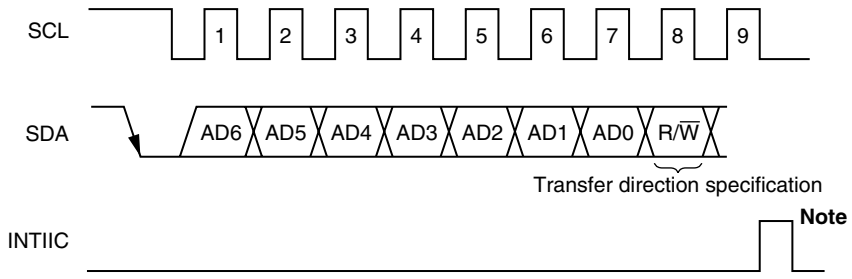
The slave address and the eighth bit, which specifies the transfer direction as described in **18.6.3 Transfer direction specification** below, are together written to the IIC register and are then output. Received addresses are written to the IIC register.

The slave address is assigned to the higher 7 bits of the IIC register.

18.6.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction. When this transfer direction specification bit has a value of 0, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of 1, it indicates that the master device is receiving data from a slave device.

Figure 18-8. Transfer Direction Specification



Note The interrupt request signal (INTIIC) is generated if a local address or extension code is received during slave device operation.

18.6.4 $\overline{\text{ACK}}$

$\overline{\text{ACK}}$ is used to confirm the serial data status of the transmitting and receiving devices.

The receiving device returns $\overline{\text{ACK}}$ for every 8 bits of data it receives.

The transmitting device normally receives $\overline{\text{ACK}}$ after transmitting 8 bits of data. When $\overline{\text{ACK}}$ is returned from the receiving device, the reception is judged as normal and processing continues. The detection of $\overline{\text{ACK}}$ is confirmed with the IICS.ACKD bit.

When the master device is the receiving device, after receiving the final data, it does not return $\overline{\text{ACK}}$ and generates the stop condition. When the slave device is the receiving device and does not return $\overline{\text{ACK}}$, the master device generates either a stop condition or a restart condition, and then stops the current transmission. Failure to return $\overline{\text{ACK}}$ may be caused by the following factors.

<1> Reception was not performed normally.

<2> The final data was received.

<3> The receiving device does not exist for the specified address.

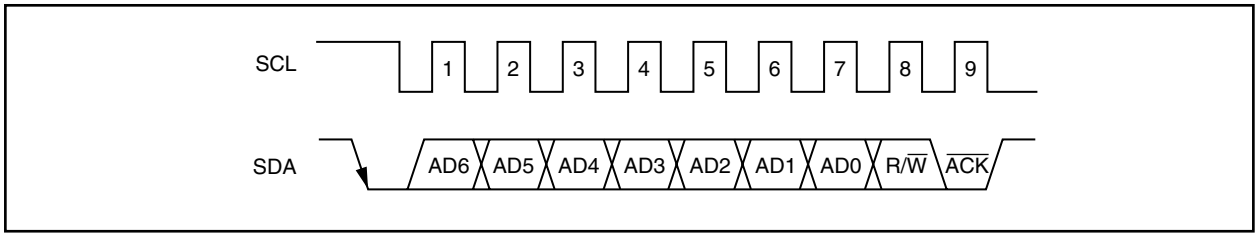
When the receiving device sets the SDA line to low level during the ninth clock, $\overline{\text{ACK}}$ is generated (normal reception).

When the IICC.ACKE bit is set to 1, automatic $\overline{\text{ACK}}$ generation is enabled. Transmission of the eighth bit following the 7 address data bits causes the IICS.TRC bit to be set. Normally, set the ACKE bit to 1 for reception (TRC bit = 0).

When the slave device is receiving (when TRC bit = 0), if the slave device cannot receive data, clear the ACKE bit to 0 to indicate to the master that no more data can be received.

Similarly, when the master device is receiving (when TRC bit = 0) and the subsequent data is not needed, clear the ACKE bit to 0 to prevent $\overline{\text{ACK}}$ from being generated. This notifies the slave device (transmitting device) of the end of the data transmission (transmission stopped).

Figure 18-9. $\overline{\text{ACK}}$



When the local address is received, $\overline{\text{ACK}}$ is automatically generated regardless of the value of the ACKE bit. No $\overline{\text{ACK}}$ is generated if the received address is not a local address (NACK).

When receiving the extension code, set the ACKE bit to 1 in advance to generate $\overline{\text{ACK}}$.

The ACK generation method during data reception is based on the wait timing setting, as described by the following.

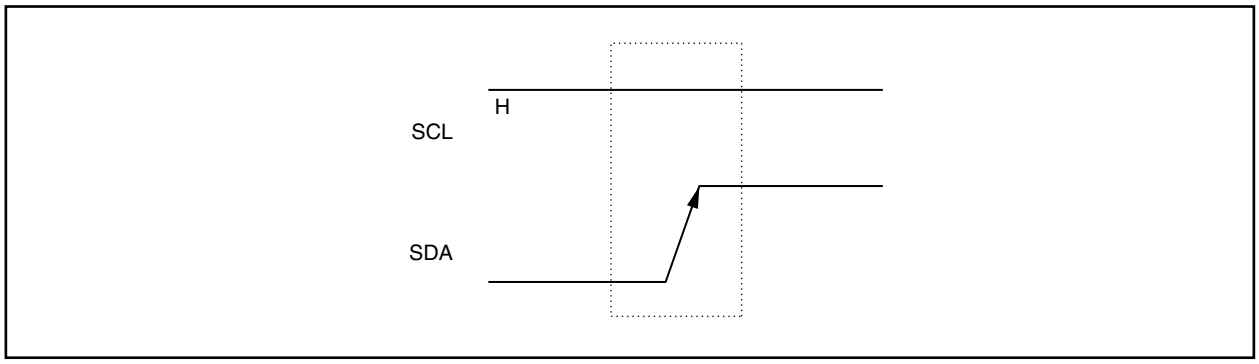
- When 8-clock wait is selected (IICC.WTIM bit = 0):
  $\overline{\text{ACK}}$ is generated at the falling edge of the SCL pin's eighth clock if the ACKE bit is set to 1 before the wait
- When 9-clock wait is selected (WTIMn bit = 1):
  $\overline{\text{ACK}}$ is generated if the ACKE bit is set to 1 in advance.

18.6.5 Stop condition

When the SCL pin is at high level, changing the SDA pin from low level to high level generates a stop condition.

A stop condition is generated when serial transfer from the master device to the slave device has been completed. Stop conditions can be detected when the device is used as a slave.

Figure 18-10. Stop Condition



A stop condition is generated when the IICC.SPT bit is set to 1. When the stop condition is detected, the IICS.SPD bit is set to 1 and the interrupt request signal (INTIIC) is generated when the IICC.SPIE bit is set to 1.

18.6.6 Wait

The wait is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCL pin to low level notifies the communication partner of the wait status. When wait status has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 18-11. Wait (1/2)

(a) When master device has a nine-clock wait and slave device has an eight-clock wait (master: transmission, slave: reception, and IICC.ACKE bit = 1)

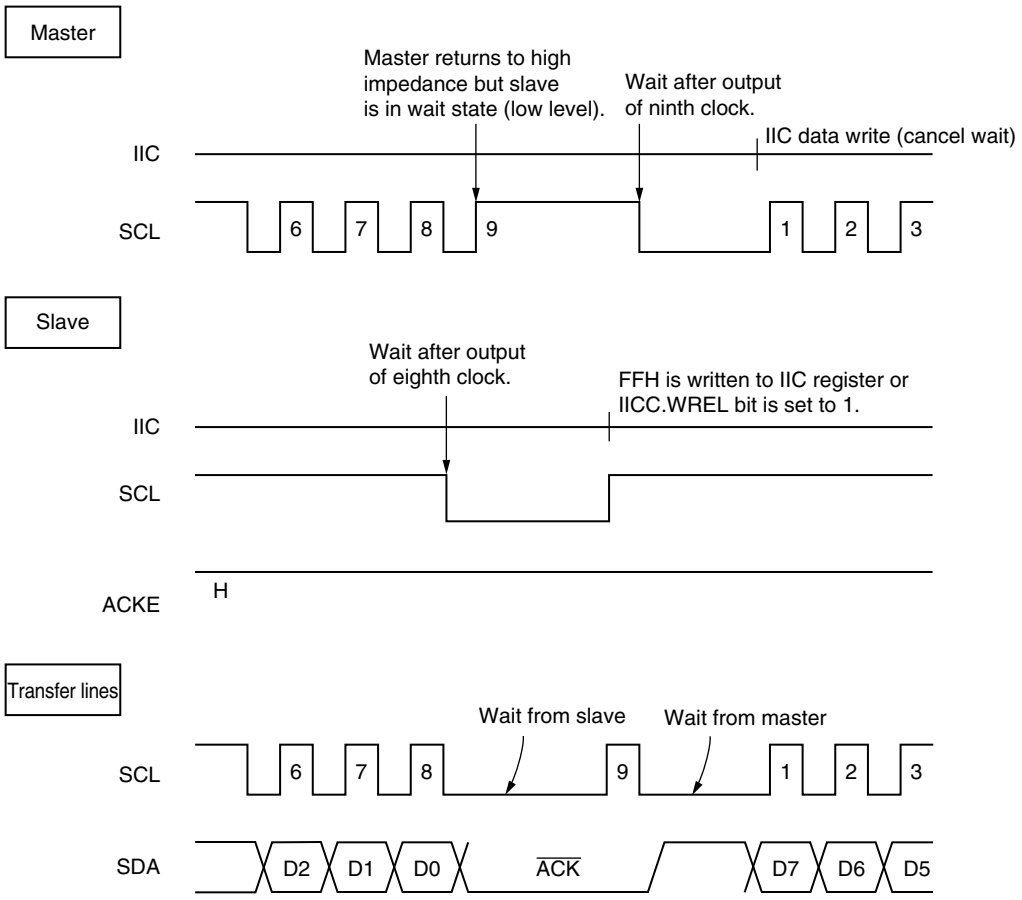
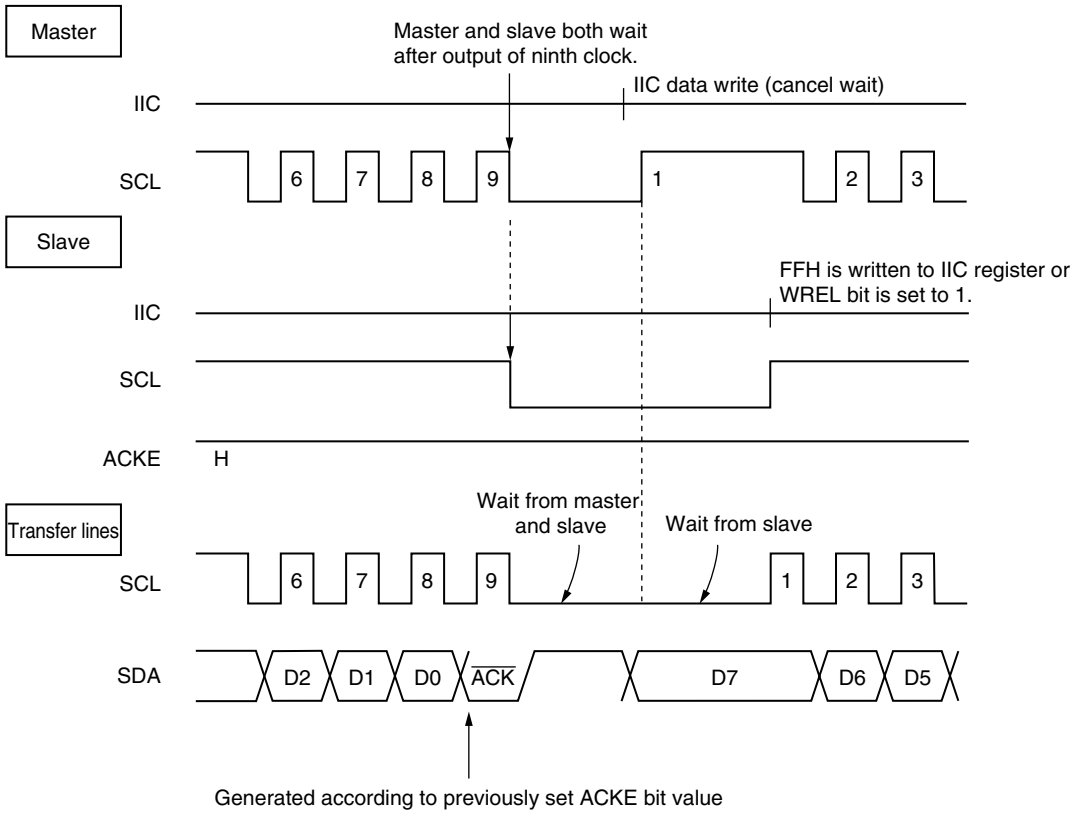


Figure 18-11. Wait (2/2)

(b) When master and slave devices both have a nine-clock wait (master: transmission, slave: reception, and ACKE bit = 1)



A wait state may be automatically set after a start condition is generated. Also, a wait state may be automatically set depending on the setting of the IICC.WTIM bit.

Normally, when the WREL bit is set to 1 or when FFH is written to the IIC register, the wait status is canceled and the transmitting side writes data to the IIC register to cancel the wait status.

The master device can also cancel the wait status via either of the following methods.

- By setting the IICC.STT bit to 1
- By setting the IICC.SPT bit to 1

<R> 18.6.7 Wait state cancellation method

In the case of I²C, wait state can be canceled normally in the following ways.

- By writing data to the IIC register
- By setting the IICC.WREL bit to 1 (wait state cancellation)
- By setting the IICC.STT bit to 1 (start condition generation)[Note]
- By setting the IICC.SPT bit to 1 (stop condition generation)[Note]

Note Master only

If any of these wait state cancellation actions is performed, I²C will cancel wait state and restart communication.

When canceling wait state and sending data (including address), write data to the IIC register.

To receive data after canceling wait state, or to complete data transmission, set the WREL bit to 1.

To generate a restart condition after canceling wait state, set the STT bit to 1.

To generate a stop condition after canceling wait state, set the SPT bit to 1.

Execute cancellation only once for each wait state.

For example, if data is written to the IIC register following wait state cancellation by setting the WREL bit to 1, conflict between the SDA line change timing and IIC register write timing may result in the data output to the SDA line may be incorrect.

Even in other operations, if communication is stopped halfway, clearing the IICC.IICE bit to 0 will stop communication, enabling wait state to be cancelled.

If the I²C bus dead-locks due to noise, etc., setting the IICC.LREL bit to 1 causes the communication operation to be exited, enabling wait state to be cancelled.

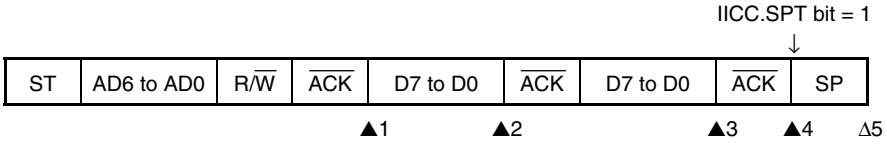
18.7 I²C Interrupt Request Signals (INTIIC)

The following shows the value of the IICS register at the INTIIC interrupt request signal generation timing and at the INTIIC signal timing.

18.7.1 Master device operation

(1) Start ~ Address ~ Data ~ Data ~ Stop (normal transmission/reception)

<1> When IICC.WTIM bit = 0



<R> ▲1: IICS register = 1000X110B

<R> ▲2: IICS register = 1000X000B

<R> ▲3: IICS register = 1000X000B (WTIM bit = 1[Note])

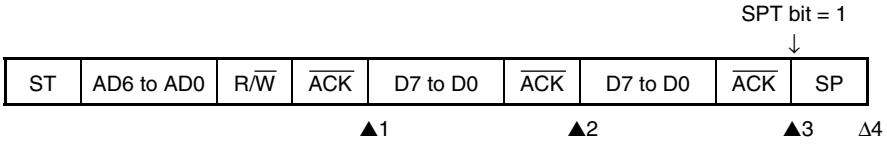
<R> ▲4: IICS register = 1000XX00B

Δ 5: IICS register = 00000001B

<R> **Note** To generate a stop condition, set the WTIM bit to 1 and change the timing of generation of the interrupt request signal (INTIIC).

Remark ▲: Always generated
Δ: Generated only when IICC.SPIE bit = 1
X: don't care

<2> When WTIM bit = 1



<R> ▲1: IICS register = 1000X110B

<R> ▲2: IICS register = 1000X100B

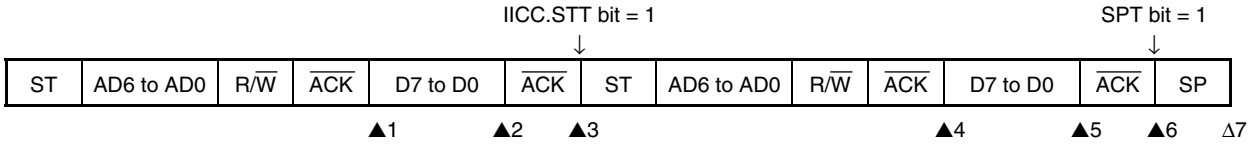
<R> ▲3: IICS register = 1000XX00B

Δ 4: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

<1> When WTIM bit = 0



<R> ▲1: IICS register = 1000X110B

<R> ▲2: IICS register = 1000X000B (WTIM bit = 1[Note 1])

<R> ▲3: IICS register = 1000XX00B (WTIM bit = 0[Note 2])

<R> ▲4: IICS register = 1000X110B

<R> ▲5: IICS register = 1000X000B (WTIM bit = 1[Note 3])

<R> ▲6: IICS register = 1000XX00B

Δ 7: IICS register = 00000001B

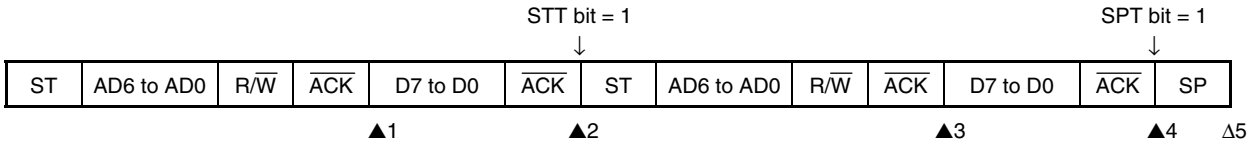
<R> **Notes 1**. Set the WTIM bit to 1 to generate a start condition and change the timing of generation of the interrupt request signal (INTIIC).

<R> **2.** Clear the WTIM bit to 0 to restore the original setting.

<R> **3.** To generate a stop condition, set the WTIM bit to 1 and change the timing of generation of the interrupt request signal (INTIIC).

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

<2> When WTIM bit = 1



<R> ▲1: IICS register = 1000X110B

<R> ▲2: IICS register = 1000XX00B

<R> ▲3: IICS register = 1000X110B

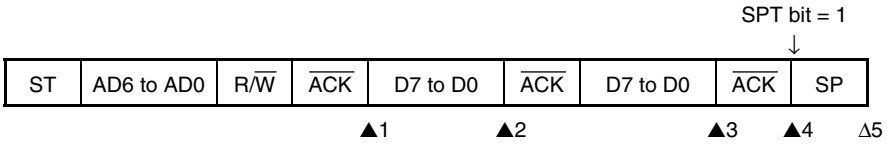
<R> ▲4: IICS register = 1000XX00B

Δ 5: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

(3) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

<1> When WTIM bit = 0



▲1: IICS register = 1010X110B

▲2: IICS register = 1010X000B

<R> ▲3: IICS register = 1010X000B (WTIM bit = 1[Note])

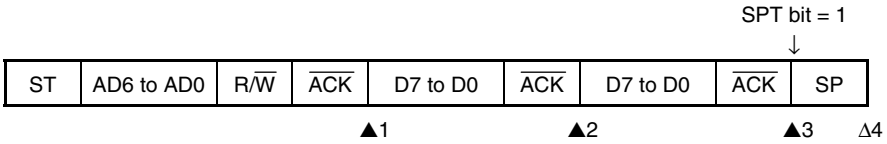
▲4: IICS register = 1010XX00B

Δ 5: IICS register = 00000001B

<R> **Note** To generate a stop condition, set the WTIM bit to 1 and change the timing of generation of the interrupt request signal (INTIIC).

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

<2> When WTIM bit = 1



▲1: IICS register = 1010X110B

▲2: IICS register = 1010X100B

▲3: IICS register = 1010XX00B

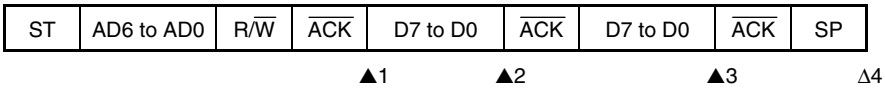
Δ 4: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

18.7.2 Slave device operation (when receiving slave address (match with address))

(1) Start ~ Address ~ Data ~ Data ~ Stop

<1> When IICC.WTIM bit = 0



▲1: IICS register = 0001X110B

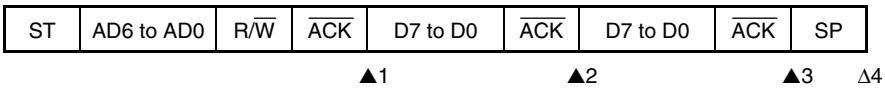
▲2: IICS register = 0001X000B

▲3: IICS register = 0001X000B

Δ 4: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when IICC.SPIE bit = 1
X: don't care

<2> When WTIM bit = 1



▲1: IICS register = 0001X110B

▲2: IICS register = 0001X100B

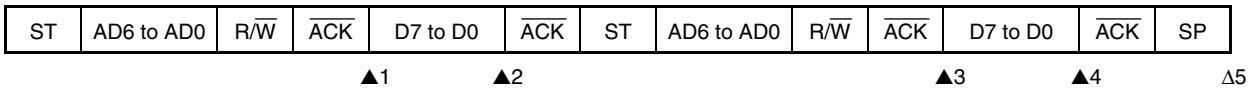
▲3: IICS register = 0001XX00B

Δ 4: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM bit = 0 (after restart, match with address)



▲1: IICS register = 0001X110B

▲2: IICS register = 0001X000B

▲3: IICS register = 0001X110B

▲4: IICS register = 0001X000B

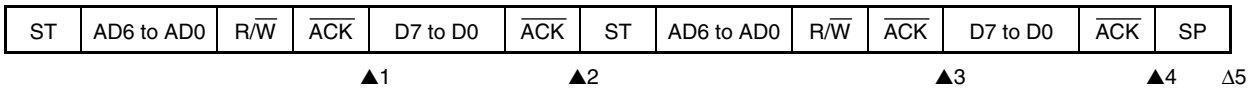
Δ 5: IICS register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE bit = 1

X: don't care

<2> When WTIM bit = 1 (after restart, match with address)



▲1: IICS register = 0001X110B

▲2: IICS register = 0001XX00B

▲3: IICS register = 0001X110B

▲4: IICS register = 0001XX00B

Δ 5: IICS register = 00000001B

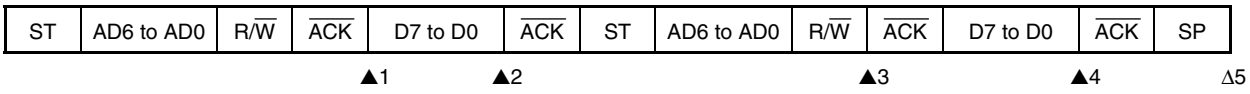
Remark ▲: Always generated

Δ: Generated only when SPIE bit = 1

X: don't care

(3) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIM bit = 0 (after restart, extension code reception)



▲1: IICS register = 0001X110B

▲2: IICS register = 0001X000B

▲3: IICS register = 0010X010B

▲4: IICS register = 0010X000B

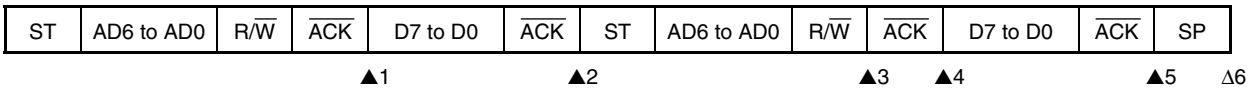
Δ 5: IICS register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE bit = 1

X: don't care

<2> When WTIM bit = 1 (after restart, extension code reception)



▲1: IICS register = 0001X110B

▲2: IICS register = 0001XX00B

▲3: IICS register = 0010X010B

▲4: IICS register = 0010X110B

▲5: IICS register = 0010XX00B

Δ 6: IICS register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE bit = 1

X: don't care

(4) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM bit = 0 (after restart, mismatch with address (= not extension code))

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1		▲2				▲3			Δ4

▲1: IICS register = 0001X110B

▲2: IICS register = 0001X000B

<R> ▲3: IICS register = 00000110B

Δ 4: IICS register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE bit = 1

X: don't care

<2> When WTIM bit = 1 (after restart, mismatch with address (= not extension code))

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1		▲2				▲3			Δ4

▲1: IICS register = 0001X110B

▲2: IICS register = 0001XX00B

<R> ▲3: IICS register = 00000110B

Δ 4: IICS register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE bit = 1

X: don't care

18.7.3 Slave device operation (when receiving extension code)

If an expansion code is received, the device always takes part in communication.

(1) Start ~ Code ~ Data ~ Data ~ Stop

<1> When IICC.WTIM bit = 0

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1: IICS register = 0010X010B

▲2: IICS register = 0010X000B

▲3: IICS register = 0010X000B

Δ 4: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when IICC.SPIE bit = 1
X: don't care

<2> When WTIM bit = 1

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICS register = 0010X010B

▲2: IICS register = 0010X110B

▲3: IICS register = 0010X100B

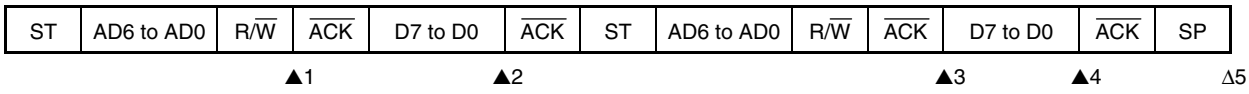
▲4: IICS register = 0010XX00B

Δ 5: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

(2) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM bit = 0 (after restart, match with address)



▲1: IICS register = 0010X010B

▲2: IICS register = 0010X000B

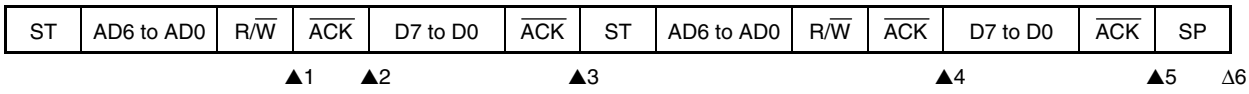
▲3: IICS register = 0001X110B

▲4: IICS register = 0001X000B

Δ 5: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

<2> When WTIM bit = 1 (after restart, match with address)



▲1: IICS register = 0010X010B

▲2: IICS register = 0010X110B

▲3: IICS register = 0010XX00B

▲4: IICS register = 0001X110B

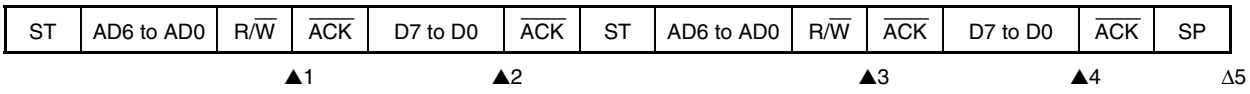
▲5: IICS register = 0001XX00B

Δ 6: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

(3) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIM bit = 0 (after restart, extension code reception)



▲1: IICS register = 0010X010B

▲2: IICS register = 0010X000B

▲3: IICS register = 0010X010B

▲4: IICS register = 0010X000B

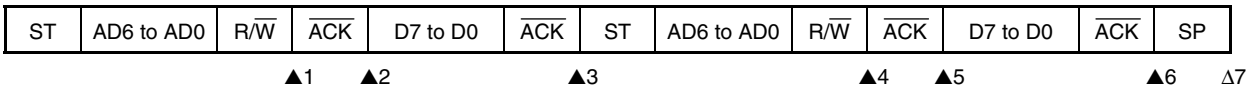
Δ 5: IICS register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE bit = 1

X: don't care

<2> When WTIM bit = 1 (after restart, extension code reception)



▲1: IICS register = 0010X010B

▲2: IICS register = 0010X110B

▲3: IICS register = 0010XX00B

▲4: IICS register = 0010X010B

▲5: IICS register = 0010X110B

▲6: IICS register = 0010XX00B

Δ 7: IICS register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE bit = 1

X: don't care

(4) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM bit = 0 (after restart, mismatch with address (= not extension code))

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1		▲2				▲3			Δ4

▲1: IICS register = 0010X010B

▲2: IICS register = 0010X000B

<R> ▲3: IICS register = 00000110B

Δ 4: IICS register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE bit = 1

X: don't care

<2> When WTIM bit = 1 (after restart, mismatch with address (= not extension code))

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1	▲2		▲3				▲4		Δ5

▲1: IICS register = 0010X010B

▲2: IICS register = 0010X110B

▲3: IICS register = 0010XX00B

<R> ▲4: IICS register = 00000110B

Δ 5: IICS register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE bit = 1

X: don't care

18.7.4 Operation without communication

(1) Start ~ Code ~ Data ~ Data ~ Stop

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

Δ1

Δ 1: IICS register = 00000001B

Remark Δ: Generated only when IICC.SPIE bit = 1

18.7.5 Arbitration loss operation (operation as slave after arbitration loss)

<R> When using the device as the master in a multimaster system, read to IICS.MSTS bit to check the result of arbitration each time the INTIIC interrupt has been generated

(1) When arbitration loss occurs during transmission of slave address data

<1> When IICC.WTIM bit = 0

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1: IICS register = 0101X110B

▲2: IICS register = 0001X000B

▲3: IICS register = 0001X000B

Δ 4: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when IICC.SPIE bit = 1
X: don't care

<2> When WTIM bit = 1

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1: IICS register = 0101X110B

▲2: IICS register = 0001X100B

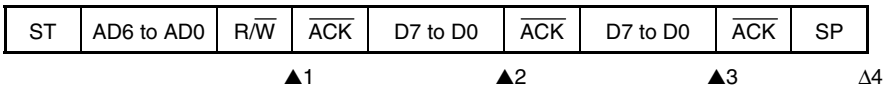
▲3: IICS register = 0001XX00B

Δ 4: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

(2) When arbitration loss occurs during transmission of extension code

<1> When WTIM bit = 0



▲1: IICS register = 0110X010B

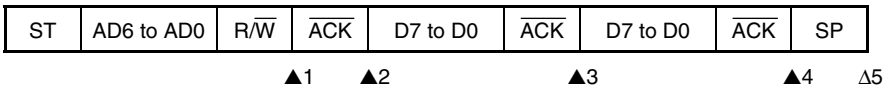
▲2: IICS register = 0010X000B

▲3: IICS register = 0010X000B

Δ 4: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

<2> When WTIM bit = 1



▲1: IICS register = 0110X010B

▲2: IICS register = 0010X110B

▲3: IICS register = 0010X100B

▲4: IICS register = 0010XX00B

Δ 5: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

18.7.6 Operation when arbitration loss occurs (no communication after arbitration loss)

<R> When using the device as the master in a multimaster system, read to IICS.MSTS bit to check the result of arbitration each time the INTIIC interrupt has been generated

(1) When arbitration loss occurs during transmission of slave address data

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 Δ2

▲1: IICS register = 01000110B

Δ 2: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when IICC.SPIE bit = 1

(2) When arbitration loss occurs during transmission of extension code

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 Δ2

▲1: IICS register = 0110X010B

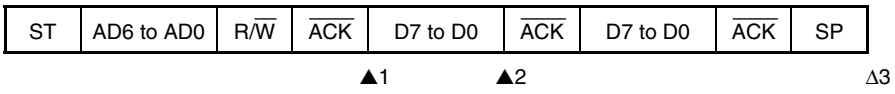
IICC.LREL bit is set to 1 by software

Δ 2: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

(3) When arbitration loss occurs during data transfer

<1> When IICC.WTIM bit = 0



▲1: IICS register = 10001110B

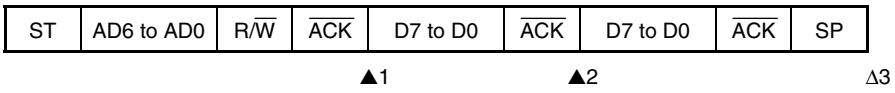
▲2: IICS register = 01000000B

Δ 3: IICS register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE bit = 1

<2> When WTIM bit = 1



▲1: IICS register = 10001110B

▲2: IICS register = 01000100B

Δ 3: IICS register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE bit = 1

(4) When loss occurs due to restart condition during data transfer

<1> Not extension code (Example: mismatches with address)

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to Dn	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1					▲2			Δ3

▲1: IICS register = 1000X110B

▲2: IICS register = 01000110B

Δ 3: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care
Dn = D6 to D0

<2> Extension code

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to Dn	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1					▲2			Δ3

▲1: IICS register = 1000X110B

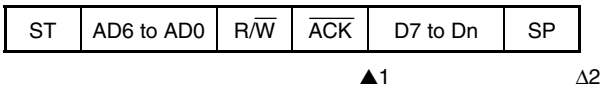
▲2: IICS register = 0110X010B

LREL bit is set to 1 by software

Δ 3: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care
Dn = D6 to D0

(5) When loss occurs due to stop condition during data transfer



▲1 Δ2

▲1: IICS register = 1000X110B

Δ 2: IICS register = 01000001B

Remark ▲: Always generated

Δ: Generated only when SPIE bit = 1

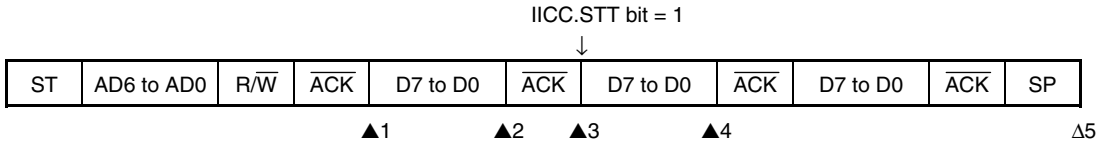
X: don't care

Dn = D6 to D0

(6) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

<R>

<1> When WTIM bit = 0



▲1: IICS register = 1000X110B

▲2: IICS register = 1000X000B (IICC.WTIM bit = 1)

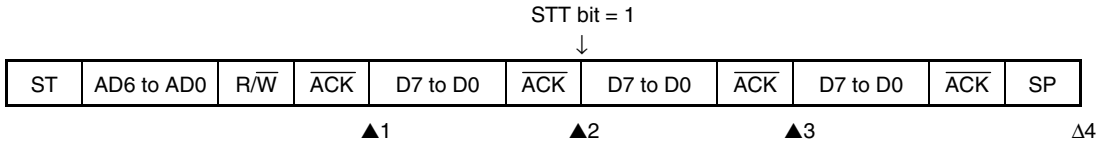
▲3: IICS register = 1000XX00B (WTIM bit = 0)

▲4: IICS register = 01000000B

Δ 5: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

<2> When WTIM bit = 1



▲1: IICS register = 1000X110B

▲2: IICS register = 1000X100B

▲3: IICS register = 01000100B

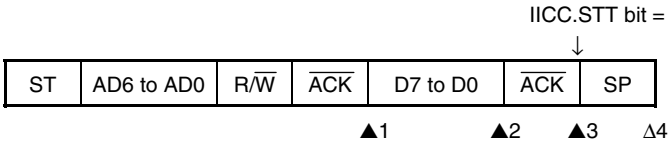
Δ 4: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

(7) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

<R>

<1> When WTIM bit = 0



▲1: IICS register = 1000X110B

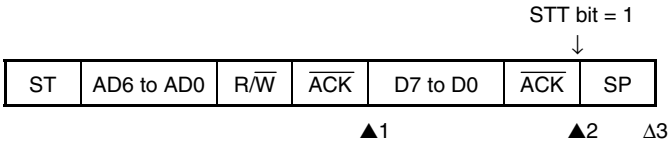
▲2: IICS register = 1000X000B (IICC.WTIM bit = 1)

▲3: IICS register = 1000XX00B (WTIM bit = 0)

Δ 4: IICS register = 01000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

<2> When WTIM bit = 1



▲1: IICS register = 1000X110B

▲2: IICS register = 1000XX00B

Δ 3: IICS register = 01000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

(8) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

<R>

<1> When WTIM bit = 0

IICC.SPT bit = 1 ↓

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
				▲1	▲2	▲3		▲4		Δ5

▲1: IICS register = 1000X110B

▲2: IICS register = 1000X000B (IICC.WTIM bit = 1)

▲3: IICS register = 1000X100B (WTIM bit = 0)

▲4: IICS register = 01000100B

Δ5: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

<2> When WTIM bit = 1

SPT bit = 1 ↓

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
				▲1		▲2		▲3		Δ4

▲1: IICS register = 1000X110B

▲2: IICS register = 1000X100B

▲3: IICS register = 01000100B

Δ 4: IICS register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE bit = 1
X: don't care

18.8 Interrupt Request Signal (INTIIC) Generation Timing and Wait Control

The setting of the IICC.WTIM bit determines the timing by which the INTIIC signal is generated and the corresponding wait control, as shown below.

Table 18-4. INTIIC Signal Generation Timing and Wait Control

WTIM Bit	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9 Notes 1, 2	8 Note 2	8 Note 2	9	8	8
1	9 Notes 1, 2	9 Note 2	9 Note 2	9	9	9

Notes 1. The slave device's INTIIC signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to the SVA register.
At this point, $\overline{ACK}$ is generated regardless of the value set to the IICC.ACKE bit. For a slave device that has received an extension code, the INTIIC signal occurs at the falling edge of the eighth clock.
When the address does not match after restart, the INTIIC signal is generated at the falling edge of the ninth clock, but no wait occurs.

2. If the received address does not match the contents of the SVA register and extension codes have not been received, neither the INTIIC signal nor a wait occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- Slave device operation: Interrupt and wait timing are determined depending on the conditions in Notes 1 and 2 above regardless of the WTIM bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- By writing data to the IIC register
- By setting the IICC.WREL bit (wait state cancellation)
- By setting the IICC.STT bit (start condition generation)
- By setting the IICC.SPT bit (stop condition generation)

Note Master only

When an 8-clock wait has been selected (WTIMn bit = 0), whether or not $\overline{ACK}$ has been generated must be determined prior to wait cancellation.

(5) Stop condition detection

The INTIIC signal is generated when a stop condition is detected.

18.9 Address Match Detection Method

When in I²C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match detection is performed automatically by hardware. An INTIIC interrupt request signal occurs when a local address has been set to the SVA register and when the address set to the SVA register matches the slave address sent by the master device, or when an extension code has been received.

18.10 Error Detection

In I²C bus mode, the status of the serial data bus (SDA) during data transmission is captured by the IIC register of the transmitting device, so the IIC register data prior to transmission can be compared with the transmitted IIC register data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

18.11 Extension Code

(1) When the higher 4 bits of the receive address are either 0000 or 1111, the extension code flag (EXC) is set for extension code reception and an interrupt request signal (INTIIC) is issued at the falling edge of the eighth clock. The local address stored in the SVA register is not affected.

(2) If 11110xx0 is set to the SVA register by a 10-bit address transfer and 11110xx0 is transferred from the master device, the results are as follows. Note that the INTIIC signal occurs at the falling edge of the eighth clock.

- Higher 4 bits of data match: IICS.EXC bit = 1
- 7 bits of data match: IICS.COI bit = 1

(3) Since the processing after the INTIIC signal occurs differs according to the data that follows the extension code, such processing is performed by software.

<R> If an extension code is received during slave operation, the device takes part in communication even if its address does not match.

For example, when operation as a slave is not desired after the extension code is received, set the IICC.LREL bit to 1. The CPU will enter the next communication wait state.

Table 18-5. Extension Code Bit Definitions

Slave Address	R/$\overline{W}$ Bit	Description
0000 000	0	General call address
0000 000	1	Start byte
0000 001	X	CBUS address
0000 010	X	Address that is reserved for different bus format
1111 0xx	X	10-bit slave address specification

18.12 Arbitration

When several master devices simultaneously generate a start condition (when the IICC.STT bit is set to 1 before the IICS.STD bit is set to 1), communication among the master devices is performed as the number of clocks is adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (IICS.ALD bit) is set (1) via the timing by which the arbitration loss occurred, and the SCL and SDA lines are both set for high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request signal (INTIIC) (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD bit = 1 setting that has been made by software.

For details of interrupt request signal generation timing, see **18.7 I²C Interrupt Request Signals (INTIIC)**.

Figure 18-12. Arbitration Timing Example

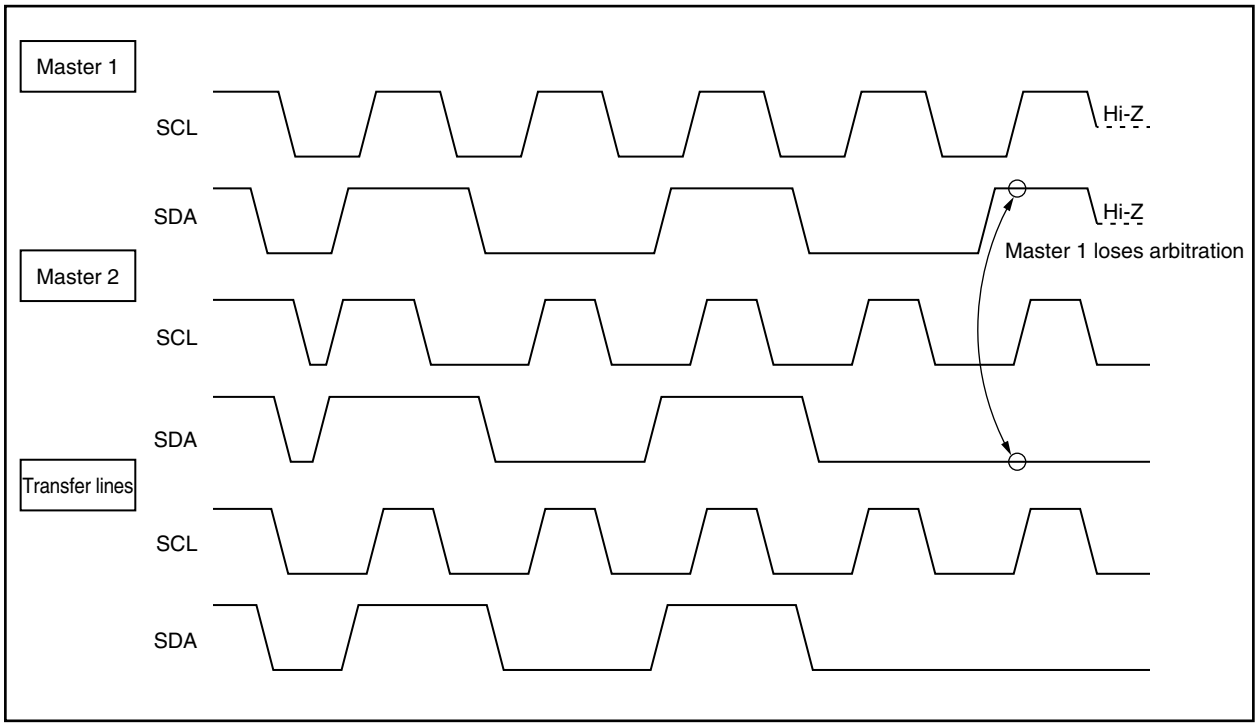


Table 18-6. Status During Arbitration and Interrupt Request Signal Generation Timing

Status During Arbitration	Interrupt Request Signal Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ transfer period after data reception	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when IICC.SPIE bit = 1)[Note 2]
When the SDA pin is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIE bit = 1)[Note 2]
When the SDA pin is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When the SCL pin is at low level while attempting to generate a restart condition	

Notes 1. When the IICC.WTIM bit = 1, an interrupt request signal occurs at the falling edge of the ninth clock. When the WTIM bit = 0 and the extension code's slave address is received, an interrupt request signal occurs at the falling edge of the eighth clock.

2. When there is a possibility that arbitration will occur, set the SPIE bit = 1 for master device operation.

18.13 Wakeup Function

The I²C bus slave function is a function that generates an interrupt request signal (INTIIC) when a local address or extension code has been received.

This function makes processing more efficient by preventing unnecessary interrupt request signals from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generate a start condition) to a slave device.

However, when a stop condition is detected, the IICC.SPIE bit is set regardless of the wake up function, and this determines whether interrupt request signals are enabled or disabled.

18.14 Communication Reservation

18.14.1 When communication reservation function is enabled (IICF.IICRSV bit = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released when the IICC.LREL bit was set to “1”).

If the IICC.STT bit is set (1) while the bus is not used, a start condition is automatically generated and wait status is set after the bus is released (after a stop condition is detected).

<R> When the IICC.SPIE bit is set to 1 and an address is written to the IIC register after release of the bus is detected by generation of an interrupt request signal (INTIIC) (stop condition detection), the device automatically starts communication as the master. Data written to the IIC register before the stop condition is detected is invalid.

When the STT bit has been set (1), the operation mode (as start condition or as communication reservation) is determined according to the bus status.

If the bus has been released .....a start condition is generated
If the bus has not been released (standby mode) .....communication reservation

To detect which operation mode has been determined for the STT bit, set the STT bit (1), wait for the wait period, then check the IICS.MSTS bit.

Wait periods, which should be set via software, are listed in Table 18-7. These wait periods can be set via the settings for the IICX.CLX, IICCL.SMC, and IICCL.CL0 bits.

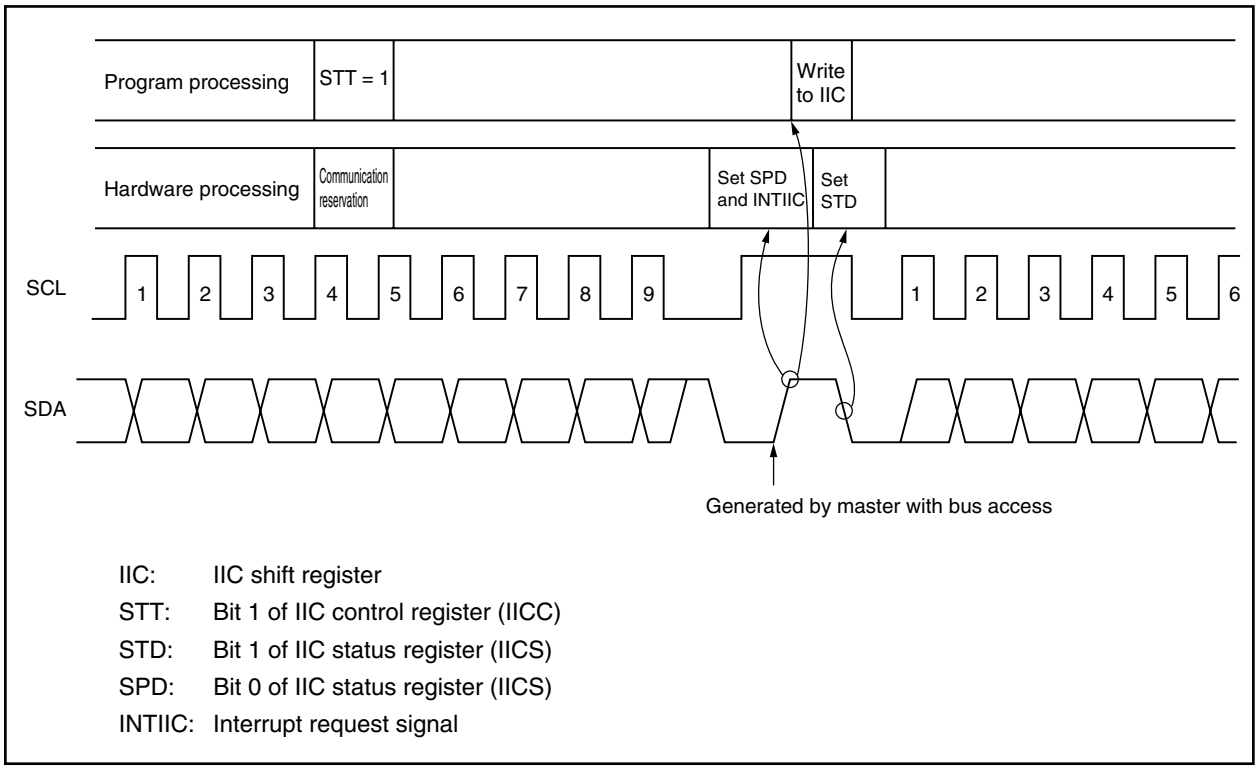
<R>

Table 18-7. Wait Periods

CLX Bit	SMC Bit	CL0 Bit	PRSCM Register	Division Clock	Wait Period
0	0	0	01H	$f_{XX}/4$	92 clocks
0	0	0	02H	$f_{XX}/8$	184 clocks
0	0	0	03H	$f_{XX}/12$	276 clocks
0	0	0	04H	$f_{XX}/16$	368 clocks
0	0	0	05H	$f_{XX}/20$	460 clocks
0	0	0	06H	$f_{XX}/24$	552 clocks
0	0	1	01H	$f_{XX}/4$	172 clocks
0	0	1	02H	$f_{XX}/8$	344 clocks
0	0	1	03H	$f_{XX}/12$	516 clocks
0	0	1	04H	$f_{XX}/16$	688 clocks
0	1	1/0	01H	$f_{XX}/4$	60 clocks
0	1	1/0	02H	$f_{XX}/8$	120 clocks
0	1	1/0	03H	$f_{XX}/12$	180 clocks
0	1	1/0	04H	$f_{XX}/16$	240 clocks
0	1	1/0	05H	$f_{XX}/20$	300 clocks
1	1	1/0	01H	$f_{XX}/4$	36 clocks
1	1	1/0	02H	$f_{XX}/8$	72 clocks
1	1	1/0	03H	$f_{XX}/12$	108 clocks
1	1	1/0	04H	$f_{XX}/16$	144 clocks
1	1	1/0	05H	$f_{XX}/20$	180 clocks

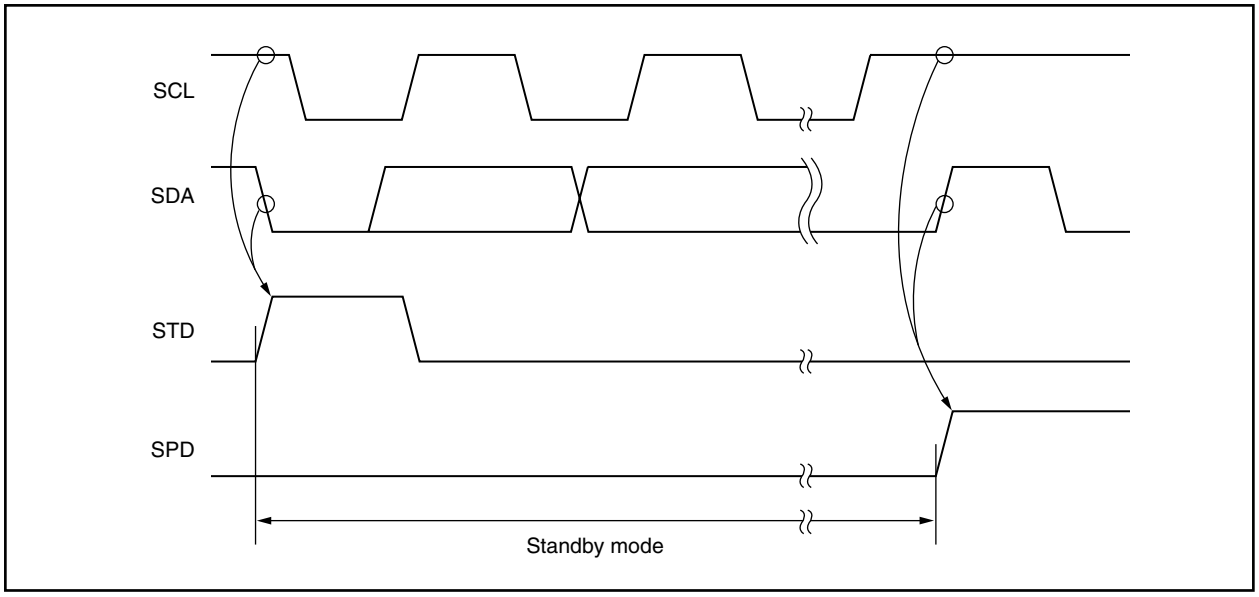
The communication reservation timing is shown below.

Figure 18-13. Communication Reservation Timing



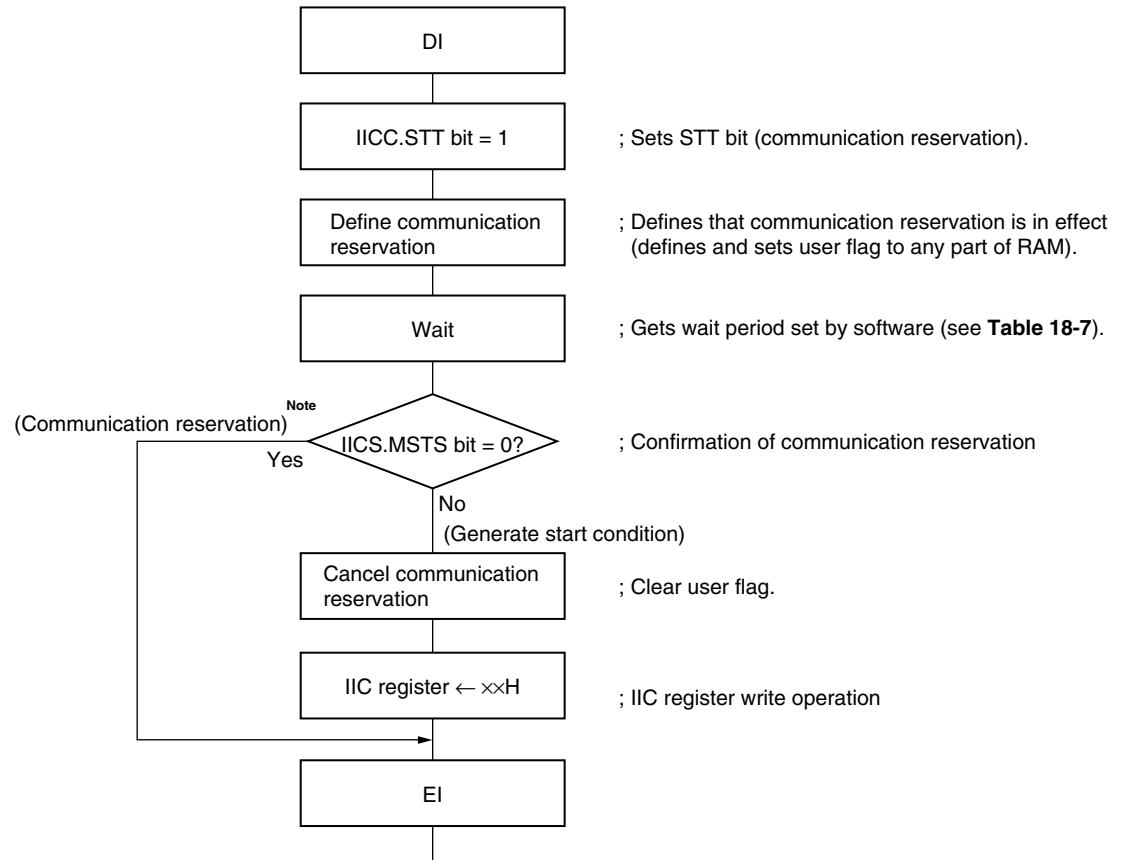
Communication reservations are accepted via the following timing. After the IICS.STD bit is set to 1, a communication reservation can be made by setting the IICC.STT bit to 1 before a stop condition is detected.

Figure 18-14. Timing for Accepting Communication Reservations



The communication reservation flowchart is illustrated below.

Figure 18-15. Communication Reservation Flowchart



Note The communication reservation operation executes a write to the IIC register when a stop condition interrupt request occurs.

18.14.2 When communication reservation function is disabled (IICF.IICRSV bit = 1)

When the IICC.STT bit is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released when the IICC.LREL bit was set to 1)

To confirm whether the start condition was generated or request was rejected, check the IICF.STCF flag. The wait time shown in Table 18-8 is required until the STCF flag is set after setting the STT bit = 1. Therefore, secure the time by software.

<R>

Table 18-8. Wait Periods

CL0 Bit	PRSCM Register	Division Clock	Wait Period
0	01H	$f_{XX}/4$	12 clocks
0	02H	$f_{XX}/8$	24 clocks
0	03H	$f_{XX}/12$	36 clocks
0	04H	$f_{XX}/16$	48 clocks
0	05H	$f_{XX}/20$	60 clocks
0	06H	$f_{XX}/24$	72 clocks
1	01H	$f_{XX}/4$	12 clocks
1	02H	$f_{XX}/8$	24 clocks
1	03H	$f_{XX}/12$	36 clocks
1	04H	$f_{XX}/16$	48 clocks

18.15 Cautions

(1) When IICF.STCEN bit = 0

Immediately after I²C operation is enabled, the bus communication status (IICF.IICBSY bit = 1) is recognized regardless of the actual bus status. To execute master communication in the status where a stop condition has not been detected, generate a stop condition and then release the bus before starting the master communication.

Use the following sequence for generating a stop condition.

<1> Set the IICCL register.
<2> Set the IICC.IICE bit.
<3> Set the IICC.SPT bit.

(2) When IICF.STCEN bit = 1

Immediately after I²C operation is enabled, the bus released status (IICBSY bit = 0) is recognized regardless of the actual bus status. To generate the first start condition (IICC.STT bit = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

<R> (3) When the IICC.IICE bit of the V850E/MA3 is set to 1 while communications with other devices are in progress, the start condition may be detected depending on the status of the communication line. Be sure to set the IICC.IICE bit to 1 when the SCL and SDA lines are high level.

<R> (4) Determine the operation clock frequency by the IICCL and IICX registers before enabling the operation (IICC.IICE bit = 1). To change the operation clock frequency, clear the IICC.IICE bit to 0 once.

<R> (5) After the IICC.STT and IICC.SPT bits have been set to 1, they must not be re-set without being cleared to 0 first.

<R> (6) If transmission has been reserved, set the IICC.SPIE bit to 1 so that an interrupt request is generated by the detection of a stop condition. After an interrupt request has been generated, the wait state will be released by writing communication data to I²C, then transferring will begin. If an interrupt is not generated by the detection of a stop condition, transmission will halt in the wait state because an interrupt request was not generated. However, it is not necessary to set the SPIE bit to 1 for the software to detect the IICS.MSTS bit.

<R> 18.16 Communication Operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the V850E/MA3 as the master in a single master system is shown below.

This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

(2) Master operation in multimaster system

In the I²C bus multimaster system, whether the bus is released or used cannot be judged by the I²C bus specifications when the bus takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the V850E/MA3 takes part in a communication with bus released state.

This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the V850E/MA3 looses in arbitration and is specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

(3) Slave operation

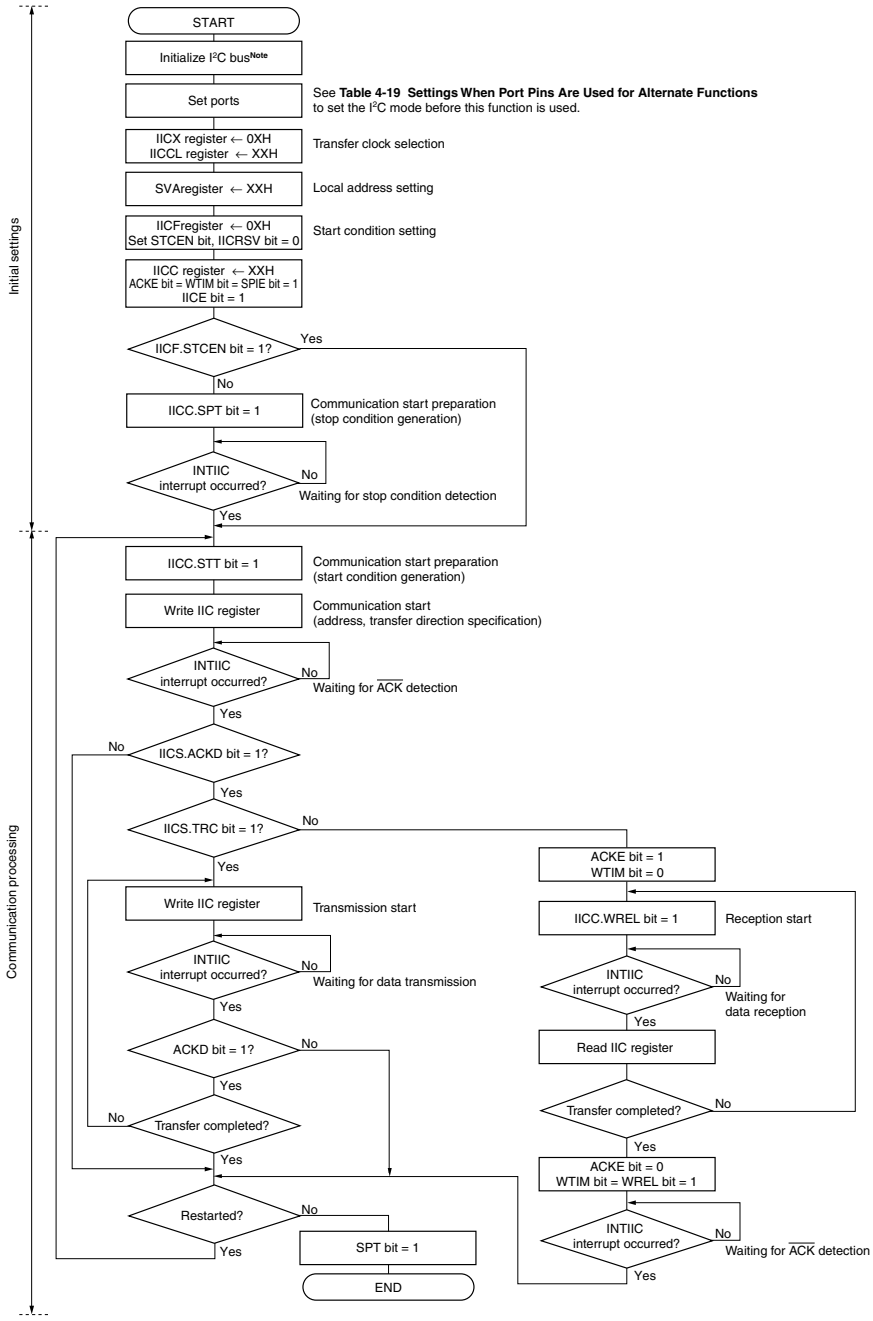
An example of when the V850E/MA3 is used as the slave of the I²C bus is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIIC interrupt occurrence (communication waiting). When the INTIIC interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

<R> 18.16.1 Master operation in single master system

Figure 18-16. Master Operation in Single Master System



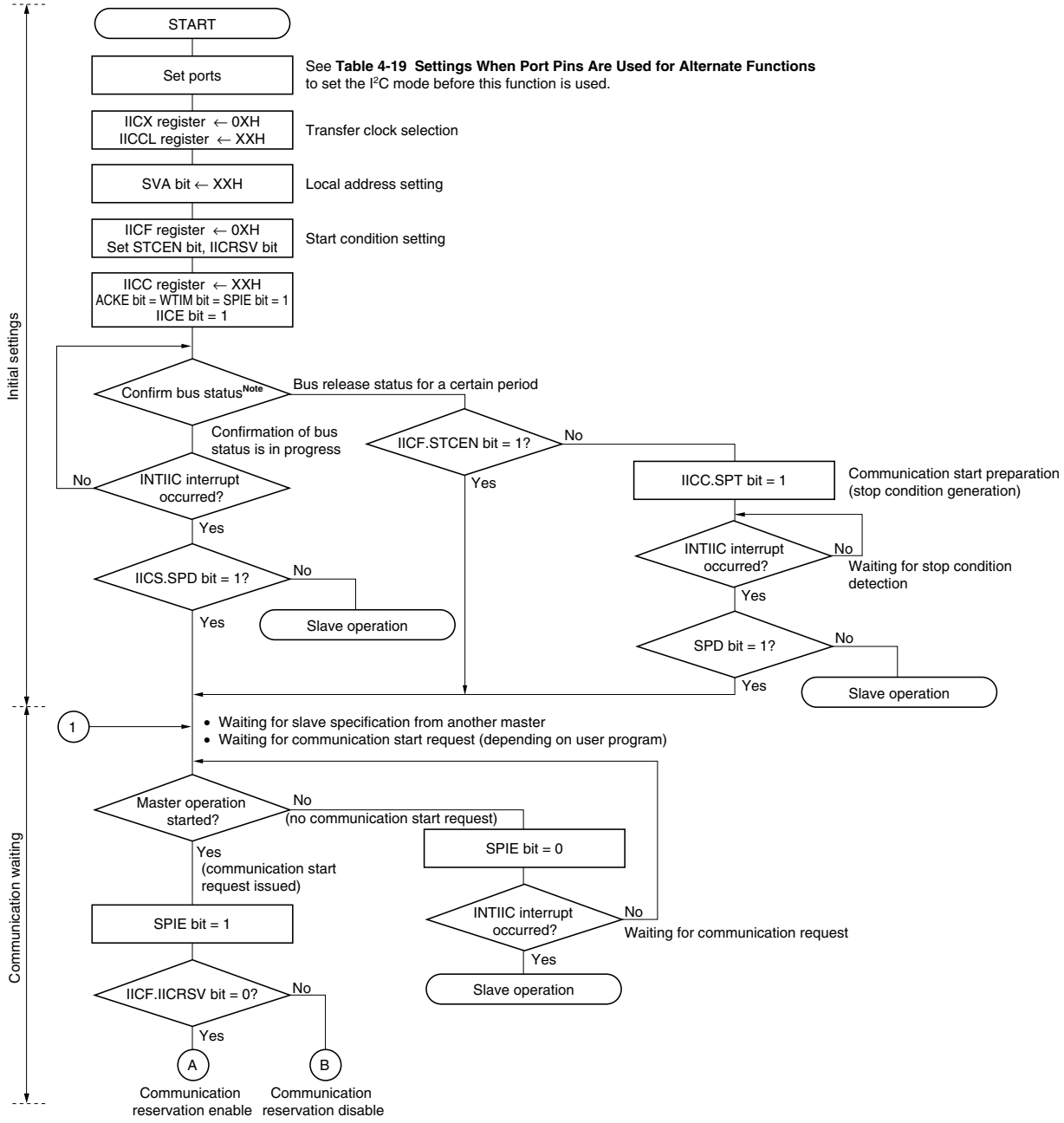
Note Release the I²C bus (SCL, SDA pins = high level) in conformity with the specifications of the product in communication.

For example, when the EEPROM outputs a low level to the SDA pin, set the SCL pin to the output port and output clock pulses from that output port until when the SDA pin is constantly high level.

Remark For the transmission and reception formats, conform to the specifications of the product in communication.

<R> 18.16.2 Master operation in multimaster system

Figure 18-17. Master Operation in Multimaster System (1/3)



Note Confirm that the bus release status (IICCL.CLD bit = 1, IICCL.DAD bit = 1) has been maintained for a certain period (1 frame, for example). When the SDA pin is constantly low level, determine whether to release the I²C bus (SCL, SDA pins = high level) by referring to the specifications of the product in communication.

Figure 18-17. Master Operation in Multimaster System (2/3)

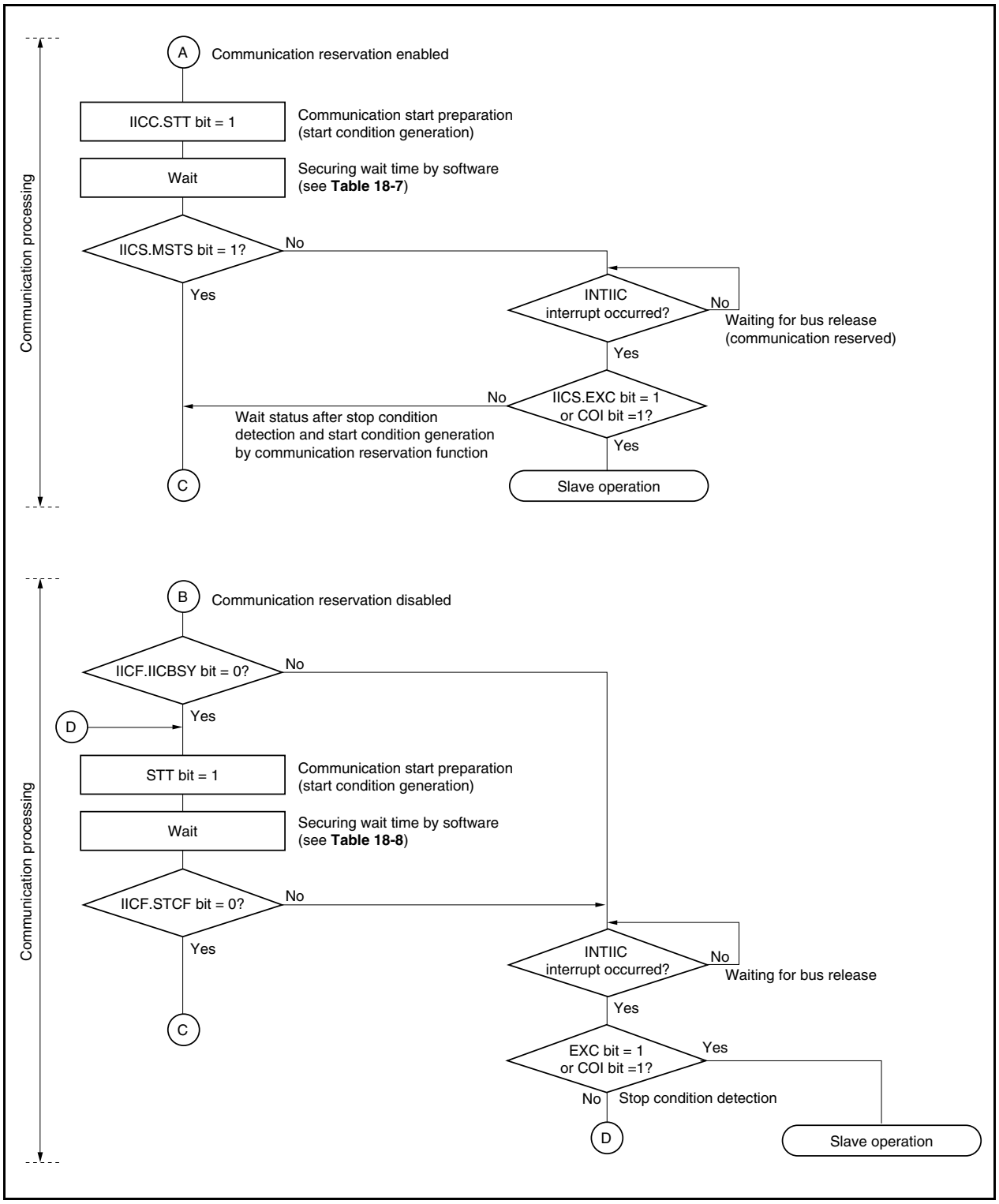
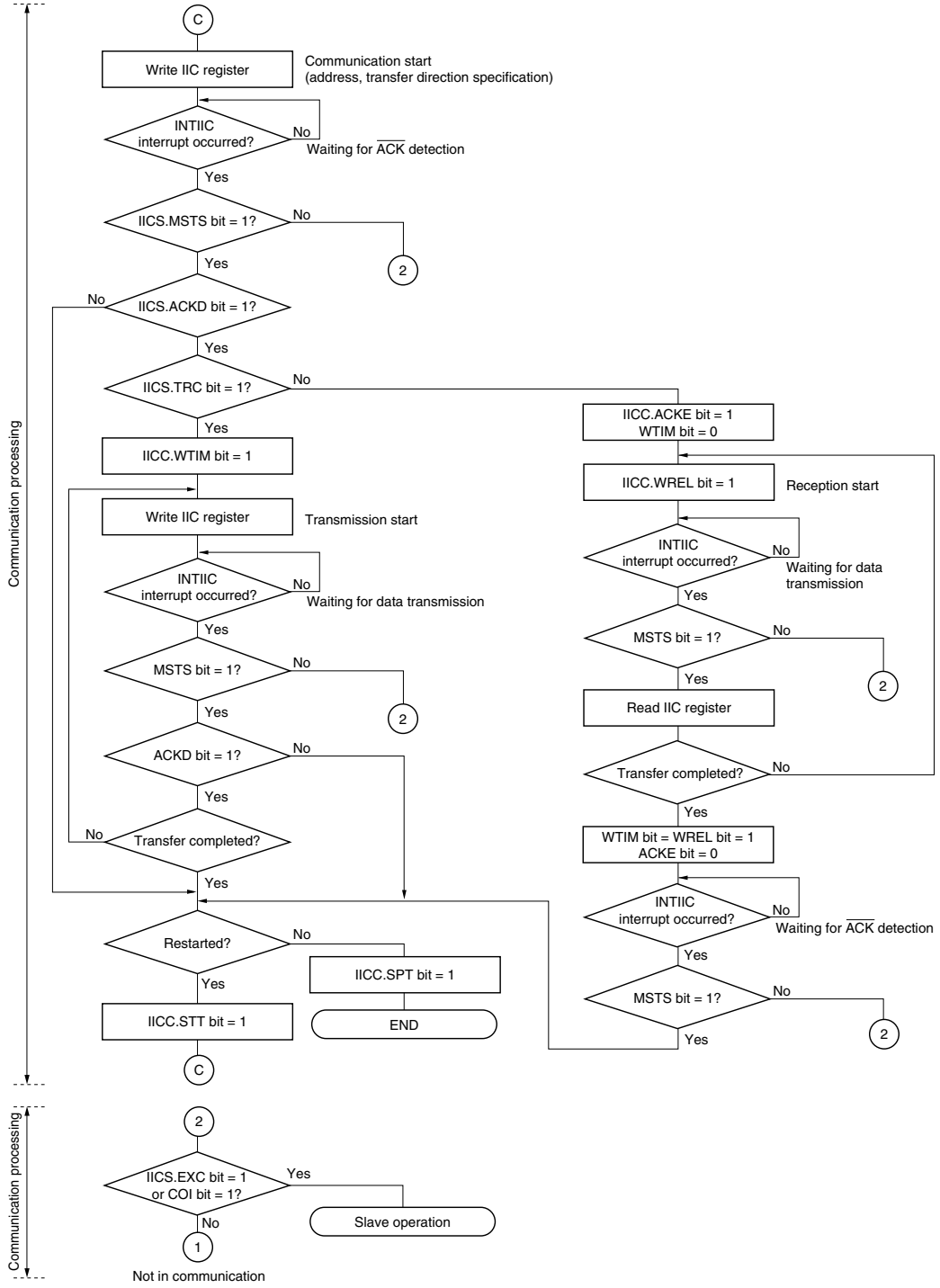


Figure 18-17. Master Operation in Multimaster System (3/3)



Remarks 1. Conform the transmission and reception formats to the specifications of the product in communication.

2. When using the V850E/MA3 as the master in the multimaster system, read the IICS.MSTS bit for each INTIIC interrupt occurrence to confirm the arbitration result.

3. When using the V850E/MA3 as the slave in the multimaster system, confirm the status using the IICS and IICF registers for each INTIIC interrupt occurrence to determine the next processing.

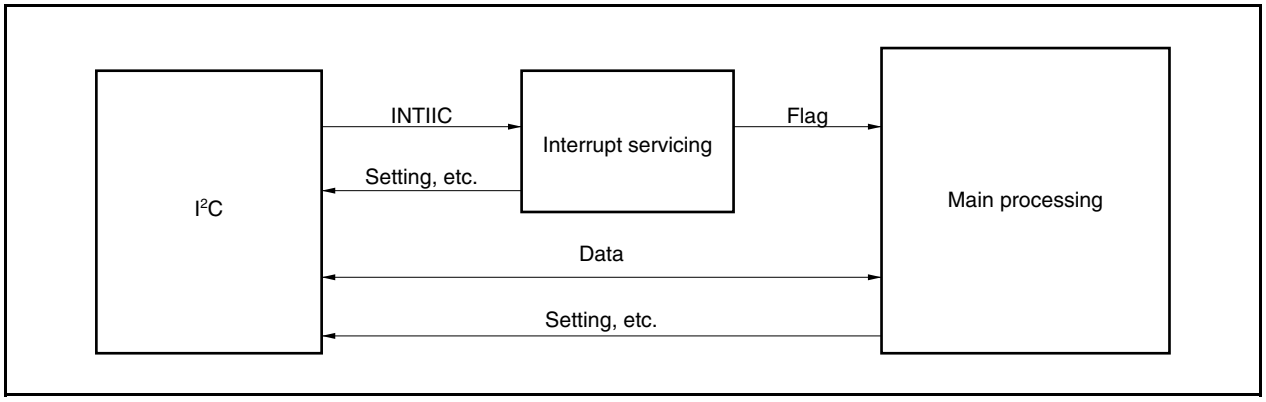
18.16.3 Slave operation

The following shows the processing procedure of the slave operation.

Basically, the operation of the slave device is event-driven. Therefore, processing by an INTIIC interrupt (processing requiring a significant change of the operation status, such as stop condition detection during communication) is necessary.

The following description assumes that data communication does not support extension codes. Also, it is assumed that the INTIIC interrupt servicing performs only status change processing and that the actual data communication is performed during the main processing.

Figure 18-18. Software Outline During Slave Operation



Therefore, the following three flags are prepared so that the data transfer processing can be performed by transmitting these flags to the main processing instead of the INTIIC signal.

(1) Communication mode flag

This flag indicates the following communication statuses.

Clear mode: Data communication not in progress

Communication mode: Data communication in progress (valid address detection stop condition detection, $\overline{\text{ACK}}$ from master not detected, address mismatch)

(2) Ready flag

This flag indicates that data communication is enabled. This is the same status as an INTIIC interrupt during normal data transfer. This flag is set in the interrupt processing block and cleared in the main processing block. The ready flag for the first data for transmission is not set in the interrupt processing block, so the first data is transmitted without clearance processing (the address match is regarded as a request for the next data).

(3) Communication direction flag

This flag indicates the direction of communication and is the same as the value of the IICS.TRC bit.

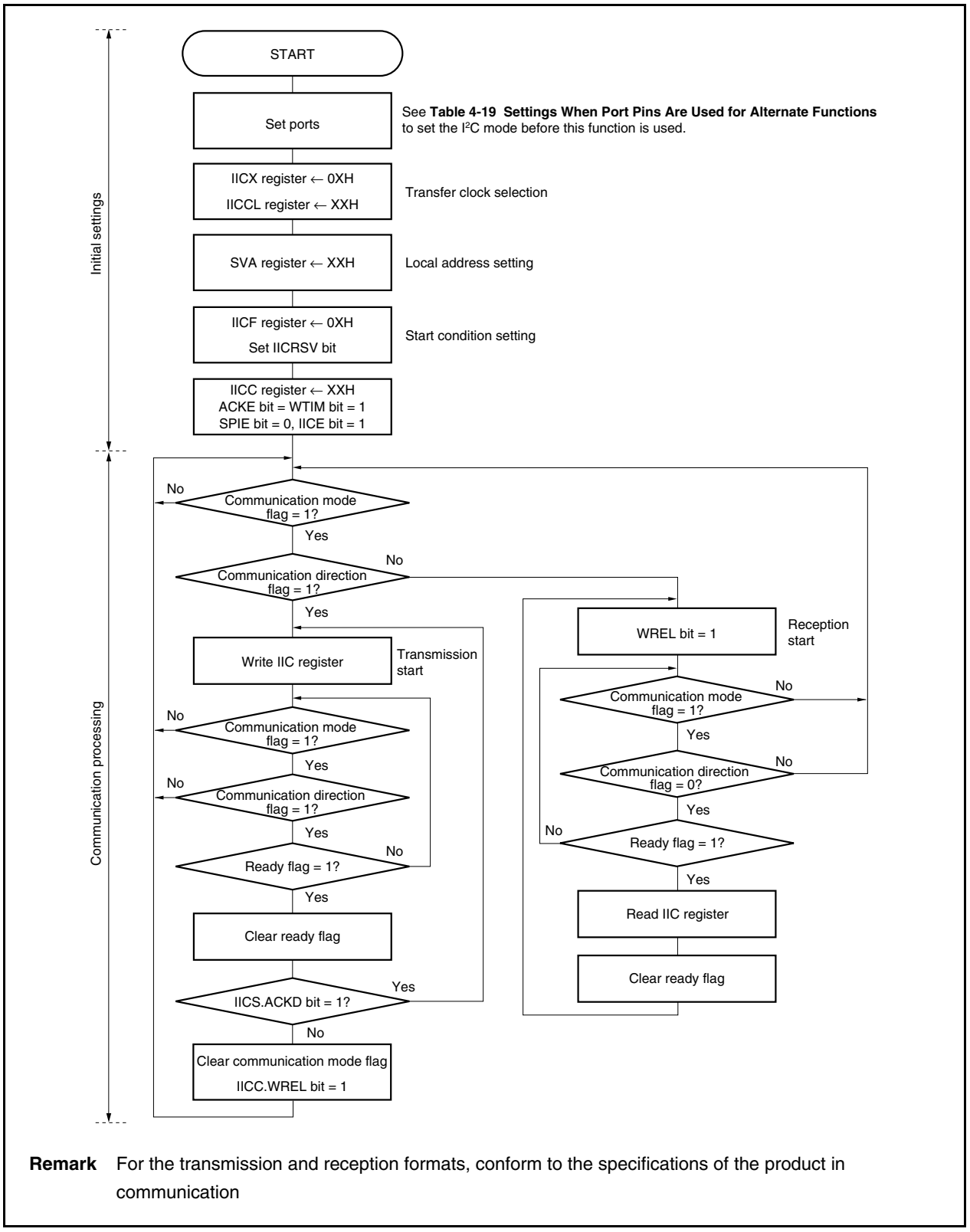
The following shows the operation of the main processing block during slave operation.

Start I²C and wait for the communication enabled status. When communication is enabled, perform transfer using the communication mode flag and ready flag (the processing of the stop condition and start condition is performed by interrupts, conditions are confirmed by flags).

For transmission, repeat the transmission operation until the master device stops returning $\overline{\text{ACK}}$. When the master device stops returning acknowledge, transfer is complete.

For reception, receive the required number of data and do not return $\overline{ACK}$ for the next data immediately after transfer is complete. After that, the master device generates the stop condition or restart condition. This causes exit from communications.

<R> **Figure 18-19. Slave Operation Flowchart (1)**



Remark For the transmission and reception formats, conform to the specifications of the product in communication

The following shows an example of the processing of the slave device by an INTIIC interrupt (it is assumed that no extension codes are used here). During an INTIIC interrupt, the status is confirmed and the following steps are executed.

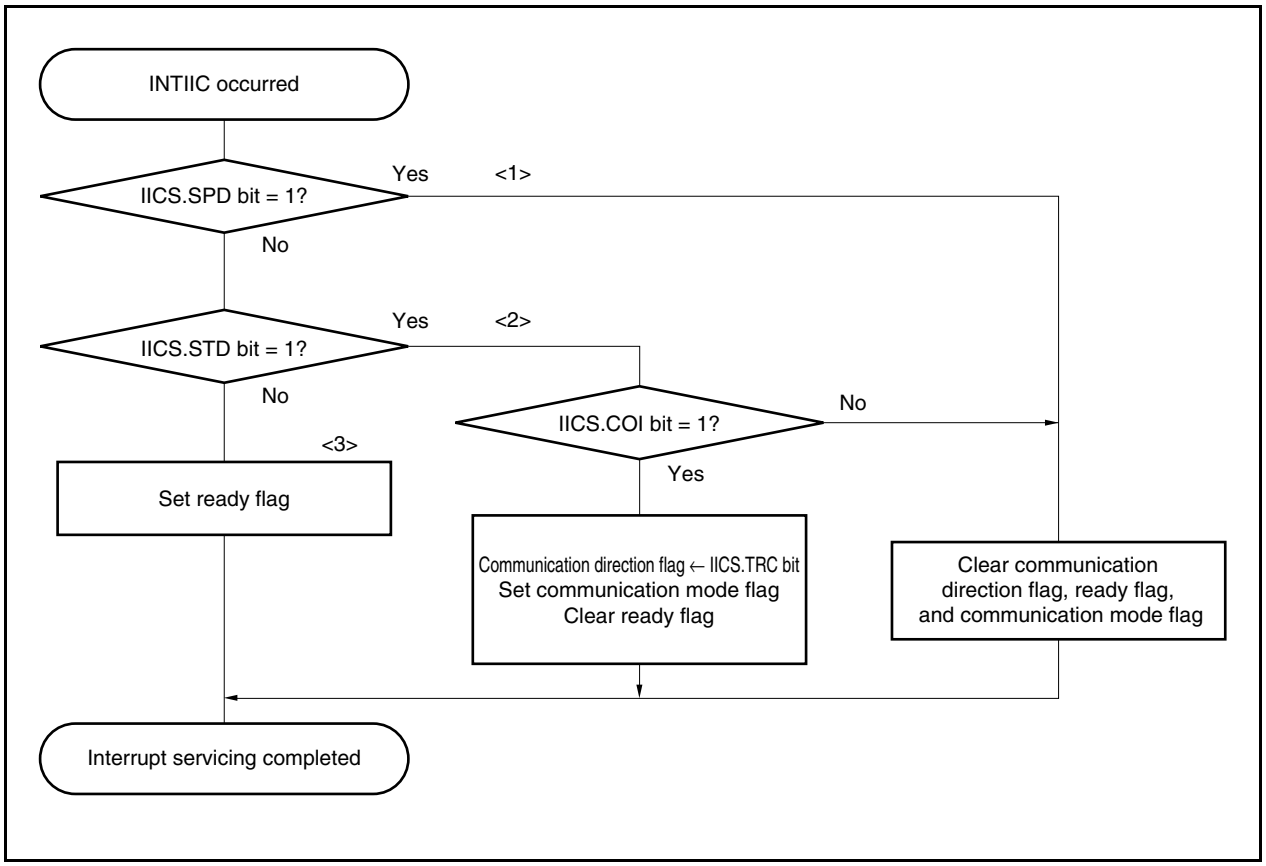
<1> When a stop condition is detected, communication is terminated.

<2> When a start condition is detected, the address is confirmed. If the address does not match, communication is terminated. If the address matches, the communication mode is set and wait is released, and operation returns from the interrupt (the ready flag is cleared).

<3> For data transmission/reception, when the ready flag is set, operation returns from the interrupt while the I²C bus remains in the wait status.

Remark <1> to <3> in the above correspond to <1> to <3> in **Figure 18-20 Slave Operation Flowchart (2)**.

<R> **Figure 18-20. Slave Operation Flowchart (2)**



18.17 Timing of Data Communication

When using I²C bus mode, the master device generates an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the IICS.TRC bit that specifies the data transfer direction and then starts serial communication with the slave device.

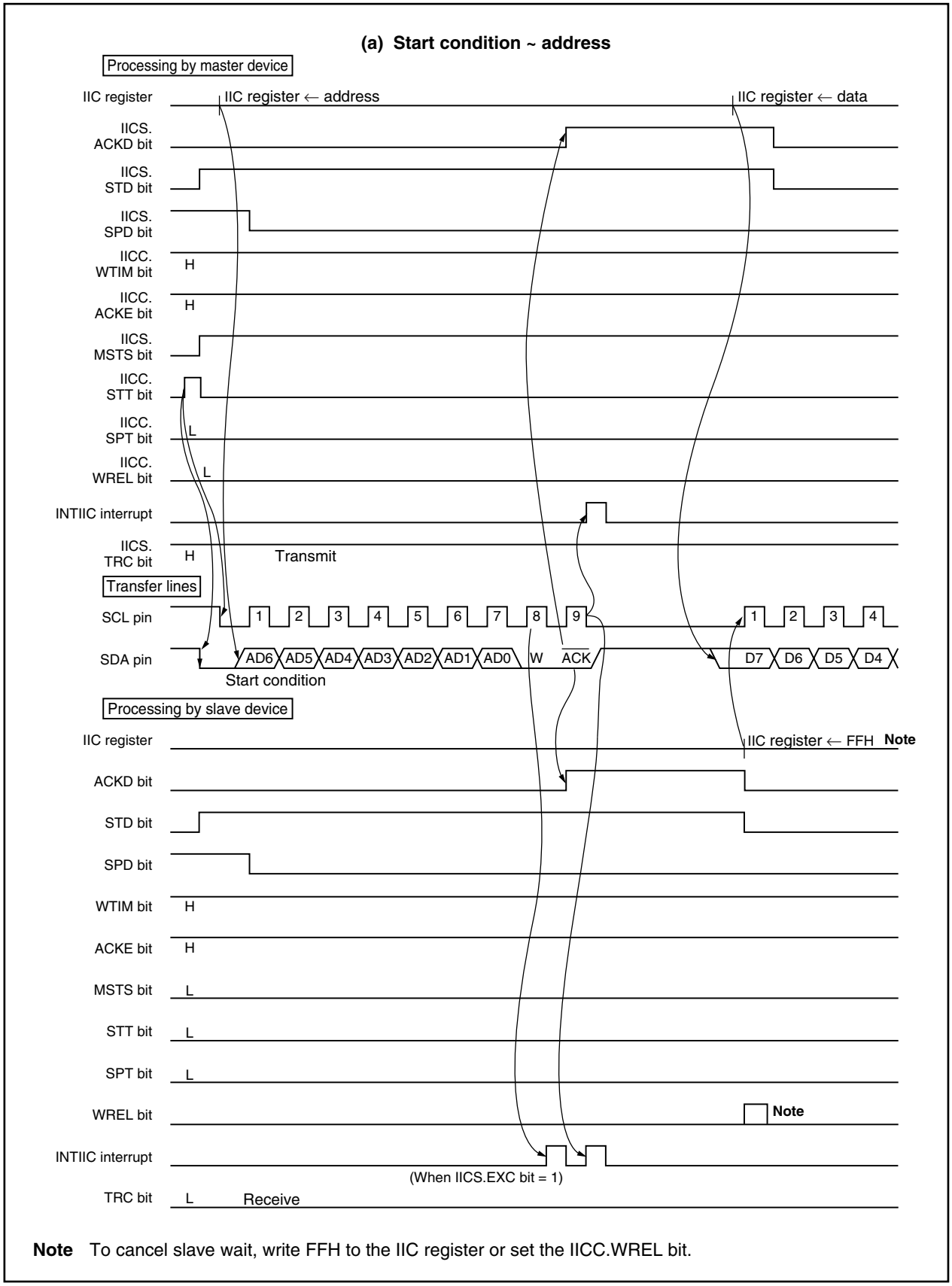
The IIC register's shift operation is synchronized with the falling edge of the serial clock (SCL pin). The transmit data is transferred to the SO latch and is output (MSB first) via the SDA pin.

Data input via the SDA pin is captured by the IIC register at the rising edge of the SCL pin.

The data communication timing is shown below.

<R>

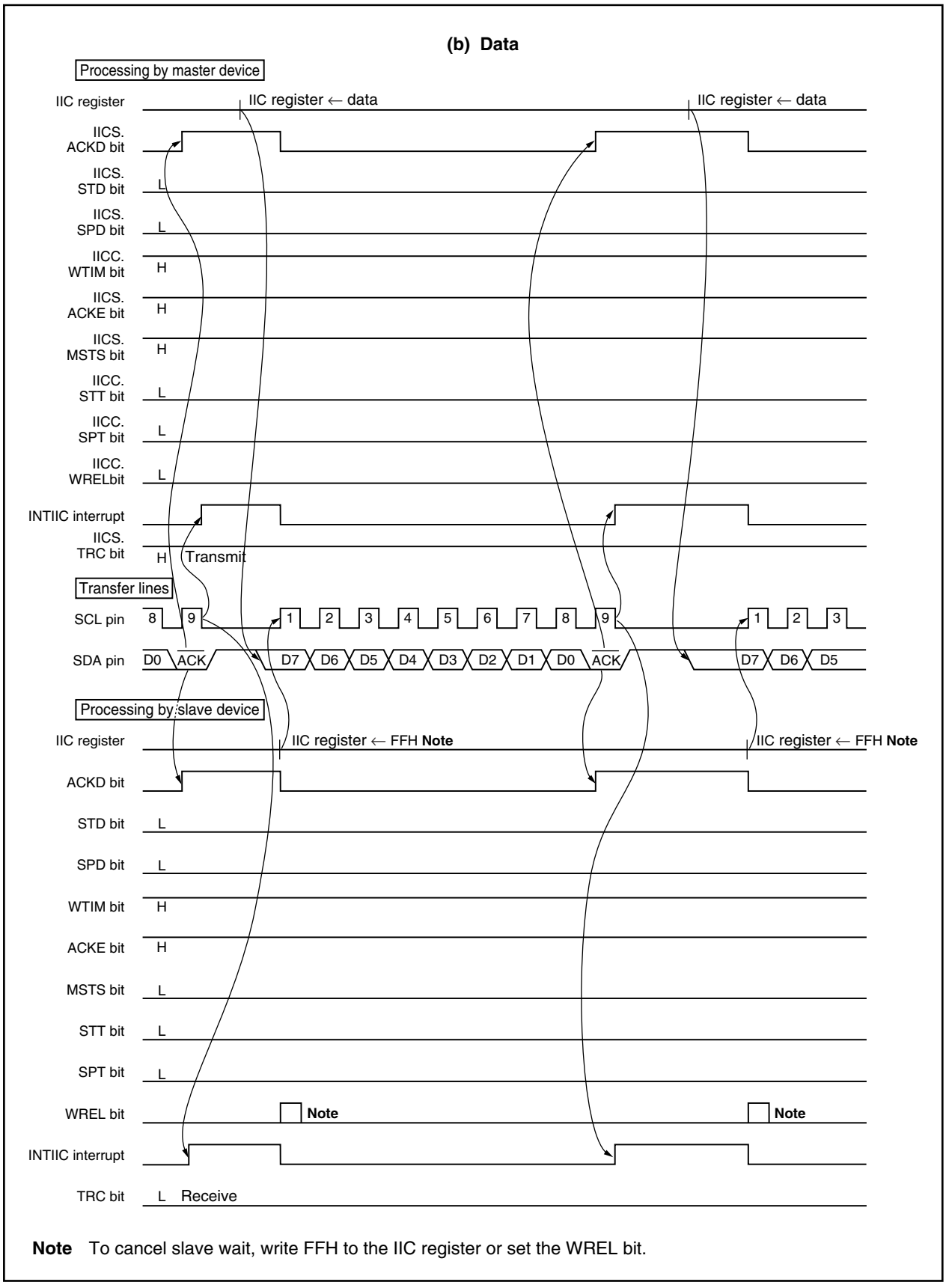
Figure 18-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)



Note To cancel slave wait, write FFH to the IIC register or set the IICC.WREL bit.

<R>

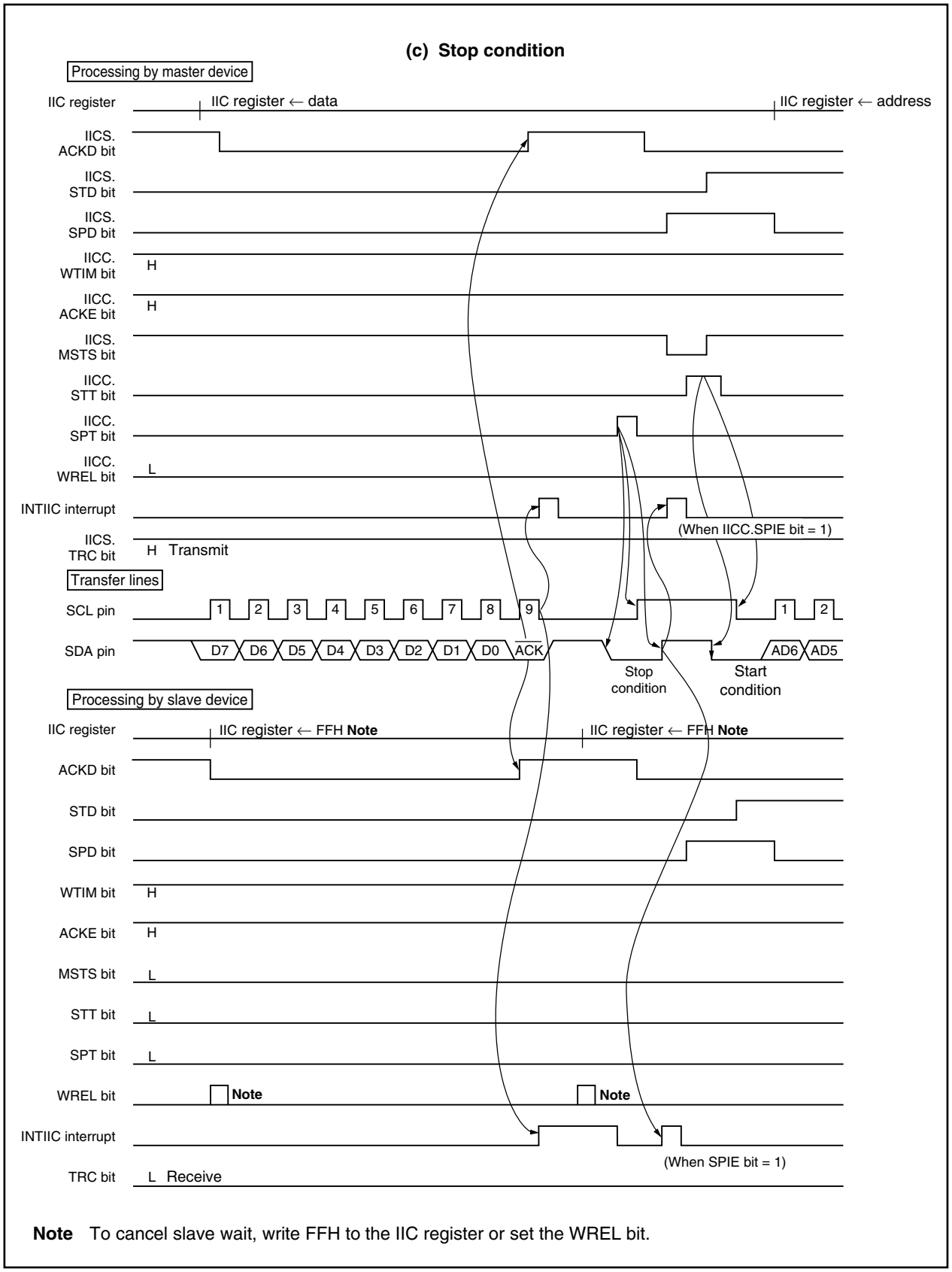
Figure 18-21. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)



Note To cancel slave wait, write FFH to the IIC register or set the WREL bit.

<R>

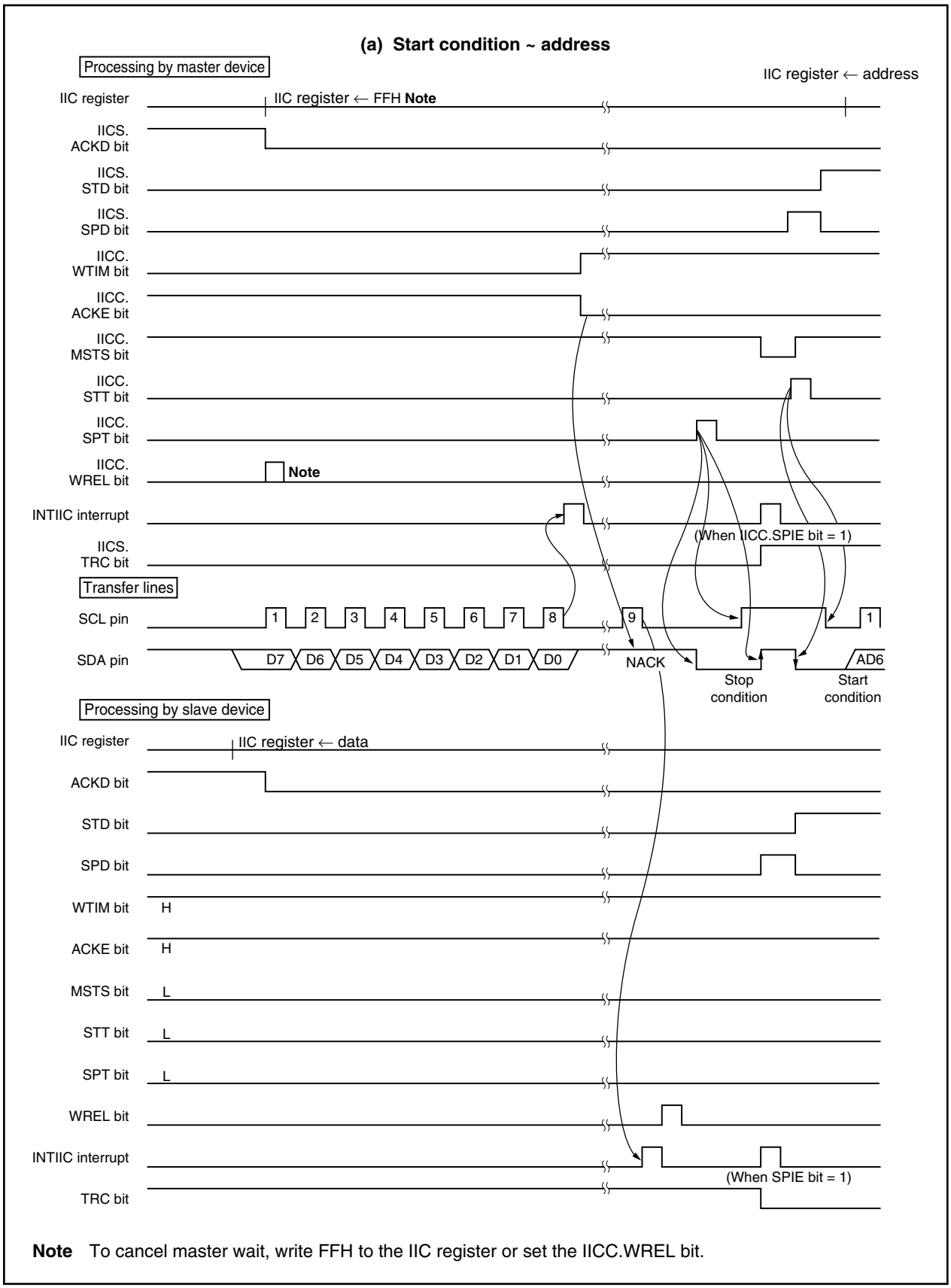
Figure 18-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)



Note To cancel slave wait, write FFH to the IIC register or set the WREL bit.

<R>

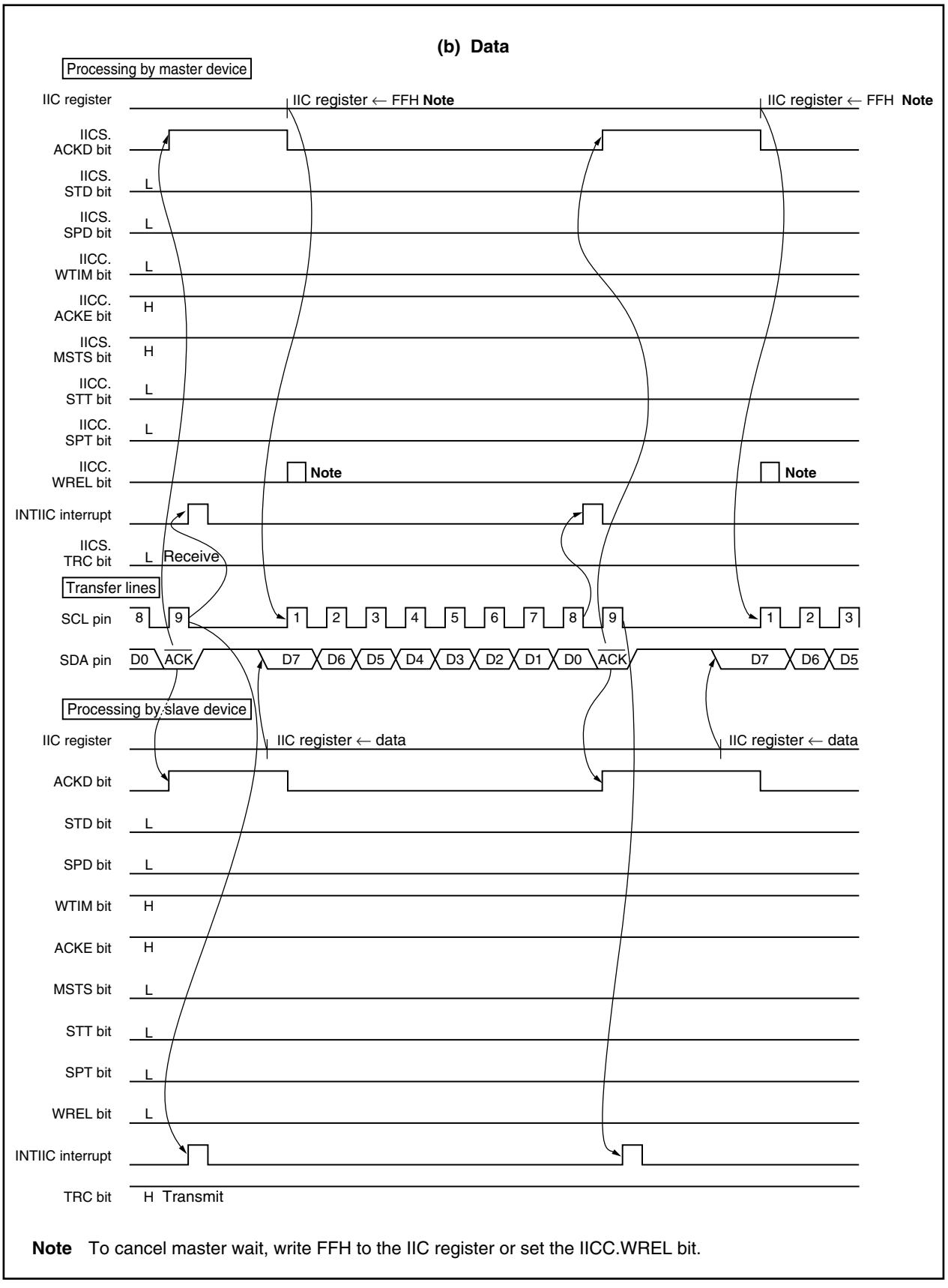
Figure 18-22. Example of Slave to Master Communication
(When 8-Clock → 9-Clock (Master)/9-Clock (Slave) Wait Is Selected) (1/3)



Note To cancel master wait, write FFH to the IIC register or set the IICC.WREL bit.

<R>

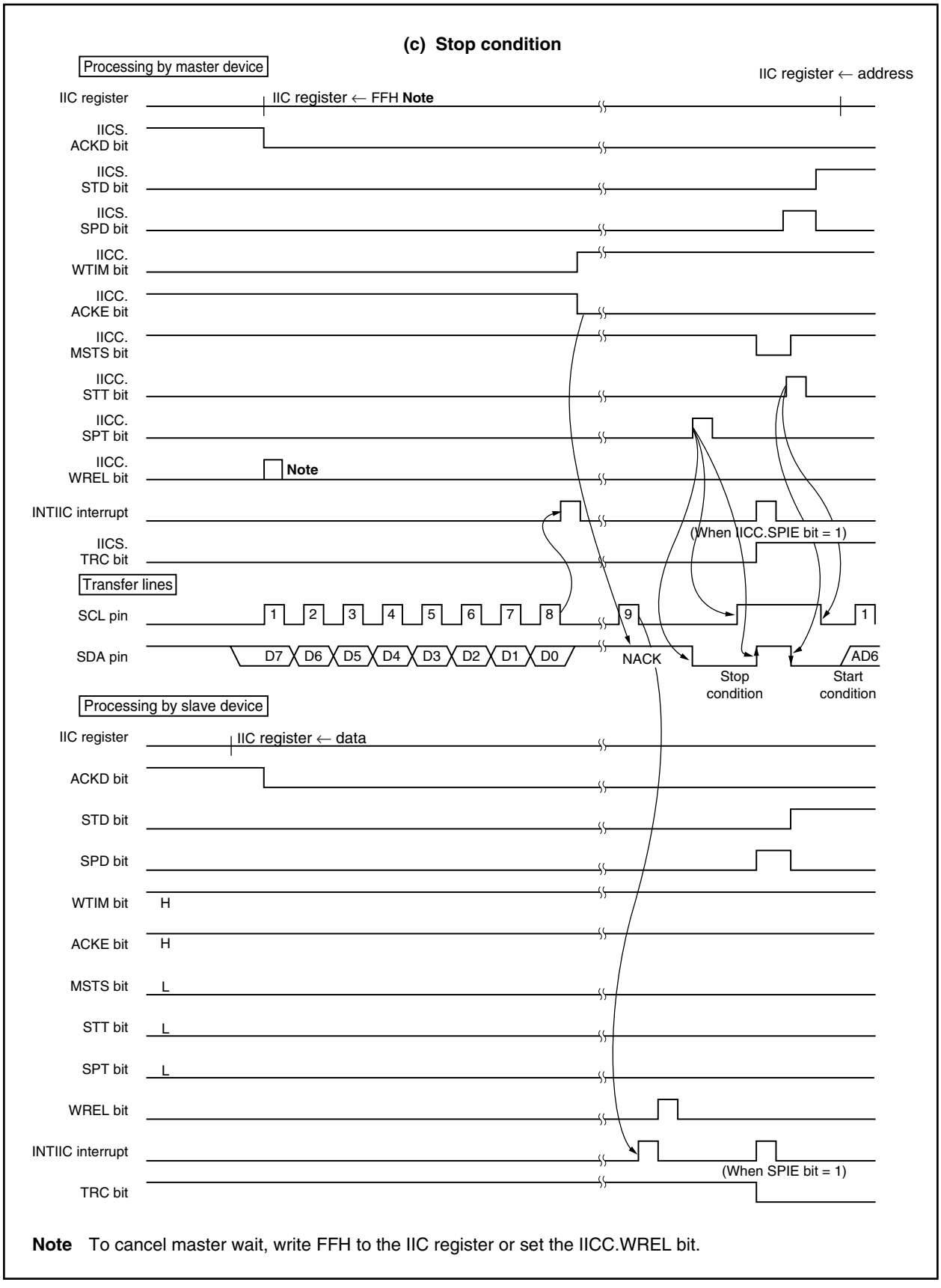
Figure 18-22. Example of Slave to Master Communication (When 8-Clock (Master)/9-Clock (Slave) Wait Is Selected) (2/3)



Note To cancel master wait, write FFH to the IIC register or set the IICC.WREL bit.

<R>

Figure 18-22. Example of Slave to Master Communication (When 8-Clock (Master)/9-Clock (Slave) Wait Is Selected) (3/3)



Note To cancel master wait, write FFH to the IIC register or set the IICC.WREL bit.

CHAPTER 19 DMA FUNCTIONS (DMA CONTROLLER)

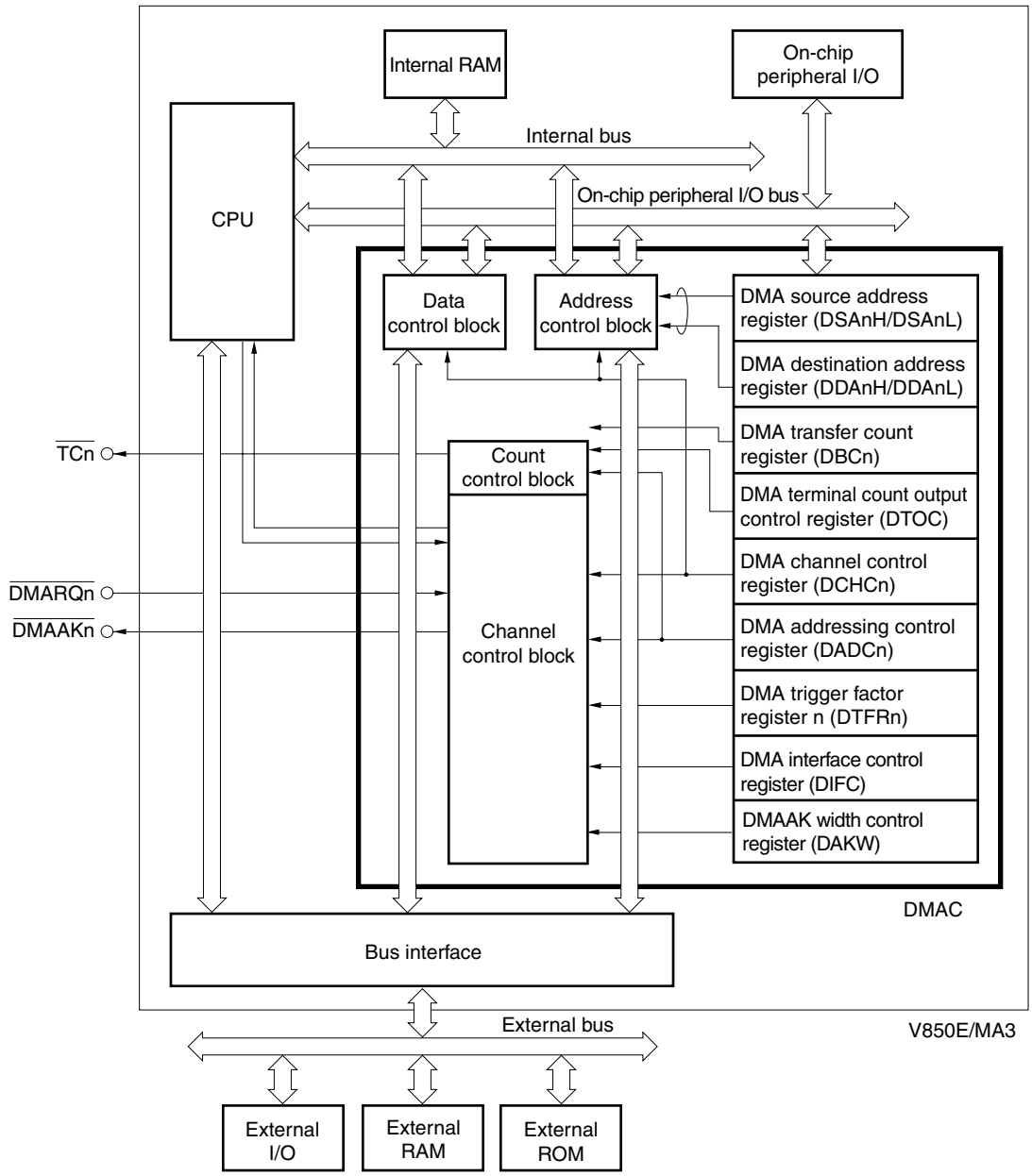
The V850E/MA3 includes a direct memory access (DMA) controller (DMAC) that executes and controls DMA transfer.

The DMAC controls data transfer between memory and I/O, or among memories, based on requests by interrupts from on-chip peripheral I/O (serial interface, timer/counter, and A/D converter) or DMA requests issued by the $\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$ pins or software triggers (memory refers to internal RAM or external memory).

19.1 Features

- 4 independent DMA channels
- Transfer unit: 8/16 bits
- Maximum transfer count: 65,536 (2^{16})
- Two types of transfer
  - Flyby (1-cycle) transfer (only in separate bus mode)
  - 2-cycle transfer
- Three transfer modes
  - Single transfer mode
  - Single-step transfer mode
  - Block transfer mode
- Transfer requests
  - Request by interrupts from on-chip peripheral I/O (serial interface, timer/counter, A/D converter)
  - Requests via $\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$ pin input
  - Requests by software trigger
- Transfer objects
  - Memory ↔ I/O
  - Memory ↔ memory
- DMA transfer end output signals ($\overline{\text{TC0}}$ to $\overline{\text{TC3}}$)
- Next address setting function

19.2 Configuration



Remark n = 0 to 3

19.3 Control Registers

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

The DSA0 to DSA3 registers set the DMA transfer source address (28 bits) for DMA channel n (n = 0 to 3). These registers are divided into two 16-bit registers, DSAnH and DSAnL.

Since these registers are configured as 2-stage FIFO buffer registers consisting of the master register and slave register, a new transfer source address for DMA transfer can be specified during DMA transfer (see **19.8 Next Address Setting Function**).

When flyby transfer is specified with the DADCn.TTYP bit, the external memory addresses are set by the DSAn register. At this time, the setting of the DDAn register is ignored (n = 0 to 3).

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

The DSA0H to DSA3H registers can be read or written in 16-bit units.

Reset input makes these registers undefined.

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

After reset: Undefined R/W Address: DSA0H FFFFF082H, DSA1H FFFFF08AH, DSA2H FFFFF092H, DSA3H FFFFF09AH

DSAnH (n = 0 to 3)

15	14	13	12	11	10	9	8
IR	0	0	0	SA27	SA26	SA25	SA24

7	6	5	4	3	2	1	0
SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16

IR	DMA transfer source specification
0	External memory, on-chip peripheral I/O
1	Internal RAM

SA27 to SA16	Set the DMA transfer source address (A27 to A16). During DMA transfer, these bits store the next DMA transfer source address. During flyby transfer, they store an external memory address.

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

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

The DSA0L to DSA3L registers can be read or written in 16-bit units.

Reset input makes these registers undefined.

After reset: Undefined R/W Address: DSA0L FFFFF080H, DSA1L FFFFF088H, DSA2L FFFFF090H, DSA3L FFFFF098H

DSAnL (n = 0 to 3)

15	14	13	12	11	10	9	8
SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8

7	6	5	4	3	2	1	0
SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0

SA15 to SA0	Set the DMA transfer source address (A15 to A0). During DMA transfer, these bits store the next DMA transfer source address. During flyby transfer, they store an external memory address.

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

The DDA0 to DDA3 registers set the DMA transfer destination address (28 bits) for DMA channel n (n = 0 to 3). They are divided into two 16-bit registers, DDAnH and DDAnL.

Since these registers are configured as 2-stage FIFO buffer registers consisting of the master register and slave register, a new transfer destination address for DMA transfer can be specified during DMA transfer (see **19.8 Next Address Setting Function**).

When flyby transfer is specified with the DADCn.TTYP bit, the setting of the DDAn register is ignored.

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

The DDA0H to DDA3H registers can be read or written in 16-bit units.

Reset input makes these registers undefined.

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

After reset: Undefined R/W Address: DDA0H FFFFF086H, DDA1H FFFFF08EH, DDA2H FFFFF096H, DDA3H FFFFF09EH

DDAnH (n = 0 to 3)

15	14	13	12	11	10	9	8
IR	0	0	0	DA27	DA26	DA25	DA24

7	6	5	4	3	2	1	0
DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16

IR	DMA transfer destination specification
0	External memory, on-chip peripheral I/O
1	Internal RAM

DA27 to DA16	Set the DMA transfer destination address (A27 to A16). During DMA transfer, these bits store the next DMA transfer destination address. This setting is ignored during flyby transfer.

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

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

The DDA0L to DDA3L registers can be read or written in 16-bit units.

Reset input makes these registers undefined.

After reset: Undefined R/W Address: DDA0L FFFFF084H, DDA1L FFFFF08CH, DDA2L FFFFF094H, DDA3L FFFFF09CH

DDAnL (n = 0 to 3)

15	14	13	12	11	10	9	8
DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8

7	6	5	4	3	2	1	0
DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

DA15 to DA0	Set the DMA transfer destination address (A15 to A0). During DMA transfer, these bits store the next DMA transfer destination address. This setting is ignored during flyby transfer.

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

The DBC0 to DBC3 registers are 16-bit registers that set the byte transfer count for DMA channel n (n = 0 to 3). These registers store the remaining transfer count during DMA transfer.

Since these registers are configured as 2-stage FIFO buffer registers consisting of the master register and slave register, a new DMA byte transfer count for DMA transfer can be specified during DMA transfer (see **19.8 Next Address Setting Function**).

These registers are decremented by 1 for each transfer, and transfer ends when a borrow occurs.

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

Reset input makes these registers undefined.

Remark If the DBCn register is read during DMA transfer after a terminal count has occurred without the register being overwritten, the value set immediately before the DMA transfer will be read out (0000H will not be read, even if DMA transfer has ended).

After reset: Undefined R/W Address: DBC0 FFFFF0C0H, DBC1 FFFFF0C2H, DBC2 FFFFF0C4H, DBC3 FFFFF0C6H

DBCn (n = 0 to 3)

15	14	13	12	11	10	9	8
BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8

7	6	5	4	3	2	1	0
BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0

BC15 to BC0	Transfer count setting (store remaining transfer count during DMA transfer)
0000H	Transfer count 1 or remaining transfer count
0001H	Transfer count 2 or remaining transfer count
:	:
FFFFH	Transfer count 65,536 (2^{16}) or remaining transfer count

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

The DADC0 to DADC3 registers are 16-bit registers that control the DMA transfer mode for DMA channel n (n = 0 to 3). These registers cannot be accessed during DMA operation.

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

Reset input clears these registers to 0000H.

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

2. Set the DADCn register when the target channels is in one of the following periods (the operation is not guaranteed if the register is set at any other time).

- **Period from system reset to the generation of the first DMA transfer request**
- **Period from completion of DMA transfer (after terminal count) to the generation of the next DMA transfer request**
- **Period from forced termination of DMA transfer (after the DCHCn.INITn bit was set to 1) to the generation of the next DMA transfer request**

After reset: 0000H R/W Address: DADC0 FFFFF0D0H, DADC1 FFFFF0D2H, DADC2 FFFFF0D4H, DADC3 FFFFF0D6H

DADCn (n = 0 to 3)

15	14	13	12	11	10	9	8
DS1	DS0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
SAD1	SAD0	DAD1	DAD0	TM1	TM0	TTYP	TDIR

DS1	DS0	Setting of transfer data size for DMA transfer
0	0	8 bits
0	1	16 bits
1	0	Setting prohibited
1	1	Setting prohibited

SAD1	SAD0	Setting of count direction of transfer source address for DMA channel n
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

DAD1	DAD0	Setting of count direction of transfer destination address for DMA channel n
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

TM1	TM0	Setting of transfer mode during DMA transfer
0	0	Single transfer mode
0	1	Single-step transfer mode
1	0	Setting prohibited
1	1	Block transfer mode

TTYP	Setting of DMA transfer type
0	2-cycle transfer
1	Flyby transfer

TDIR	Setting of transfer direction during transfer between I/O and memory
0	Memory → I/O (read)
1	I/O → memory (write)
This setting is valid only during flyby transfer and ignored during 2-cycle transfer.	

Caution Be sure to clear bits 13 to 8 to “0”. If they are set to 1, the operation is not guaranteed.

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

The DCHC0 to DCHC3 registers are 8-bit registers that control the DMA transfer operating mode for DMA channel n (n = 0 to 3).

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

Reset input clears these registers to 00H.

Cautions 1. If transfer has been completed with the MLEn bit set to 1 and if the next transfer request is made by DMA transfer (hardware DMA) that is started by $\overline{\text{DMARQn}}$ pin input or an interrupt from the on-chip peripheral I/O, the next transfer is executed with the TCn bit set to 1 (not automatically cleared to 0).

2. Set the MLEn bit when the target channel is in one of the following periods (the operation is not guaranteed if the bit is set at any other time).

- **Period from system reset to the generation of the first DMA transfer request**
- **Period from completion of DMA transfer (after terminal count) to the generation of the next DMA transfer request**
- **Period from forced termination of DMA transfer (after the INITn bit of the DCHCn register was set to 1) to the generation of the next DMA transfer request**

3. If DMA transfer is forcibly terminated in the last transfer cycle with the MLEn bit set to 1, the operation is performed in the same manner as when transfer is completed (the TCn bit is set to 1 and the $\overline{\text{TCn}}$ signal is output). (The Enn bit is cleared to 0 upon forced termination, regardless of the value of the MLEn bit.)
In this case, the Enn bit must be set to 1 and the TCn bit must be read (cleared to 0) when the next DMA transfer request is made.

4. Upon completion of DMA transfer (during terminal count), each bit is updated with the Enn bit cleared to 0 and then the TCn bit set to 1. If the statuses of the TCn bit and Enn bit are polled and if the DCHCn register is read while each bit is updated, therefore, a value indicating the status "transfer not completed and prohibited" (TCn bit = 0 and Enn bit = 0) may be read (this is not abnormal).

5. The TCn bit does not have to be read (cleared (0)) after DMA transfer has been completed (after terminal count) only if both of the following conditions are satisfied. If either of these conditions is not satisfied, be sure to read (clear (0)) the TCn bit before the next DMA transfer request is generated.

- **The MLEn bit is set (1) when DMA transfer is completed (at terminal count).**
- **The next DMA transfer start source is the $\overline{\text{DMARQn}}$ pin input or an interrupt request from the on-chip peripheral I/O (hardware DMA).**

If the above two conditions are not satisfied, the operation cannot be guaranteed if the next DMA transfer request is generated when the TCn bit is set (1).

After reset: 00H R/W Address: DCHC0 FFFFF0E0H, DCHC1 FFFFF0E2H, DCHC2 FFFFF0E4H, DCHC3 FFFFF0E6H

	<7>	6	5	4	<3>	<2>	<1>	<0>
DCHCn	TCn	0	0	0	MLEn	INITn	STGn	Enn

(n = 0 to 3)

TCn[Note 1]	Status bit that indicates whether DMA transfer via DMA channel n has ended or not
0	DMA transfer has not ended.
1	DMA transfer has ended.
This bit is set (1) at the last DMA transfer and cleared (0) when it is read.	

MLEn	When this bits is set (1) at DMA transfer completion (at the terminal count output), the Enn bit is not cleared (0) and the DMA transfer enabled state is retained. If the next DMA transfer start factor is the $\overline{\text{DMARQn}}$ pin input or the interrupt from an on-chip peripheral I/O (hardware DMA), the DMA transfer request is acknowledged even if the TCn bit is not read. If the next DMA transfer start factor is input by setting the STGn bit to 1 (software DMA), the DMA transfer request is acknowledged if the TCn bit is read and cleared (0). When this bit is cleared (0) at DMA transfer completion (at the terminal count output), the Enn bit is cleared (0) and the DMA transfer disabled state is entered. At the next DMA transfer request, the TCn bit must be read and the Enn bit must be set (1).

INITn[Note 2]	If this bit is set (1) during DMA transfer or after DMA is forcibly suspended by NMI input, DMA transfer is forcibly terminated.

STGn[Note 2]	If this bit is set (1) in the DMA transfer enabled state (TCn bit = 0, Enn bit = 1), DMA transfer is started.

Enn	Setting whether DMA transfer via DMA channel n is to be enabled or disabled
0	DMA transfer disabled
1	DMA transfer enabled
• This bit is cleared (0) when DMA transfer ends. It is also cleared (0) when DMA transfer is forcibly terminated by inputting NMI or setting (1) the INITn bit. • If the Enn bit is set (1), do not set it until DMA transfer has been completed the number of times set by the DBCn register or DMA transfer is forcibly terminated by the INITn bit.	

Notes 1. TCn bit is read-only.

2. INITn and STGn bits are write-only. If these bits are read, 0 is read.

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

19.3.6 DMA terminal count output control register (DTOC)

The DMA terminal count output control register (DTOC) is an 8-bit register that controls the terminal count output from each DMA channel and DMA transfer during NMI input. Terminal count signals from each DMA channel can be brought together and output from the $\overline{TC0}$ pin.

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

Reset input sets this register to 01H.

Caution To change the setting of the DTOC register, be sure to stop all the DMA operations.

After reset: 01H R/W Address: FFFFF8A0H

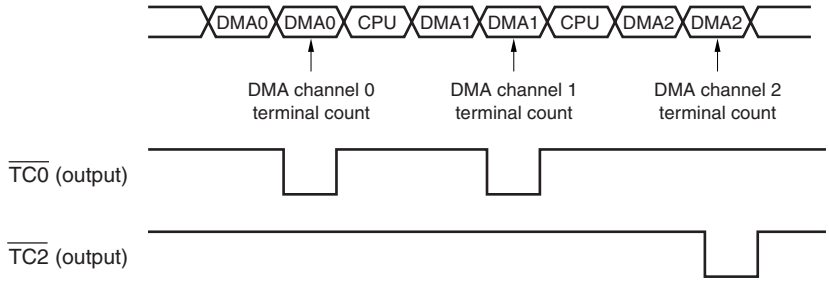
	7	6	5	4	3	2	1	0
DTOC	DMSTPM	0	0	0	TCO3	TCO2	TCO1	TCO0

DMSTPM	Control of DMA transfer when NMI is input
0	Forcibly abort DMA transfer when NMI is input.
1	Do not abort DMA transfer when NMI is input.

- When the DMSTPM bit = 0, processing of NMI can be executed immediately after the DMA cycle under execution has been completed. To execute DMA transfer that has been forcibly stopped, however, set the DCHCn.INITn bit (1) to forcibly terminate DMA transfer, re-initialize, and then execute it (n = 0 to 3).
- When the DMSTPM bit = 1, NMI processing is held pending in the block transfer mode until DMA transfer has been executed the number of times specified in advance. In the single transfer mode and single-step transfer mode, NMI processing is executed after completion of the DMA cycle under execution.
  To stop DMA, set the DCHCn.INITn bit (1) and forcibly terminate DMA transfer (n = 0 to 3).

TCO3-TCO0	$\overline{TC0}$ pin status flag
0	Channel n terminal count signal not output from $\overline{TC0}$ pin.
1	Channel n terminal count signal output from $\overline{TC0}$ pin.

The following shows an example of the case when the DTOC register is set to 03H.



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

The DTFR0 to DTFR3 registers are 8-bit registers that control the DMA transfer start trigger via interrupt requests from on-chip peripheral I/O.

The interrupt requests set by these registers serve as DMA transfer startup factors.

These registers can be read or written in 8-bit or 1-bit units. However, bit 7 (DFn) and bits 5 to 0 (IFCn5 to IFCn0) can be read or written in 8-bit units.

Reset input clears these registers to 00H.

Cautions 1. To change the setting of the DTFRn register, be sure to stop the DMA operation.

2. An interrupt request from an on-chip peripheral I/O input in the standby mode (IDLE or software STOP mode) is held pending as a DMA transfer start factor. The held DMA start factor is executed after restoring to the normal operation mode.

3. If the start source of DMA transfer has been changed by using the IFCn5 to IFCn0 bits, be sure to clear the DFn bit (0) with an instruction immediately after.

4. If the transmission enable interrupt request signal (INTSTn) of UARTA is used as the start source of DMA transfer, for example, the start source of DMA transfer that is started by the interrupt request signal generated when the last transfer data is transferred to the UARTAn transmit shift register is held. In this case, clear the DFn bit (0) and clear the DMA transfer start source.

After reset: 00H R/W Address: DTFR0 FFFFF810H, DTFR1 FFFFF812H, DTFR2 FFFFF814H, DTFR3 FFFFF816H

DTFRn (n = 0 to 3)

	<7>	6	5	4	3	2	1	0
DTFRn	DFn	0	IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0

DFn[Note]	DMA transfer request flag
0	DMA transfer not requested
1	DMA transfer requested

Note Only 0 can be written to this flag.

If the interrupt specified as the DMA transfer start factor occurs and it is necessary to clear the DMA transfer request while DMA transfer is disabled (including when it is suspended by NMI, and is forcibly terminated by software), stop the operation of the source causing the interrupt, and then clear the DFn bit (for example, disable reception in the case of serial reception). If it is clear that the interrupt will not occur until DMA transfer is resumed next, it is not necessary to stop the operation of the source causing the interrupt.

Cautions 1. For the IFCn5 to IFCn0 bits, see Table 19-1 DMA Start Factors.

2. Be sure to set bit 6 to 0. The operation is not guaranteed when it is set to 1.

Table 19-1. DMA Start Factor (1/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt Source
0	0	0	0	0	0	DMA request from on-chip peripheral I/O disabled
0	0	0	0	0	1	INTP000/INTCCP00
0	0	0	0	1	0	INTP001/INTCCP01
0	0	0	0	1	1	INTP004/INTCC101
0	0	0	1	0	0	INTP005/INTCC100
0	0	0	1	0	1	INTP106
0	0	0	1	1	0	INTP107
0	0	0	1	1	1	INTP010/INTCCQ0
0	0	1	0	0	0	INTP011/INTCCQ1
0	0	1	0	0	1	INTP012/INTCCQ2
0	0	1	0	1	0	INTP013/INTCCQ3
0	0	1	0	1	1	INTP114
0	0	1	1	0	0	INTP115
0	0	1	1	0	1	INTP021/INTCCP10
0	0	1	1	1	0	INTP022/INTCCP11
0	0	1	1	1	1	INTP124
0	1	0	0	0	0	INTP125
0	1	0	0	0	1	INTP126
0	1	0	0	1	0	INTP130
0	1	0	0	1	1	INTP131
0	1	0	1	0	0	INTP132
0	1	0	1	0	1	INTP133
0	1	0	1	1	0	INTP134
0	1	0	1	1	1	INTP137
0	1	1	0	0	0	INTP050/INTCCP20
0	1	1	0	0	1	INTP051/INTCCP21
0	1	1	0	1	0	INTCMD0
0	1	1	0	1	1	INTCMD1
0	1	1	1	0	0	INTCMD2
0	1	1	1	0	1	INTCMD3
0	1	1	1	1	0	INTCM100
0	1	1	1	1	1	INTCM101
1	0	0	0	0	0	INTOVQ
1	0	0	0	0	1	INTSER0/INTCSIER0
1	0	0	0	1	0	INTSR0/INTCSIR0
1	0	0	0	1	1	INTST0/INTCSIT0
1	0	0	1	0	0	INTSER1/INTCSIER1
1	0	0	1	0	1	INTSR1/INTCSIR1
1	0	0	1	1	0	INTST1/INTCSIT1

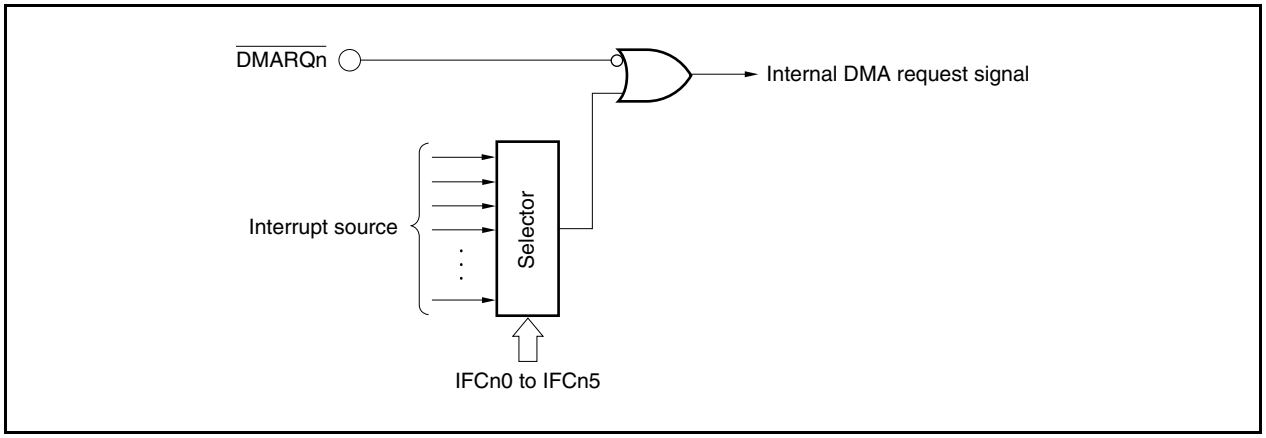
Remark n = 0 to 3

Table 19-1. DMA Start Factor (2/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt Source
1	0	0	1	1	1	INTSER2/INTCSIER2
1	0	1	0	0	0	INTSR2/INTCSIR2
1	0	1	0	0	1	INTST2/INTCSIT2
1	0	1	0	1	0	INTSER3
1	0	1	0	1	1	INTSR3/INTIIC
1	0	1	1	0	0	INTST3
1	0	1	1	0	1	INTAD
Other than above						Setting prohibited

Remark n = 0 to 3

The relationship between the $\overline{\text{DMARQn}}$ signal and the interrupt source that serves as a DMA transfer trigger is as follows (n = 0 to 3).



Cautions 1. If a $\overline{\text{DMARQn}}$ pin is specified as the DMA transfer start factor, clear the DTFRn register to 00H. If an interrupt request is specified as the DMA transfer start factor, mask the $\overline{\text{DMARQn}}$ signal input on the port side (PMC0 register, etc.). In this case, an interrupt request will be generated with the start of DMA transfer. To prevent an interrupt request from being generated, mask the interrupt by setting the interrupt request control register. DMA transfer starts even if an interrupt is masked.

2. The start request of DMA transfer by the $\overline{\text{DMARQn}}$ signal input in the standby mode (IDLE or software STOP mode) does not serve as the start source of DMA transfer.

3. If the frequency of the CPU clock falls below the clock of each on-chip peripheral I/O because of the setting of the PCC.CK1 and PCC.CK0 bits, the start source of DMA transfer may not be acknowledged.

19.3.8 DMA interface control register (DIFC)

The DIFC register is an 8-bit register that of each DMA channel and controls the mask function (DMA mask mode) of the $\overline{\text{DMARQn}}$ signal (n = 0 to 3).

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF8A8H

	7	6	5	4	3	2	1	0
DIFC	0	0	0	0	DRMK3	DRMK2	DRMK1	DRMK0

DRMKn	Specification of DMA mask mode (mask bit of $\overline{\text{DMARQn}}$ signal) (n = 0-3)
0	Do not mask the $\overline{\text{DMARQn}}$ signal.
1	Mask input of the $\overline{\text{DMARQn}}$ signal for the duration of 3 bus clocks from the rise of the $\overline{\text{DMAAKn}}$ signal.

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

<R> **2. The mask function of the $\overline{\text{DMARQn}}$ signal (DMA mask mode) is to mask only the $\overline{\text{DMARQn}}$ signal being serviced so that the $\overline{\text{DMARQn}}$ signal is not inadvertently generated two times. After the $\overline{\text{DMAAKn}}$ signal of the same channel as the $\overline{\text{DMARQn}}$ signal is asserted, the $\overline{\text{DMARQn}}$ signal is masked for a mask period. Masking of the other channels is not affected.**

If only the $\overline{\text{DMARQn}}$ signal of a high priority is masked and if a contention with other $\overline{\text{DMARQn}}$ signal occurs, the other $\overline{\text{DMARQn}}$ signal is not acknowledged first during the mask period, regardless of the priority of the $\overline{\text{DMARQn}}$ signal.

19.3.9 DMAAK width control register (DAKW)

This 8-bit register controls the active width of the $\overline{\text{DMAAK}}$n signal of each DMA channel (n = 0 to 3).

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF8ACH

	7	6	5	4	3	2	1	0
DAKW	0	0	0	0	0	DAKW2	DAKW1	DAKW0

DAKW2	DAKW1	DAKW0	Specification of active extension width of $\overline{\text{DMAAK}}$n signal (n = 0 to 3)
0	0	0	0CPU clock
0	0	1	1CPU clock
0	1	0	2CPU clock
0	1	1	3CPU clock
1	0	0	4CPU clock
1	0	1	5CPU clock
1	1	0	6CPU clock
1	1	1	7CPU clock

- The active width of the $\overline{\text{DMAAK}}$n signal is extended in synchronization with the CPU clock (f_{CPU}).
- Set the active width of the $\overline{\text{DMAAK}}$n signal at least one CPU clock width wider than the period of BUSCLK.

Cautions 1. During flyby transfer, the active width of the $\overline{\text{DMAAK}}$n signal is not extended regardless of the set value of the DAKW register.

2. The set value of the DAKW register is reflected on all of signals $\overline{\text{DMAAK}}$0 to $\overline{\text{DMAAK}}$3.

3. If one or both of the following conditions is satisfied when the active width of the $\overline{\text{DMAAK}}$n signal is extended, the next $\overline{\text{DMAAK}}$n signal is asserted while an active extension width of the $\overline{\text{DMAAK}}$n signal is appended. As a result, the active cycle of two or more $\overline{\text{DMAAK}}$n signals is used as one $\overline{\text{DMAAK}}$n signal. Consequently, the number of times the $\overline{\text{DMAAK}}$n signal is asserted may be less than the actual number of DMA cycles.

- **During block transfer or single-step transfer**
- **If the next $\overline{\text{DMARQ}}$n signal is asserted before the first $\overline{\text{DMAAK}}$n signal is deasserted during single transfer**

4. Be sure to clear bits 7 to 3 to "0". The operation cannot be guaranteed if these bits are set to 1.

19.4 Transfer Modes

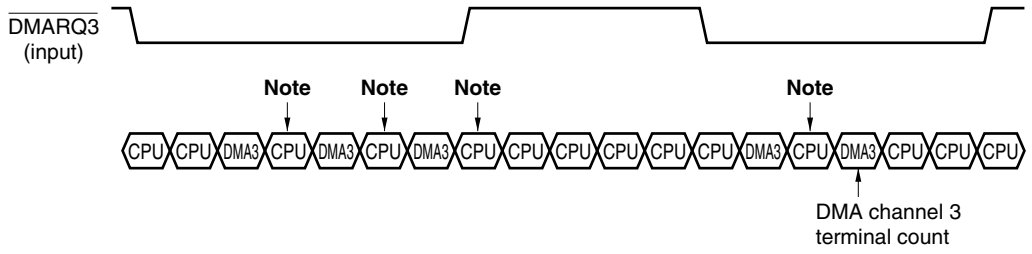
19.4.1 Single transfer mode

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

When the DMAC has released the bus, if another higher priority DMA transfer request is issued, the higher priority DMA request always takes precedence. If another DMA transfer request with a lower priority occurs one clock after single transfer has been completed, however, this request does not take precedence even if the previous DMA transfer request signal with a higher priority remains active. DMA transfer with the newly requested lower priority request is executed after the CPU bus has been released.

Figures 19-1 to 19-4 show examples of single transfer.

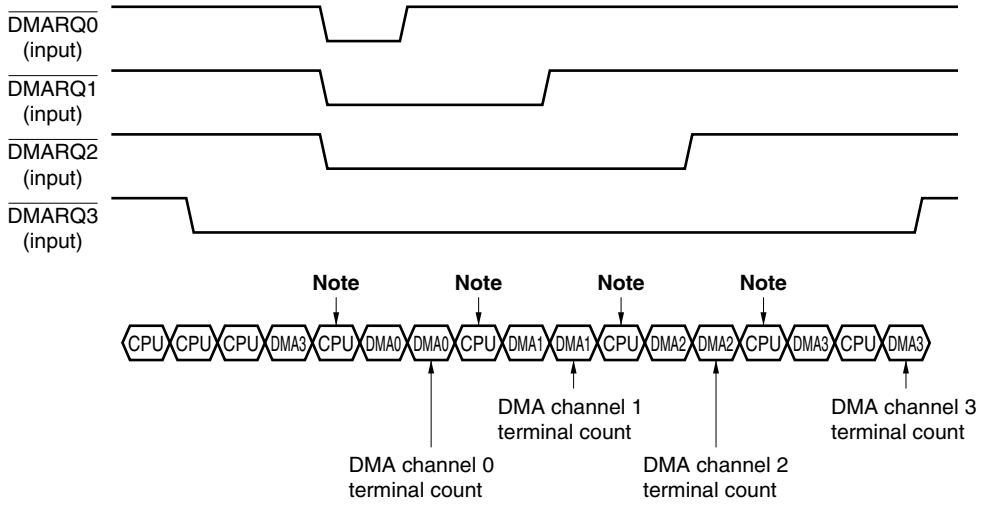
Figure 19-1. Single Transfer Example 1



Note The bus is always released.

Figure 19-2 shows an example of a single transfer in which a higher priority DMA request is issued. DMA channels 0 to 2 are in the block transfer mode and channel 3 is in the single transfer mode.

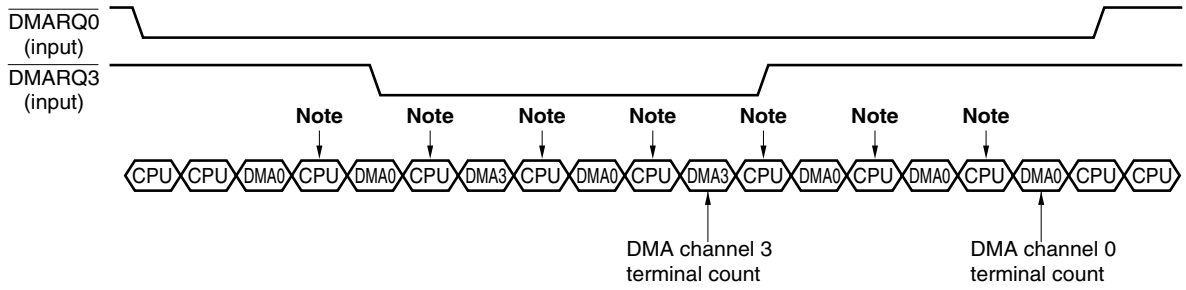
Figure 19-2. Single Transfer Example 2



Note The bus is always released.

Figure 19-3 is an example of single transfer where a DMA transfer request with a lower priority is issued one clock after single transfer has been completed. DMA channels 0 and 3 are used for single transfer. If two DMA transfer request signals become active at the same time, two DMA transfer operations are alternately executed.

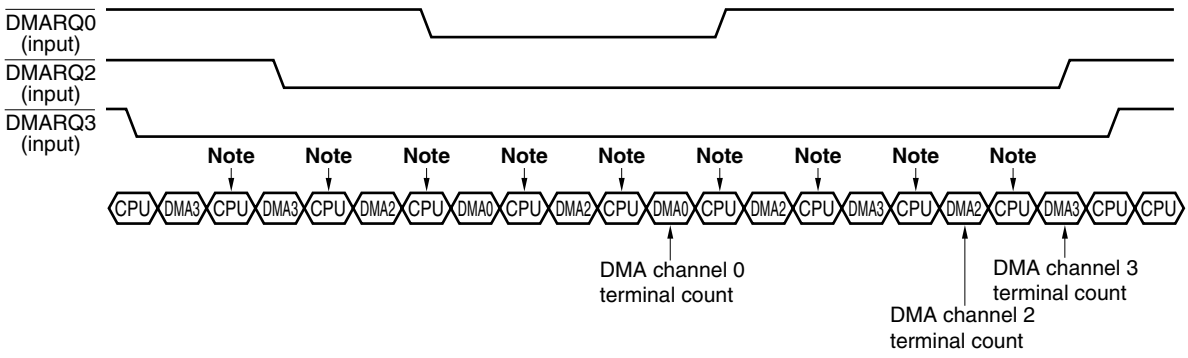
Figure 19-3. Single Transfer Example 3



Note The bus is always released.

Figure 19-4 is an example of single transfer where two or more DMA transfer requests with a lower priority are issued one clock after single transfer has been completed. DMA channels 0, 2, and 3 are used for single transfer. If three or more DMA transfer request signals become active at the same time, two DMA transfer operations are alternately executed, starting from the one with the highest priority.

Figure 19-4. Single Transfer Example 4



Note The bus is always released.

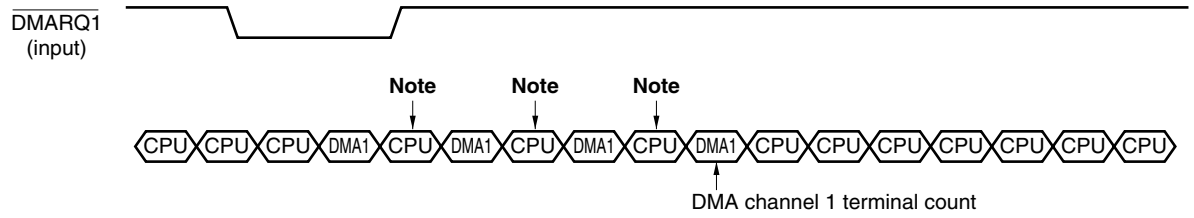
19.4.2 Single-step transfer mode

In single-step transfer mode, the DMAC releases the bus at each byte/halfword transfer. If there is a subsequent DMA transfer request signal ($\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$), transfer is performed again. This operation continues until a terminal count occurs.

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

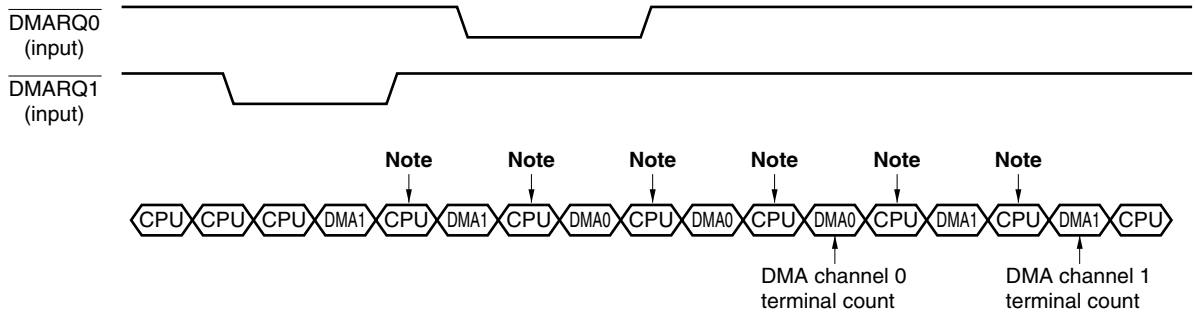
The following shows an example of a single-step transfer. Figure 19-6 shows an example of single-step transfer made in which a higher priority DMA request is issued. DMA channels 0 and 1 are in the single-step transfer mode.

Figure 19-5. Single-Step Transfer Example 1



Note The bus is always released.

Figure 19-6. Single-Step Transfer Example 2



Note The bus is always released.

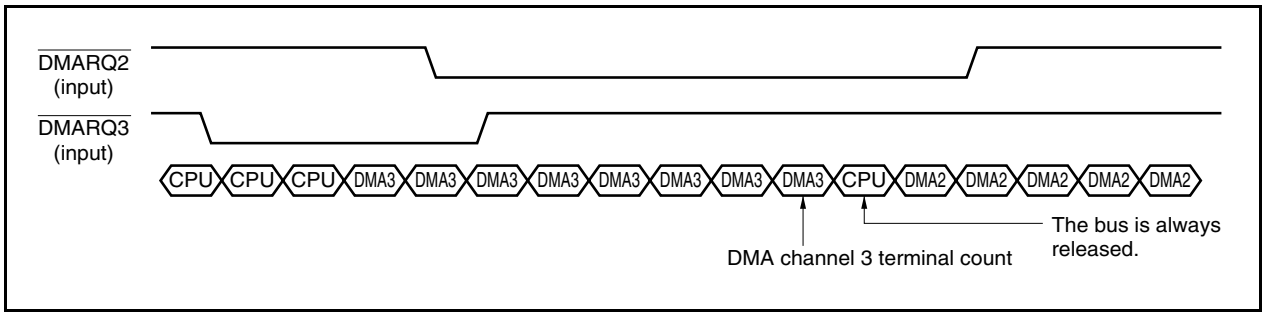
19.4.3 Block transfer mode

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

After the block transfer ends and the DMAC releases the bus, another DMA transfer can be acknowledged. The bus cycle of the CPU is not inserted during block transfer, but bus hold and refresh cycles are inserted between DMA transfer operations.

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

Figure 19-7. Block Transfer Example



19.5 Transfer Types

19.5.1 2-cycle transfer

In 2-cycle transfer, data transfer is performed in two cycles, a read cycle (source to DMAC) and a write cycle (DMAC to destination).

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

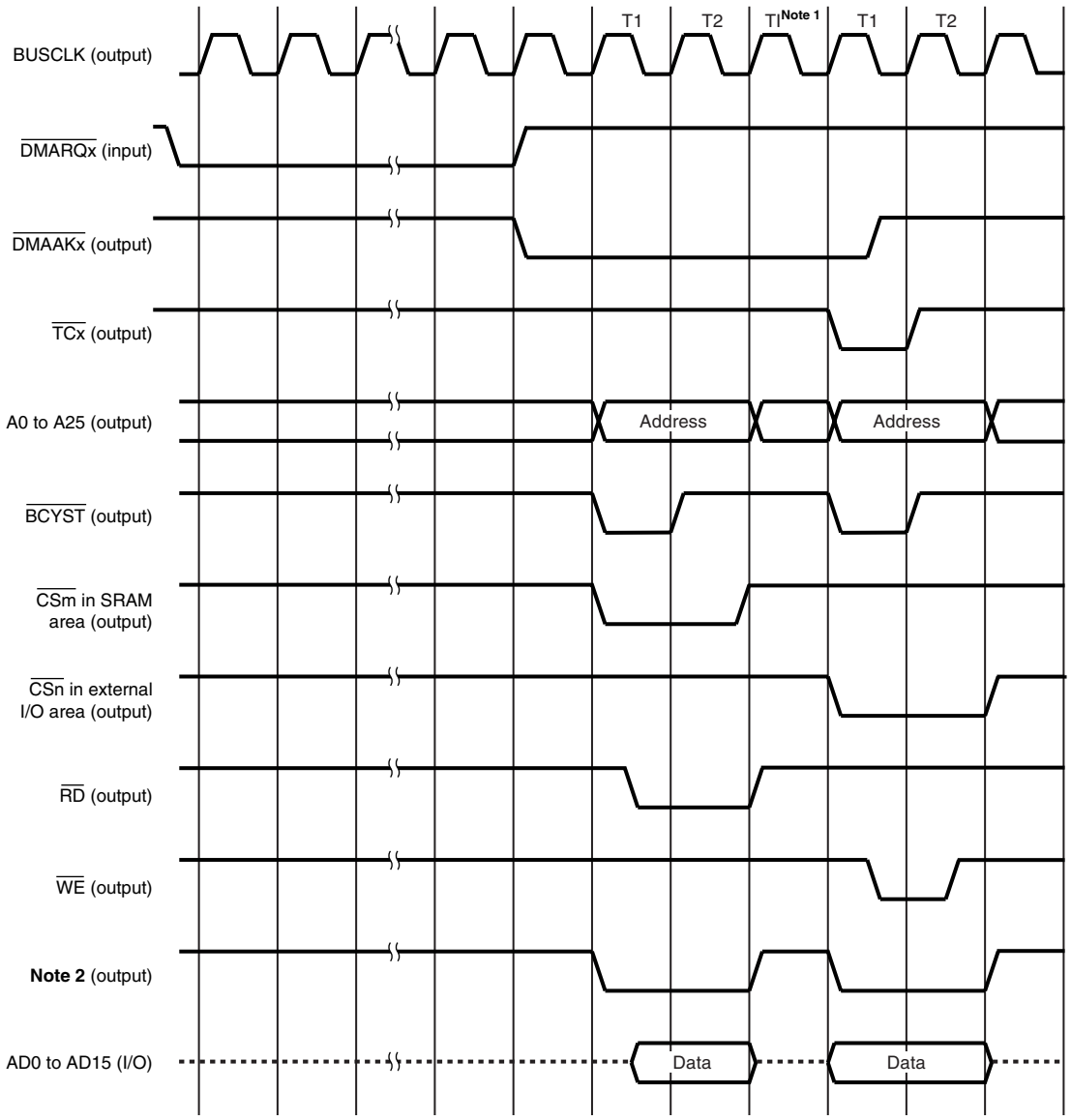
Cautions 1. An idle cycle of 1 to 2 clocks is always inserted between a read cycle and a write cycle.

2. For 2-cycle transfer from the on-chip peripheral I/O to the internal RAM, refer to Caution 4 of Table 19-3 Relationship Between Transfer Type and Transfer Object.

The DMA access timing when no data exists in the write buffer is shown in **Figures 19-8** to **19-11**. Refer to **5.6 Write Buffer Function** for the write buffer.

Figure 19-8. SRAM, External ROM, and External I/O Access Timing During 2-Cycle DMA Transfer (SRAM → External I/O): BMC Register = 01H (1/2)

(a) BCC register setting for SRAM: BCm1, BCm0 bits = 00, BCC register setting for external I/O: BCn1, BCn0 bits = 00



Notes 1. Idle state (TI) independent of the BCC register settings.

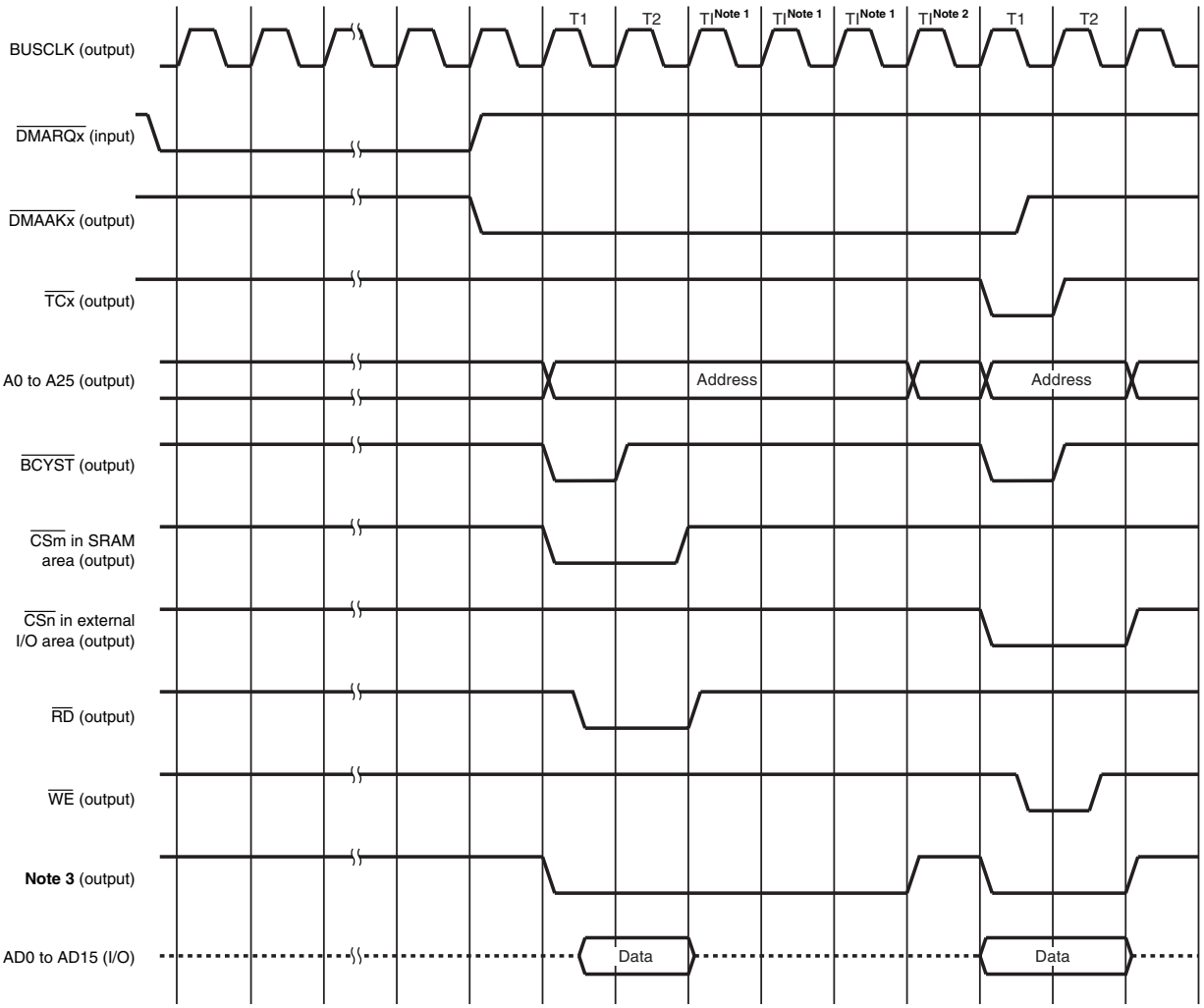
2. UBE, LBE

Remarks 1. The broken lines indicate the high-impedance state.

2. n = 0 to 7, m = 0 to 7 (n ≠ m)
x = 0 to 3

Figure 19-8. SRAM, External ROM, and External I/O Access Timing During 2-Cycle DMA Transfer (SRAM → External I/O): BMC Register = 01H (2/2)

(b) BCC register setting for SRAM: BCm1, BCm0 bits = 11, BCC register setting for external I/O: BCn1, BCn0 bits = 00



Notes 1. Idle state (TI) inserted depending on the BCC register settings.

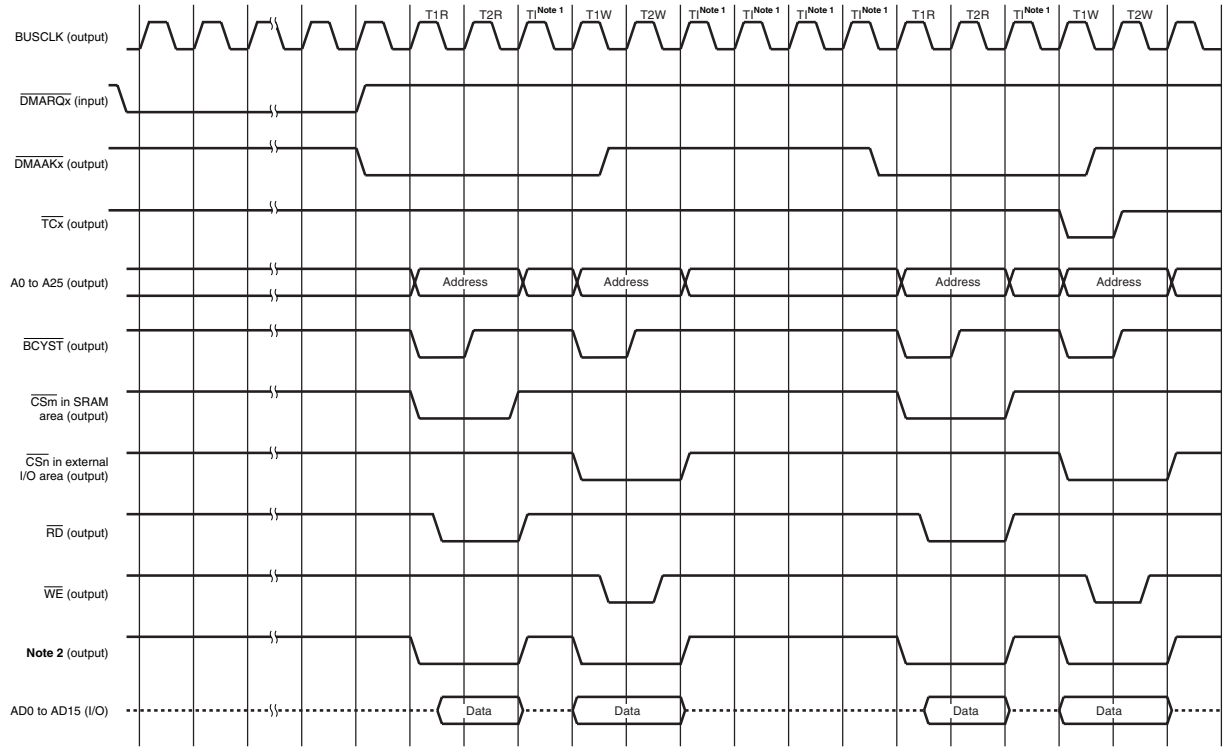
2. Idle state (TI) independent of the BCC register settings.

3. $\overline{UBE}$, $\overline{LBE}$

Remarks 1. The broken lines indicate the high-impedance state.

2. n = 0 to 7, m = 0 to 7 (n ≠ m)
x = 0 to 3

Figure 19-9. Timing During 2-Cycle DMA Transfer (SRAM → External I/O): In Single-Step Transfer Mode (BCC Register Setting for SRAM: BCm1, BCm0 Bits = 00, BCC Register Setting for External I/O: BCn1, BCn0 Bits = 00)



Notes 1. Idle state (TI) independent of the BCC register settings.

2. $\overline{UBE}$, $\overline{LBE}$

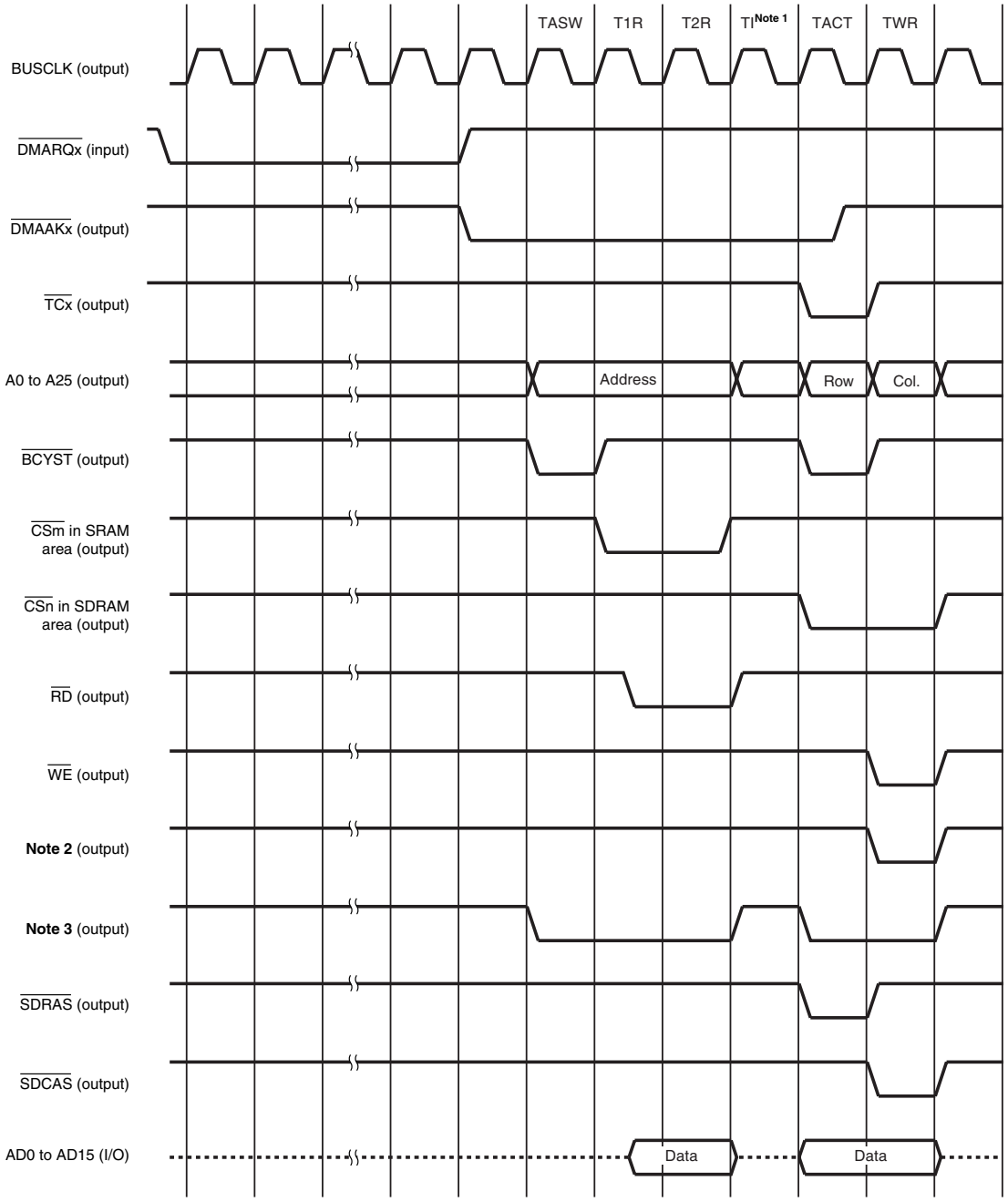
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 0 to 7, m = 0 to 7 (n ≠ m)

x = 0 to 3

Figure 19-10. Timing During 2-Cycle DMA Transfer (SRAM → SDRAM): BMC Register = 01H (1/3)

(a) In single transfer mode (CSDC register setting for SRAM: CSDCm bit = 1, ASC register setting for SRAM: ACm1, ACm0 bits = 01)



Notes 1. Idle state (TI) independent of the BCC register settings.

2. $\overline{\text{UWE}}$/UDQM, $\overline{\text{LWE}}$/LDQM

3. $\overline{\text{UBE}}$/UDQM, $\overline{\text{LBE}}$/LDQM

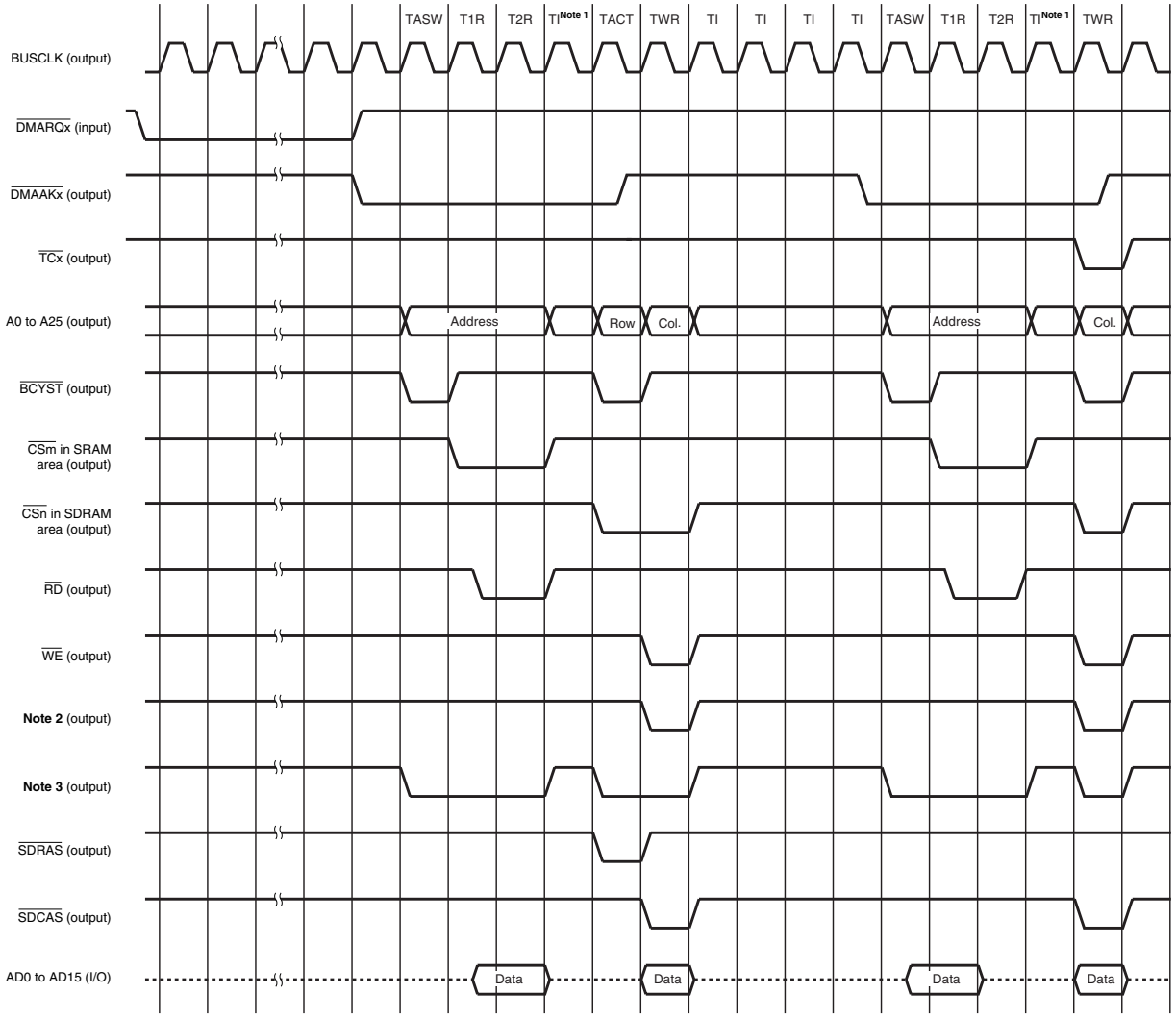
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, 6, m = 0, 4, 6, 7 (n ≠ m), x = 0 to 3

3. Col.: Column address
Row: Row address

Figure 19-10. Timing During 2-Cycle DMA Transfer (SRAM → SDRAM): BMC Register = 01H (2/3)

(b) In single-step transfer mode (CSDC register setting for SRAM: CSDCm bit = 1, ASC register setting for SRAM: ACm1, ACm0 bits = 01)



Notes 1. Idle state (TI) independent of the BCC register settings.

2. $\overline{\text{UWE}}$/UDQM, $\overline{\text{LWE}}$/LDQM

3. $\overline{\text{UBE}}$/UDQM, $\overline{\text{LBE}}$/LDQM

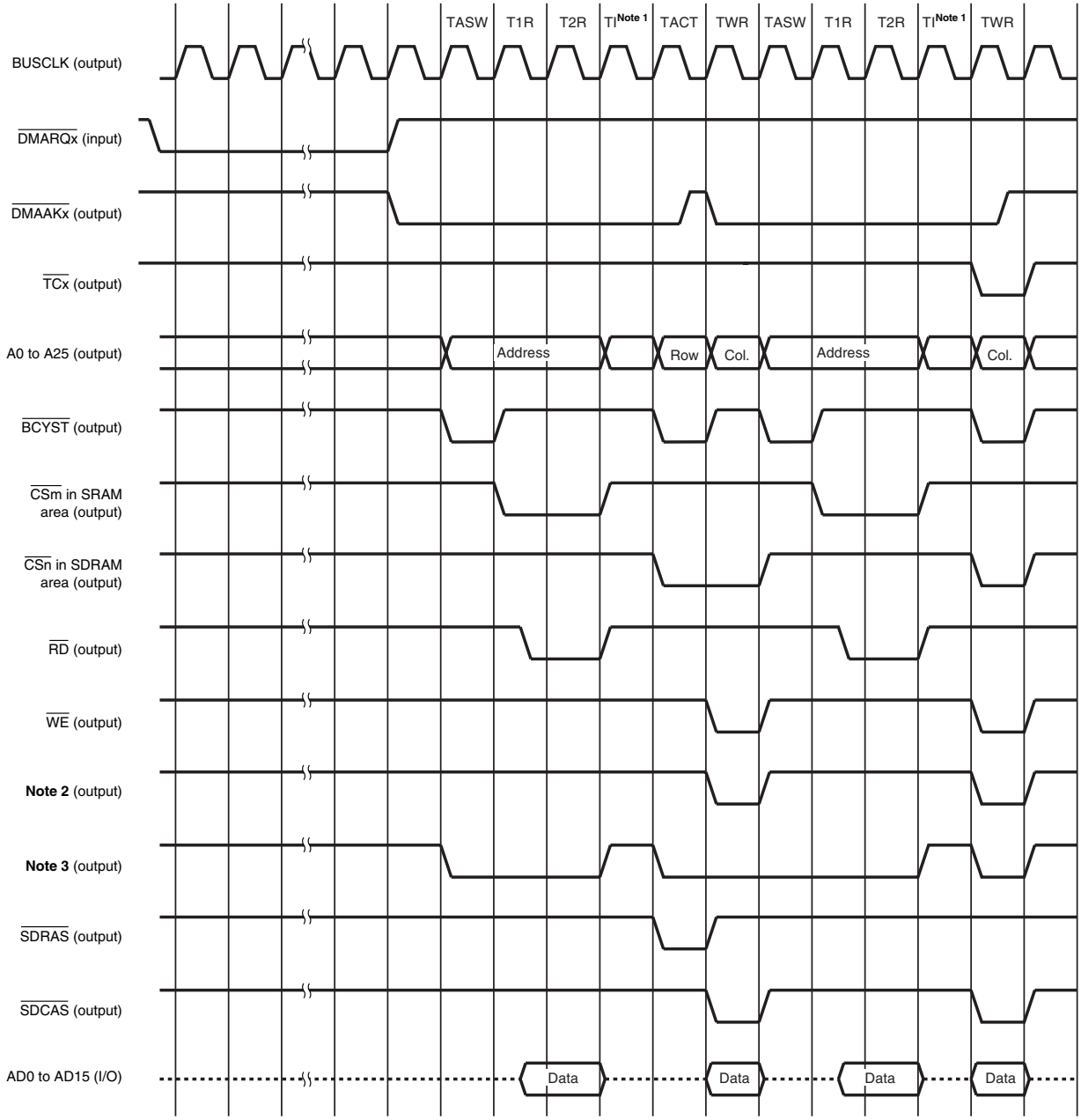
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, 6, m = 0, 4, 6, 7 (n ≠ m), x = 0 to 3

3. Col.: Column address
Row: Row address

Figure 19-10. Timing During 2-Cycle DMA Transfer (SRAM → SDRAM): BMC Register = 01H (3/3)

(c) In block transfer mode (CSDC register setting for SRAM: CSDCm bit = 1, ASC register setting for SRAM: ACm1, ACm0 bits = 01)



Notes 1. Idle state (TI) independent of the BCC register settings.

2. $\overline{\text{UWE}}$/UDQM, $\overline{\text{LWE}}$/LDQM

3. $\overline{\text{UBE}}$/UDQM, $\overline{\text{LBE}}$/LDQM

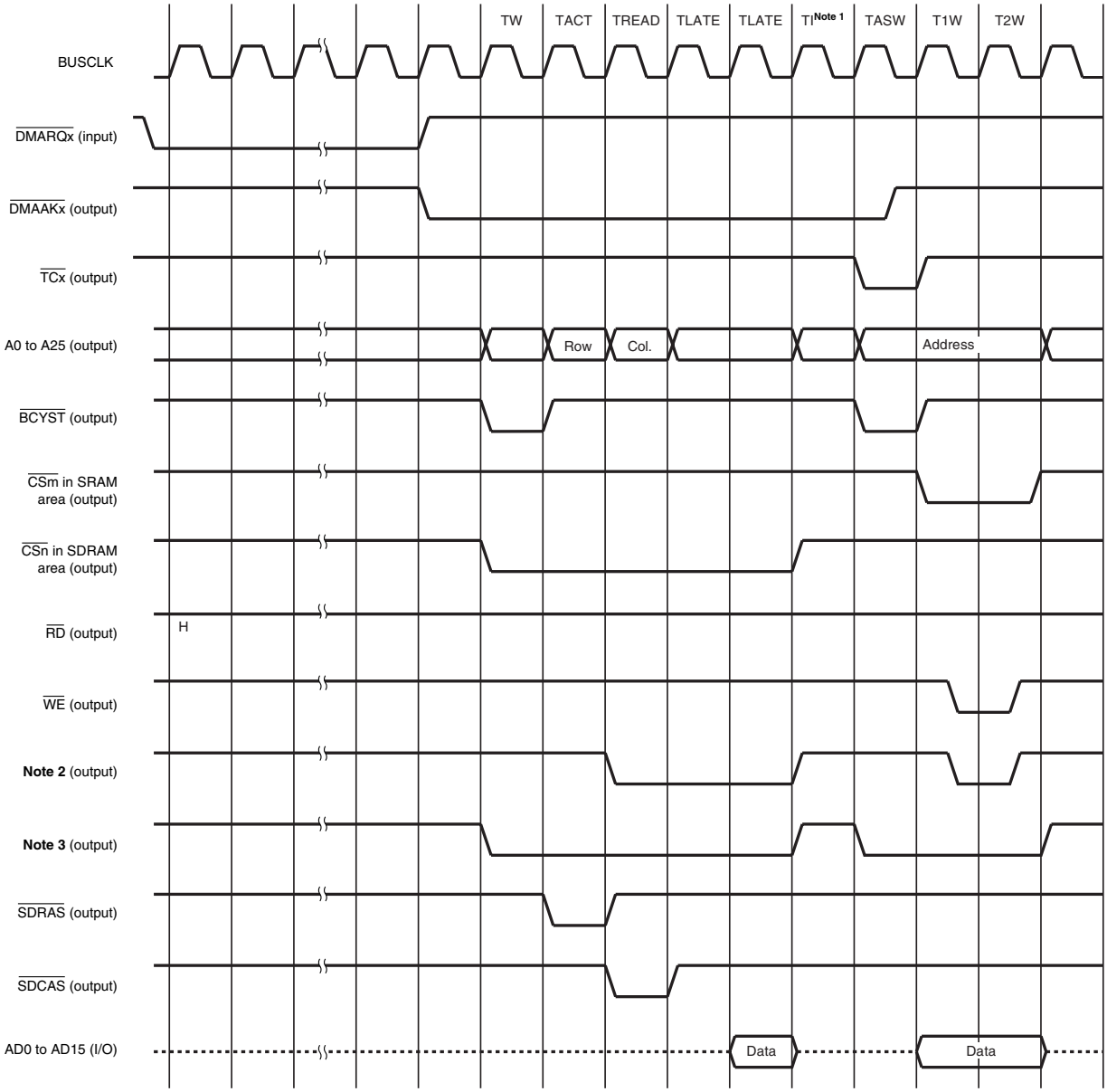
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, 6, m = 0, 4, 6, 7 (n ≠ m), x = 0 to 3

3. Col.: Column address
Row: Row address

Figure 19-11. Timing During 2-Cycle DMA Transfer (SDRAM → SRAM): BMC Register = 01H (1/3)

(a) In single transfer mode (CSDC register setting for SRAM: CSDCm bit = 1, ASC register setting for SRAM: ACm1, ACm0 bits = 01) (SCRn register setting for SDRAM: Latency = 2, BCW = 2)



Notes 1. Idle state (TI) independent of the BCC register settings.

2. $\overline{\text{UWE}}$/UDQM, $\overline{\text{LWE}}$/LDQM

3. $\overline{\text{UBE}}$/UDQM, $\overline{\text{LBE}}$/LDQM

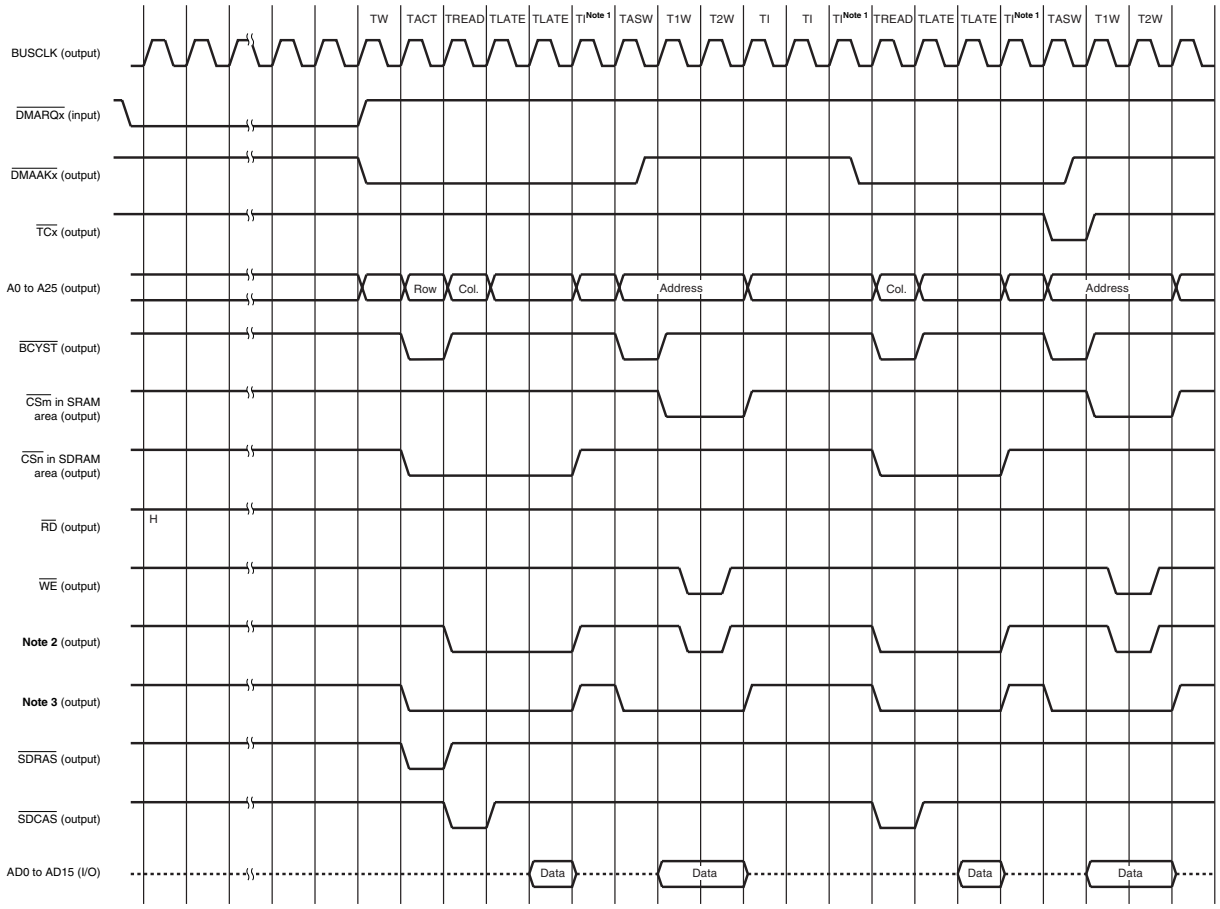
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, 6, m = 0, 4, 6, 7 (n ≠ m), x = 0 to 3

3. Col.: Column address
Row: Row address

Figure 19-11. Timing During 2-Cycle DMA Transfer (SDRAM → SRAM): BMC Register = 01H (2/3)

(b) In single-step transfer mode (CSDC register setting for SRAM: CSDCm bit = 1,
ASC register setting for SRAM: ACm1, ACm0 bits = 01)
(SCRn register setting for SDRAM: Latency = 2, BCW = 2)



Notes 1. Idle state (TI) independent of the BCC register settings.

2. $\overline{\text{UWE}}$/UDQM, $\overline{\text{LWE}}$/LDQM

3. $\overline{\text{UBE}}$/UDQM, $\overline{\text{LBE}}$/LDQM

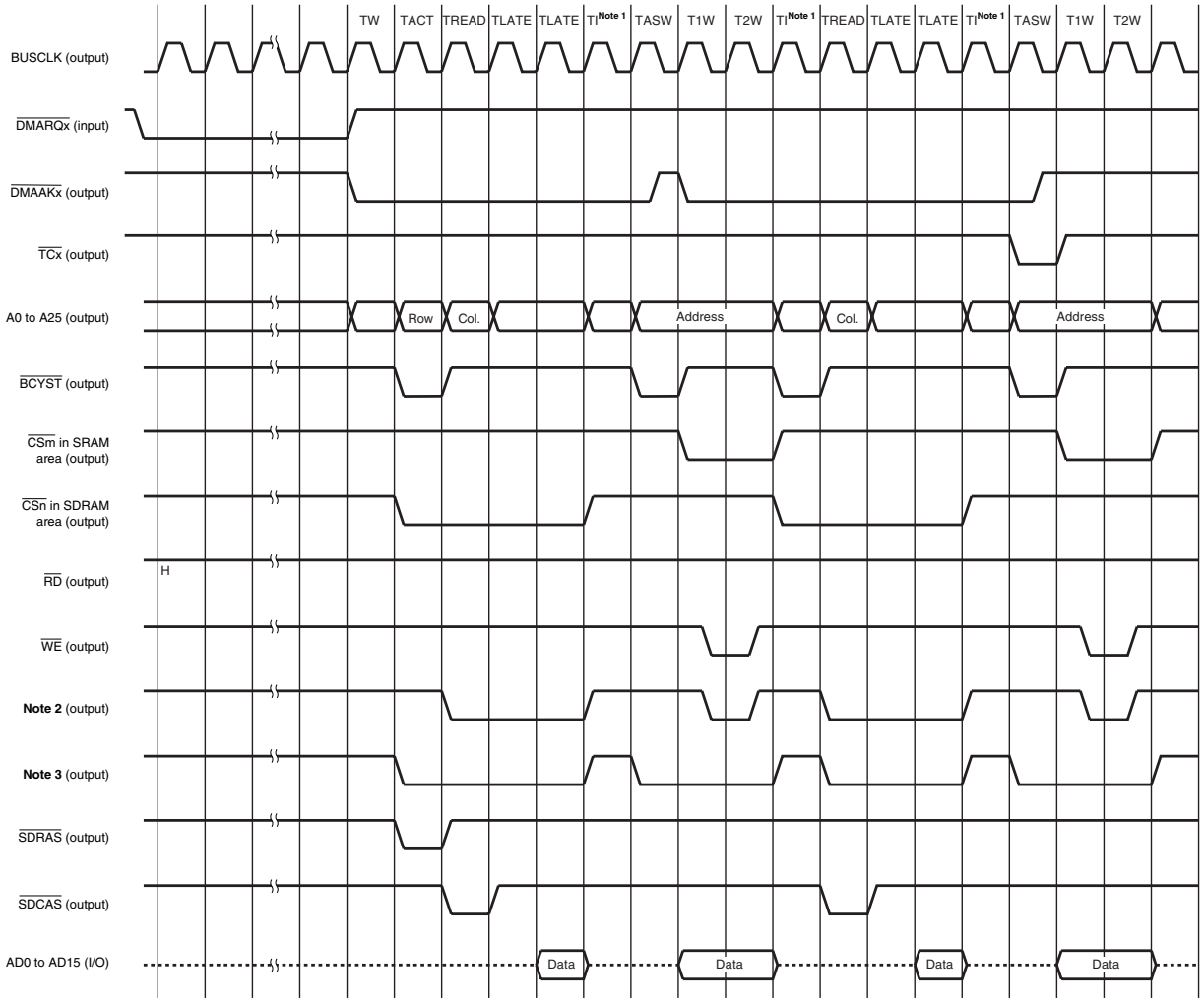
Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, 6, m = 0, 4, 6, 7 (n ≠ m), x = 0 to 3

3. Col.: Column address
Row: Row address

Figure 19-11. Timing During 2-Cycle DMA Transfer (SDRAM → SRAM): BMC Register = 01H (3/3)

(c) In block transfer mode (CSDC register setting to SRAM: CSDCm bit = 1,
ASC register setting for SRAM: ACm1, ACm0 bits = 01)
(SCRn register setting to SDRAM: Latency = 2, BCW = 2)



Notes 1. Idle state (TI) independent of the BCC register settings.

2. $\overline{\text{UWE}}$/UDQM, $\overline{\text{LWE}}$/LDQM

3. $\overline{\text{UBE}}$/UDQM, $\overline{\text{LBE}}$/LDQM

Remarks 1. The broken lines indicate the high-impedance state.

2. n = 1, 3, 4, 6, m = 0, 4, 6, 7 (n ≠ m), x = 0 to 3

3. Col.: Column address
Row: Row address

(1) Active width of $\overline{\text{DMAAK}}$n signal

The active width of the $\overline{\text{DMAAK}}$n signal output from the DMA controller is from the start of the read operation to the end of the write operation of the DMA controller (n = 0 to 3). However, the $\overline{\text{DMAAK}}$n signal is not asserted in cycles to write or read the internal RAM. When the external memory is written, asserting the $\overline{\text{DMAAK}}$n signal ends as soon as data has been transferred to the write buffer. Therefore, handshaking between the $\overline{\text{DMARQ}}$n and $\overline{\text{DMAAK}}$n signals must be completed before the external I/O or external memory is written. If data of three buffers has already been stored in the write buffer, for example, the data transferred by DMA is stored in the fourth stage of the buffer and asserting the $\overline{\text{DMAAK}}$n signal ends. After a write operation of the stored data has been executed three times, transfer to the external I/O or external memory that is controlled by the $\overline{\text{DMAAK}}$n signal whose assertion has ended is executed.

(2) $\overline{\text{DMAAK}}$n signal active width extension function

The $\overline{\text{DMAAK}}$n signal is output in synchronization with the internal bus cycle during 2-cycle transfer, and is not synchronized with the external bus cycle (n = 0 to 3).

Depending on the target of DMA transfer, the configuration may allow the $\overline{\text{DMAAK}}$n signal to be asserted only for the duration of two internal system clocks (f_{CLK}). In this case, assertion of the $\overline{\text{DMAAK}}$n signal may not be sampled with BUSCLK if the internal system clock is divided and the bus clock (BUSCLK) is used (e.g., BMC register = 02H: internal system clock divided by three).

To sample assertion of the $\overline{\text{DMAAK}}$n signal with BUSCLK, extend the active width of the $\overline{\text{DMAAK}}$n signal by setting the DAKW register.

The minimum value of the active width of the $\overline{\text{DMAAK}}$n during 2-cycle transfer is shown in the table below.

Table 19-2. Minimum Value of Active Width of $\overline{\text{DMAAK}}$n Signal During 2-Cycle Transfer

<R>

		Transfer Destination		
		External I/O/External Memory	On-Chip Peripheral I/O	Internal RAM
Transfer source	External I/O/external memory	Read cycle + 4 internal system clocks	Read cycle + (5+i) internal system clocks	Read cycle + 1 internal system clock
	On-chip peripheral I/O	(6+i) internal system clocks	(7+2i) internal system clocks	(3+i) internal system clocks
	Internal RAM	2 internal system clocks	(3+i) internal system clocks	–

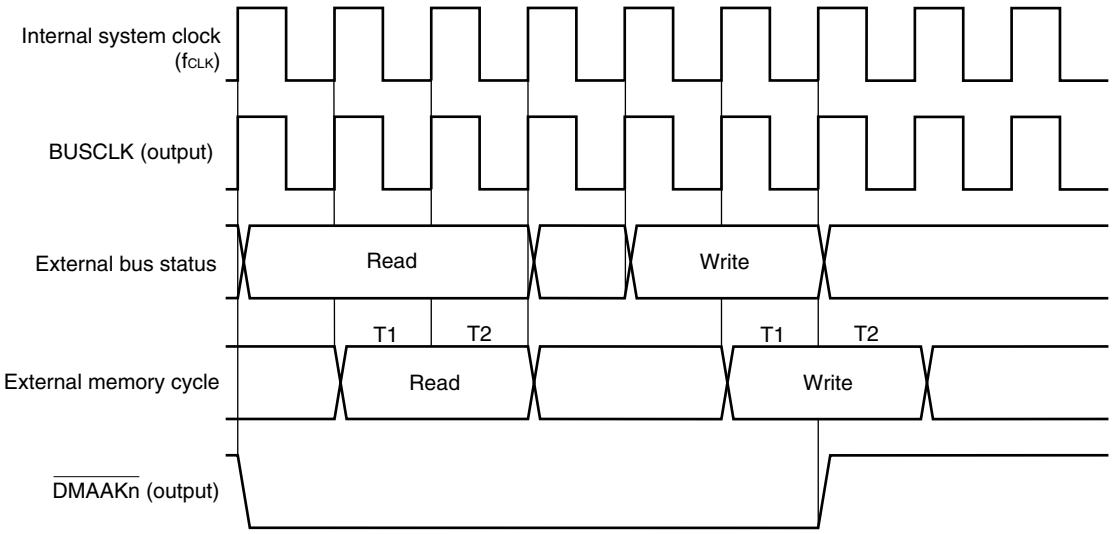
Caution The function to extend the active width of the $\overline{\text{DMAAK}}$n signal can be used only during 2-cycle transfer (n = 0 to 3). The active width of the $\overline{\text{DMAAK}}$n signal is not extended regardless of the DAKW register. The operation is not guaranteed if it is used during flyby transfer. During flyby transfer, the $\overline{\text{DMAAK}}$n signal synchronized with the bus cycle is output.

Remark i: Number of wait cycles set by VSWC register

(3) Outline of 2-cycle transfer timing

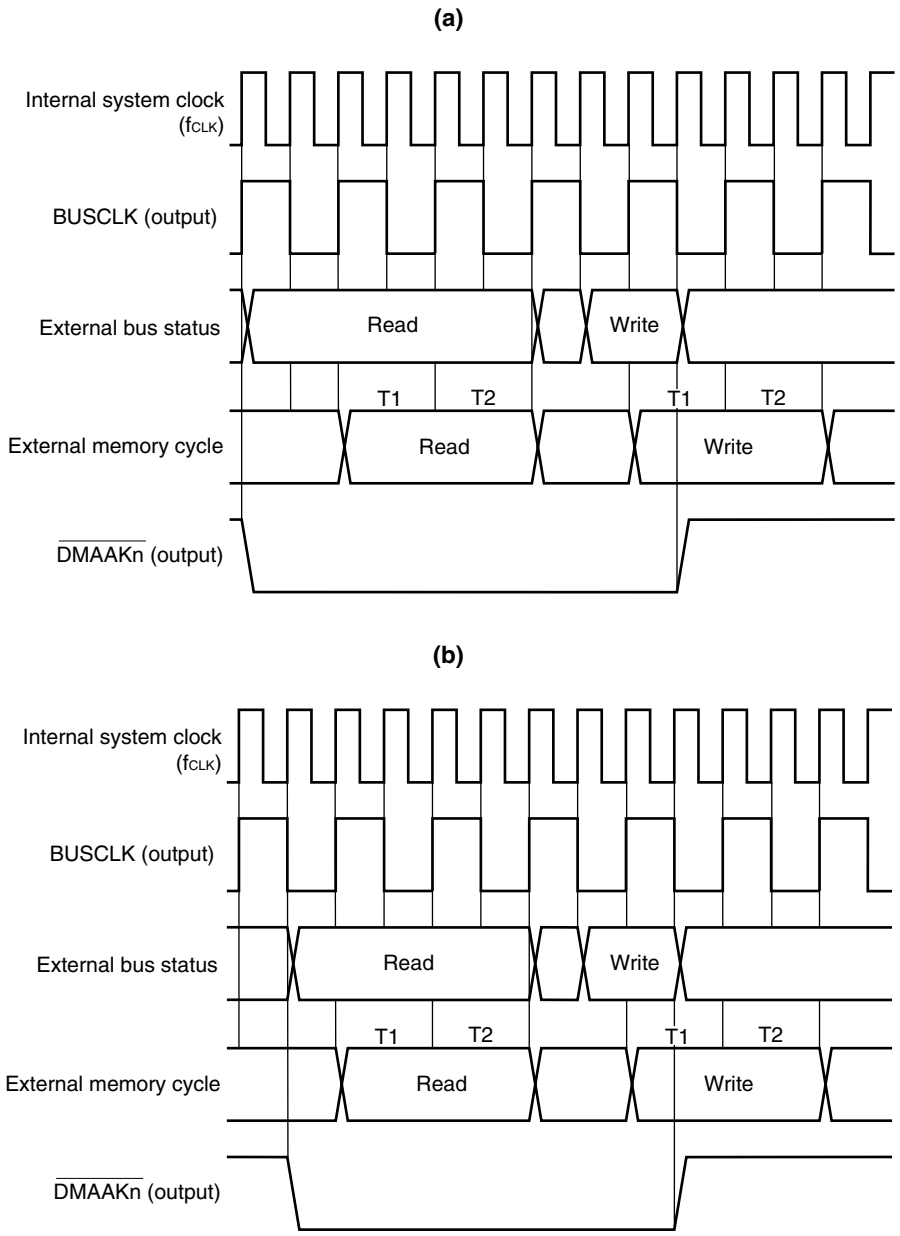
The timing of 2-cycle transfer when no data exists in the write buffer is outlined below. Refer to **5.6 Write Buffer Function** for the write buffer.

Figure 19-12. Outline of Timing During 2-Cycle Transfer (SRAM → SRAM): Divided by 1 (0 SRAM Waits)



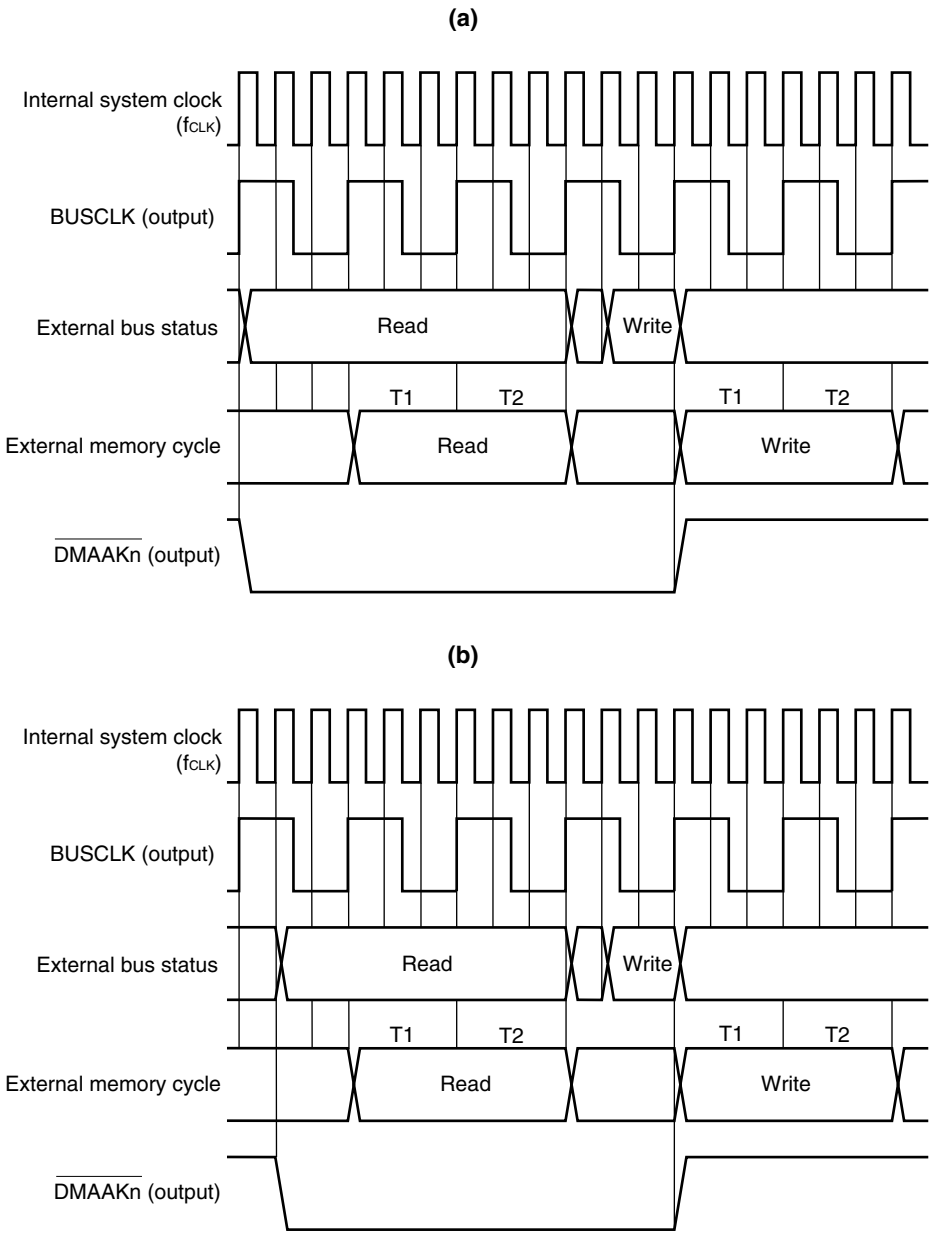
Remark n = 0 to 3

Figure 19-13. Outline of Timing During 2-Cycle Transfer (SRAM → SRAM): Divided by 2 (0 SRAM Waits)



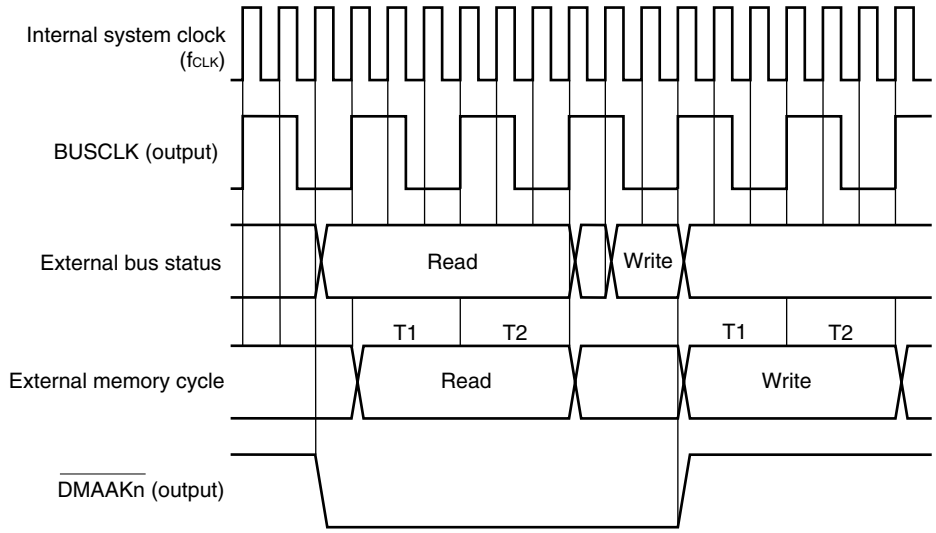
Remark n = 0 to 3

Figure 19-14. Outline of Timing During 2-Cycle Transfer (SRAM → SRAM): Divided by 3 (0 SRAM Waits) (1/2)



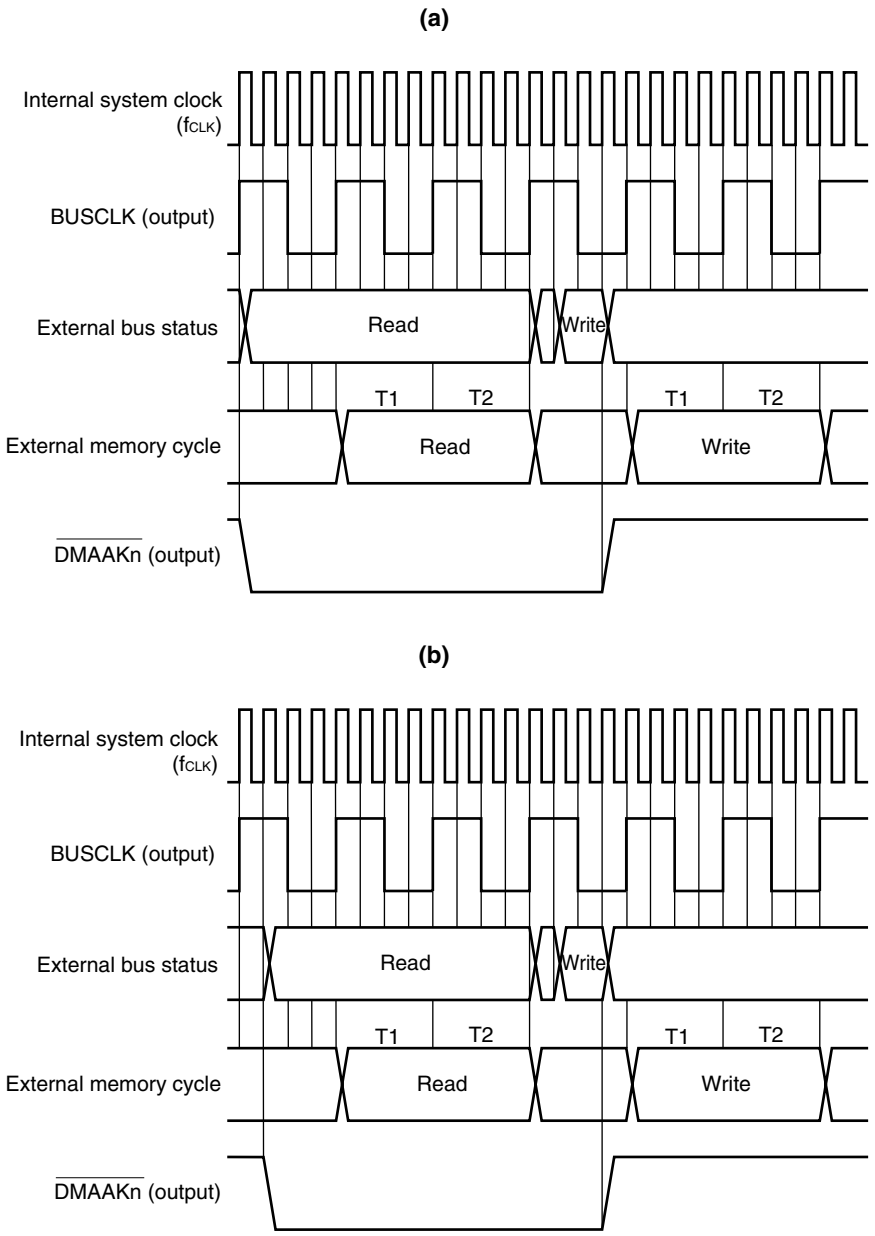
Remark n = 0 to 3

Figure 19-14. Outline of Timing During 2-Cycle Transfer (SRAM → SRAM): Divided by 3 (0 SRAM Waits) (2/2)



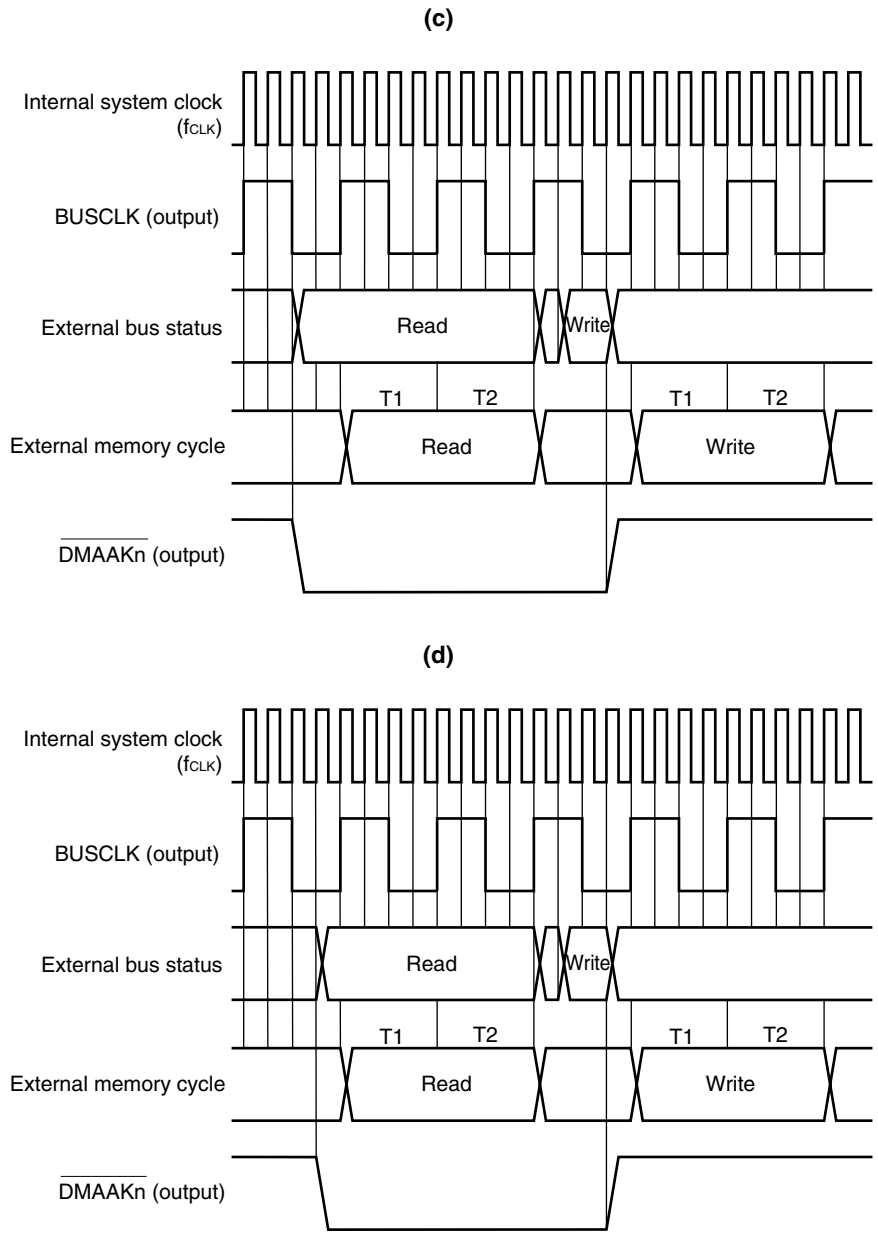
Remark n = 0 to 3

Figure 19-15. Outline of Timing During 2-Cycle Transfer (SRAM → SRAM): Divided by 4 (0 SRAM Waits) (1/2)



Remark n = 0 to 3

Figure 19-15. Outline of Timing During 2-Cycle Transfer (SRAM → SRAM): Divided by 4 (0 SRAM Waits) (2/2)



Remark n = 0 to 3

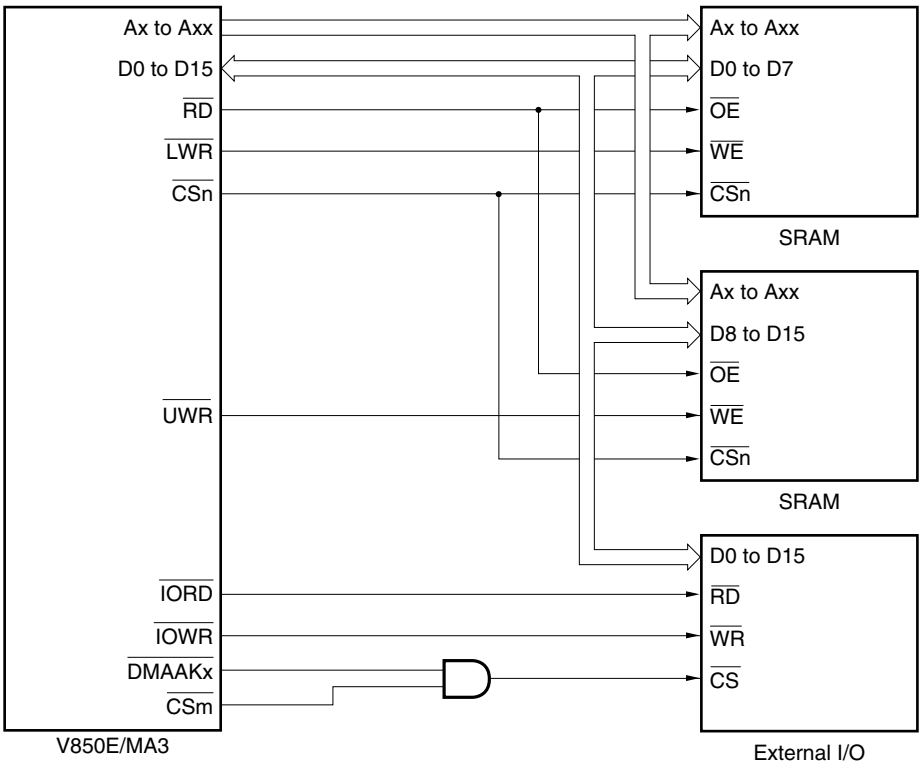
19.5.2 Flyby transfer

Since data is transferred in 1 cycle during a flyby transfer, a memory address is always output irrespective of whether it is a source address or a destination address, and read/write signals of the memory and external I/O become active at the same time. Therefore, the external I/O is selected by the $\overline{\text{DMAAK0}}$ to $\overline{\text{DMAAK3}}$ signals. Flyby transfer can be used only in the separate bus mode.

To perform a normal access to the external I/O by means other than DMA transfer, externally AND the $\overline{\text{CSm}}$ and $\overline{\text{DMAAKx}}$ signals (m = 0 to 7, x = 0 to 3), and connect the resultant signal to the chip select signal of the external I/O. A circuit example of a normal access, other than DMA transfer, to external I/O is shown below.

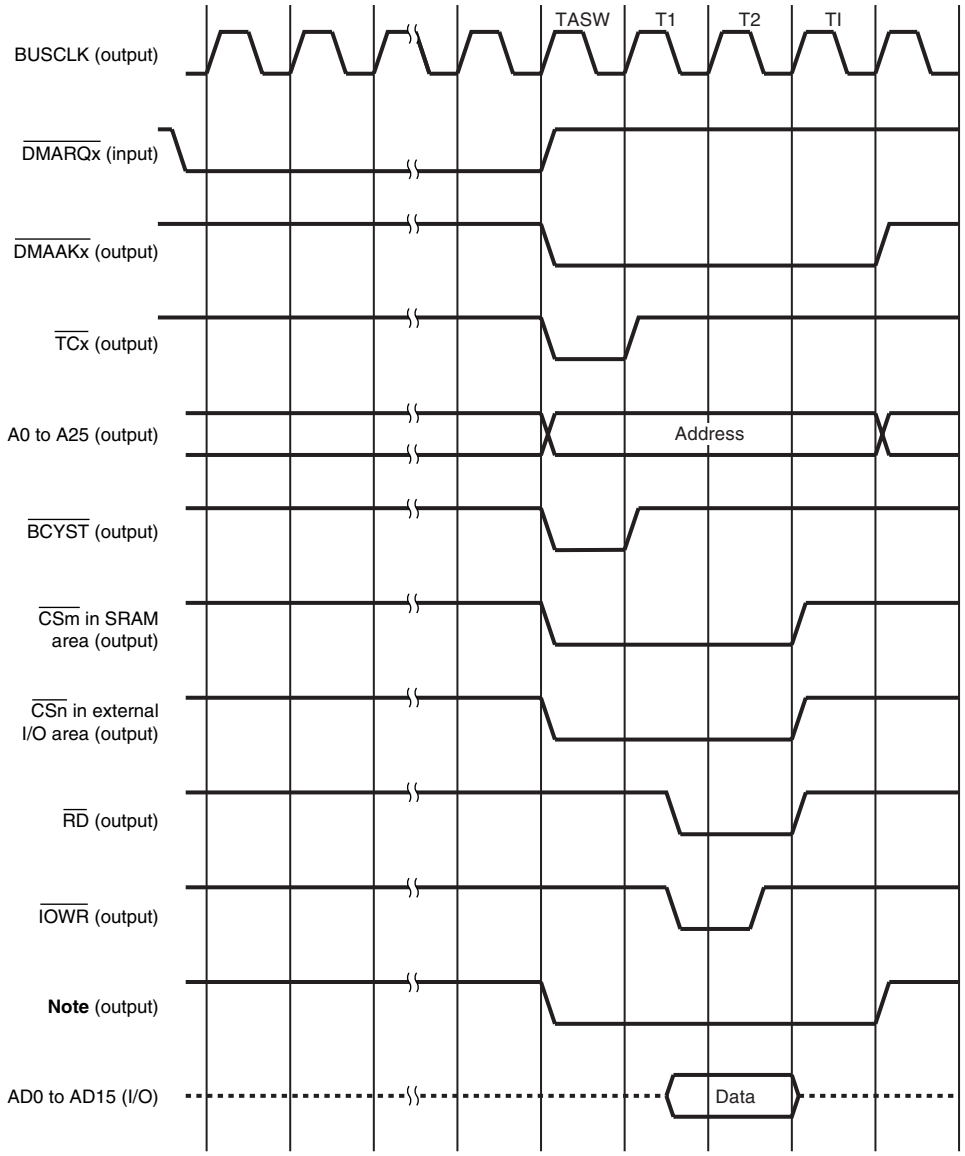
Caution If the memory is connected in the multiplexed bus mode, flyby transfer cannot be executed. Flyby transfer whose transfer object is SDRAM cannot be executed.

Figure 19-16. Circuit Example When Flyby Transfer Is Performed Between External I/O and SRAM



Remark n = 0 to 7, m = 0 to 7 (n ≠ m)
x = 0 to 3

Figure 19-17. Timing During DMA Flyby Transfer (SRAM → External I/O): In Single Transfer Mode (TASW = 1, TI = 1 Inserted)



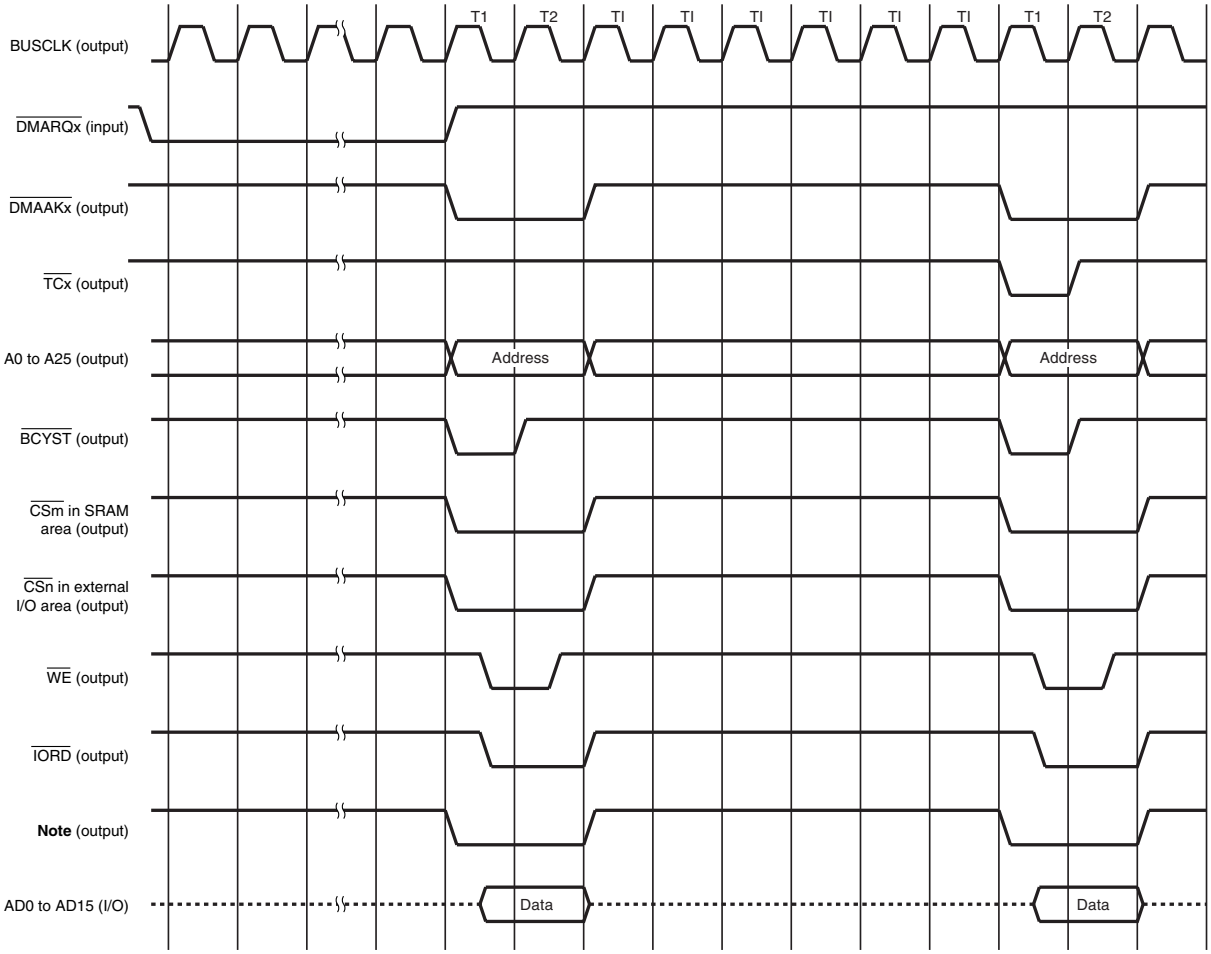
Note UBE, LBE

Remarks 1. The broken lines indicate the high-impedance state.

2. n = 0 to 7, m = 0 to 7 (n ≠ m), x = 0 to 3

Figure 19-18. Timing During DMA Flyby Transfer (External I/O → SRAM) (1/2)

(a) In single-step transfer mode



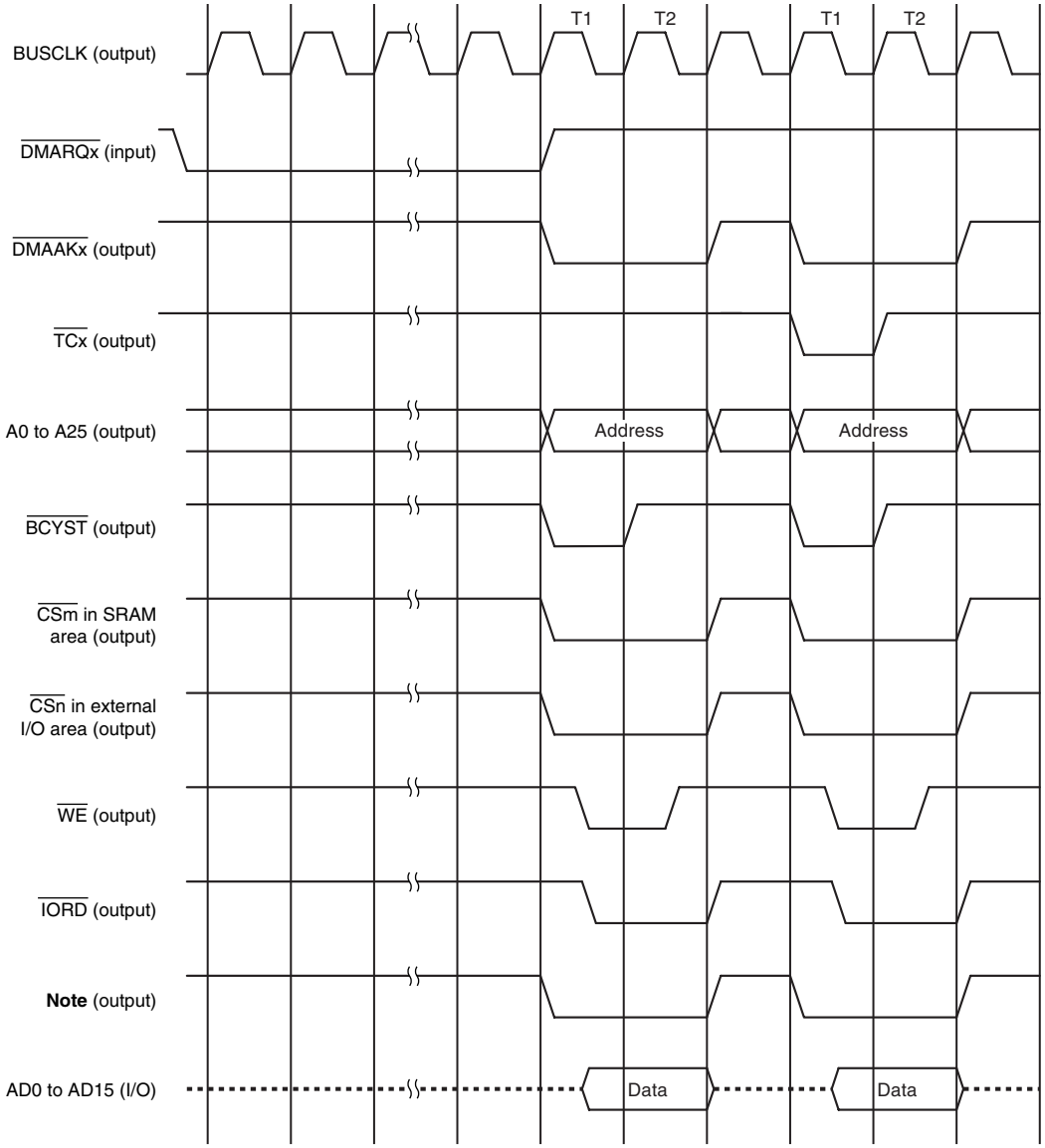
Note UBE, LBE

Remarks 1. The broken lines indicate the high-impedance state.

2. n = 0 to 7, m = 0 to 7 (n ≠ m), x = 0 to 3

Figure 19-18. Timing During DMA Flyby Transfer (External I/O → SRAM) (2/2)

(b) In block transfer mode



<R>

Note UBE, LBE

Remarks 1. The broken lines indicate the high-impedance state.

2. n = 0 to 7, m = 0 to 7 (n ≠ m), x = 0 to 3

19.6 Transfer Object

19.6.1 Transfer type and transfer object

Table 19-3 lists the relationships between the transfer type and transfer object. The mark "√" means "transfer possible", and the mark "×" means "transfer impossible".

Table 19-3. Relationship Between Transfer Type and Transfer Object

		Destination									
		2-Cycle Transfer					Flyby Transfer[Note 3]				
		Internal ROM	On-chip Peripheral I/O[Note 1]	External I/O	Internal RAM	External Memory	Internal ROM	On-chip Peripheral I/O	External I/O	Internal RAM	External Memory
Source	On-chip peripheral I/O[Note 1]	×	√	√	√	√[Note 2]	×	×	×	×	×
	External I/O	×	√	√	√	√	×	×	×	×	√[Note 4]
	Internal RAM	×	√	√	×	√	×	×	×	×	×
	External memory	×	√[Note 2]	√	√	√[Note 2]	×	×	√[Note 4]	×	×
	Internal ROM	×	×	×	×	×	×	×	×	×	×

Notes 1. If the transfer object is the on-chip peripheral I/O, only the single transfer mode can be used.

2. Transfer can also be executed between a little-endian area and a big-endian area.

3. Can be used only in separate bus mode.

4. Flyby transfer of which transfer object is SDRAM cannot be executed.

Cautions 1. The operation is not guaranteed for combinations of transfer destination and source marked with "×" in Table 19-3.

2. In the case of flyby transfer, make the data bus width the same for the source and destination.

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

4. When 2-cycle DMA transfer from the on-chip peripheral I/O to the internal RAM is executed during DMA transfer of 2 cycles, $\overline{TCn}$ signal output may be asserted twice, instead of once, after transfer of the last data, and the DMA transfer end interrupt (INTDMAn) may occur twice (n = 0 to 3). However, DMA transfer itself is normally ended. No such problem occurs during flyby transfer.

To prevent this, do not use the $\overline{TCn}$ signal. For the excessive DMA transfer end interrupts (INTDMAn), execute the following processing <1> and <2> in that order in the DMA transfer end interrupt servicing routine.

After processing <2>, by executing the application processing that is normally executed and recovery from the interrupt, the second DMA transfer end interrupt (INTDMAn) occurrence can be suppressed.

<1> Write 00H to the WAS register

<2> Clear the DMAICn.DMAIFn bit to 0 in the same channel as the DMA transfer end interrupt (INTDMAn) currently being serviced (n = 0 to 3).

Remark During 2-cycle DMA transfer, if the data bus width of the transfer source and that of the transfer destination are different, the operation becomes as follows.
If DMA transfer is executed to transfer data of an on-chip peripheral I/O register (as a transfer source or destination), be sure to specify the same transfer size as the register size. For example, to execute DMA transfer of an 8-bit register, be sure to specify byte (8-bit) transfer.

<16-bit transfer>

- Transfer from a 16-bit bus to an 8-bit bus
  A read cycle (16 bits) is generated and then a write cycle (8 bits) is generated twice consecutively.
- Transfer from an 8-bit bus to a 16-bit bus
  A read cycle (8 bits) is generated twice consecutively and then a write cycle (16 bits) is generated. Data is written to the transfer destination in the order of the lower and higher byes in little endian, and in the order of the higher and lower bytes in big endian.

<8-bit transfer>

- Transfer from a 16-bit bus to an 8-bit bus
  A read cycle (the higher 8 bits are high impedance) is generated and then a write cycle (8 bits) is generated.
- Transfer from an 8-bit bus to a 16-bit bus
  A read cycle (8 bits) is generated and then a write cycle (the higher 8 bits are high impedance) is generated. Data is written to the transfer destination in the order of the lower and higher byes in little endian, and in the order of the higher and lower bytes in big endian.

19.6.2 External bus cycles during DMA transfer

The external bus cycles during DMA transfer are shown below.

Table 19-4. External Bus Cycles During DMA Transfer

Transfer Type	Transfer Object	External Bus Cycle	
2-cycle transfer	On-chip peripheral I/O, internal RAM	None	–
	External I/O	Yes	SRAM cycle
	External memory	Yes	Memory access cycle set by the BCT register
Flyby transfer	Between external memory and external I/O	Yes	DMA flyby transfer cycle accessing memory that is set as external memory by the BCT register

19.7 DMA Channel Priorities

The DMA channel priorities are fixed as follows.

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

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

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

Caution If DMA is started by inputting the same signal to more than one $\overline{\text{DMARQn}}$ pin (n = 0 to 3), a DMA channel with a lower priority may be acknowledged before a DMA channel with a higher priority.

19.8 Next Address Setting Function

The DSAnH, DSAnL, DDAnH, DDAnL, and CBCn registers are two-stage FIFO buffer registers consisting of a master register and a slave register (n = 0 to 3).

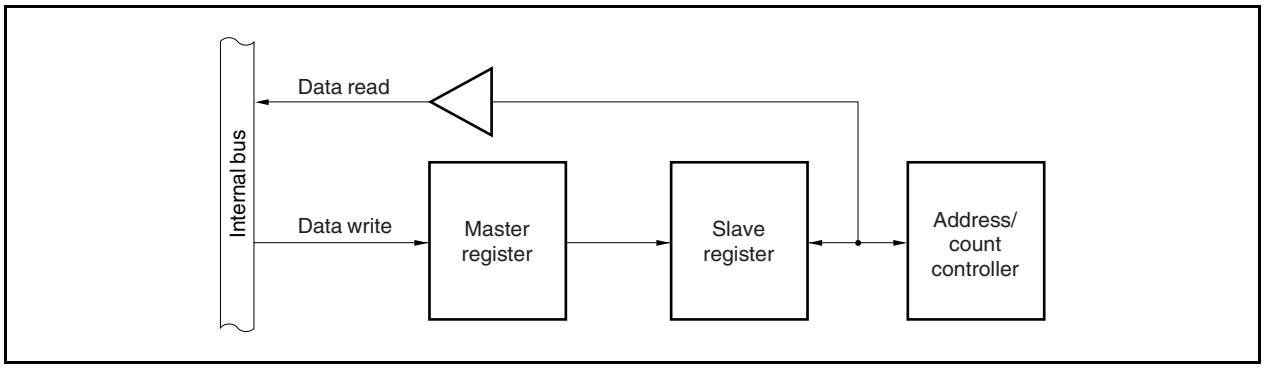
When the terminal count is issued, these registers are automatically rewritten with the value that was set immediately before.

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

Note To make new DMA transfer setting, confirm that DMA transfer has been started. If a new setting is made before the start of DMA transfer, the set value is overwritten to both the master and slave registers.

Figure 19-9 shows the configuration of the buffer register.

Figure 19-19. Buffer Register Configuration



The actual DMA transfer is executed in accordance with the contents of the slave register.

The set value to be reflected upon the master register and slave register differs as follows, depending on the timing (period) of setting.

(1) Period from system reset to the start of the first DMA transfer

The set values are reflected on both the master and slave registers.

(2) During DMA transfer (period start of DMA transfer to completion of DMA transfer)

The set value is reflected only on the master register and not on the slave register (the slave register holds the set value for the next DMA transfer).

After completion of DMA transfer, however, the contents of the master register are automatically overwritten to the slave register.

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

To check that DMA transfer has been started, confirm that the $\overline{\text{DMAAKn}}$ signal has been asserted or that the first transfer has been executed, by reading the DBCn register (n = 0 to 3).

(3) Period from completion of DMA transfer to start of next DMA transfer

The set value is reflected on both the master and slave registers.

Remark "Completion of DMA transfer" means either of the following cases.

- Completion of DMA transfer (terminal count)
- Forced termination of DMA transfer (setting INITn bit of DCHCn register to 1).

19.9 DMA Transfer Start Factors

There are 3 types of DMA transfer start factors, as shown below.

(1) Request from an external pin ($\overline{\text{DMARQn}}$)

Requests from the $\overline{\text{DMARQn}}$ pin are sampled each time the BUSCLK signal rises (n = 0 to 3).
Hold the request from $\overline{\text{DMARQn}}$ pin until the corresponding $\overline{\text{DMAAKn}}$ signal becomes active.
If a state whereby the DCHCn.Enn bit = 1 and TCn bit = 0 is set, the $\overline{\text{DMARQn}}$ signal becomes valid. If the $\overline{\text{DMARQn}}$ signal set by the DTFRn register becomes active in this status, DMA transfer starts.

(2) Request from software

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

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

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

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

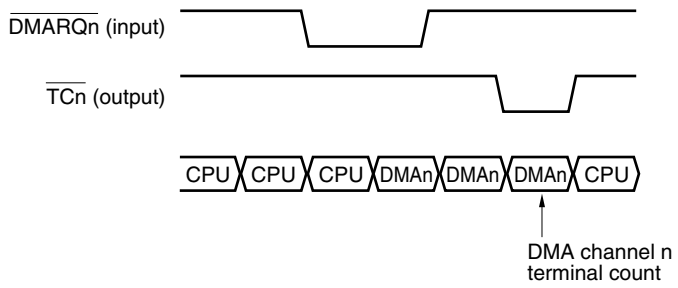
- Enn bit = 1
- TCn bit = 0

19.10 Terminal Count Output upon DMA Transfer End

The terminal count signal ($\overline{TCn}$) becomes active for one clock of BUSCLK during the last DMA transfer cycle (n = 3 to 0).

The $\overline{TCn}$ signal becomes active at the clock following the clock in which the $\overline{BCYST}$ signal becomes active during the last DMA transfer cycle.

Figure 19-20. Terminal Count Signal ($\overline{TCn}$) Timing Example (1)

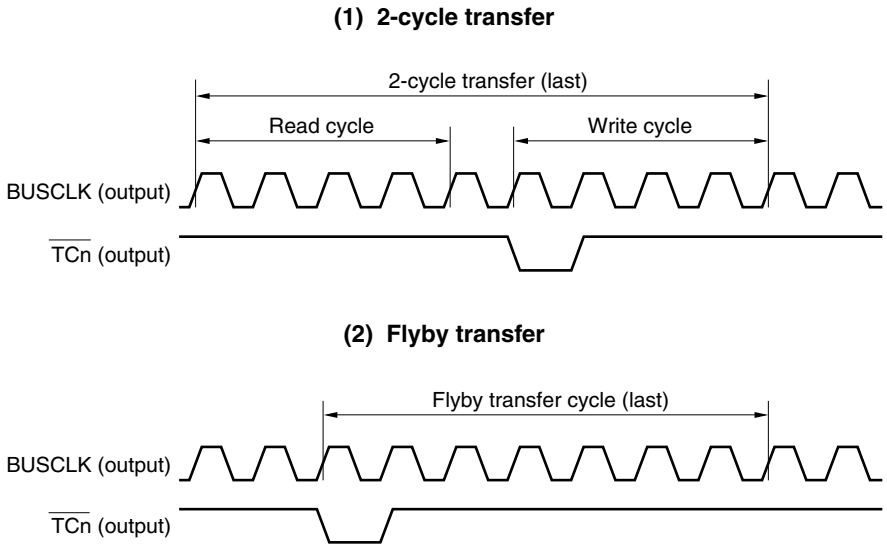


Remark n = 0 to 3

The $\overline{TCn}$ signal becomes active for one clock at the beginning of the write cycle of the last DMA transfer when 2-cycle transfer is executed.

When flyby transfer is executed, the $\overline{TCn}$ signal becomes active for one clock at the beginning of the last DMA transfer cycle.

Figure 19-21. Terminal Count Signal ($\overline{TCn}$) Timing Example (2)



Remark n = 0 to 3

19.11 Forcible Interruption

If the DTOC.DMSTPM bit is 0, DMA transfer can be forcibly interrupted by NMI input during DMA transfer.

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

Forcibly terminate and initialize DMA by using the DCHCn.INITn bit.

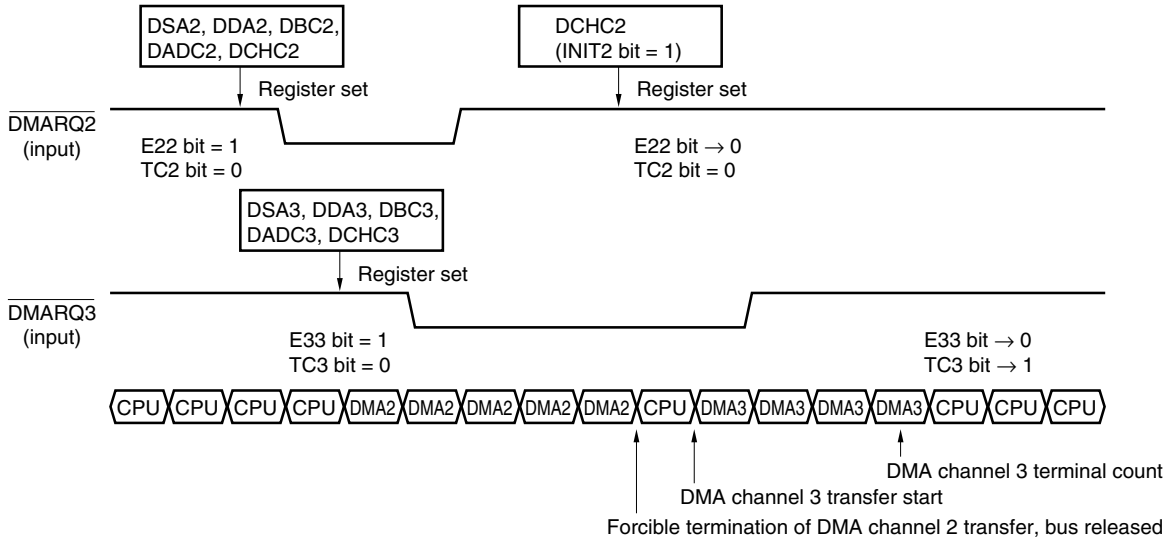
19.12 Forcible Termination

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

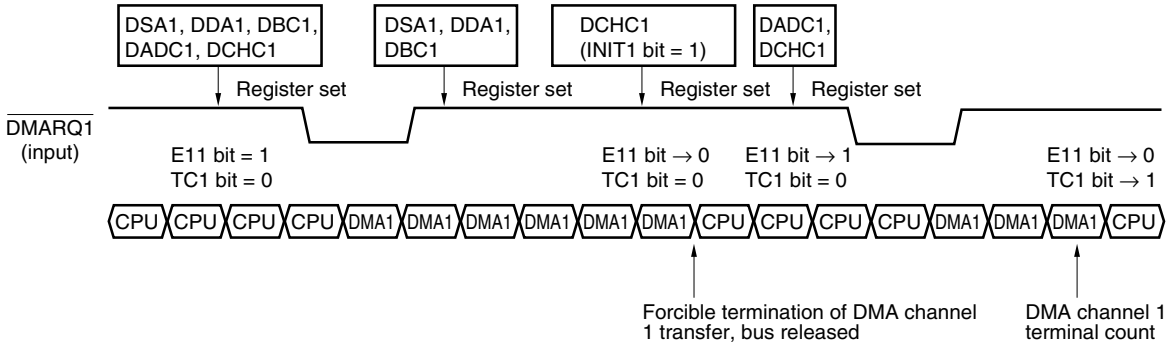
An example of forcible termination by the INITn bit is illustrated below (n = 0 to 3).

Figure 19-22. Example of Forcible Termination of DMA Transfer

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



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



Remark The values of the DSAn, DDAn, and DBCn registers (n = 0 to 3) are retained even when DMA transfer is forcibly stopped, because these registers are FIFO-format buffer registers. The next transfer condition can be set to these registers even while DMA transfer is in progress. On the other hand, the setting of the DADCn and DCHCn registers is invalid during DMA transfer because these registers are not buffer registers (see **19.8 Next Address Setting Function**, **19.3.4 DMA addressing control registers 0 to 3 (DADC0 to DADC3)**, and **19.3.5 DMA channel control registers 0 to 3 (DCHC0 to DCHC3)**).

19.13 Times Related to DMA Transfer

The overhead before and after DMA transfer and minimum number of execution internal system clocks for DMA transfer are listed below. In the case of external memory access, the time depends on the type of external memory connected.

Table 19-5. Number of Minimum Execution Internal System Clocks in DMA Cycle

DMA Cycle		Minimum Number of Execution Clocks
<1> Time to respond to DMA request		4 internal system clocks[Note 1]
<2> Memory access	External memory access	Differs depending on the memory connected
	Internal RAM access	2 internal system clocks[Note 2]
	On-chip peripheral I/O register access	4 internal system clocks + Number of wait cycles specified by VSWC register

<R>

Notes 1. If an external interrupt (INTPn) is specified as a factor of starting DMA transfer, noise elimination time is added (n = 000, 001, 004, 005, 010 to 013, 021, 022, 050, 051, 106, 107, 114, 115, 124 to 126, 130 to 134, 137).

2. Two clocks are required for the DMA cycle.

The minimum execution clock in the DMA cycle in each transfer mode is as follows.

2-cycle transfer

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

Note One clock is always inserted between the read cycle and write cycle of DMA transfer. However, two clocks are inserted when the undivided external bus clock is used.

Flyby transfer: DMA response time (<1>) + External memory access (<2>)

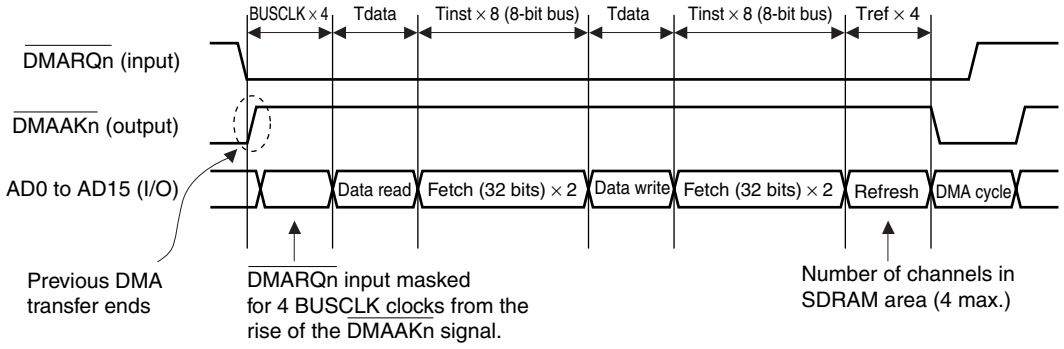
19.14 Maximum Response Time for DMA Transfer Request

The response time for a DMA transfer request becomes the longest under the following conditions (state in which all the refresh cycles for the SDRAM are enabled).

<R> **Caution The wait time caused by the following conditions is not included.**

- **When DMA transfer triggered by other than the $\overline{\text{DMARQn}}$ pin input occurs**
- **External bus hold**

Condition	Instruction fetch from an external memory using 8-bit data bus width Execution of a bit manipulation instruction (SET1, CLR1, or NOT1) A branch instruction (JR, JARL, JMP, or Bcond) follows a bit manipulation instruction Either DMA transfer source or destination is internal RAM The DIFC.DRMKn bit = 1 SDRAM is used in four CSm spaces
Response time	Tinst ×16 + Tdata × 2 + Tref × 4 + BUSCLK × 4



Remarks 1. Tinst: Number of clocks per bus cycle during instruction fetch
Tdata: Number of clocks per bus cycle during data access
Tref: Number of clocks per refresh cycle
BUSCLK × 4: Time for masking $\overline{\text{DMARQn}}$ input

2. n = 0 to 3
m = 1, 3, 4, 6

<R>

19.15 Cautions

(1) Memory boundary

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

(2) Transfer of misaligned data

DMA transfer of 32-/16-bit bus width misaligned data is not supported. If the source or the destination address is set to an odd address, the LSB of the address is forcibly handled as “0”.

(3) Bus arbitration for CPU

When an external device is targeted for DMA transfer, the CPU can access the internal ROM and internal RAM (when not subject to DMA transfer).

The CPU can access the internal ROM during DMA transfer between the on-chip peripheral I/O and internal RAM.

(4) Holding $\overline{\text{DMARQn}}$ signal

Be sure to keep the $\overline{\text{DMARQn}}$ signal active until the $\overline{\text{DMAAKn}}$ signal becomes active (n = 0 to 3).

(5) $\overline{\text{DMAAKn}}$ signal output

When the transfer object is internal RAM, the $\overline{\text{DMAAKn}}$ signal is not output during a DMA cycle for internal RAM (for example, if 2-cycle transfer is performed from internal RAM to an external memory, the $\overline{\text{DMAAKn}}$ signal is output only during a DMA write cycle for the external memory).

If the transfer object is the on-chip peripheral I/O, the $\overline{\text{DMAAKn}}$ signal is output even in the DMA cycle executed on the on-chip peripheral I/O.

(6) DMA start factors

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

(7) Program execution and DMA transfer with internal RAM

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

19.15.1 Suspension factors

DMA transfer is suspended if the following factors are issued.

- Bus hold
- Refresh cycle

If the factor that is suspending DMA transfer disappears, DMA transfer promptly restarts.

19.16 DMA Transfer End

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

CHAPTER 20 INTERRUPT/EXCEPTION PROCESSING FUNCTION

The V850E/MA3 is provided with a dedicated interrupt controller (INTC) for interrupt servicing and can process a total of 77 interrupt requests.

An interrupt is an event that occurs independently of program execution, and an exception is an event whose occurrence is dependent on program execution.

The V850E/MA3 can process interrupt requests from the on-chip peripheral hardware and external sources. Moreover, exception processing can be started by the TRAP instruction (software exception) or by generation of an exception event (i.e. fetching of an illegal opcode) (exception trap).

20.1 Features

○ Interrupts
- Non-maskable interrupts: 2 sources (external: 1 source, internal: 1 source)

Caution P20 is fixed to NMI input. The level of the NMI pin is read when the P2.P20 bit is read, regardless of the value of the PM2 and PMC2 registers. Set the valid edge of the NMI pin by using the NMIR and NMIF registers (default value: falling edge detection).

- Maskable interrupts External: 25 sources, internal: 49/50 sources (See **Table 1-1**)
- 8 levels of programmable priorities (maskable interrupts)
- Multiple interrupt control according to priority
- Masks can be specified for each maskable interrupt request.
- Noise elimination, edge detection, and valid edge specification for external interrupt request signals.

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

Interrupt sources are listed in Table 20-1.

Table 20-1. Interrupt Source List (1/3)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Control Register	Generating Source	Generating Unit				
Reset	Interrupt	RESET	–	$\overline{\text{RESET}}$ pin input	Pin	–	0000H	00000000H	Undefined
			–	WDT overflow (WDTRES)	WDT				
Non-maskable	Interrupt	NMI	–	NMI pin valid edge input	Pin		0010H	00000010H	nextPC
		INTWDT	–	WDT overflow	WDT		0020H	00000020H	Undefined
Software exception	Exception	TRAP0n[Note 1]	–	TRAP instruction	–	–	004nH	00000040H	nextPC
	Exception	TRAP1n[Note 1]	–	TRAP instruction	–	–	005nH	00000050H	nextPC
Exception trap	Exception	ILGOP/ DBG0	–	Illegal instruction code/ DBTRAP instruction	–	–	0060H	00000060H	nextPC
<R> Maskable	Interrupt	INTWDTM	WDTIC	WDT overflow (interval timer mode)	WDT	0	0080H	00000080H	nextPC
	Interrupt	INTP000/ INTCCP00	P00IC0	$\overline{\text{INTP000}}$ pin/ TP0CCR0 capture input/ compare match	Pin/ TMP0	1	0090H	00000090H	nextPC
	Interrupt	INTP001/ INTCCP01	P00IC1	$\overline{\text{INTP001}}$ pin/ TP0CCR1 capture input/ compare match	Pin/ TMP0	2	00A0H	000000A0H	nextPC
	Interrupt	INTP004/ INTCC101	P00IC4	$\overline{\text{INTP004}}$ pin/ CC101 capture input/ compare match	Pin/ TMENC10	3	00B0H	000000B0H	nextPC
	Interrupt	INTP005/ INTCC100	P00IC5	$\overline{\text{INTP005}}$ pin/ CC100 capture input/ compare match	Pin/ TMENC10	4	00C0H	000000C0H	nextPC
	Interrupt	INTP106	P10IC6	$\overline{\text{INTP106}}$ pin	Pin	5	00D0H	000000D0H	nextPC
	Interrupt	INTP107	P10IC7	$\overline{\text{INTP107}}$ pin	Pin	6	00E0H	000000E0H	nextPC
	Interrupt	INTP010/ INTCCQ0	P01IC0	$\overline{\text{INTP010}}$ pin/ TQ0CCR0 capture input/ compare match[Note 2]	Pin/ TMQ0	7	00F0H	000000F0H	nextPC
	Interrupt	INTP011/ INTCCQ1	P01IC1	$\overline{\text{INTP011}}$ pin/ TQ0CCR1 capture input/ compare match	Pin/ TMQ0	8	0100H	00000100H	nextPC
	Interrupt	INTP012/ INTCCQ2	P01IC2	$\overline{\text{INTP012}}$ pin/ TQ0CCR2 capture input/ compare match	Pin/ TMQ0	9	0110H	00000110H	nextPC
	Interrupt	INTP013/ INTCCQ3	P01IC3	$\overline{\text{INTP013}}$ pin/ TQ0CCR3 capture input/ compare match	Pin/ TMQ0	10	0120H	00000120H	nextPC
	Interrupt	INTP114	P11IC4	$\overline{\text{INTP114}}$ pin	Pin	11	0130H	00000130H	nextPC
	Interrupt	INTP115	P11IC5	$\overline{\text{INTP115}}$ pin	Pin	12	0140H	00000140H	nextPC
	Interrupt	INTP021/ INTCCP10	P02IC1	$\overline{\text{INTP021}}$ pin/ TP1CCR0 capture input/ compare match	Pin/ TMP1	13	0150H	00000150H	nextPC

Notes 1. n = 0 to FH

2. When TMQ0 is used in the 6-phase PWM output mode, it functions as INTCCQ0 (crest interrupt) from the TMQ0 option (TMQOP0).

Table 20-1. Interrupt Source List (2/3)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Control Register	Generating Source	Generating Unit				
Maskable	Interrupt	INTP022/ INTCCP11	P02IC2	$\overline{\text{INTP022}}$ pin/ TP1CCR1 capture input/ compare match	Pin/ TMP1	14	0160H	00000160H	nextPC
	Interrupt	INTP124	P12IC4	$\overline{\text{INTP124}}$ pin	Pin	15	0170H	00000170H	nextPC
	Interrupt	INTP125	P12IC5	$\overline{\text{INTP125}}$ pin	Pin	16	0180H	00000180H	nextPC
	Interrupt	INTP126	P12IC6	$\overline{\text{INTP126}}$ pin	Pin	17	0190H	00000190H	nextPC
	Interrupt	INTP130	P13IC0	$\overline{\text{INTP130}}$ pin	Pin	18	01A0H	000001A0H	nextPC
	Interrupt	INTP131	P13IC1	$\overline{\text{INTP131}}$ pin	Pin	19	01B0H	000001B0H	nextPC
	Interrupt	INTP132	P13IC2	$\overline{\text{INTP132}}$ pin	Pin	20	01C0H	000001C0H	nextPC
	Interrupt	INTP133	P13IC3	$\overline{\text{INTP133}}$ pin	Pin	21	01D0H	000001D0H	nextPC
	Interrupt	INTP134	P13IC4	$\overline{\text{INTP134}}$ pin	Pin	22	01E0H	000001E0H	nextPC
	Interrupt	INTP137	P13IC7	$\overline{\text{INTP137}}$ pin	Pin	23	01F0H	000001F0H	nextPC
	Interrupt	INTP050/ INTCCP20	P05IC0	$\overline{\text{INTP050}}$ pin/ TP2CCR0 capture input/ compare match	Pin/ TMP2	24	0200H	00000200H	nextPC
	Interrupt	INTP051/ INTCCP21	P05IC1	$\overline{\text{INTP051}}$ pin/ TP2CCR1 capture input/ compare match	Pin/ TMP2	25	0210H	00000210H	nextPC
	Interrupt	INTCMD0	CMICD0	CMD0 compare match	TMD0	26	0220H	00000220H	nextPC
	Interrupt	INTCMD1	CMICD1	CMD1 compare match	TMD1	27	0230H	00000230H	nextPC
	Interrupt	INTCMD2	CMICD2	CMD2 compare match	TMD2	28	0240H	00000240H	nextPC
	Interrupt	INTCMD3	CMICD3	CMD3 compare match	TMD3	29	0250H	00000250H	nextPC
	Interrupt	INTCM100	CM10IC0	CM100 compare match	TMENC10	30	0260H	00000260H	nextPC
	Interrupt	INTCM101	CM10IC1	CM101 compare match	TMENC10	31	0270H	00000270H	nextPC
	Interrupt	INTOVP0	OVPIC0	TMP0 overflow	TMP0	32	0280H	00000280H	nextPC
	Interrupt	INTOVQ	OVQIC	TMQ0 overflow/underflow[Note]	TMQ0	33	0290H	00000290H	nextPC
	Interrupt	INTOVP1	OVPIC1	TMP1 overflow	TMP1	34	02A0H	000002A0H	nextPC
	Interrupt	INTOVP2	OVPIC2	TMP2 overflow	TMP2	35	02B0H	000002B0H	nextPC
	Interrupt	INTDMA0	DMAIC0	DMA channel 0 transfer completion	DMA0	36	02C0H	000002C0H	nextPC
	Interrupt	INTDMA1	DMAIC1	DMA channel 1 transfer completion	DMA1	37	02D0H	000002D0H	nextPC
	Interrupt	INTDMA2	DMAIC2	DMA channel 2 transfer completion	DMA2	38	02E0H	000002E0H	nextPC
	Interrupt	INTDMA3	DMAIC3	DMA channel 3 transfer completion	DMA3	39	02F0H	000002F0H	nextPC
	Interrupt	INTSER0/ INTCSIER0	SEIC0	UARTA0 receive error/ CSIB0 receive error	UARTA0/ CSIB0	40	0300H	00000300H	nextPC
	Interrupt	INTSR0/ INTCSIR0	SRIC0	UARTA0 reception completion/CSIB0 reception completion	UARTA0/ CSIB0	41	0310H	00000310H	nextPC

Note When TMQ0 is used in the 6-phase PWM output mode, it functions as INTOVQ (valley interrupt) from the TMQ0 option (TMQOP0).

Table 20-1. Interrupt Source List (3/3)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Control Register	Generating Source	Generating Unit				
Maskable	Interrupt	INTST0/ INTCSIT0	STIC0	UARTA0 transmission enable/CSIB0 transmission enable	UARTA0/ CSIB0	42	0320H	00000320H	nextPC
	Interrupt	INTSER1/ INTCSIER1	SEIC1	UARTA1 receive error/ CSIB1 receive error	UARTA1/ CSIB1	43	0330H	00000330H	nextPC
	Interrupt	INTSR1/ INTCSIR1	SRIC1	UARTA1 reception completion/CSIB1 reception completion	UARTA1/ CSIB1	44	0340H	00000340H	nextPC
	Interrupt	INTST1/ INTCSIT1	STIC1	UARTA1 transmission enable/CSIB1 transmission enable	UARTA1/ CSIB1	45	0350H	00000350H	nextPC
	Interrupt	INTSER2/ INTCSIER2	SEIC2	UARTA2 receive error/ CSIB2 receive error	UARTA2/ CSIB2	46	0360H	00000360H	nextPC
	Interrupt	INTSR2/ INTCSIR2	SRIC2	UARTA2 reception completion/CSIB2 reception completion	UARTA2/ CSIB2	47	0370H	00000370H	nextPC
	Interrupt	INTST2/ INTCSIT2	STIC2	UARTA2 transmission enable/CSIB2 transmission enable	UARTA2/ CSIB2	48	0380H	00000380H	nextPC
	Interrupt	INTSER3	SEIC3	UARTA3 receive error	UARTA3	49	0390H	00000390H	nextPC
	Interrupt	INTSR3/ INTIIC[Note]	SRIC3	UARTA3 reception completion/I^2C serial transfer completion	UARTA3/ I^2C	50	03A0H	000003A0H	nextPC
	Interrupt	INTST3	STIC3	UARTA3 transmission enable	UARTA3	51	03B0H	000003B0H	nextPC
	Interrupt	INTAD	ADIC	A/D conversion completion	ADC	52	03C0H	000003C0H	nextPC

Note I^2C bus versions (Y products) only (see **Table 1-1**)

Remarks 1. Default Priority: The priority order when two or more maskable interrupt request signals are generated at the same time. The highest priority is 0.

Restored PC: The value of the program counter (PC) saved to EIPC, FEPC, or DBPC of the CPU when interrupt servicing is started. Note, however, that the restored PC when a non-maskable or maskable interrupt is acknowledged while one of the following instructions is being executed does not become the nextPC. (If an interrupt is acknowledged during interrupt execution, execution stops, and then resumes after the interrupt servicing has finished. In this case, the address of the aborted instruction is the restore PC.)

- Load instructions (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)
- Division instructions (DIV, DIVH, DIVU, DIVHU)
- PREPARE, DISPOSE instructions (only if an interrupt is generated before the stack pointer is updated)

nextPC: The PC value 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).

20.2 Non-Maskable Interrupts

A non-maskable interrupt request signal is acknowledged unconditionally, even when interrupts are in the interrupt disabled (DI) status. An NMI is not subject to priority control and takes precedence over all the other interrupt request signals.

The V850E/MA3 has the following two non-maskable interrupt request signals.

- NMI pin input (NMI)
- Non-maskable interrupt request signal generated by overflow of watchdog timer (INTWDT)

The valid edge of the NMI pin is selected from the rising edge, falling edge, or both rising/falling edge by the NMIR and NMIF registers.

INTWDT functions when the WDTM.WDTM4 and WDTM.WDTM3 bits are set to “10”.

If two or more non-maskable interrupts occur at the same time, the interrupt with a higher priority is serviced, as follows (the interrupt with a lower priority is ignored).

INTWDT > NMI

If a new NMI or INTWDT request is issued while an NMI is being serviced, it is serviced as follows.

(1) If new NMI request is issued while NMI is being serviced

The new NMI request is held pending, regardless of the value of the PSW.NP bit in the CPU. The pending NMI interrupt is acknowledged after the NMI currently under execution has been serviced (after the RETI instruction has been executed). However, even if two or more new NMI requests are generated during NMI servicing, only one NMI request is acknowledged.

(2) If INTWDT request is issued while NMI is being serviced

The INTWDT request is held pending if the PSW.NP bit remains set (1) while the NMI is being serviced. The pending INTWDT request is acknowledged after the NMI currently under execution has been serviced (after the RETI instruction has been executed).

If the PSW.NP bit is cleared (0) while the NMI is being serviced, the newly generated INTWDT request is executed (the NMI servicing is stopped).

Caution If a non-maskable interrupt request is generated, the values of the PC and PSW are saved to the NMI status save registers (FEPC and FEPSW). At this time, execution can be returned by the RETI instruction only from an NMI. Execution cannot be returned while INTWDT is being serviced. Therefore, reset the system after the interrupt has been serviced.

Figure 20-1. Non-Maskable Interrupt Request Signal Acknowledgment Operation (1/2)

(a) NMI and INTWDT request signals generated at the same time

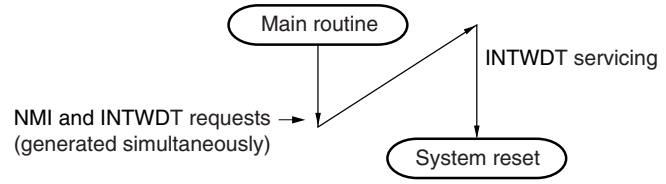
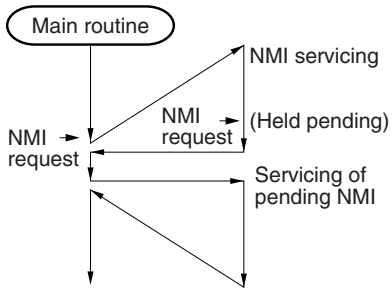
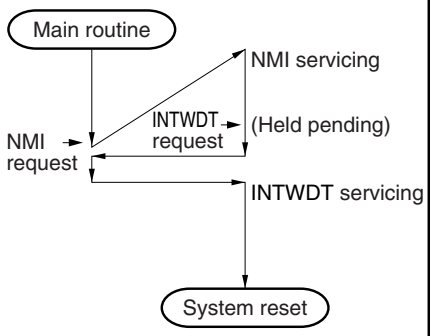
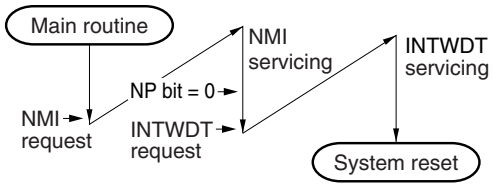
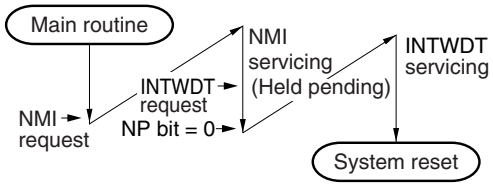
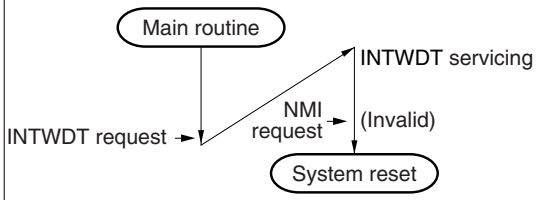
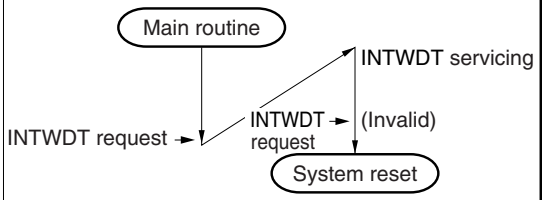


Figure 20-1. Non-Maskable Interrupt Request Signal Acknowledgment Operation (2/2)

(b) Non-maskable interrupt request signal generated during non-maskable interrupt servicing

Non-maskable interrupt being serviced	Non-maskable interrupt request signal generated during non-maskable interrupt servicing	
	NMI	INTWDT
NMI	• NMI request generated during NMI servicing 	• INTWDT request generated during NMI servicing (PSW.NP bit = 1 retained before INTWDT request)  • INTWDT request generated during NMI servicing (NP bit = 0 set before INTWDT request)  • INTWDT request generated during NMI servicing (NP bit = 0 set after INTWDT request) 
INTWDT	• NMI request generated during INTWDT servicing 	• INTWDT request generated during INTWDT servicing 

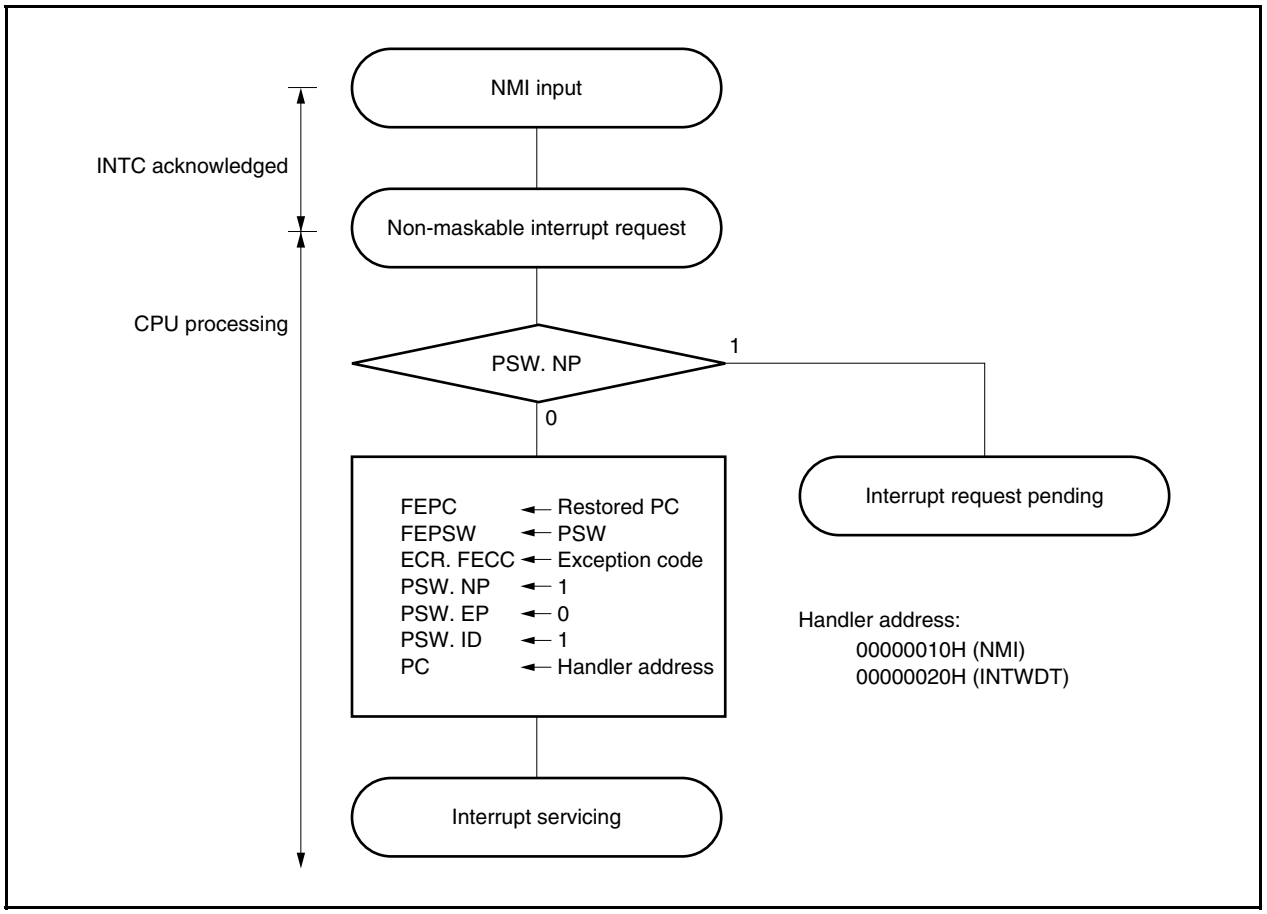
20.2.1 Operation

If a non-maskable interrupt request signal 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, 0020H) to the higher halfword (FECC) of ECR.
(4) Sets the PSW.NP and PSW.ID bits to 1 and clears the PSW.EP bit to 0.
(5) Loads the handler address (00000010H, 00000020H) of the non-maskable interrupt routine to the PC, and transfers control.

The following shows the non-maskable interrupt servicing.

Figure 20-2. Non-Maskable Interrupt Servicing



20.2.2 Restore

(1) NMI input

Execution is restored from the NMI servicing by the RETI instruction.

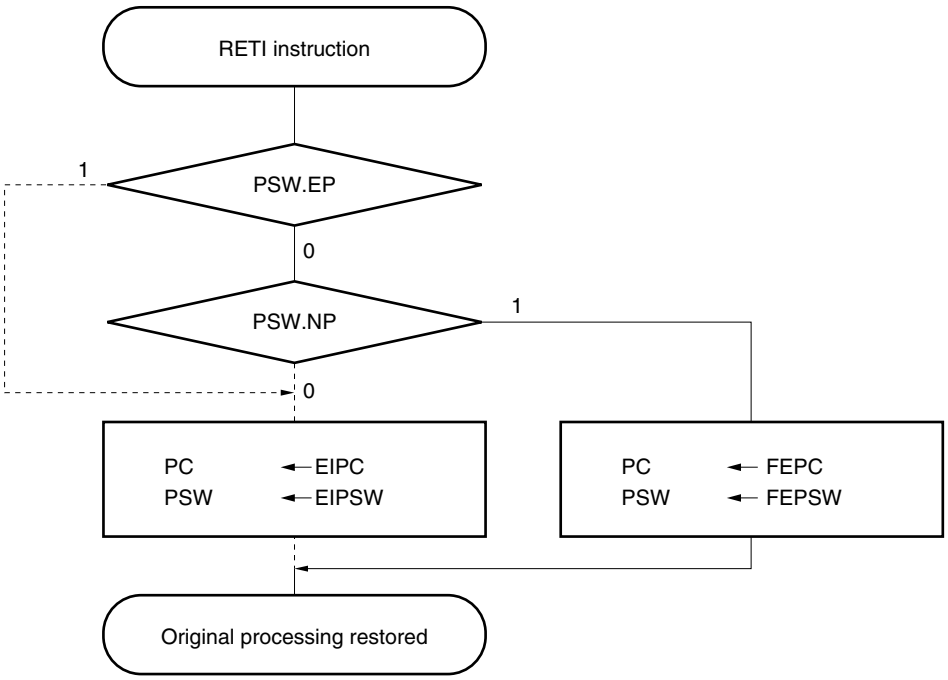
When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

<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 20-3. RETI Instruction Processing



Caution When the PSW.EP bit and PSW.NP bit are changed by the LDSR instruction during NMI servicing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP bit back to 0 and PSW.NP bit back to 1 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

(2) INTWDT signal

Execution cannot be restored by the RETI instruction. Reset the system after interrupt servicing has been completed.

20.2.3 Non-maskable interrupt status flag (NP)

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

This flag is set when a non-maskable interrupt request signal has been acknowledged, and masks non-maskable interrupt requests to prohibit multiple interrupts from being acknowledged.

After reset: 00000020H

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

NP	Non-maskable interrupt servicing status
0	No non-maskable interrupt servicing
1	Non-maskable interrupt servicing in progress

20.3 Maskable Interrupts

Maskable interrupt request signals can be masked by interrupt control registers. The V850E/MA3 has 75 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 nested.

To enable multiple interrupt servicing, however, save EIPC and EIPSW to memory or registers before executing the EI instruction, and execute the DI instruction before the RETI instruction to restore the original values of EIPC and EIPSW.

20.3.1 Operation

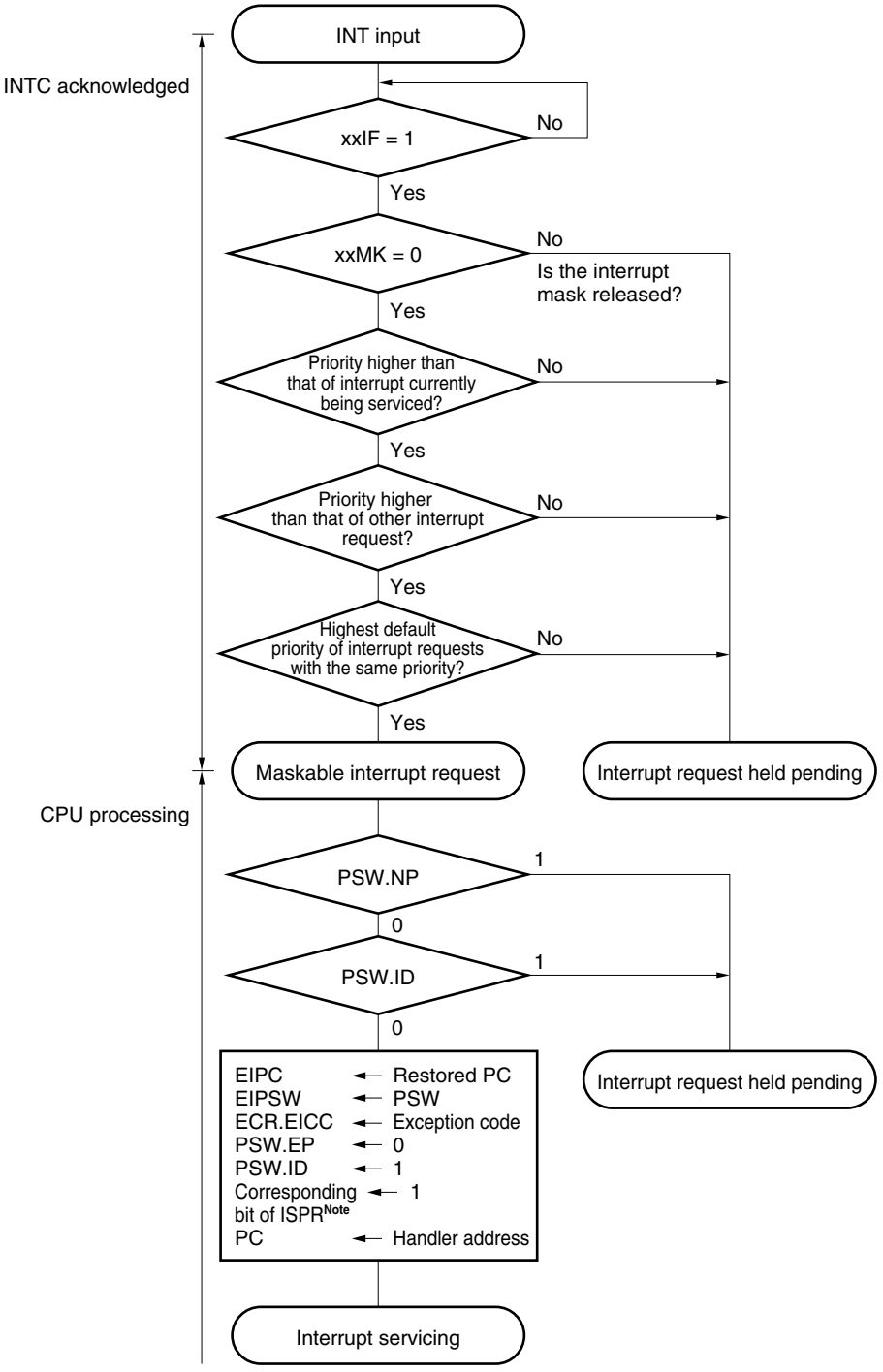
If a maskable interrupt occurs, the CPU performs the following processing, and transfers control to a handler routine.

<1> Saves the restored PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower halfword of ECR (EICC).
<4> Sets the PSW.ID bit 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 PSW.ID bit = 1) are held pending inside the INTC. In this case, servicing a new maskable interrupt is started in accordance with the priority of the pending maskable interrupt request signal if either the maskable interrupt is unmasked or NP and ID bits are cleared to 0 by using the RETI or LDSR instruction.

How maskable interrupts are serviced is illustrated below.

Figure 20-4. Maskable Interrupt Servicing



Note For the ISPR register, see **20.3.6 In-service priority register (ISPR)**.

20.3.2 Restore

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

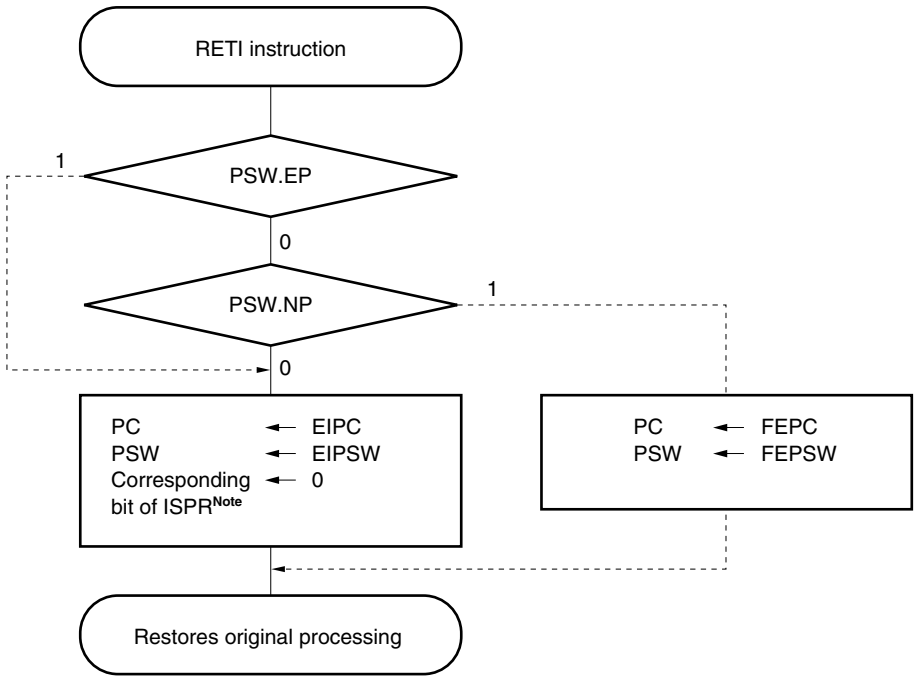
When the RETI instruction is executed, the CPU performs the following steps, and transfers control to the address of the restored PC.

<1> 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 to the address of the restored PC and PSW.

The following illustrates the processing of the RETI instruction.

Figure 20-5. RETI Instruction Processing



Note For the ISPR register, see **20.3.6 In-service priority register (ISPR)**.

Caution When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during maskable interrupt servicing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set EP bit back to 0 and NP bit back to 0 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

20.3.3 Priorities of maskable interrupts

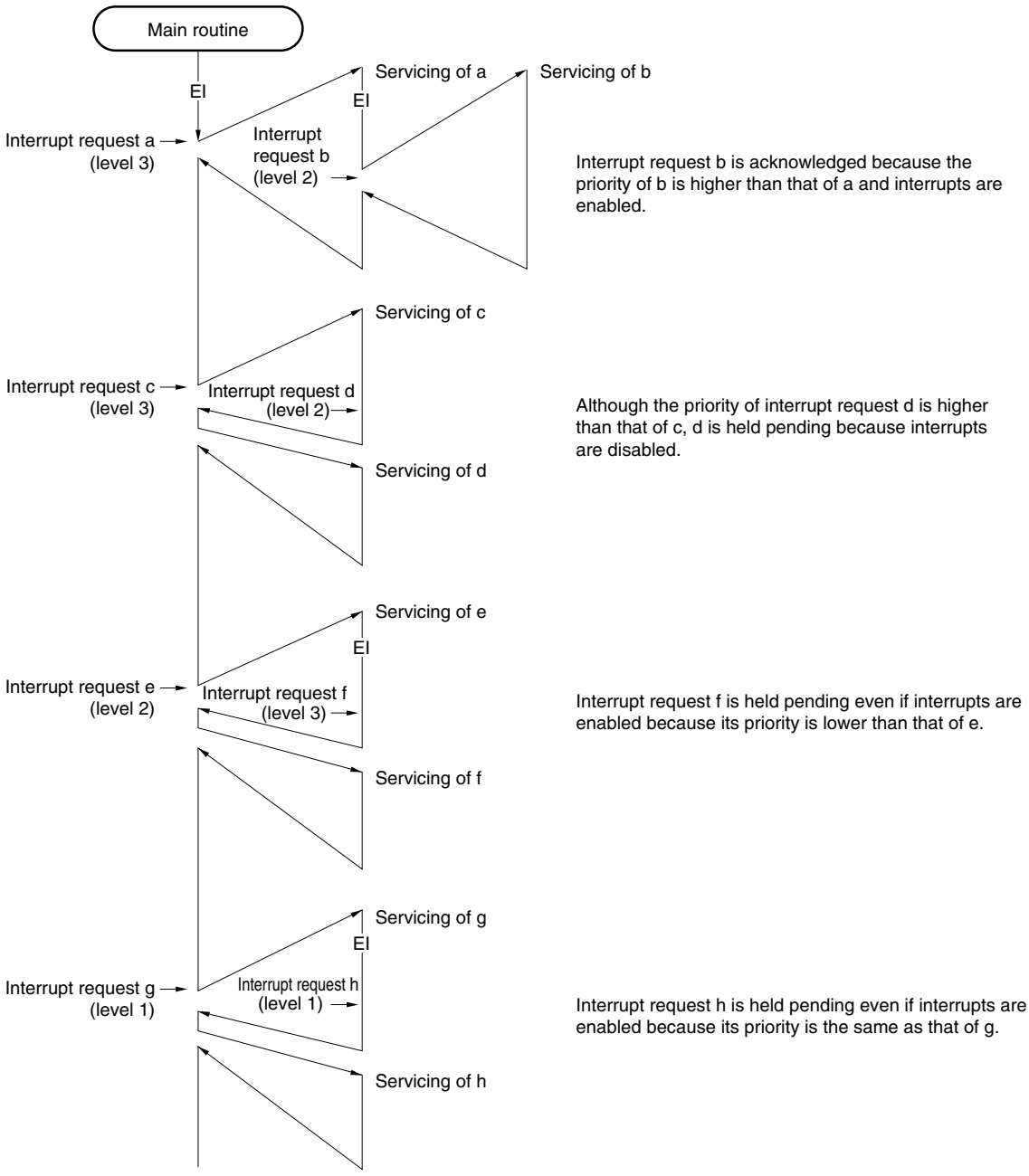
The V850E/MA3 provides multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced. Multiple interrupts can be controlled by priority levels.

There are two types of priority level control: control based on the default priority levels, and control based on the programmable priority levels that are specified by the interrupt priority level specification bit (xxPRn) of the interrupt control register (xxICn). When two or more interrupts having the same priority level specified by the xxPRn bit are generated at the same time, interrupts are serviced in order depending on the priority level allocated to each interrupt request signal type (default priority level) beforehand. For more information, see **Table 20-1 Interrupt Source List**. Programmable priority control customizes interrupt request signals into eight levels by the setting of the priority level specification flag.

Note that when an interrupt request signal is acknowledged, the PSW.ID flag is automatically set to 1. Therefore, when multiple interrupts are to be used, clear the ID flag to 0 beforehand (for example, by placing the EI instruction in the interrupt service program) to set the interrupt enable mode.

Remark xx: Identification name of each peripheral unit (see **Table 20-2**)
n: Peripheral unit number (see **Table 20-2**)

Figure 20-6. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (1/2)

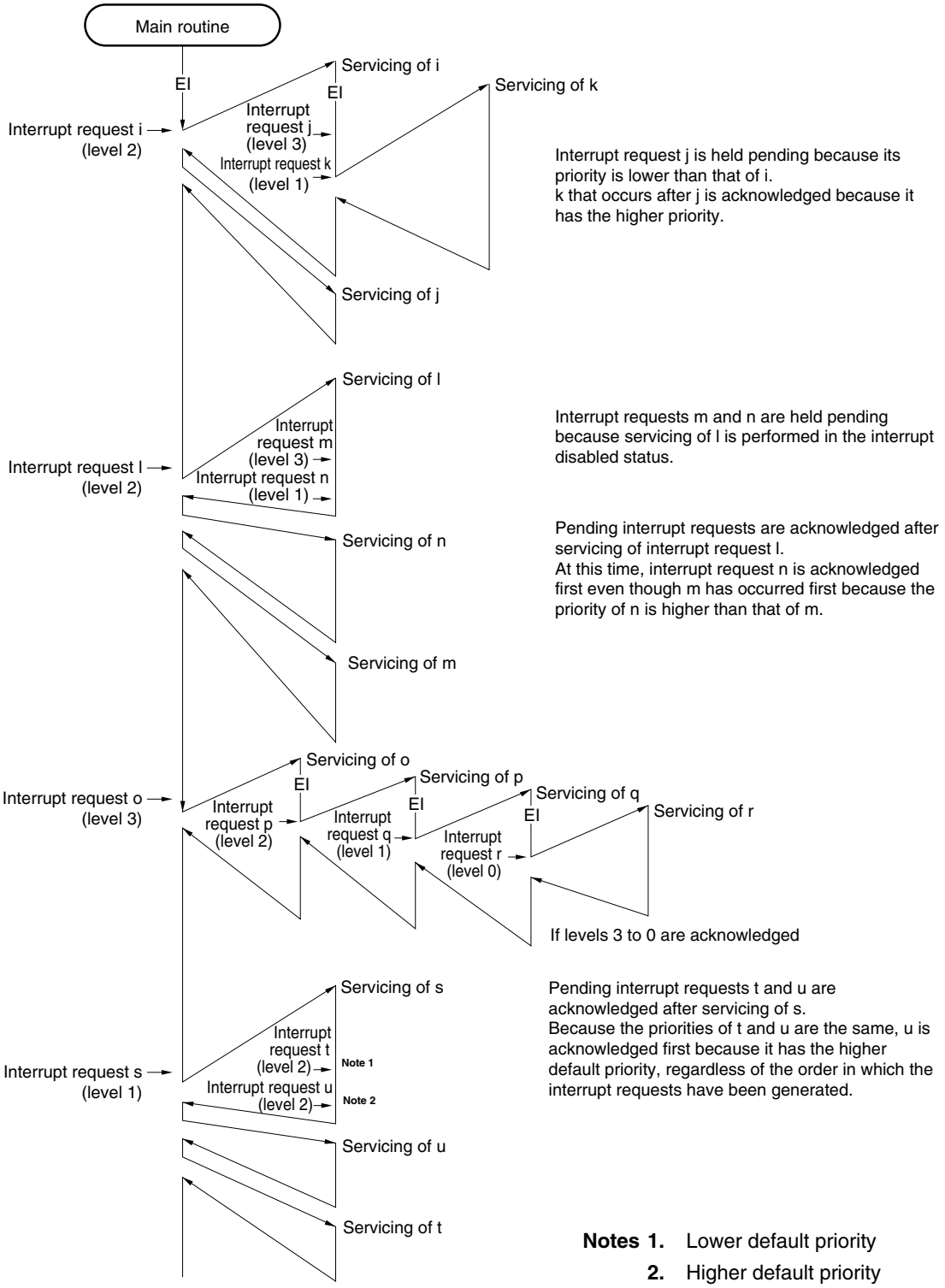


Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Remarks 1. a to u in the figure are the temporary names of interrupt request signals shown for the sake of explanation.

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

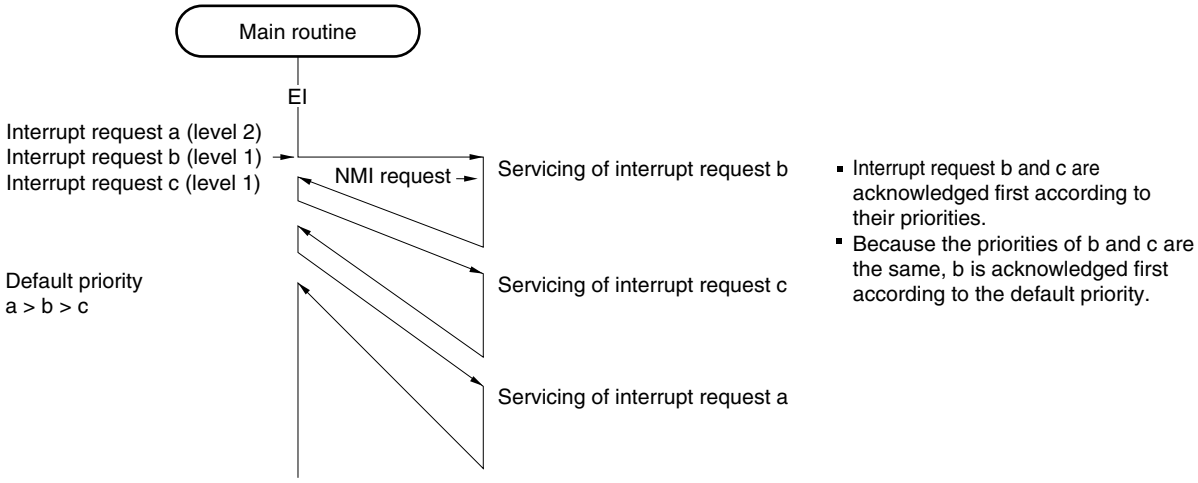
Figure 20-6. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (2/2)



Notes 1. Lower default priority
2. Higher default priority

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

Figure 20-7. Example of Servicing Interrupt Request Signals Simultaneously Generated



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

Remarks 1. a to c in the figure are the temporary names of interrupt request signals shown for the sake of explanation.

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

20.3.4 Interrupt control register (xxICn)

The xxICn register is assigned to each maskable interrupt request signal and sets the control conditions for each maskable interrupt request.

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

Reset input sets this register to 47H.

Caution Be sure to read the xxICn.xxIFn bit while interrupts are disabled (DI). If the xxIFn bit is read while interrupts are enabled (EI), an incorrect value may be read if there is a conflict between acknowledgment of the interrupt and reading of the bit.

After reset: 47H R/W Address: FFFFF110H to FFFFF178H

	<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 signal not generated
1	Interrupt request signal generated

xxMKn	Interrupt mask flag
0	Enable interrupt servicing
1	Disable interrupt servicing (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 Automatically reset by hardware when interrupt request signal is acknowledged.

Remark xx: Identifying name of each peripheral unit (see **Table 20-2**)
n: Peripheral unit number (see **Tables 20-2**)

The following tables list the addresses and bits of the interrupt control registers.

Table 20-2. Addresses and Bits of Interrupt Control Registers (1/2)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF110H	WDTIC	WDTIF	WDTMK	0	0	0	WDTPR2	WDTPR1	WDTPR0
FFFFF112H	P00IC0	P00IF0	P00MK0	0	0	0	P00PR02	P00PR01	P00PR00
FFFFF114H	P00IC1	P00IF1	P00MK1	0	0	0	P00PR12	P00PR11	P00PR10
FFFFF116H	P00IC4	P00IF4	P00MK4	0	0	0	P00PR42	P00PR41	P00PR40
FFFFF118H	P00IC5	P00IF5	P00MK5	0	0	0	P00PR52	P00PR51	P00PR50
FFFFF11AH	P10IC6	P10IF6	P10MK6	0	0	0	P10PR62	P10PR61	P10PR60
FFFFF11CH	P10IC7	P10IF7	P10MK7	0	0	0	P10PR72	P10PR71	P10PR70
FFFFF11EH	P01IC0	P01IF0	P01MK0	0	0	0	P01PR02	P01PR01	P01PR00
FFFFF120H	P01IC1	P01IF1	P01MK1	0	0	0	P01PR12	P01PR11	P01PR10
FFFFF122H	P01IC2	P01IF2	P01MK2	0	0	0	P01PR22	P01PR21	P01PR20
FFFFF124H	P01IC3	P01IF3	P01MK3	0	0	0	P01PR32	P01PR31	P01PR30
FFFFF126H	P11IC4	P11IF4	P11MK4	0	0	0	P11PR42	P11PR41	P11PR40
FFFFF128H	P11IC5	P11IF5	P11MK5	0	0	0	P11PR52	P11PR51	P11PR50
FFFFF12AH	P02IC1	P02IF1	P02MK1	0	0	0	P02PR12	P02PR11	P02PR10
FFFFF12CH	P02IC2	P02IF2	P02MK2	0	0	0	P02PR22	P02PR21	P02PR20
FFFFF12EH	P12IC4	P12IF4	P12MK4	0	0	0	P12PR42	P12PR41	P12PR40
FFFFF130H	P12IC5	P12IF5	P12MK5	0	0	0	P12PR52	P12PR51	P12PR50
FFFFF132H	P12IC6	P12IF6	P12MK6	0	0	0	P12PR62	P12PR61	P12PR60
FFFFF134H	P13IC0	P13IF0	P13MK0	0	0	0	P13PR02	P13PR01	P13PR00
FFFFF136H	P13IC1	P13IF1	P13MK1	0	0	0	P13PR12	P13PR11	P13PR10
FFFFF138H	P13IC2	P13IF2	P13MK2	0	0	0	P13PR22	P13PR21	P13PR20
FFFFF13AH	P13IC3	P13IF3	P13MK3	0	0	0	P13PR32	P13PR31	P13PR30
FFFFF13CH	P13IC4	P13IF4	P13MK4	0	0	0	P13PR42	P13PR41	P13PR40
FFFFF13EH	P13IC7	P13IF7	P13MK7	0	0	0	P13PR72	P13PR71	P13PR70
FFFFF140H	P05IC0	P05IF0	P05MK0	0	0	0	P05PR02	P05PR01	P05PR00
FFFFF142H	P05IC1	P05IF1	P05MK1	0	0	0	P05PR12	P05PR11	P05PR10
FFFFF144H	CMICD0	CMIF0	CMMK0	0	0	0	CMPR02	CMPR01	CMPR00
FFFFF146H	CMICD1	CMIF1	CMMK1	0	0	0	CMPR12	CMPR11	CMPR10
FFFFF148H	CMICD2	CMIF2	CMMK2	0	0	0	CMPR22	CMPR21	CMPR20
FFFFF14AH	CMICD3	CMIF3	CMMK3	0	0	0	CMPR32	CMPR31	CMPR30
FFFFF14CH	CM10IC0	CM10IF0	CM10MK0	0	0	0	CM10PR02	CM10PR01	CM10PR00
FFFFF14EH	CM10IC1	CM10IF1	CM10MK1	0	0	0	CM10PR12	CM10PR11	CM10PR10
FFFFF150H	OVPIC0	OVPIF0	OVPMK0	0	0	0	OVPPR02	OVPPR01	OVPPR00
FFFFF152H	OVQIC	OVQIF	OVQMK	0	0	0	OVQPR2	OVQPR1	OVQPR0
FFFFF154H	OVPIC1	OVPIF1	OVPMK1	0	0	0	OVPPR12	OVPPR11	OVPPR10
FFFFF156H	OVPIC2	OVPIF2	OVPMK2	0	0	0	OVPPR22	OVPPR21	OVPPR20
FFFFF158H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00

Table 20-2. Addresses and Bits of Interrupt Control Registers (2/2)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF15AH	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF15CH	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFF15EH	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30
FFFFF160H	SEIC0	SEIF0	SEMK0	0	0	0	SEPR02	SEPR01	SEPR00
FFFFF162H	SRIC0	SRIF0	SRMK0	0	0	0	SRPR02	SRPR01	SRPR00
FFFFF164H	STIC0	STIF0	STMK0	0	0	0	STPR02	STPR01	STPR00
FFFFF166H	SEIC1	SEIF1	SEMK1	0	0	0	SEPR12	SEPR11	SEPR10
FFFFF168H	SRIC1	SRIF1	SRMK1	0	0	0	SRPR12	SRPR11	SRPR10
FFFFF16AH	STIC1	STIF1	STMK1	0	0	0	STPR12	STPR11	STPR10
FFFFF16CH	SEIC2	SEIF2	SEMK2	0	0	0	SEPR22	SEPR21	SEPR20
FFFFF16EH	SRIC2	SRIF2	SRMK2	0	0	0	SRPR22	SRPR21	SRPR20
FFFFF170H	STIC2	STIF2	STMK2	0	0	0	STPR22	STPR21	STPR20
FFFFF172H	SEIC3	SEIF3	SEMK3	0	0	0	SEPR32	SEPR31	SEPR30
FFFFF174H	SRIC3	SRIF3	SRMK3	0	0	0	SRPR32	SRPR31	SRPR30
FFFFF176H	STIC3	STIF3	STMK3	0	0	0	STPR32	STPR31	STPR30
FFFFF178H	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0

20.3.5 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 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 input 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	1	1	1	1
	7	6	5	4	3	2	1	0
(IMR3L)	1	1	1	ADMK	STMK3	SRMK3	SEMK3	STMK2

After reset: FFFFH R/W Address: IMR2 FFFFF104H
IMR2L FFFFF104H, IMR2H FFFFF105H

	15	14	13	12	11	10	9	8
IMR2 (IMR2H[Note])	SRMK2	SEMK2	STMK1	SRMK1	SEMK1	STMK0	SRMK0	SEMK0
	7	6	5	4	3	2	1	0
(IMR2L)	DMAMK3	DMAMK2	DMAMK1	DMAMK0	OVPMK2	OVPMK1	OVQMK	OVPMK0

After reset: FFFFH R/W Address: IMR1 FFFFF102H
IMR1L FFFFF102H, IMR1H FFFFF103H

	15	14	13	12	11	10	9	8
IMR1 (IMR1H[Note])	CM10MK1	CM10MK0	CMMK3	CMMK2	CMMK1	CMMK0	P05MK1	P05MK0
	7	6	5	4	3	2	1	0
(IMR1L)	P13MK7	P13MK4	P13MK3	P13MK2	P13MK1	P13MK0	P12MK6	P12MK5

After reset: FFFFH R/W Address: IMR0 FFFFF100H
IMR0L FFFFF100H, IMR0H FFFFF101H

	15	14	13	12	11	10	9	8
IMR0 (IMR0H[Note])	P12MK4	P02MK2	P02MK1	P11MK5	P11MK4	P01MK3	P01MK2	P01MK1
	7	6	5	4	3	2	1	0
(IMR0L)	P01MK0	P10MK7	P10MK6	P00MK5	P00MK4	P00MK1	P00MK0	WDTMK

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 15 to 5 of the IMR3 register (bits 7 to 0 of the IMR3H register and bits 7 to 5 of the IMR3L register) to 1. If these bits are not 1, the operation cannot be guaranteed.

Remark xx: Identification name of each peripheral unit (see **Table 20-2**)
n: Peripheral unit number (see **Table 20-2**)

20.3.6 In-service priority register (ISPR)

The ISPR register holds the priority level of the maskable interrupt currently acknowledged. When an interrupt request signal is acknowledged, the bit of this register corresponding to the priority level of that interrupt request signal 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 reset to 0 by hardware. However, it is not reset to 0 when execution is returned from non-maskable interrupt servicing or exception processing.

This register is read-only, in 8-bit or 1-bit units.

Reset input clears this register to 00H.

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

After reset: 00H R Address: FFFFF1FAH

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0

ISPRn	Priority of interrupt currently being acknowledged
0	Interrupt request signal with priority n is not acknowledged
1	Interrupt request signal with priority n is being acknowledged

Remark n: 0 to 7 (priority level)

20.3.7 Maskable interrupt status flag (ID)

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

This flag is set to 00000020H after reset.

After rest: 00000020H

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

ID	Maskable interrupt servicing specification[Note]
0	Maskable interrupt request signal acknowledgment enabled
1	Maskable interrupt request signal acknowledgment disabled (pending)

Note Interrupt disable flag (ID) function

ID is set (1) by the DI instruction and cleared (0) by the EI instruction. Its value is also modified by the RETI instruction or LDSR instruction when referencing the PSW.

Non-maskable interrupts and exceptions are acknowledged regardless of this flag. When a maskable interrupt is acknowledged, the ID flag is automatically set (1) by hardware.

An interrupt request signal generated during the acknowledgment disabled period (ID flag = 1) can be acknowledged when the xxICn.xxIFn bit is set (1), and the ID flag is cleared (0).

20.4 External Interrupt Request Input Pins (NMI, $\overline{\text{INTPn}}$)

20.4.1 Noise elimination

(1) Noise elimination of NMI pin

NMI pin incorporates a noise eliminator using analog delay. The delay time is Typ. 80 ns. A signal input that changes within the delay time is not internally acknowledged.

(2) Noise elimination of $\overline{\text{INTPn}}$ pin

The $\overline{\text{INTPn}}$ pin incorporates a noise eliminator using analog delay (n = 000, 001, 004, 005, 010 to 013, 021, 022, 050, 051, 106, 107, 114, 115, 124 to 126, 130 to 134, and 137). The delay time is Typ. 80 ns. A signal input that changes within the delay time is not internally acknowledged.

20.4.2 Edge detection

The valid edge of the NMI and $\overline{\text{INTPn}}$ pins can be selected by program (n = 000, 001, 004, 005, 010 to 013, 021, 022, 050, 051, 106, 107, 114, 115, 124 to 126, 130 to 134, and 137). Moreover, a level trigger can be selected for the $\overline{\text{INTPn}}$ pin. 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 NMI and INTPn signals become interrupt sources.

The valid edge and level trigger are specified by NMIR, NMIF, INTR0 to INTR3, INTR5, INTF0 to INTF3, and INTF5 registers.

Caution To change the mode from the external interrupt input ($\overline{\text{INTPn}}$) mode to the port mode or other alternate functions, mask the external interrupt input by using the xxICa.xxMKa bit.

Remark xx: Identifying name of each peripheral unit (See **Table 20-2**)
a: Peripheral unit number (See **Table 20-2**)

(1) NMI rising edge specification register (NMIR), NMI falling edge specification register (NMIF)

The NMIR and NMIF registers are registers that specify the valid edge of an NMI pin. The NMI valid edge can be specified to be either the rising edge, the falling edge, or both the rising and falling edges.

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

Reset input clears these registers to 00H.

After reset: 00H R/W Address: FFFFFC3EH

	7	6	5	4	3	2	1	<0>
NMIR	0	0	0	0	0	0	0	NMIR0

After reset: 00H R/W Address: FFFFFC1EH

	7	6	5	4	3	2	1	<0>
NMIF	0	0	0	0	0	0	0	NMIF0

Remark For how to specify a valid edge, see **Table 20-3**.

Table 20-3. Valid Edge Specification

NMIF0	NMIR0	Valid Edge Specification
0	0	Falling edge
0	1	Rising edge
1	0	Setting prohibited
1	1	Both rising and falling edges

(2) 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 $\overline{\text{INTP000}}$, $\overline{\text{INTP001}}$, $\overline{\text{INTP004}}$, $\overline{\text{INTP005}}$, $\overline{\text{INTP106}}$, and $\overline{\text{INTP107}}$ 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 input clears the INTR0 register to 00H and sets the INTF0 register to F3H.

Caution Before setting the $\overline{\text{INTP000}}$, $\overline{\text{INTP001}}$, $\overline{\text{INTP004}}$, $\overline{\text{INTP005}}$, $\overline{\text{INTP106}}$, or $\overline{\text{INTP107}}$ pin in the trigger mode, set the PMC0 register.
If the PMC0 register is set after the INTR0 and INTF0 registers have been set, an illegal interrupt may occur, depending on the timing of setting the PMC0 register.

After reset: 00H R/W Address: FFFFFC20H

	<7>	<6>	<5>	<4>	3	2	<1>	<0>
INTR0	INTR07	INTR06	INTR05	INTR04	0	0	INTR01	INTR00

After reset: F3H R/W Address: FFFFFC00H

	<7>	<6>	<5>	<4>	3	2	<1>	<0>
INTF0	INTF07	INTF06	INTF05	INTF04	0	0	INTF01	INTF00

Remark For the valid edge specification, see **Table 20-4**.

Table 20-4. Valid Edge Specification of $\overline{\text{INTP000}}$, $\overline{\text{INTP001}}$, $\overline{\text{INTP004}}$, $\overline{\text{INTP005}}$, $\overline{\text{INTP106}}$, and $\overline{\text{INTP107}}$ Pins

INTF0n	INTR0n	Valid Edge Specification (n = 0, 1, 4 to 7)
0	0	Level detection (low level detection)[Note]
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Note When a lower priority level-detection interrupt request (INTPm) occurs while another interrupt is being serviced and this newly generated level-detection interrupt request becomes inactive before the current interrupt servicing is complete, this new interrupt request (INTPm) is held pending (m = 000, 001, 004, 005, 106, 107). To avoid acknowledging this INTPm interrupt request, clear the Px0IFn bit of the interrupt control register.

Remark n = 0, 1, 4 to 7: Control of $\overline{\text{INTP000}}$, $\overline{\text{INTP001}}$, $\overline{\text{INTP004}}$, $\overline{\text{INTP005}}$, $\overline{\text{INTP106}}$, and $\overline{\text{INTP107}}$ pins

(3) External interrupt rising edge specification register 1 (INTR1), external interrupt falling edge specification register 1 (INTF1)

The INTR1 and INTF1 registers are 8-bit registers that specify the trigger mode of the $\overline{\text{INTP010}}$ to $\overline{\text{INTP013}}$, $\overline{\text{INTP114}}$, and $\overline{\text{INTP115}}$ 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 input clears the INTR1 register to 00H and sets the INTF1 register to 3FH.

Caution Before setting the $\overline{\text{INTP010}}$ to $\overline{\text{INTP013}}$, $\overline{\text{INTP114}}$, or $\overline{\text{INTP115}}$ pin in the trigger mode, set the PMC1 register.

If the PMC1 register is set after the INTR1 and INTF1 registers have been set, an illegal interrupt may occur, depending on the timing of setting the PMC1 register.

After reset: 00H R/W Address: FFFFFC22H

	7	6	<5>	<4>	<3>	<2>	<1>	<0>
INTR1	0	0	INTR15	INTR14	INTR13	INTR12	INTR11	INTR10

After reset: 3FH R/W Address: FFFFFC02H

	7	6	<5>	<4>	<3>	<2>	<1>	<0>
INTF1	0	0	INTF15	INTF14	INTF13	INTF12	INTF11	INTF10

Remark For the valid edge specification, see **Table 20-5**.

Table 20-5. Valid Edge Specification of $\overline{\text{INTP010}}$ to $\overline{\text{INTP013}}$, $\overline{\text{INTP114}}$, and $\overline{\text{INTP115}}$ Pins

INTF1n	INTR1n	Valid Edge Specification (n = 0 to 5)
0	0	Level detection (low level detection)[Note]
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Note When a lower priority level-detection interrupt request (INTPm) occurs while another interrupt is being serviced and this newly generated level-detection interrupt request becomes inactive before the current interrupt servicing is complete, this new interrupt request (INTPm) is held pending (m = 010 to 013, 114, 115). To avoid acknowledging this INTPm interrupt request, clear the Px1IFn bit of the interrupt control register.

Remark n = 0 to 5: Control of $\overline{\text{INTP010}}$ to $\overline{\text{INTP013}}$, $\overline{\text{INTP114}}$, and $\overline{\text{INTP115}}$ pins

(4) External interrupt rising edge specification register 2 (INTR2), external interrupt falling edge specification register 2 (INTF2)

The INTR2 and INTF2 registers are 8-bit registers that specify the trigger mode of the $\overline{\text{INTP021}}$, $\overline{\text{INTP022}}$, and $\overline{\text{INTP124}}$ to $\overline{\text{INTP126}}$ 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 input clears the INTR2 register to 00H and sets the INTF2 register to 76H.

Caution Before setting the $\overline{\text{INTP021}}$, $\overline{\text{INTP022}}$, or $\overline{\text{INTP124}}$ to $\overline{\text{INTP126}}$ pin in the trigger mode, set the PMC2 register.
If the PMC2 register is set after the INTR2 and INTF2 registers have been set, an illegal interrupt may occur, depending on the timing of setting the PMC2 register.

After reset: 00H R/W Address: FFFFFC24H

	7	<6>	<5>	<4>	3	<2>	<1>	0
INTR2	0	INTR26	INTR25	INTR24	0	INTR22	INTR21	0

After reset: 76H R/W Address: FFFFFC04H

	7	<6>	<5>	<4>	3	<2>	<1>	0
INTF2	0	INTF26	INTF25	INTF24	0	INTF22	INTF21	0

Remark For the valid edge specification, see **Table 20-6**.

Table 20-6. Valid Edge Specification of $\overline{\text{INTP021}}$, $\overline{\text{INTP022}}$, and $\overline{\text{INTP024}}$ to $\overline{\text{INTP126}}$ Pins

INTF2n	INTR2n	Valid Edge Specification (n = 1, 2, 4 to 6)
0	0	Level detection (low level detection)[Note]
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Note When a lower priority level-detection interrupt request (INTPm) occurs while another interrupt is being serviced and this newly generated level-detection interrupt request becomes inactive before the current interrupt servicing is complete, this new interrupt request (INTPm) is held pending (m = 021, 022, 124 to 126). To avoid acknowledging this INTPm interrupt request, clear the Px2IFn bit of the interrupt control register.

Remark n = 1, 2, 4 to 6: Control of $\overline{\text{INTP021}}$, $\overline{\text{INTP022}}$, and $\overline{\text{INTP124}}$ to $\overline{\text{INTP126}}$ pins

(5) External interrupt rising edge specification register 3 (INTR3), external interrupt falling edge specification register 3 (INTF3)

The INTR3 and INTF3 registers are 8-bit registers that specify the trigger mode of the $\overline{\text{INTP130}}$ to $\overline{\text{INTP134}}$ and $\overline{\text{INTP137}}$ 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 input clears the INTR3 register to 00H and sets the INTF3 register to 9FH.

Caution Before setting the $\overline{\text{INTP130}}$ to $\overline{\text{INTP134}}$ or $\overline{\text{INTP137}}$ pin in the trigger mode, set the PMC3 register.

If the PMC3 register is set after the INTR3 and INTF3 registers have been set, an illegal interrupt may occur, depending on the timing of setting the PMC3 register.

After reset: 00H R/W Address: FFFFFC26H

	<7>	6	5	<4>	<3>	<2>	<1>	<0>
INTR3	INTR37	0	0	INTR34	INTR33	INTR32	INTR31	INTR30

After reset: 9FH R/W Address: FFFFFC06H

	<7>	6	5	<4>	<3>	<2>	<1>	<0>
INTF3	INTF37	0	0	INTF34	INTF33	INTF32	INTF31	INTF30

Remark For the valid edge specification, see **Table 20-7**.

Table 20-7. Valid Edge Specification of $\overline{\text{INTP130}}$ to $\overline{\text{INTP134}}$, and $\overline{\text{INTP137}}$ Pins

INTF3n	INTR3n	Valid Edge Specification (n = 0 to 4, 7)
0	0	Level detection (low level detection)[Note]
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Note When a lower priority level-detection interrupt request (INTPm) occurs while another interrupt is being serviced and this newly generated level-detection interrupt request becomes inactive before the current interrupt servicing is complete, this new interrupt request (INTPm) is held pending (m = 130 to 134, 137). To avoid acknowledging this INTPm interrupt request, clear the P13IFn bit of the interrupt control register.

Remark n = 0 to 4, 7: Control of $\overline{\text{INTP130}}$ to $\overline{\text{INTP134}}$, and $\overline{\text{INTP137}}$ pins

(6) External interrupt rising edge specification register 5 (INTR5), external interrupt falling edge specification register 5 (INTF5)

The INTR5 and INTF5 registers are 8-bit registers that specify the trigger mode of the $\overline{\text{INTP050}}$ and $\overline{\text{INTP051}}$ 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 input clears the INTR5 register to 00H and sets the INTF5 register to 03H.

Caution Before setting the $\overline{\text{INTP050}}$ or $\overline{\text{INTP051}}$ pin in the trigger mode, set the PMC5 register. If the PMC5 register is set after the INTR5 and INTF5 registers have been set, an illegal interrupt may occur, depending on the timing of setting the PMC5 register.

After reset: 00H R/W Address: FFFFFC2AH

	7	6	5	4	3	2	<1>	<0>
INTR5	0	0	0	0	0	0	INTR51	INTR50

After reset: 03H R/W Address: FFFFFC0AH

	7	6	5	4	3	2	<1>	<0>
INTF5	0	0	0	0	0	0	INTF51	INTF50

Remark For the valid edge specification, see **Table 20-8**.

Table 20-8. Valid Edge Specification of $\overline{\text{INTP050}}$ and $\overline{\text{INTP051}}$ Pins

INTF5n	INTR5n	Valid Edge Specification (n = 0, 1)
0	0	Level detection (low level detection)[Note]
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Note When a lower priority level-detection interrupt request (INTPm) occurs while another interrupt is being serviced and this newly generated level-detection interrupt request becomes inactive before the current interrupt servicing is complete, this new interrupt request (INTPm) is held pending (m = 050, 051). To avoid acknowledging this INTPm interrupt request, clear the P05IFn bit of the interrupt control register.

Remark n = 0, 1: Control of $\overline{\text{INTP050}}$ and $\overline{\text{INTP051}}$ pins

20.5 Software Exception

A software exception is generated when the CPU executes the TRAP instruction, and can always be acknowledged.

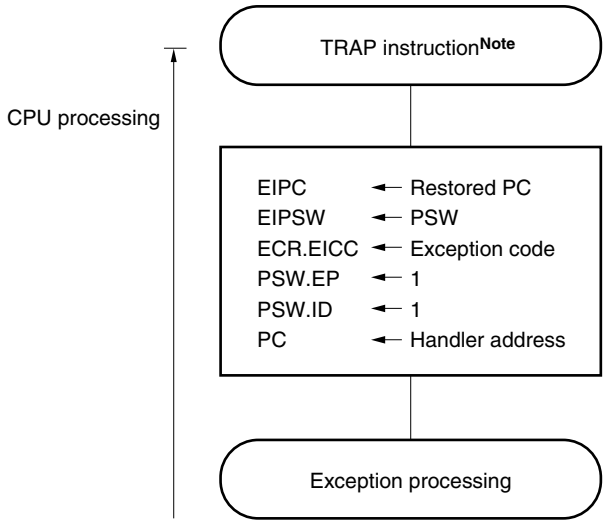
20.5.1 Operation

If a software exception occurs, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower 16 bits (EICC) of ECR (interrupt source).
<4> Sets the PSW.EP and PSW.ID bits to 1
<5> Sets the handler address (00000040H or 00000050H) corresponding to the software exception to the PC, and transfers control.

The following illustrates the processing of a software exception.

Figure 20-8. Software Exception Processing



Note TRAP instruction format: TRAP vector (the vector is a value from 00H to 1FH.)

The handler address is determined by the TRAP instruction's operand (vector). If the vector is 00H to 0FH, it becomes 00000040H, and if the vector is 10H to 1FH, it becomes 00000050H.

20.5.2 Restore

Recovery from software exception processing is carried out by the RETI instruction.

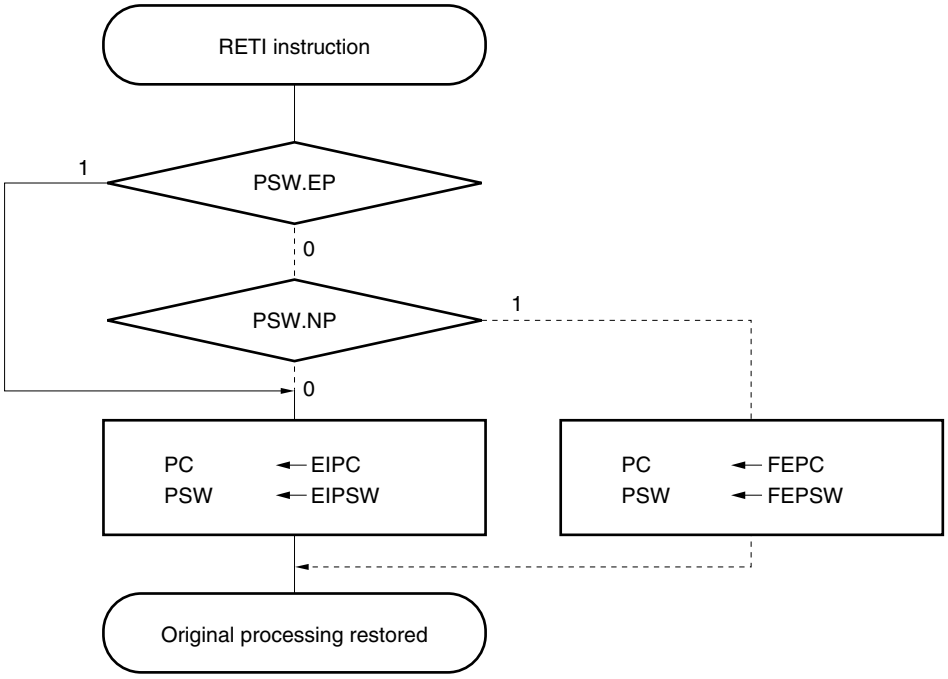
By executing the RETI instruction, the CPU carries out the following processing and shifts control to the restored PC's address.

<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 following illustrates the processing of the RETI instruction.

Figure 20-9. RETI Instruction Processing



Caution When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during the software exception processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set the EP bit back to 1 and clear the NP bit to 0 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

20.5.3 Exception status flag (EP)

The EP flag is a status flag used to indicate that exception processing is in progress. It is set when an exception occurs.

The EP flag is allocated to the PSW.

This flag is set to 00000020H after reset.

After reset: 00000020H

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

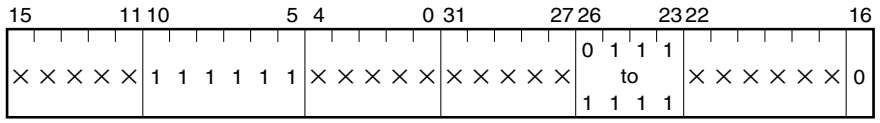
EP	Exception processing status
0	Exception processing not in progress
1	Exception processing in progress

20.6 Exception Trap

An exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. In the V850E/MA3, an illegal opcode exception (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

20.6.1 Illegal opcode

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.

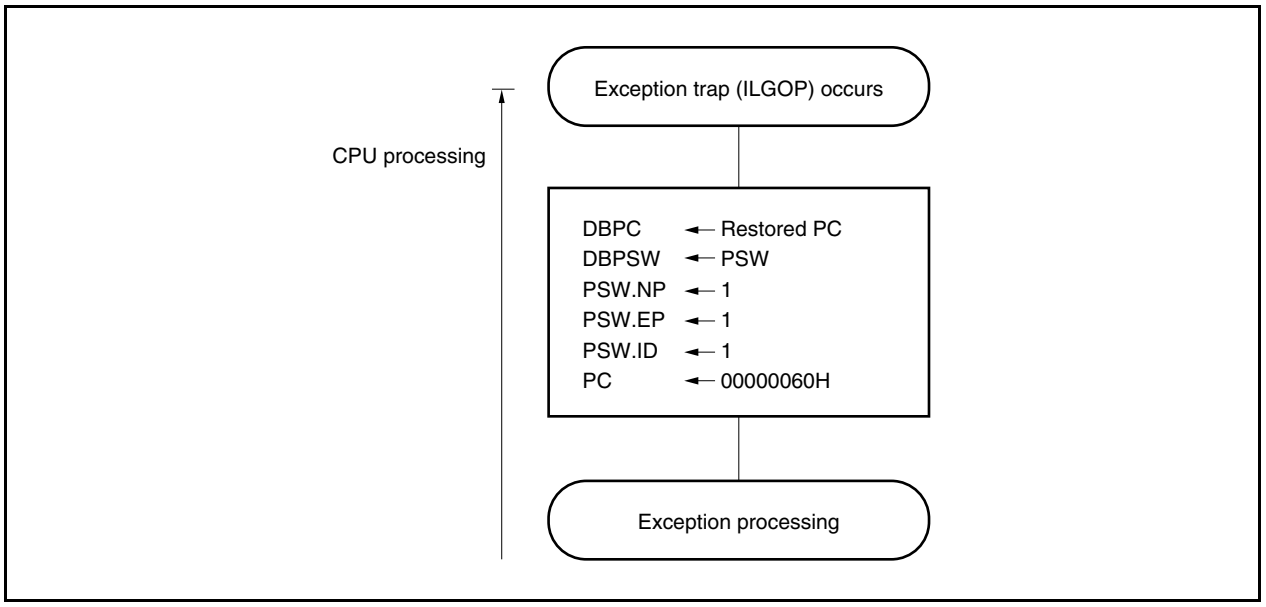
(1) Operation

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 to 1.
<4> Sets the handler address (00000060H) corresponding to the exception trap to the PC, and transfers control.

The following illustrates the processing of the exception trap.

Figure 20-10. Exception Trap Processing



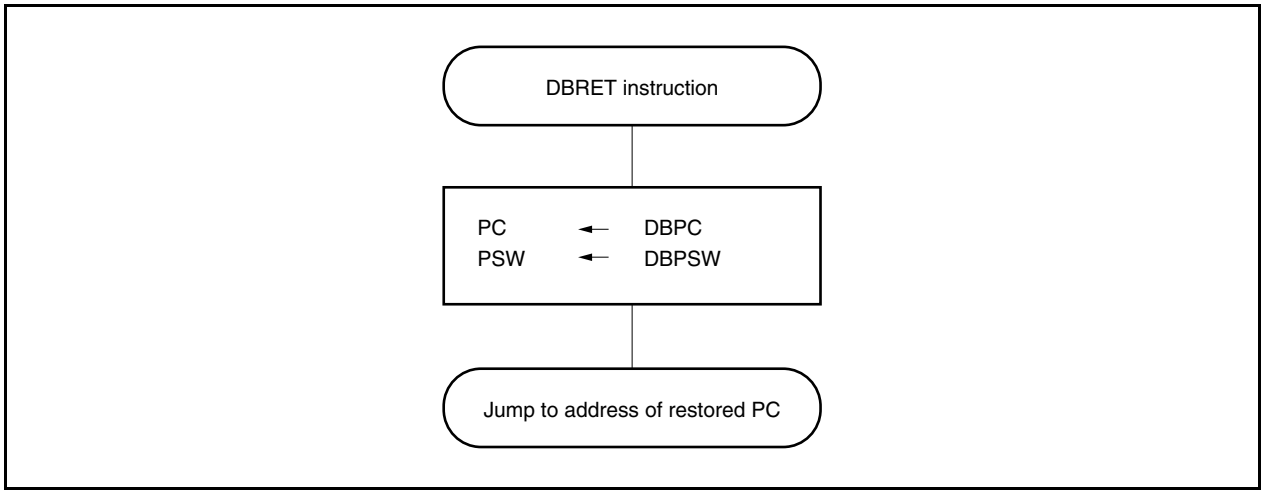
(2) Restore

Recovery from an exception trap is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

<1> Loads the restored PC and PSW from DBPC and DBPSW.
<2> Transfers control to the address indicated by the restored PC and PSW.

The following illustrates the restore processing from an exception trap.

Figure 20-11. Restore Processing from Exception Trap



20.6.2 Debug trap

The debug trap is an exception that can be acknowledged anytime and is generated by execution of the DBTRAP instruction.

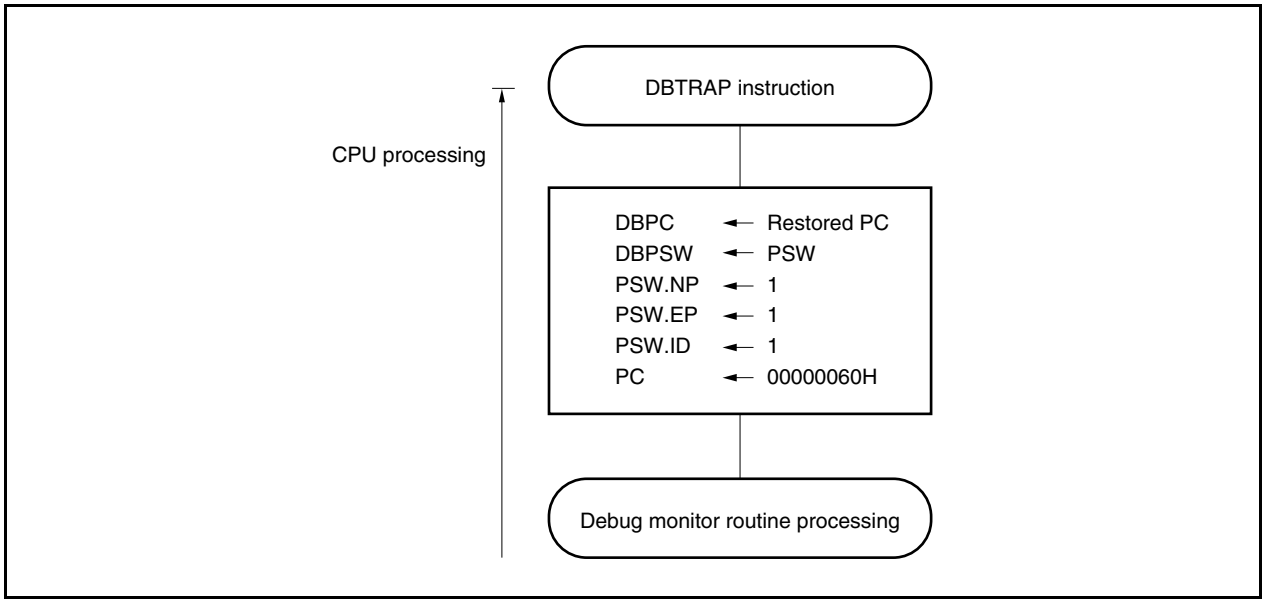
When the debug trap is generated, the CPU performs the following processing.

(1) Operation

<1> Saves the restored PC to DBPC.
<2> Saves the current PSW to DBPSW.
<3> Sets the PSW.NP, PSW.EP and PSW.ID bits to 1.
<4> Sets the handler address (00000060H) corresponding to the debug trap to the PC and transfers control.

The following illustrates the processing of the debug trap.

Figure 20-12. Debug Trap Processing



(2) Restore

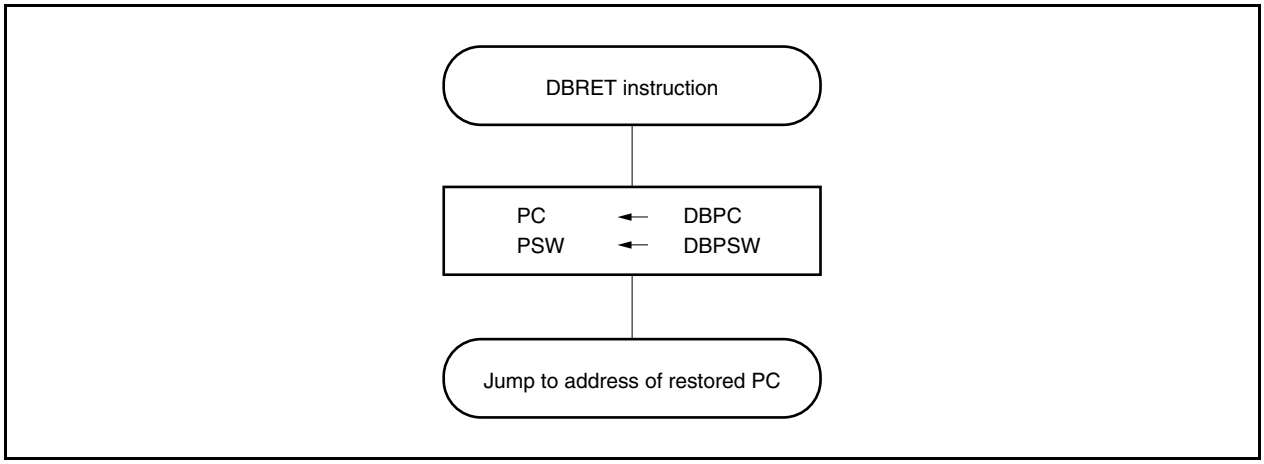
Recovery from a debug trap is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

<1> Loads the restored PC and PSW from DBPC and DBPSW.
<2> Transfers control to the address indicated by the restored PC and PSW.

Caution DBPC and DBPSW can be accessed during the period between when the DBTRAP instruction is executed and when the DBRET instruction is executed.

The following illustrates the restore processing from a debug trap.

Figure 20-13. Restore Processing from Debug Trap



20.7 Multiple Interrupt Servicing Control

Multiple interrupt servicing control is a process by which an interrupt request signal 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 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 (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 request signal 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.

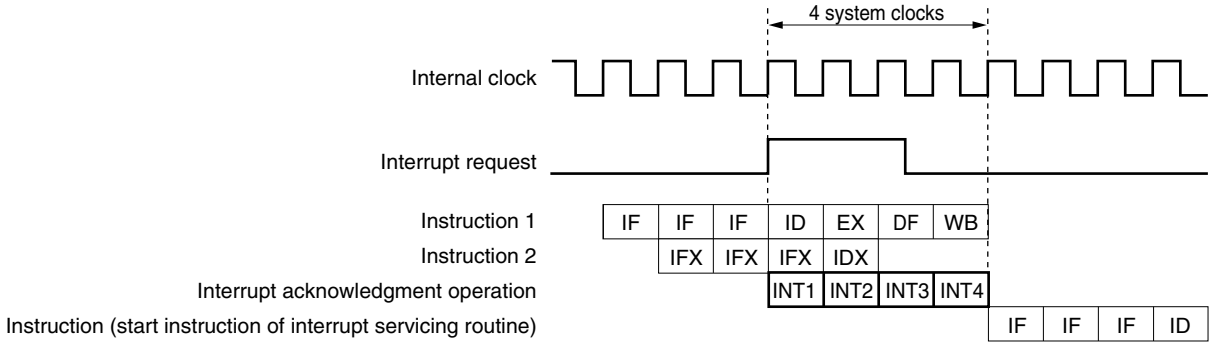
Caution In a non-maskable interrupt servicing routine (time until the RETI instruction is executed), maskable interrupts are suspended and not acknowledged.

Remark xx: Identification name of each peripheral unit (see **Table 20-2**)
n: Peripheral unit number (see **Table 20-2**)

20.8 Interrupt Latency Time

The V850E/MA3 interrupt latency time (from interrupt request generation to start of interrupt servicing) is described below.

Figure 20-14. Pipeline Operation at Interrupt Request Acknowledgment (Outline)



Remark INT1 to INT4: Interrupt acknowledgment processing
IFX: Invalid instruction fetch
IDX: Invalid instruction decode

Interrupt latency time (internal system clock)			Condition
	Internal interrupt	External interrupt INTPn	
Minimum	4	4 + Analog delay time	The following cases are exceptions. • In IDLE/software STOP mode • External bus access • Two or more interrupt request non-sample instructions are executed in succession • Access to on-chip peripheral I/O register
Maximum	8	8 + Analog delay time	

Remark n = 000, 001, 004, 005, 010 to 013, 021, 022, 050, 051, 106, 107, 114, 115, 124 to 126, 130 to 134, 137

20.9 Periods in Which CPU Does Not Acknowledge Interrupts

The CPU acknowledges an interrupt while an instruction is being executed. However, no interrupt will be acknowledged between an interrupt request non-sample instruction and the next instruction (interrupt is held pending).

The interrupt request non-sample instructions are as follows.

- EI instruction
- DI instruction
- LDSR reg2, 0x5 instruction (for PSW)
- The store instruction for the command register (PRCMD)
- The store, SET1, NOT1, and CLR1 instructions for the following registers.
  - Interrupt-related registers:
    Interrupt control register (xxICn), interrupt mask registers 0 to 3 (IMR0 to IMR3)
  - Power save control register (PSC)

20.10 Cautions

(1) Caution on INTWDT

If the non-maskable interrupt (INTWDT) is generated because WDT overflows, execute reset processing in the interrupt routine. Note that the RETI instruction cannot be executed at this time.

(2) Caution when port is used as external interrupt input pin ($\overline{\text{INTPn}}$)

When a port is used as external interrupt input pins ($\overline{\text{INTPn}}$), note that the interrupts related to the timer/counter and serial interface that are multiplexed with the port are not generated (n = 000, 001, 004, 005, 010 to 013, 021, 022, 050, 051, 106, 107, 114, 115, 124 to 126, 130 to 134, 137).

CHAPTER 21 STANDBY FUNCTION

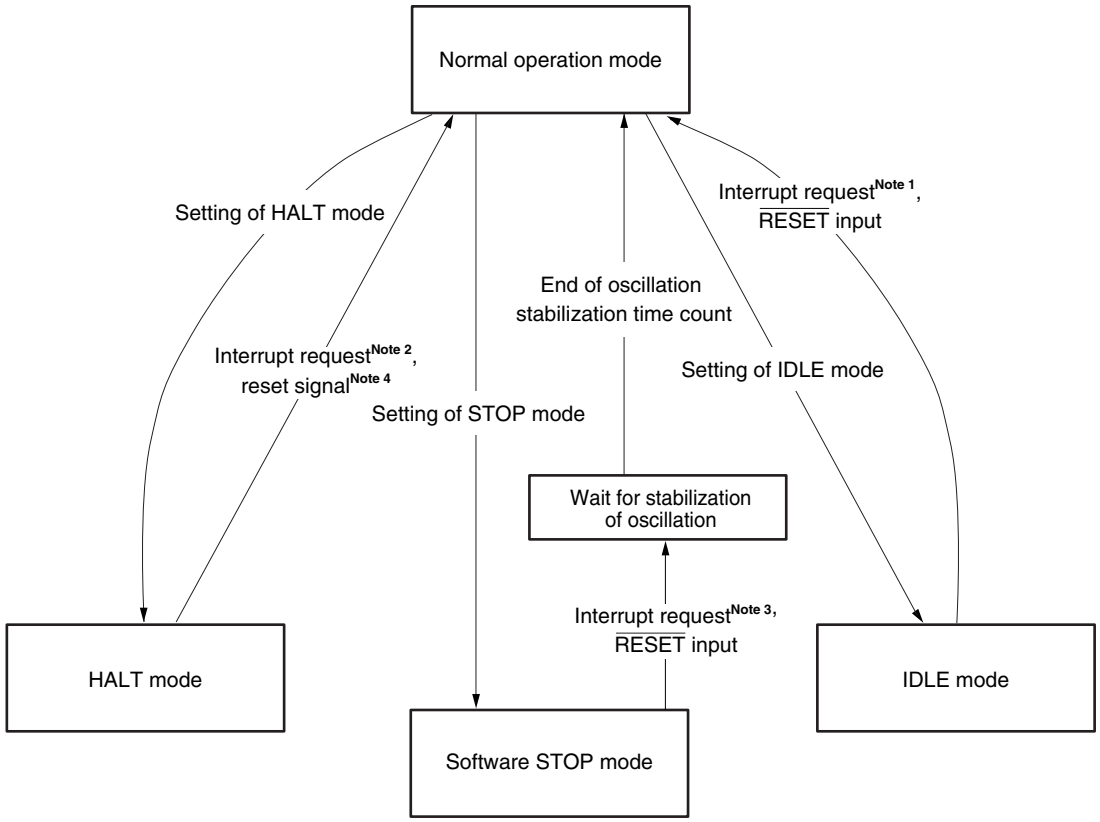
21.1 Overview

The power consumption of the system can be effectively reduced by using the standby modes in combination and selecting the appropriate mode for the application. The available standby modes are listed in Table 21-1.

Table 21-1. Standby Modes

Mode	Functional Outline
HALT mode	Mode to stop only the operating clock of the CPU
IDLE mode	Mode to stop all the operations of the internal circuit except the oscillator, CSIB in the slave mode, and UARTA when external clock is selected
Software STOP mode	Mode to stop all the operations of the internal circuit except CSIB in the slave mode and UARTA when external clock is selected

Figure 21-1. Status Transition



Notes 1. Non-maskable interrupt request signal (NMI pin input), unmasked external interrupt request Note 5, or unmasked internal interrupt request signal (CSIB-related interrupt request signal in the slave mode and UARTA-related interrupt request signal when external clock is selected) from peripheral functions operable in IDLE mode.

2. Non-maskable interrupt request signal (NMI pin input, non-maskable interrupt request signal (INTWDT) generation by overflow) or unmasked maskable interrupt request signal.

3. Non-maskable interrupt request signal (NMI pin input), unmasked external interrupt request signal Note 5, or unmasked internal interrupt request signal (CSIB-related interrupt request signal in the slave mode and UARTA-related interrupt request signal when external clock is selected) from peripheral functions operable in software STOP mode.

4. $\overline{\text{RESET}}$ pin input, reset signal (WDTRES) generation by watchdog timer overflow

5. INTPn (n = 000, 001, 004, 005, 010 to 013, 021, 022, 050, 051, 106, 107, 114, 115, 124 to 126, 130 to 134, 137)

When it is specified that the level of the $\overline{\text{INTPn}}$ pin is to be detected, the software STOP mode and IDLE mode cannot be released.

21.2 Control Registers

(1) Power save control register (PSC)

The PSC register is an 8-bit register that controls the standby function. The STB bit of this register is used to specify the standby mode. This register is a special register (see **3.4.9 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 input clears this register to 00H.

After reset: 00H R/W After reset: FFFFF1FEH

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

NMIM	Control of non-maskable interrupt request (NMI) from NMI pin[Note 1]
0	Standby mode release enabled by NMI request
1	Standby mode release disabled by NMI request

INTM	Control of all maskable interrupt requests[Note 1] (INTxx[Note 2])
0	Standby mode release enabled by INTxx request
1	Standby mode release disabled by INTxx request

STB	Setting operation mode[Note 3]
0	Normal mode
1	Standby mode

Notes 1. Setting these bits is valid only in the IDLE/software STOP mode.

2. For details, see **Tables 20-1 Interrupt Source List**.

3. For the setting procedure, see **21.7 Procedure for Setting and Restoring from IDLE and Software STOP Modes**.

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

2. To set the IDLE mode or software STOP mode, set the PSMR.PSM bit first and then set the STB bit to 1.

21.3 HALT Mode

21.3.1 Setting and operation status

The HALT mode is set when a dedicated instruction (HALT) is executed in the normal operation mode.

In the HALT mode, the clock generator continues operating. Only clock supply to the CPU is stopped; clock supply to the other on-chip peripheral functions continues.

As a result, program execution is stopped, and the internal RAM retains the contents before the HALT mode was set. The on-chip peripheral functions that are independent of instruction processing by the CPU continue operating.

Table 21-3 shows the operation status in the HALT mode.

The average power consumption of the system can be reduced by using the HALT mode in combination with the normal operation mode for intermittent operation.

Cautions 1. Insert five or more NOP instructions after the HALT instruction.

2. If the HALT instruction is executed while an unmasked interrupt request signal is being held pending, the HALT mode is set but is released immediately by the pending interrupt request signal.

21.3.2 Releasing HALT mode

The HALT mode is released by a non-maskable interrupt request signal (NMI pin input, non-maskable interrupt request signal (INTWDT) generation by overflow), an unmasked maskable interrupt request signal, and reset signal ($\overline{\text{RESET}}$ pin input, reset signal generation by watchdog timer overflow (WDTRES)).

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

The HALT mode is released by a non-maskable interrupt request signal or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request signal. If the HALT mode is set in an interrupt servicing routine, however, an interrupt request signal that is issued later is serviced as follows.

(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 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, execution branches to the handler address.

Table 21-2. Operation After Releasing HALT Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address	
Unmasked maskable interrupt request signal	Execution branches to the handler address or the next instruction is executed	The next instruction is executed

(2) Releasing HALT mode by $\overline{\text{RESET}}$ pin input or WDTRES signal generation

The same operation as the normal reset operation is performed.

Table 21-3. Operation Status in HALT Mode

Setting of HALT Mode / Item		Operation Status
Clock generator		Operates
System clock (f_{XX})		Supply
CPU		Stops operation
DMA		Operable
Interrupt controller		Operable
ROM correction		Operable
Timer	TMP0 to TMP2	Operable
	TMQ0	Operable
	TMD0 to TMD3	Operable
	TMENC10	Operable
Watchdog timer		Operable
Serial interface	CSIB0 to CSIB2	Operable
	I^2C[Note]	Operable
	UARTA0 to UARTA3	Operable
A/D converter		Operable
D/A converter		Operable
External bus interface		See **CHAPTER 5 BUS CONTROL FUNCTION**.
Port function		Retains status before HALT mode was set.
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the HALT mode was set.

Note I^2C bus versions (Y products) only (see **Table 1-1**)

21.4 IDLE Mode

21.4.1 Setting and operation status

The IDLE mode is set by clearing the PSMR.PSM bit to 0 and setting the PSC.STB bit to 1 in the normal operation mode.

In the IDLE mode, the clock generator continues operation but clock supply to the CPU and other on-chip peripheral functions stops.

As a result, program execution stops and the contents of the internal RAM before the IDLE mode was set are retained. The CPU and other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate with an external clock continue operating.

Table 21-5 shows the operation status in the IDLE mode.

The IDLE mode can reduce the power consumption more than the HALT mode because it stops the operation of the on-chip peripheral functions. The clock generator does not stop, so the normal operation mode can be restored without waiting for the oscillation stabilization time after the IDLE mode has been released, in the same manner as when the HALT mode is released.

Caution For the IDLE mode setting procedure, see 21.7 Procedure for Setting and Restoring from IDLE and Software STOP Modes.

21.4.2 Releasing IDLE mode

The IDLE mode is released by a non-maskable interrupt request signal (NMI pin input), unmasked external interrupt request signal ($\overline{\text{INTPn}}$ pin input)[Note], unmasked internal interrupt request signal (CSIB-related interrupt request signal in the slave mode and UARTA-related interrupt request signal when external clock is selected) from the peripheral functions operable in the IDLE mode, or $\overline{\text{RESET}}$ input (n = 000, 001, 004, 005, 010 to 013, 021, 022, 050, 051, 106, 107, 114, 115, 124 to 126, 130 to 134, 137).

After the IDLE mode has been released, the normal operation mode is restored.

Note When it is specified that the level of the $\overline{\text{INTPn}}$ pin is to be detected, the IDLE mode cannot be released.

(1) Releasing IDLE mode by non-maskable interrupt request signal (NMI pin input) or unmasked maskable interrupt request signal

The IDLE mode is released by a non-maskable interrupt request signal (NMI pin input) or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request signal. If the IDLE mode is set in an interrupt servicing routine, however, an interrupt request signal that is issued later is processed as follows.

Caution When PSC.INTM bit = 1, the IDLE mode cannot be released by the unmasked maskable interrupt request signal.

(a) If an interrupt request with a priority lower than or same as the interrupt request signal currently being serviced is generated, the IDLE mode is released, but the newly generated interrupt is not acknowledged. The interrupt request signal itself is retained. Therefore, execution starts at the next instruction after the IDLE instruction.

(b) If an interrupt request signal with a priority higher than that of the interrupt request signal currently being serviced is issued (including a non-maskable interrupt request signal), the IDLE mode is released and that interrupt request signal is acknowledged. Therefore, execution branches to the handler address.

Table 21-4. Operation After Releasing IDLE Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
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 IDLE mode by $\overline{\text{RESET}}$ pin input

The same operation as the normal reset operation is performed.

Table 21-5. Operation Status in IDLE Mode

Item \ Setting of IDLE Mode		Operation Status
Clock generator		Operates
System clock (f_{XX})		Stops supply
CPU		Stops operation
DMA		Stops operation
Interrupt controller		Stops operation
ROM correction		Stops operation
Timer	TMP0 to TMP2	Stops operation
	TMQ0	Stops operation
	TMD0 to TMD3	Stops operation
	TMENC10	Stops operation
Watchdog timer		Stops operation
Serial interface	CSIB0 to CSIB2	Operable when SCKn input clock is selected as operation clock (in slave mode) (n = 0 to 2)
	I^2C[Note]	Stops operation
	UARTA0 to UARTA3	Operable when ASCKn input clock is selected as operation clock (when external clock is selected) (n = 0 to 2)
A/D converter		Stops operation
<R> D/A converter		Operable (Retains output value before IDLE mode was set.)
External bus interface		See **CHAPTER 5 BUS CONTROL FUNCTION**.
Port function		Retains status before IDLE mode was set.
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the IDLE mode was set.

Note I^2C bus versions (Y products) only (see **Table 1-1**)

21.5 Software STOP Mode

21.5.1 Setting and operation status

The software STOP mode is set when the PSMR.PSM bit is set to 1 and the PSC.STB bit is set to 1 in the normal operation mode.

In the software 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 software STOP mode was set are retained. The CPU and other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate with an external clock continue operating.

Table 21-7 shows the operation status in the software STOP mode.

Because the software STOP stops operation of the clock generator, it reduces the power consumption to a level lower than the IDLE mode. If the external clock is not used, the power consumption can be minimized with only leakage current flowing.

Caution For the software STOP mode setting procedure, see 21.7 Procedure for Setting and Restoring from IDLE and Software STOP Modes.

21.5.2 Releasing software STOP mode

The software STOP mode is released by a non-maskable interrupt request signal (NMI pin input), unmasked external interrupt request signal ($\overline{\text{INTPn}}$ pin input)[Note], unmasked internal interrupt request signal (CSIB-related interrupt request signal in the slave mode and UARTA-related interrupt request signal when external clock is selected) from the peripheral functions operable in the software STOP mode, or $\overline{\text{RESET}}$ pin input (n = 000, 001, 004, 005, 010 to 013, 021, 022, 050, 051, 106, 107, 114, 115, 124 to 126, 130 to 134, 137).

After the software STOP mode has been released, the normal operation mode is restored after the oscillation stabilization time has been secured.

Note When it is specified that the level of the $\overline{\text{INTPn}}$ pin is to be detected, the software STOP mode cannot be released.

(1) Releasing software STOP mode by non-maskable interrupt request signal (NMI pin input) or unmasked maskable interrupt request signal

The software STOP mode is released by a non-maskable interrupt request signal (NMI pin input) or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request signal. If the software STOP mode is set in an interrupt servicing routine, however, an interrupt request signal that is issued later is serviced as follows.

Caution When PSC.INTM bit = 1, the software STOP mode cannot be released by the 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 software STOP mode is released, but the newly generated interrupt is not acknowledged. The interrupt request signal itself is retained. Therefore, execution starts at the next instruction after the STOP 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 software STOP mode is released and that interrupt request signal is acknowledged. Therefore, execution branches to the handler address.

Table 21-6. Operation After Releasing Software STOP 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 after securing oscillation stabilization time	The next instruction is executed after securing oscillation stabilization time

(2) Releasing software STOP mode by $\overline{\text{RESET}}$ pin input

The same operation as the normal reset operation is performed.

Table 21-7. Operation Status in Software STOP Mode

Item \ Setting of Software STOP Mode		Operation Status
Clock generator		Stops operation
System clock (f_{XX})		Stops supply
CPU		Stops operation
DMA		Stops operation
Interrupt controller		Stops operation
ROM correction		Stops operation
Timer	TMP0 to TMP2	Stops operation
	TMQ0	Stops operation
	TMD0 to TMD3	Stops operation
	TMENC10	Stops operation
Watchdog timer		Stops operation
Serial interface	CSIB0 to CSIB2	Operable when SCKn input clock is selected as operation clock (in slave mode) (n = 0 to 2)
	I^2CNote	Stops operation
	UARTA0 to UARTA3	Operable when ASCKn input clock is selected as operation clock (when external clock is selected) (n = 0 to 2)
A/D converter		Stops operation
<R> D/A converter		Operable (Retains output value before software STOP mode was set.)
External bus interface		See **CHAPTER 5 BUS CONTROL FUNCTION**.
Port function		Retains status before software STOP mode was set.
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the software STOP mode was set.

Note I^2C bus versions (Y products) only (see **Table 1-1**)

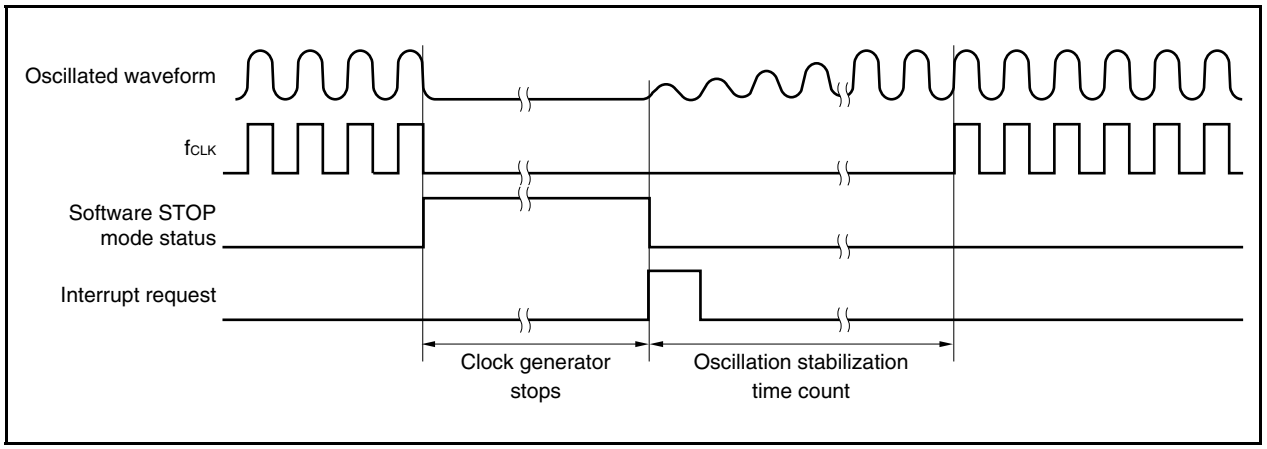
21.6 Securing Oscillation Stabilization Time

When the software STOP mode is released, the oscillation stabilization time set by the OSTS register elapses. When releasing the software STOP mode by $\overline{\text{RESET}}$ pin input, however, secure the oscillation stabilization time by the low-level width of the $\overline{\text{RESET}}$ signal because the oscillation stabilization time is not inserted.

The timer for counting the oscillation stabilization time is shared with the watchdog timer, so oscillation stabilization time equal to the overflow time of the watchdog timer elapses.

The operation performed when the software STOP mode is released by an interrupt request signal is shown below.

Figure 21-2. Oscillation Stabilization Time



Caution For details of the OSTS register, see 7.3 (5) Oscillation stabilization time select register (OSTS).

21.7 Procedure for Setting and Restoring from IDLE and Software STOP Modes

To set the IDLE mode or software STOP mode, be sure to select a multiplication factor for the clock generator by using the CKC register so that the system clock (fxx) is less than 25 MHz. To restore operation from the IDLE mode or software STOP mode, be sure to restore the multiplication factor to the original set value by using the CKC register.

(1) Procedure of setting IDLE and software STOP mode

Set the IDLE or software STOP mode in the following sequence.

In the following example, "[(number)]" in the sequence corresponds to a line number.

[To set]

<1> Prepare data to be set to the CKC.CKDIV1 and CKDIV0 bits [(1)].

<2> Write the data prepared in <1> to the PRCMD register [(2)].

<3> Write the data prepared in <1> to the CKC register (by using the following instructions) [(3)].

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

<4> Insert 10 or more NOP instructions [(4) to (13)].

<5> Prepare data to be set to the PSC register [(14)].

<6> Write the data prepared in <5> to the PRCMD register [(15)].

<7> Write the data prepared in <5> to the PSC register (by using the following instructions) [(16)].

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

<8> Insert five or more NOP instructions [(17) to (21)].

[To restore]

<9> Prepare data to be set to the CKC.CKDIV1 and CKDIV0 bits [(22)].

<10> Write the data prepared in <9> to the PRCMD register [(23)].

<11> Write the data prepared in <9> to the CKC register (by using the following instructions) [(24)].

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

A description example is shown below.

[Example] At 80 MHz operation

[To set]

```
   ST.B        r11, PSMR[r0]       ; Sets PSMR register.
                                   ; (r11 = 0: IDLE mode, r11 = 1: STOP mode)
   (1)  MOV    0x01, r12
   (2)  ST.B   r12, PRCMD[r0]      ; Writes PRCMD register.
   (3)  ST.B   r12, CKC[r0]        ; Sets CKC register (fxx = 2.5 x fx = 20 MHz).
                                   ; fxx = 25 MHz or less
   (4)  NOP                        ; Dummy instructions (10 instructions)
   (5)  NOP
   (6)  NOP
   (7)  NOP
   (8)  NOP
   (9)  NOP
   (10) NOP
   (11) NOP
   (12) NOP
   (13) NOP
   (14) MOV    0x02, r10
   (15) ST.B   r10, PRCMD[r0]      ; Writes PRCMD register.
   (16) ST.B   r10, PSC[r0]        ; Sets PSC register (sets standby mode).
   (17) NOP                        ; Dummy instructions (5 instructions).
   (18) NOP
   (19) NOP
   (20) NOP
   (21) NOP
```

[To restore]

```
   (22) MOV    0x03, r10
   (23) ST.B   r10, PRCMD[r0]      ; Writes PRCMD register.
   (24) ST.B   r10, CKC[r0]        ; Sets CKC register (fxx = 10 x fx = 80 MHz).
                                   ; Sets original multiplication factor.
```

(Normal application codes follow.)

Cautions 1. No special sequence is necessary for reading special registers.

2. Stop all DMA transfer operations before executing this processing.

3. Because the CKC and PSC registers are special registers, it is assumed that <2> and <3>, <6> and <7>, and <10> and <11> are performed by successive store instructions. If another instruction is placed between <2> and <3>, <6> and<7>, and <10> and <11>, the above sequence may not be established if an interrupt is acknowledged by that instruction, causing malfunctioning.

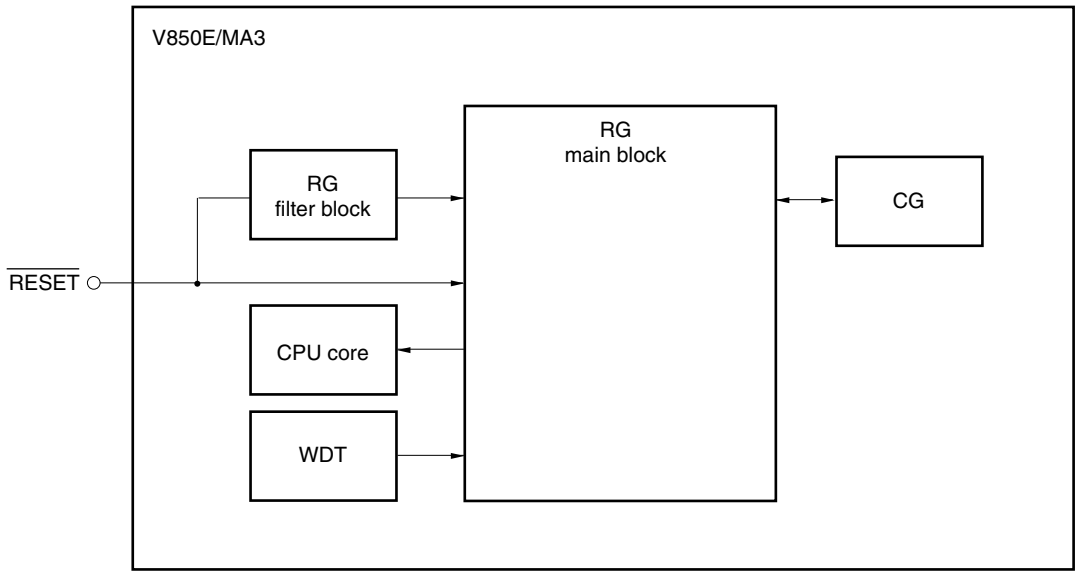
4. The multiplication factor of the system clock (fxx) is changed within the duration of 10 system clocks after the CKC.CKDIV1 and CKDIV0 bits have been set.

CHAPTER 22 RESET FUNCTIONS

22.1 Overview

- System reset by $\overline{\text{RESET}}$ pin input
- System reset (WDTRES) by watchdog timer (WDT) overflow
- Forced reset by on-chip debug function (DCU) and reset mask function (see **CHAPTER 24 ON-CHIP DEBUG FUNCTION (DCU)**.)

22.2 Configuration



Remark RG: Reset generator
WDT: Watchdog timer (see **CHAPTER 13 WATCHDOG TIMER FUNCTIONS**)
CG: Clock generator (see **CHAPTER 7 CLOCK GENERATOR**)

22.3 Control Register

(1) Watchdog timer reset status register (WDRES)

The WDRES register can be used to check whether the V850E/MA3 is reset by the watchdog timer (WDTRES).

This register is a special register and indicates the status of WDTRES.

This register is set only by an 8-bit units when it is written, and by an 8-bit or 1-bit units when it is read.

To write data to the WDRES register, a specific sequence using the PRCMD register as a command register is necessary. If the WDRES register is written in an illegal sequence, writing is invalid, the protect error bit (bit 0 of the SYS register: PRERR) is set to 1, and the write operation is not performed.

$\overline{\text{RESET}}$ input clears this register to 00H.

After reset: 00H R/W Address: FFFFF82AH

	7	6	5	4	3	2	1	<0>
WDRES	0	0	0	0	0	0	0	WRESF

WRESF	WDTRES detection flag
0	WDTRES did not occur.
1	WDTRES occurred.
Set (1) condition:	Occurrence of reset due to overflow of watchdog timer (WDT)
Clear (0) condition:	Writing 0 to this flag by instruction or $\overline{\text{RESET}}$ pin input Only 0 can be written to the WRESF bit.

Caution Write 0 to the WRESF bit after confirming that WRESF bit is 1 (read) in order to avoid conflict with setting the flag.

Remark The WRESF bit can be read/written but it can only be cleared by writing 0, 1 cannot be written to it.

22.4 Operation

When a low level is input to the $\overline{\text{RESET}}$ pin or when the watchdog timer overflows (WDTRES), the V850E/MA3 is reset, and each hardware unit is initialized to a specific status.

The reset status is released when the $\overline{\text{RESET}}$ pin input goes from low to high or if the WDTRES signal is automatically released. After the reset status has been released, the CPU starts program execution.

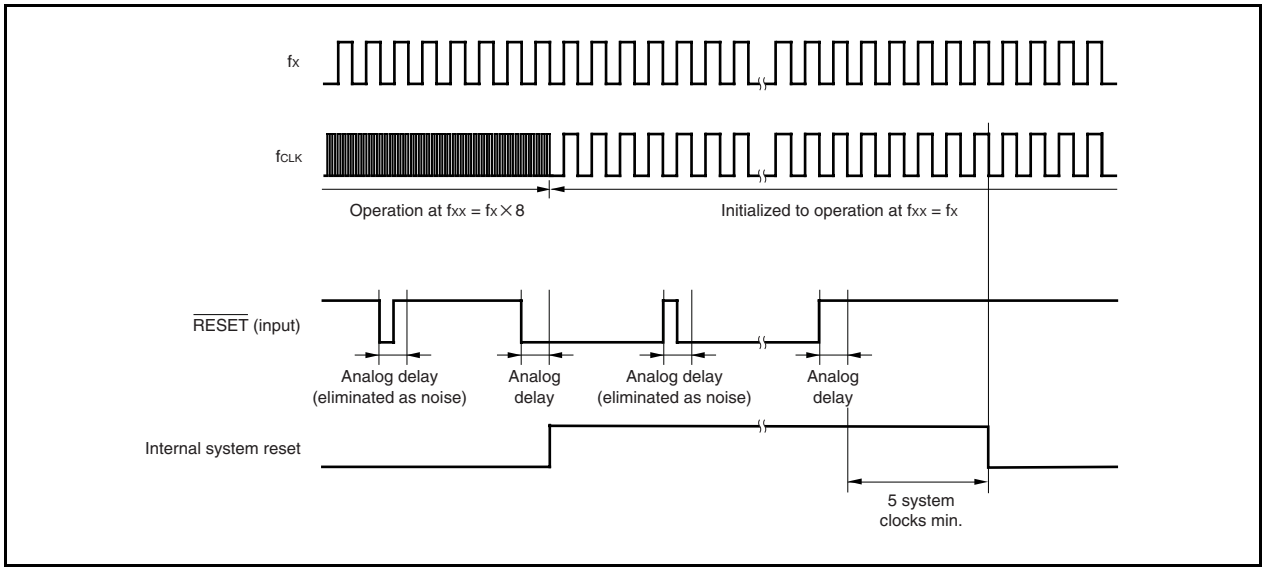
When the reset status is released by $\overline{\text{RESET}}$ pin input, the oscillation stabilization time is not inserted. When inputting the $\overline{\text{RESET}}$ signal with the clock generator stopped (power-on reset or reset input to release the IDLE or software STOP mode), therefore, the oscillation stabilization time must be secured by the low-level width of the $\overline{\text{RESET}}$ pin input.

The status of each hardware unit during the reset period and after the reset status is released is shown below.

Hardware	During Reset Period	After Reset Is Released
Clock generator (in clock-through mode)	Oscillation/supply continues However, the following setting is initialized. • System clock (f_{XX}) is initialized to $1 \times f_X$. • Internal system clock (f_{CLK})/CPU clock (f_{CPU}) is initialized to f_{XX}.	
Clock generator (in PLL mode)	Oscillation/supply stops However, the following setting is initialized. • System clock (f_{XX}) is initialized to $1.25 \times f_X$. • Internal system clock (f_{CLK})/CPU clock (f_{CPU}) is initialized to f_{XX}.	
CPU	Stops operation	Operable
Internal RAM	Undefined if power-on reset or writing data to RAM (by CPU or DMA) conflicts with reset input (data destroyed). Otherwise, retains value immediately before reset input	
On-chip debug function	Operable	
On-chip peripheral I/O registers	Initialized to specific status	
On-chip peripheral functions other than above	Stops operation	Can start operation
Pin function	See **2.2 Pin Status**.	

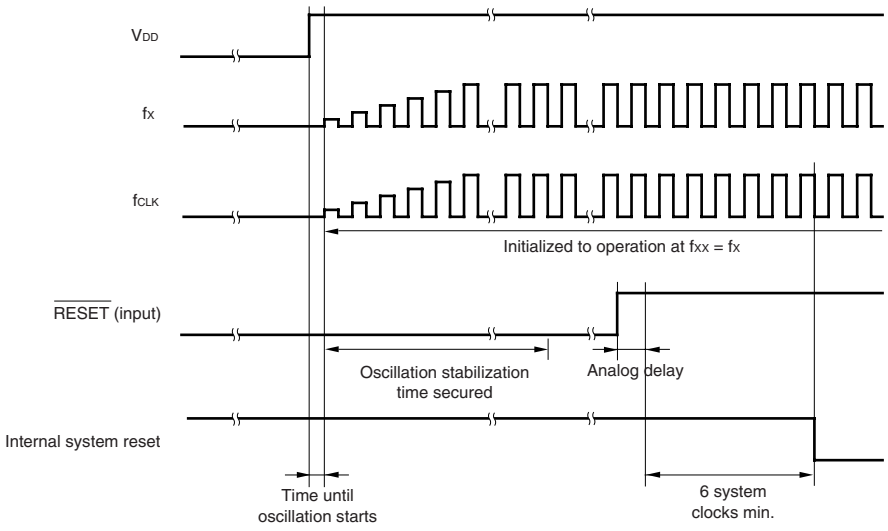
The reset operation by $\overline{\text{RESET}}$ pin input is illustrated below.

Figure 22-1. Reset Operation by $\overline{\text{RESET}}$ Pin Input



The reset operation at power on is illustrated below.

Figure 22-2. Reset Operation at Power On



Cautions 1. Secure the oscillation stabilization time by the low-level width of the $\overline{\text{RESET}}$ signal.

<R> **2. Turn on power in the order of V_{DD} (power to the internal circuitry) and EV_{DD} (power to the external circuitry).**

If the power to the internal circuitry (V_{DD}) is outside the guaranteed operating range (2.3 to 2.7 V) in the power-on/off sequence when a voltage is applied to the power supplies for the external circuitry (EV_{DD}, CV_{DD}, AV_{DD0}, and AV_{DD1}), the following phenomena may occur.

- **A current of about 130 mA (TYP.) flows into the EV_{DD} pin.**
- **An undefined level is output from the following pins.**
  TDO/$\overline{\text{TC3}}$/P27 pin
  ANO0/P80 pin
  ANO1/P81 pin

For details, see 26.2 Power-On/Off Sequence.

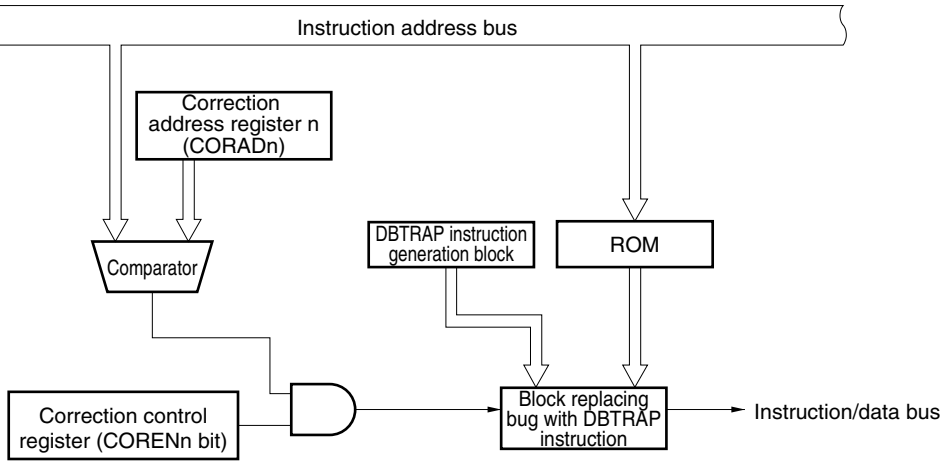
CHAPTER 23 ROM CORRECTION FUNCTION

23.1 Overview

The ROM correction function is used to replace part of the program in the mask ROM or flash memory with the program of an external memory or the internal RAM.

By using this function, program bugs found in the mask ROM or flash memory can be corrected at up to four places.

Figure 23-1. Block Diagram of ROM Correction



Remark n = 0 to 3

23.2 Control Registers

(1) Correction address registers 0 to 3 (CORAD0 to CORAD3)

The CORAD0 to CORAD3 registers are used to set the first address of the program to be corrected.

The program can be corrected at up to four places because four CORADn registers are provided (n = 0 to 3).

These registers can be read or written in 32-bit units.

If the higher 16 bits of the CORADn register are used as the CORADnH register, and the lower 16 bits as the CORADnL register, these registers can be read or written in 16-bit units.

Reset input clears these registers to 00000000H.

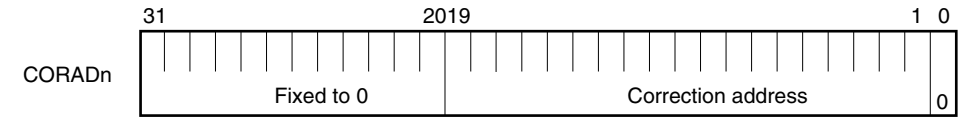
Because the ROM capacity differs depending on the product, set correction addresses in the following ranges.

<R> μPD703131A, 703131AY, 703132A, 703132AY, 703136A, 703136AY (256 KB): 0000000H to 003FFFEH

<R> μPD703133A, 703133AY, 703134A, 703134AY, 70F3134A, 70F3134AY (512 KB): 0000000H to 007FFFEH

Fix bits 0 and 20 to 31 to 0.

After reset: 00000000H R/W Address: CORAD0 FFFFF840H,
CORAD0L FFFFF840H, CORAD0H FFFFF842H,
CORAD1 FFFFF844H,
CORAD1L FFFFF844H, CORAD1H FFFFF846H,
CORAD2 FFFFF848H,
CORAD2L FFFFF848H, CORAD2H FFFFF84AH,
CORAD3 FFFFF84CH,
CORAD3L FFFFF84CH, CORAD3H FFFFF84EH



(n = 0 to 3)

(2) Correction control register (CORCN)

The CORCN register disables or enables the correction operation at the address set to the CORADn register (n = 0 to 3).

Each channel can be enabled or disabled by this register.

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

Reset input clears this register to 00H.

After reset: 00H R/W Address: FFFFF880H

	7	6	5	4	<3>	<2>	<1>	<0>
CORCN	0	0	0	0	COREN3	COREN2	COREN1	COREN0

CORENn	Enables/disables correction operation
0	Disabled
1	Enabled

Remark n = 0 to 3

Table 23-1. Correspondence Between CORCN Register Bits and CORADn Registers

CORCN Register Bit	Corresponding CORADn Register
COREN3	CORAD3
COREN2	CORAD2
COREN1	CORAD1
COREN0	CORAD0

23.3 ROM Correction Operation and Program Flow

<1> If the address to be corrected and the fetch address of the internal ROM match, the fetch code is replaced by the DBTRAP instruction.

<2> When the DBTRAP instruction is executed, execution branches to address 00000060H.

<3> Software processing after branching causes the result of ROM correction to be judged (the fetch address and ROM correction operation are confirmed) and execution to branch to the correction software.

<4> After the correction software has been executed, the return address is set, and return processing is started by the DBRET instruction.

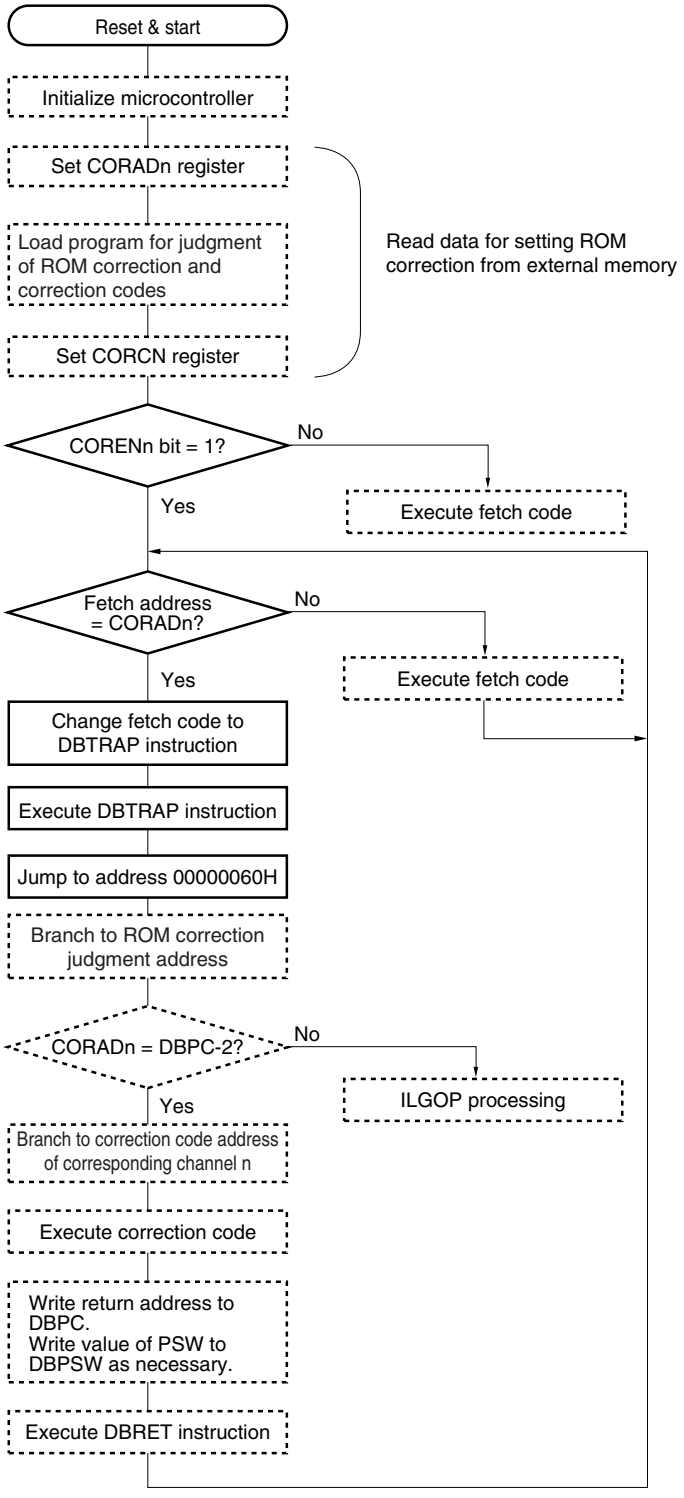
Cautions 1. When setting an address to be corrected to the CORADn register, clear the higher bits to 0 in accordance with the capacity of the internal ROM.

2. The ROM correction function cannot be used to correct the data of the internal ROM. It can only be used to correct instruction codes. If ROM correction is used to correct data, that data is replaced with the DBTRAP instruction code.

<R> **3. Use of ROM correction is prohibited if self-programming is performed in the μPD70F3134A or 70F3134AY.**

<R> **4. When DMA transfer is executed in the internal RAM, do not execute instructions allocated in the internal RAM.**

Figure 23-2. ROM Correction Operation and Program Flow



Remarks 1. : Processing by user program (software)

: Processing by ROM correction (hardware)

2. n = 0 to 3

CHAPTER 24 ON-CHIP DEBUG FUNCTION (DCU)

An on-chip debug unit is provided in the V850E/MA3 and realizes standalone on-chip debugging of the V850E/MA3 by connecting an N-Wire type emulator.

Caution The debug function is supported by the V850E/MA3, but whether this function can be used or not depends on the debugger used.

24.1 Function Overview

24.1.1 On-chip debug unit type

The on-chip debug unit incorporated in the V850E/MA3 is RCU1 (run control unit 1). The on-chip unit incorporated differs depending on the microcontroller, and also features different functions.

24.1.2 Debug function

(1) Debug interface

This interface establishes communication with the host machine by using the $\overline{\text{TRST}}$, TCK, TMS, TDI, and TDO signals, via an N-Wire type emulator. The communication specifications of N-Wire are used for this interface. It does not support a boundary scan function.

(2) On-chip debug

On-chip debugging can be performed by providing wiring and connectors for debugging on the target system. Connect an N-Wire type emulator to the emulator connector.

(3) Forced reset function

The V850E/MA3 can be forcibly reset.

(4) Break reset function

The CPU can be started in the debug mode immediately after resetting the CPU has been cleared.

(5) Forced break function

Execution of the user program can be forcibly stopped (however, the handler of the illegal instruction code exception (first address: 00000060H) cannot be used).

(6) Hardware break function

Two common instruction fetch/access breakpoints can be used. By using the instruction breakpoint, program execution can be suspended at an arbitrary address. By using the access breakpoint, program execution can be suspended by data-accessing an arbitrary address. In addition to these two breakpoints, a software break function is available. Up to four software breakpoints can be set in the internal ROM area. The number of software breakpoints that can be set in the internal RAM area differs depending on the debugger used.

(7) Debug monitor function

During debugging, a memory space for debugging that differs from the user memory space is used (background monitor format). The user program can be executed starting from any address.

While execution of the user program is stopped, the user resources (such as memory and I/O) can be read or written, and the user program can be downloaded.

(8) Mask function

The $\overline{\text{RESET}}$, WDTRES, NMI, INWDT, $\overline{\text{WAIT}}$, and $\overline{\text{HLDRQ}}$ signals can be masked.

The mask functions of the debugger (ID850NWC or ID850NW) and corresponding functions of the V850E/MA3 are shown below.

Mask Function of Debugger (ID850NWC, ID850NW)	Corresponding Functions of V850E/MA3
NMI0	NMI pin input
NMI1	Generation of non-maskable interrupt request signal (INTWDT)
NMI2	×
STOP	×
HOLD	$\overline{\text{HLDRQ}}$ pin input
RESET	Generation of reset signal (WDTRES) by $\overline{\text{RESET}}$ pin input or overflow of watchdog timer
WAIT	$\overline{\text{WAIT}}$ pin input

(9) Timer function

The execution time of the user program can be measured.

24.1.3 ROM security function

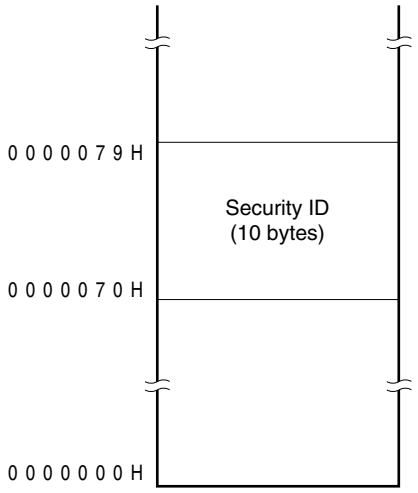
(1) Security ID

The flash memory versions of the V850E/MA3 perform authentication using a 10-byte ID code to prevent the contents of the ROM from being read by an unauthorized person during on-chip debugging by the N-Wire emulator.

Set the ID code in the 10-byte internal ROM area from 0000070H to 0000079H to allow the debugger perform ID authentication.

If the IDs match, the security is released and reading ROM and using the N-Wire emulator are enabled.

- Set the 10-byte ID code to 0000070H to 0000079H.
- Bit 7 of 0000079H is the N-Wire emulator enable flag.
  (0: Disable, 1: Enable)
- When the N-Wire emulator is started, the debugger requests ID input. When the ID code input on the debugger and the ID code set in 0000070H to 0000079H match, the debugger starts.
- Debugging cannot be performed if the N-Wire emulator enable flag is 0, even if the ID codes match.



Caution When the data in the flash memory has been deleted, all the bits are set to 0xFF. Therefore, ID code is FFFFFFFFFFFFFFFFFFFFH.

(2) Setting

When the ID code is “112233445566778899AAH”

Address	Value
0x70	0xAA
0x71	0x99
0x72	0x88
0x73	0x77
0x74	0x66
0x75	0x55
0x76	0x44
0x77	0x33
0x78	0x22
0x79	0x11

When the CA850 is used, continue using the handler at address 0x60 (ILGOP) to input the security code and the data of the system reserved area (00H).

Caution Description of a link directive is not necessary because the handler name at address 0x60 is used.

[Program example]

```
#-------------------------------------
#       ILGOP  handler
#-------------------------------------
        .section        "ILGOP"    --Interrupt handler address 0x60
                                   -- Input ILGOP handler code
        .org                       0x10 -- Skip handler address to 0x70
#-------------------------------------
#       SECURITYID (continue ILGOP  handler)
#-------------------------------------
        .word                      0x778899aa    --0-3 byte code
        .word                      0x33445566    --4-7 byte code
        .hword                     0x1122        --8-9 byte code
```

24.2 Selecting On-Chip Debug Function and Port Function (Including Alternate Functions)

In the V850E/MA3, pins P06, P07, P26, and P27 also function as on-chip debug pins.

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

$\overline{\text{TRST}}$ Pin Low-Level Input	$\overline{\text{TRST}}$ Pin High-Level Input
P06/$\overline{\text{DMARQ2}}$/$\overline{\text{INTP106}}$	TMS
P07/$\overline{\text{DMARQ3}}$/$\overline{\text{INTP107}}$	TCK
P26/$\overline{\text{TC2}}$/$\overline{\text{INTP126}}$	TDI
P27/$\overline{\text{TC3}}$	TDO

Caution When the TMS, TCK, TDI, and TDO pins are used for the on-chip debug function, other alternate functions cannot be used.

24.3 Connection with N-Wire Type Emulator

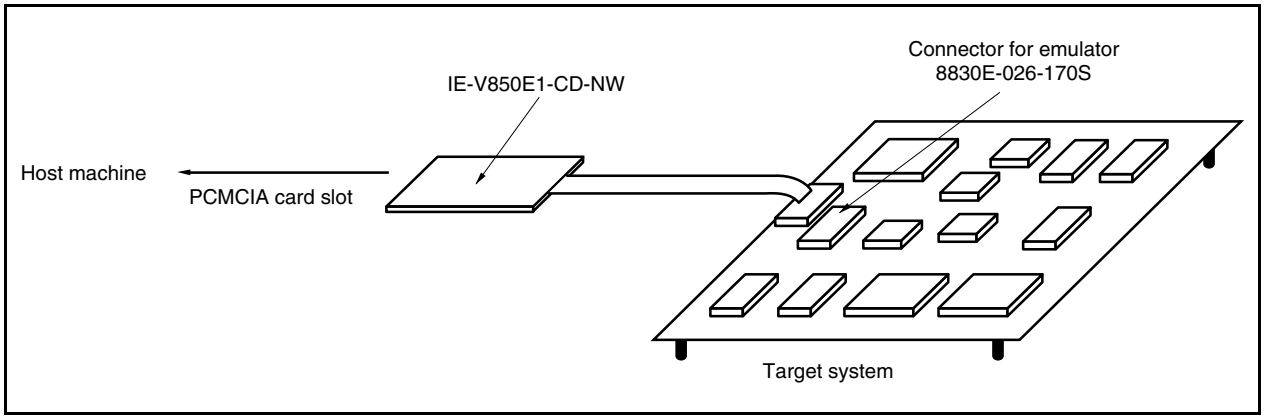
24.3.1 KEL connector

When the IE-V850E1-CD-NW is used, use of the following connector is recommended.

○ Part number
- 8830E-026-170S: Straight type
- 8830E-026-170L: Right-angle type

An emulator and the connection circuit must be mounted on the target system.

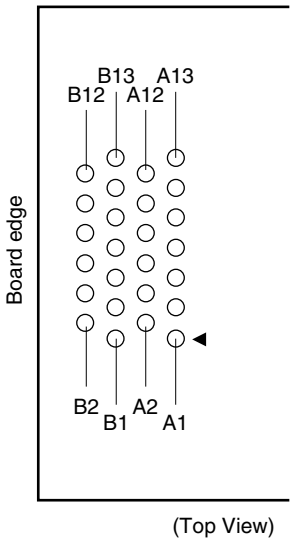
Figure 24-1. Example of Connecting N-Wire Type Emulator (IE-V850E1-CD-NW (N-Wire Card))



(1) Pin configuration

The following shows the pin configuration of the emulator connector (target system side).

Figure 24-2. Pin Configuration of Emulator Connector (on Target System Side)



Caution Design the board based on the dimensions of the connector when actually mounting the connector on the board.

(2) Pin functions

The following table shows the pin functions of the emulator connector (on the target system side).

Table 24-1. Emulator Connector Pin Functions (on Target System Side)

Pin No.	Pin Name	I/O	Pin Function
A1	(Reserved 1)	–	(Connect this pin to GND)
A2	(Reserved 2)	–	(Connect this pin to GND)
A3	(Reserved 3)	–	(Connect this pin to GND)
A4	(Reserved 4)	–	(Connect this pin to GND)
A5	(Reserved 5)	–	(Connect this pin to GND)
A6	(Reserved 6)	–	(Connect this pin to GND)
A7	DDI	Output	Data output for debug serial interface
A8	DCK	Output	Clock output for debug serial interface
A9	DMS	Output	Transfer mode select output for debug serial interface
A10	DDO	Input	Data input for debug serial interface
A11	$\overline{\text{DRST}}$	Output	DCU reset output
A12	(Reserved 7)	–	(Leave this pin open)
A13	FLMD0	Output	Flash download control signal
B1	GND	–	–
B2	GND	–	–
B3	GND	–	–
B4	GND	–	–
B5	GND	–	–
B6	GND	–	–
B7	GND	–	–
B8	GND	–	–
B9	GND	–	–
B10	GND	–	–
B11	PORT0_IN	–	(Connect this pin to GND)
B12	PORT1_IN	–	(Connect this pin to GND)
B13	V_{DD}	–	+3.3 V input (for monitoring power to target)

Cautions 1. The connection of pins not supported by the V850E/MA3 depends on the emulator used.

2. The pattern on the target board must satisfy the following conditions.

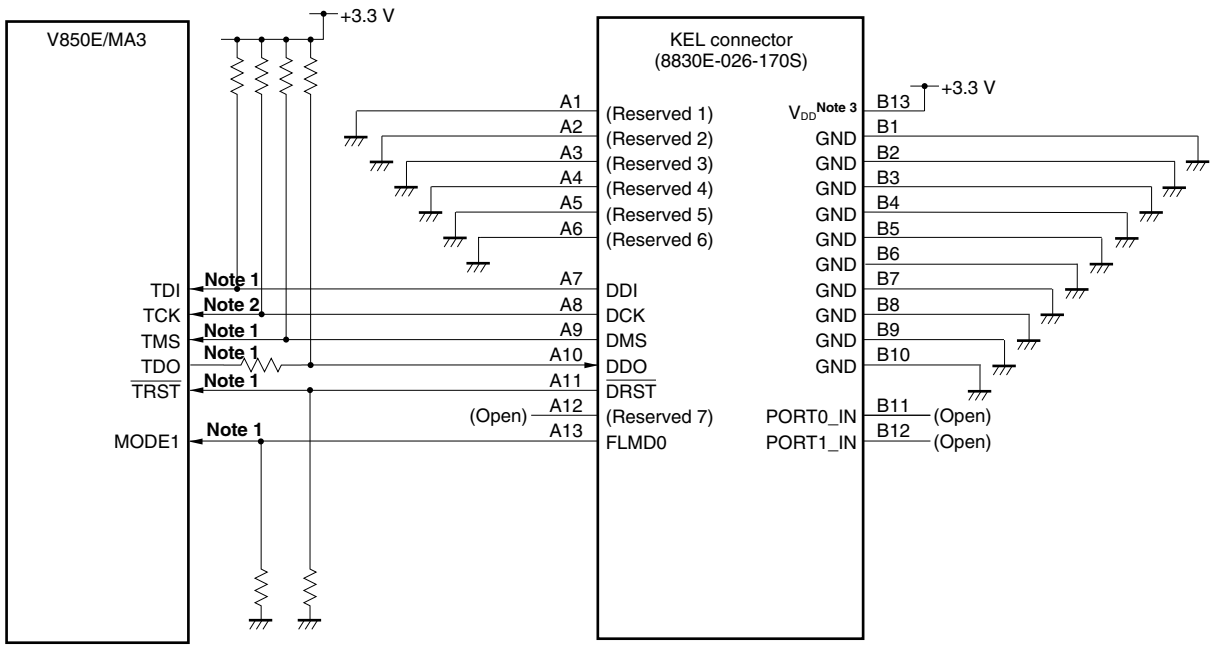
- **Keep the pattern length to within 100 mm.**
- **Shield the clock signal with GND.**

Remark Input/output is when viewed from the emulator side.

(3) Recommended circuit example

The following figure shows an example of the recommended circuit of the emulator connector (on the target system side).

Figure 24-3. Example of Recommended Emulator Connector



Notes 1. Keep the pattern length to within 100 mm.

2. Shield the clock signal with GND.

3. For detecting power to the target board.

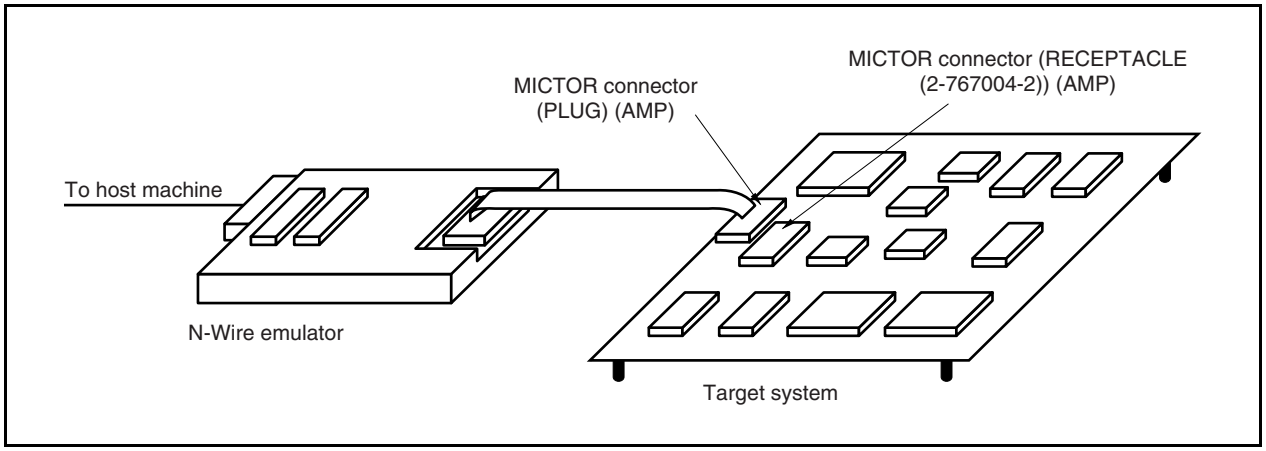
Caution The recommended circuit example shown above assumes that a 3.3 V interface is used.

24.3.2 AMP connector

○ Part number
- 2-767004-2 (MICTOR connector): 38-pin type

A connector for the emulator and a connection circuit must be provided on the target system.

Figure 24-4. Connecting N-Wire Type Emulator

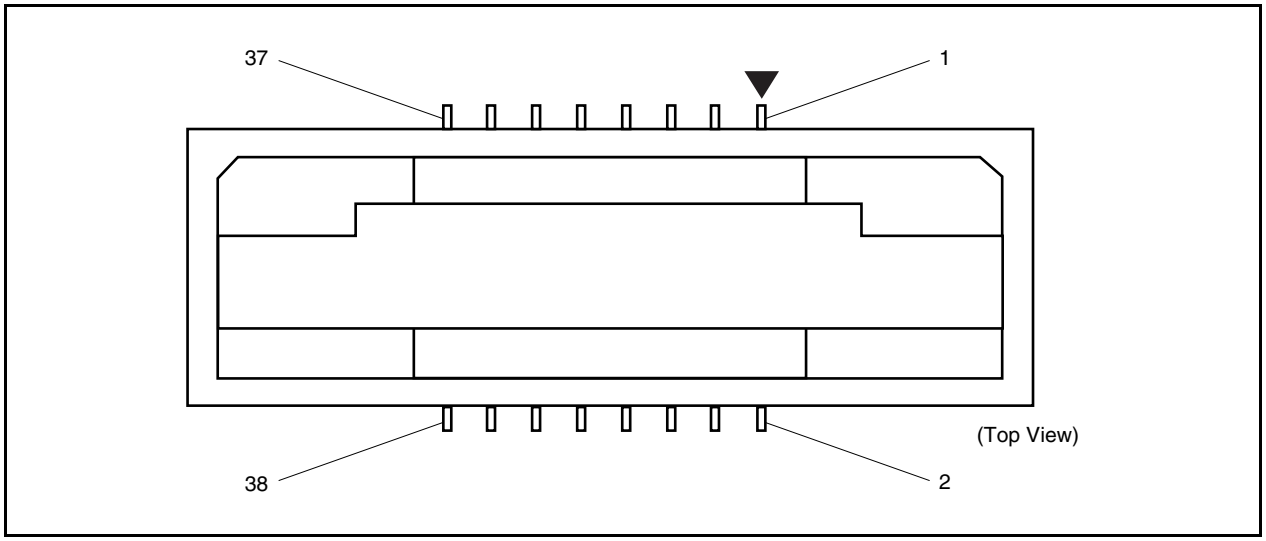


(1) Pin configuration

The following shows the pin configuration of the emulator connector (target system side).

Remark The following connector is recommended.
- 2-767004-2 (AMP): 38-pin type

Figure 24-5. Pin Configuration of Emulator Connector (on Target System Side)



(2) Pin functions

The following table shows the pin functions of the emulator connector (on the target system side).

Table 24-2. Emulator Connector Pin Functions (on Target System Side)

Pin No.	Pin Name	I/O	Pin Function
1	GND	–	–
2	GND	–	–
3	DCK	Output	Clock output for debug serial interface
4	V_{DD}	–	+3.3 V input (for monitoring power to target)
5	DMS	Output	Transfer mode select output for debug serial interface
6	$\overline{\text{DRST}}$	Output	DCU reset output
7	DDI	Output	Data output for debug serial interface
8	PORT0_OUT	Output	(Leave this pin open)
9	DDO	Input	Data input for debug serial interface
10	FLMD0	Output	Flash download control signal
11	(Reserved 1)	–	(Leave this pin open)
12	PORT2_OUT	Output	(Leave this pin open)
13	(Reserved 2)	–	(Leave this pin open)
14	PORT0_IN	Input	(Connect this pin to GND)
15	(Reserved 3)	–	(Leave this pin open)
16	PORT1_IN	Input	(Connect this pin to GND)
17	(Reserved 4)	–	(Connect this pin to GND)
18	PORT2_IN	Input	(Connect this pin to GND)
19	(Reserved 5)	–	(Connect this pin to GND)
20	(Reserved 14)	–	(Leave this pin open)
21	(Reserved 6)	–	(Connect this pin to GND)
22	(Reserved 15)	–	(Connect this pin to GND)
23	(Reserved 7)	–	(Connect this pin to GND)
24	(Reserved 16)	–	(Connect this pin to GND)
25	(Reserved 8)	–	(Connect this pin to GND)
26	(Reserved 17)	–	(Connect this pin to GND)
27	(Reserved 9)	–	(Connect this pin to GND)
28	(Reserved 18)	–	(Connect this pin to GND)
29	(Reserved 10)	–	(Connect this pin to GND)
30	(Reserved 19)	–	(Connect this pin to GND)
31	(Reserved 11)	–	(Connect this pin to GND)
32	(Reserved 20)	–	(Connect this pin to GND)
33	(Reserved 12)	–	(Connect this pin to GND)
34	(Reserved 21)	–	(Connect this pin to GND)
35	(Reserved 13)	–	(Connect this pin to GND)
36	(Reserved 22)	–	(Connect this pin to GND)
37	GND	–	–
38	GND	–	–

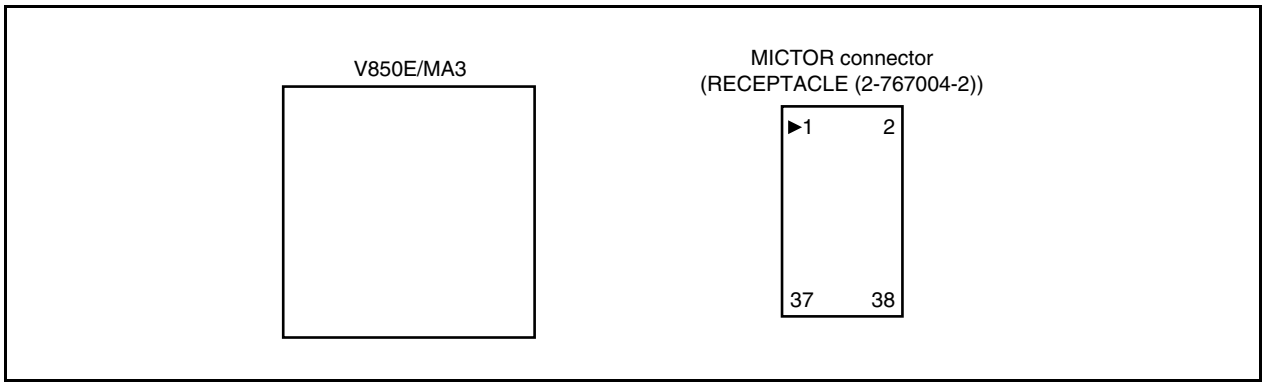
Remarks 1. Input/output is when viewed from the emulator side.

2. Cautions are given on the next page.

Cautions 1. The connection of pins not supported by the V850E/MA3 depends on the emulator used.

2. The pattern on the target board must satisfy the following conditions.

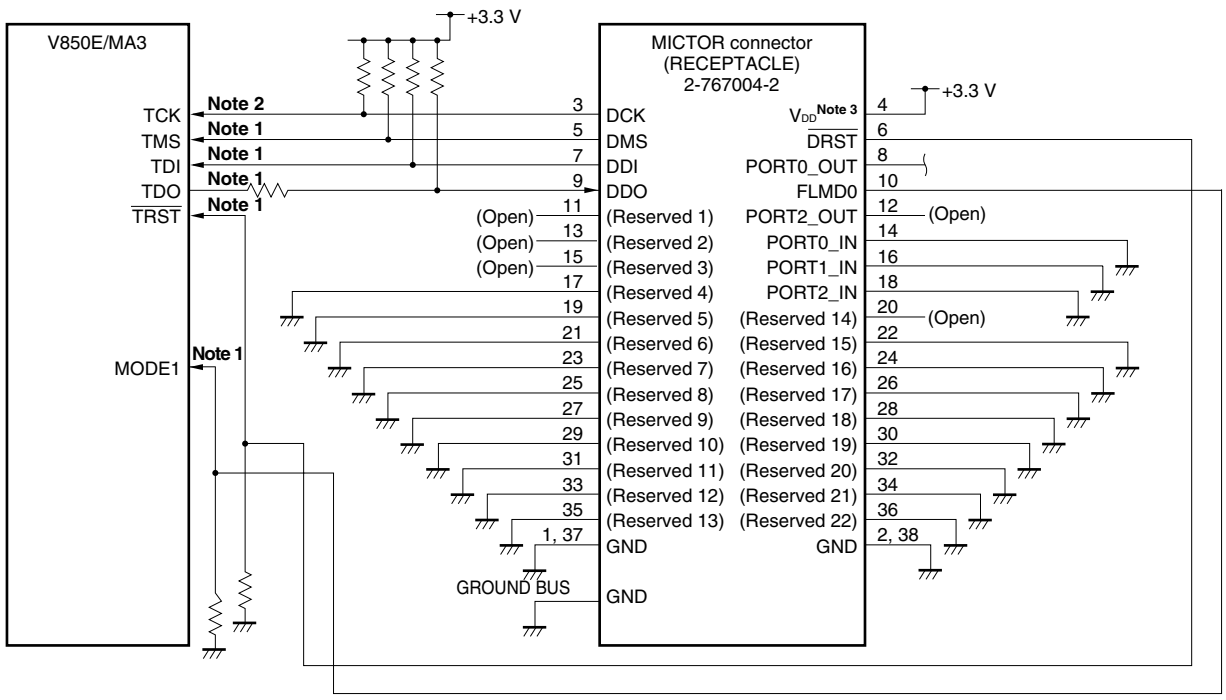
- **Lay out the pattern with the odd pins facing the device (V850E/MA3).**
- **Keep the pattern length to within 100 mm.**
- **Shield the clock signal with GND.**



(3) Recommended circuit example

The following figure shows an example of the recommended circuit of the emulator connector (on the target system side).

Figure 24-6. Example of Recommended Emulator Connector



Notes 1. Keep the pattern length to within 100 mm.

2. Shield the TCK signal by GND.

3. For detecting power to the target board

Caution The recommended circuit example shown above assumes that a 3.3 V interface is used.

<R> 24.4 Cautions

(1) The flash memory of the device used in debugging is rewritten during debugging, so the number of flash memory rewrites cannot be guaranteed. Therefore, do not use the device used in debugging for a mass production product.

(2) Even if reset is masked by using the mask function, the I/O buffers (port pins, etc.) are set to the reset state when the $\overline{\text{RESET}}$ signal is input.

(3) The software breakpoint set in the internal ROM is realized by the ROM correction function, and therefore, it is temporarily disabled when the target is reset or when reset is caused by watchdog timer overflow (WDTRES). The breakpoint is enabled again once a break has occurred because of a hardware break or forced break, but no software break occurs until then.

(4) $\overline{\text{RESET}}$ signal input during a break is masked.

(5) The ROM correction function cannot be emulated.

CHAPTER 25 FLASH MEMORY

The following products are the flash memory versions of the V850E/MA3.

Caution There are differences in noise immunity and noise radiation between the flash memory and mask ROM versions. When preproducing an application set with the flash memory version and then mass producing it with the mask ROM version, be sure to conduct sufficient evaluations on the commercial samples (CS) (not engineering samples (ES)) of the mask ROM versions.
For the electrical specifications related to the flash memory rewriting, see CHAPTER 26 ELECTRICAL SPECIFICATIONS.

<R> • μPD70F3134A, 70F3134AY: Products with 512 KB flash memory

Writing to flash memory can be performed with memory mounted on the target system (on board). A dedicated flash programmer is connected to the target system to perform writing.

The following can be considered as the development environment and the applications using flash memory.

- Software can be changed after the V850E/MA3 is solder-mounted on the target system.
- Small scale production of various models is made easier by differentiating software.
- Data adjustment in starting mass production is made easier.

25.1 Features

- All block batch erase, or erase in block units (block 0: 64 KB, block 1: 448 KB)
- Communication via serial interface from the dedicated flash programmer
- Erase/write voltage: V_{DD} = 2.5 V
- On-board programming

25.2 Writing with Flash Programmer

Writing can be performed either on-board or off-board by the dedicated flash programmer.

(1) On-board programming

The contents of the flash memory are rewritten after the V850E/MA3 is mounted on the target system. Mount connectors, etc., on the target system (recommended target connector: 7616-5002SC (Sumitomo 3M Ltd.)) to connect the dedicated flash programmer.

(2) Off-board programming

Writing to flash memory is performed by the dedicated program adapter (FA Series), etc., before mounting the V850E/MA3 on the target system.

Remarks 1. The FA Series is a product of Naito Densei Machida Mfg. Co., Ltd.

2. When the flash memory is written with a dedicated flash programmer, the operation is always performed at a frequency multiplied by 10 in the PLL mode.

3. When the flash memory is written with power supplied from the dedicated flash programmer to the V850E/MA3, be sure to input 4 MHz frequency to the X1 pin.
If power is supplied from an external source, input a frequency in the range of 4 MHz to 8 MHz to the X1 pin.

Table 25-1. Wiring of V850E/MA3 Flash Writing Adapter (FA-144GJ-UEN-A)

Flash Programmer (PG-FP4) Connection Pin			Name of FA Board Pin	CSIB0 + HS Used		CSIB0 Used		UARTA0 Used	
Signal Name	I/O[Note 1]	Pin Function		Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	SI	SO0	55	SO0	55	TXD0	55
SO/TxD	Output	Transmit signal	SO	SI0	54	SI0	54	RXD0	54
SCK	Output	Transfer clock	SCK	SCK0	53	SCK0	53	Not needed	Not needed
CLK	Output	Clock to V850E/MA3	CLKOUT	X1	58	X1	58	X1	58
/RESET	Output	Reset signal	/RESET	$\overline{\text{RESET}}$	66	$\overline{\text{RESET}}$	66	$\overline{\text{RESET}}$	66
FLMD0	Output	Write mode setting	FLMD0	MODE1	65	MODE1	65	MODE1	65
FLMD1	Output	Write mode setting	FLMD1	MODE0 or unused[Note 2]	64	MODE0 or unused[Note 2]	64	MODE0 or unused[Note 2]	64
HS	Input	Handshake signal for CSI0 + HS	RESERVE/ HS	PCM0	91	Not needed	Not needed	Not needed	Not needed
VDD	–	VDD voltage generation/ voltage monitor	VDD	EV_{DD}	8, 37, 98, 112, 134	EV_{DD}	8, 37, 98, 112, 134	EV_{DD}	8, 37, 98, 112, 134
				CV_{DD}	56	CV_{DD}	56	CV_{DD}	56
				AV_{DD0}	72	AV_{DD0}	72	AV_{DD0}	72
				AV_{DD1}	67	AV_{DD1}	67	AV_{DD1}	67
VDD2	–	Write voltage	VDD2	V_{DD}	23, 62, 81, 124	V_{DD}	23, 62, 81, 124	V_{DD}	23, 62, 81, 124
GND	–	Ground	GND	V_{SS}	24, 63, 82, 125	V_{SS}	24, 63, 82, 125	V_{SS}	24, 63, 82, 125
				EV_{SS}	9, 38, 99, 113, 135	EV_{SS}	9, 38, 99, 113, 135	EV_{SS}	9, 38, 99, 113, 135
				CV_{SS}	59	CV_{SS}	59	CV_{SS}	59
				AV_{SS0}	71	AV_{SS0}	71	AV_{SS0}	71
				AV_{SS1}	70	AV_{SS1}	70	AV_{SS1}	70

Notes 1. Input/output is when viewed from the flash programmer (PG-FP4) side.

2. When unused, be sure to connect the MODE0 pin to V_{SS} via a resistor on the board side.

Figure 25-1. Wiring Example of V850E/MA3 Flash Writing Adapter (FA-144GJ-UEN-A) (1/2)

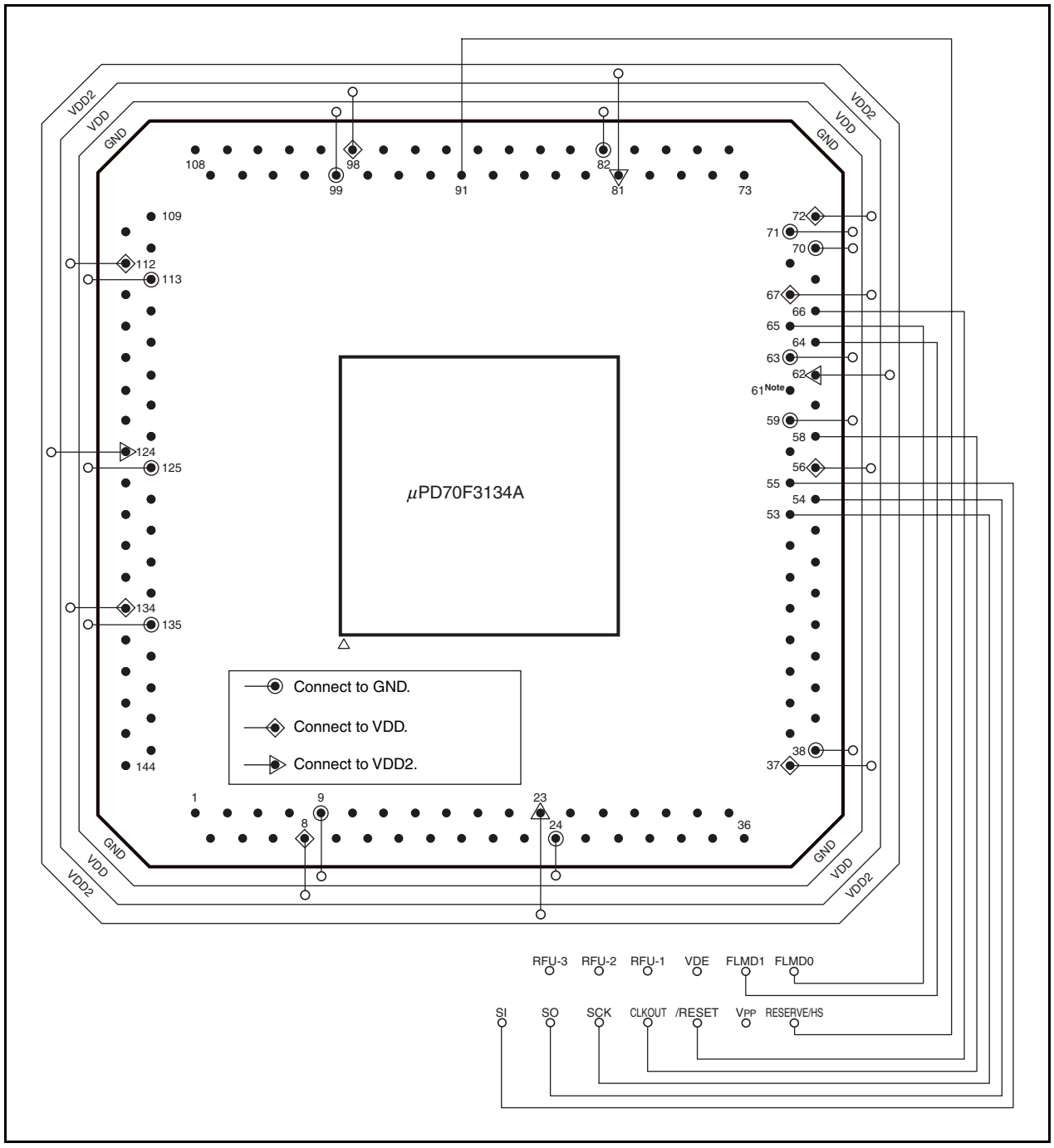


Figure 25-1. Wiring Example of V850E/MA3 Flash Writing Adapter (FA-144GJ-UEN-A) (2/2)

Note The PSEL pin must be fixed to the high level or low level, depending on the frequency input to the X1 pin.

Input Level of PSEL Pin	Clock Frequency Input to X1 Pin
L	4.0 to 5.5 MHz
H	5.5 to 8.0 MHz

Remarks 1. Process the pins not shown above in the same manner as when they are not used (see **2.3 Pin I/O Circuits and Recommended Connection of Unused Pins**). To connect a pin to EV_{DD} or EV_{SS} via resistor, using a resistor of 1 kΩ to 10 kΩ is recommended.

2. This adapter is designed for a 144-pin plastic LQFP package for single-power flash memory.

3. This figure shows the wiring of CSIB0 supporting handshaking.

4. To write the flash memory with a flash programmer, the operation is always performed at a frequency multiplied by 10 in the PLL mode.

5. When the flash memory is written with power supplied from the flash programmer (PG-FP4) to the V850E/MA3, be sure to set the frequency input to the X1 pin to 4 MHz.
If supplying power from an external source, input a frequency in the range of 4 MHz to 8 MHz to the X1 pin.

Table 25-2. Wiring of V850E/MA3 Flash Writing Adapter (FA-161F1-EN4-A)

Flash Programmer (PG-FP4) Connection Pin			Name of FA Board Pin	CSIB0 + HS Used		CSIB0 Used		UARTA0 Used	
Signal Name	I/O[Note 1]	Pin Function		Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	SI	SO0	L8	SO0	L8	TXD0	L8
SO/TxD	Output	Transmit signal	SO	SI0	P7	SI0	P7	RXD0	P7
SCK	Output	Transfer clock	SCK	SCK0	N7	SCK0	N7	Not needed	Not needed
CLK	Output	Clock to V850E/MA3	CLKOUT	X1	P10	X1	P10	X1	P10
/RESET	Output	Reset signal	/RESET	$\overline{\text{RESET}}$	P12	$\overline{\text{RESET}}$	P12	$\overline{\text{RESET}}$	P12
FLMD0	Output	Write mode setting	FLMD0	MODE1	M10	MODE1	M10	MODE1	M10
FLMD1	Output	Write mode setting	FLMD1	MODE0 or unused[Note 2]	L9	MODE0 or unused[Note 2]	L9	MODE0 or unused[Note 2]	L9
HS	Input	Handshake signal for CSI0 + HS	RESERVE/HS	PCM0	G11	Not needed	Not needed	Not needed	Not needed
VDD	–	VDD voltage generation/ voltage monitor	VDD	EV_{DD}	A12, C6, F4, F12, P1	EV_{DD}	A12, C6, F4, F12, P1	EV_{DD}	A12, C6, F4, F12, P1
				CV_{DD}	P9	CV_{DD}	P9	CV_{DD}	P9
				AV_{DD0}	L10	AV_{DD0}	L10	AV_{DD0}	L10
				AV_{DD1}	N13	AV_{DD1}	N13	AV_{DD1}	N13
VDD2	–	Write voltage	VDD2	V_{DD}	C8, J1, K14, M8	V_{DD}	C8, J1, K14, M8	V_{DD}	C8, J1, K14, M8
GND	–	Ground	GND	V_{SS}	B8, J4, K13, N8	V_{SS}	B8, J4, K13, N8	V_{SS}	B8, J4, K13, N8
				EV_{SS}	A1, A5, B1, B5, B14, C1, C11, C14, D1, D14, E14, L1, M1, N1, P2, P5	EV_{SS}	A1, A5, B1, B5, B14, C1, C11, C14, D1, D14, E14, L1, M1, N1, P2, P5	EV_{SS}	A1, A5, B1, B5, B14, C1, C11, C14, D1, D14, E14, L1, M1, N1, P2, P5
				CV_{SS}	N10	CV_{SS}	N10	CV_{SS}	N10
				AV_{SS0}	M11	AV_{SS0}	M11	AV_{SS0}	M11
				AV_{SS1}	N12	AV_{SS1}	N12	AV_{SS1}	N12

Notes 1. Input/output is when viewed from the flash programmer (PG-FP4) side.

2. When unused, be sure to connect the MODE0 pin to EV_{SS} via a resistor on the board side.

Caution The clock cannot be supplied from the CLK pin of the flash programmer. Provide an oscillator on the board to supply the clock.

Figure 25-2. Wiring Example of V850E/MA3 Flash Writing Adapter (FA-161F1-EN4-A) (1/2)

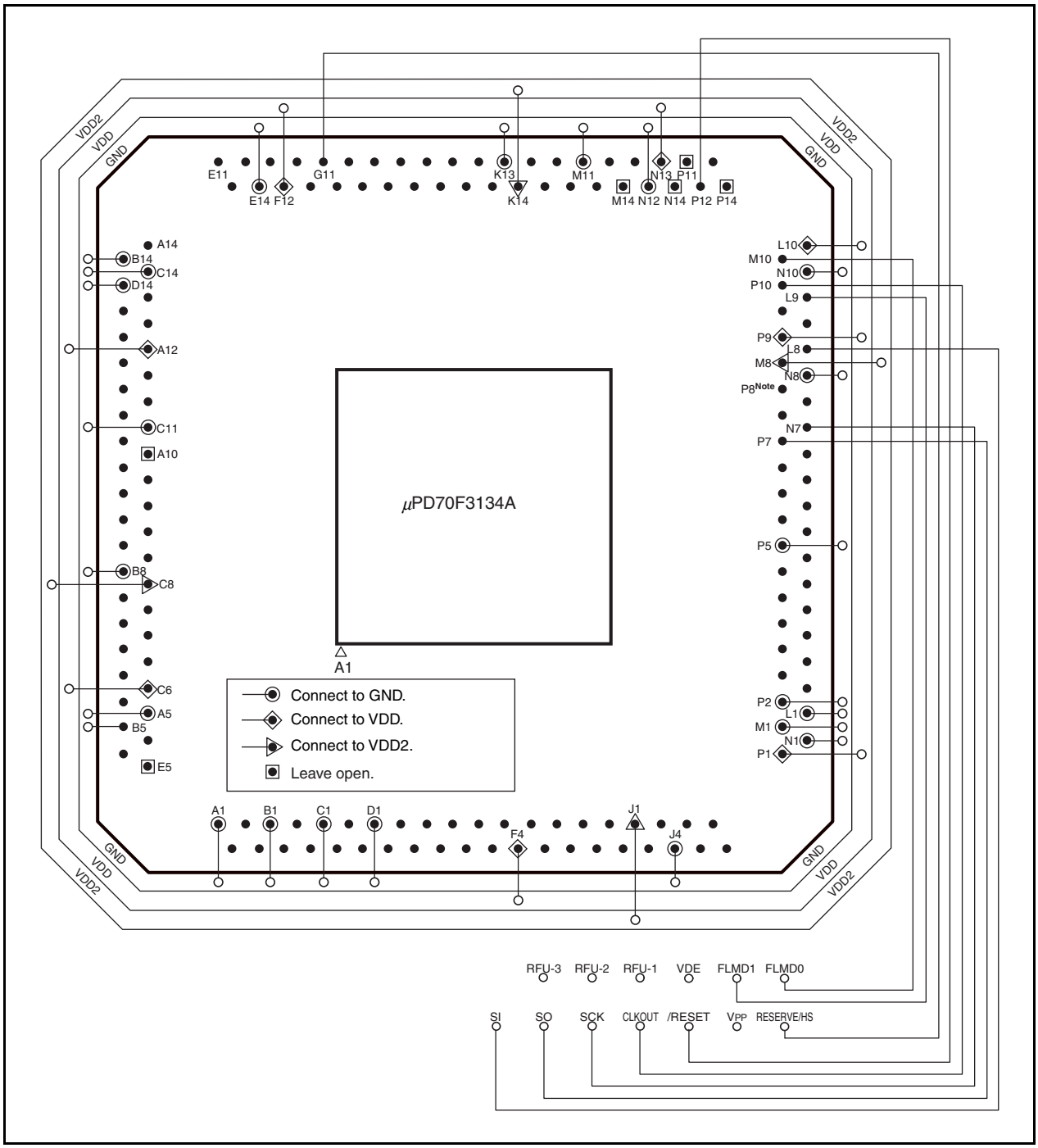


Figure 25-2. Wiring Example of V850E/MA3 Flash Writing Adapter (FA-161F1-EN4-A) (2/2)

Note The PSEL pin must be fixed to the high level or low level, depending on the frequency input to the X1 pin.

Input Level of PSEL Pin	Clock Frequency Input to X1 Pin
L	4.0 to 5.5 MHz
H	5.5 to 8.0 MHz

Remarks 1. Process the pins not shown above in the same manner as when they are not used (see **2.3 Pin I/O Circuits and Recommended Connection of Unused Pins**). To connect a pin to EV_{DD} or EV_{SS} via a resistor, using a resistor of 1 kΩ to 10 kΩ is recommended.

2. This adapter is designed for a 161-pin plastic FBGA package for single-power flash memory.

3. This figure shows the wiring of CSIB0 supporting handshaking.

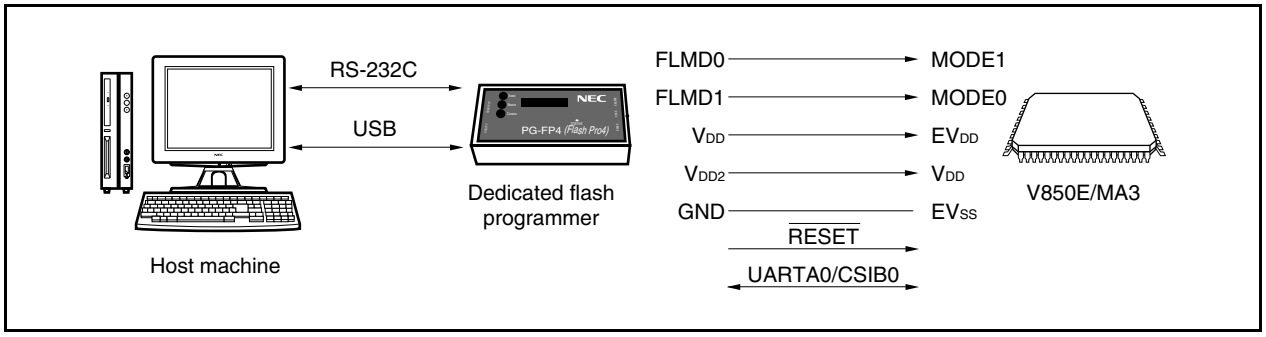
4. To write the flash memory with a flash programmer, the operation is always performed at a frequency multiplied by 10 in the PLL mode.

5. When the flash memory is written with power supplied from the flash programmer (PG-FP4) to the V850E/MA3, be sure to set the frequency input to the X1 pin to 4 MHz.
If supplying power from an external source, input a frequency in the range of 4 MHz to 8 MHz to the X1 pin.

25.3 Programming Environment

The following shows the environment required for writing programs to the flash memory of the V850E/MA3.

Figure 25-3. Environment Required for Writing Program to Flash Memory



A host machine is required for controlling the dedicated flash programmer.

UARTA0 or CSIB0 is used for the interface between the dedicated flash programmer and the V850E/MA3 to perform writing, erasing, etc. A dedicated program adapter (FA Series) is required for off-board writing.

25.4 Communication Mode

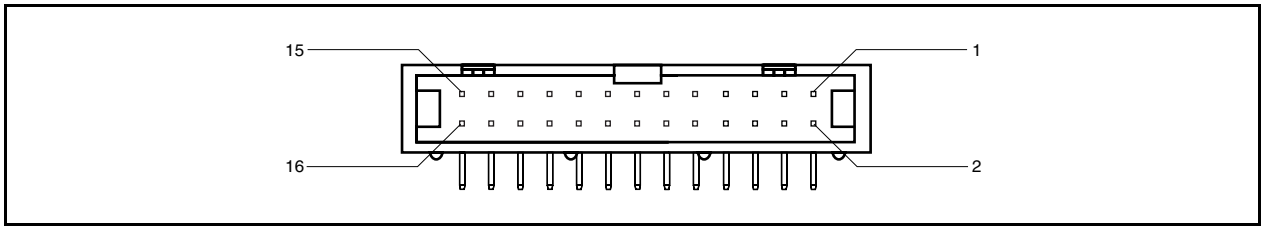
Communication between the dedicated flash programmer and the V850E/MA3 is performed via serial communication using UARTA0 or CSIB0.

Remark The recommended target connector is as follows.

- 7616-5002SC (Sumitomo 3M Ltd.)

The following figure outlines the connector (when viewed from the connector insertion side).

Figure 25-4. Target Connector Outline (Viewed from Connector Insertion Side)



(1) UARTA0

Transfer rate: 9,600 to 153,600 bps (LSB first)

Table 25-3. Wiring Correspondence Between Dedicated Flash Programmer and V850E/MA3

Pin No.	Dedicated Flash Programmer (PG-FP4)	I/O (PG-FP4 Side)	V850E/MA3		
			Pin Name	Pin No.	
				GJ	F1
1	GND	–	EV_{SS}	9, 38, 99, 113, 135	**Note**
2	$\overline{RESET}$	Output	$\overline{RESET}$	66	P12
3	SI/RxD	Input	TXD0	55	L8
4	V_{DD}	–	EV_{DD}	8, 37, 98, 112, 134	A12, C6, F4, F12, P1
5	SO/TxD	Output	RXD0	54	P7
6	V_{PP}	×	NC	–	–
7	SCK	×	NC	–	–
8	H/S	×	NC	–	–
9	CLK	Output	X1	58	P10
10	VDE	×	NC	–	–
11	V_{DD2}	–	V_{DD}	23, 62, 81, 124	C8, J1, K14, M8
12	FLMD1	Output	MODE0	64	L9
13	RFU-1	×	NC	–	–
14	FLMD0	Output	MODE1	65	M10
15	Not used	×	NC	–	–
16	Not used	×	NC	–	–

Note A1, A5, B1, B5, B14, C1, C11, C14, D1, D14, E14, L1, M1, N1, P2, P5

Remark NC: No Connection
GJ: 144-pin plastic LQFP (fine pitch) (20×20)
F1: 161-pin plastic FBGA (13×13)

(2) CSIB0

Transfer rate: 2.4 kHz to 2,500 kHz (MSB first)

Table 25-4. Wiring Correspondence Between Dedicated Flash Programmer and V850E/MA3

Pin No.	Dedicated Flash Programmer (PG-FP4)	I/O (PG-FP4 Side)	V850E/MA3		
			Pin Name	Pin No.	
				GJ	F1
1	GND	–	EV_{SS}	9, 38, 99, 113, 135	**Note**
2	$\overline{RESET}$	Output	$\overline{RESET}$	66	P12
3	SI/RxD	Input	SO0	55	L8
4	V_{DD}	–	EV_{DD}	8, 37, 98, 112, 134	A12, C6, F4, F12, P1
5	SO/TxD	Output	SI0	54	P7
6	V_{PP}	×	NC	–	–
7	SCK	Output	SCK0	53	N7
8	H/S	×	NC	–	–
9	CLK	Output	X1	58	P10
10	VDE	×	NC	–	–
11	V_{DD2}	–	V_{DD}	23, 62, 81, 124	C8, J1, K14, M8
12	FLMD1	Output	MODE0	64	L9
13	RFU-1	×	NC	–	–
14	FLMD0	Output	MODE1	65	M10
15	Not used	×	NC	–	–
16	Not used	×	NC	–	–

Note A1, A5, B1, B5, B14, C1, C11, C14, D1, D14, E14, L1, M1, N1, P2, P5

Remark NC: No Connection
GJ: 144-pin plastic LQFP (fine pitch) (20×20)
F1: 161-pin plastic FBGA (13×13)

(3) CSIB0 + HS

Transfer rate: 2.4 kHz to 2,500 kHz (MSB first)

Table 25-5. Wiring Correspondence Between Dedicated Flash Programmer and V850E/MA3

Pin No.	Dedicated Flash Programmer (PG-FP4)	I/O (PG-FP4 Side)	V850E/MA3		
			Pin Name	Pin No.	
				GJ	F1
1	GND	–	EV_{SS}	9, 38, 99, 113, 135	**Note**
2	$\overline{RESET}$	Output	$\overline{RESET}$	66	P12
3	SI/RxD	Input	SO0	55	L8
4	V_{DD}	–	EV_{DD}	8, 37, 98, 112, 134	A12, C6, F4, F12, P1
5	SO/TxD	Output	SI0	54	P7
6	V_{PP}	×	NC	–	–
7	SCK	Output	SCK0	53	N7
8	H/S	Input	PCM0	91	G11
9	CLK	Output	X1	58	P10
10	VDE	×	NC	–	–
11	V_{DD2}	–	V_{DD}	23, 62, 81, 124	C8, J1, K14, M8
12	FLMD1	Output	MODE0	64	L9
13	RFU-1	×	NC	–	–
14	FLMD0	Output	MODE1	65	M10
15	Not used	×	NC	–	–
16	Not used	×	NC	–	–

Note A1, A5, B1, B5, B14, C1, C11, C14, D1, D14, E14, L1, M1, N1, P2, P5

Remark NC: No Connection
GJ: 144-pin plastic LQFP (fine pitch) (20×20)
F1: 161-pin plastic FBGA (13×13)

25.5 Pin Connection

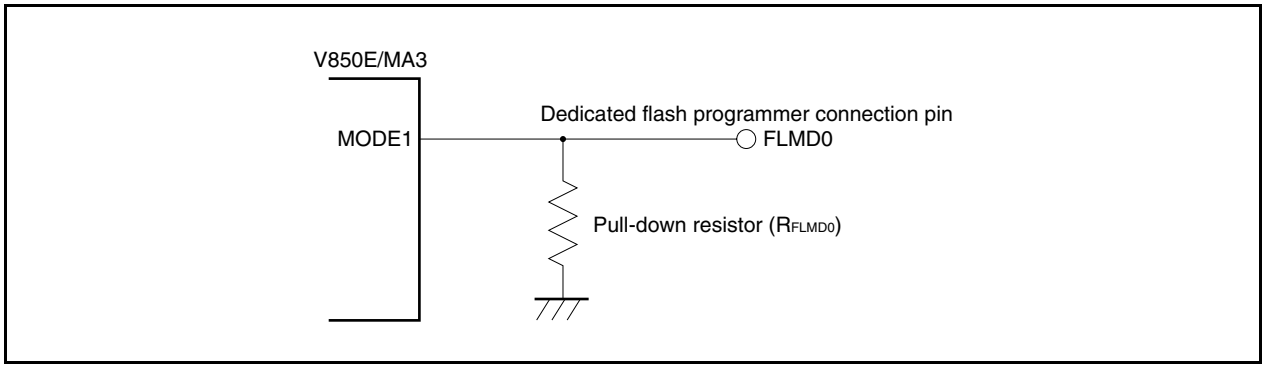
When performing on-board writing, install a connector on the target system (recommended target connector: 7616-5002SC (Sumitomo 3M Ltd.)) to connect the dedicated flash programmer. Also, design a function to switch from the normal operation mode (single-chip mode) to the flash memory programming mode.

In the flash memory programming mode, all the pins not used for flash memory programming enter the same status as they were immediately after reset. Therefore, because all the ports become output high-impedance, pin connection is required when the external device does not acknowledge the output high-impedance status.

25.5.1 MODE1 pin

In the normal operation mode, 0 V is input to the MODE1 pin. In the flash memory programming mode, the MODE1 pin becomes high level. The following shows an example of the connection of the MODE1 pin.

Figure 25-5. Example of MODE1 Pin Connection



25.5.2 Serial interface pins

The following shows the pins used by each serial interface.

Table 25-6. Pins Used by Serial Interface

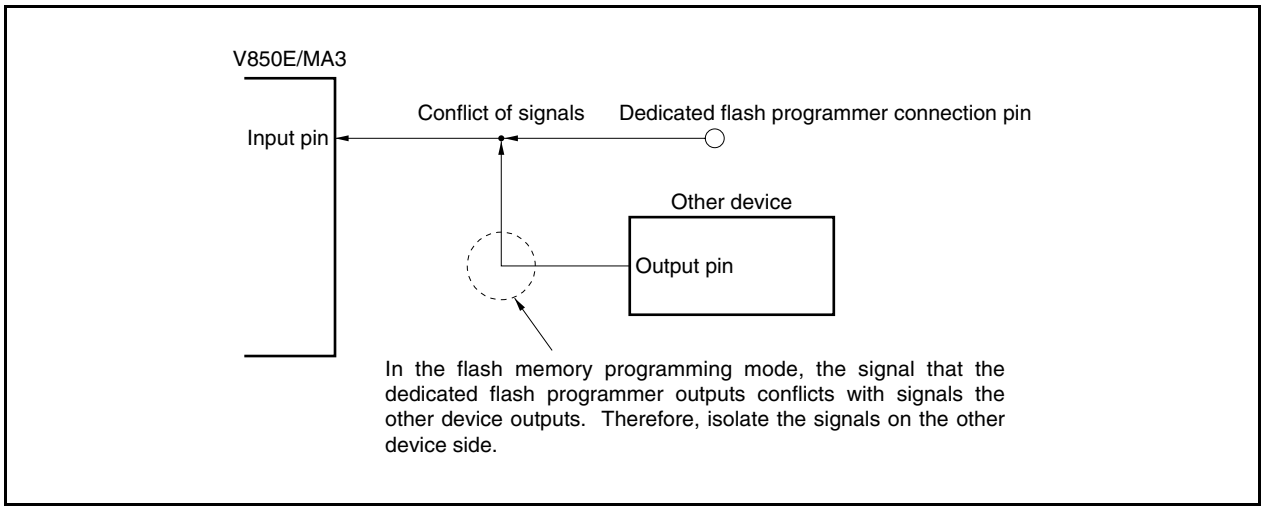
Serial Interface	Pins Used
UARTA0	TXD0, RXD0
CSIB0	SO0, SI0, SCK0
CSIB0 + HS	SO0, SI0, SCK0, PCM0

When connecting a dedicated flash programmer to a serial interface pin that is connected to other devices on-board, care should be taken to avoid the conflict of signals and the malfunction of other devices, etc.

(1) Conflict of signals

When connecting a dedicated flash programmer (output) to a serial interface pin (input) 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.

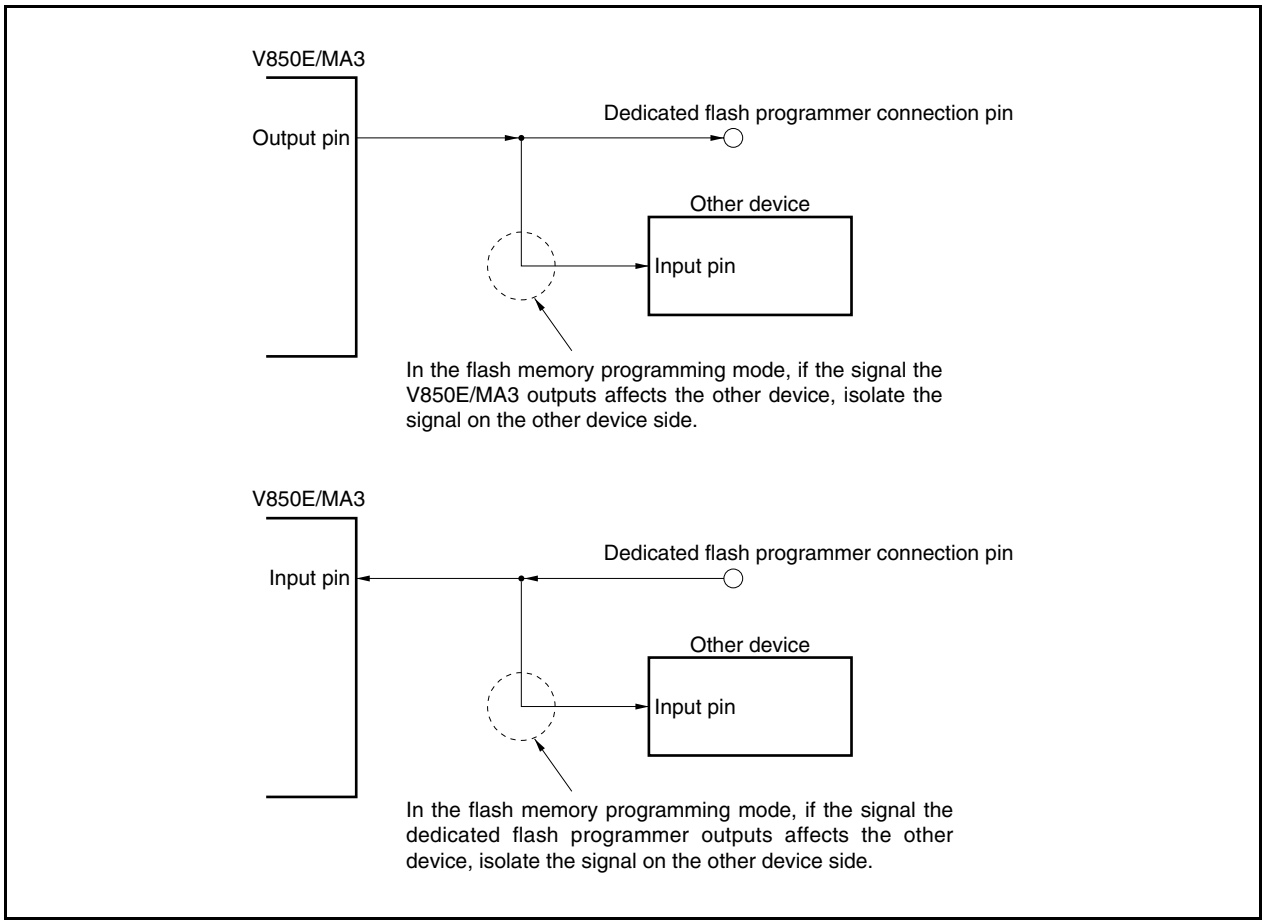
Figure 25-6. Conflict of Signals (Serial Interface Input Pin)



(2) Malfunction of other device

When connecting a dedicated flash programmer (output or input) to a serial interface pin (input or output) connected to another device (input), the signal output to the other device may cause the device to malfunction. To avoid this, isolate the connection to the other device.

Figure 25-7. Malfunction of Other Device

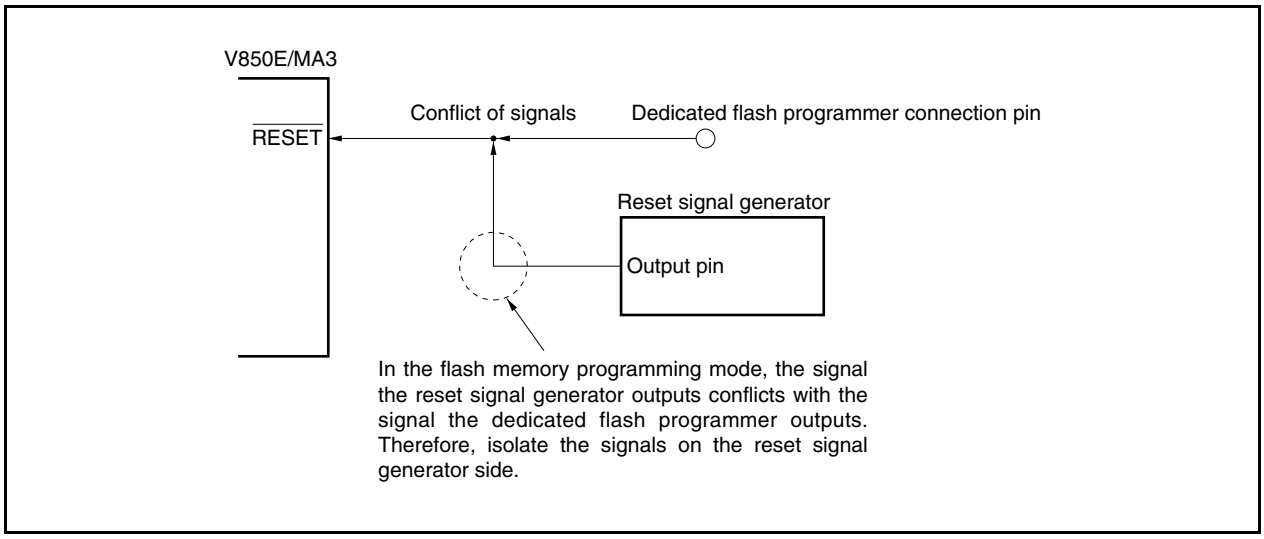


25.5.3 $\overline{\text{RESET}}$ pin

When connecting the reset signals of the dedicated flash programmer to the $\overline{\text{RESET}}$ pin, which is connected to the reset signal generator on-board, a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the reset signal generator.

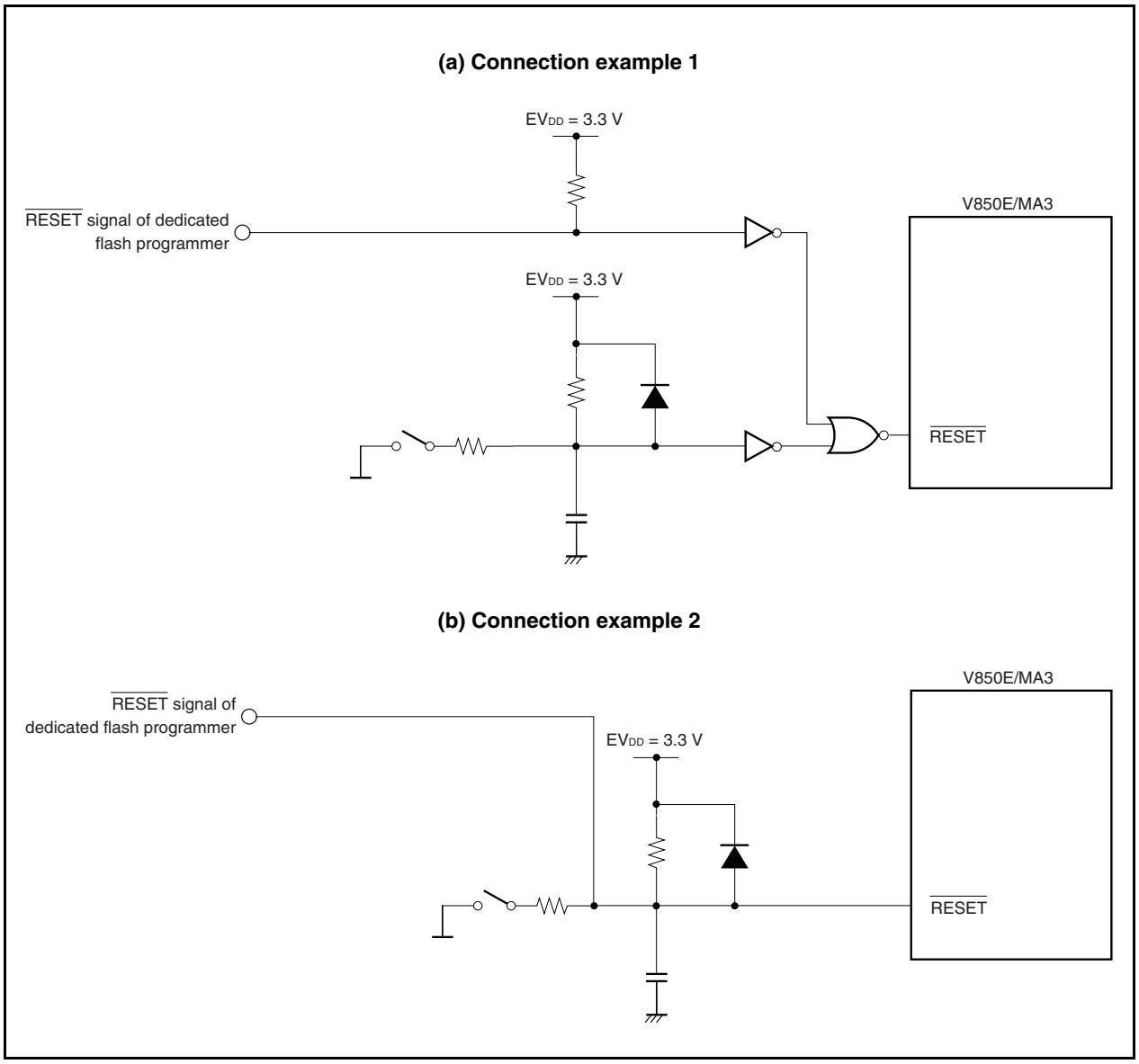
When the reset signal is input from the user system in flash memory programming mode, the programming operations will not be performed correctly. Therefore, do not input signals other than the reset signal from the dedicated flash programmer.

Figure 25-8. Conflict of Signals ($\overline{\text{RESET}}$ Pin)



Connect the reset signal of the dedicated flash programmer to the reset signal of the V850E/MA3 at the location where the two reset signals are the same.

Figure 25-9. Example of $\overline{\text{RESET}}$ Pin Connection



25.5.4 NMI pin

Do not change the signal input to the NMI pin in flash memory programming mode. If it is changed in flash memory programming mode, programming may not be performed correctly.

25.5.5 MODE0, MODE1 pins

If MODE0 is set as a low-level input and MODE1 is set as a high-level input, when reset is released, these pins change to the flash memory programming mode.

25.5.6 Port pins

When the flash memory programming mode is set, all the port pins except the pins that communicate with the dedicated flash programmer become output high-impedance. These pins must be connected according to the recommended connection of unused pins (see **2.3 Pin I/O Circuits and Recommended Connection of Unused Pins**). If a problem such as prohibition of output high-impedance state occurs in the external device connected to the port, connect the port to V_{DD} or V_{SS} via a resistor.

25.5.7 Other signal pins

Connect X1 and X2 in the same status as in the normal operation mode.

25.5.8 Power supply

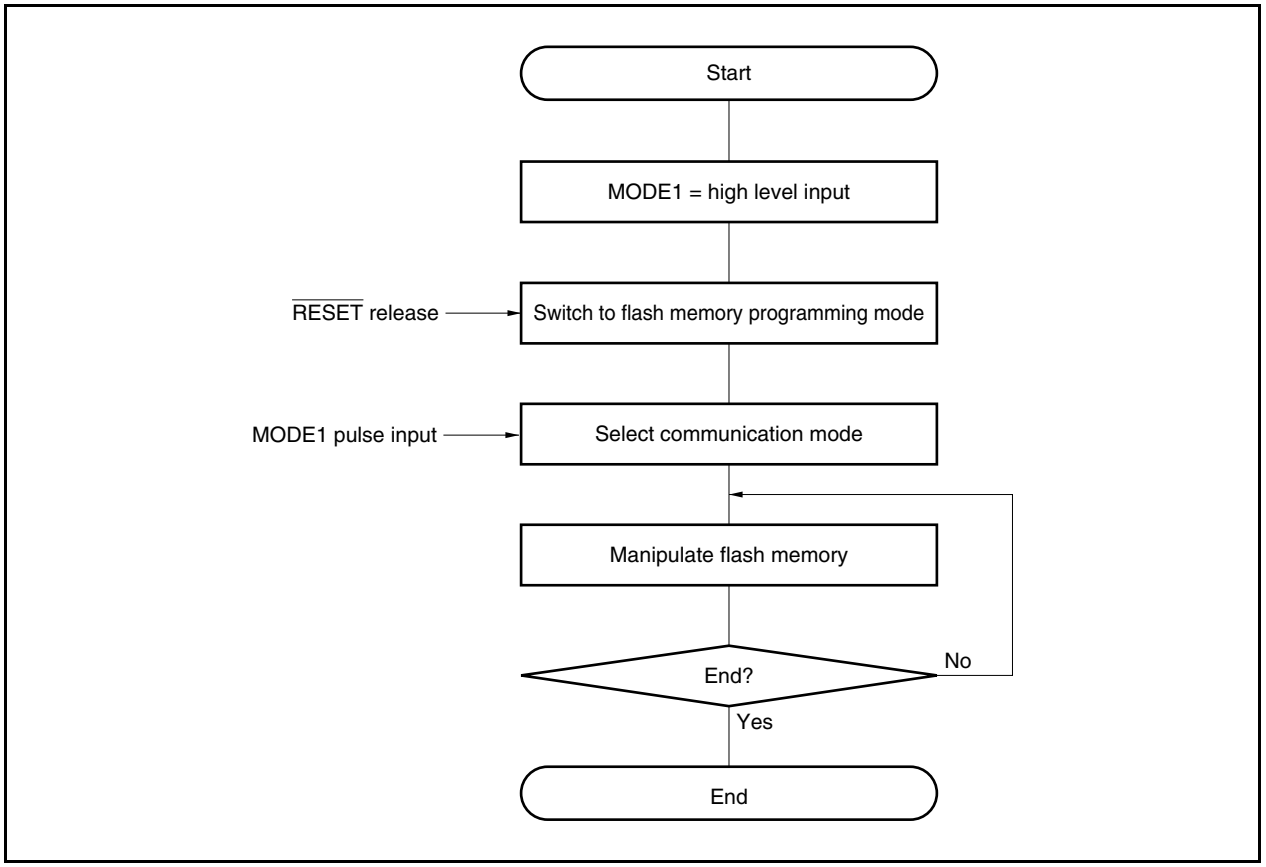
Supply the same power (V_{DD}, V_{SS}, AV_{DD0}, AV_{SS0}, AV_{DD1}, AV_{SS1}, CV_{DD}, CV_{SS}, EV_{DD}, and EV_{SS}) as in normal operation mode. Connect V_{DD}, GND, and V_{DD2} of the dedicated flash programmer to EV_{DD}, EV_{SS}, and V_{DD}. (V_{DD} and V_{DD2} of the dedicated flash programmer are provided with a power supply monitoring function.)

25.6 Programming Method

25.6.1 Flash memory control

The following shows the procedure for manipulating the flash memory.

Figure 25-10. Flash Memory Operation Flow



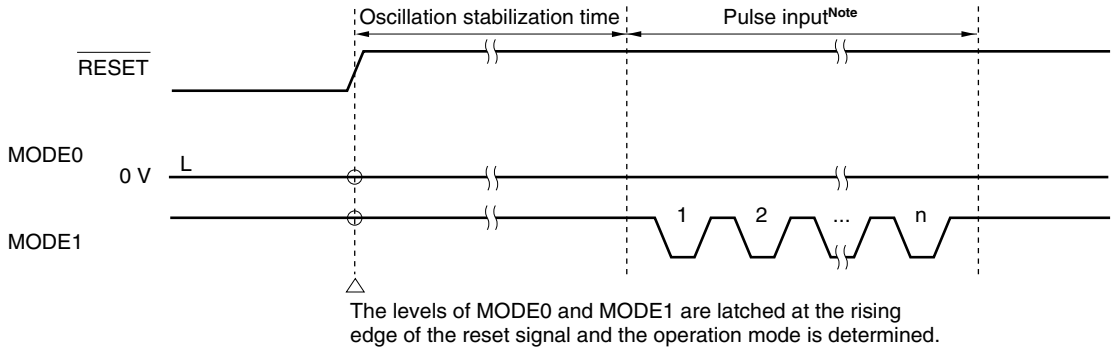
25.6.2 Flash memory programming mode

When rewriting the contents of flash memory using the dedicated flash programmer, set the V850E/MA3 to the flash memory programming mode. To switch to this mode, set the MODE0 and MODE1 pins before releasing reset.

When performing on-board writing, switch modes using a jumper, etc.

- MODE0: Low-level input
- MODE1: High-level input

Figure 25-11. Flash Memory Programming Mode



Note For the number of pulses, see **Table 25-7**.

25.6.3 Selection of communication mode

In the V850E/MA3, the communication mode is selected by inputting pulses (11 pulses max.) to the MODE1 pin after switching to the flash memory programming mode. The MODE1 pulse is generated by the dedicated flash programmer.

The following shows the relationship between the number of pulses and the communication mode.

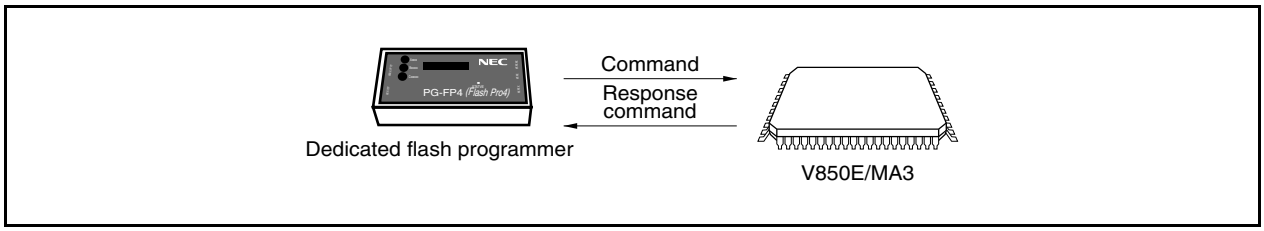
Table 25-7. List of Communication Modes

MODE1 Pulses	Communication Mode	Remarks
0	UARTA0	Transfer rate: 9,600 bps (at reset), LSB first
8	CSIB0	V850E/MA3 performs slave operation, MSB first
11	CSIB0 + HS	
Other	RFU (reserved)	Setting prohibited

25.6.4 Communication commands

The V850E/MA3 communicates with the dedicated flash programmer by means of commands. A command sent from the dedicated flash programmer to the V850E/MA3 is called the “command”. The response signal sent from the V850E/MA3 to the dedicated flash programmer is called the “response command”.

Figure 25-12. Communication Command



The following shows the commands for controlling the flash memory of the V850E/MA3. All of these commands are issued from the dedicated flash programmer, and the V850E/MA3 performs the various processing corresponding to the commands.

Table 25-8. Flash Memory Control Command

Category	Command Name	Function
Verify	Verify command	Compares the contents of the entire memory and the input data.
Erase	Chip erase command	Erases the contents of the entire memory.
	Block erase command	Erases the contents of the specified block.
Blank check	Block blank check command	Checks the erase state of the specified block.
Data write	Write command	Writes data by the specification of the write address and the number of bytes to be written, and executes verify check.
System setting and control	Status command	Acquires the status of operations.
	Oscillating frequency setting command	Sets the oscillation frequency.
	Baud rate setting command	Sets the baud rate when UARTA0 is used
	Silicon signature command	Reads outs the silicon signature information.
	Version acquisition command	Read outs the device version information.
	Check sum command	Read outs the data check sum value of the specified block.
	Reset command	Escapes from each state.

The V850E/MA3 sends back response commands for the commands issued from the dedicated flash programmer. The following shows the response commands the V850E/MA3 sends out.

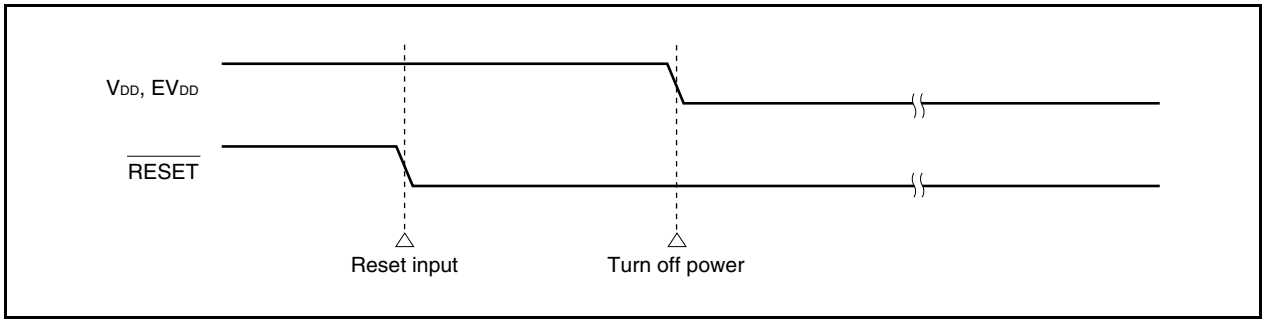
Table 25-9. Response Commands

Response Command Name	Function
ACK (acknowledge)	Acknowledges command/data, etc.
NAK (not acknowledge)	Acknowledges illegal command/data, etc.

25.6.5 Turning off power

When all the communication commands have been executed or to stop processing during rewrite processing, make the $\overline{\text{RESET}}$ pin low and then turn off power (V_{DD}, EV_{DD}) as shown below.

Figure 25-13. Turning Off Power



CHAPTER 26 ELECTRICAL SPECIFICATIONS

26.1 Normal Operation Mode

Absolute Maximum Ratings (T_A = 25°C)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}	V_{DD} pin	−0.5 to +3.6	V
	V_{SS}	V_{SS} pin	−0.5 to +0.5	V
	EV_{DD}	EV_{DD} pin	−0.5 to +4.6	V
	EV_{SS}	EV_{SS} pin	−0.5 to +0.5	V
	CV_{DD}	CV_{DD} pin	−0.5 to +4.6	V
	CV_{SS}	CV_{SS} pin	−0.5 to +0.5	V
	AV_{DD0}	AV_{DD0} pin, EV_{DD} − 0.5 V < AV_{DD0} < EV_{DD} + 0.5 V	−0.5 to +4.6	V
	AV_{SS0}	AV_{SS0} pin	−0.5 to +0.5	V
	AV_{DD1}	AV_{DD1} pin, EV_{DD} − 0.5 V < AV_{DD1} < EV_{DD} + 0.5 V	−0.5 to +4.6	V
	AV_{SS1}	AV_{SS1} pin	−0.5 to +0.5	V
Input voltage	V_I	Excluding X1 pin, V_{DD} = 2.5 V ± 0.2 V, EV_{DD} = CV_{DD} = AV_{DD0} = AV_{DD1} = 3.3 V±0.3 V	−0.5 to EV_{DD} + 0.5[Note]	V
Clock input voltage	V_K	X1 pin, V_{DD} = 2.5 V ±0.2 V, CV_{DD} = 3.3 V ±0.3 V	−0.5 to CV_{DD} + 0.5[Note]	V
Output current, low	I_{OL}	Per pin	4.0	mA
		Total of all pins	100	mA
Output current, high	I_{OH}	Per pin	−4.0	mA
		Total of all pins	−100	mA
Output voltage	V_O	V_{DD} = 2.5 V ± 0.2 V, EV_{DD} = CV_{DD} = AV_{DD0} = AV_{DD1} = 3.3 V ±0.3 V	−0.5 to EV_{DD} + 0.5[Note]	V
Analog input voltage	V_{IAN}	P70/ANI0 to P77/ANI7 pins, V_{DD} = 2.5 V ±0.2 V, AV_{DD0} = 3.3 V ±0.3 V	−0.5 to AV_{DD0} + 0.5[Note]	V
Analog output voltage	V_{OAN}	P80/ANO0, P81/ANO1, V_{DD} = 2.5 V ±0.2 V, AV_{DD1} = 3.3 V ±0.3 V	−0.5 to AV_{DD1} + 0.5[Note]	V
Operating ambient temperature	T_A		−40 to +85	°C
Storage temperature	T_{stg}	144-pin plastic LQFP version	−55 to +150	°C
		161-pin plastic FBGA version	−40 to +125	°C

Note Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

Cautions 1. Do not directly connect the output (or I/O) pins of IC products to each other, or to V_{DD}, 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} = CV_{DD} = CV_{SS} = AV_{DD0} = AV_{SS0} = AV_{DD1} = AV_{SS1} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	fc = 1 MHz Unmeasured pins returned to 0 V.			15	pF
I/O capacitance	C_{IO}				15	pF
Output capacitance	C_O				15	pF

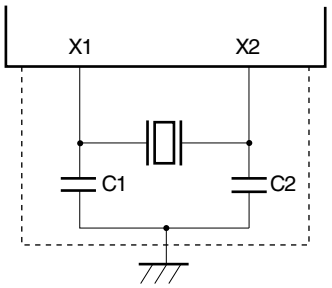
Operating Conditions

Operation Mode	System clock Frequency (f_{XX})	Operating Ambient Temperature (T_A)	Supply Voltage (V_{DD})
Clock through mode	5 to 25 MHz	−40 to +85°C	V_{DD} = 2.5 V ±0.2 V EV_{DD} = 3.3 V ±0.3 V CV_{DD} = 3.3 V ±0.3 V AV_{DD0} = 3.3 V ±0.3 V AV_{DD1} = 3.3 V ±0.3 V
PLL mode	5 to 80 MHz		

Recommended Oscillator

(a) Ceramic resonator

(i) Kyocera (T_A = –40 to +85°C)



Type	Part Number	Oscillation frequency f_X (MHz)	Recommended Circuit Constant		Oscillation Voltage Range		Oscillation Stabilization Time (MAX.) T_{OS} (ms)
			C1	C2	MIN. (V)	MAX. (V)	
Surface mounting	PBRC4.00HR	4.00	On chip (30 pF)	On chip (30 pF)	3.0	3.6	0.04
	PBRC5.00HR	5.00	On chip (30 pF)	On chip (30 pF)	3.0	3.6	0.02
	PBRC6.60HR	6.60	On chip (30 pF)	On chip (30 pF)	3.0	3.6	0.03
	PBRC8.00HR	8.00	On chip (30 pF)	On chip (30 pF)	3.0	3.6	0.03

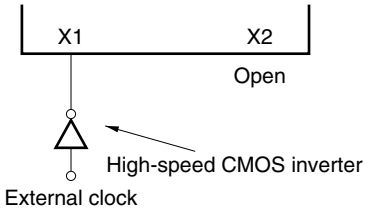
Cautions 1. Keep the oscillator as close to the X1 and X2 pins as possible.

2. Do not route other signal lines through the area enclosed by broken lines.

3. 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/MA3 within the DC characteristics and AC characteristics for internal operation conditions.

(b) External clock



Cautions 1. Keep the high-speed CMOS as close to the X1 pin as possible.

2. Sufficiently evaluate the matching of the V850E/MA3 and high-speed CMOS inverter.

DC Characteristics

(T_A = –40 to +85°C, V_{DD} = 2.5 V ±0.2 V, EV_{DD} = CV_{DD} = AV_{DD0} = AV_{DD1} = 3.3 V ±0.3 V, V_{SS} = EV_{SS} = CV_{SS} = AV_{SS0} = AV_{SS1} = 0 V) (1/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH}	Pins other than **Notes 1, 2, 3, 4, 5**	2.0		3.6	V
		Note 1	$0.8EV_{DD}$		3.6	V
		Notes 3, 5	$0.7EV_{DD}$		3.6	V
		Note 2	$0.7AV_{DD0}$		3.6	V
		Note 4	$0.7AV_{DD1}$		3.6	V
Input voltage, low	V_{IL}	Pins other than **Notes 1, 2, 3, 4, 5**	0		0.8	V
		Note 1	0		$0.2EV_{DD}$	V
		Notes 3, 5	0		$0.3EV_{DD}$	V
		Note 2	0		$0.3AV_{DD0}$	V
		Note 4	0		$0.3AV_{DD1}$	V
Clock input voltage, high	V_{XH}	X1 pin	$0.7CV_{DD}$		CV_{DD}	V
Clock input voltage, low	V_{XL}	X1 pin	0		$0.3CV_{DD}$	V
Shumitt-triggered input threshold voltage	V_{T^+}	**Note 1**, falling edge		2.0		V
	V_{T^-}	**Note 1**, falling edge		1.3		V
Output voltage, high	V_{OH}	I_{OH} = –1.0 mA	EV_{DD} – 1.0			V
		I_{OH} = –100 μA	EV_{DD} – 0.5			V
Output voltage, low	V_{OL}	I_{OL} = 3 mA			0.4	V
Input leakage current, high	I_{LIH}	$V_I = EV_{DD}$, pins other than **Notes 2, 4**			10	μA
Input leakage current, low	I_{LIL}	V_I = 0 V, pins other than **Notes 2, 4**			–10	μA
Output leakage current, high	I_{LOH}	$V_O = EV_{DD}$			10	μA
Output leakage current, low	I_{LOL}	V_O = 0 V			–10	μA
Analog pin input leakage current	I_{LWASN}	**Notes 2, 4**			±10	μA

Notes 1. P00, P01, P04 to P07, P10 to P15, P20 to P22, P24 to P26, P30 to P34, P37, P41, P42, P44, P45, P50, P51, $\overline{TRST}$, $\overline{RESET}$, CKSEL, PSEL, MODE0, MODE1 pins and their alternate-function pins. However, these pins do not have hysteresis characteristics in port mode.

2. P70 to P77 pins and their alternate-function pins.

3. P27 pin and its alternate-function pins.

4. P80, P81 pins and their alternate-function pins.

5. PCM1, PCD1 pins and their alternate-function pins.

Remark The TYP. value is a reference value at T_A = 25°C, V_{DD} = 2.5 V, EV_{DD} = 3.3 V. Current flowing through the pull-up resistor is not included.

DC Characteristics (T_A = –40 to +85°C, V_{DD} = 2.5 V ±0.2 V, EV_{DD} = CV_{DD} = AV_{DD0} = AV_{DD1} = 3.3 V ±0.3 V, V_{SS} = EV_{SS} = CV_{SS} = AV_{SS0} = AV_{SS1} = 0 V) (2/2)

	Parameter		Symbol	Conditions		MIN.	TYP.	MAX.	Unit
<R>	Supply current	In normal operation mode	I_{DD1}	V_{DD} pin	Mask ROM versions		1.25 × fxx + 20	1.5 × fxx + 60	mA
					Flash memory versions		2.25 × fxx + 50	3.2 × fxx + 94	mA
			I_{EDD1}	EV_{DD} + CV_{DD} pins			0.9 × fxx + 5	0.95 × fxx + 24	mA
<R>		In HALT mode	I_{DD2}	V_{DD} pin	Mask ROM versions		1.1 × fxx + 24	1.2 × fxx + 44	mA
					Flash memory versions		1.1 × fxx + 30	1.2 × fxx + 54	mA
			I_{EDD2}	EV_{DD} + CV_{DD} pins			0.65 × fxx + 3	0.7 × fxx + 24	mA
		In IDLE mode	I_{DD3}	V_{DD} pin	Mask ROM versions		10	20	mA
					Flash memory versions		18	30	mA
			I_{EDD3}	EV_{DD} + CV_{DD} pins			0.35	3	mA
		In STOP mode	I_{DD4}	V_{DD} pin				850	μA
			I_{EDD4}	EV_{DD} + CV_{DD} pins				50	μA

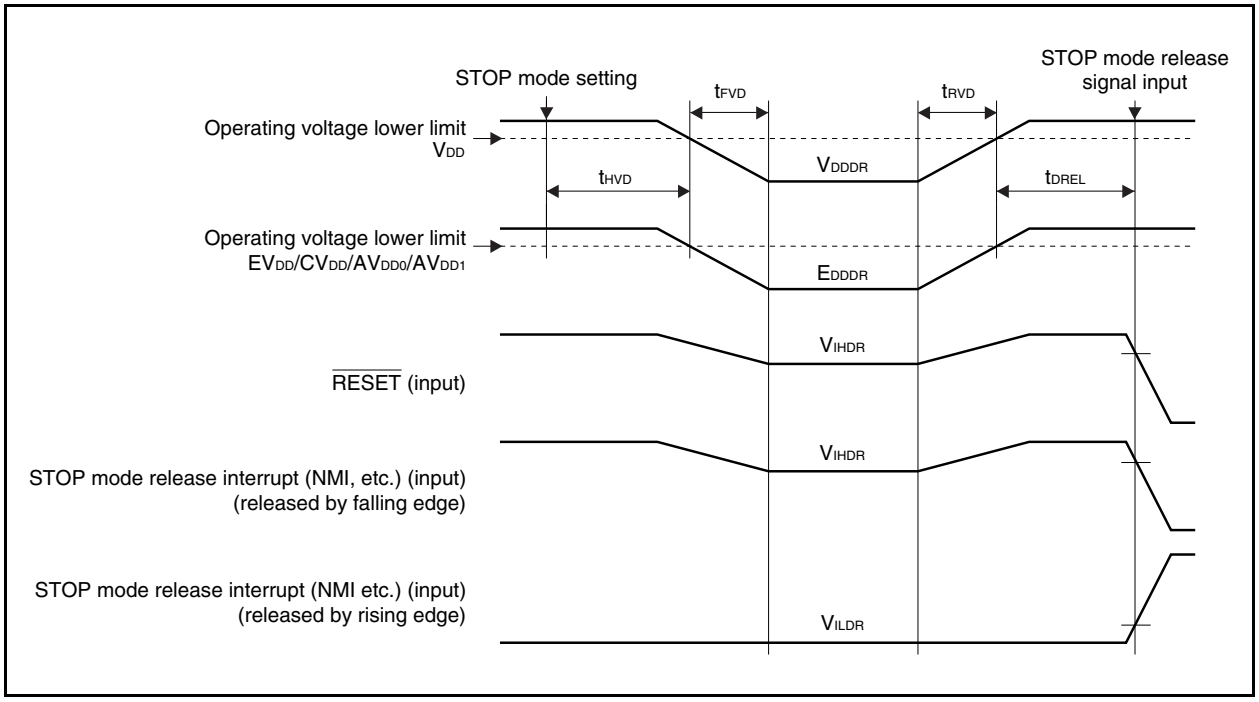
Remarks 1. The TYP. value is a reference value at T_A = 25°C, V_{DD} = 2.5 V, EV_{DD} = 3.3 V. Current flowing through the pull-up resistor is not included.

2. fxx: System clock frequency (MHz)

Data Retention Characteristics (T_A = –40 to +85°C, V_{DD} = 2.5 V ±0.2 V, EV_{DD} = CV_{DD} = AV_{DD0} = AV_{DD1} = 3.3 V ±0.3 V, V_{SS} = EV_{SS} = CV_{SS} = AV_{SS0} = AV_{SS1} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Data retention voltage	V_{DDDR}	STOP mode, V_{DD} = V_{DDDR}		1.8		2.7	V
	EV_{DDDR}	STOP mode, EV_{DD} = CV_{DD} = AV_{DD0} = AV_{DD1} = EV_{DDDR}		V_{DDDR}		3.6	V
Data retention current	I_{DDDR}	V_{DD} = V_{DDDR}	V_{DD} pin			850	μA
			EV_{DD} + CV_{DD} pins			50	μA
Supply voltage rise time	t_{RVD}			200			μs
Supply voltage fall time	t_{FVD}			200			μs
Supply voltage retention time (for STOP mode setting)	t_{HVD}			0			ms
STOP mode release signal input time	t_{DREL}			0			ns
Data retention input voltage, high	V_{IHDR}	All input pins		0.8EV_{DDDR}		EV_{DDDR}	V
Data retention input voltage, low	V_{ILDR}	All input pins		0		0.2EV_{DDDR}	V

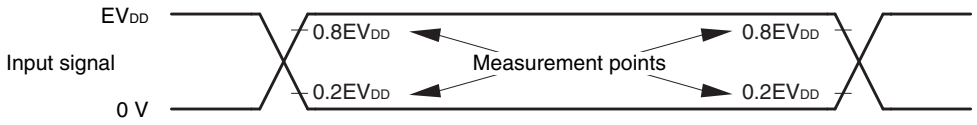
Caution Shifting to STOP mode and restoring from STOP mode must be performed within the rated operating range.



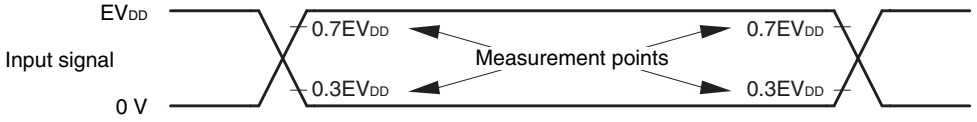
AC Characteristics (T_A = –40 to +85°C, V_{DD} = 2.5 V ±0.2 V, EV_{DD} = CV_{DD} = AV_{DD0} = AV_{DD1} = 3.3 V ±0.3 V, V_{SS} = EV_{SS} = CV_{SS} = AV_{SS0} = AV_{SS1} = 0 V, output pin load capacitance: C_L = 50 pF)

AC Test Input Measurement Points

(a) P00, P01, P04 to P07, P10 to P15, P20 to P22, P24 to P26, P30 to P34, P37, P41, P42, P44, P45, P50, P51, $\overline{\text{TRST}}$, $\overline{\text{RESET}}$, CKSEL, PSEL, MODE0, MODE1 pins and their alternate-function pins.



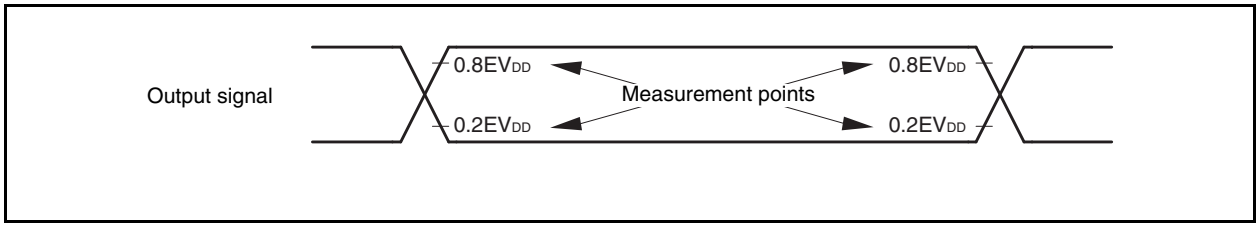
(b) P27, P70 to P77, P80, P81, PCM1, PCD0, X1 pins and their alternate-function pins.



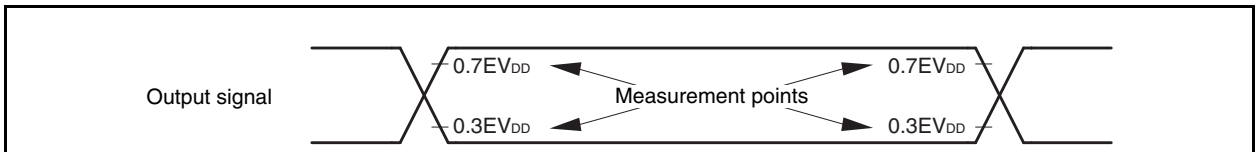
Remark EV_{DD} should be read as AV_{DD0} when the A/D input pin is used and as AV_{DD1} when the D/A port pin is used.

AC Test Output Measurement Point

(a) P00, P01, P04 to P07, P10 to P15, P20 to P22, P24 to P26, P30 to P34, P37, P40 to P45, P50, P51, $\overline{\text{TRST}}$, $\overline{\text{RESET}}$, CKSEL, PSEL, MODE0, MODE1 pins and their alternate-function pins



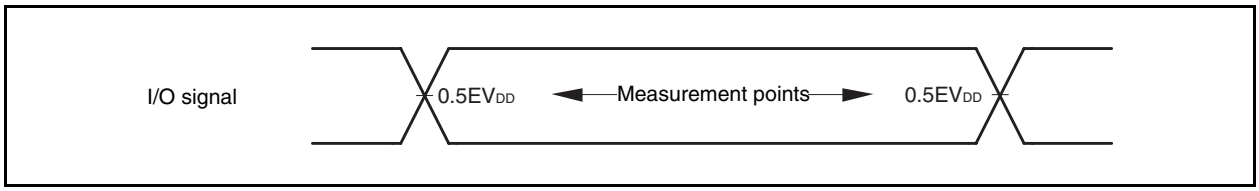
(b) P27, P70 to P77, P80, P81, PCM1, PCD0 pins and their alternate-function pins



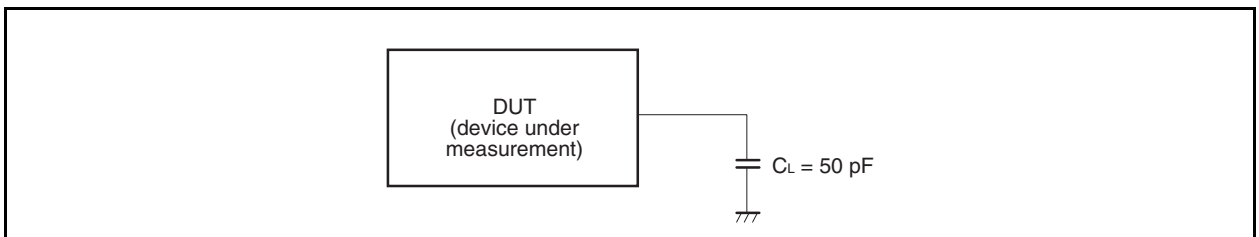
Remark EV_{DD} should be read as AV_{DD1} when the D/A output pin is used.

AC Test Bus Access Measurement Points

Bus control pins during accessing external bus (A0 to A25, AD0 to AD15, $\overline{\text{CS0}}$ to $\overline{\text{CS7}}$, $\overline{\text{IOWR}}$, $\overline{\text{IORD}}$, $\overline{\text{LBE}}/\overline{\text{LWR}}$/LDQM, $\overline{\text{UBE}}/\overline{\text{UWR}}$/UDQM, $\overline{\text{RD}}$, $\overline{\text{WR}}/\overline{\text{WE}}$, $\overline{\text{ASTB}}$, $\overline{\text{BCYST}}$, $\overline{\text{WAIT}}$, $\overline{\text{BUSCLK}}$, $\overline{\text{HLDAK}}$, $\overline{\text{HLDRQ}}$, $\overline{\text{REFRQ}}$, $\overline{\text{SDCKE}}$, SDCLK, SDCAS, $\overline{\text{SDRAS}}$, $\overline{\text{DMAAK0}}$ to $\overline{\text{DMAAK3}}$, $\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$)



Load Conditions



Caution If the load capacitance exceeds 50 pF due to the circuit configuration, bring the load capacitance of the device to 50 pF or less by inserting a buffer or by some other means.

(1) Clock timing (1/2)

Parameter	Symbol		Conditions		MIN.	MAX.	Unit
X1 input cycle	<1>	t_{CYX}	Clock through mode		40	200	ns
			PLL mode	PSEL = L	181.8	250	ns
				PSEL = H	125	181.8	ns
X1 input high-level width	<2>	t_{WXH}	Clock through mode		16		ns
			PLL mode		50		ns
X1 input low-level width	<3>	t_{WXL}	Clock through mode		16		ns
			PLL mode		50		ns
X1 rise time	<4>	t_{XR}	Clock through mode			4	ns
			PLL mode			6	ns
X1 fall time	<5>	t_{XF}	Clock through mode			4	ns
			PLL mode			6	ns
CPU operating frequency	–	f_{CPU}			5	80	MHz
BUSCLK output cycle	<6>	t_{CYK1}			20	200	ns
BUSCLK high-level width	<7>	t_{WKH1}			0.5T – 6		ns
BUSCLK low-level width	<8>	t_{WKL1}			0.5T – 6		ns
BUSCLK rise time	<9>	t_{KR1}				6	ns
BUSCLK fall time	<10>	t_{KF1}				6	ns
SDCLK output cycle	<11>	t_{CYK2}			20	200	ns
SDCLK high-level width	<12>	t_{WKH2}			0.5T – 6		ns
SDCLK low-level width	<13>	t_{WKL2}			0.5T – 6		ns
SDCLK rise time	<14>	t_{KR2}				6	ns
SDCLK fall time	<15>	t_{KF2}				6	ns

Remarks 1. $T = t_{CYKn}$ (n = 1, 2)

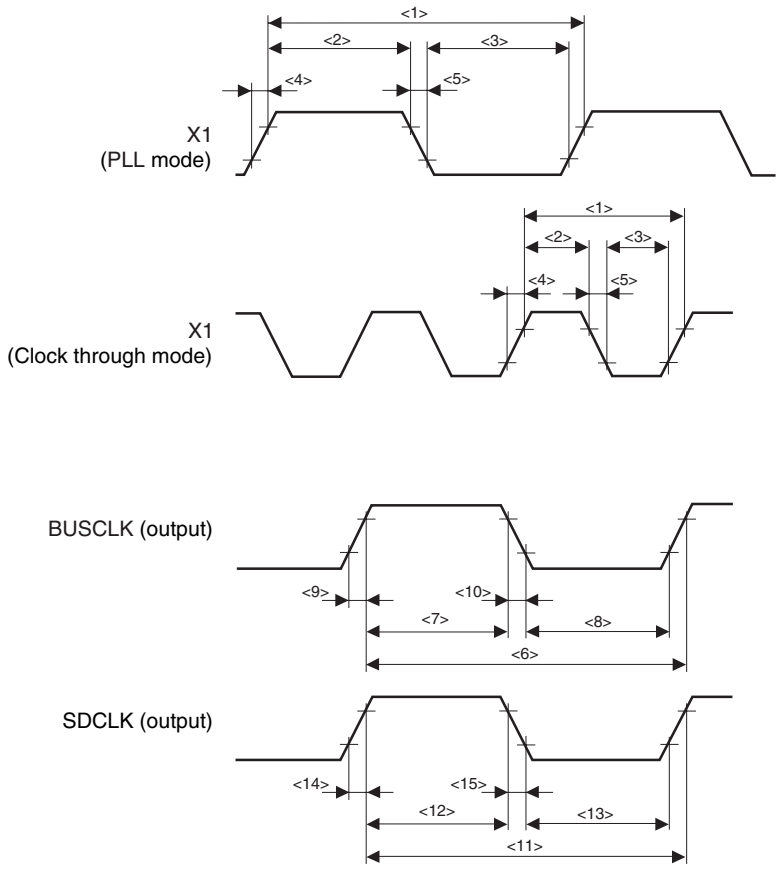
2. The phase difference between BUSCLK and SDCLK cannot be defined.

3. X1↓ → SDCLK delay time and X1↓ → BUSCLK delay time cannot be defined.

4. The division ratio of BUSCLK is set by the BMC.CKM1 and BMC.CKM0 bits.

<R> **5.** The values of BUSCLK and SDCLK in the above specifications are when a clock with a duty factor of 1:1 is input. If the duty factor of the X1 input clock changes in the clock through mode, the duty factor of BUSCLK and SDCLK also changes, affecting the timing of memory access. In the PLL mode, the memory access timing is not affected.

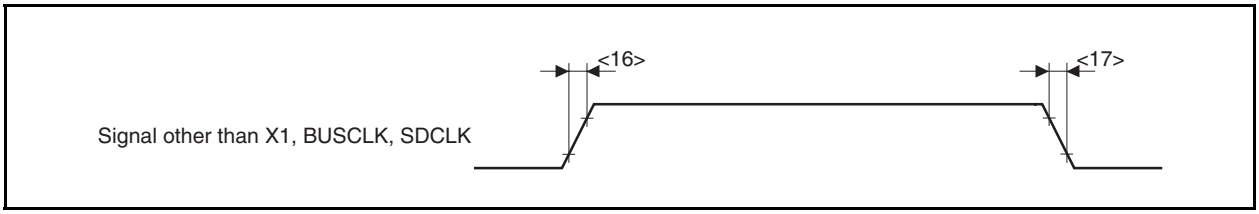
(1) Clock timing (2/2)



Remark BUSCLK cycle differs depending on the bus cycle.

(2) Output waveform (other than X1, BUSCLK, SDCLK)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output rise time	<16>	t_{OR}			6	ns
Output fall time	<17>	t_{OF}			6	ns

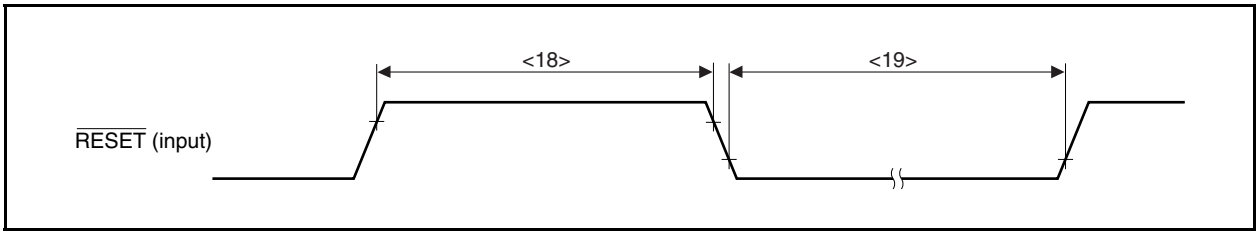


(3) Reset timing

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ pin high-level width	<18>	t_{WRSH}		500		ns
$\overline{\text{RESET}}$ pin low-level width	<19>	t_{WRSL}	When power is on, STOP mode released	500 + T_{OS}		ns
			Other than when power is on, STOP mode released	500		ns

Caution Sufficiently evaluate the oscillation stabilization time.

Remark T_{OS}: Oscillation stabilization time



(4) SRAM, external ROM, external I/O access timing (in separate bus mode)

(a) Access timing (SRAM, external ROM, external I/O) (1/2)

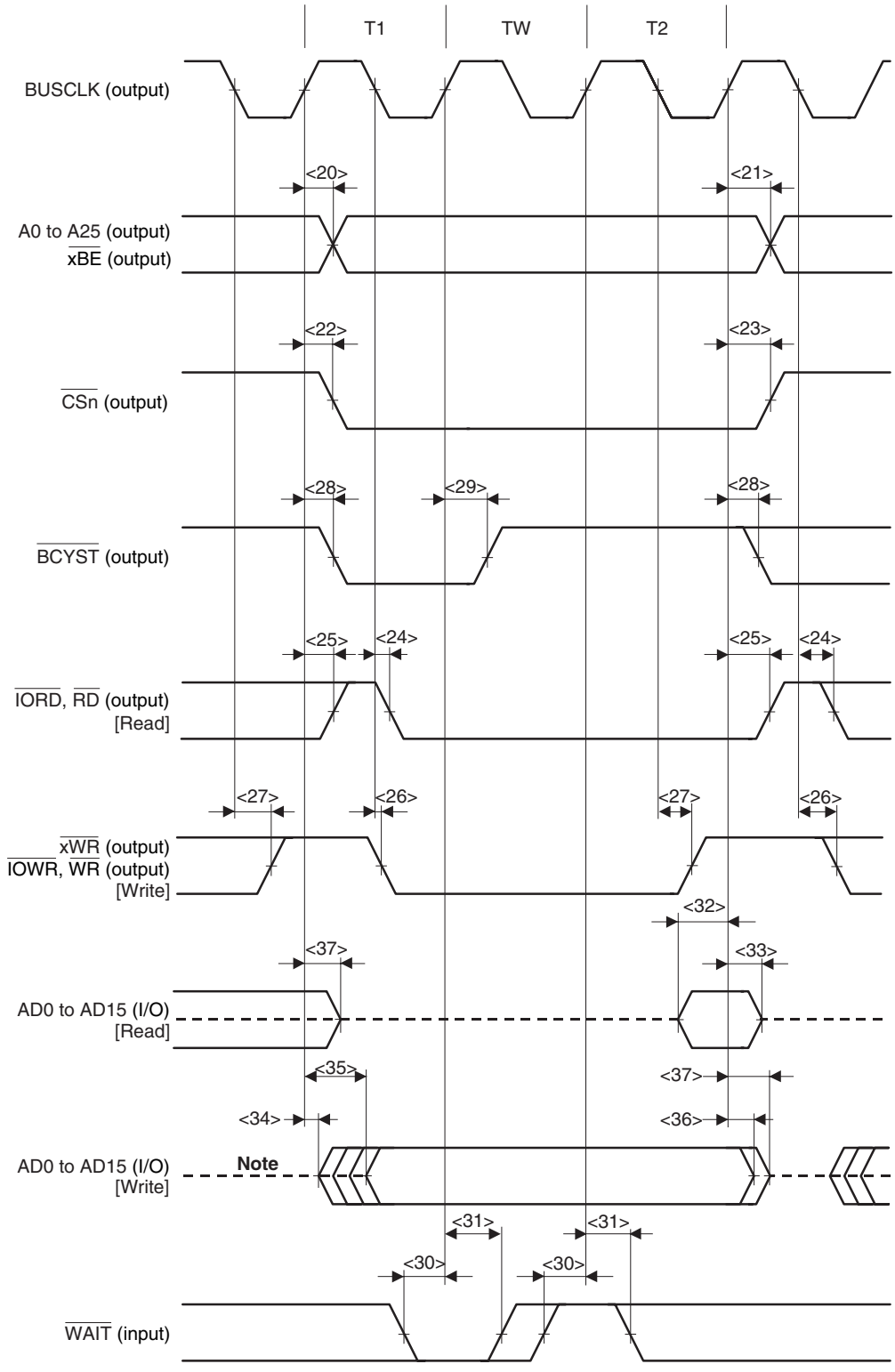
Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address, $\overline{xBE}$ output delay time (from BUSCLK↑)	<20>	t_{DKA}		0	15	ns
Address, $\overline{xBE}$ output hold time (from BUSCLK↑)	<21>	t_{HKA}		0	13	ns
$\overline{CSn}$↓ delay time (from BUSCLK↑)	<22>	t_{DKCSL}		0	15	ns
$\overline{CSn}$↑ delay time (from BUSCLK↑)	<23>	t_{HKCSH}		0	13	ns
$\overline{RD}$, $\overline{IORD}$↓ delay time (from BUSCLK↓)	<24>	t_{DKRDL}		0	13	ns
$\overline{RD}$, $\overline{IORD}$↑ delay time (from BUSCLK↑)	<25>	t_{HKRDH}		0	13	ns
$\overline{xWR}$, $\overline{IOWR}$, $\overline{WR}$↓ delay time (from BUSCLK↓)	<26>	t_{DKWRL}		0	13	ns
$\overline{xWR}$, $\overline{IOWR}$, $\overline{WR}$↑ delay time (from BUSCLK↓)	<27>	t_{HKWRH}		0	13	ns
$\overline{BCYST}$↓ delay time (from BUSCLK↑)	<28>	t_{DKBSL}		0	13	ns
$\overline{BCYST}$↑ delay time (from BUSCLK↑)	<29>	t_{HKBSH}		0	13	ns
$\overline{WAIT}$ setup time (to BUSCLK↑)	<30>	t_{SWK}		12		ns
$\overline{WAIT}$ hold time (from BUSCLK↑)	<31>	t_{HKW}		0		ns
Data input setup time (to BUSCLK↑)	<32>	t_{SKID}		13		ns
Data input hold time (from BUSCLK↑)	<33>	t_{HKID}		0		ns
Data output delay time (from BUSCLK↑)	<34>	t_{DKOD1}		0		ns
	<35>	t_{DKOD2}		0	16	ns
Data float delay time (from BUSCLK↑)	<36>	t_{HKOD1}		0		ns
	<37>	t_{HKOD2}			16	ns

Remarks 1. Satisfy at least one of data input hold times t_{HRDID} or t_{HKID}.

2. n = 0 to 7

x = U, L

(a) Access timing (SRAM, external ROM, external I/O) (2/2)



Note The T0 cycle is inserted before the T1 cycle.

Remarks 1. This is the timing when the number of waits set by the DWC0 and DWC1 registers is 0.

2. The broken lines indicate high impedance.

3. n = 0 to 7, x = U, L

(b) Read timing (SRAM, external ROM, external I/O) (1/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Data input setup time (to address, $\overline{xBE}$, $\overline{CSn}$)	<38>	t_{SAID}	**Note 1**		$(2 + w + w_D + w_{AS})T - 25$	ns
Data input setup time (to $\overline{RD}$, $\overline{IORD}$)	<39>	t_{SRDID}			$(1.5 + w + w_D)T - 25$	ns
$\overline{RD}$, $\overline{IORD}$ low-level width	<40>	t_{WRDL}		$(1.5 + w + w_D)T - 4$		ns
$\overline{RD}$, $\overline{IORD}$ high-level width	<41>	t_{WRDH}		$(0.5 + w_{AS} + i)T - 4$		ns
Delay time from address, $\overline{xBE}$, $\overline{CSn}$ to $\overline{RD}$, $\overline{IORD}$↓	<42>	t_{DARD}	**Note 1**	$(0.5 + w_{AS})T - 6$		ns
Delay time from $\overline{RD}$, $\overline{IORD}$↑ to address, $\overline{xBE}$	<43>	t_{DRDA}		$iT - 4$		ns
Data input hold time (from $\overline{RD}$, $\overline{IORD}$↑)	<44>	t_{HRDID}		0		ns
Delay time from $\overline{RD}$, $\overline{IORD}$↑ to data output	<45>	t_{DRDOD}		$(1 + i)T - 16$		ns
$\overline{WAIT}$ setup time (to address)	<46>	t_{SAW}	**Notes 1, 2**		$(1 + w_{AS})T - 24$	ns
$\overline{WAIT}$ setup time (to $\overline{BCYST}$↓)	<47>	t_{SBSW}	**Note 2**		$(1 + w_{AS})T - 21$	ns
<R> $\overline{WAIT}$ hold time (from $\overline{BCYST}$↑)	<48>	t_{HBSW}	**Note 2**	$w_{AS}T - 4$		ns
$\overline{WAIT}$ high-level width	<49>	t_{WWH}		$T - 4$		ns
Data output hold time (from $\overline{xWR}$, $\overline{IOWR}$, $\overline{WR}$↑)	<50>	t_{HWROD}		$(0.5 + i)T - 4$		ns

<R> **Notes 1.** When the CSDC.CSDCp bit = 1, output of the falling edge of the $\overline{CSp}$ signal is delayed by 1 cycle.

2. During the first $\overline{WAIT}$ sampling when the number of waits set by the DWC0 and DWC1 registers is 0.

Remarks 1. $T = t_{CYK1}$

2. w: Number of waits set by $\overline{WAIT}$

3. w_D: Number of waits set by the DWC0 and DWC1 registers

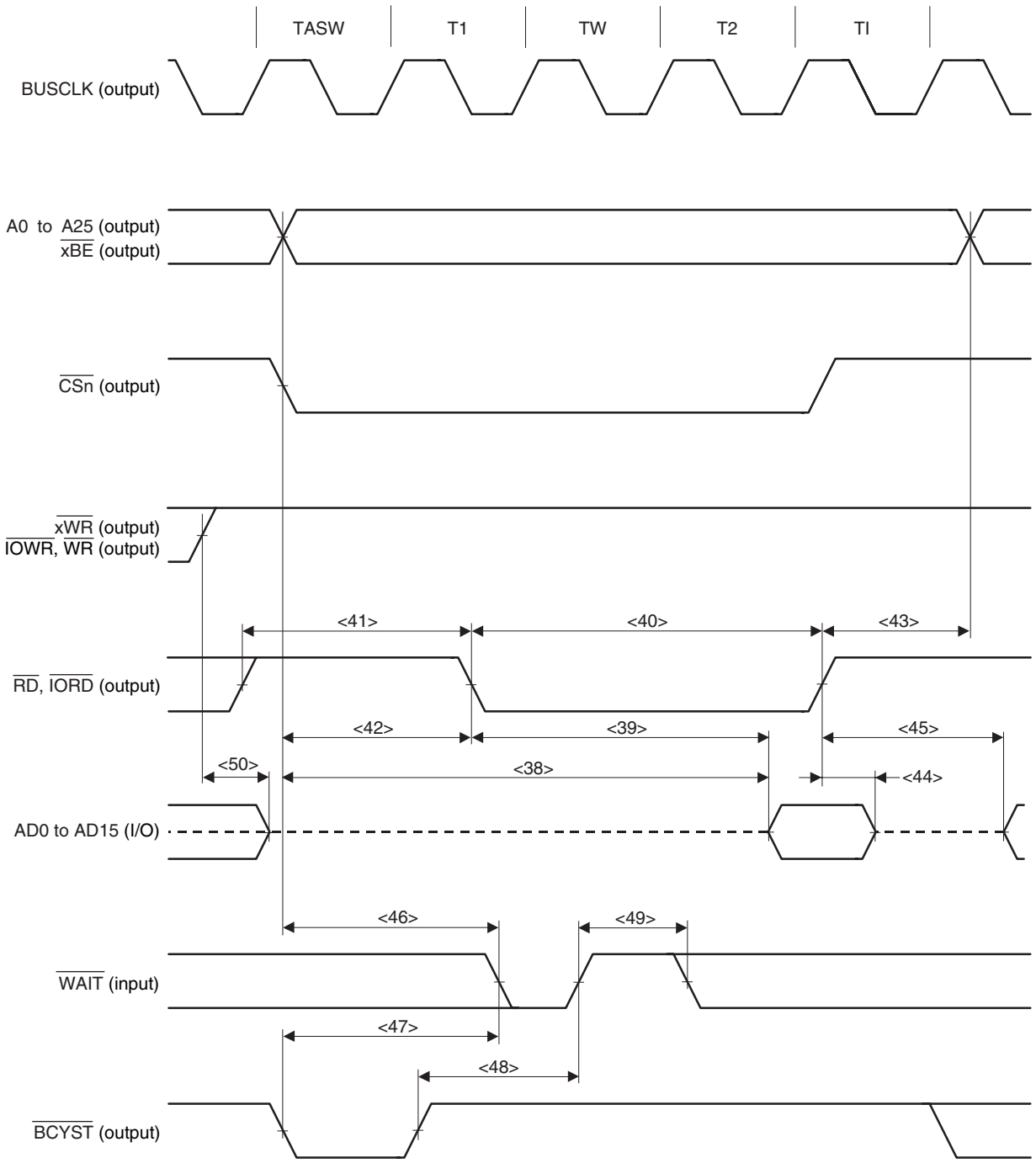
4. Satisfy at least one of data input hold times t_{HRDID} or t_{HKID}.

5. n = 0 to 7
p = 0, 4, 6, 7
x = U, L

6. i: Number of idle states

7. w_{AS}: Number of address setup waits set by the ASC register

(b) Read timing (SRAM, external ROM, external I/O) (2/2)



Remarks 1. This is the timing when the number of waits set by the DWC0 and DWC1 registers is 0, the number of idle states set by the BCC register is 1, and the number of waits set by the ASC register is 1.

2. The broken lines indicate high impedance.

3. n = 0 to 7
x = U, L

(c) Write timing (SRAM, external ROM, external I/O) (1/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{WAIT}}$ setup time (to address)	<46>	t_{SAW}	**Notes 1, 2**		$(1 + w_{AS})T - 24$	ns
$\overline{\text{WAIT}}$ setup time (to $\overline{\text{BCYST}}\downarrow$)	<47>	t_{SBSW}	**Note 1**		$(1 + w_{AS})T - 21$	ns
<R> $\overline{\text{WAIT}}$ hold time (from $\overline{\text{BCYST}}\uparrow$)	<48>	t_{HBSW}	**Note 1**	$w_{AS}T - 4$		ns
$\overline{\text{WAIT}}$ high-level width	<49>	t_{WWH}	**Note 3**	$T - 4$		ns
Delay time from address, $\overline{\text{xBE}}$, $\overline{\text{CSn}}$ to $\overline{\text{xWR}}$, $\overline{\text{IOWR}}$, $\overline{\text{WR}}\downarrow$	<51>	t_{DAWR}	**Note 2**	$(0.5 + w_{AS})T - 6$		ns
Address, $\overline{\text{xBE}}$, $\overline{\text{CSn}}$ setup time (to $\overline{\text{xWR}}$, $\overline{\text{IOWR}}$, $\overline{\text{WR}}\uparrow$)	<52>	t_{SAWR}	**Note 2**	$(1.5 + w + w_D + w_{AS})T - 6$		ns
Delay time from $\overline{\text{xWR}}$, $\overline{\text{IOWR}}$, $\overline{\text{WR}}\uparrow$ to address	<53>	t_{DWRA}		$(0.5 + i)T - 4$		ns
Delay time from $\overline{\text{xWR}}$, $\overline{\text{IOWR}}$, $\overline{\text{WR}}\uparrow$ to $\overline{\text{CSn}}$	<54>	t_{DWRCS}		$0.5T - 4$		ns
$\overline{\text{xWR}}$, $\overline{\text{IOWR}}$, $\overline{\text{WR}}$ high-level width	<55>	t_{WWRH}		$(0.5 + i + w_{AS})T - 4$		ns
$\overline{\text{xWR}}$, $\overline{\text{IOWR}}$, $\overline{\text{WR}}$ low-level width	<56>	t_{WWRL}		$(1 + w + w_D)T - 4$		ns
Data output setup time (to $\overline{\text{xWR}}$, $\overline{\text{IOWR}}$, $\overline{\text{WR}}\uparrow$)	<57>	t_{SODWR}		$(1.5 + w_{AS} + w + w_D)T - 6$		ns
Data output hold time (from $\overline{\text{xWR}}$, $\overline{\text{IOWR}}$, $\overline{\text{WR}}\uparrow$)	<50>	t_{HWROD}		$(0.5 + i)T - 4$		ns

Notes 1. During the first $\overline{\text{WAIT}}$ sampling when the number of waits set by the DWC0 and DWC1 registers is 0.

<R> **2.** When the CSDC.CSDCp bit = 1, output of the falling edge of the $\overline{\text{CSp}}$ signal is delayed by 1 cycle.

3. Time required for wait release

Remark 1. $T = t_{CYK1}$

2. w: Number of waits set by $\overline{\text{WAIT}}$

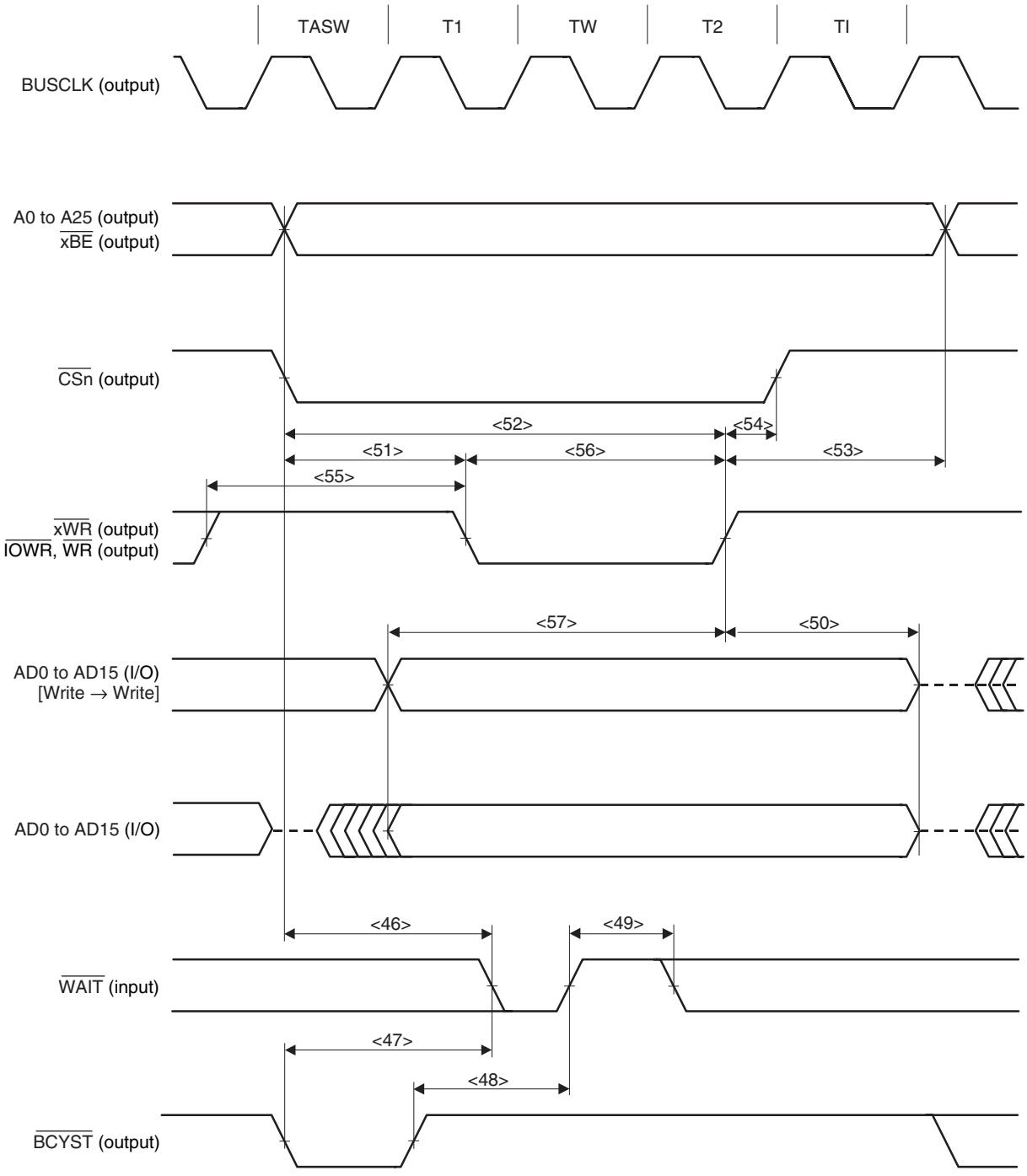
3. w_D: Number of waits set by the DWC0 and DWC1 registers

4. n = 0 to 7
p = 0, 4, 6, 7
x = U, L

5. i: Number of idle states

6. w_{AS}: Number of address setup waits set by the ASC register

(c) Write timing (SRAM, external ROM, external I/O) (2/2)



Remarks 1. This is the timing when the number of waits set by the DWC0 and DWC1 registers is 0, the number of idle states set by the BCC register is 1, and the number of waits set by the ASC register is 1.

2. The broken lines indicate high impedance.

3. n = 0 to 7
x = U, L

(d) DMA flyby transfer timing (SRAM → external I/O transfer) (1/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{WAIT}$ setup time (to BUSCLK↑)	<30>	t_{SWK}		12		ns
$\overline{WAIT}$ hold time (from BUSCLK↑)	<31>	t_{HKW}		0		ns
$\overline{RD}$ low-level width	<40>	t_{WRDL}		$(1.5 + w + w_D)T - 4$		ns
$\overline{RD}$ high-level width	<41>	t_{WRDH}		$(0.5 + i + w_{AS})T - 4$		ns
Delay time from address, $\overline{xBE}$, $\overline{CSn}$ to $\overline{RD}$↓	<42>	t_{DARD}	**Note 1**	$(0.5 + w_{AS})T - 6$		ns
Delay time from $\overline{RD}$↑ to address, $\overline{xBE}$	<43>	t_{DRDA}		$iT - 4$		ns
Delay time from $\overline{RD}$↑ to data output	<45>	t_{DRDOD}		$(1 + i)T - 16$		ns
$\overline{WAIT}$ setup time (to address)	<46>	t_{SAW}	**Note 2**		$(1 + w_{AS})T - 24$	ns
$\overline{WAIT}$ setup time (to $\overline{BCYST}$↓)	<47>	t_{SBSW}	**Note 2**		$(1 + w_{AS})T - 21$	ns
<R> $\overline{WAIT}$ hold time (from $\overline{BCYST}$↑)	<48>	t_{HBSW}	**Note 2**	$w_{AS}T - 4$		ns
$\overline{WAIT}$ high-level width	<49>	t_{WWH}		$T - 4$		ns
Delay time from address, $\overline{xBE}$, $\overline{CSn}$ to $\overline{IOWR}$↓	<51>	t_{DAWR}	**Note 1**	$(0.5 + w_{AS})T - 6$		ns
Address, $\overline{xBE}$, $\overline{CSn}$ setup time (to $\overline{IOWR}$↑)	<52>	t_{SAWR}	**Note 1**	$(1.5 + w + w_D + w_{AS})T - 6$		ns
Delay time from $\overline{IOWR}$↑ to address, $\overline{xBE}$	<53>	t_{DWRA}		$(0.5 + i)T - 4$		ns
$\overline{IOWR}$ high-level width	<55>	t_{WWRH}		$(0.5 + i + w_{AS})T - 4$		ns
$\overline{IOWR}$ low-level width	<56>	t_{WWRL}		$(1 + w + w_D)T - 4$		ns
Delay time from $\overline{IOWR}$↑ to $\overline{RD}$↑	<58>	t_{DIWRRD}		$0.5T - 4$		ns
Delay time from $\overline{DMAAKm}$↓ to $\overline{IOWR}$↓	<59>	t_{DDAWR}		$(0.5 + w_{AS})T - 8$		ns
Delay time from $\overline{IOWR}$↑ to $\overline{DMAAKm}$↑	<60>	t_{DWRDA}		$(0.5 + i)T - 8$		ns

<R> **Notes 1.** When the CSDC.CSDCp bit = 1, output of the falling edge of the $\overline{CSp}$ signal is delayed by 1 cycle.

2. During the first $\overline{WAIT}$ sampling when the number of waits set by the FWC register is 0.

Remarks 1. $T = t_{CYK1}$

2. w: Number of waits set by $\overline{WAIT}$

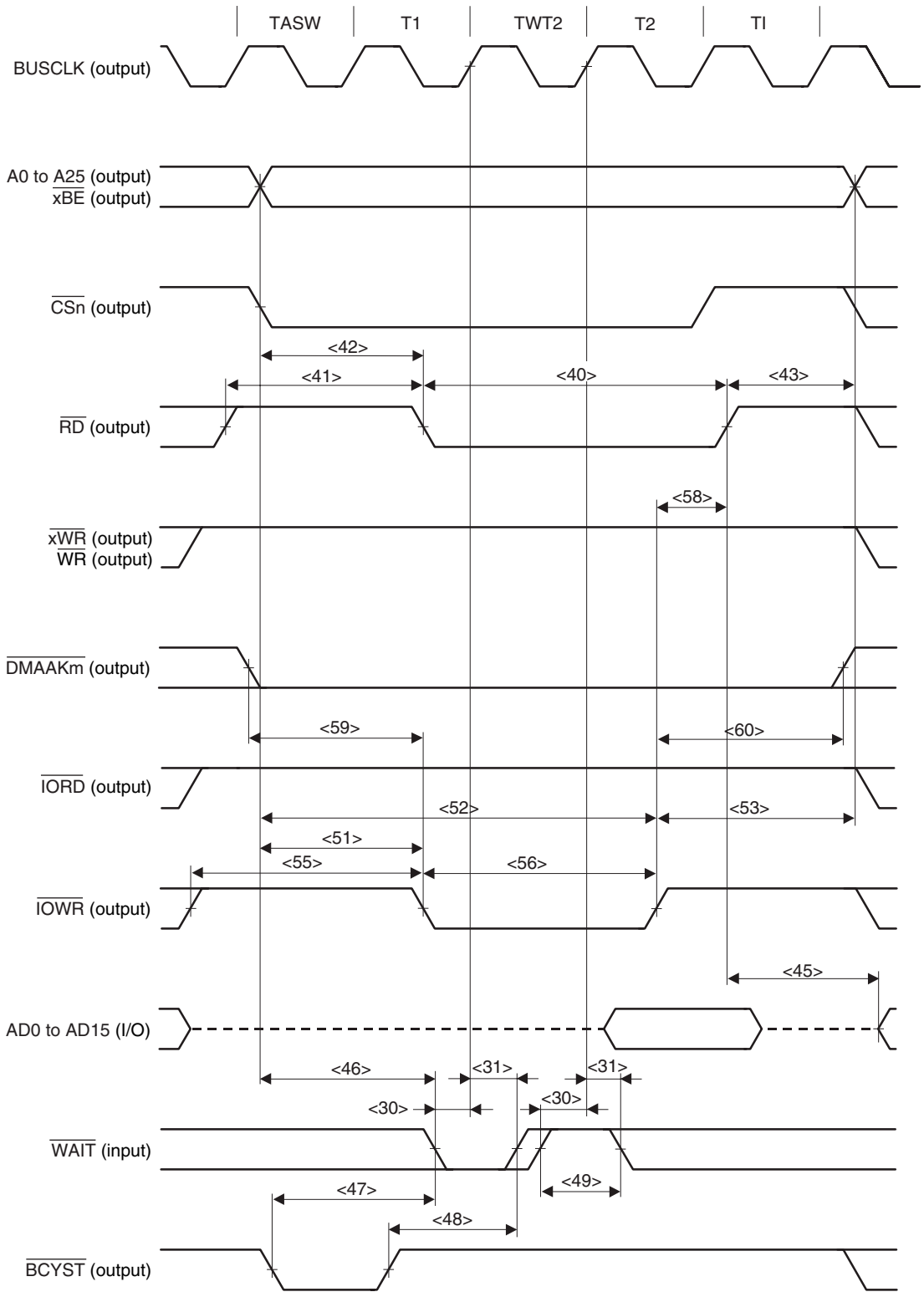
3. w_D: Number of waits set by the FWC register

4. n = 0, 1, 3, 4, 6, 7
m = 0 to 3
p = 0, 4, 6, 7
x = U, L

5. i: Number of idle states

6. w_{AS}: Number of address setup waits set by the ASC register

(d) DMA flyby transfer timing (SRAM → external I/O transfer) (2/2)



Remarks 1. This is the timing when the number of waits set by the FWC register is 0, the number of idle states set by the FIC register is 1, and the number of waits set by the ASC register is 1.

2. The broken lines indicate high impedance.

3. n = 0, 1, 3, 4, 6, 7, m = 0 to 3, x = U, L

(e) DMA flyby transfer timing (external I/O → SRAM transfer) (1/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{WAIT}$ setup time (to BUSCLK↑)	<30>	t_{SWK}		12		ns
$\overline{WAIT}$ hold time (from BUSCLK↑)	<31>	t_{HKW}		0		ns
$\overline{IORD}$ low-level width	<40>	t_{WRDL}		$(1.5 + w + w_D)T - 4$		ns
$\overline{IORD}$ high-level width	<41>	t_{WRDH}		$(0.5 + i + w_{AS})T - 4$		ns
Delay time from address, $\overline{xBE}$, $\overline{CSn}$ to $\overline{IORD}$↓	<42>	t_{DARD}	**Note 1**	$(0.5 + w_{AS})T - 6$		ns
Delay time from $\overline{IORD}$↑ to address, $\overline{xBE}$	<43>	t_{DRDA}		$iT - 4$		ns
Delay time from $\overline{IORD}$↑ to data output	<45>	t_{DRDOD}		$(1 + i)T - 16$		ns
$\overline{WAIT}$ setup time (to address)	<46>	t_{SAW}	**Note 2**		$(1 + w_{AS})T - 24$	ns
$\overline{WAIT}$ setup time (to $\overline{BCYST}$↓)	<47>	t_{SBSW}	**Note 2**		$(1 + w_{AS})T - 21$	ns
<R> $\overline{WAIT}$ hold time (from $\overline{BCYST}$↑)	<48>	t_{HBSW}	**Note 2**	$w_{AS}T - 4$		ns
$\overline{WAIT}$ high-level width	<49>	t_{WWH}		$T - 4$		ns
Delay time from address, $\overline{xBE}$, $\overline{CSn}$ to $\overline{xWR}$, $\overline{WR}$↓	<51>	t_{DAWR}	**Note 1**	$(0.5 + w_{AS})T - 6$		ns
Address, $\overline{xBE}$, $\overline{CSn}$ setup time (to $\overline{xWR}$, $\overline{WR}$↑)	<52>	t_{SAWR}	**Note 1**	$(1.5 + w + w_D + w_{AS})T - 6$		ns
Delay time from $\overline{xWR}$, $\overline{WR}$↑ to address, $\overline{xBE}$	<53>	t_{DWRA}		$(0.5 + i)T - 4$		ns
$\overline{xWR}$, $\overline{WR}$ high-level width	<55>	t_{WWRH}		$(0.5 + i + w_{AS})T - 4$		ns
$\overline{xWR}$, $\overline{WR}$ low-level width	<56>	t_{WWRL}		$(1 + w + w_D)T - 4$		ns
Delay time from $\overline{xWR}$, $\overline{WR}$↑ to $\overline{IORD}$↑	<61>	t_{DWRIRD}		$0.5T - 4$		ns
Delay time from $\overline{DMAAKm}$↓ to $\overline{IORD}$↓	<62>	t_{DDARD}		$(0.5 + w_{AS})T - 8$		ns
<R> Delay time from $\overline{IORD}$↑ to $\overline{DMAAKm}$↑	<63>	t_{DRDDA}		$iT - 4$		ns

<R> **Notes 1.** When the CSDC.CSDCp bit = 1, output of the falling edge of the $\overline{CSp}$ signal is delayed by 1 cycle.

2. During the first $\overline{WAIT}$ sampling when the number of waits set by the FWC register is 0.

Remarks 1. $T = t_{CYK1}$

2. w: Number of waits set by $\overline{WAIT}$

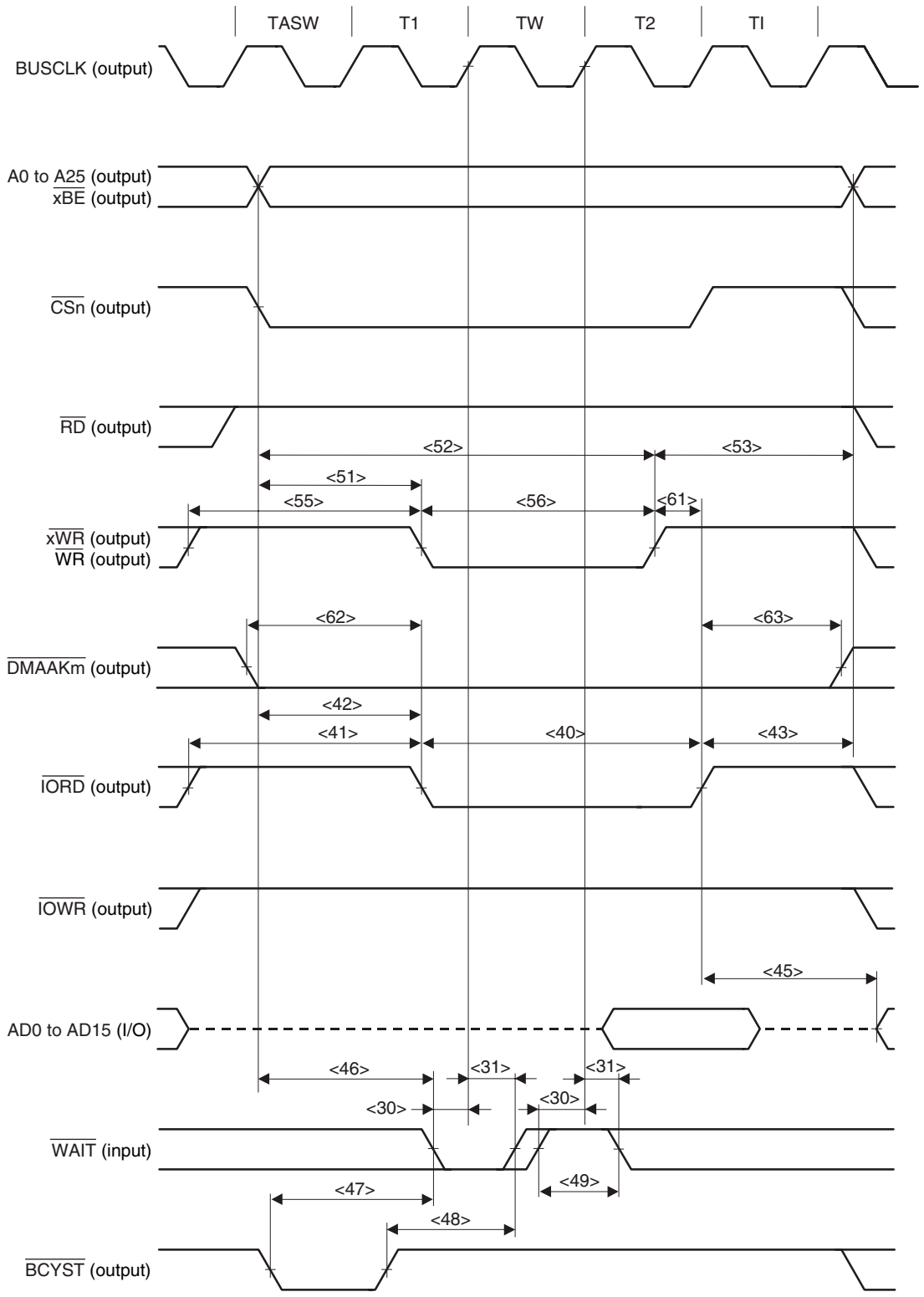
3. w_D: Number of waits set by the FWC register

4. n = 0, 1, 3, 4, 6, 7
m = 0 to 3
p = 0, 4, 6, 7
x = U, L

5. i: Number of idle states

6. w_{AS}: Number of address setup waits set by the ASC register

(e) DMA flyby transfer timing (external I/O → SRAM transfer) (2/2)



Remarks 1. This is the timing when the number of waits set by the FWC register is 0, the number of idle states set by the FIC register is 1, and the number of waits set by the ASC register is 1.

2. The broken lines indicate high impedance.

3. n = 0, 1, 3, 4, 6, 7, m = 0 to 3, x = U, L

(5) SRAM, external ROM access timing (in multiplexed bus mode)

(a) Read timing (BUSCLK asynchronous) (SRAM, external ROM) (1/3)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address, $\overline{xBE}$, $\overline{CSn}$ setup time (to ASTB↓)	<64>	t_{SAST}		$(0.5 + w_{AS})T - 8$		ns
Address hold time (from ASTB↓)	<65>	t_{HSTA}		$(0.5 + w_{AH})T - 8$		ns
Delay time from $\overline{RD}$↓ to address float	<66>	t_{FRDA}			13	ns
Data input setup time (to address, $\overline{xBE}$, $\overline{CSn}$)	<67>	t_{SAID}			$(2 + w + w_D + w_{AS} + w_{AH})T - 25$	ns
Data input setup time (to $\overline{RD}$↓)	<68>	t_{SRDID}			$(1 + w + w_D)T - 25$	ns
Delay time from ASTB↓ to $\overline{RD}$↓	<69>	t_{DSTRD}		$(0.5 + w_{AH})T - 8$		ns
Data input hold time (from $\overline{RD}$↑)	<44>	t_{HRDID}		0		ns
Delay time from $\overline{RD}$↑ to address, $\overline{xBE}$	<70>	t_{DRDA}		$(1 + i)T - 8$		ns
Delay time from $\overline{RD}$↑ to ASTB↑	<71>	t_{DRRST}		$0.5T - 8$		ns
Delay time from $\overline{RD}$↑ to ASTB↓	<72>	t_{DRDST}		$(1.5 + i)T - 8$		ns
$\overline{RD}$ low-level width	<73>	t_{WRD}		$(1 + w + w_D)T - 4$		ns
ASTB high-level width	<74>	t_{WSTH}		$(1 + i + w_{AS})T - 4$		ns
$\overline{WAIT}$ setup time (to address)	<75>	t_{SAWT1}	Note		$(1.5 + w + w_D + w_{AS} + w_{AH})T - 25$	ns
$\overline{WAIT}$ setup time (to ASTB↓)	<76>	t_{SSTWT1}	Note		$(w + w_D + w_{AH})T - 25$	ns

Note $w, w_D \geq 1$

Remarks 1. $T = t_{CYK1}$

2. w: Number of waits set by $\overline{WAIT}$

3. w_D: Number of waits set by the DWC0 and DWC1 registers
When the programmable wait clock is inserted, the sampling timing differs.

4. Satisfy at least one of data input hold times t_{HRDID} and t_{HKID}.

5. n = 0 to 7
x = U, L

6. i: Number of idle states

7. w_{AS}: Number of address setup waits set by the ASC register

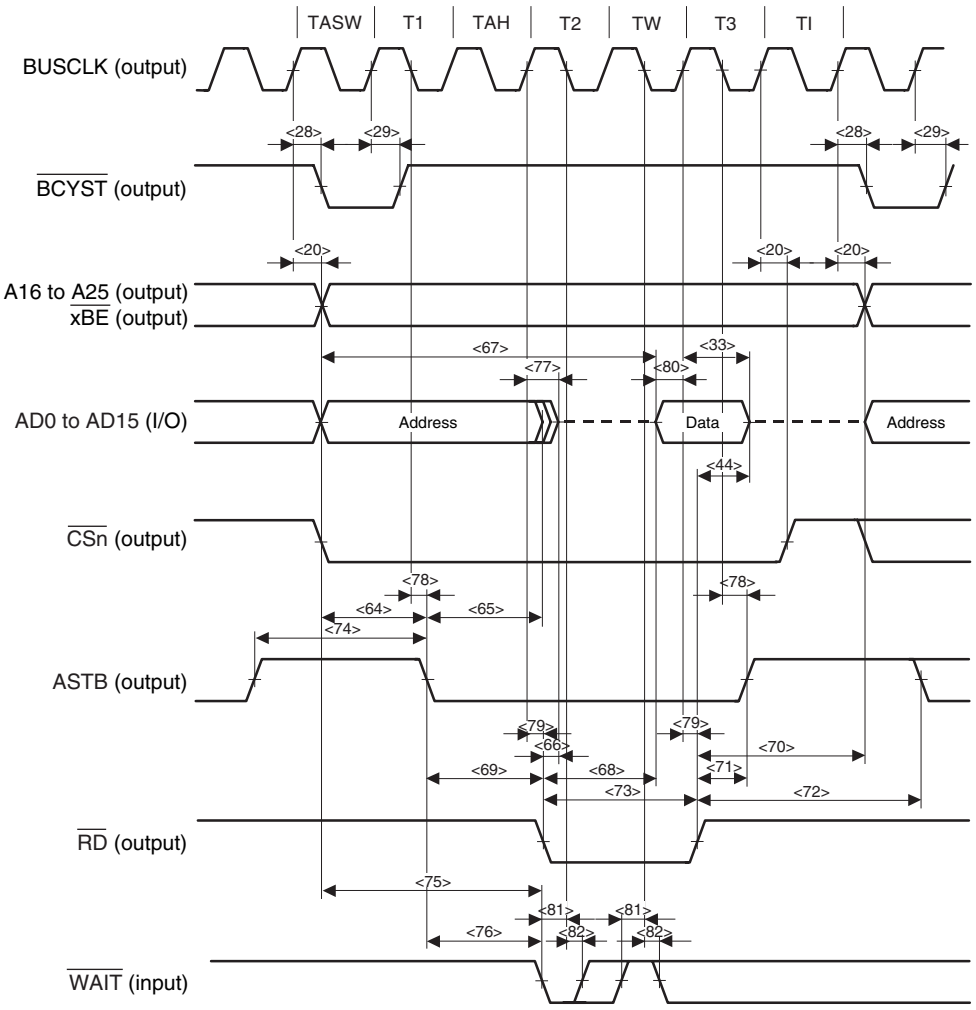
8. w_{AH}: Number of address hold waits set by the AHC register

(a) Read timing (BUSCLK synchronous) (SRAM, external ROM) (2/3)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{BCYST}}$↓ delay time (from BUSCLK↑)	<28>	t_{DKBSL}		0	13	ns
$\overline{\text{BCYST}}$↑ delay time (from BUSCLK↑)	<29>	t_{HKBSH}		0	13	ns
Address, $\overline{\text{xBE}}$, $\overline{\text{CSn}}$ output delay time (from BUSCLK↑)	<20>	t_{DKA}		0	15	ns
Address float delay time (from BUSCLK↑)	<77>	t_{FKA}		0	15	ns
ASTB delay time (from BUSCLK↓)	<78>	t_{DKST}		0	13	ns
$\overline{\text{RD}}$ delay time (from BUSCLK↑)	<79>	t_{DKRD}		0	13	ns
Data input setup time (to BUSCLK↑)	<80>	t_{SIDK}		12		ns
Data input hold time (from BUSCLK↑)	<33>	t_{HKID}		0		ns
$\overline{\text{WAIT}}$ setup time (to BUSCLK↓)	<81>	t_{SWTK}		12		ns
$\overline{\text{WAIT}}$ hold time (from BUSCLK↓)	<82>	t_{HKWT}		0		ns

Remark n = 0 to 7
x = U, L

(a) Read timing (BUSCLK asynchronous/BUSCLK synchronous) (SRAM, external ROM) (3/3)



Remarks 1. This is the timing when the number of waits set by the DWC0 and DWC1 registers is 0, the number of idle states set by the BCC register is 1, the number of waits set by the ASC register is 1, and the number of waits set by the AHC register is 1.

2. The broken lines indicate high impedance.

3. $\overline{IORD}$ and $\overline{IOWR}$ are not supported in the multiplexed bus mode.

4. n = 0 to 7
x = U, L

(b) Write timing (BUSCLK asynchronous) (SRAM, external ROM) (1/3)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address, $\overline{xBE}$, $\overline{CSn}$ setup time (to ASTB↓)	<64>	t_{SAST}		$(0.5 + w_{AS})T - 8$		ns
Address hold time (from ASTB↓)	<65>	t_{HSTA}		$(0.5 + w_{AH})T - 8$		ns
Delay time from ASTB↓ to $\overline{xWR}$, $\overline{WR}$↓	<83>	t_{DSTWR}		$(0.5 + w_{AH})T - 8$		ns
Delay time from $\overline{xWR}$, $\overline{WR}$↑ to ASTB↑	<84>	t_{DWRST}		$0.5T - 8$		ns
$\overline{xWR}$, $\overline{WR}$ low-level width	<56>	t_{WWRL}		$(1 + w + w_D)T - 4$		ns
ASTB high-level width	<85>	t_{WSTH}		$(1 + i + w_{AS})T - 8$		ns
Time from $\overline{xWR}$, $\overline{WR}$↓ to data output	<86>	t_{DWROD}			4	ns
Data output setup time (to $\overline{xWR}$, $\overline{WR}$↑)	<87>	t_{SODWR}		$(1 + w + w_D)T - 11$		ns
Data output hold time (from $\overline{xWR}$, $\overline{WR}$↑)	<88>	t_{HWROD}		$(1 + i)T - 8$		ns
$\overline{WAIT}$ setup time (to address)	<75>	t_{SAWT1}	**Note**		$(1.5 + w + w_D + w_{AS} + w_{AH})T - 25$	ns
$\overline{WAIT}$ setup time (to ASTB↓)	<76>	t_{SSTWT1}	**Note**		$(w + w_D + w_{AH})T - 25$	ns

Note $w, w_D \geq 1$

Remarks 1. $T = t_{CYK1}$

2. w: Number of waits set by $\overline{WAIT}$

3. w_D: Number of waits set by the DWC0 and DWC1 registers
When a programmable wait clock is inserted, the sampling timing differs.

4. Satisfy at least one of data input hold times t_{HRDID} or t_{HKID}.

5. n = 0 to 7
x = U, L

6. i: Number of idle states

7. w_{AS}: Number of address setup waits set by the ASC register

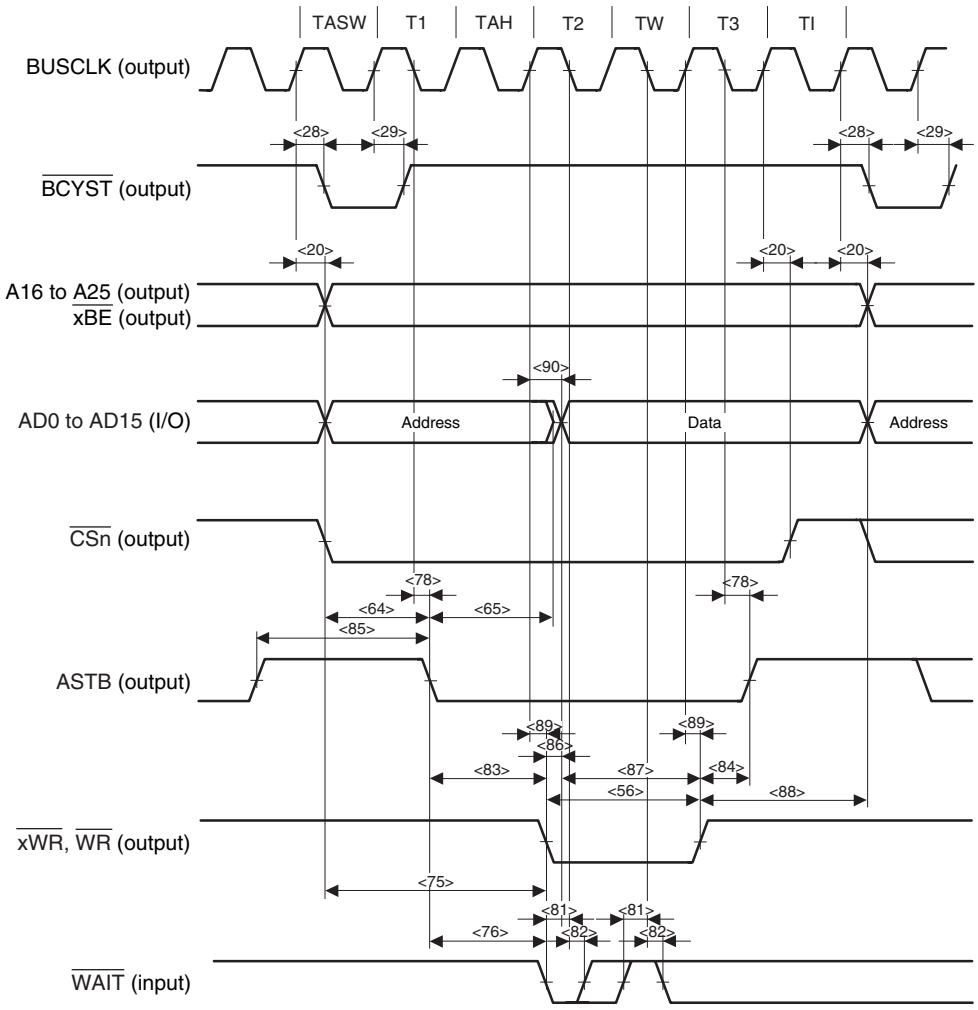
8. w_{AH}: Number of address hold waits set by the AHC register

(b) Write timing (BUSCLK synchronous) (SRAM, external ROM) (2/3)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{BCYST}}$↓ delay time (from BUSCLK↑)	<28>	t_{DKBSL}		0	13	ns
$\overline{\text{BCYST}}$↑ delay time (from BUSCLK↑)	<29>	t_{HKBSH}		0	13	ns
Address, $\overline{\text{xBE}}$, $\overline{\text{CSn}}$ output delay time (from BUSCLK↑)	<20>	t_{DKA}		0	15	ns
ASTB delay time (from BUSCLK↓)	<78>	t_{DKST}		0	13	ns
$\overline{\text{xWR}}$, $\overline{\text{WR}}$ delay time (from BUSCLK↑)	<89>	t_{DKWR}		0	13	ns
Data output delay time (from BUSCLK↑)	<90>	t_{DKOD}		0	13	ns
$\overline{\text{WAIT}}$ setup time (to BUSCLK↓)	<81>	t_{SWTK}		12		ns
$\overline{\text{WAIT}}$ hold time (from BUSCLK↓)	<82>	t_{HKWT}		0		ns

Remark n = 0 to 7
x = U, L

(b) Write timing (BUSCLK asynchronous/BUSCLK synchronous) (SRAM, external ROM) (3/3)



Remarks 1. This is the timing when the number of waits set by the DWC0 and DWC1 registers is 0, the number of idle states set by the BCC register is 1, the number of waits set by the ASC register is 1, and the number of waits set by the AHC register is 1.

2. IORD and IOWR are not supported in multiplexed bus mode.

3. n = 0 to 7

x = U, L

(6) Page ROM access timing (1/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{WAIT}$ setup time (to BUSCLK↑)	<30>	t_{SWK}		12		ns
$\overline{WAIT}$ hold time (from BUSCLK↑)	<31>	t_{HKW}		0		ns
Data input setup time (to BUSCLK↑)	<32>	t_{SKID}		13		ns
Data input hold time (from BUSCLK↑)	<33>	t_{HKID}		0		ns
Off-page data input setup time (to address, $\overline{CSn}$↓)	<38>	t_{SAID}			$(2 + w + w_D + w_{AS})T - 25$	ns
Off-page data input setup time (to $\overline{RD}$)	<39>	t_{SRDID}			$(1.5 + w + w_D)T - 25$	ns
Data input hold time (from $\overline{RD}$↑)	<44>	t_{HRDID}		0		ns
Delay time from $\overline{RD}$↑ to data output	<45>	t_{DRDOD}		$(1 + i)T - 16$		ns
$\overline{WAIT}$ high-level width	<49>	t_{WWH}		$T - 4$		ns
On-page data input setup time (to address)	<91>	t_{SOAID}			$(2 + w + w_{PR} + w_{AS})T - 25$	ns

Remarks 1. $T = t_{CYK1}$

2. w: Number of waits set by $\overline{WAIT}$

3. w_D: Number of waits set by the DWC0 and DWC1 registers

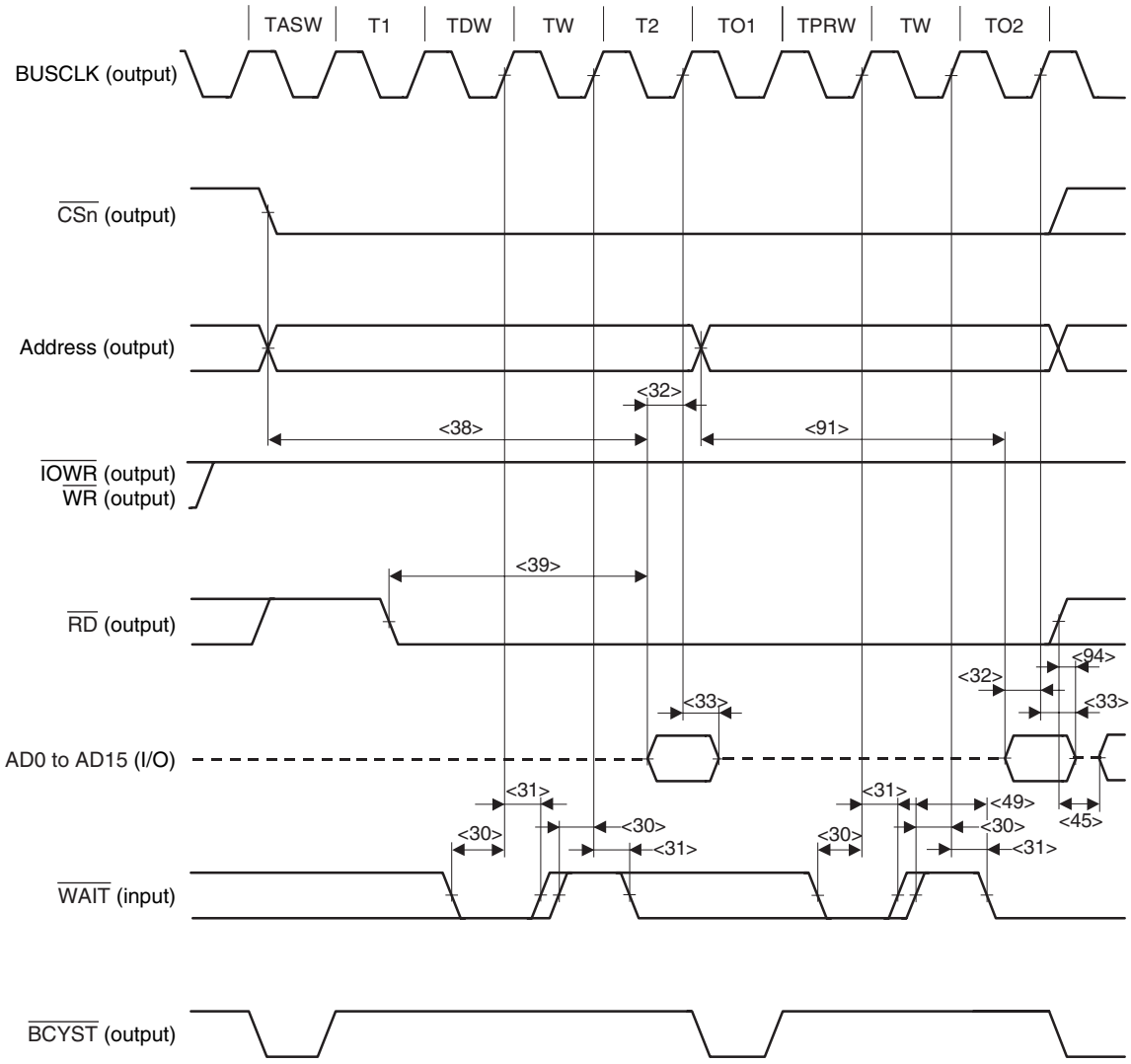
4. w_{PR}: Number of waits set by the PRC register

5. i: Number of idle states inserted when a read cycle is followed by the write cycle

6. w_{AS}: Number of address setup waits set by the ASC register

7. Satisfy at least one of data input hold times t_{HRDID} or t_{HKID}.

(6) Page ROM access timing (2/2)



Remarks 1. Timing when the following.
Number of waits set by the DWC0 and DWC1 registers (TDW): 1
Number of waits set by the PRC register (TPRW): 1
Number of waits set by the ASC register (TASW): 1

2. The broken lines indicate high impedance.

3. n = 0 to 7

(7) SDRAM access timing

(a) Read timing (SDRAM access) (1/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address delay time (from SDCLK↑)	<92>	t_{DSKA}		2	15	ns
$\overline{\text{BCYST}}$ delay time (from SDCLK↑)	<93>	t_{DSKBC}		0	15	ns
$\overline{\text{CSn}}$ delay time (from SDCLK↑)	<94>	t_{DSKCS}		1.5	15	ns
$\overline{\text{SDRAS}}$ delay time (from SDCLK↑)	<95>	t_{DSKRAS}		1	15	ns
$\overline{\text{SDCAS}}$ delay time (from SDCLK↑)	<96>	t_{DSKCAS}		1	15	ns
xDQM delay time (from SDCLK↑)	<97>	t_{DSKDQM}		2	15	ns
SDCKE delay time (from SDCLK↑)	<98>	t_{DSKCKE}		1.5	15	ns
Data input setup time (SDRAM read, to SDCLK↑)	<99>	t_{SDRMSK}		12		ns
Data input hold time (SDRAM read, from SDCLK↑)	<100>	t_{HSKDRM}		0		ns
Delay time from SDCLK↑ to data output	<101>	t_{DSDOD}		(1 + i)T – 5		ns

Caution If a SRAM (external I/O) cycle that uses the $\overline{\text{xWR}}$ signal is generated immediately after the cycle to read SDRAM, SRAM (external I/O) may be written by mistake. In this case, see 5.4.2 Chip select signal delay control register (CSDC).

However, SRAM (external I/O) is not written by mistake if synchronization is designed so that the $\overline{\text{xWR}}$ signal is sampled with BUSCLK.

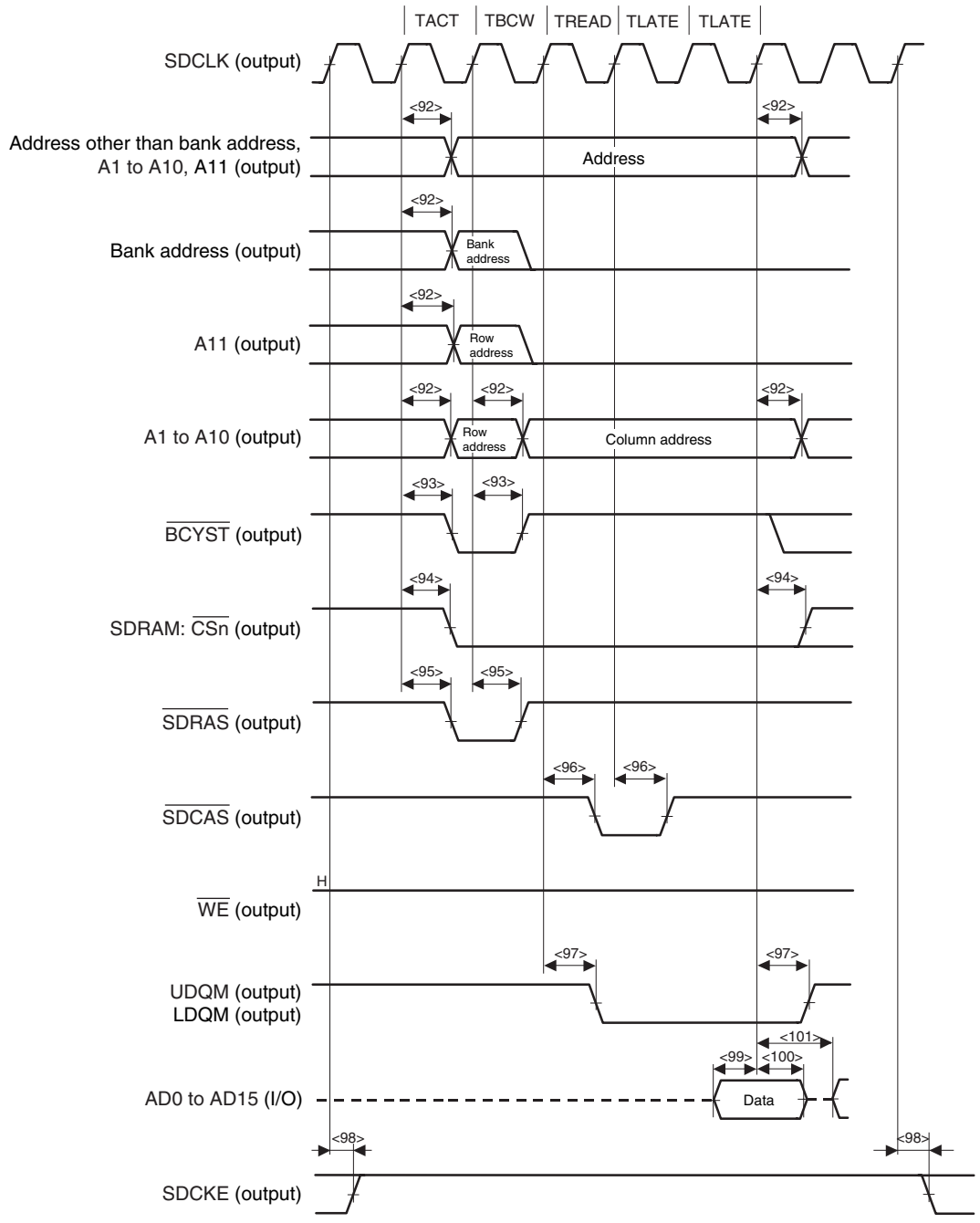
Remarks 1. T = t_{CYK2}

2. i: Number of idle states

3. n = 1, 3, 4, 6

x = U, L

(a) Read timing (SDRAM access) (2/2)



Caution A glitch may be generated when $\overline{\text{BCYST}}$ successively outputs low levels.

Remarks 1. Number of waits set by the BCWn1 and BCWn0 bits of the SCRn register (TBCW): 2

2. The broken lines indicate high impedance.

3. n = 1, 3, 4, 6

(b) Write timing (SDRAM access) (1/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address delay time (from SDCLK↑)	<92>	t_{DSKA}		2	15	ns
$\overline{BCYST}$ delay time (from SDCLK↑)	<93>	t_{DSKBC}		0	15	ns
$\overline{CSn}$ delay time (from SDCLK↑)	<94>	t_{DSKCS}		1.5	15	ns
$\overline{SDRAS}$ delay time (from SDCLK↑)	<95>	t_{DSKRAS}		1	15	ns
$\overline{SDCAS}$ delay time (from SDCLK↑)	<96>	t_{DSKCAS}		1	15	ns
xDQM delay time (from SDCLK↑)	<97>	t_{DSKDQM}		2	15	ns
SDCKE delay time (from SDCLK↑)	<98>	t_{DSKCKE}		1.5	15	ns
$\overline{WE}$ delay time (from SDCLK↑)	<102>	t_{DSKWE}		2	15	ns
Data output delay time (from SDCLK↑)	<103>	t_{DSKDT}		2	15	ns
Data float delay time (from SDCLK↑)	<104>	t_{HZSKDT}		2	15	ns

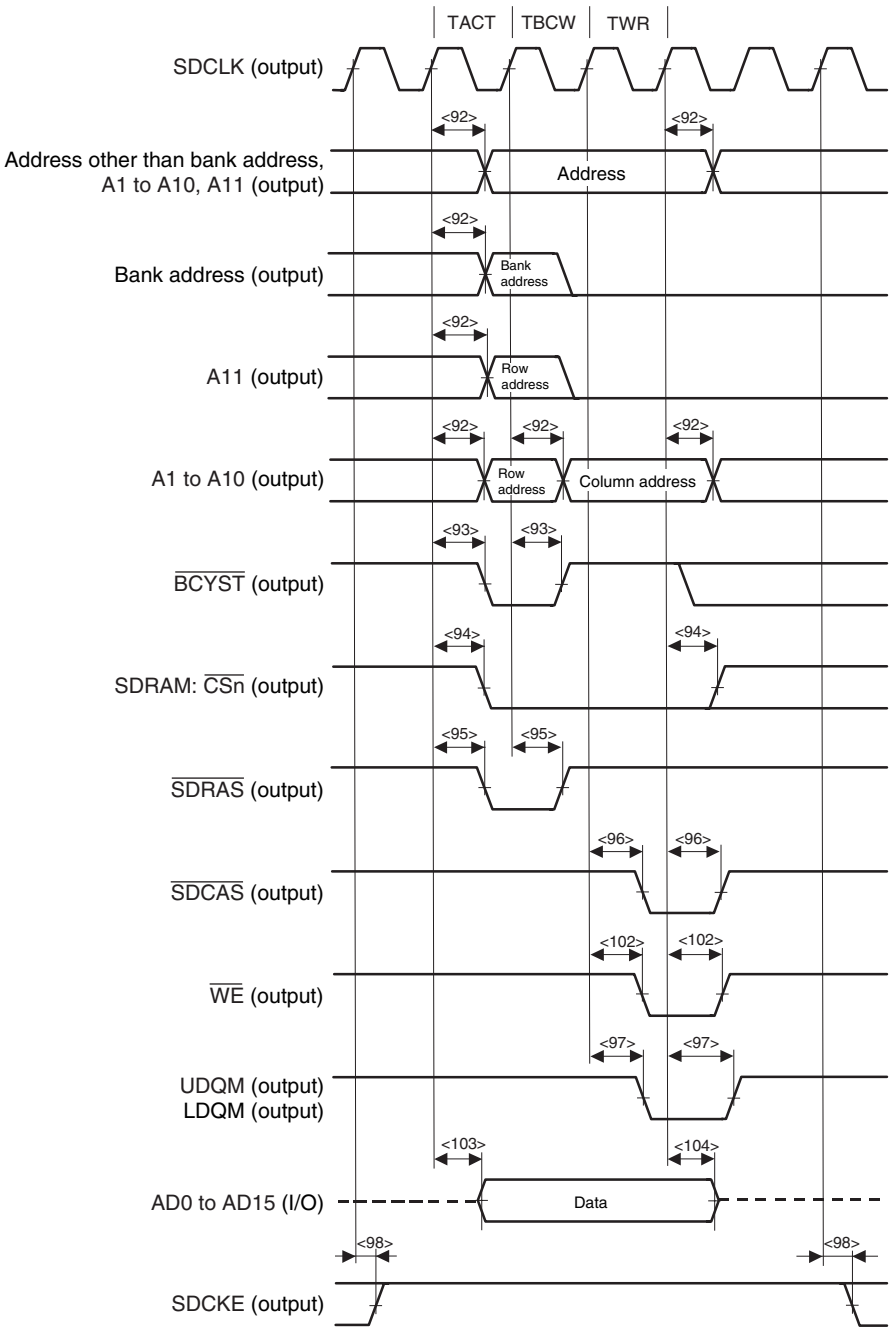
Caution If a SRAM (external I/O) cycle that uses the $\overline{xWR}$ or $\overline{WR}$ signal is generated immediately after the cycle to write SDRAM, SRAM (external I/O) may be written by mistake. In this case, see 5.4.2 Chip select signal delay control register (CSDC).

However, SRAM (external I/O) is not written by mistake if synchronization is designed so that the $\overline{xWR}$ or $\overline{WR}$ signal is sampled with BUSCLK.

Remark n = 1, 3, 4, 6

x = U, L

(b) Write timing (SDRAM access) (2/2)



Caution A glitch may be generated when $\overline{\text{BCYST}}$ successively outputs low levels.

Remarks 1. Number of waits (TBCW) set by the BCWn1 and BCWn0 bits of the SCRn register: 2

2. The broken lines indicate high impedance.

3. n = 1, 3, 4, 6

(8) DMAC timing (1/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{DMARQn}}$ setup time (to BUSCLK↑)	<105>	t_{SDRK}	2-cycle transfer	12		ns
$\overline{\text{DMARQn}}$ hold time 1	<106>	t_{HKDR1}	2-cycle transfer	Up to $\overline{\text{DMAAKn}}$↓		ns
$\overline{\text{DMARQn}}$ hold time 2 (from $\overline{\text{DMAAKn}}$↓)	<107>	t_{HKDR2}	NOMASK		$3T_{CPU} - 27$ [Note 1]	ns
					$6T_{CPU} - 27$ [Note 2]	ns
			MASK		$2T_{BUS} + 5T_{CPU} - 27$ [Note 1]	ns
					$2T_{BUS} + 8T_{CPU} - 27$ [Note 2]	ns
$\overline{\text{DMARQn}}$ hold time 3 after $\overline{\text{DMAAKn}}$↑ (from BUSCLK↑)	<108>	t_{HKDR3}	2-cycle transfer, MASK		$2T_{BUS} + 2T_{CPU} - 15$	ns
$\overline{\text{DMAAKn}}$ low-level width	<109>	t_{WDAL}	2-cycle transfer, NOMASK	$2T_{CPU} - 6$ [Note 3]		ns
				$5T_{CPU} - 6$ [Note 4]		ns
			2-cycle transfer, MASK	$2T_{CPU} - 6$ [Note 3]		ns
				$5T_{CPU} - 6$ [Note 4]		ns
$\overline{\text{TCn}}$ output delay time (from BUSCLK↑)	<110>	t_{DKTC}	2-cycle transfer	0	16	ns
$\overline{\text{TCn}}$ output hold time (from BUSCLK↑)	<111>	t_{HKTC}	2-cycle transfer	0	16	ns

Notes 1. This is the second DMA transfer request disabling timing for single transfer.
The access is as follows (when the number of waits for external memory access = 0).

Transfer source	Transfer destination
Internal RAM	External memory

2. This is the second DMA transfer request disabling timing for single transfer.
The access is other than that shown in the Table in **Note 1**.

3. The access is as follows.

Transfer source	Transfer destination
Internal RAM	external memory
external memory	Internal RAM

4. The access is other than that shown in the Table in **Note 3**.

Remarks 1. n = 0 to 3

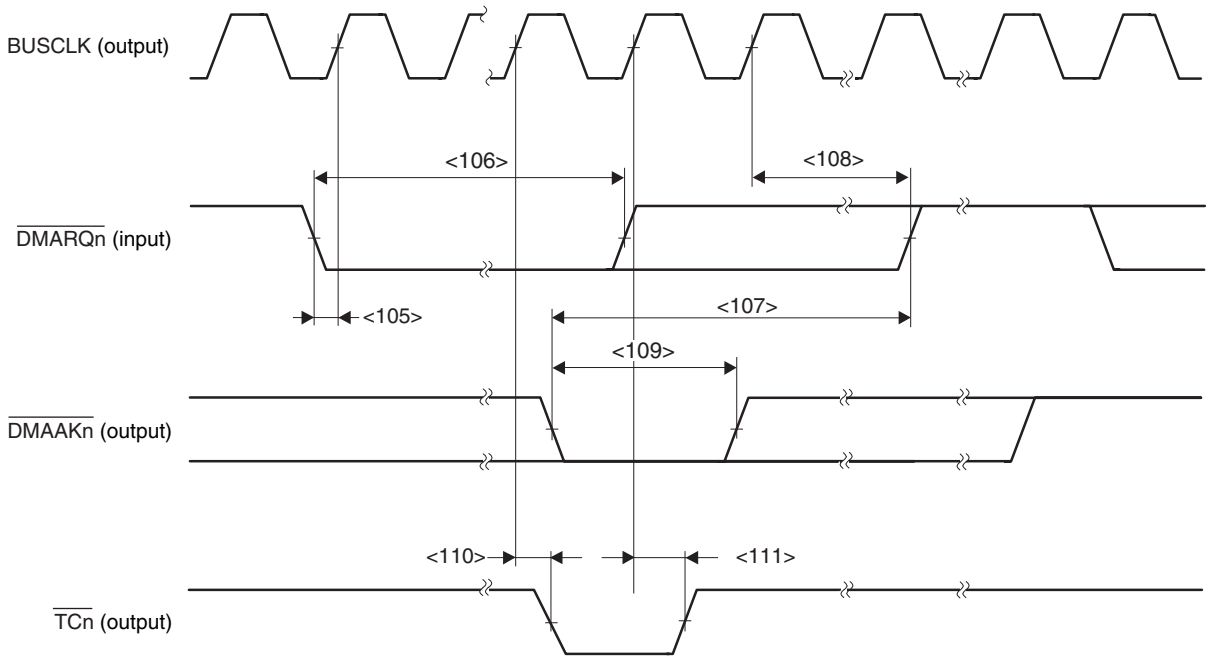
2. T_{BUS} = 1 × BUSCLK cycle

3. T_{CPU} = 1 × internal system clock cycle

4. NOMASK: Masking of $\overline{\text{DMARQn}}$ is not selected (DIFC.DRMKn bit = 0).

5. MASK: Masking of $\overline{\text{DMARQn}}$ is selected (DIFC.DRMKn bit = 1).

(8) DMAC timing (2/2)



Remarks 1. In 2-cycle transfer, the $\overline{TCn}$ signal is output in a write cycle. The time from the $\overline{DMAAKn}$ signal falling edge to $\overline{TCn}$ signal output cannot be defined.

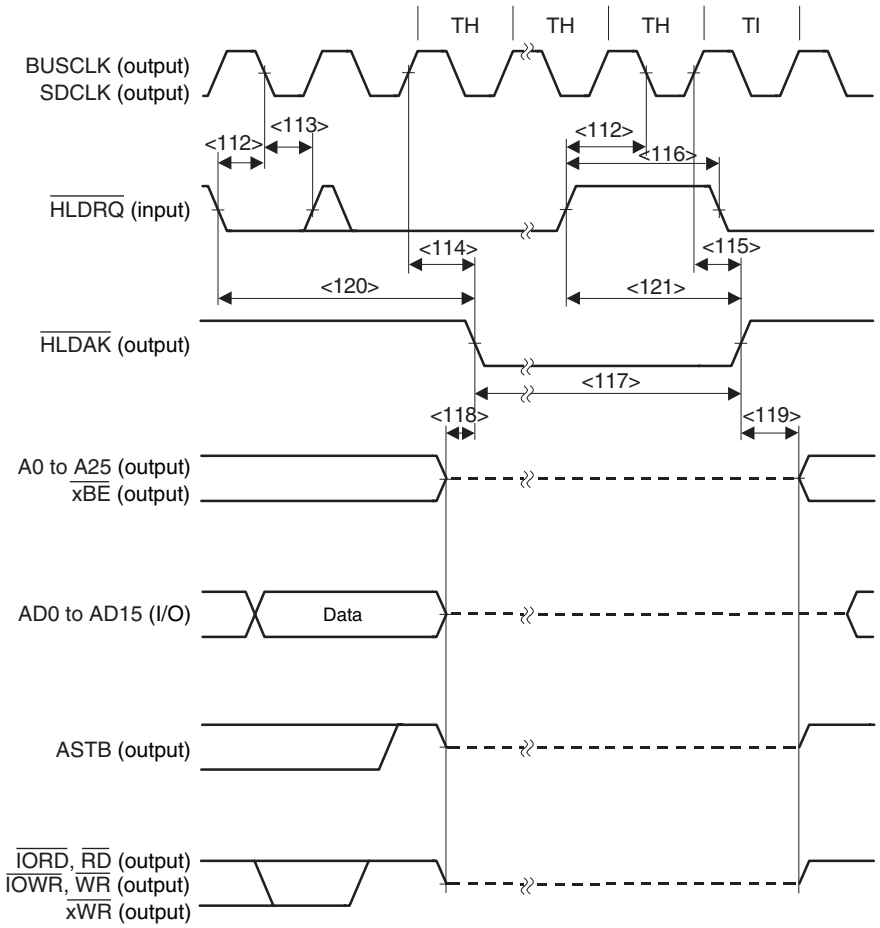
2. n = 0 to 3

(9) Bus hold timing (1/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{HLDRQ}}$ setup time (to BUSCLK↓)	<112>	t_{SHRK}		8		ns
$\overline{\text{HLDRQ}}$ setup time (to SDCLK↓)				8		ns
$\overline{\text{HLDRQ}}$ hold time (from BUSCLK↓)	<113>	t_{HKHR}		0		ns
$\overline{\text{HLDRQ}}$ hold time (from SDCLK↓)				0		ns
$\overline{\text{HLDAK}}$↓ delay time (from BUSCLK↑)	<114>	t_{DKHAL}		0	18	ns
$\overline{\text{HLDAK}}$↓ delay time (from SDCLK↑)				0	18	ns
$\overline{\text{HLDAK}}$↑ delay time (from BUSCLK↑)	<115>	t_{DKHAH}		0	13	ns
$\overline{\text{HLDAK}}$↑ delay time (from SDCLK↑)				0	13	ns
$\overline{\text{HLDRQ}}$ high-level width	<116>	t_{WHQH}		T + 4		ns
$\overline{\text{HLDAK}}$ low-level width	<117>	t_{WHAL}		T – 11		ns
Delay time from bus float to $\overline{\text{HLDAK}}$↓	<118>	t_{DHCF}		0		ns
Delay time from $\overline{\text{HLDAK}}$↑ to bus output	<119>	t_{DHAC}		0	15	ns
Delay time from $\overline{\text{HLDRQ}}$↓ to $\overline{\text{HLDAK}}$↓	<120>	t_{DHQHA1}		2T		ns
Delay time from $\overline{\text{HLDRQ}}$↑ to $\overline{\text{HLDAK}}$↑	<121>	t_{DHQHA2}		T	2T + 21	ns

Remark T = t_{CYKn} (n = 1, 2)

(9) Bus hold timing (2/2)



Remarks 1. The broken lines indicate high impedance.

2. n = 0 to 7

x = U, L

(10) Interrupt timing (1/2)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
NMI high-level width	<122>	t_{WNIH}	**Note 1**	500		ns
NMI low-level width	<123>	t_{WNIL}	**Note 1**	500		ns
$\overline{\text{INTPa}}$ pin high-level width	<124>	t_{WITH0}	**Note 1**	500		ns
$\overline{\text{INTPa}}$ pin low-level width	<125>	t_{WITL0}	**Note 1**	500		ns
INTPPb, INTPQk pins high-level width	<126>	t_{WITH1}	**Notes 1, 2**	$3T_0 + 500$		ns
INTPPb, INTPQk pins low-level width	<127>	t_{WITL1}	**Notes 1, 2**	$3T_0 + 500$		ns
INTP10, INTP11 pins high-level width	<128>	t_{WITH2}	**Note 2**	$3T_1 + 15$		ns
INTP10, INTP11 pins low-level width	<129>	t_{WITL2}	**Note 2**	$3T_1 + 15$		ns

Notes 1. Noise is eliminated for at least 30 ns.

2. When timer P, timer Q, or timer ENC1 is selected as an alternate function in the port function settings, noise is digitally eliminated.

Remarks 1. a = 000, 001, 004, 005, 010 to 013, 021, 022, 050, 051, 106, 107, 114, 115, 124 to 126, 130 to 134, 137

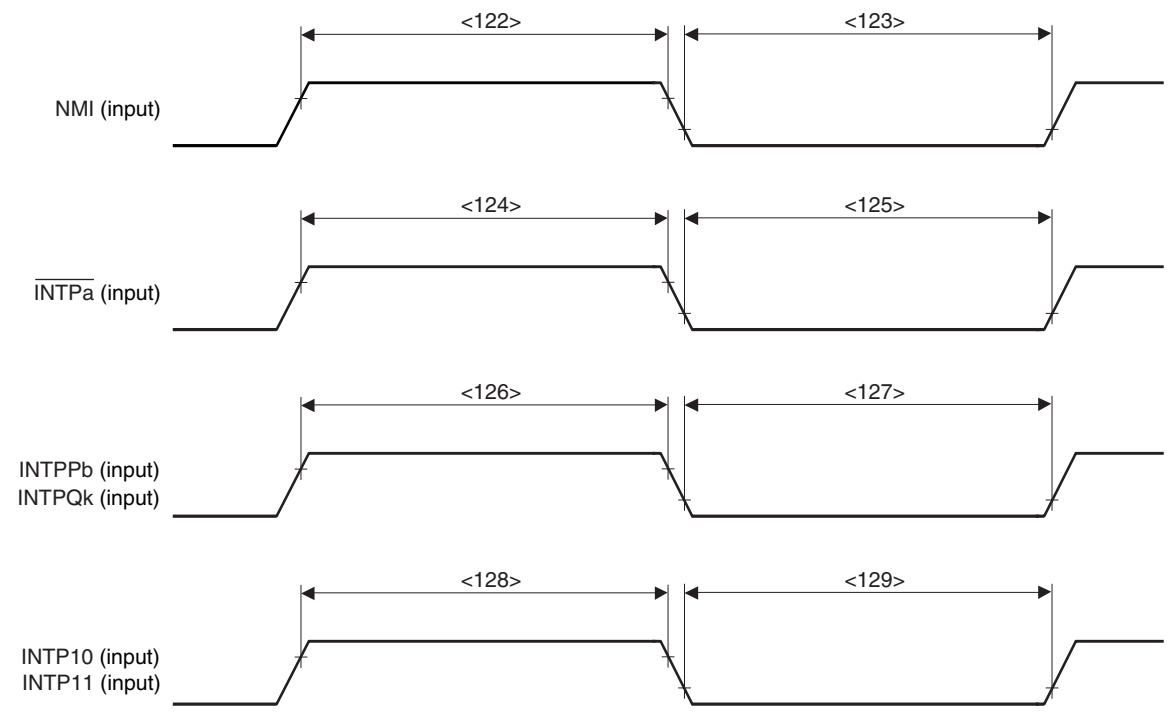
b = 00, 01, 10, 11, 20, 21

k = 0 to 3

2. T_0: TPnCTL0.TPnCKS2 to TPnCTL0.TPnCKS0 set values for INTPPb (n = 0 to 2)
TQ0CTL0.TQ0CKS2 to TQ0CTL0.TQ0CKS0 set values for INTPQk

3. T_1: PRM10.PRM102 to PRM10.PRM100 set values

(10) Interrupt timing (2/2)



Remark a = 000, 001, 004, 005, 010 to 013, 021, 022, 050, 051, 106, 107, 114, 115, 124 to 126, 130 to 134, 137
b = 00, 01, 10, 11, 20, 21
k = 0 to 3

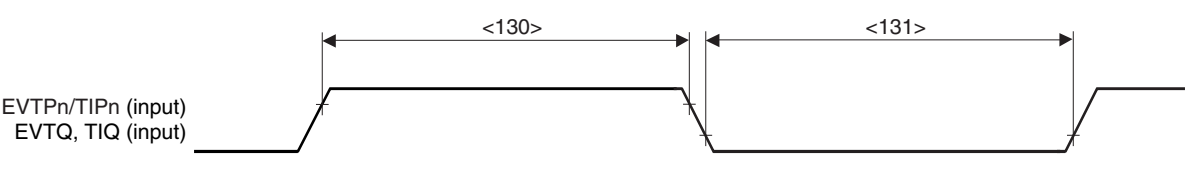
(11) Timer P (TMP), timer Q (TMQ) timing

	Parameter	Symbol		Conditions	MIN.	MAX.	Unit
<R>	EVTPn/TIPn, EVTQ, TIQ high-level width	<130>	t_{WTIH}	**Note**	$3T_0 + 500$		ns
<R>	EVTPn/TIPn, EVTQ, TIQ low-level width	<131>	t_{WTIL}	**Note**	$3T_0 + 500$		ns

<R> **Note** A noise of at least 40 ns is eliminated.

Remarks 1. n = 0 to 2

2. T_0: TPnCTL0.TPnCKS2 to TPnCTL0.TPnCKS0 set values for EVTPn/TIPn
TQ0CTL0.TQ0CKS2 to TQ0CTL0.TQ0CKS0 set values for EVTQ, TIQ



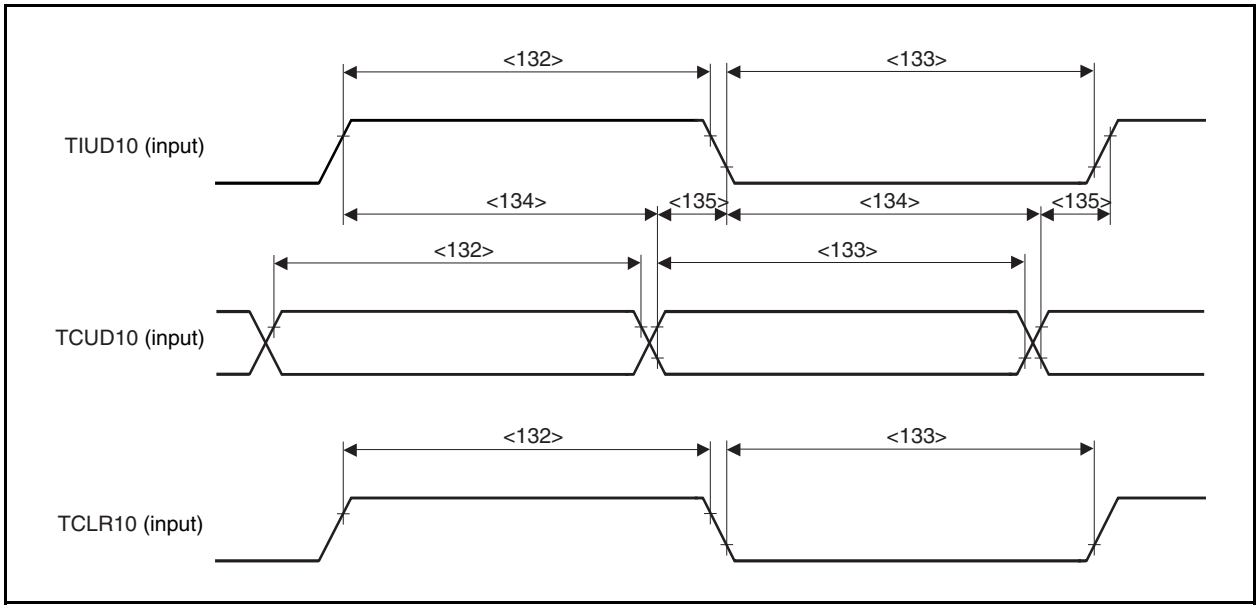
(12) Timer ENC1 (TMENC1) timing

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
TIUD10, TCUD10, TCLR10 pins high-level width	<132>	t_{WUDIH}		4T + 10		ns
TIUD10, TCUD10, TCLR10 pins low-level width	<133>	t_{WUDIL}		4T + 10		ns
Time difference from TIUD10 to TCUD10	<134>	t_{PHUD1}	**Note**	4T + 15		ns
Time difference from TCUD10 to TIUD10	<135>	t_{PHUD2}	**Note**	4T + 15		ns

Note The effect when noise is input is not included.

Remarks 1. T = 1/fxx (fxx: System clock)

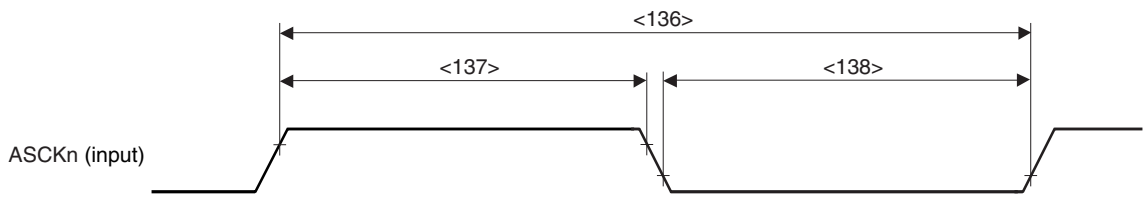
2. The digital noise eliminator is valid for TIUD10, TCUD10, and TCLR10 pins.



(13) UARTA timing

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
ASCKn cycle time	<136>	t_{KCY}		25		ns
ASCKn high-level width	<137>	t_{KH}		10		ns
ASCKn low-level width	<138>	t_{KL}		10		ns

Remark n = 0 to 2



Remark n = 0 to 2

(14) CSIB timing (1/3)

(a) Master mode

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
SCKn cycle	<139>	t_{CYSK1}	Output	100		ns
SCKn high-level width	<140>	t_{WSK1H}	Output	$0.5t_{CYSK1} - 20$		ns
SCKn low-level width	<141>	t_{WSK1L}	Output	$0.5t_{CYSK1} - 20$		ns
SIn setup time (to SCKn↑)	<142>	t_{SSISK}		30		ns
SIn setup time (to SCKn↓)				30		ns
SIn hold time (from SCKn↑)	<143>	t_{HSKSI}		0		ns
SIn hold time (from SCKn↓)				0		ns
SOn output delay time (from SCKn↓)	<144>	t_{DSKSO}			20	ns
SOn output delay time (from SCKn↑)					20	ns
SOn output hold time (from SCKn↑)	<145>	t_{HSKSO}		$0.5t_{CYSK1} - 5$		ns
SOn output hold time (from SCKn↓)				$0.5t_{CYSK1} - 5$		ns

Remark n = 0 to 2

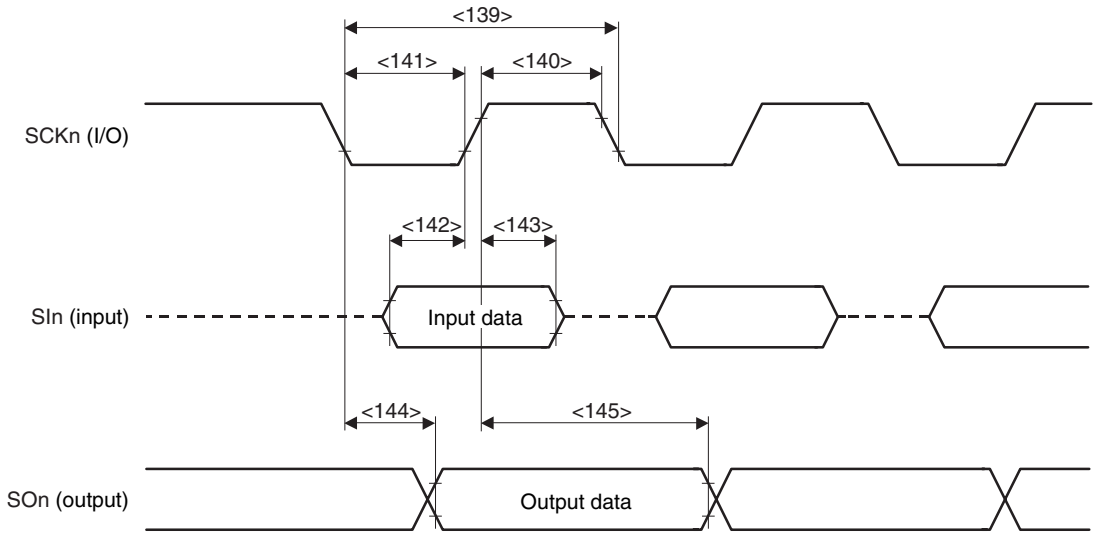
(b) Slave mode

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
SCKn cycle	<139>	t_{CYSK1}	Input	100		ns
SCKn high-level width	<140>	t_{WSK1H}	Input	30		ns
SCKn low-level width	<141>	t_{WSK1L}	Input	30		ns
SIn setup time (to SCKn↑)	<142>	t_{SSISK}		10		ns
SIn setup time (to SCKn↓)				10		ns
SIn hold time (from SCKn↑)	<143>	t_{HSKSI}		10		ns
SIn hold time (from SCKn↓)				10		ns
SOn output delay time (from SCKn↓)	<144>	t_{DSKSO}			30	ns
SOn output delay time (from SCKn↑)					30	ns
SOn output hold time (from SCKn↑)	<145>	t_{HSKSO}		t_{WSK1H}		ns
SOn output hold time (from SCKn↓)				t_{WSK1H}		ns

Remark n = 0 to 2

(14) CSIB timing (2/3)

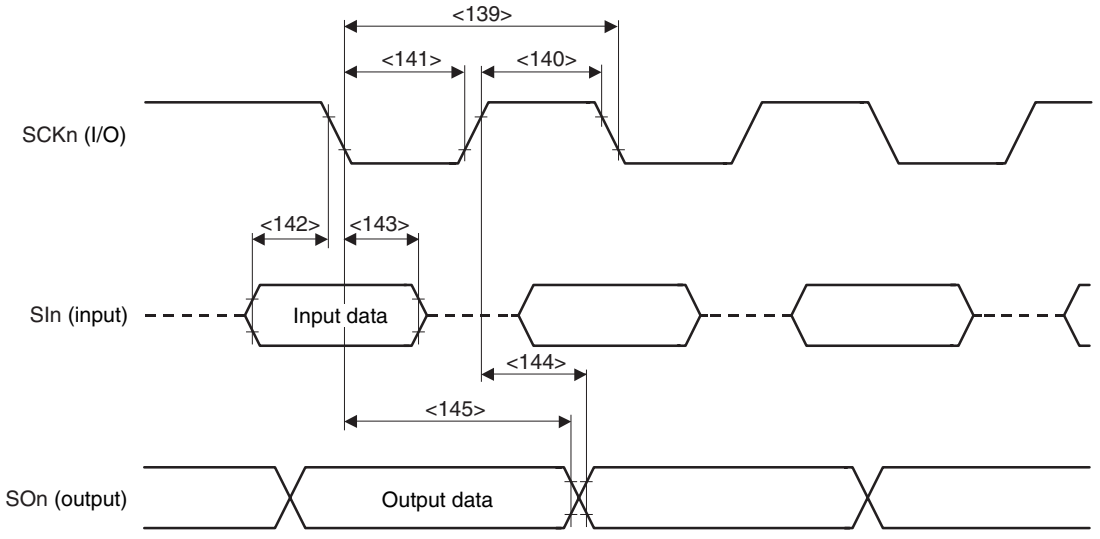
(c) Timing when CBnCTL1.CBnCKP, CBnCTL1.CBnDAP bits = 00



Remarks 1. The broken lines indicate high impedance.

2. n = 0 to 2

(d) Timing when CBnCTL1.CBnCKP, CBnCTL1.CBnDAP bits = 01

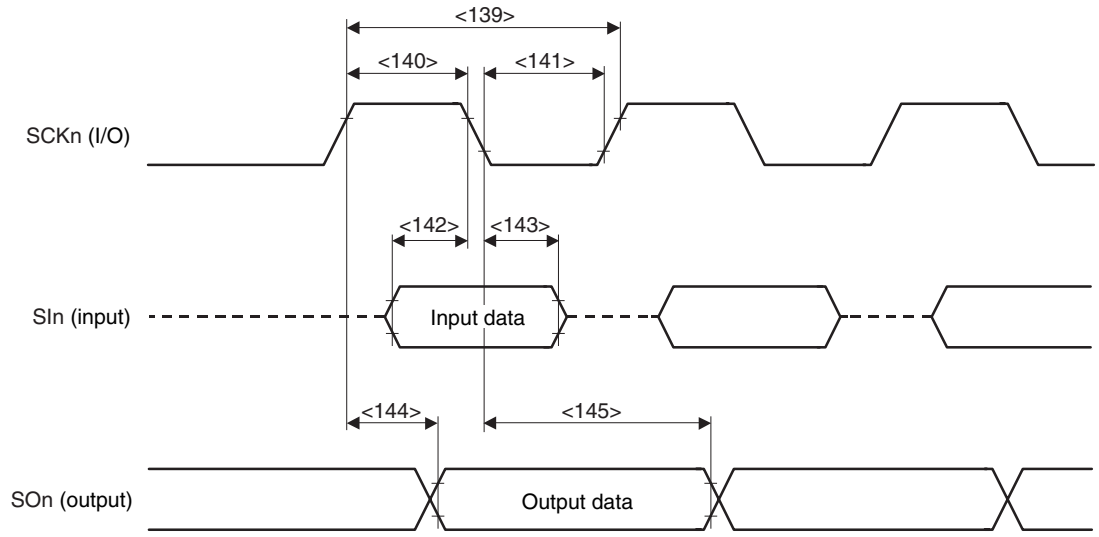


Remarks 1. The broken lines indicate high impedance.

2. n = 0 to 2

(14) CSIB timing (3/3)

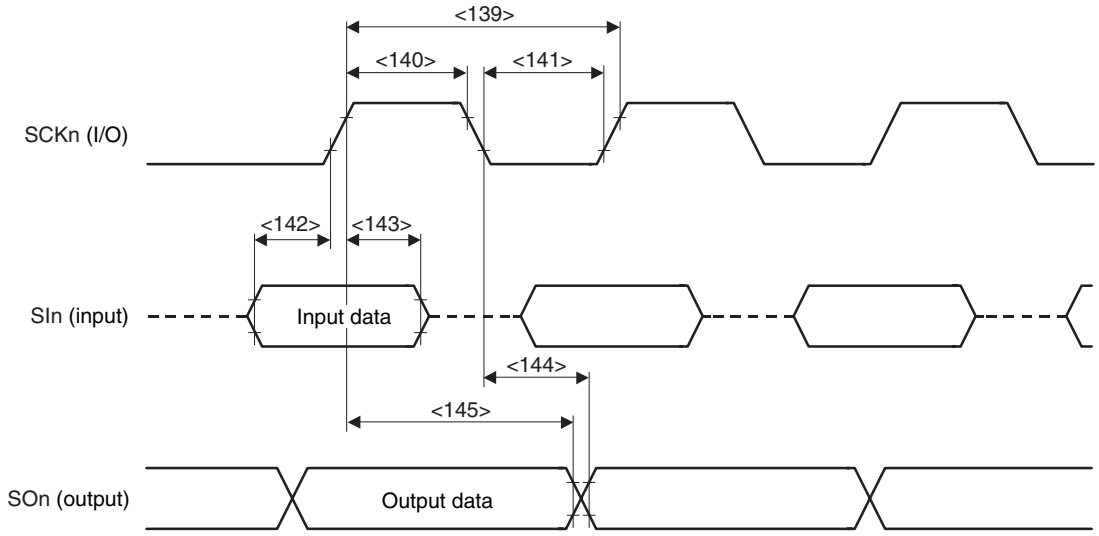
(e) Timing when CBnCTL1.CBnCKP, CBnCTL1.CBnDAP bits = 10



Remarks 1. The broken lines indicate high impedance.

2. n = 0 to 2

(f) Timing when CBnCTL1.CBnCKP, CBnCTL1.CBnDAP bits = 11



Remarks 1. The broken lines indicate high impedance.

2. n = 0 to 2

(15) I^2C bus mode (I^2C bus versions (Y products) only) (1/2)

Parameter		Symbol		Standard Mode		High-Speed Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCL clock frequency		–	f_{CLK}	0	100	0	400	kHz
Bus free time (between stop condition and start condition)		<146>	t_{BUF}	4.7	–	1.3	–	μs
Hold time[Note 1]		<147>	$t_{HD:STA}$	4.0	–	0.6	–	μs
SCL clock low-level width		<148>	t_{LOW}	4.7	–	1.3	–	μs
SCL clock high-level width		<149>	t_{HIGH}	4.0	–	0.6	–	μs
Start/restart condition setup time		<150>	$t_{SU:STA}$	4.7	–	0.6	–	μs
Data hold time	CBUS-compatible master	<151>	$t_{HD:DAT}$	5.0	–	–	–	μs
	I^2C mode			0[Note 2]	–	0[Note 2]	0.9[Note 3]	μs
Data setup time		<152>	$t_{SU:DAT}$	250	–	100[Note 4]	–	ns
SDA, SCL signal rise time		<153>	t_R	–	1000	20 + 0.1Cb[Note 5]	300	ns
SDA, SCL signal fall time		<154>	t_F	–	300	20 + 0.1Cb[Note 5]	300	ns
Stop condition setup time		<155>	$t_{SU:STO}$	4.0	–	0.6	–	μs
Pulse width of spike suppressed by input filter		<156>	t_{SP}	–	–	0	50	ns
Each bus line capacitive load		–	Cb	–	400	–	400	pF

Notes 1. The first clock pulse is generated after a hold time under the start condition.

2. The system must internally supply a hold time of at least 300 ns for the SDA signal (at $V_{IHmin.}$ of SCL signal) to fill the undefined area at the falling edge of SCL.

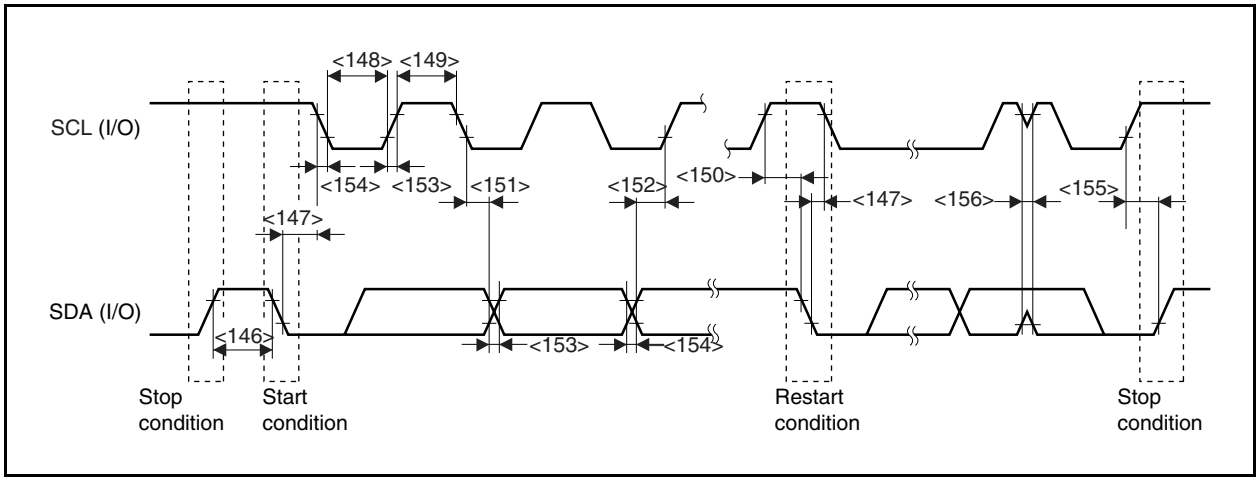
3. If the system does not extend the low hold time (t_{LOW}) of the SCL signal, the maximum data hold time ($t_{HD:DAT}$) must be satisfied.

4. The high-speed mode I^2C bus can be used in the standard mode I^2C bus system. In this case, make sure that the following conditions are satisfied.

- If system does not extend the hold time of the SCL signal in the low status
  $t_{SU:DAT} \geq 250$ ns
- If system extends the hold time of SCL signal in the low status
  Sends the next data bit to the SDA line before the SCL line is released ($t_{Rmax.} + t_{SU:DAT} = 1000 + 250 = 1250$ ns: standard mode I^2C bus specification).

5. Cb: Total capacitance of one bus line (unit: pF)

(15) I^2C bus mode (I^2C bus versions (Y products) only) (2/2)



A/D Converter Characteristics (T_A = –40 to +85°C, V_{DD} = 2.5 V ±0.2 V, EV_{DD} = CV_{DD} = AV_{DD0} = AV_{DD1} = 3.3 V ±0.3 V, V_{SS} = EV_{SS} = CV_{SS} = AV_{SS0} = AV_{SS1} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	–		10	10	10	bit
Overall error[Note 1]	–				±0.8	%FSR
Quantization error	–				±1/2	LSB
Conversion time[Note 2]	t_{CONV}		2.25		5.00	μs
Sampling time	t_{SAMP}		Conversion clock[Note 4] × 4/26			clock
Integral linearity error[Note 1]	–				±0.5	%FSR
Differential linearity error[Note 1]	–				±0.5	%FSR
Zero-scale error[Note 3]	–				±5	LSB
Full-scale error[Note 3]	–				±5	LSB
Analog input voltage	V_{WASN}		AV_{SS0}		AV_{DD0}	V
AV_{DD0} supply current	AI_{DD0}				12	mA

Notes 1. Excluding quantization error (±0.05%FSR)

2. Conversion time only for the analog block. The conversion time set by the ADM1.FR2 to ADM1.FR0 bits is the value including the time of transfer to the A/D controller block.

3. Excluding quantization error (±0.5 LSB)

4. Conversion clock is the number of clocks set by the ADM1 register.

Remark LSB: Least Significant Bit
FSR: Full Scale Range
%FSR is the ratio to the full-scale value.

D/A Converter Characteristics (T_A = –40 to +85°C, V_{DD} = 2.5 V ±0.2 V, EV_{DD} = CV_{DD} = AV_{DD0} = AV_{DD1} = 3.3 V ±0.3 V, V_{SS} = EV_{SS} = CV_{SS} = AV_{SS0} = AV_{SS1} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	–		8	8	8	bit
Overall error[Note 1]	–	Load condition: 2 MΩ, 20 pF AV_{DD1} = EV_{DD}			±1.18	%FSR
Settling time	–				3	μs
Output resistance	–			3.5		kΩ
AV_{DD1} supply current[Note 2]	AI_{DD1}				5	mA

Notes 1. Excluding quantization error (±0.2%FSR).

2. Current value when conversion value is 55H or ABH.

<R> ## 26.2 Power-On/Off Sequence

If the power to the internal circuitry (V_{DD}) is outside the guaranteed operating range (2.3 to 2.7 V) in the power-on/off sequence when a voltage is applied to the power supplies for the external circuitry (EV_{DD}, CV_{DD}, AV_{DD0}, and AV_{DD1}), the following phenomena may occur.

[Phenomena]

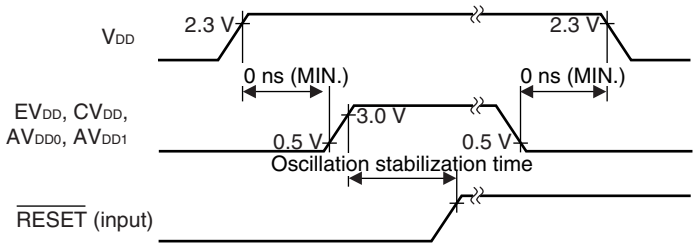
- A current of about 130 mA (TYP.) flows into the EV_{DD} pin.
- An undefined level is output from the following pins.
  TDO/$\overline{TC3}$/P27 pin
  ANO0/P80 pin
  ANO1/P81 pin

Therefore, the following power-on/off sequence is recommended. By turning on/off power in this sequence, the above phenomena can be avoided.

[Recommended sequence]

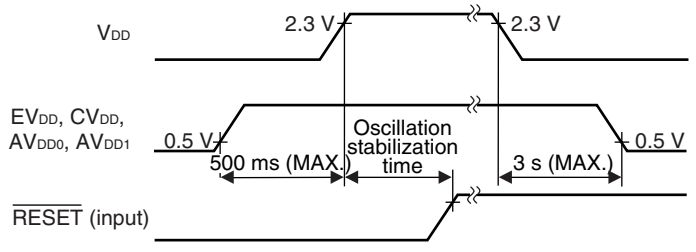
- To turn on
  Keep the voltage on the EV_{DD}, CV_{DD}, AV_{DD0}, and AV_{DD1} pins to within 0.5 V until the voltage on the V_{DD} pin reaches the guaranteed operation range (2.3 to 2.7 V).
- To turn off
  Keep the voltage on the V_{DD} pin in the guaranteed operation range (2.3 to 2.7 V) until the voltage on the EV_{DD}, CV_{DD}, AV_{DD0}, and AV_{DD1} pin falls below 0.5 V.

(a) Time of recommended sequence



Caution Turn on/off power within this time to avoid the above phenomena.

(b) Time if external power is turned on first or later



Caution If the time shown in (a) Time of the recommended sequence cannot be satisfied, the above phenomena may occur. Keep this time even if the above phenomena do not affect the application.

<R> 26.3 Flash Memory Programming Mode (μPD70F3134A, 70F3134AY Only)

Basic Characteristics (T_A = –40 to +85°C, V_{DD} = 2.5 V ±0.2 V, EV_{DD} = CV_{DD} = AV_{DD0} = AV_{DD1} = 3.3 V ±0.3 V, V_{SS} = EV_{SS} = CV_{SS} = AV_{SS0} = AV_{SS1} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operating frequency	f_{CPU1}	When self programming	10		80	MHz
	f_{CPU2}	When PG-FP4 is used	40		80 Note 1	MHz
Supply voltage	V_{DD}		2.3		2.7	V
Rewrite count	C_{WRT}	1 erasure + 1 rewrite after erasure = 1 rewrite, **Note 2**	100	100	100	Times
Programming temperature	T_{PRG}		–20		70	°C

Notes 1. 40 MHz when power is supplied from PG-FP4.

2. 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

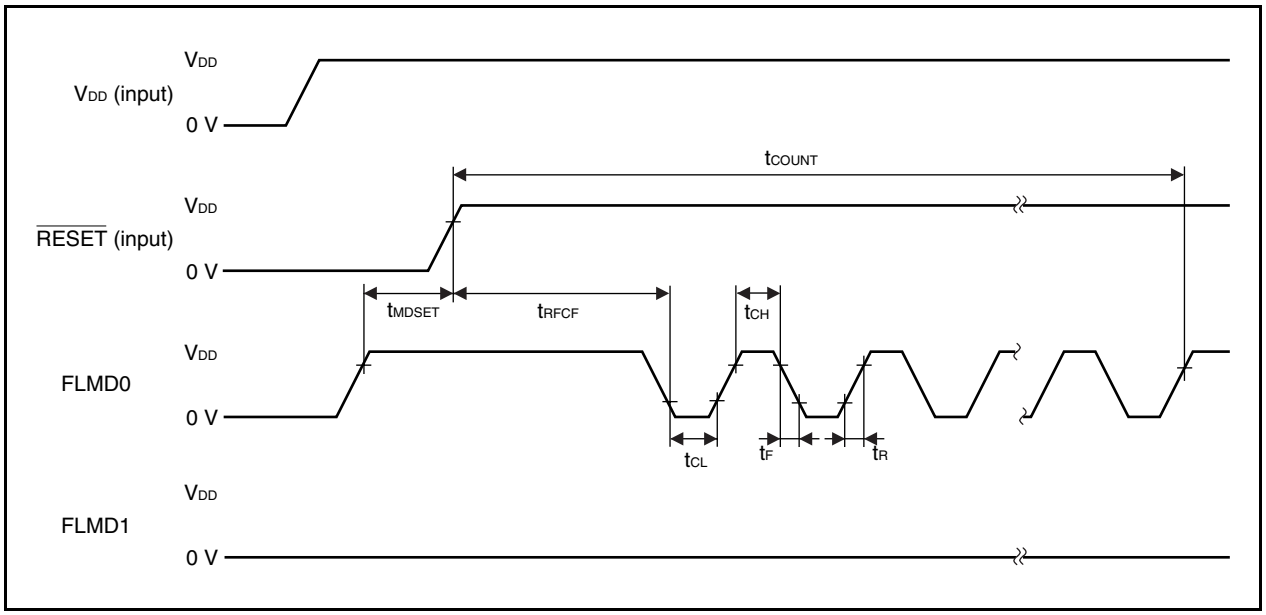
Shipped product → E → P→E→P→E→P: 3 rewrites

Remarks 1. When the PG-FP4 is used, a time parameter required for writing/erasing by downloading parameter files is automatically set. Do not change the settings otherwise specified.

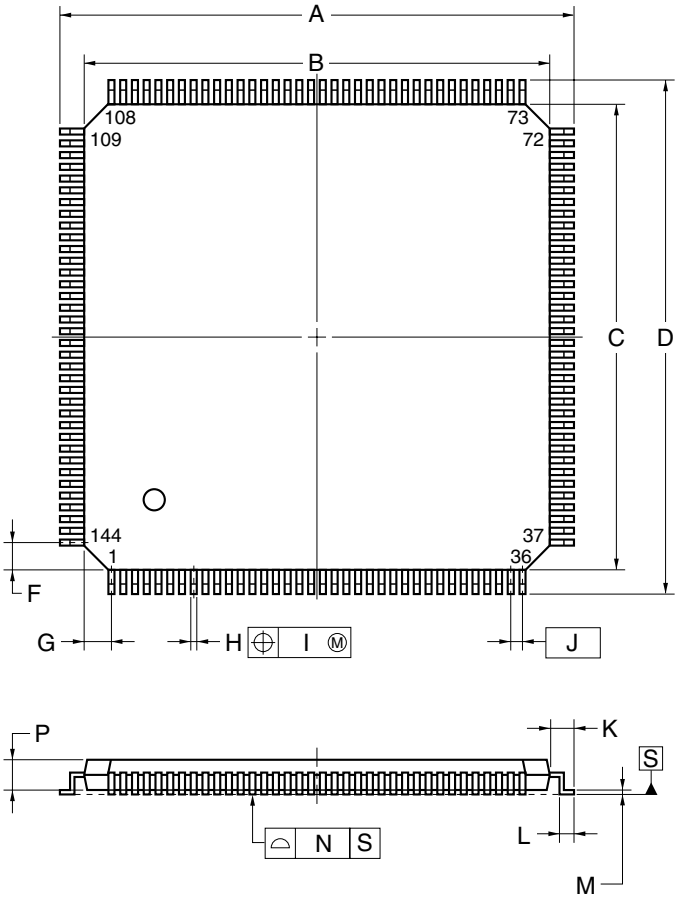
2. Block 0 = 00000H to 0FFFFH, block 1 = 10000H to 7FFFFH

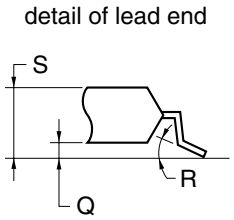
<R> **Serial Write Operation Characteristics (T_A = –20 to +70°C, V_{DD} = 2.5 V ±0.2 V, EV_{DD} = CV_{DD} = AV_{DD0} = AV_{DD1} = 3.3 V ±0.3 V, V_{SS} = EV_{SS} = CV_{SS} = AV_{SS0} = AV_{SS1} = 0 V)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
FLMD0, FLMD1 setup time	t_{MDSET}		2			μs
FLMD0 count start time from $\overline{\text{RESET}}$↑	t_{RFCF}		8.5			ms
Count execution time	t_{COUNT}				120	ms
FLMD0 counter high-level width	t_{CH}			50		μs
FLMD0 counter low-level width	t_{CL}			50		μs
FLMD0 counter rise time	t_R				50	ns
FLMD0 counter fall time	t_F				50	ns



144-PIN PLASTIC LQFP (FINE PITCH) (20x20)





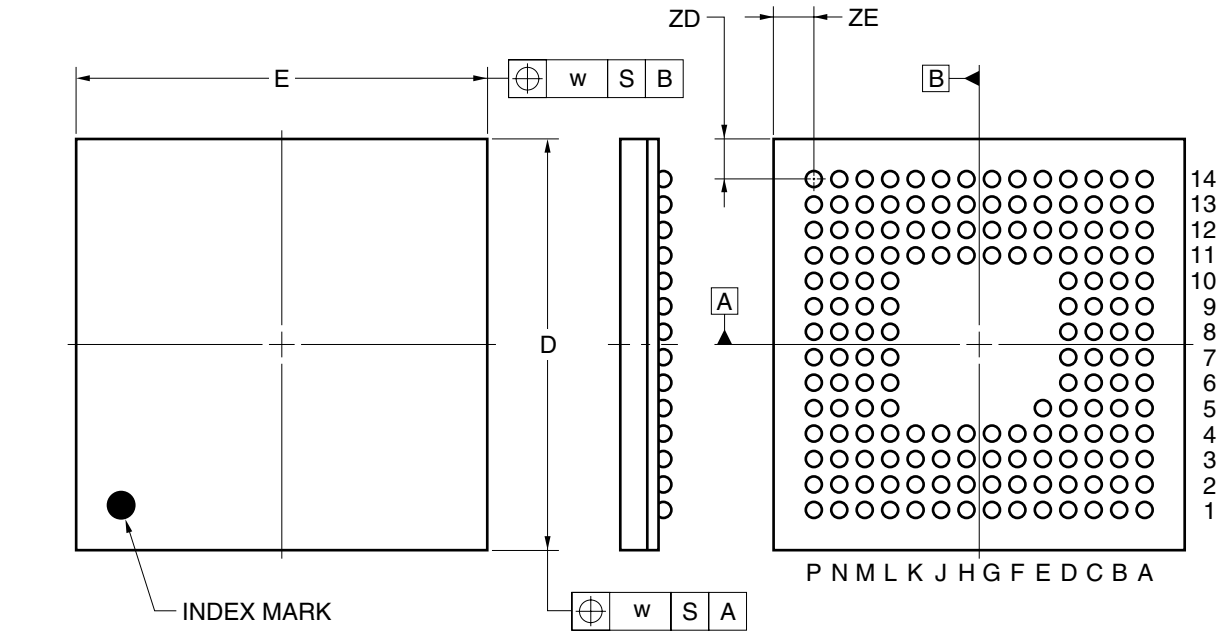
NOTE

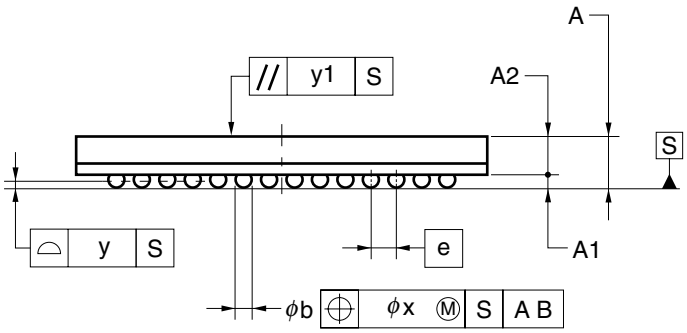
Each lead centerline is located within 0.08 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	22.0±0.2
B	20.0±0.2
C	20.0±0.2
D	22.0±0.2
F	1.25
G	1.25
H	0.22±0.05
I	0.08
J	0.5 (T.P.)
K	1.0±0.2
L	0.5±0.2
M	$0.17^{+0.03}_{-0.07}$
N	0.08
P	1.4
Q	0.10±0.05
R	$3^{\circ}{}^{+4^{\circ}}_{-3^{\circ}}$
S	1.5±0.1

S144GJ-50-UEN

161-PIN PLASTIC FBGA (13x13)





ITEM	MILLIMETERS
D	13.00±0.10
E	13.00±0.10
w	0.20
A	1.48±0.10
A1	0.35±0.06
A2	1.13
e	0.80
b	$0.50^{+0.05}_{-0.10}$
x	0.08
y	0.10
y1	0.20
ZD	1.30
ZE	1.30

P161F1-80-EN4-1

<R>

CHAPTER 28 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 28-1. Surface Mounting Type Soldering Conditions (1/2)

(1) μPD703131AGJ-xxx-UEN 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703131AYGJ-xxx-UEN 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703132AGJ-xxx-UEN 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703132AYGJ-xxx-UEN 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703133AGJ-xxx-UEN 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703133AYGJ-xxx-UEN 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703134AGJ-xxx-UEN 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703134AYGJ-xxx-UEN 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD70F3134AGJ-UEN 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD70F3134AYGJ-UEN 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703136AGJ-xxx-UEN 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703136AYGJ-xxx-UEN 144-pin plastic LQFP (fine pitch) (20 × 20)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: 3 times or less, Exposure limit: 3 days[Note] (after that, prebake at 125°C for 36 to 72 hours)	IR60-363-3
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Table 28-1. Surface Mounting Type Soldering Conditions (2/2)

(2) μPD703131AGJ-xxx-UEN-A 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703131AYGJ-xxx-UEN-A 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703132AGJ-xxx-UEN-A 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703132AYGJ-xxx-UEN-A 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703133AGJ-xxx-UEN-A 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703133AYGJ-xxx-UEN-A 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703134AGJ-xxx-UEN-A 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703134AYGJ-xxx-UEN-A 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD70F3134AGJ-UEN-A 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD70F3134AYGJ-UEN-A 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703136AGJ-xxx-UEN-A 144-pin plastic LQFP (fine pitch) (20 × 20)
μPD703136AYGJ-xxx-UEN-A 144-pin plastic LQFP (fine pitch) (20 × 20)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: 3 times or less, Exposure limit: 3 days[Note] (after that, prebake at 125°C for 36 to 72 hours)	IR60-363-3
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, please contact an NEC Electronics sales representative.

3. Soldering conditions of following products are undefined.

- μPD703131AF1-xxx-EN4: 161-pin plastic FBGA (13 × 13)
- μPD703131AYF1-xxx-EN4: 161-pin plastic FBGA (13 × 13)
- μPD703132AF1-xxx-EN4: 161-pin plastic FBGA (13 × 13)
- μPD703132AYF1-xxx-EN4: 161-pin plastic FBGA (13 × 13)
- μPD703133AF1-xxx-EN4: 161-pin plastic FBGA (13 × 13)
- μPD703133AYF1-xxx-EN4: 161-pin plastic FBGA (13 × 13)
- μPD703134AF1-xxx-EN4: 161-pin plastic FBGA (13 × 13)
- μPD703134AYF1-xxx-EN4: 161-pin plastic FBGA (13 × 13)
- μPD703136AF1-xxx-EN4: 161-pin plastic FBGA (13 × 13)
- μPD703136AYF1-xxx-EN4: 161-pin plastic FBGA (13 × 13)
- μPD70F3134AF1-EN4: 161-pin plastic FBGA (13 × 13)
- μPD70F3134AF1-EN4-A: 161-pin plastic FBGA (13 × 13)
- μPD70F3134AYF1-EN4: 161-pin plastic FBGA (13 × 13)
- μPD70F3134AYF1-EN4-A: 161-pin plastic FBGA (13 × 13)

APPENDIX A REGISTER INDEX

(1/9)

Symbol	Name	Unit	Page
ADCR0	A/D conversion result register 0	ADC	624
ADCR0H	A/D conversion result register 0H	ADC	624
ADCR1	A/D conversion result register 1	ADC	624
ADCR1H	A/D conversion result register 1H	ADC	624
ADCR2	A/D conversion result register 2	ADC	624
ADCR2H	A/D conversion result register 2H	ADC	624
ADCR3	A/D conversion result register 3	ADC	624
ADCR3H	A/D conversion result register 3H	ADC	624
ADCR4	A/D conversion result register 4	ADC	624
ADCR4H	A/D conversion result register 4H	ADC	624
ADCR5	A/D conversion result register 5	ADC	624
ADCR5H	A/D conversion result register 5H	ADC	624
ADCR6	A/D conversion result register 6	ADC	624
ADCR6H	A/D conversion result register 6H	ADC	624
ADCR7	A/D conversion result register 7	ADC	624
ADCR7H	A/D conversion result register 7H	ADC	624
ADIC	Interrupt control register 52	INTC	882
ADM0	A/D converter mode register 0	ADC	619
ADM1	A/D converter mode register 1	ADC	621
ADM2	A/D converter mode register 2	ADC	623
ADTS	A/D trigger select register	ADC	626
AHC	Address hold wait control register	BCU	225
ASC	Address setup wait control register	BCU	224
BCC	Bus cycle control register	BCU	230
BCP	Bus cycle period control register	BCU	226
BCT0	Bus cycle type configuration register 0	BCU	200
BCT1	Bus cycle type configuration register 1	BCU	200
BEC	Endian configuration register	BCU	205
BMC	Bus mode control register	BCU	221
CB0CTL0	CSIB0 control register 0	CSIB	692
CB0CTL1	CSIB0 control register 1	CSIB	695
CB0CTL2	CSIB0 control register 2	CSIB	696
CB0RX	CSIB0 receive data register (16 bits)	CSIB	691
CB0RXL	CSIB0 receive data register L (8 bits)	CSIB	691
CB0STR	CSIB0 status register	CSIB	698
CB0TX	CSIB0 transmit data register (16 bits)	CSIB	691
CB0TXL	CSIB0 transmit data register L (8 bits)	CSIB	691
CB1CTL0	CSIB1 control register 0	CSIB	692
CB1CTL1	CSIB1 control register 1	CSIB	695

(2/9)

Symbol	Name	Unit	Page
CB1CTL2	CSIB1 control register 2	CSIB	696
CB1RX	CSIB1 receive data register	CSIB	691
CB1RXL	CSIB1 receive data register L	CSIB	691
CB1STR	CSIB1 status register	CSIB	698
CB1TX	CSIB1 transmit data register	CSIB	691
CB1TXL	CSIB1 transmit data register L	CSIB	691
CB2CTL0	CSIB2 control register 0	CSIB	692
CB2CTL1	CSIB2 control register 1	CSIB	695
CB2CTL2	CSIB2 control register 2	CSIB	696
CB2RX	CSIB2 receive data register	CSIB	691
CB2RXL	CSIB2 receive data register L	CSIB	691
CB2STR	CSIB2 status register	CSIB	698
CB2TX	CSIB2 transmit data register	CSIB	691
CB2TXL	CSIB2 transmit data register L	CSIB	691
CC100	Capture/compare register 100	Timer	535
CC101	Capture/compare register 101	Timer	536
CCR10	Capture/compare control register 10	Timer	530
CKC	Clock control register	CG	309
CM100	Compare register 100	Timer	534
CM101	Compare register 101	Timer	534
CM10IC0	Interrupt control register 30	INTC	881
CM10IC1	Interrupt control register 31	INTC	881
CMD0	Compare register D0	Timer	518
CMD1	Compare register D1	Timer	518
CMD2	Compare register D2	Timer	518
CMD3	Compare register D3	Timer	518
CMICD0	Interrupt control register 26	INTC	881
CMICD1	Interrupt control register 27	INTC	881
CMICD2	Interrupt control register 28	INTC	881
CMICD3	Interrupt control register 29	INTC	881
CORAD0	Correction address register 0	CPU	920
CORAD0H	Correction address register 0H	CPU	920
CORAD0L	Correction address register 0L	CPU	920
CORAD1	Correction address register 1	CPU	920
CORAD1H	Correction address register 1H	CPU	920
CORAD1L	Correction address register 1L	CPU	920
CORAD2	Correction address register 2	CPU	920
CORAD2H	Correction address register 2H	CPU	920
CORAD2L	Correction address register 2L	CPU	920
CORAD3	Correction address register 3	CPU	920
CORAD3H	Correction address register 3H	CPU	920
CORAD3L	Correction address register 3L	CPU	920
CORCN	Correction control register	CPU	921
CSC0	Chip area select control register 0	BCU	197

(3/9)

Symbol	Name	Unit	Page
CSC1	Chip area select control register 1	BCU	197
CSDC	Chip select signal delay control register	BCU	202
DA0CS0	D/A conversion value setting register 0	DAC	654
DA0CS1	D/A conversion value setting register 1	DAC	654
DA0M	D/A converter mode register	DAC	653
DADC0	DMA addressing control register 0	DAC	818
DADC1	DMA addressing control register 1	DAC	818
DADC2	DMA addressing control register 2	DAC	818
DADC3	DMA addressing control register 3	DAC	818
DAKW	DMAAK width control register	DMAC	827
DBC0	DMA transfer count register 0	DMAC	817
DBC1	DMA transfer count register 1	DMAC	817
DBC2	DMA transfer count register 2	DMAC	817
DBC3	DMA transfer count register 3	DMAC	817
DCHC0	DMA channel control register 0	DMAC	820
DCHC1	DMA channel control register 1	DMAC	820
DCHC2	DMA channel control register 2	DMAC	820
DCHC3	DMA channel control register 3	DMAC	820
DDA0H	DMA destination address register 0H	DMAC	815
DDA0L	DMA destination address register 0L	DMAC	816
DDA1H	DMA destination address register 1H	DMAC	815
DDA1L	DMA destination address register 1L	DMAC	816
DDA2H	DMA destination address register 2H	DMAC	815
DDA2L	DMA destination address register 2L	DMAC	816
DDA3H	DMA destination address register 3H	DMAC	815
DDA3L	DMA destination address register 3L	DMAC	816
DIFC	DMA interface control register	DMAC	826
DMAIC0	Interrupt control register 36	INTC	881
DMAIC1	Interrupt control register 37	INTC	882
DMAIC2	Interrupt control register 38	INTC	882
DMAIC3	Interrupt control register 39	INTC	882
DSA0H	DMA source address register 0H	DMAC	813
DSA0L	DMA source address register 0L	DMAC	814
DSA1H	DMA source address register 1H	DMAC	813
DSA1L	DMA source address register 1L	DMAC	814
DSA2H	DMA source address register 2H	DMAC	813
DSA2L	DMA source address register 2L	DMAC	814
DSA3H	DMA source address register 3H	DMAC	813
DSA3L	DMA source address register 3L	DMAC	814
DTFR0	DMA trigger factor register 0	DMAC	823
DTFR1	DMA trigger factor register 1	DMAC	823
DTFR2	DMA trigger factor register 2	DMAC	823
DTFR3	DMA trigger factor register 3	DMAC	823
DTOC	DMA terminal count output control register	DMAC	822
DWC0	Data wait control register 0	BCU	222

(4/9)

Symbol	Name	Unit	Page
DWC1	Data wait control register 1	BCU	222
FIC	DMA flyby transfer idle control register	DMAC	231
FWC	DMA flyby transfer wait control register	DMAC	227
HZA0CTL0	High impedance output control register 00	Timer	560
HZA0CTL1	High impedance output control register 01	Timer	560
IIC	IIC shift register	I^2C	755
IICC	IIC control register	I^2C	740
IICCL	IIC clock selection register	I^2C	751
IICX	IIC function expansion register	I^2C	752
IICF	IIC flag register	I^2C	749
IICS	IIC status register	I^2C	746
IMR0	Interrupt mask register 0	INTC	883
IMR0H	Interrupt mask register 0H	INTC	883
IMR0L	Interrupt mask register 0L	INTC	883
IMR1	Interrupt mask register 1	INTC	883
IMR1H	Interrupt mask register 1H	INTC	883
IMR1L	Interrupt mask register 1L	INTC	883
IMR2	Interrupt mask register 2	INTC	883
IMR2H	Interrupt mask register 2H	INTC	883
IMR2L	Interrupt mask register 2L	INTC	883
IMR3	Interrupt mask register 3	INTC	883
IMR3H	Interrupt mask register 3H	INTC	883
IMR3L	Interrupt mask register 3L	INTC	883
INTF0	External interrupt falling edge specification register 0	INTC	887
INTF1	External interrupt falling edge specification register 1	INTC	888
INTF2	External interrupt falling edge specification register 2	INTC	889
INTF3	External interrupt falling edge specification register 3	INTC	890
INTF5	External interrupt falling edge specification register 5	INTC	891
INTR0	External interrupt rising edge specification register 0	INTC	887
INTR1	External interrupt rising edge specification register 1	INTC	888
INTR2	External interrupt rising edge specification register 2	INTC	889
INTR3	External interrupt rising edge specification register 3	INTC	890
INTR5	External interrupt rising edge specification register 5	INTC	891
ISPR	In-service priority register	INTC	884
LBS	Local bus sizing control register	BCU	204
NMIF	NMI falling edge specification register	INTC	886
NMIR	NMI rising edge specification register	INTC	886
OSTS	Oscillation stabilization time select register	CG	312
OVPIC0	Interrupt control register 32	INTC	881
OVPIC1	Interrupt control register 34	INTC	881
OVPIC2	Interrupt control register 35	INTC	881
OVQIC	Interrupt control register 33	INTC	881
P0	Port 0 register	Port	93
P00IC0	Interrupt control register 1	INTC	881
P00IC1	Interrupt control register 2	INTC	881

(5/9)

Symbol	Name	Unit	Page
P00IC4	Interrupt control register 3	INTC	881
P00IC5	Interrupt control register 4	INTC	881
P01IC0	Interrupt control register 7	INTC	881
P01IC1	Interrupt control register 8	INTC	881
P01IC2	Interrupt control register 9	INTC	881
P01IC3	Interrupt control register 10	INTC	881
P02IC1	Interrupt control register 13	INTC	881
P02IC2	Interrupt control register 14	INTC	881
P05IC0	Interrupt control register 24	INTC	881
P05IC1	Interrupt control register 25	INTC	881
P1	Port 1 register	Port	101
P10IC6	Interrupt control register 5	INTC	881
P10IC7	Interrupt control register 6	INTC	881
P11IC4	Interrupt control register 11	INTC	881
P11IC5	Interrupt control register 12	INTC	881
P12IC4	Interrupt control register 15	INTC	881
P12IC5	Interrupt control register 16	INTC	881
P12IC6	Interrupt control register 17	INTC	881
P13IC0	Interrupt control register 18	INTC	881
P13IC1	Interrupt control register 19	INTC	881
P13IC2	Interrupt control register 20	INTC	881
P13IC3	Interrupt control register 21	INTC	881
P13IC4	Interrupt control register 22	INTC	881
P13IC7	Interrupt control register 23	INTC	881
P2	Port 2 register	Port	108
P3	Port 3 register	Port	119
P4	Port 4 register	Port	129
P5	Port 5 register	Port	135
P7	Port 7 register	Port	140
P8	Port 8 register	Port	142
PAH	Port AH register	Port	148
PAHH	Port AHH register	Port	148
PAHL	Port AHL register	Port	148
PAL	Port AL register	Port	145
PALH	Port ALH register	Port	145
PALL	Port ALL register	Port	145
PBD	Port BD register	Port	174
PCC	Processor clock control register	CG	308
PCD	Port CD register	Port	170
PCM	Port CM register	Port	165
PCS	Port CS register	Port	155
PCT	Port CT register	Port	160
PDL	Port DL register	Port	152
PDLH	Port DLH register	Port	152

(6/9)

Symbol	Name	Unit	Page
PDLL	Port DLL register	Port	152
PFC0	Port 0 function control register	Port	95
PFC1	Port 1 function control register	Port	103
PFC2	Port 2 function control register	Port	110
PFC3	Port 3 function control register	Port	121
PFC4	Port 4 function control register	Port	131
PFC5	Port 5 function control register	Port	136
PFCCS	Port CS function control register	Port	157
PFCCT	Port CT function control register	Port	162
PFCE0	Port 0 function control expansion register	Port	95
PFCE1	Port 1 function control expansion register	Port	103
PFCE2	Port 2 function control expansion register	Port	110
PFCE3	Port 3 function control expansion register	Port	121
PFCE5	Port 5 function control expansion register	Port	136
PM0	Port 0 mode register	Port	93
PM1	Port 1 mode register	Port	101
PM2	Port 2 mode register	Port	108
PM3	Port 3 mode register	Port	119
PM4	Port 4 mode register	Port	129
PM5	Port 5 mode register	Port	135
PMAH	Port AH mode register	Port	149
PMAHH	Port AH mode register H	Port	149
PMAHL	Port AH mode register L	Port	149
PMAL	Port AL mode register	Port	146
PMALH	Port AL mode register H	Port	146
PMALL	Port AL mode register L	Port	146
PMBD	Port BD mode register	Port	174
PMC0	Port 0 mode control register	Port	94
PMC1	Port 1 mode control register	Port	102
PMC2	Port 2 mode control register	Port	109
PMC3	Port 3 mode control register	Port	120
PMC4	Port 4 mode control register	Port	130
PMC5	Port 5 mode control register	Port	136
PMC7	Port 7 mode control register	Port	140
PMCAH	Port AH mode control register	Port	149
PMCAHH	Port AH mode control register H	Port	149
PMCAHL	Port AH mode control register L	Port	149
PMCAL	Port AL mode control register	Port	146
PMCALH	Port AL mode control register H	Port	146
PMCALL	Port AL mode control register L	Port	146
PMCBD	Port BD mode control register	Port	175
PMCCD	Port CD mode control register	Port	171
PMCCM	Port CM mode control register	Port	166
PMCCS	Port CS mode control register	Port	156

(7/9)

Symbol	Name	Unit	Page
PMCCT	Port CT mode control register	Port	161
PMCD	Port CD mode register	Port	170
PMCDL	Port DL mode control register	Port	153
PMCDLH	Port DL mode control register H	Port	153
PMCDLL	Port DL mode control register L	Port	153
PMCM	Port CM mode register	Port	165
PMCS	Port CS mode register	Port	155
PMCT	Port CT mode register	Port	160
PMDL	Port DL mode register	Port	153
PMDLH	Port DL mode register H	Port	153
PMDLL	Port DL mode register L	Port	153
PRC	Page ROM configuration register	MEMC	265
PRCMD	Command register	CPU	82
PRM10	Prescaler mode register 10	Timer	532
PRSCM	Prescaler compare register	I^2C	753
PRSM	Prescaler mode register	I^2C	752
PSC	Power save control register	CPU	310, 905
PSMR	Power save mode register	CPU	311
RFS1	SDRAM refresh control register 1	MEMC	298
RFS3	SDRAM refresh control register 3	MEMC	298
RFS4	SDRAM refresh control register 4	MEMC	298
RFS6	SDRAM refresh control register 6	MEMC	298
SCR1	SDRAM configuration register 1	MEMC	275
SCR3	SDRAM configuration register 3	MEMC	275
SCR4	SDRAM configuration register 4	MEMC	275
SCR6	SDRAM configuration register 6	MEMC	275
SEIC0	Interrupt control register 40	INTC	882
SEIC1	Interrupt control register 43	INTC	882
SEIC2	Interrupt control register 46	INTC	882
SEIC3	Interrupt control register 49	INTC	882
SESA10	Valid edge select register 10	Timer	530
SRIC0	Interrupt control register 41	INTC	882
SRIC1	Interrupt control register 44	INTC	882
SRIC2	Interrupt control register 47	INTC	882
SRIC3	Interrupt control register 50	INTC	882
STATUS10	Status register 10	Timer	533
STIC0	Interrupt control register 42	INTC	882
STIC1	Interrupt control register 45	INTC	882
STIC2	Interrupt control register 48	INTC	882
STIC3	Interrupt control register 51	INTC	882
SVA	Slave address register	I^2C	755
SYS	System status register	CPU	83
TMC10	Timer control register 10	Timer	529
TMCD0	Timer mode control register D0	Timer	520

(8/9)

Symbol	Name	Unit	Page
TMCD1	Timer mode control register D1	Timer	520
TMCD2	Timer mode control register D2	Timer	520
TMCD3	Timer mode control register D3	Timer	520
TMD0	Timer D0	Timer	517
TMD1	Timer D1	Timer	517
TMD2	Timer D2	Timer	517
TMD3	Timer D3	Timer	517
TMENC10	Timer ENC10	Timer	526
TP0CCR0	TMP0 capture/compare register 0	Timer	326
TP0CCR1	TMP0 capture/compare register 1	Timer	328
TP0CNT	TMP0 counter read buffer register	Timer	330
TP0CTL0	TMP0 control register 0	Timer	319
TP0CTL1	TMP0 control register 1	Timer	320
TP0IOC0	TMP0 I/O control register 0	Timer	322
TP0IOC1	TMP0 I/O control register 1	Timer	323
TP0IOC2	TMP0 I/O control register 2	Timer	324
TP0OPT0	TMP0 option register 0	Timer	325
TP1CCR0	TMP1 capture/compare register 0	Timer	326
TP1CCR1	TMP1 capture/compare register 1	Timer	328
TP1CNT	TMP1 counter read buffer register	Timer	330
TP1CTL0	TMP1 control register 0	Timer	319
TP1CTL1	TMP1 control register 1	Timer	320
TP1IOC0	TMP1 I/O control register 0	Timer	322
TP1IOC1	TMP1 I/O control register 1	Timer	323
TP1IOC2	TMP1 I/O control register 2	Timer	324
TP1OPT0	TMP1 option register 0	Timer	325
TP2CCR0	TMP2 capture/compare register 0	Timer	326
TP2CCR1	TMP2 capture/compare register 1	Timer	328
TP2CNT	TMP2 counter read buffer register	Timer	330
TP2CTL0	TMP2 control register 0	Timer	319
TP2CTL1	TMP2 control register 1	Timer	320
TP2IOC0	TMP2 I/O control register 0	Timer	322
TP2IOC1	TMP2 I/O control register 1	Timer	323
TP2IOC2	TMP2 I/O control register 2	Timer	324
TP2OPT0	TMP2 option register 0	Timer	325
TQ0CCR0	TMQ0 capture/compare register 0	Timer	419
TQ0CCR1	TMQ0 capture/compare register 1	Timer	421
TQ0CCR2	TMQ0 capture/compare register 2	Timer	423
TQ0CCR3	TMQ0 capture/compare register 3	Timer	425
TQ0CNT	TMQ0 counter read buffer register	Timer	427
TQ0CTL0	TMQ0 control register 0	Timer	413
TQ0CTL1	TMQ0 control register 1	Timer	414
TQ0DTC	TMQ0 dead time compare register	Timer	553
TQ0IOC0	TMQ0 I/O control register 0	Timer	415

(9/9)

Symbol	Name	Unit	Page
TQ0IOC1	TMQ0 I/O control register 1	Timer	416
TQ0IOC2	TMQ0 I/O control register 2	Timer	417
TQ0IOC3	TMQ0 I/O control register 3	Timer	558
TQ0OPT0	TMQ0 option register 0	Timer	418, 554
TQ0OPT1	TMQ0 option register 1	Timer	555
TQ0OPT2	TMQ0 option register 2	Timer	556
TUM10	Timer unit mode register 10	Timer	528
UA0CTL0	UARTA0 control register 0	UARTA	664
UA0CTL1	UARTA0 control register 1	UARTA	680
UA0CTL2	UARTA0 control register 2	UARTA	681
UA0OPT0	UARTA0 option control register 0	UARTA	666
UA0RX	UARTA0 receive data register	UARTA	668
UA0STR	UARTA0 status register	UARTA	666
UA0TX	UARTA0 transmit data register	UARTA	668
UA1CTL0	UARTA1 control register 0	UARTA	664
UA1CTL1	UARTA1 control register 1	UARTA	680
UA1CTL2	UARTA1 control register 2	UARTA	681
UA1OPT0	UARTA1 option control register 0	UARTA	666
UA1RX	UARTA1 receive data register	UARTA	668
UA1STR	UARTA1 status register	UARTA	666
UA1TX	UARTA1 transmit data register	UARTA	668
UA2CTL0	UARTA2 control register 0	UARTA	664
UA2CTL1	UARTA2 control register 1	UARTA	680
UA2CTL2	UARTA2 control register 2	UARTA	681
UA2OPT0	UARTA2 option control register 0	UARTA	666
UA2RX	UARTA2 receive data register	UARTA	668
UA2STR	UARTA2 status register	UARTA	666
UA2TX	UARTA2 transmit data register	UARTA	668
UA3CTL0	UARTA3 control register 0	UARTA	664
UA3CTL1	UARTA3 control register 1	UARTA	680
UA3CTL2	UARTA3 control register 2	UARTA	681
UA3OPT0	UARTA3 option control register 0	UARTA	666
UA3RX	UARTA3 receive data register	UARTA	668
UA3STR	UARTA3 status register	UARTA	666
UA3TX	UARTA3 transmit data register	UARTA	668
VSWC	System wait control register	BCU	84
WAS	Write access synchronization control register	BCU	220
WDCS	Watchdog timer clock select register	WDT	610
WDRES	Watchdog timer reset status register	WDT	612, 916
WDTIC	Interrupt control register 0	INTC	881
WDTM	Watchdog timer mode register	WDT	611

APPENDIX B INSTRUCTION SET LIST

B.1 Conventions

(1) Register symbols used to describe operands

Register Symbol	Explanation
reg1	General-purpose registers: Used as source registers.
reg2	General-purpose registers: Used mainly as destination registers. Also used as source register in some instructions.
reg3	General-purpose registers: Used mainly to store the remainders of division results and the higher order 32 bits of multiplication results.
bit#3	3-bit data for specifying the bit number
immX	X bit immediate data
dispX	X bit displacement data
regID	System register number
vector	5-bit data that specifies the trap vector (00H to 1FH)
cccc	4-bit data that shows the conditions code
sp	Stack pointer (SP)
ep	Element pointer (r30)
listX	X item register list

(2) Register symbols used to describe opcodes

Register Symbol	Explanation
R	1-bit data of a code that specifies reg1 or regID
r	1-bit data of the code that specifies reg2
w	1-bit data of the code that specifies reg3
d	1-bit displacement data
I	1-bit immediate data (indicates the higher bits of immediate data)
i	1-bit immediate data
cccc	4-bit data that shows the condition codes
CCCC	4-bit data that shows the condition codes of Bcond instruction
bbb	3-bit data for specifying the bit number
L	1-bit data that specifies a program register in the register list
S	1-bit data that specifies a system register in the register list

(3) Register symbols used in operations

Register Symbol	Explanation
←	Input for
GR [ ]	General-purpose register
SR [ ]	System register
zero-extend (n)	Expand n with zeros until word length.
sign-extend (n)	Expand n with signs until word length.
load-memory (a, b)	Read size b data from address a.
store-memory (a, b, c)	Write data b into address a in size c.
load-memory-bit (a, b)	Read bit b of address a.
store-memory-bit (a, b, c)	Write c to bit b of address a.
saturated (n)	Execute saturated processing of n (n is a 2's complement). If, as a result of calculations, n ≥ 7FFFFFFFH, let it be 7FFFFFFFH. n ≤ 80000000H, let it be 80000000H.
result	Reflects the results in a flag.
Byte	Byte (8 bits)
Halfword	Half word (16 bits)
Word	Word (32 bits)
+	Addition
–	Subtraction
ll	Bit concatenation
×	Multiplication
÷	Division
%	Remainder from division results
AND	Logical product
OR	Logical sum
XOR	Exclusive OR
NOT	Logical negation
logically shift left by	Logical shift left
logically shift right by	Logical shift right
arithmetically shift right by	Arithmetic shift right

(4) Register symbols used in execution clock

Register Symbol	Explanation
i	If executing another instruction immediately after executing the first instruction (issue).
r	If repeating execution of the same instruction immediately after executing the first instruction (repeat).
l	If using the results of instruction execution in the instruction immediately after the execution (latency).

(5) Register symbols used in flag operations

Identifier	Explanation
(Blank)	No change
0	Clear to 0
X	Set or cleared in accordance with the results.
R	Previously saved values are restored.

(6) Condition codes

Condition Name (cond)	Condition Code (cccc)	Condition Formula	Explanation
V	0 0 0 0	OV = 1	Overflow
NV	1 0 0 0	OV = 0	No overflow
C/L	0 0 0 1	CY = 1	Carry Lower (Less than)
NC/NL	1 0 0 1	CY = 0	No carry Not lower (Greater than or equal)
Z/E	0 0 1 0	Z = 1	Zero Equal
NZ/NE	1 0 1 0	Z = 0	Not zero Not equal
NH	0 0 1 1	(CY or Z) = 1	Not higher (Less than or equal)
H	1 0 1 1	(CY or Z) = 0	Higher (Greater than)
N	0 1 0 0	S = 1	Negative
P	1 1 0 0	S = 0	Positive
T	0 1 0 1	–	Always (Unconditional)
SA	1 1 0 1	SAT = 1	Saturated
LT	0 1 1 0	(S xor OV) = 1	Less than signed
GE	1 1 1 0	(S xor OV) = 0	Greater than or equal signed
LE	0 1 1 1	((S xor OV) or Z) = 1	Less than or equal signed
GT	1 1 1 1	((S xor OV) or Z) = 0	Greater than signed

B.2 Instruction Set (in Alphabetical Order)

(1/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010010iiiii	GR[reg2]←GR[reg2]+sign-extend(imm5)		1	1	1	×	×	×	×	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1	×	×	×	×	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]		1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND zero-extend(imm16)		1	1	1		0	0	×	
Bcond	disp9	ddddd1011dddcccc **Note 1**	if conditions are satisfied then PC←PC+sign-extend(disp9)	When conditions are satisfied	3 Note 2	3 Note 2	3 Note 2					
				When conditions are not satisfied	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 wwwww01101000010	GR[reg3]←GR[reg2] (23 : 16) ll GR[reg2] (31 : 24) ll GR[reg2] (7 : 0) ll GR[reg2] (15 : 8)		1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 wwwww01101000000	GR[reg3]←GR[reg2] (7 : 0) ll GR[reg2] (15 : 8) ll GR [reg2] (23 : 16) ll GR[reg2] (31 : 24)		1	1	1	×	0	×	×	
CALLT	imm6	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))		5	5	5					
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		4	4	4	R	R	R	R	R
DBRET		0000011111100000 0000000101000110	PC←DBPC PSW←DBPSW		4	4	4	R	R	R	R	R

(2/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
DBTRAP		1111100001000000	DBPC←PC+2 (returned PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	4	4	4					
DI		0000011111100000 0000000101100000	PSW.ID←1	1	1	1					
DISPOSE	imm5,list12	0000011001iiiiiL LLLLLLLLLLL00000	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	imm5,list12,[reg1]	0000011001iiiiiL 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	rrrrr11110ddddddd ddddddddddddddd0 **Note 7**	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	3	3	3					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	4	4	4					
JR	disp22	0000011110ddddddd ddddddddddddddd0 **Note 7**	PC←PC+sign-extend(disp22)	3	3	3					
LD.B	disp16[reg1],reg2	rrrrr111000RRRRR ddddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 11					
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRRR ddddddddddddddd1 **Notes 8, 10**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 11					

(3/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
LD.H	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Halfword))		1	1	Note 11					
LDSR	reg2,regID	rrrrr111111RRRRR 0000000000100000 **Note 12**	SR[regID]←GR[reg2]	Other than regID = PSW	1	1	1					
				regID = PSW	1	1	1	×	×	×	×	×
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Halfword)		1	1	Note 11					
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←Load-memory(adr,Word)		1	1	Note 11					
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]		1	1	1					
	imm5,reg2	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	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1]		1	2 Note 14	2					
	imm9,reg2,reg3	r r r r r 1 1 1 1 1 i i i i i wwwww01001IIII00 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xsign-extend(imm9)		1	2 Note 14	2					
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2] Note 6 xGR[reg1] Note 6		1	1	2					
	imm5,reg2	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	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1]		1	2 Note 14	2					
	imm9,reg2,reg3	r r r r r 1 1 1 1 1 i i i i i wwwww01001IIII10 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xzero-extend(imm9)		1	2 Note 14	2					
NOP		0000000000000000	Pass at least one clock cycle doing nothing.		1	1	1					
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])		1	1	1		0	×	×	
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,Z flag)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,Z flag)		3 Note 3	3 Note 3	3 Note 3				×	

(4/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
OR	reg1,reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×	
ORI	imm16,reg1,reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×	
PREPARE	list12,imm5	0000011110iiiiiL LLLLLLLLLLL00001	Store-memory(sp–4,GR[reg in list12],Word) sp←sp–4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5)	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	list12,imm5, sp/imm[Note 15]	0000011110iiiiiL LLLLLLLLLLLff011 imm16/imm32 **Note 16**	Store-memory(sp–4,GR[reg in list12],Word) GR[reg in list 12]←Load-memory(sp,Word) sp←sp+4 repeat 2 step above until all regs in list12 is loaded PC←GR[reg1]	n+2 Note 4 Note 17	n+2 Note 4 Note 17	n+2 Note 4 Note 17					
RETI		0000011111100000 0000000101000000	if PSW.EP=1 then PC ←EIPC PSW ←EIPSW else if PSW.NP=1 then PC ←FEPC PSW ←FEPSW else PC ←EIPC PSW ←EIPSW	4	4	4	R	R	R	R	R
SAR	reg1,reg2	rrrrr111111RRRRR 0000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010101iiiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend (imm5)	1	1	1	×	0	×	×	
SASF	cccc,reg2	rrrrr1111110cccc 0000001000000000	if conditions are satisfied then GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000001H else GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000000H	1	1	1					
SATADD	reg1,reg2	rrrrr000110RRRRR	GR[reg2]←saturated(GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×
	imm5,reg2	rrrrr010001iiiii	GR[reg2]←saturated(GR[reg2]+sign-extend(imm5)	1	1	1	×	×	×	×	×
SATSUB	reg1,reg2	rrrrr000101RRRRR	GR[reg2]←saturated(GR[reg2]–GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16,reg1,reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiii	GR[reg2]←saturated(GR[reg1]–sign-extend(imm16)	1	1	1	×	×	×	×	×
SATSUBR	reg1,reg2	rrrrr000100RRRRR	GR[reg2]←saturated(GR[reg1]–GR[reg2])	1	1	1	×	×	×	×	×
SETF	cccc,reg2	rrrrr1111110cccc 0000000000000000	If conditions are satisfied then GR[reg2]←00000001H else GR[reg2]←00000000H	1	1	1					

(5/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SET1	bit#3,disp16[reg1]	00bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,1)	3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 Note 3	3 Note 3	3 Note 3				×	
SHL	reg1,reg2	rrrrr111111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010110iiiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×	
SHR	reg1,reg2	rrrrr111111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010100iiiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SLD.B	disp7[ep],reg2	rrrrr0110ddddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.BU	disp4[ep],reg2	rrrrr0000110dddd **Note 18**	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.H	disp8[ep],reg2	rrrrr1000ddddddd **Note 19**	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	Note 9					
SLD.HU	disp5[ep],reg2	rrrrr0000111dddd **Notes 18, 20**	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,Halfword))	1	1	Note 9					
SLD.W	disp8[ep],reg2	rrrrr1010ddddddd0 **Note 21**	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	Note 9					
SST.B	reg2,disp7[ep]	rrrrr0111ddddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1					
SST.H	reg2,disp8[ep]	rrrrr1001ddddddd **Note 19**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Halfword)	1	1	1					
SST.W	reg2,disp8[ep]	rrrrr1010ddddddd1 **Note 21**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1					
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1					
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Halfword)	1	1	1					
ST.W	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Word)	1	1	1					
STSR	regID,reg2	rrrrr111111RRRRR 0000000001000000	GR[reg2]←SR[regID]	1	1	1					

(6/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1,reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]–GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]–GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	00000000010RRRRR	adr←(PC+2) + (GR [reg1] logically shift left by 1) PC←(PC+2) + (sign-extend (Load-memory (adr,Halfword)) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←sign-extend (GR[reg1] (7 : 0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←sign-extend (GR[reg1] (15 : 0))	1	1	1					
TRAP	vector	0 0 0 0 0 1 1 1 1 1 1 i i i i i 0000000100000000	EIPC ←PC+4 (Return PC) EIPSW ←PSW ECR.EICC ←Exception code (40H to 4FH, 50H to 5FH) PSW.EP ←1 PSW.ID ←1 PC ←00000040H (when vector is 00H to 0FH (exception code: 40H to 4FH)) 00000050H (when vector is 10H to 1FH (exception code: 50H to 5FH))	4	4	4					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit (adr,bit#3))	3 Note 3	3 Note 3	3 Note 3				×	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←GR[reg1] Z flag←Not (Load-memory-bit (adr,reg2))	3 Note 3	3 Note 3	3 Note 3				×	
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16,reg1,reg2	rrrrr110101RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1] XOR zero-extend (imm16)	1	1	1		0	×	×	
ZXB	reg1	00000000100RRRRR	GR[reg1]←zero-extend (GR[reg1] (7 : 0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←zero-extend (GR[reg1] (15 : 0))	1	1	1					

Notes 1. ddddddddd: Higher 8 bits of disp9.

2. 4 if there is an instruction that rewrites the contents of the PSW immediately before.

3. If there is no wait state (3 + the number of read access wait states).

4. n is the total number of list12 load registers. (According to the number of wait states. Also, if there are no wait states, n is the total number of list12 registers. If n = 0, same operation as when n = 1)

5. RRRRR: other than 00000.

6. The lower halfword data only are valid.

7. ddddddddddddddddddddd: The higher 21 bits of disp22.

8. ddddddddddddddd: The higher 15 bits of disp16.

9. According to the number of wait states (1 if there are no wait states).

10. b: bit 0 of disp16.

11. According to the number of wait states (2 if there are no wait states).

Notes 12. In this instruction, for convenience of mnemonic description, the source register is made reg2, but the reg1 field is used in the opcode. Therefore, the meaning of register specification in the mnemonic description and in the opcode differs from other instructions.

rrrrr = regID specification

RRRRR = reg2 specification

13. iiiii: Lower 5 bits of imm9.

IIII: Higher 4 bits of imm9.

14. In the case of reg2 = reg3 (the lower 32 bits of the results are not written in the register) or reg3 = r0 (the higher 32 bits of the results are not written in the register), shortened by 1 clock.

15. sp/imm: specified by bits 19 and 20 of the sub-opcode.

16. ff = 00: Load sp in ep.

01: Load sign expanded 16-bit immediate data (bits 47 to 32) in ep.

10: Load 16-bit logically left shifted 16-bit immediate data (bits 47 to 32) in ep.

11: Load 32-bit immediate data (bits 63 to 32) in ep.

17. If imm = imm32, n + 3 clocks.

18. rrrrr: Other than 00000.

19. ddddddd: Higher 7 bits of disp8.

20. dddd: Higher 4 bits of disp5.

21. dddddd: Higher 6 bits of disp8.

22. Do not make a combination that satisfies all the following conditions when using the “MUL reg1, reg2, reg3” instruction and “MULU reg1, reg2, reg3” instruction. Operation is not guaranteed when an instruction that satisfies the following conditions is executed.

- Reg1 = reg3
- Reg1 ≠ reg2
- Reg1 ≠ r0
- Reg3 ≠ r0

APPENDIX C REVISION HISTORY

C.1 Major Revisions in This Edition

(1/5)

Page	Description
Throughout	• Addition of part numbers μPD703136A, 703136AY • Deletion of part numbers μPD703131, 703131Y, 703132, 703132Y, 703133, 703133Y, 703134, 703134Y, 70F3134, 70F3134Y • Addition of following lead-free products μPD703131AGJ-xxx-UEN-A, 703131AYGJ-xxx-UEN-A, 703132AGJ-xxx-UEN-A, 703132AYGJ-xxx-UEN-A, 703133AGJ-xxx-UEN-A, 703133AYGJ-xxx-UEN-A, 703134AGJ-xxx-UEN-A, 703134AYGJ-xxx-UEN-A, 70F3134AGJ-UEN-A, 70F3134AYGJ-UEN-A, 70F3134AF1-EN4-A, 70F3134AYF1-EN4-A • Under development → Mass production μPD703131AGJ-xxx-UEN, 703131AYGJ-xxx-UEN, 703132AGJ-xxx-UEN, 703132AYGJ-xxx-UEN, 703133AGJ-xxx-UEN, 703133AYGJ-xxx-UEN, 703134AGJ-xxx-UEN, 703134AYGJ-xxx-UEN
p. 21	Modification of description in **Table 1-1 V850E/MA3 Product List**
p. 22	Addition of description to Internal memory of Memory space in **1.2 Features**
p. 24	Modification of description in **1.4 Ordering Information**
p. 30	Addition of **Note** to **1.6.1 Internal block diagram**
p. 32	Addition of description to **1.6.2 (4) RAM**
p. 53	Modification of **Note** in **Table 3-2 System Register Numbers**
p. 58	Modification of description in **3.2.2 (6) Exception/debug trap status saving registers (DBPC, DBPSW)**
p. 65	Addition of **3.4.5 (2) (a) Internal RAM (8 KB)**
p. 69	Modification of description in **Figure 3-10 Recommended Memory Map**
p. 77	Deletion of **Note** in **3.4.8 On-chip peripheral I/O registers**
p. 80	Addition of **Note** to **3.4.8 On-chip peripheral I/O registers**
p. 81	Modification of **Caution** in **3.4.9 (1) Setting data to special registers**
p. 83	Addition of (iii) to **3.4.9 (3) (a) Set condition (PRERR flag = 1)**
p. 84	Modification of description in **3.4.10 System wait control register (VSWC)**
p. 88	Modification of description in **Table 4-2 Writing/Reading Pn Register**
p. 190	Addition of description to **4.6.2 Cautions on bit manipulation instruction for port n register (Pn)**
p. 196	Addition of **Note** to **5.3 Memory Block Function**
p. 202	Partial deletion of description in **5.4.2 (1) Chip select signal delay control register (CSDC)**
p. 300	Addition of **Caution** to **6.3.6 (1) (a) Notes on changing refresh interval**
p. 319	Modification of **Note** in **8.4 (1) TMPn control register 0 (TPnCTL0)**
p. 321	Addition of **Caution** in **8.4 (2) TMPn control register 1 (TPnCTL1)**
p. 322	Addition of **Caution** to **8.4 (3) TMPn I/O control register 0 (TPnIOC0)**
p. 323	Addition of **Caution** to **8.4 (4) TMPn I/O control register 1 (TPnIOC1)**
p. 325	Addition of description to **8.4 (6) TMPn option register 0 (TPnOPT0)**

(2/5)

Page	Description
p. 327	Addition of description to **8.4 (7) (a) Function as compare register**
p. 327	Addition of description to **8.4 (7) (b) Function as capture register**
p. 329	Addition of description to **8.4 (8) (a) Function as compare register**
p. 329	Addition of description to **8.4 (8) (b) Function as capture register**
p. 334	Addition of **Note** to **Figure 8-2 Flowchart of Basic Operation for Anytime Write**
p. 337	Addition of **Note** to **Figure 8-4 Flowchart of Basic Operation for Batch Write**
pp. 340, 341	Modification of description in **Figure 8-8 Register Setting for Interval Timer Mode Operation**
p. 342	Addition of description to **Figure 8-9 Software Processing Flow in Interval Timer Mode**
p. 349	Addition of **8.6.1 (3) Operation by external event count input (EVTPn)**
p. 350	Addition of description to **8.6.2 External event count mode (TPnMD2 to TPnMD0 bits = 001)**
p. 354	Addition of **Caution** to **8.6.2 (2) Operation timing in external event count mode**
p. 359	Modification of **Figure 8-21 Basic Timing in External Trigger Pulse Output Mode**
p. 359	Addition of description to **8.6.3 External trigger pulse output mode (TPnMD2 to TPnMD0 bits = 010)**
p. 360	Addition of **Note** to **Figure 8-22 Setting of Registers in External Trigger Pulse Output Mode**
p. 366	Modification of description in **8.6.3 (2) (b) 0%/100% output of PWM waveform**
p. 372	Addition of **Note** to **Figure 8-26 Setting of Registers in One-Shot Pulse Output Mode**
p. 374	Modification of **Figure 8-27 Software Processing Flow in One-Shot Pulse Output Mode**
p. 375	Modification of **8.6.4 (2) (a) Note on rewriting TPnCCRa register**
p. 379	Addition of **Note** to **Figure 8-30 Setting of Registers in PWM Output Mode**
p. 384	Modification of description in **8.6.5 (2) (b) 0%/100% output of PWM waveform**
p. 404	Modification of description in **8.6.7 Pulse width measurement mode (TPnMD2 to TPnMD0 bits = 110)**
p. 413	Modification of **Note** in **9.4 (1) TMQ0 control register 0 (TQ0CTL0)**
p. 415	Addition of **Caution** to **9.4 (3) TMQ0 I/O control register 0 (TQ0IOC0)**
p. 416	Addition of **Caution** to **9.4 (4) TMQ0 I/O control register 1 (TQ0IOC1)**
p. 418	Addition of description to **9.4 (6) TMQ0 option register 0 (TQ0OPT0)**
p. 420	Addition of description to **9.4 (7) (a) Function as compare register**
p. 420	Addition of description to **9.4 (7) (b) Function as capture register**
p. 422	Addition of description to **9.4 (8) (a) Function as compare register**
p. 422	Addition of description to **9.4 (8) (b) Function as capture register**
p. 424	Addition of description to **9.4 (9) (a) Function as compare register**
p. 424	Addition of description to **9.4 (9) (b) Function as capture register**
p. 426	Addition of description to **9.4 (10) (a) Function as compare register**
p. 426	Addition of description to **9.4 (10) (b) Function as capture register**
p. 431	Addition of **Note** to **Figure 9-2 Flowchart of Basic Operation for Anytime Write**
p. 434	Addition of **Note** to **Figure 9-4 Flowchart of Basic Operation for Batch Write**
pp. 437, 438	Modification of description in **Figure 9-8 Register Setting for Interval Timer Mode Operation**
p. 440	Addition of description to **Figure 9-9 Software Processing Flow in Interval Timer Mode**
p. 446	Addition of **9.6.1 (3) Operation by external event count input (EVTQ)**
p. 447	Addition of description to **9.6.2 External event count mode (TQ0MD2 to TQ0MD0 bits = 001)**
p. 451	Addition of **Caution** to **9.6.2 (2) Operation timing in external event count mode**
p. 457	Modification of **Figure 9-21 Basic Timing in External Trigger Pulse Output Mode**

(3/5)

Page	Description
p. 458	Addition of description to **9.6.3 External trigger pulse output mode (TQ0MD2 to TQ0MD0 bits = 010)**
p. 459	Addition of **Note** to **Figure 9-22 Setting of Registers in External Trigger Pulse Output Mode**
p. 465	Modification of description in **9.6.3 (2) (b) 0%/100% output of PWM waveform**
p. 472	Addition of **Note** to **Figure 9-26 Setting of Registers in One-Shot Pulse Output Mode**
pp. 474, 475	Modification of **Figure 9-27 Software Processing Flow in One-Shot Pulse Output Mode**
p. 481	Addition of **Note** to **Figure 9-30 Setting of Registers in PWM Output Mode**
p. 487	Modification of description in **9.6.5 (2) (b) 0%/100% output of PWM waveform**
p. 510	Modification of description in **9.6.7 Pulse width measurement mode (TQ0MD2 to TQ0MD0 bits = 110)**
p. 551	Modification of **Figure 12-1 Block Diagram of Motor Control**
p. 552	Modification of **Figure 12-2 TMQ0 Option**
p. 556	Addition of **Caution** to **12.3 (3) TMQ0 option register 2 (TQ0OPT2)**
p. 565	Modification of **Figure 12-5 Outline of 6-Phase PWM Output Mode**
p. 566	Modification of description in **Figure 12-6 Timing Chart of 6-Phase PWM Output Mode**
p. 568	Modification of **Figure 12-7 Interrupt and Up/Down Flag**
p. 576	Addition of description to **12.4.3 Interrupt culling function**
p. 601	Addition of description to **12.4.5 (1) (b) Setting of TMQ0 register**
p. 647	Addition of **14.8.7 Variation of A/D conversion results**
p. 650	Addition of description to **14.9 (6) Differential linearity error**
p. 656	Deletion of **Note** from (6) in **15.4.3 Cautions**
pp. 664, 665	Modification of description in **16.4 (1) UARTAn control register 0 (UAnCTL0)**
p. 666	Addition of description to **16.4 (4) UARTAn option control register 0 (UAnOPT0)**
p. 666	Deletion of **Caution** in **16.4 (5) UARTAn status register (UAnSTR)**
p. 675	Modification of description in **16.6.4 UART reception**
p. 676	Modification of **Caution** in **16.6.5 Reception errors**
p. 693	Modification of description in **17.4 (1) CSIBn control register 0 (CBnCTL0)**
p. 694	Addition of **17.4 (1) (a) How to use CBnSCE bit**
p. 698	Addition of **Caution** to **17.4 (4) CSIBn status register (CBnSTR)**
p. 699	Addition of **17.5.1 Single transfer mode (master mode, transmission mode)**
p. 701	Addition of **17.5.2 Single transfer mode (master mode, reception mode)**
p. 703	Addition of **17.5.3 Single transfer mode (master mode, transmission/reception mode)**
p. 705	Addition of **17.5.4 Single transfer mode (slave mode, transmission mode)**
p. 707	Addition of **17.5.5 Single transfer mode (slave mode, reception mode)**
p. 709	Addition of **17.5.6 Single transfer mode (slave mode, transmission/reception mode)**
p. 711	Addition of **17.5.7 Continuous transfer mode (master mode, transmission mode)**
p. 713	Addition of **17.5.8 Continuous transfer mode (master mode, reception mode)**
p. 716	Addition of **17.5.9 Continuous transfer mode (master mode, transmission/reception mode)**
p. 720	Addition of **17.5.10 Continuous transfer mode (slave mode, transmission mode)**
p. 722	Addition of **17.5.11 Continuous transfer mode (slave mode, reception mode)**
p. 725	Addition of **17.5.12 Continuous transfer mode (slave mode, transmission/reception mode)**
p. 729	Addition of **17.5.13 Reception error**
pp. 730, 731	Addition of **Caution** to **17.5.14 Clock timing**

(4/5)

Page	Description
p. 736	Modification of **Figure 18-2 I^2C Block Diagram**
p. 739	Addition of **18.3 (13) Stop condition generator**
pp. 740 to 744	Modification of description in **18.4 (1) IIC control register (IICC)**
pp. 746, 747	Modification of description in **18.4 (2) IIC status register (IICS)**
p. 750	Addition of description to **18.4 (3) IIC flag register (IICF)**
p. 751	Addition of description to **18.4 (4) IIC clock selection register (IICCL)**
p. 752	Addition of description to **18.4 (5) IIC function expansion register (IICX)**
p. 753	Addition of description to **18.4 (7) Prescaler compare register (PRSCM)**
p. 755	Addition of description to **18.4 (9) IIC shift register (IIC)**
p. 755	Addition of description to **18.4 (10) Slave address register (SVA)**
p. 764	Addition of **18.6.7 Wait state cancellation method**
p. 765	Modification of description in **18.7.1 (1) Start ~ Address ~ Data ~ Data ~ Stop (normal transmission/reception)**
p. 766	Modification of description in **18.7.1 (2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)**
p. 767	Addition of **Note** to **18.7.1 (3) <1> When WTIM bit = 0**
p. 771	Modification of description in **18.7.2 (4) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop**
p. 775	Modification of description in **18.7.3 (4) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop**
p. 776	Addition of description to **18.7.5 Arbitration loss operation (operation as slave after arbitration loss)**
p. 778	Addition of description to **18.7.6 Operation when arbitration loss occurs (no communication after arbitration loss)**
p. 782	Addition of **18.7.6 (6) <1> When WTIM bit = 0**
p. 783	Addition of **18.7.6 (7) <1> When WTIM bit = 0**
p. 784	Addition of **18.7.6 (8) <1> When WTIM bit = 0**
p. 787	Addition of description to (3) in **18.11 Extension Code**
pp. 790, 791	Modification of description in **18.14.1 When communication reservation function is enabled (IICF.IICRSV bit = 0)**
p. 794	Addition of **Table 18-8 Wait Periods**
p. 795	Addition of (3) to (6) in **18.15 Cautions**
p. 796	Addition of description in **18.16 Communication Operations**
p. 797	Modification of description in **18.16.1 Master operation in single master system**
p. 798	Modification of description in **18.16.2 Master operation in multimaster system**
p. 802	Modification of **Figure 18-19 Slave Operation Flowchart (1)**
p. 803	Modification of **Figure 18-20 Slave Operation Flowchart (2)**
pp. 805 to 807	Modification of **Figure 18-21 Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave)**
pp. 808 to 810	Modification of **Figure 18-22 Example of Slave to Master Communication (When 8-Clock (Master)/9-Clock (Slave) Wait Is Selected)**
p. 826	Addition of **Caution** to **19.3.8 DMA interface control register (DIFC)**
p. 842	Modification of description in **Table 19-2 Minimum Value of Active Width of $\overline{\text{DMAAK}}$n Signal During 2-Cycle Transfer**
pp. 851, 852	Modification of description in **Figure 19-18 Timing During DMA Flyby Transfer (External I/O → SRAM)**
p. 861	Modification of description in **Table 19-5 Number of Minimum Execution Internal System Clocks in DMA Cycle**

(5/5)

Page	Description
p. 862	Deletion of **Caution** in **19.14 Maximum Response Time for DMA Transfer Request**
p. 863	Deletion of (8) and (9) in **19.15 Cautions**
p. 865	Modification of description in **Table 20-1 Interrupt Source List**
p. 909	Modification of description in **Table 21-5 Operation Status in IDLE Mode**
p. 911	Modification of description in **Table 21-7 Operation Status in Software STOP Mode**
p. 918	Addition of **Caution** to **Figure 22-2 Reset Operation at Power On**
p. 921	Deletion and modification of **Caution** in **23.3 ROM Correction Operation and Program Flow**
p. 935	Deletion of description in **24.4 Cautions**
p. 960	Addition of supply current specifications of mask ROM versions in normal operation mode (I_{DD1}) and HALT mode (I_{DD2}) to **DC Characteristics** in **26.1 Normal Operation Mode**
p. 964	Addition of **Remark** to **(1) Clock timing** of **AC Characteristics** in **26.1 Normal Operation Mode**
p. 969	Modification of specifications of $\overline{\text{WAIT}}$ hold time (from $\overline{\text{BCYST}}\uparrow$) ($t_{HBSW}$) and deletion of **Note** in **(4) (b) Read timing (SRAM, external ROM, external I/O)** of **AC Characteristics** in **26.1 Normal Operation Mode**
p. 971	Modification of specifications of $\overline{\text{WAIT}}$ hold time (from $\overline{\text{BCYST}}\uparrow$) ($t_{HBSW}$) and deletion of **Note** in **(4) (c) Write timing (SRAM, external ROM, external I/O)** of **AC Characteristics** in **26.1 Normal Operation Mode**
p. 973	Modification of specifications of $\overline{\text{WAIT}}$ hold time (from $\overline{\text{BCYST}}\uparrow$) ($t_{HBSW}$) and deletion of **Note** in **(4) (d) DMA flyby transfer timing (SRAM → external I/O transfer)** of **AC Characteristics** in **26.1 Normal Operation Mode**
p. 975	Modification of specifications of $\overline{\text{WAIT}}$ hold time (from $\overline{\text{BCYST}}\uparrow$) ($t_{HBSW}$) and Delay time from $\overline{\text{IORD}}\uparrow$ to $\overline{\text{DMAAK}}\text{m}\uparrow$ (t_{DRDDA}), and deletion of **Note** in **(4) (e) DMA flyby transfer timing (external I/O → SRAM transfer)** of **AC Characteristics** in **26.1 Normal Operation Mode**
p. 994	Addition of **Note** to **(11) Timer P (TMP), timer Q (TMQ) timing** of **AC Characteristics** in **26.1 Normal Operation Mode**
p. 1003	Addition of **26.2 Power-On/Off Sequence**
p. 1005	Addition of **Serial Write Operation Characteristics** to **26.3 Flash Memory Programming Mode (μPD70F3134A, 70F3134AY Only)**
p. 1008	Addition of **CHAPTER 28 RECOMMENDED SOLDERING CONDITIONS**
p. 1034	Addition of **C.2 Revision History of Preceding Editions**

<R> C.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/6)

Edition	Description	Chapter
3rd	• Modification of description on output from the SCL and SDA pins N-ch open drain output → Dummy open drain output (P-ch side is always off) • Addition of part numbers μPD703131A, 703131AY, 703132A, 703132AY, 703133A, 703133AY, 703134A, 703134AY, 70F3134A, 70F3134AY • Under development → Mass production μPD703131GJ-xxx-UEN, μPD703132GJ-xxx-UEN, μPD70F3134GJ-UEN, μPD70F3134AGJ-UEN, μPD70F3134YGJ-UEN, μPD70F3134AYGJ-UEN • Addition of registers Chip select signal delay control register (CSDC) DMAAK width control register (DAKW) Write access synchronization control register (WAS)	Throughout
	Addition of **Table 1-1 V850E/MA3 Product List**	**CHAPTER 1 INTRODUCTION**
	Modification of description in **1.4 Ordering Information**	
	Addition of **Caution** to **2.4 Pin I/O Circuits**	**CHAPTER 2 PIN FUNCTIONS**
	Addition of **Note** to **Table 3-2 System Register Numbers**	**CHAPTER 3 CPU FUNCTION**
	Addition of **Note** to **3.4.8 On-chip peripheral I/O registers**	
	Modification of description in **3.4.9 Special registers**	
	Modification of description in **3.4.9 (1) Setting data to special registers**	
	Addition of **Remark** to **3.4.10 System wait control register (VSWC)**	
	Addition of description to **3.4.11 (1) Registers to be set first**	
	Addition of description to **3.4.11 (2) Restriction on conflict between sld instruction and interrupt request**	
	Addition of **Caution** to **Table 4-3 Alternate-Function Pins of Port 0**	**CHAPTER 4 PORT FUNCTIONS**
	Addition of **Caution** to **Table 4-4 Alternate-Function Pins of Port 1**	
	Modification of **Figure 4-7 Block Diagram of P10 to P13 Pins**	
	Addition of description to **4.3.3 Port 2**	
	Addition of **Caution** to **Table 4-5 Alternate-Function Pins of Port 2**	
	Modification of **Figure 4-12 Block Diagram of P25 Pin**	
	Addition of **Caution** to **Table 4-6 Alternate-Function Pins of Port 3**	
	Modification of **Note** in **4.3.4 (1) (f) Setting of alternate functions of port 3 pins**	
	Modification of **Figure 4-18 Block Diagram of P33 Pin**	
	Modification of **Figure 4-20 Block Diagram of P37 Pin**	
	Addition of **Caution** to **Table 4-7 Alternate-Function Pins of Port 4**	
	Addition of **Caution** to **Table 4-8 Alternate-Function Pins of Port 5**	
	Addition of **Caution** to **4.3.13 (1) (d) Port CT function control register (PFCCT)**	
	Modification of **Note** in **Table 4-19 Using Alternate Function of Port Pins (4/9)**	
	Addition of **Note** in **Table 4-19 Using Alternate Function of Port Pins (8/9)**	
	Modification of **Caution 2** in **Table 4-21 Noise Elimination Time of Timer ENC1 Input Pins**	
	Modification of description in **4.6 Cautions**	

(2/6)

Edition	Description	Chapter
3rd	Modification of **Caution 1** in **5.4.1 (1) Bus cycle type configuration registers 0, 1 (BCT0, BCT1)**	**CHAPTER 5 BUS CONTROL FUNCTION**
	Change of the number of clocks for operand data access to internal ROM in the table in **5.5.1 Number of clocks for access**	
	Addition of **5.6 Write Buffer Function**	
	Addition of **Caution 4** to **5.8.1 (2) Address setup wait control register (ASC)**	
	Modification of bit description in **5.8.1 (5) DMA flyby transfer wait control register (FWC)**	
	Modification of description in **Table 5-2 (a) In separate bus mode**	
	Addition of **5.10.4 Bus hold timing**	
	Addition of **5.10.5 Bus hold timing (SRAM)**	
	Addition of **5.10.6 Bus hold timing (SDRAM)**	
	Deletion of description in **5.11 Bus Priority**	
	Modification of timing in **Figure 6-3 (e) Write (32-bit access (1/2))**	**CHAPTER 6 MEMORY ACCESS CONTROL FUNCTION**
	Modification of timing in **Figure 6-3 (f) Write (16-bit access (1/2))**	
	Addition of description to **Figure 6-4 (b) Read (successive 16-bit access)**	
	Modification of description in **6.2.3 On-page**	
	Modification of description on register in **6.2.4 Page ROM configuration register (PRC)**	
	Addition of **6.2.5 Page ROM access**	
	Deletion of a part of description on CAS latency in **6.3.1 Features**	
	Addition of **6.3.3 (1) Output of each address and connection of SDRAM** and **(2) Bank address output**	
	Modification of **Caution 4** and description on register in **6.3.4 SDRAM configuration registers 1, 3, 4, 6 (SCR1, SCR3, SCR4, SCR6)**	
	Addition of **Figure 6-10 SDRAM Single Read Cycle**	
	Deletion of a part of description in **6.3.5 (2) SDRAM single write cycle**	
	Addition of **Figure 6-11 SDRAM Single Write Cycle**	
	Deletion of a part of description in **6.3.5 (3) (c) CAS latency setting when read**	
	Addition of **Figure 6-12 SDRAM Access Timing**	
	Modification of description in **6.3.6 (1) SDRAM refresh control registers 1, 3, 4, 6 (RFS1, RFS3, RFS4, RFS6)**	
	Addition of **Figure 6-14 CBR (Automatic) Refresh Cycle (16-Bit Bus Width)**	
	Addition of **Figure 6-15 Self Timing (16-Bit Bus Width)**	
	Modification of description in **6.3.8 SDRAM initialization sequence**	
	Addition of **Figure 6-16 SDRAM Register Write Operation Timing (16-Bit Bus Width)**	
	Modification of **Cautions** in **7.3 (2) Clock control register (CKC)**	**CHAPTER 7 CLOCK GENERATOR**
	Modification and addition of **Notes** and addition of **Caution 2** in **7.3 (3) Power save control register (PSC)**	
	Addition of **Caution 4** to **7.3 (5) Oscillation stabilization time select register (OSTS)**	
	Modification of **CHAPTER 8 16-BIT TIMER/EVENT COUNTER P (TMP)**	**CHAPTER 8 16-BIT TIMER/EVENT COUNTER P (TMP)**
	Modification of **CHAPTER 9 16-BIT TIMER/EVENT COUNTER Q (TMQ)**	**CHAPTER 9 16-BIT TIMER/EVENT COUNTER Q (TMQ)**

(3/6)

Edition	Description	Chapter
3rd	Modification of description in **11.4 (5) Prescaler mode register 10 (PRM10)**	**CHAPTER 11 16-BIT 2-PHASE ENCODER INPUT UP/DOWN COUNTER/GENERAL-PURPOSE TIMER (TMENC1)**
	Addition of description to **11.5.1 (5) Timer output operation**	
	Modification of description in **Table 11-4 List of Count Operations in UDC Mode**	
	Modification of description in **12.1 Functional Overview**	**CHAPTER 12 MOTOR CONTROL FUNCTION**
	Modification of **Figure 12-1 Block Diagram of Motor Control**	
	Modification of **Figure 12-2 TMQ0 Option**	
	Addition of **Caution** to **12.3 (3) TMQ0 option register 2 (TQ0OPT2)**	
	Addition of description to **12.3 (4) TMQ0I/O control register 3 (TQ0IOC3)**	
	Modification of description in **12.3 (5) High-impedance output control registers 00, 01 (HZA0CTLn)**	
	Modification of **Figure 12-5 Outline of 6-Phase PWM Output Mode**	
	Deletion from **Cautions** and modification in **Figure 12-6 Timing Chart of 6-Phase PWM Output Mode**	
	Modification of **Figure 12-7 Interrupt and Up/Down Flag**	
	Modification of description in **Figure 12-12 (b) In vicinity of 0% output (TQ0CCRm register = i (M + 1 (a/2, TQ0CCR0 register = M, TQ0DTC register = a)**	
	Modification of description in **Figure 12-13 (b) In vicinity of 100% output (TQ0CCRm register = i (a/2, TQ0CCR0 register = M, TQ0DTC register = a)**	
	Modification of **Caution** in **12.4.3 Interrupt culling function**	
	Deletion of a portion of the timings in **Figure 12-18 Crest/Valley Interrupt Output**	
	Deletion of a portion of the timings in **Figure 12-19 Crest Interrupt Output**	
	Deletion of a portion of the timings in **Figure 12-20 Valley Interrupt Output**	
	Addition of description on batch rewrite mode (transfer mode) in **12.4.4 Operation to rewrite register with transfer function**	
	Modification of **Note** in **Figure 12-21 Timing of Reflecting Rewritten Value**	
	Addition of description to **12.4.4 (2) Batch rewrite mode (transfer mode)**	
	Modification of description in **Figure 12-26 Basic Operation in Batch Mode**	
	Modification of description in **12.4.4 (2) (d) Transferring TQ0OPT1 register value**	
	Addition of description to **12.4.4 (3) Intermittent batch rewrite mode (transfer culling mode)**	
	Modification of description in **Figure 12-31 Basic Operation in Intermittent Batch Rewrite Mode**	
	Modification of **Figure 12-34 Rewriting TQ0CCR1 Register (TQ0OPT1.TQ0ICE bit = 1, TQ0OPT1.TQ0IOE bit = 0, TQ0OPT1.TQ0ID4 to TQ0OPT1.TQ0ID0 bits = 00001)**	
	Modification of **Figure 12-35 Rewriting TQ0CCR1 Register (TQ0OPT1.TQ0ICE bit = 1, TQ0OPT1.TQ0IOE bit = 1, TQ0OPT1.TQ0ID4 to TQ0OPT1.TQ0ID0 bits = 00001)**	
	Modification of description in **12.4.4 (4) Rewriting TQ0OPT0.TQ0CMS bit**	
	Addition of description to **12.4.5 (1) (b) Setting of TMQ0 register**	
	Modification of **Note 2** in **13.3 (2) Watchdog timer mode register (WDTM)**	**CHAPTER 13 WATCHDOG TIMER FUNCTIONS**
	Deletion of description in **13.4.1 Operation as watchdog timer**	
	Deletion of description in **13.4.2 Operation as interval timer**	

(4/6)

Edition	Description	Chapter
3rd	Addition of description to **14.2 (13) AV_{DD0} pin**	**CHAPTER 14 A/D CONVERTER**
	Modification of **Caution 2** in **14.3 (3) A/D converter mode register 2 (ADM2)**	
	Addition of **Figure 14-12 Timer Trigger Scan Operation Timing: 8-Channel Scan (ANI0 to ANI7)**	
	Modification of description on settling time in **15.1 Functions**	**CHAPTER 15 D/A CONVERTER**
	Addition of **Caution** to **15.3 (1) D/A converter mode register (DA0M)**	
	Addition of **Caution** to **15.3 (2) D/A conversion value setting registers 0, 1 (DA0CS0, DA0CS1)**	
	Modification of (5) and addition of (6) in **15.4.3 Cautions**	
	Modification of **Note** in **Figure 16-2 UARTA2/CSIB2 Mode Switch Settings**	**CHAPTER 16 ASYNCHRONOUS SERIAL INTERFACE A (UARTA)**
	Modification of **Note** in **Figure 16-3. UARTA3/I^2C Mode Switch Settings**	
	Modification of description in **16.4 (1) UARTAn control register 0 (UAnCTL0)**	
	Addition of **Caution** in **16.4 (5) UARTAn status register (UAnSTR)**	
	Modification of **Cautions** in **16.6.4 UART reception**	
	Addition of **Caution** in **16.6.5 Reception errors**	
	Modification of **Figure 16-10 Noise Filter Circuit**	
	Addition of **Figure 16-11 Timing of RXDn Signal Judged as Noise**	
	Modification of **Caution** in **16.7 (2) UARTAn control register 1 (UAnCTL1)**	
	Modification of **Caution** in **16.7 (3) UARTAn control register 2 (UAnCTL2)**	
	Modification of description in **Table 16-3 Baud Rate Generator Setting Data**	
	Addition of **16.8 Cautions**	
	Modification of **Note** in **Figure 17-2 UARTA2/CSIB2 Mode Switch Settings**	**CHAPTER 17 CLOCKED SERIAL INTERFACE B (CSIB)**
	Addition of **Remark** to **17.3 (2) CSIBn transmit data register (CBnTX)**	
	Modification of description in **17.4 (1) CSIBn control register 0 (CBnCTL0)**	
	Modification of **Caution** in **17.4 (2) CSIBn control register 1 (CBnCTL1)**	
	Modification of description in **17.5 Operation**	
	Modification of description in **17.6 (1) SCKn pin**	
	Modification of description in **17.7 Operation Flow**	
	Modification of description in **CHAPTER 18 I^2C BUS**	**CHAPTER 18 I^2C BUS**
	Modification of **Note** in **Figure 18-1 UARTA3/I^2C Mode Switch Settings**	
	Modification of description in **18.2 (2) I^2C bus mode (multimaster supported)**	
	Modification of **Figure 18-2 I^2C Block Diagram**	
	Modification of description in **18.4 (1) IIC control register (IICC)**	
	Modification of description in **18.4 (4) IIC clock selection register (IICCL)**	
	Addition of description in **18.4 (9) IIC shift register (IIC)**	
	Modification of description in **18.5.1 Pin configuration**	
	Addition of **Caution** to **18.6.1 Start condition**	
	Addition of description in **Figure 18-11 Wait Signal**	
	Modification of **Notes** in **Table 18-3 INTIIC Signal Generation Timing and Wait Control**	
	Addition of description on slave device operation in **18.8 (1) During address transmission/reception**	

(5/6)

Edition	Description	Chapter
3rd	Addition of **18.14.2 When communication reservation function is disabled (IICF.IICRSV bit = 1)**	**CHAPTER 18 I^2C BUS**
	Modification of description in **18.15 (1) When IICF.STCEN bit = 0**	
	Addition of description to **18.15 (2) When IICF.STCEN bit = 1**	
	Modification of description in **18.16 Communication Operations**	
	Modification of description in **Figure 18-22 (c) Stop condition**	
	Deletion of description in **19.3.1 DMA source address registers 0 to 3 (DSA0 to DSA3)**	**CHAPTER 19 DMA FUNCTIONS (DMA CONTROLLER)**
	Deletion of description in **19.3.2 DMA destination address registers 0 to 3 (DDA0 to DDA3)**	
	Deletion of description in **19.3.3 DMA transfer count registers 0 to 3 (DBC0 to DBC3)**	
	Addition of **Caution 5** and modification of description in **19.3.5 DMA channel control registers 0 to 3 (DCHC0 to DCHC3)**	
	Addition of **Caution** and modification of description in **19.3.6 DMA terminal count output control register (DTOC)**	
	Modification and addition of **Cautions** in **19.3.7 DMA trigger factor registers 0 to 3 (DTFR0 to DTFR3)**	
	Modification of description in **19.3.8 DMA interface control register (DIFC)**	
	Addition of **19.3.9 DMAAK width control register (DAKW)**	
	Addition of description and figure to **19.5.1 2-cycle transfer**	
	Addition of description and figure to **19.5.2 Flyby transfer**	
	Addition of **Note 4** and **Caution 4** in **Table 19-3 Relationship Between Transfer Type and Transfer Object**	
	Addition of description in **19.8 (2) During DMA transfer (period start of DMA transfer to completion of DMA transfer)**	
	Modification of description in **19.13 Times Related to DMA Transfer**	
	Modification of description in **19.14 Maximum Response Time for DMA Transfer Request**	
	Modification of description in **19.15 (6) DMA start factors**	
	Modification of description in **19.15 (8) Restrictions related to automatic clearing of DCHCn.TCn bit (non-A products only (See Table 1-1))**	
	Modification of description in **19.15 (9) Restrictions related to DMA transfer when the number of transfers is set to two or more (non-A products only (See Table 1-1))**	
	Addition of **20.2.2 (2) With INTWDT signal**	**CHAPTER 20 INTERRUPT/ EXCEPTION PROCESSING FUNCTION**
	Modification of **Caution** in **20.3.6 In-service priority register (ISPR)**	
	Addition of **Caution** to **20.4.2 Edge detection**	
	Addition of **Caution** to **20.6.2 (2) Restore**	
	Modification of description in **20.9 Periods in Which CPU Does Not Acknowledge Interrupts**	
	Addition of (2) to **20.10 Cautions**	
	Modification of description in **Table 21-1 Standby Modes**	**CHAPTER 21 STANDBY FUNCTION**
	Addition and modification of **Notes** in **Figure 21-1 Status Transition**	
	Addition of **Notes** and **Caution 2** to **21.2 (1) Power save control register (PSC)**	
	Addition of **Caution 2** in **21.3.1 Setting and operation status**	
	Modification of description in **21.3.2 Releasing HALT mode**	
	Modification of **Caution** in **21.4.1 Setting and operation status**	

(6/6)

Edition	Description	Chapter
3rd	Addition of description to **21.4.2 Releasing IDLE mode**	**CHAPTER 21 STANDBY FUNCTION**
	Addition of **Caution** and description to **21.4.2 (1) Releasing IDLE mode by non-maskable interrupt request signal (NMI pin input) or unmasked maskable interrupt request signal**	
	Modification of description in **Table 21-5 Operation Status in IDLE Mode**	
	Modification of **Caution** in **21.5.1 Setting and operation status**	
	Addition of description in **21.5.2 Releasing software STOP mode**	
	Addition of **Caution** and description to **21.5.2 (1) Releasing software STOP mode by non-maskable interrupt request signal (NMI pin input) or unmasked maskable interrupt request signal**	
	Modification of description in **Table 21-7 Operation Status in Software STOP Mode**	
	Addition of **21.7 Procedure for Setting and Restoring from IDLE and Software STOP Modes**	
	Addition and deletion of **Cautions** in **23.3 ROM Correction Operation and Program Flow**	**CHAPTER 23 ROM CORRECTION FUNCTION**
	Modification of **Figure 23-2 ROM Correction Operation and Program Flow**	
	Addition of **24.1.2 (6) Hardware break function**	**CHAPTER 24 ON-CHIP DEBUG FUNCTION (DCU)**
	Addition of **24.1.2 (8) Mask function**	
	Addition of **24.1.2 (9) Timer function**	
	Modification of description in **24.1.3 ROM security function**	
	Addition of **24.2 Selecting On-Chip Debug Function and Port Function (Including Alternate Functions)**	
	Modification of description in **Table 24-1 Emulator Connector Pin Functions (on Target System Side)**	
	Modification of **Figure 24-3 Example of Recommended Emulator Connector**	
	Modification of description in **Table 24-2 Emulator Connector Pin Functions (on Target System Side)**	
	Modification of **Figure 24-6 Example of Recommended Emulator Connector**	
	Addition of **24.4 Cautions**	
	Addition of description in **25.2 (2) Off-board programming**	**CHAPTER 25 FLASH MEMORY**
	Deletion of description **25.5.2 (2) Malfunction of other device**	
	Modification of **Figure 25-10 Flash Memory Operation Flow**	
	Deletion of description in **Table 25-8 Flash Memory Control Command**	
	Addition of **CHAPTER 26 ELECTRICAL SPECIFICATIONS**	**CHAPTER 26 ELECTRICAL SPECIFICATIONS**
	Addition of **CHAPTER 27 PACKAGE DRAWINGS**	**CHAPTER 27 PACKAGE DRAWINGS**
	Modification of description in **B.2 Instruction Set (in Alphabetical Order)**	**APPENDIX B INSTRUCTION SET LIST**
	Addition of **APPENDIX C REVISION HISTORY**	**APPENDIX C 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
Podbielski Strasse 166 B
30177 Hanover
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. 52180
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
Limburglaan 5
5616 HR 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/

NEC Electronics Shanghai Ltd.
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/

NEC Electronics Hong Kong Ltd.
12/F., Cityplaza 4,
12 Taikoo Wan Road, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

Seoul Branch
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-2719-2377

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/