

V850ES/JC3-L, V850ES/JE3-L

User's Manual: Hardware

RENESAS MCU

V850ES/Jx3-L Microcontrollers

— Preliminary —

V850ES/JC3-L

V850ES/JE3-L

μ PD70F3797	μ PD70F3802	μ PD70F3805
μ PD70F3798	μ PD70F3803	μ PD70F3806
μ PD70F3799	μ PD70F3804	μ PD70F3807
μ PD70F3800	μ PD70F3838	μ PD70F3808
μ PD70F3801	μ PD70F3839	μ PD70F3840

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (<http://www.renesas.com>).

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas 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 Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
 - “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.
 - “High Quality”: 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; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

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.

How to Use This Manual

- Readers** This manual is intended for users who wish to understand the functions of the V850ES/JC3-L, V850ES/JE3-L and design application systems using these products.
- Purpose** This manual is intended to give users an understanding of the hardware functions of the V850ES/JC3-L, V850ES/JE3-L shown in the **Organization** below.
- Organization** This manual is divided into two parts: Hardware (this manual) and Architecture (**V850ES Architecture User's Manual**).

Hardware

- Pin functions
- CPU function
- On-chip peripheral functions
- Flash memory programming
- Electrical specifications (Target)

Architecture

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

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

To understand the overall functions of the V850ES/JC3-L, V850ES/JE3-L
→ Read this manual according to the **CONTENTS**.

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

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

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

To know the electrical specifications of the V850ES/JC3-L (40-pin)
→ See **CHAPTER 30 ELECTRICAL SPECIFICATIONS (V850ES/JC3-L (40-pin)) (Target)**

To know the electrical specifications of the V850ES/JC3-L (48-pin)
→ See **CHAPTER 31 ELECTRICAL SPECIFICATIONS (V850ES/JC3-L (48-pin)) (Target)**

To know the electrical specifications of the V850ES/JE3-L
→ See **CHAPTER 32 ELECTRICAL SPECIFICATIONS (V850ES/JE3-L) (Target)**

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.

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Active low representation:	$\overline{\text{xxx}}$ (overscore over pin or signal name)
Memory map address:	Higher addresses on the top and lower addresses on the bottom
Note:	Footnote for item marked with Note in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numeric representation:	Binary ... xxxx or xxxxB Decimal ... xxxx Hexadecimal ... xxxxH
Prefix indicating power of 2 (address space, memory capacity):	K (kilo): $2^{10} = 1,024$ M (mega): $2^{20} = 1,024^2$ G (giga): $2^{30} = 1,024^3$

Related Documents

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

Documents related to V850ES/JC3-L, V850ES/JE3-L

Document Name	Document No.
V850ES Architecture User's Manual	U15943E
V850ES/JC3-L, V850ES/JE3-L Hardware User's Manual	This manual

Documents related to development tools

Document Name	Document No.	
QB-V850ESJX3L In-Circuit Emulator	To be prepared	
QB-V850MINI, QB-V850MINIL On-Chip Debug Emulator	U17638E	
QB-MINI2 On-Chip Debug Emulator with Programming Function	U18371E	
CA850 Ver. 3.20 C Compiler Package	Operation	U18512E
	C Language	U18513E
	Assembly Language	U18514E
	Link Directives	U18415E
PM+ Ver. 6.30 Project Manager	U18416E	
ID850QB Ver. 3.40 Integrated Debugger	Operation	U18604E
SM850 Ver. 2.50 System Simulator	Operation	U16218E
SM850 Ver. 2.00 or Later System Simulator	External Part User Open Interface Specification	U14873E
SM+ System Simulator	Operation	U18601E
	User Open Interface	U18212E
RX850 Ver. 3.20 Real-Time OS	Basics	U13430E
	Installation	U17419E
	Technical	U13431E
	Task Debugger	U17420E
RX850 Pro Ver. 3.21 Real-Time OS	Basics	U18165E
	Installation	U17421E
	Task Debugger	U17422E
AZ850 Ver. 3.30 System Performance Analyzer	U17423E	
PG-FP5 Flash Memory Programmer	U18865E	

Other Documents

Document Name	Document No.
SEMICONDUCTOR SELECTION GUIDE – Products and Packages –	X13769X
Semiconductor Device Mount Manual	Note
Quality Grades on Renesas Semiconductor Devices	C11531E
Renesas Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

Note See the “Semiconductor Device Mount Manual” website (<http://www2.renesas.com/pkg/en/mount/index.html>).

Caution The related documents listed above are subject to change without notice. Be sure to use the latest version of each document when designing.

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

IECUBE is a registered trademark of Renesas Electronics Corporation in Japan and Germany.

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

EEPROM is a trademark of Renesas Electronics Corporation

Applilet is a registered trademark of Renesas Electronics in Japan, Germany, Hong Kong, China, the Republic of Korea, the United Kingdom, and the United States of America.

Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

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

PC/AT is a trademark of International Business Machines Corporation.

SPARCstation is a trademark of SPARC International, Inc.

Solaris and SunOS are trademarks of Sun Microsystems, Inc.

TRON is an abbreviation of The Realtime Operating System Nucleus.

ITRON is an abbreviation of Industrial TRON.

Table of Contents

CHAPTER 1 INTRODUCTION	19
1.1 General	19
1.2 Features	23
1.3 Application Fields	25
1.4 Ordering Information	25
1.4.1 V850ES/JC3-L	25
1.4.2 V850ES/JE3-L	25
1.5 Pin Configuration (Top View)	26
1.6 Function Block Configuration	31
1.6.1 Internal block diagram.....	31
1.6.2 Internal units	34
CHAPTER 2 PIN FUNCTIONS	37
2.1 List of Pin Functions	37
2.2 Pin States	45
2.3 Pin I/O Circuit Types, I/O Buffer Power Supplies, and Connection of Unused Pins	46
2.4 Cautions	49
CHAPTER 3 CPU FUNCTION	50
3.1 Features	50
3.2 CPU Register Set	51
3.2.1 Program register set	52
3.2.2 System register set	53
3.3 Operation Modes	59
3.4 Address Space	60
3.4.1 CPU address space	60
3.4.2 Memory map	61
3.4.3 Areas	63
3.4.4 Wraparound of data space.....	68
3.4.5 Recommended use of address space.....	68
3.4.6 Peripheral I/O registers	72
3.4.7 Special registers	82
3.4.8 Registers to be set first	86
3.4.9 Cautions.....	87
CHAPTER 4 PORT FUNCTIONS	89
4.1 Features	89
4.1.1 V850ES/JC3-L (40-pin)	89
4.1.2 V850ES/JC3-L (48-pin)	89
4.1.3 V850ES/JE3-L	89
4.2 Basic Port Configuration	90
4.2.1 V850ES/JC3-L (40-pin)	90
4.2.2 V850ES/JC3-L (48-pin)	91
4.2.3 V850ES/JE3-L	92
4.3 Port Configuration	93
4.3.1 Port 0	99
4.3.2 Port 1 (V850ES/JC3-L (48-pin), V850ES/JE3-L).....	105

4.3.3	Port 3	106
4.3.4	Port 4	114
4.3.5	Port 5	116
4.3.6	Port 7	121
4.3.7	Port 9	124
4.3.8	Port CM.....	136
4.3.9	Port DL.....	137
4.4	Block Diagrams	139
4.5	Port Register Settings When Alternate Function Is Used.....	171
4.6	Cautions	177
4.6.1	Cautions on setting port pins.....	177
4.6.2	Cautions on bit manipulation instruction for port n register (Pn).....	180
4.6.3	Cautions on on-chip debug pins.....	181
4.6.4	Cautions on P05/INTP2/DRST pin.....	181
4.6.5	Cautions on P10, P11, and P53 pins when power is turned on	181
4.6.6	Hysteresis characteristics	181
CHAPTER 5	CLOCK GENERATOR	182
5.1	Overview	182
5.2	Configuration.....	183
5.3	Registers	185
5.4	Operations	190
5.4.1	Operation of each clock	190
5.4.2	External clock signal input.....	190
5.5	PLL Function	191
5.5.1	Overview	191
5.5.2	Registers.....	191
5.5.3	Usage	195
5.6	How to Connect a Resonator	196
5.6.1	Main clock oscillator.....	196
5.6.2	Subclock oscillator	196
CHAPTER 6	16-BIT TIMER/EVENT COUNTER P (TMP)	199
6.1	Overview	199
6.2	Configuration.....	200
6.2.1	Pins used by TMPn.....	202
6.2.2	Register configuraiton	203
6.2.3	Interrupts.....	206
6.3	Registers	207
6.4	Operations	219
6.4.1	Interval timer mode (TPnMD2 to TPnMD0 bits = 000)	227
6.4.2	External event count mode (TPnMD2 to TPnMD0 bits = 001)	238
6.4.3	External trigger pulse output mode (TPnMD2 to TPnMD0 bits = 010)	247
6.4.4	One-shot pulse output mode (TPnMD2 to TPnMD0 bits = 011).....	259
6.4.5	PWM output mode (TPnMD2 to TPnMD0 bits = 100)	267
6.4.6	Free-running timer mode (TPnMD2 to TPnMD0 bits = 101)	276
6.4.7	Pulse width measurement mode (TPnMD2 to TPnMD0 bits = 110).....	292
6.4.8	Timer output operations	296
6.5	Selector (V850ES/JE3-L only)	297
6.6	Cautions	298

CHAPTER 7 16-BIT TIMER/EVENT COUNTER Q (TMQ)	299
7.1 Functions	299
7.2 Configuration	300
7.2.1 Pins used by TMQ0	302
7.2.2 Interrupts.....	302
7.3 Registers	303
7.4 Operations	318
7.4.1 Interval timer mode (TQ0MD2 to TQ0MD0 bits = 000)	325
7.4.2 External event count mode (TQ0MD2 to TQ0MD0 bits = 001)	336
7.4.3 External trigger pulse output mode (TQ0MD2 to TQ0MD0 bits = 010)	347
7.4.4 One-shot pulse output mode (TQ0MD2 to TQ0MD0 bits = 011)	362
7.4.5 PWM output mode (TQ0MD2 to TQ0MD0 bits = 100)	372
7.4.6 Free-running timer mode (TQ0MD2 to TQ0MD0 bits = 101).....	383
7.4.7 Pulse width measurement mode (TQ0MD2 to TQ0MD0 bits = 110)	403
7.4.8 Timer output operations	408
7.5 Cautions	409
 CHAPTER 8 16-BIT INTERVAL TIMER M (TMM)	 410
8.1 Features	410
8.2 Configuration	411
8.3 Registers	412
8.4 Operation	414
8.4.1 Interval timer mode	414
8.4.2 Cautions.....	418
 CHAPTER 9 WATCH TIMER	 419
9.1 Functions	419
9.2 Configuration	420
9.3 Control Registers	422
9.4 Operation	426
9.4.1 Watch timer operations	426
9.4.2 Interval timer operations.....	427
9.5 Cautions	429
 CHAPTER 10 REAL-TIME COUNTER	 430
10.1 Functions	430
10.2 Configuration	431
10.2.1 Pin configuration	433
10.2.2 Interrupt functions	433
10.3 Registers	434
10.4 Operation	449
10.4.1 Initial settings	449
10.4.2 Rewriting each counter during real-time counter operation.....	450
10.4.3 Reading each counter during real-time counter operation	451
10.4.4 Changing INTRTC0 interrupt setting during real-time counter operation	452
10.4.5 Changing INTRTC1 interrupt setting during real-time counter operation	453
10.4.6 Initial INTRTC2 interrupt settings	454
10.4.7 Changing INTRTC2 interrupt setting during real-time counter operation	455
10.4.8 Initializing real-time counter	456

10.4.9 Watch error correction example of real-time counter	457
CHAPTER 11 WATCHDOG TIMER 2	461
11.1 Functions	461
11.2 Configuration.....	462
11.3 Registers	463
11.4 Operation	465
CHAPTER 12 REAL-TIME OUTPUT FUNCTION (RTO).....	466
12.1 Function	466
12.2 Configuration.....	467
12.3 Registers	469
12.4 Operation	471
12.5 Usage.....	472
12.6 Cautions	472
CHAPTER 13 A/D CONVERTER	473
13.1 Overview	473
13.2 Functions	473
13.3 Configuration.....	474
13.4 Registers	477
13.5 Operation	488
13.5.1 Basic operation	488
13.5.2 Conversion timing	489
13.5.3 Trigger modes.....	490
13.5.4 Operation mode	492
13.5.5 Power-fail compare mode	498
13.6 Cautions	505
13.7 How to Read A/D Converter Characteristics Table	510
CHAPTER 14 D/A CONVERTER (V850ES/JC3-L (48-pin), V850ES/JE3-L)	514
14.1 Functions	514
14.2 Configuration.....	515
14.3 Registers	516
14.4 Operation	518
14.4.1 Operation in normal mode.....	518
14.4.2 Operation in real-time output mode.....	518
14.4.3 Cautions.....	519
CHAPTER 15 ASYNCHRONOUS SERIAL INTERFACE A (UARTA).....	520
15.1 Features	520
15.2 Configuration.....	521
15.2.1 Pin functions of each channel	523
15.3 Mode Switching of UARTA and Other Serial Interfaces.....	524
15.3.1 UARTA0 and CSIB4 mode switching.....	524
15.3.2 UARTA1 and I ² C02 mode switching.....	525
15.3.3 UARTA2 and I ² C00 mode switching.....	526
15.4 Registers	527
15.5 Interrupt Request Signals	534
15.6 Operation	535

15.6.1 Data format	535
15.6.2 UART transmission	537
15.6.3 Continuous transmission procedure.....	538
15.6.4 UART reception	540
15.6.5 Reception errors	542
15.6.6 Parity types and operations.....	544
15.6.7 LIN transmission/reception format	545
15.6.8 SBF transmission.....	547
15.6.9 SBF reception	548
15.6.10 Receive data noise filter	549
15.7 Dedicated Baud Rate Generator.....	550
15.8 Cautions.....	558
CHAPTER 16 CLOCKED SERIAL INTERFACE B (CSIB).....	559
16.1 Features	559
16.2 Configuration.....	560
16.2.1 Pin functions of each channel	561
16.3 Mode Switching of CSIB and Other Serial Interfaces.....	562
16.3.1 CSIB0 and I ² C01 mode switching	562
16.3.2 CSIB4 and UARTA0 mode switching.....	563
16.4 Registers	564
16.5 Interrupt Request Signals	573
16.6 Operation	574
16.6.1 Single transfer mode (master mode, transmission mode).....	574
16.6.2 Single transfer mode (master mode, reception mode)	576
16.6.3 Single transfer mode (master mode, transmission/reception mode)	578
16.6.4 Single transfer mode (slave mode, transmission mode)	580
16.6.5 Single transfer mode (slave mode, reception mode).....	582
16.6.6 Single transfer mode (slave mode, transmission/reception mode).....	585
16.6.7 Continuous transfer mode (master mode, transmission mode).....	587
16.6.8 Continuous transfer mode (master mode, reception mode).....	589
16.6.9 Continuous transfer mode (master mode, transmission/reception mode)	592
16.6.10 Continuous transfer mode (slave mode, transmission mode)	596
16.6.11 Continuous transfer mode (slave mode, reception mode).....	598
16.6.12 Continuous transfer mode (slave mode, transmission/reception mode).....	601
16.6.13 Reception errors.....	604
16.6.14 Clock timing.....	605
16.7 Output Pins	607
16.8 Baud Rate Generator	608
16.8.1 Baud rate generation	609
16.9 Cautions.....	610
CHAPTER 17 I²C BUS	611
17.1 Mode Switching of I²C Bus and Other Serial Interfaces.....	612
17.1.1 UARTA2 and I ² C00 mode switching.....	612
17.1.2 CSIB0 and I ² C01 mode switching	613
17.1.3 UARTA1 and I ² C02 mode switching.....	614
17.2 Features	615
17.3 Configuration.....	616
17.4 Registers.....	620

17.5.1 Pin configuration	636
17.6 I²C Bus Definitions and Control Methods	637
17.6.1 Start condition	637
17.6.2 Addresses	638
17.6.3 Transfer direction specification	639
17.6.4 $\overline{\text{ACK}}$	640
17.6.5 Stop condition	641
17.6.6 Wait state	642
17.6.7 Wait state cancellation method	644
17.7 I²C Interrupt Request Signals (INTIICn).....	645
17.7.1 Master device operation.....	645
17.7.2 Slave device operation (when receiving slave address data (address match))	648
17.7.3 Slave device operation (when receiving extension code).....	652
17.7.4 Operation without communication	656
17.7.5 Operation when arbitration loss occurs (operation as slave after arbitration loss)	656
17.7.6 Operation when arbitration loss occurs (no communication after arbitration loss)	658
17.8 Interrupt Request Signal (INTIICn) Generation Timing and Wait Control	665
17.9 Address Match Detection Method	667
17.10 Error Detection	667
17.11 Extension Code	667
17.12 Arbitration	668
17.13 Wakeup Function	669
17.14 Communication Reservation	670
17.14.1 When communication reservation function is enabled (IICFn.IICRSVn bit = 0).....	670
17.14.2 When communication reservation function is disabled (IICFn.IICRSVn bit = 1)	674
17.15 Cautions	675
17.16 Communication Operations	676
17.16.1 Master operation in single master system	677
17.16.2 Master operation in multimaster system	678
17.16.3 Slave operation	681
17.17 Timing of Data Communication	684
CHAPTER 18 DMA FUNCTION (DMA CONTROLLER)	691
18.1 Features	691
18.2 Configuration	692
18.3 Registers	694
18.4 Transfer Sources and Destinations	702
18.5 Transfer Modes	702
18.6 Transfer Types	703
18.7 DMA Channel Priorities	704
18.8 Time Related to DMA Transfer	705
18.9 DMA Transfer Start Factors	706
18.10 End of DMA Transfer	707
18.11 Cautions	707
CHAPTER 19 INTERRUPT SERVICING/EXCEPTION PROCESSING FUNCTION.....	712
19.1 Features	712
19.2 Non-Maskable Interrupts	716
19.2.1 Operation	718
19.2.2 Restoration.....	719

19.2.3 NP flag	720
19.3 Maskable Interrupts	721
19.3.1 Operation	721
19.3.2 Restoration.....	723
19.3.3 Priorities of maskable interrupts	724
19.3.4 Interrupt control register (xxICn)	728
19.3.5 Interrupt mask registers 0 to 3 (IMR0 to IMR3)	730
19.3.6 In-service priority register (ISPR)	732
19.3.7 ID flag	733
19.3.8 Watchdog timer mode register 2 (WDTM2).....	733
19.4 Software Exception.....	734
19.4.1 Operation	734
19.4.2 Restoration.....	735
19.4.3 EP flag	736
19.5 Exception Trap	737
19.5.1 Illegal opcode.....	737
19.6 Multiple Interrupt Servicing Control.....	741
19.7 External Interrupt Request Input Pins (NMI, INTP0 to INTP7)	742
19.7.1 Noise elimination.....	742
19.7.2 Edge detection	742
19.8 Interrupt Response Time of CPU.....	748
19.9 Periods in Which Interrupts Are Not Acknowledged by CPU	749
19.10Cautions	749
19.10.1 Restored PC.....	749
CHAPTER 20 KEY INTERRUPT FUNCTION	750
20.1 Function	750
20.2 Pin Functions.....	751
20.3 Registers	751
20.4 Cautions	752
CHAPTER 21 STANDBY FUNCTION	753
21.1 Overview	753
21.2 Registers	755
21.3 HALT Mode	760
21.3.1 Setting and operation status	760
21.3.2 Releasing HALT mode	760
21.4 IDLE1 Mode.....	762
21.4.1 Setting and operation status	762
21.4.2 Releasing IDLE1 mode	763
21.5 IDLE2 Mode.....	765
21.5.1 Setting and operation status	765
21.5.2 Releasing IDLE2 mode	766
21.5.3 Securing setup time when releasing IDLE2 mode.....	768
21.6 STOP Mode/Low-Voltage STOP Mode	769
21.6.1 Setting and operation status	769
21.6.2 Releasing STOP mode/low-voltage STOP mode.....	773
21.6.3 Re-setting after release of low-voltage STOP mode	774
21.6.4 Securing oscillation stabilization time when releasing STOP mode	775
21.7 Subclock Operation Mode/Low-Voltage Subclock Operation Mode	776

21.7.1	Setting and operation status	776
21.7.2	Releasing subclock operation mode	780
21.7.3	Releasing low-voltage subclock operation mode	780
21.8	Sub-IDLE Mode/Low-Voltage Sub-IDLE Mode	781
21.8.1	Setting and operation status	781
21.8.2	Releasing sub-IDLE mode/low-voltage sub-IDLE mode	784
CHAPTER 22	RESET FUNCTION.....	785
22.1	Overview	785
22.2	Configuration.....	786
22.3	Register to Check Reset Source	787
22.4	Operation	788
22.4.1	Reset operation via $\overline{\text{RESET}}$ pin	788
22.4.2	Reset operation by watchdog timer 2.....	791
22.4.3	Reset operation by low-voltage detector.....	793
22.4.4	Operation immediately after reset ends	794
22.4.5	Reset function operation	796
22.5	Cautions	797
CHAPTER 23	CLOCK MONITOR	798
23.1	Functions	798
23.2	Configuration.....	798
23.3	Registers	799
23.4	Operation	800
CHAPTER 24	LOW-VOLTAGE DETECTOR (LVI).....	803
24.1	Functions	803
24.2	Configuration.....	803
24.3	Registers	804
24.4	Operation	806
24.4.1	To use for internal reset signal.....	806
24.4.2	To use for interrupt.....	807
CHAPTER 25	CRC FUNCTION.....	808
25.1	Functions	808
25.2	Configuration.....	808
25.3	Registers	809
25.4	Operation	810
25.5	Usage.....	811
CHAPTER 26	REGULATOR	813
26.1	Outline	813
26.2	Operation	815
CHAPTER 27	OPTION BYTE.....	816
27.1	Program Example.....	817
CHAPTER 28	FLASH MEMORY	818
28.1	Features	818
28.2	Memory Configuration.....	819

28.3	Functional Outline	821
28.4	Rewriting by Dedicated Flash Memory Programmer	824
28.4.1	Programming environment.....	824
28.4.2	Communication mode	825
28.4.3	Flash memory control	827
28.4.4	Selection of communication mode	828
28.4.5	Communication commands.....	829
28.4.6	Pin connection in on-board programming	830
28.5	Rewriting by Self Programming	834
28.5.1	Overview.....	834
28.5.2	Features.....	835
28.5.3	Standard self programming flow	836
28.5.4	Flash functions.....	837
28.5.5	Pin processing	837
28.5.6	Internal resources used.....	838
CHAPTER 29	ON-CHIP DEBUG FUNCTION	839
29.1	Debugging with DCU	841
29.1.1	Connection circuit example.....	841
29.1.2	Interface signals.....	842
29.1.3	Mask function.....	843
29.1.4	Registers.....	844
29.1.5	Operation	845
29.1.6	Cautions.....	846
29.2	Debugging Without Using DCU	847
29.2.1	Circuit connection examples	847
29.2.2	Mask function.....	850
29.2.3	Allocation of user resources.....	851
29.2.4	Cautions.....	858
29.3	ROM Security Function	859
29.3.1	Security ID	859
29.3.2	Setting.....	860
CHAPTER 30	ELECTRICAL SPECIFICATIONS (V850ES/JC3-L (40-pin)) (Target)	861
30.1	Absolute Maximum Ratings	861
30.2	Capacitance	862
30.3	Operating Conditions	863
30.4	Oscillator Characteristics	864
30.4.1	Main clock oscillator characteristics	864
30.4.2	Subclock oscillator characteristics	865
30.4.3	PLL characteristics.....	866
30.4.4	Internal oscillator characteristics	866
30.5	Regulator Characteristics	866
30.6	DC Characteristics	867
30.6.1	Pin characteristics.....	867
30.6.2	Supply current characteristics	869
30.6.3	Data retention characteristics (in STOP mode).....	870
30.7	AC Characteristics	871
30.7.1	Measurement conditions	871
30.7.2	Power on/power off/reset timing.....	872

30.8 Peripheral Function Characteristics	873
30.8.1 Interrupt timing	873
30.8.2 Key return timing.....	873
30.8.3 Timer timing	873
30.8.4 UART timing.....	874
30.8.5 CSIB timing.....	874
30.8.6 I ² C bus mode.....	876
30.8.7 A/D converter	877
30.8.8 LVI circuit characteristics	878
30.9 Flash Memory Programming Characteristics	879
CHAPTER 31 ELECTRICAL SPECIFICATIONS (V850ES/JC3-L (48-pin)) (Target)	881
31.1 Absolute Maximum Ratings	881
31.2 Capacitance	882
31.3 Operating Conditions	883
31.4 Oscillator Characteristics.....	884
31.4.1 Main clock oscillator characteristics	884
31.4.2 Subclock oscillator characteristics	885
31.4.3 PLL characteristics.....	886
31.4.4 Internal oscillator characteristics	886
31.5 Regulator Characteristics	886
31.6 DC Characteristics	887
31.6.1 Pin characteristics.....	887
31.6.2 Supply current characteristics	889
31.6.3 Data retention characteristics (in STOP mode).....	890
31.7 AC Characteristics	891
31.7.1 Measurement conditions	891
31.7.2 Power on/power off/reset timing.....	892
31.8 Peripheral Function Characteristics	893
31.8.1 Interrupt timing	893
31.8.2 Key return timing.....	893
31.8.3 Timer timing	893
31.8.4 UART timing.....	894
31.8.5 CSIB timing.....	894
31.8.6 I ² C bus mode.....	896
31.8.7 A/D converter	897
31.8.8 D/A converter	898
31.8.9 LVI circuit characteristics	898
31.9 Flash Memory Programming Characteristics	899
CHAPTER 32 ELECTRICAL SPECIFICATIONS (V850ES/JE3-L) (Target).....	901
32.1 Absolute Maximum Ratings	901
32.2 Capacitance	902
32.3 Operating Conditions	903
32.4 Oscillator Characteristics.....	904
32.4.1 Main clock oscillator characteristics	904
32.4.2 Subclock oscillator characteristics	905
32.4.3 PLL characteristics.....	906
32.4.4 Internal oscillator characteristics	906
32.5 Regulator Characteristics	906

32.6 DC Characteristics	907
32.6.1 Pin characteristics	907
32.6.2 Supply current characteristics	909
32.6.3 Data retention characteristics (in STOP mode)	910
32.7 AC Characteristics	911
32.7.1 Measurement conditions	911
32.7.2 Power on/power off/reset timing	912
32.8 Peripheral Function Characteristics	913
32.8.1 Interrupt timing	913
32.8.2 Key return timing	913
32.8.3 Timer timing	913
32.8.4 UART timing	914
32.8.5 CSIB timing	914
32.8.6 I ² C bus mode	916
32.8.7 A/D converter	917
32.8.8 D/A converter	918
32.8.9 LVI circuit characteristics	918
32.9 Flash Memory Programming Characteristics	919
 APPENDIX A DEVELOPMENT TOOLS	 925
A.1 Software Package	927
A.2 Language Processing Software	927
A.3 Control Software	927
A.4 Debugging Tools (Hardware)	928
A.4.1 When using IECUBE QB-V850ESJX3L, QB-V850ESSX2	928
A.4.2 When using MINICUBE QB-V850MINIL	931
A.4.3 When using MINICUBE2 QB-MINI2	932
A.5 Debugging Tools (Software)	932
A.6 Embedded Software	933
A.7 Flash Memory Writing Tools	933
 APPENDIX B MAJOR DIFFERENCES BETWEEN PRODUCTS	 934
 APPENDIX C REGISTER INDEX	 935
 APPENDIX D INSTRUCTION SET LIST	 945
D.1 Conventions	945
D.2 Instruction Set (in Alphabetical Order)	948

CHAPTER 1 INTRODUCTION

The V850ES/JC3-L and V850ES/JE3-L are one of the products in the Renesas Electronics V850 single-chip microcontroller series designed for low-power operation for real-time control applications.

1.1 General

The V850ES/JC3-L and V850ES/JE3-L are 32-bit single-chip microcontrollers that include the V850ES CPU core and peripheral functions such as ROM/RAM, timer/counters, serial interfaces, an A/D converter, a D/A converter.

In addition to high real-time response characteristics and 1-clock-pitch basic instructions, the The V850ES/JC3-L and V850ES/JE3-L feature multiply instructions, saturated operation instructions, bit manipulation instructions, etc., realized by a hardware multiplier, as optimum instructions for digital servo control applications. Moreover, as a real-time control system, the V850ES/JC3-L and V850ES/JE3-L enable an extremely high cost-performance for applications that require super low power consumption, such as PC peripheral device, ECR peripheral device, and industrial instrument.

Table 1-1. V850ES/JC3-L Product List (1/2)

Generic Name		V850ES/JC3-L				
Part Number		μ PD70F3797	μ PD70F3798	μ PD70F3799	μ PD70F3800	μ PD70F3838
Internal memory	Flash memory	16 KB	32 KB	64 KB	128 KB	256 KB
	RAM	8 KB	8 KB	8 KB	8 KB	16 KB
Memory space		64 MB				
General-purpose register		32 bits \times 32 registers				
Clock	Main clock (oscillation frequency)	Ceramic/crystal (in PLL mode: $f_x = 2.5$ to 5 MHz (multiplied by 4), in clock through mode: $f_x = 2.5$ to 10 MHz) External clock (in PLL mode: $f_x = 2.5$ to 5 MHz (multiplied by 4), in clock through mode: $f_x = 2.5$ to 5 MHz)				
	Subclock (oscillation frequency)	Crystal ($f_{XT} = 32.768$ kHz)				
	Internal oscillator	$f_R = 220$ kHz (TYP.)				
	Minimum instruction execution time	50 ns (main clock (f_{xx}) = 20 MHz)				
I/O port		I/O: 27 (5 V tolerant/N-ch open-drain output selectable: 17)				
Timer	16-bit TMP	6 channels				
	16-bit TMQ	1 channel				
	16-bit TMM	1 channel				
	Watch timer	1 channel				
	RTC	1 channel				
	WDT	1 channel				
Real-time output port		4 bits \times 1 channel, 2 bits \times 1 channel, or 6 bits \times 1 channel				
10-bit A/D converter		5 channels				
8-bit D/A converter		-				
Serial interface	CSIB	1 channel				
	UARTA/CSIB	-				
	CSIB/I ² C bus	1 channel				
	UARTA/I ² C bus	1 channel				
	UARTA	1 channel				
DMA controller		4 channels (transfer target: on-chip peripheral I/O, internal RAM)				
Interrupt source	External	6 (6) ^{Note}				
	Internal	42				
Power save function		HALT/IDLE1/IDLE2/STOP/subclock/sub-IDLE/ low-voltage STOP/low-voltage subclock/low-voltage sub-IDLE mode				
Reset source		$\overline{\text{RESET}}$ pin input, watchdog timer 2 (WDT2), clock monitor (CLM), low-voltage detector (LVI)				
CRC function		16-bit error detection code generated for 8-bit unit data				
On-chip debug		MINICUBE [®] , MINICUBE2 supported				
Operating power supply voltage		2.2 to 3.6 V @ 5 MHz, 2.7 to 3.6 V @ 20 MHz				
Operating ambient temperature		-40 to +85°C				
Package		40-pin WQFN (6 \times 6 mm)				

Note The figure in parentheses indicates the number of external interrupts that can release STOP mode.

Table 1-1. V850ES/JC3-L Product List (2/2)

Generic Name		V850ES/JC3-L				
Part Number		μ PD70F3801	μ PD70F3802	μ PD70F3803	μ PD70F3804	μ PD70F3839
Internal memory	Flash memory	16 KB	32 KB	64 KB	128 KB	256 KB
	RAM	8 KB	8 KB	8 KB	8 KB	16 KB
Memory space		64 MB				
General-purpose register		32 bits \times 32 registers				
Clock	Main clock (oscillation frequency)	Ceramic/crystal (in PLL mode: $f_x = 2.5$ to 5 MHz (multiplied by 4), in clock through mode: $f_x = 2.5$ to 10 MHz) External clock (in PLL mode: $f_x = 2.5$ to 5 MHz (multiplied by 4), in clock through mode: $f_x = 2.5$ to 5 MHz)				
	Subclock (oscillation frequency)	Crystal ($f_{XT} = 32.768$ kHz)				
	Internal oscillator	$f_R = 220$ kHz (TYP.)				
	Minimum instruction execution time	50 ns (main clock (f_{xx}) = 20 MHz)				
I/O port		I/O: 34 (5 V tolerant/N-ch open-drain output selectable: 20)				
Timer	16-bit TMP	6 channels				
	16-bit TMQ	1 channel				
	16-bit TMM	1 channel				
	Watch timer	1 channel				
	RTC	1 channel				
	WDT	1 channel				
Real-time output port		4 bits \times 1 channel, 2 bits \times 1 channel, or 6 bits \times 1 channel				
10-bit A/D converter		6 channels				
8-bit D/A converter		1 channel				
Serial interface	CSIB	2 channels				
	UARTA/CSIB	1 channel				
	CSIB/I ² C bus	1 channel				
	UARTA/I ² C bus	2 channels				
	UARTA	-				
DMA controller		4 channels (transfer target: on-chip peripheral I/O, internal RAM)				
Interrupt source	External	6 (6) ^{Note}				
	Internal	46				
Power save function		HALT/IDLE1/IDLE2/STOP/subclock/sub-IDLE/ low-voltage STOP/low-voltage subclock/low-voltage sub-IDLE mode				
Reset source		$\overline{\text{RESET}}$ pin input, watchdog timer 2 (WDT2), clock monitor (CLM), low-voltage detector (LVI)				
CRC function		16-bit error detection code generated for 8-bit unit data				
On-chip debug		MINICUBE, MINICUBE2 supported				
Operating power supply voltage		2.2 to 3.6 V @5 MHz, 2.7 to 3.6 V @20 MHz				
Operating ambient temperature		-40 to +85°C				
Package		48-pin LQFP (7 \times 7 mm), 48-pin WQFN (7 \times 7 mm)				

Note The figure in parentheses indicates the number of external interrupts that can release STOP mode.

Table 1-1. V850ES/JE3-L Product List

Generic Name		V850ES/JE3-L				
Part Number		μ PD70F3805	μ PD70F3806	μ PD70F3807	μ PD70F3808	μ PD70F3840
Internal memory	Flash memory	16 KB	32 KB	64 KB	128 KB	256 KB
	RAM	8 KB	8 KB	8 KB	8 KB	16 KB
Memory space		64 MB				
General-purpose register		32 bits \times 32 registers				
Clock	Main clock (oscillation frequency)	Ceramic/crystal (in PLL mode: $f_x = 2.5$ to 5 MHz (multiplied by 4), in clock through mode: $f_x = 2.5$ to 10 MHz) External clock (in PLL mode: $f_x = 2.5$ to 5 MHz (multiplied by 4), in clock through mode: $f_x = 2.5$ to 5 MHz)				
	Subclock (oscillation frequency)	Crystal ($f_{XT} = 32.768$ kHz)				
	Internal oscillator	$f_R = 220$ kHz (TYP.)				
	Minimum instruction execution time	50 ns (main clock (f_{xx}) = 20 MHz)				
I/O port		I/O: 50 (5 V tolerant/N-ch open-drain output selectable: 28)				
Timer	16-bit TMP	6 channels				
	16-bit TMQ	1 channel				
	16-bit TMM	1 channel				
	Watch timer	1 channel				
	RTC	1 channel				
	WDT	1 channel				
Real-time output port		4 bits \times 1 channel, 2 bits \times 1 channel, or 6 bits \times 1 channel				
10-bit A/D converter		10 channels				
8-bit D/A converter		1 channel				
Serial interface	CSIB	3 channels				
	UARTA/CSIB	1 channel				
	CSIB/I ² C bus	1 channel				
	UARTA/I ² C bus	2 channels				
	UARTA	-				
DMA controller		4 channels (transfer target: on-chip peripheral I/O, internal RAM)				
Interrupt source	External	9 (9) ^{Note}				
	Internal	48				
Power save function		HALT/IDLE1/IDLE2/STOP/subclock/sub-IDLE/ low-voltage STOP/low-voltage subclock/low-voltage sub-IDLE mode				
Reset source		$\overline{\text{RESET}}$ pin input, watchdog timer 2 (WDT2), clock monitor (CLM), low-voltage detector (LVI)				
CRC function		16-bit error detection code generated for 8-bit unit data				
On-chip debug		MINICUBE, MINICUBE2 supported				
Operating power supply voltage		2.2 to 3.6 V @5 MHz, 2.7 to 3.6 V @20 MHz				
Operating ambient temperature		-40 to +85°C				
Package		64-pin LQFP (10 \times 10 mm)				

Note The figure in parentheses indicates the number of external interrupts that can release STOP mode.

1.2 Features

- Minimum instruction execution time: 50 ns (operating on main clock (f_{XX}) of 20 MHz: $V_{DD} = 2.7$ to 3.6 V)
200 ns (operating on main clock (f_{XX}) of 5 MHz: $V_{DD} = 2.2$ to 3.6 V)
30.5 μ s (operating on subclock (f_{XT}) of 32.768 kHz)
- General-purpose registers: 32 bits \times 32 registers
- CPU features:
 - Signed multiplication ($16 \times 16 \rightarrow 32$): 1 to 2 clocks
 - Signed multiplication ($32 \times 32 \rightarrow 64$): 1 to 5 clocks
 - Saturated operations (overflow and underflow detection functions included)
 - Most instructions can be executed in 1 clock cycle by using 32-bit RISC-based 5-stage pipeline architecture
 - Instruction fetching from internal ROM and accessing internal RAM for data can be executed separately, by using Harvard architecture
 - High code efficiency achieved by using variable length instructions
 - 32-bit shift instruction: 1 clock cycle
 - Bit manipulation instructions
 - Load/store instructions with long/short format
- Memory space: 64 MB of linear address space (for programs and data)
 - Internal memory:
 - RAM: 8/16 KB (see **Tables 1-1** and **1-2**)
 - Flash memory: 16/32/64/128/256 KB (see **Tables 1-1** and **1-2**)
- Interrupts and exceptions:

		Internal			external:		
		maskable	Non-maskable	total	maskable	Non-maskable	total
V850ES/JC3-L (40-pin)	μ PD70F3797	1	42	43	1	5	6
	μ PD70F3798						
	μ PD70F3799						
	μ PD70F3800						
	μ PD70F3838						
V850ES/JC3-L (48-pin)	μ PD70F3801	1	46	47	1	5	6
	μ PD70F3802						
	μ PD70F3803						
	μ PD70F3804						
	μ PD70F3839						
V850ES/JE3-L	μ PD70F3805	1	48	49	1	8	9
	μ PD70F3806						
	μ PD70F3807						
	μ PD70F3808						
	μ PD70F3840						

Software exceptions: 32 sources

Exception trap: 2 sources

- Ports: I/O ports: 27/34/50 (see **Tables 1-1** and **1-2**)

- Timer function:
 - 16-bit interval timer M (TMM): 1 channel
 - 16-bit timer/event counter P (TMP): 6 channels^{Note}
 - 16-bit timer/event counter Q (TMQ): 1 channel
 - Watch timer: 1 channel
 - Watchdog timer: 1 channel

Note The available functions differ for each product. For details, see **CHAPTER 6 16-BIT TIMER/EVENT COUNTER P (TMP)**.

- Real-time counter: 1 channel
- Real-time output port: 6 bits × 1 channel
- Serial interface: For details about the number of incorporated channels, see **Tables 1-1** and **1-2**.
 - Asynchronous serial interface A (UARTA)
 - Clocked serial interface B (CSIB)
 - I²C bus interface (I²C)
- A/D converter: 10-bit resolution: 5/6/10 channels (see **Tables 1-1** and **1-2**)
- D/A converter: 8-bit resolution: 0/1 channels (see **Tables 1-1** and **1-2**)
- DMA controller: 4 channels
- DCU (debug control unit): JTAG interface
- Clock generator:
 - During main clock or subclock operation
 - 7-level CPU clock (f_{xx}, f_{xx}/2, f_{xx}/4, f_{xx}/8, f_{xx}/16, f_{xx}/32, f_{xt})
 - Clock-through mode/PLL mode selectable
- Internal oscillator clock: 220 kHz (TYP.)
- Power-save functions:
 - HALT/IDLE1/IDLE2/STOP/low-voltage STOP/subclock/sub-IDLE/low-voltage subclock/low-voltage sub-IDLE mode
- Package:
 - V850ES/JC3-L
 - 40-pin plastic WQFN (6 × 6)
 - 48-pin plastic LQFP (fine pitch) (7 × 7)
 - 48-pin plastic WQFN (7 × 7)
 - V850ES/JE3-L
 - 64-pin plastic LQFP (fine pitch) (8 × 8)
- Power supply voltage:
 - V_{DD} = 2.2 V to 3.6 V (5 MHz)
 - V_{DD} = 2.7 V to 3.6 V (20 MHz)

1.3 Application Fields

Digital cameras, electrical power meters, mobile terminals, digital home electronics, other consumer devices

1.4 Ordering Information

1.4.1 V850ES/JC3-L

Part Number	Package	Internal Flash Memory
μ PD70F3797K8-4B4-AX	40-pin plastic WQFN (6 × 6)	16 KB
μ PD70F3798K8-4B4-AX	40-pin plastic WQFN (6 × 6)	32 KB
μ PD70F3799K8-4B4-AX	40-pin plastic WQFN (6 × 6)	64 KB
μ PD70F3800K8-4B4-AX	40-pin plastic WQFN (6 × 6)	128 KB
μ PD70F3838K8-4B4-AX	40-pin plastic WQFN (6 × 6)	256 KB
μ PD70F3801GA-GAM-AX	48-pin plastic LQFP (fine pitch) (7 × 7)	16 KB
μ PD70F3802GA-GAM-AX	48-pin plastic LQFP (fine pitch) (7 × 7)	32 KB
μ PD70F3803GA-GAM-AX	48-pin plastic LQFP (fine pitch) (7 × 7)	64 KB
μ PD70F3804GA-GAM-AX	48-pin plastic LQFP (fine pitch) (7 × 7)	128 KB
μ PD70F3839GA-GAM-AX	48-pin plastic LQFP (fine pitch) (7 × 7)	256 KB
μ PD70F3801K8-5B4-AX	48-pin plastic WQFN (7 × 7)	16 KB
μ PD70F3802K8-5B4-AX	48-pin plastic WQFN (7 × 7)	32 KB
μ PD70F3803K8-5B4-AX	48-pin plastic WQFN (7 × 7)	64 KB
μ PD70F3804K8-5B4-AX	48-pin plastic WQFN (7 × 7)	128 KB
μ PD70F3839K8-5B4-AX	48-pin plastic WQFN (7 × 7)	256 KB

Remark The V850ES/JC3-L is a lead-free product.

1.4.2 V850ES/JE3-L

Part Number	Package	Internal Flash Memory
μ PD70F3805GB-GAH-AX	64-pin plastic LQFP (fine pitch) (10 × 10)	16 KB
μ PD70F3806GB-GAH-AX	64-pin plastic LQFP (fine pitch) (10 × 10)	32 KB
μ PD70F3807GB-GAH-AX	64-pin plastic LQFP (fine pitch) (10 × 10)	64 KB
μ PD70F3808GB-GAH-AX	64-pin plastic LQFP (fine pitch) (10 × 10)	128 KB
μ PD70F3840GB-GAH-AX	64-pin plastic LQFP (fine pitch) (10 × 10)	256 KB

Remark The V850ES/JE3-L is a lead-free product.

1.5 Pin Configuration (Top View)

• V850ES/JC3-L

40-pin plastic WQFN (6 × 6)

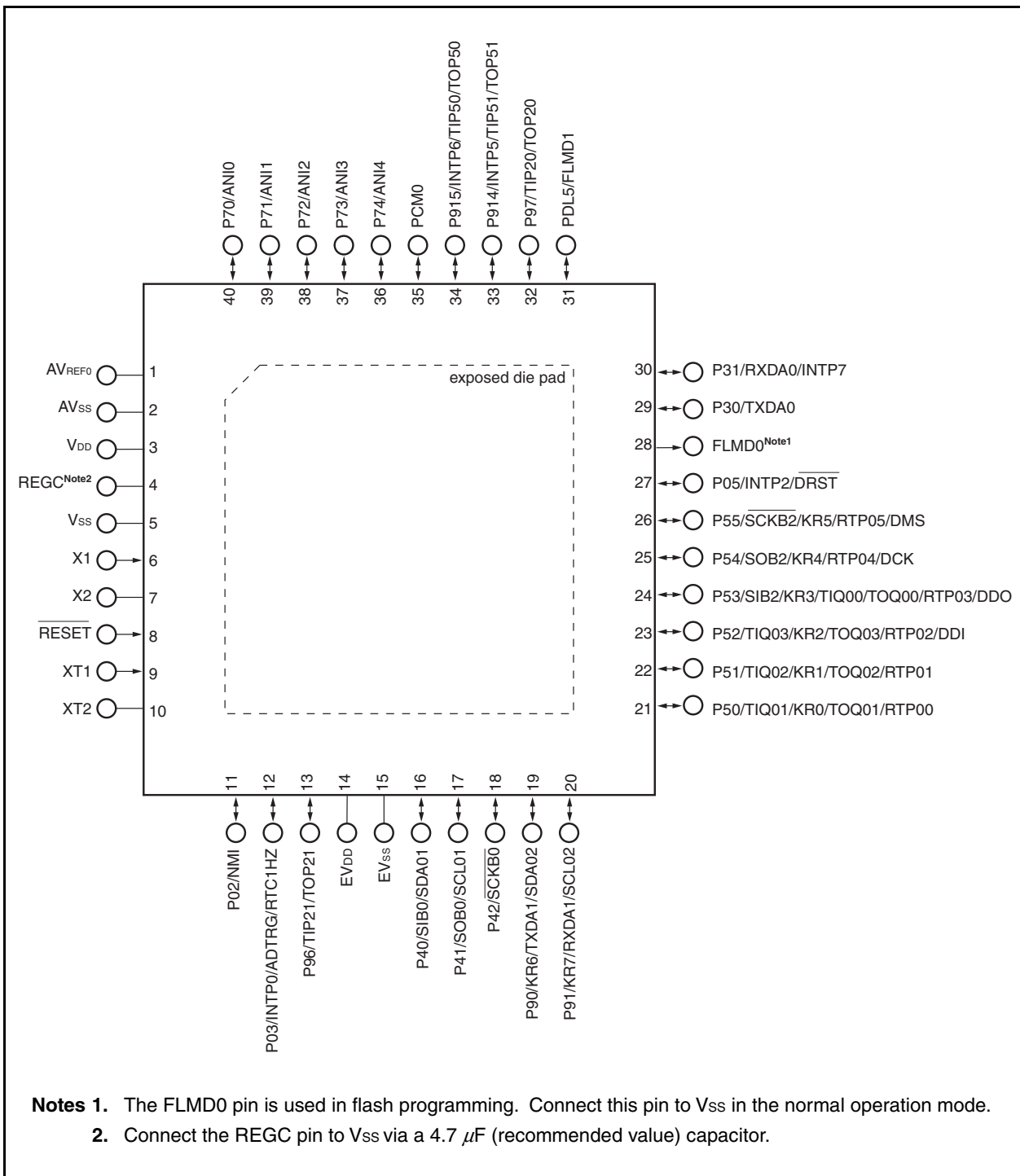
μ PD70F3797K8-4B4-AX

μ PD70F3798K8-4B4-AX

μ PD70F3799K8-4B4-AX

μ PD70F3800K8-4B4-AX

μ PD70F3838K8-4B4-AX



• V850ES/JC3-L

48-pin plastic WQFN (7 × 7)

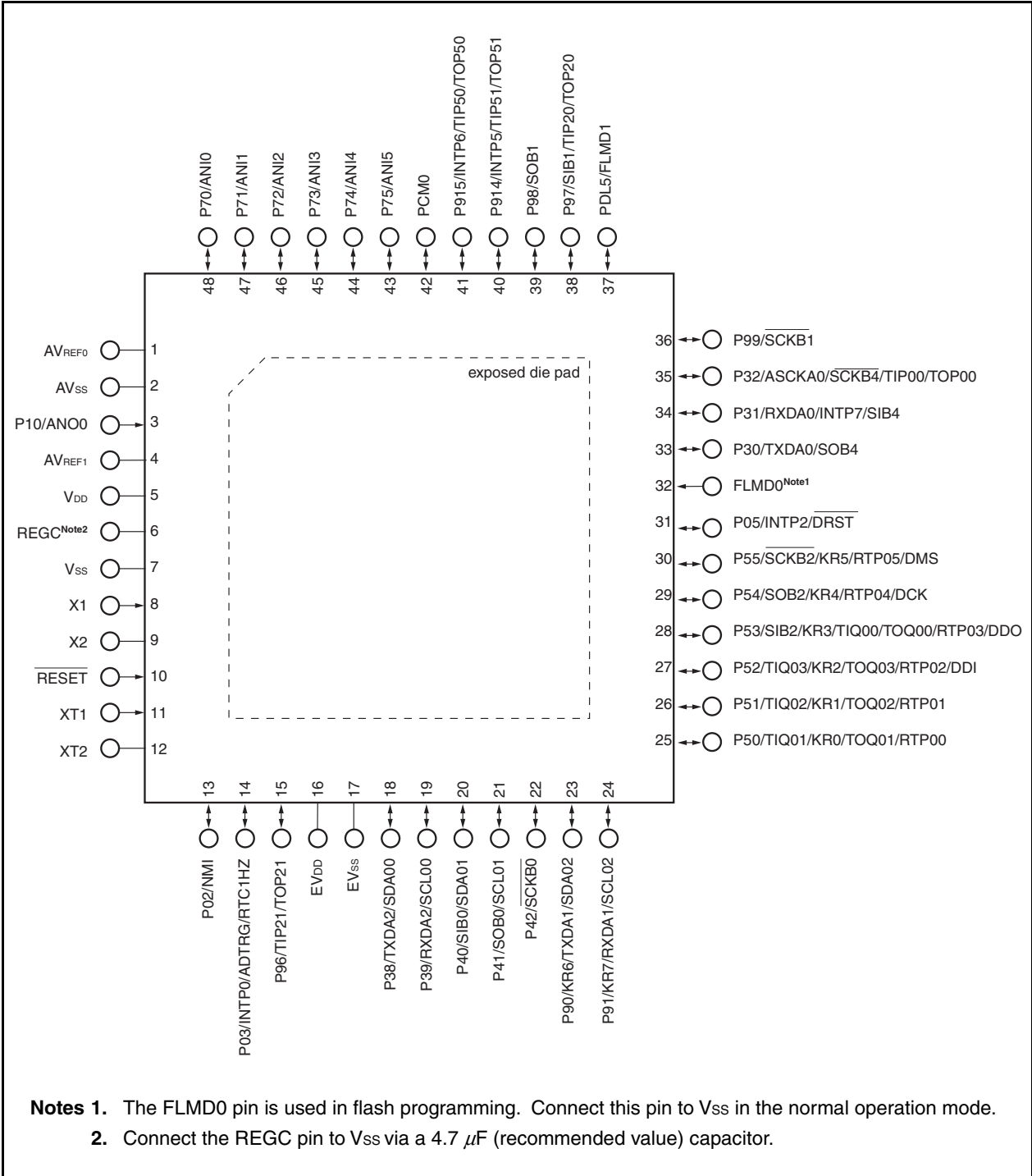
μ PD70F3801K8-5B4-AX

μ PD70F3802K8-5B4-AX

μ PD70F3803K8-5B4-AX

μ PD70F3804K8-5B4-AX

μ PD70F3839K8-5B4-AX



• V850ES/JC3-L

48-pin plastic LQFP (fine pitch) (7 × 7)

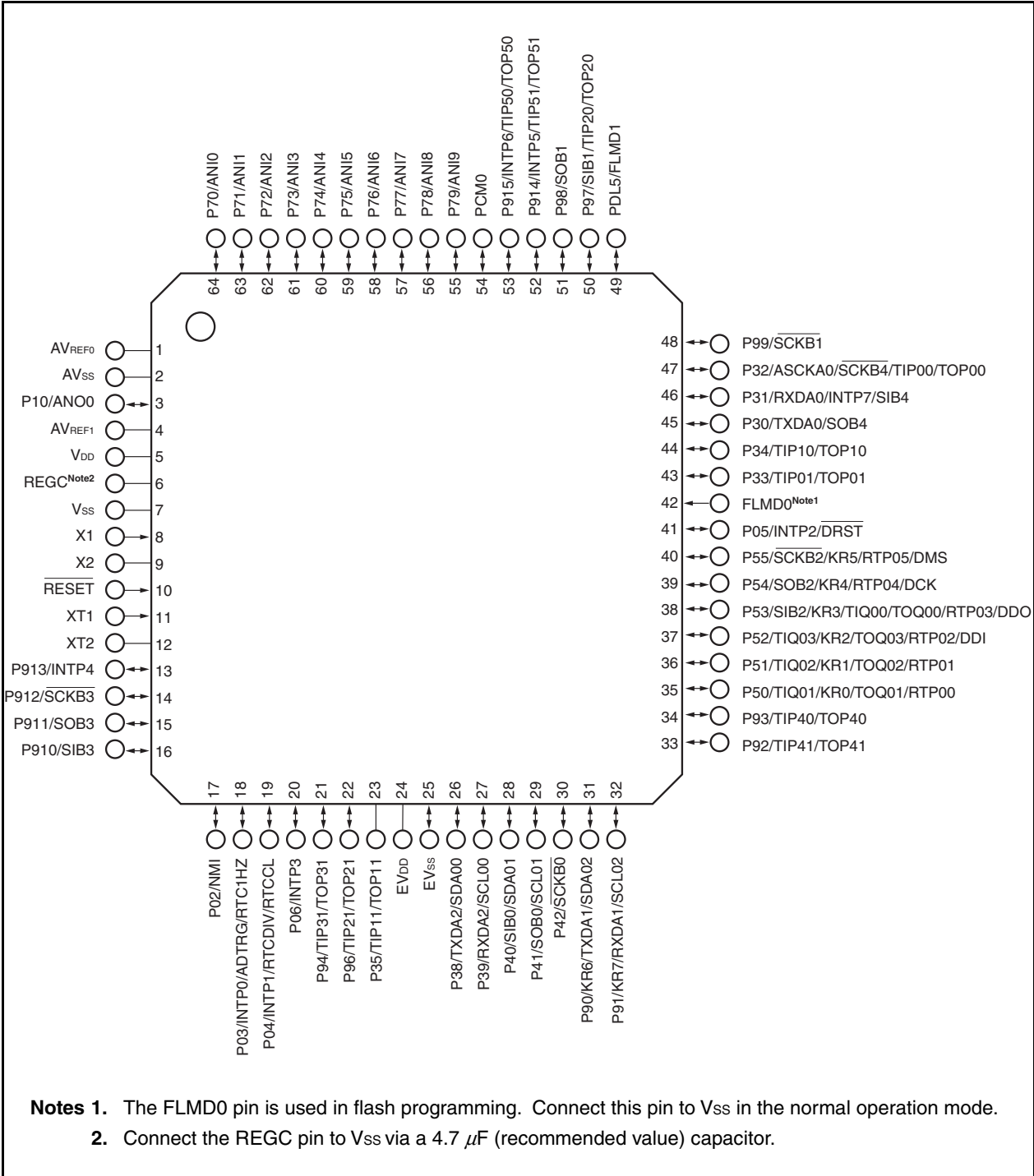
μ PD70F3801GA-GAM-AX

μ PD70F3802GA-GAM-AX

μ PD70F3803GA-GAM-AX

μ PD70F3804GA-GAM-AX

μ PD70F3839GA-GAM-AX



Notes 1. The FLMD0 pin is used in flash programming. Connect this pin to Vss in the normal operation mode.

2. Connect the REGC pin to Vss via a 4.7 μF (recommended value) capacitor.

• V850ES/JE3-L

64-pin plastic LQFP (fine pitch) (10 × 10)

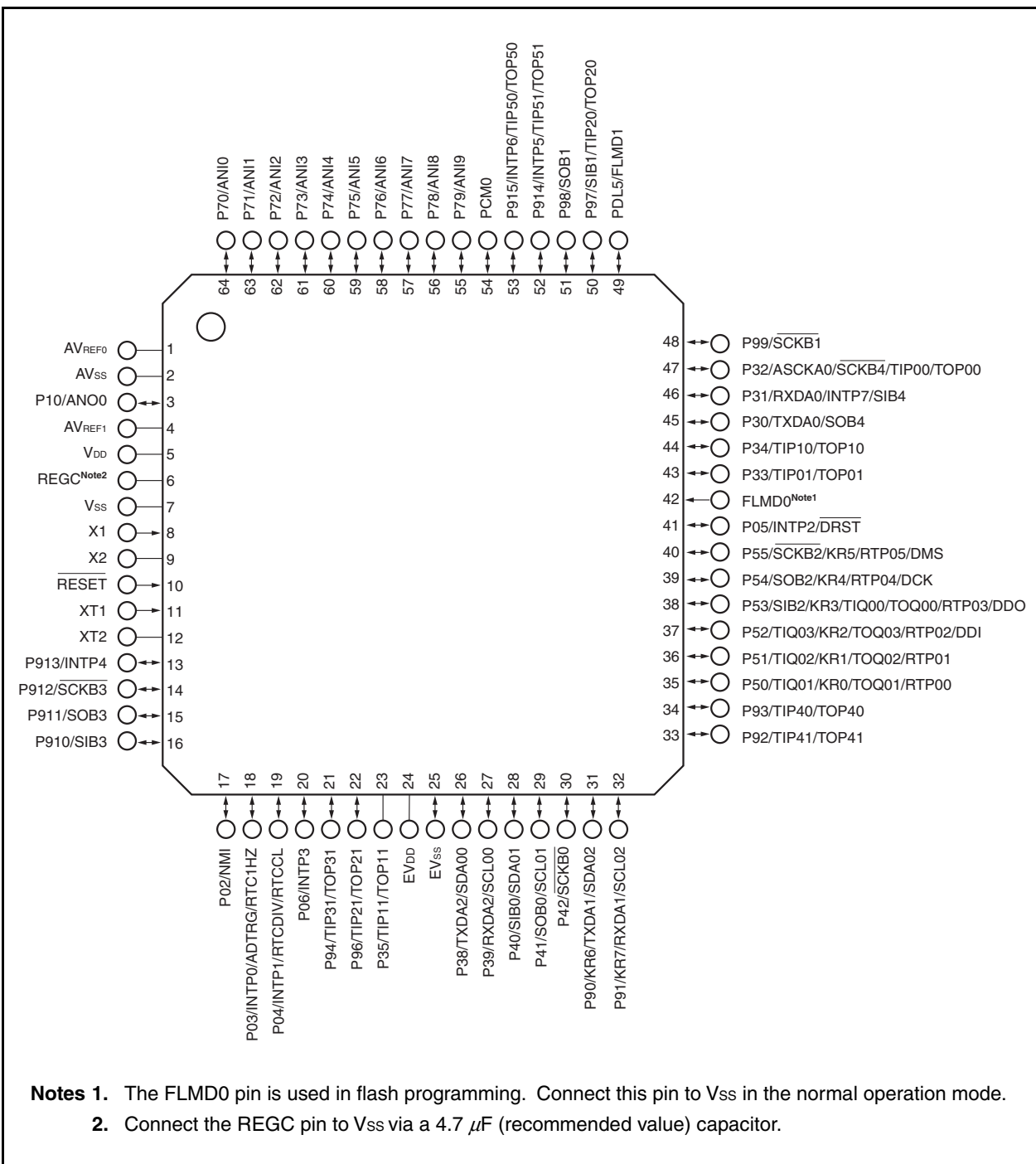
μ PD70F3805GB-GAH-AX

μ PD70F3806GB-GAH-AX

μ PD70F3807GB-GAH-AX

μ PD70F3808GB-GAH-AX

μ PD70F3840GB-GAH-AX



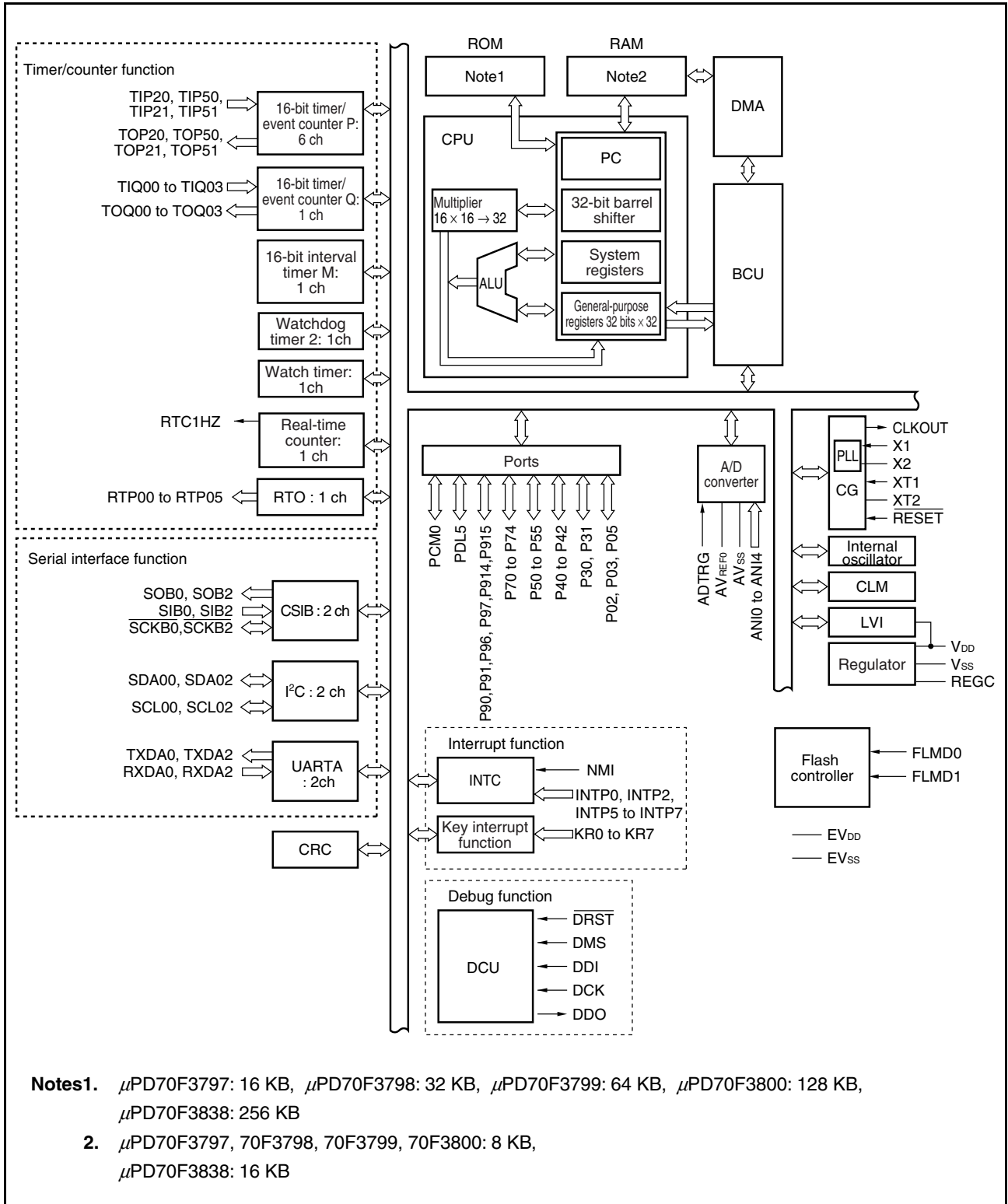
Pin functions

ADTRG:	A/D trigger input	$\overline{\text{RESET}}$:	Reset
ANI0 to ANI9:	Analog input	RTC1HZ,	Real-time Counter Clock Output
ANO0:	Analog output	RTCCL, RTCDIV	
ASCKA0:	Asynchronous serial clock	RTP00 to RTP05:	Real-time output port
AV _{REF0} , AV _{REF1} :	Analog reference voltage	RXDA0 to RXDA2:	Receive data
AV _{SS} :	Analog V _{SS}	SCKB0 to SCKB4:	Serial clock
DCK:	Debug clock	SCL00 to SCL02:	Serial clock
DDI:	Debug data input	SDA00 to SDA02:	Serial data
DDO:	Debug data output	SIB0 to SIB4:	Serial input
DMS:	Debug mode select	SOB0 to SOB4:	Serial output
$\overline{\text{DRST}}$:	Debug reset	TIP00, TIP01,	Timer input
EV _{DD} :	Power supply for external pin	TIP10, TIP11,	
EV _{SS} :	Ground for external pin	TIP20, TIP21,	
FLMD0, FLMD1:	Flash programming mode	TIP31,	
INTP0 to INTP7:	External interrupt input	TIP40, TIP41,	
KR0 to KR7:	Key return	TIP50, TIP51,	
NMI:	Non-maskable interrupt request	TIQ00 to TIQ03:	
P02 to P06:	Port 0	TOP00, TOP01,	Timer output
P10:	Port 1	TOP10, TOP11,	
P30 to P35,		TOP20, TOP21,	
P38, P39:	Port 3	TOP31,	
P40 to P42:	Port 4	TOP40, TOP41,	
P50 to P55:	Port 5	TOP50, TOP51,	
P70 to P79:	Port 7	TOQ00 to TOQ03:	
P90 to P94,		TXDA0 to TXDA2:	Transmit data
P96 to P915:	Port 9	V _{DD} :	Power supply
PCM0:	Port CM	V _{SS} :	Ground
PDL15:	Port DL	X1, X2:	Crystal for main clock
REGC:	Regulator control	XT1, XT2:	Crystal for subclock

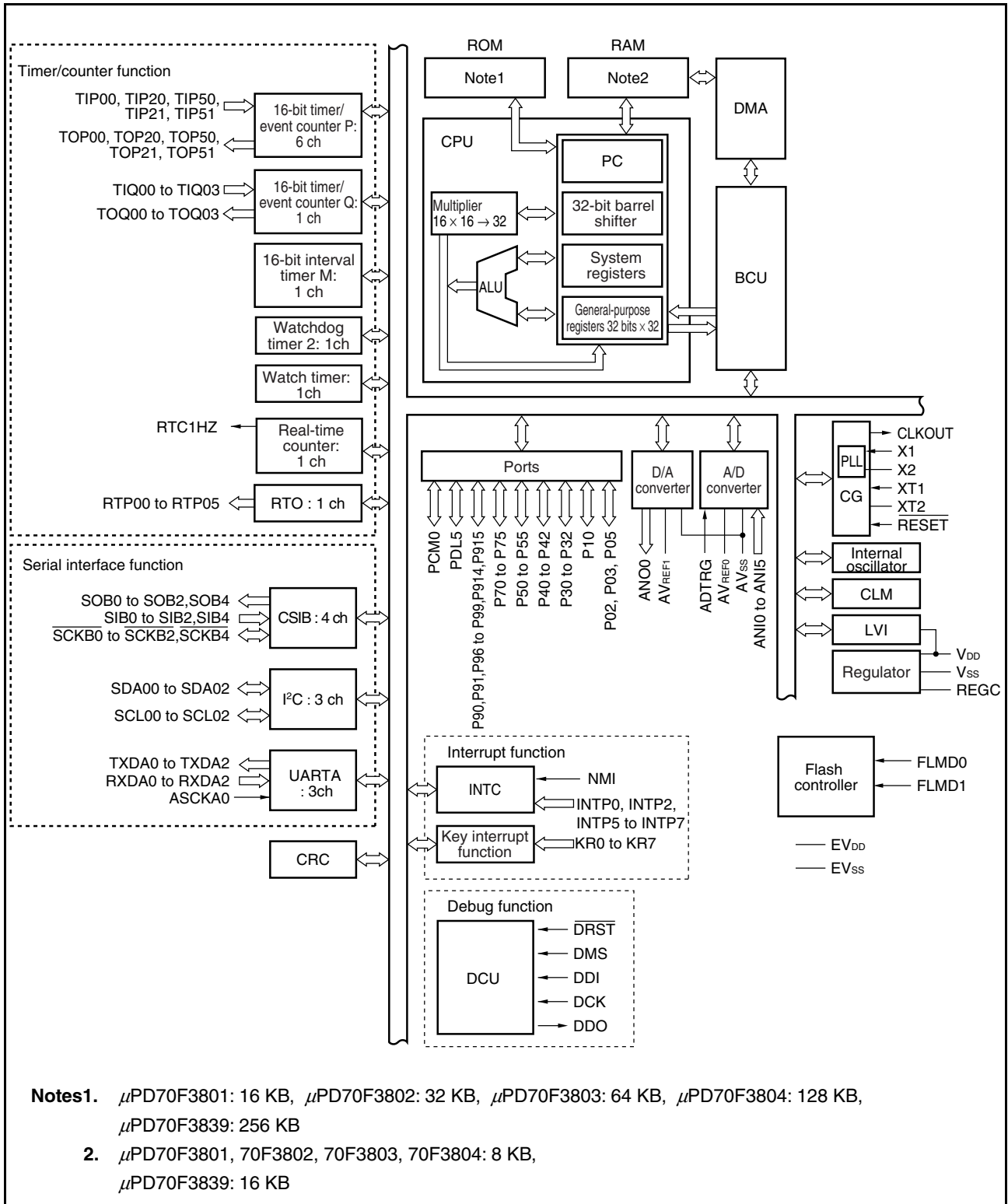
1.6 Function Block Configuration

1.6.1 Internal block diagram

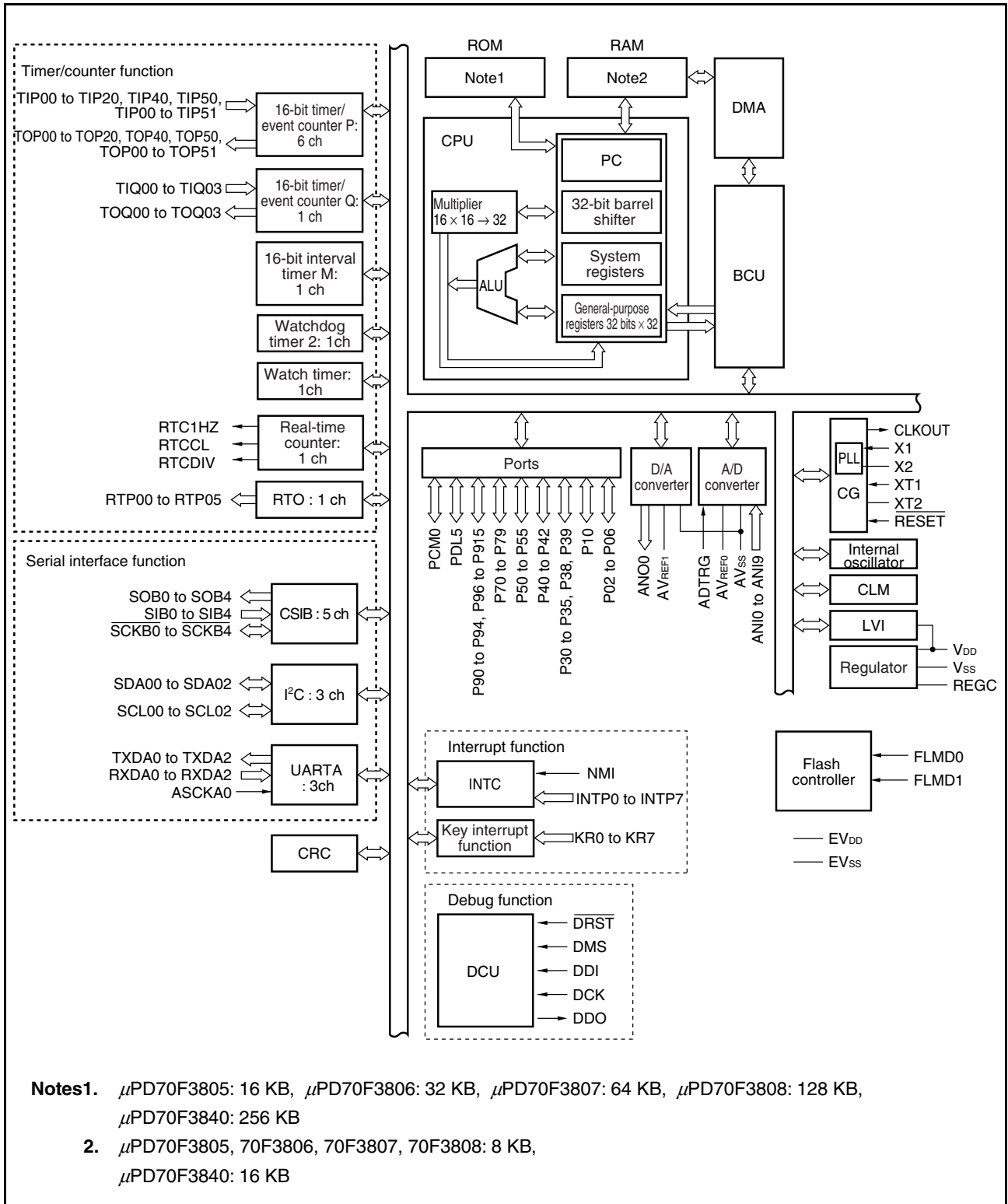
(1) V850ES/JC3-L (40-pin)



(2) V850ES/JC3-L (48-pin)



(3) V850ES/JE3-L



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 \times 16 bits \rightarrow 32 bits) and a barrel shifter (32 bits) contribute to faster complex processing.

(2) Bus control unit (BCU)

The BCU controls the internal bus.

(3) Flash memory (ROM)

This is a 256/128/64/32/16 KB flash memory mapped to addresses 0000000H to 003FFFFH/0000000H to 001FFFFH/0000000H to 000FFFFH/0000000H-0007FFFH/0000000H-0003FFFH.

It can be accessed from the CPU in one clock during instruction fetch.

(4) RAM

This is a 16/8 KB RAM mapped to addresses 3FFB000H to 3FFEFFFH/3FFD000H to 3FFEFFFH. It can be accessed from the CPU in one clock during data access.

(5) Interrupt controller (INTC)

This controller handles hardware interrupt requests (NMI, INTP0 to INTP7) from on-chip peripheral hardware and external hardware. Eight levels of interrupt priorities can be specified for these interrupt requests, and multiplexed interrupt servicing control can be performed.

(6) Clock generator (CG)

A main clock oscillator and subclock oscillator are provided and generate the main clock oscillation frequency (f_x) and subclock frequency (f_{XT}), respectively. There are two modes: In the clock-through mode, f_x is used as the main clock frequency (f_{xx}) as is. In the PLL mode, f_x is used multiplied by 4.

The CPU clock frequency (f_{CPU}) can be selected from among f_{xx} , $f_{xx}/2$, $f_{xx}/4$, $f_{xx}/8$, $f_{xx}/16$, $f_{xx}/32$, and f_{XT} .

(7) Internal oscillator

An internal oscillator is provided on chip. The oscillation frequency is 220 kHz (TYP). The internal oscillator supplies the clock for watchdog timer 2 and timer M.

(8) Timer/counter

Six-channel 16-bit timer/event counter P (TMP), one-channel 16-bit timer/event counter Q (TMQ), and one-channel 16-bit interval timer M (TMM), are provided on chip.

(9) Watch timer

This timer counts the reference time period (0.5 s) for counting the clock (the 32.768 kHz subclock or the 32.768 kHz f_{BRG} clock from the prescaler). The watch timer can also be used as an interval timer based on the main clock.

(10) Real-time counter (for watch)

The real-time counter counts the reference time (one second) for watch counting based on the subclock (32.768 kHz) or main clock. This can simultaneously be used as the interval timer based on the main clock. Hardware counters dedicated to year, month, day of week, day, hour, minute, and second are provided, and can count up to 99 years.

(11) Watchdog timer 2

A watchdog timer is provided on chip to detect inadvertent program loops, system abnormalities, etc.

The internal oscillator clock, the main clock, or the subclock can be selected as the source clock.

Watchdog timer 2 generates a non-maskable interrupt request signal (INTWDT2) or a system reset signal (WDT2RES) after an overflow occurs.

(12) Serial interface

The V850ES/JC3-L, V850ES/JE3-L include three kinds of serial interfaces: asynchronous serial interface A (UARTA), 3-wire variable-length serial interface B (CSIB), and an I²C bus interface (I²C).

(13) A/D converter

This 10-bit A/D converter includes 10/6/5 analog input pins. Conversion is performed using the successive approximation method.

(14) D/A converter

A 1/0-channel, 8-bit-resolution D/A converter that uses the R-2R ladder method is provided on chip.

(15) DMA controller

A 4-channel DMA controller is provided on chip. This controller transfers data between the internal RAM and on-chip peripheral I/O devices in response to interrupt requests sent by on-chip peripheral I/O.

(16) Key interrupts function

A key interrupt request signal (INTKR) can be generated by inputting a falling edge to the key input pins (8 channels).

(17) Real-time output function

The real-time output function transfers preset 6-bit data to output latches upon the occurrence of a timer compare register match signal.

(18) CRC function

A CRC operation circuit that generates a 16-bit CRC (Cyclic Redundancy Check) code upon the setting of 8-bit data is provided on-chip.

(19) DCU (debug control unit)

An on-chip debug function that uses the JTAG (Joint Test Action Group) communication specifications is provided. Switching between the normal port function and on-chip debugging function is done with the control pin input level and the OCDM register.

(20) Ports

The following general-purpose port functions and control pin functions are available.

Table 1-2. Port Functions (V850ES/JC3-L (40-pin))

Port	I/O	Alternate Function
P0	3-bit I/O	NMI, external interrupt, A/D converter trigger, debug reset, real-time counter output
P3	2-bit I/O	External interrupt, serial interface
P4	3-bit I/O	Serial interface
P5	6-bit I/O	Timer I/O, real-time output, key interrupt input, serial interface, debug I/O
P7	5-bit I/O	A/D converter analog input
P9	6-bit I/O	Serial interface, key interrupt input, timer I/O, external interrupt
PCM	1-bit I/O	–
PDL	1-bit I/O	–

Table 1-3. Port Functions (V850ES/JC3-L (48-pin))

Port	I/O	Alternate Function
P0	3-bit I/O	NMI, external interrupt, A/D converter trigger, debug reset, real-time counter output
P1	1-bit I/O	D/A converter analog output
P3	5-bit I/O	External interrupt, serial interface, timer I/O
P4	3-bit I/O	Serial interface
P5	6-bit I/O	Timer I/O, real-time output, key interrupt input, serial interface, debug I/O
P7	6-bit I/O	A/D converter analog input
P9	8-bit I/O	Serial interface, key interrupt input, timer I/O, external interrupt
PCM	1-bit I/O	–
PDL	1-bit I/O	–

Table 1-4. Port Functions (V850ES/JE3-L)

Port	I/O	Alternate Function
P0	5-bit I/O	NMI, external interrupt, A/D converter trigger, debug reset, real-time counter output
P1	1-bit I/O	D/A converter analog output
P3	8-bit I/O	External interrupt, serial interface, timer I/O
P4	3-bit I/O	Serial interface
P5	6-bit I/O	Timer I/O, real-time output, key interrupt input, serial interface, debug I/O
P7	10-bit I/O	A/D converter analog input
P9	15-bit I/O	Serial interface, key interrupt input, timer I/O, external interrupt
PCM	1-bit I/O	–
PDL	1-bit I/O	–

CHAPTER 2 PIN FUNCTIONS

2.1 List of Pin Functions

The functions of the pins in the V850ES/JC3-L, V850ES/JE3-L are described below.

There are 2 or 3 types of pin I/O buffer power supplies: AV_{REF0} , AV_{REF1} , and EV_{DD} . The relationship between these power supplies and the pins is described below.

Table 2-1. Pin I/O Buffer Power Supplies (V850ES/JC3-L : 40-pin products)

Power Supply	Corresponding Pins
AV_{REF0}	Port 7
EV_{DD}	\overline{RESET} , ports 0, 3 to 5, 9, CM, DL, FLMD0

Table 2-2. Pin I/O Buffer Power Supplies (V850ES/JC3-L : 48-pin products)

Power Supply	Corresponding Pins
AV_{REF0}	Port 7
AV_{REF1}	Port 1
EV_{DD}	\overline{RESET} , ports 0, 3 to 5, 9, CM, DL, FLMD0

Table 2-3. Pin I/O Buffer Power Supplies (V850ES/JE3-L : 64-pin products)

Power Supply	Corresponding Pins
AV_{REF0}	Port 7
AV_{REF1}	Port 1
EV_{DD}	\overline{RESET} , ports 0, 3 to 5, 9, CM, DL, FLMD0

(1) Port functions

(1/2)

Function	Pin No.			I/O	Description	Alternate Function
	JC3L (40)	JC3L (48)	JE3L			
P02	11	13	17	I/O	Port 0 (refer to 4.3.1) 5-bit I/O port Input/output can be specified in 1-bit units. N-ch open-drain output can be specified in 1-bit units. 5 V tolerant.	NMI
P03	12	14	18			INTP0/ADTRG/RTC1HZ
P04	-	-	19			INTP1/RTCDIV/RTCCCL
P05 ^{Note}	27	31	41			INTP2/ $\overline{\text{DRST}}$
P06	-	-	20			INTP3
P10	-	3	3	I/O	Port 1 (refer to 4.3.2) 1-bit I/O port (V850ES/JC3-L: 48-pin, V850ES/JE3-L only). Input/output can be specified in 1-bit units.	ANO0
P30	-	33	45	I/O	Port 3 (refer to 4.3.3) 2-bit I/O port (V850ES/JC3-L : 40-pin) 5-bit I/O port (V850ES/JC3-L : 48-pin) 8-bit I/O port (V850ES/JE3-L) Input/output can be specified in 1-bit units. N-ch open-drain output can be specified in 1-bit units. 5 V tolerant.	TXDA0/SOB4
	29	-	-			TXDA0
P31	-	34	46			RXDA0/INTP7/SIB4
	30	-	-			RXDA0/INTP7
P32	-	35	47			ASCKA0/ $\overline{\text{SCKB4}}$ /TIP00/TOP00
P33	-	-	43			TIP01/TOP01
P34	-	-	44			TIP10/TOP10
P35	-	-	23			TIP11/TOP11
P38	-	18	26			TXDA2/SDA00
P39	-	19	27			RXDA2/SCL00
P40	16	20	28			I/O
P41	17	21	29	SOB0/SCL01		
P42	18	22	30	$\overline{\text{SCKB0}}$		
P50	21	25	35	I/O	Port 5 (refer to 4.3.5) 6-bit I/O port Input/output can be specified in 1-bit units. N-ch open-drain output can be specified in 1-bit units. 5 V tolerant.	TIQ01/KR0/TOQ01/RTP00
P51	22	26	36			TIQ02/KR1/TOQ02/RTP01
P52	23	27	37			TIQ03/KR2/TOQ03/RTP02/DDI
P53	24	28	38			SIB2/KR3/TIQ00/TOQ00/RTP03/DDO
P54	25	29	39			SOB2/KR4/RTP04/DCK
P55	26	30	40			$\overline{\text{SCKB2}}$ /KR5/RTP05/DMS

Note Incorporates a pull-down resistor. It can be disconnected by clearing the OCDM.OCDM0 bit to 0.

Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

(2/2)

Function	Pin No.			I/O	Description	Alternate Function
	JC3L (40)	JC3L (48)	JE3L			
P70	40	48	64	I/O	Port 7 (refer to 4.3.6) 5-bit I/O port (V850ES/JC3-L : 40-pin) 6-bit I/O port (V850ES/JC3-L : 48-pin) 10-bit I/O port (V850ES/JE3-L) Input/output can be specified in 1-bit units.	ANI0
P71	39	47	63			ANI1
P72	38	46	62			ANI2
P73	37	45	61			ANI3
P74	36	44	60			ANI4
P75	–	43	59			ANI5
P76	–	–	58			ANI6
P77	–	–	57			ANI7
P78	–	–	56			ANI8
P79	–	–	55			ANI9
P90	19	23	31	I/O	Port 9 (refer to 4.3.7) 6-bit I/O port (V850ES/JC3-L : 40-pin) 8-bit I/O port (V850ES/JC3-L : 48-pin) 15-bit I/O port (V850ES/JE3-L) Input/output can be specified in 1-bit units. N-ch open-drain output can be specified in 1-bit units. 5 V tolerant.(P90 to P96)	KR6/TXDA1/SDA02
P91	20	24	32			KR7/RXDA1/SCL02
P92	–	–	33			TIP41/TOP41/TXDA4
P93	–	–	34			TIP40/TOP40/RXDA4
P94	–	–	21			TIP31/TOP31/TXDA5
P96	13	15	22			TXDC0/TIP21/TOP21
P97	–	38	50			SIB1/TIP20/TOP20
	32	–	–			TIP20/TOP20
P98	–	39	51			SOB1
P99	–	36	48			SCKB1
P910	–	–	16			SIB3
P911	–	–	15			SOB3
P912	–	–	14			SCKB3
P913	–	–	13			INTP4
P914	33	40	52			INTP5/TIP51/TOP51
P915	34	41	53	INTP6/TIP50/TOP50		
PCM0	35	42	54	I/O	Port CM (refer to 4.3.8) 1-bit I/O port Input/output can be specified in 1-bit units.	–
PDL5	31	37	49	I/O	Port DL (refer to 4.3.9) 1-bit I/O port Input/output can be specified in 1-bit units.	FLMD1

Remark JC3L (40): V850ES/JC3-L (40-pin products)

JC3L (48): V850ES/JC3-L (48-pin products)

JE3L: V850ES/JE3-L

(2) Non-port functions

(1/5)

Function	Pin No.			I/O	Description	Alternate Function
	JC3L (40)	JC3L (48)	JE3L			
ADTRG	12	14	18	Input	A/D converter external trigger input. 5 V tolerant.	P03/INTP0/RTC1HZ
ANI0	40	48	64	Input	Analog voltage input for A/D converter	P70
ANI1	39	47	63			P71
ANI2	38	46	62			P72
ANI3	37	45	61			P73
ANI4	36	44	60			P74
ANI5	–	43	59			P75
ANI6	–	–	58			P76
ANI7	–	–	57			P77
ANI8	–	–	56			P78
ANI9	–	–	55			P79
ANO0	–	3	3	Output	Analog voltage output for D/A converter	P10
ASCKA0	–	35	47	Input	UARTA0 baud rate clock input. 5 V tolerant.	P32/SCKB4/TIP00/TOP00
AV _{REF0}	1	1	1	–	Reference voltage input for A/D converter/positive power supply for port 7	–
AV _{REF1}	–	4	4		Reference voltage input for D/A converter/positive power supply for port 1	–
AV _{SS}	2	2	2	–	Ground potential for A/D and D/A converters (same potential as V _{SS})	–
DCK	25	29	39	Input	Debug clock input. 5 V tolerant.	P54/SOB2/KR4/RTP04
DDI	23	27	37	Input	Debug data input. 5 V tolerant.	P52/TIQ03/KR2/TOQ03/RTP02
DDO ^{Note}	24	28	38	Output	Debug data output. N-ch open-drain output selectable. 5 V tolerant.	P53/SIB2/KR3/TIQ00/TOQ00/ RTP03
DMS	26	30	40	Input	Debug mode select input. 5 V tolerant.	P55/SCKB2/KR5/RTP05
DRST	27	31	41	Input	Debug reset input. 5 V tolerant.	P05/INTP2
EV _{DD}	14	16	24	–	Positive power supply for external (same potential as V _{DD})	–
EV _{SS}	15	17	25	–	Ground potential for external (same potential as V _{SS})	–
FLMD0	28	32	42	Input	Flash memory programming mode setting pin	–
FLMD1	31	37	49			PDL5

Note In the on-chip debug mode, high-level output is forcibly set.

Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

(2/5)

Function	Pin No.			I/O	Description	Alternate Function
	JC3L (40)	JC3L (48)	JE3L			
INTP0	12	14	18	Input	External interrupt request input (maskable, analog noise elimination). Analog noise elimination or digital noise elimination selectable for INTP3 pin. 5 V tolerant.	P03/ADTRG/RTC1HZ
INTP1	–	–	19			P04/RTCDIV/RTCCL
INTP2	27	31	41			P05/ $\overline{\text{DRST}}$
INTP3	–	–	20			P06
INTP4	–	–	13			P913
INTP5	33	40	52			P914/TIP51/TOP51
INTP6	34	41	53			P915/TIP50/TOP50
INTP7	–	34	46			P31/RXDA0/SIB4
	30	–	–	P31/RXDA0		
KR0 ^{Note 1}	21	25	35	Input	Key interrupt input (on-chip analog noise eliminator). 5 V tolerant.	P50/TIQ01/TOQ01/RTP00
KR1 ^{Note 1}	22	26	36			P51/TIQ02/TOQ02/RTP01
KR2 ^{Note 1}	23	27	37			P52/TIQ03/TOQ03/RTP02/DDI
KR3 ^{Note 1}	24	28	38			P53/SIB2/TIQ00/TOQ00/RTP03/DDO
KR4 ^{Note 1}	25	29	39			P54/SOB2/RTP04/DCK
KR5 ^{Note 1}	26	30	40			P55/ $\overline{\text{SCKB2}}$ /RTP05/DMS
KR6 ^{Note 1}	19	23	31			P90/TXDA1/SDA02
KR7 ^{Note 1}	20	24	32			P91/RXDA1/SCL02
NMI ^{Note 2}	11	13	17	Input	External interrupt input (non-maskable, analog noise elimination). 5 V tolerant.	P02
REGC	4	6	6	–	Connection of regulator output stabilization capacitance (4.7 μF (recommended value))	–
$\overline{\text{RESET}}$	8	10	10	Input	System reset input	–
RTC1HZ	12	14	18	Output	Real-time counter correction clock (1 Hz) output	P03/INTP0/ADTRG
RTCCL	–	–	19	Output	Real-time counter clock (32 kHz primary oscillation) output	P04/INTP1/RTCDIV
RTCDIV	–	–	19	Output	Real-time counter clock (32 kHz division) output	P04/INTP1/RTCCL
RTP00	21	25	35	Output	Real-time output port. N-ch open-drain output selectable. 5 V tolerant.	P50/TIQ01/KR0/TOQ01
RTP01	22	26	36			P51/TIQ02/KR1/TOQ02
RTP02	23	27	37			P52/TIQ03/KR2/TOQ03/DDI
RTP03	24	28	38			P53/SIB2/KR3/TIQ00/TOQ00/DDO
RTP04	25	29	39			P54/SOB2/KR4/DCK
RTP05	26	30	40			P55/ $\overline{\text{SCKB2}}$ /KR5/DMS

Notes1. Connect a pull-up resistor externally.

- The NMI pin alternately functions as the P02 pin. It functions as the P02 pin after reset. To enable the NMI function, set the PMC0.PMC02 bit to 1. The initial setting of the NMI pin is "No edge detected". Select the NMI pin valid edge using the INTF0 and INTR0 registers.

Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

(3/5)

Function	Pin No.			I/O	Description	Alternate Function
	JC3L (40)	JC3L (48)	JE3L			
RXDA0	–	34	46	Input	Serial receive data input (UARTA0 to UARTA2) 5 V tolerant.	P31/INTP7/SIB4
	30	–	–			P31/INTP7
RXDA1	20	24	32			P91/KR7/SCL02
RXDA2	–	19	27			P39/SCL00
SCKB0	18	22	30	I/O	Serial clock I/O (CSIB0 to CSIB4) N-ch open-drain output selectable. 5 V tolerant.	P42
SCKB1	–	36	48			P99
SCKB2	26	30	40			P55/KR5/RTP05/DMS
SCKB3	–	–	14			P912
SCKB4	–	35	47			P32/ASCKA0/TIP00/TOP00
SCL00	–	19	27	I/O	Serial clock I/O (I ² C00 to I ² C02) N-ch open-drain output selectable. 5 V tolerant.	P39/RXDA2
SCL01	17	21	29			P41/SOB0
SCL02	20	24	32			P91/KR7/RXDA1
SDA00	–	18	26	I/O	Serial transmit/receive data I/O (I ² C00 to I ² C02) N-ch open-drain output selectable. 5 V tolerant.	P38/TXDA2
SDA01	16	20	28			P40/SIB0
SDA02	19	23	31			P90/KR6/TXDA1
SIB0	16	20	28	Input	Serial receive data input (CSIB0 to CSIB4) 5 V tolerant.	P40/SDA01
SIB1	–	38	50			P97/TIP20/TOP20
SIB2	24	28	38			P53/KR3/TIQ00/TOQ00/RTP03/DDO
SIB3	–	–	16			P910
SIB4	–	34	46			P31/RXDA0/INTP7
SOB0	17	21	29	Output	Serial transmit data output (CSIB0 to CSIB4) N-ch open-drain output selectable. 5 V tolerant.	P41/SCL01
SOB1	–	39	51			P98
SOB2	25	29	39			P54/KR4/RTP04/DCK
SOB3	–	–	15			P911
SOB4	–	33	45			P30/TXDA0

Remark JC3L (40): V850ES/JC3-L (40-pin products)

JC3L (48): V850ES/JC3-L (48-pin products)

JE3L: V850ES/JE3-L

(4/5)

Function	Pin No.			I/O	Description	Alternate Function
	JC3L (40)	JC3L (48)	JE3L			
TIP00	–	35	47	Input	External event count input/capture trigger input/external trigger input (TMP0). 5 V tolerant.	P32/ASCKA0/SCKB4/TOP00
TIP01	–	–	43		Capture trigger input (TMP0). 5 V tolerant.	P33/TOP01
TIP10	–	–	44		External event count input/capture trigger input/external trigger input (TMP1). 5 V tolerant.	P34/TOP10
TIP11	–	–	43		Capture trigger input (TMP1). 5 V tolerant.	P35/TOP11
TIP20	–	38	50		External event count input/capture trigger input/external trigger input (TMP2). 5 V tolerant.	P97/SIB1/TOP20
	32	–	–			P97/TOP20
TIP21	13	15	22		Capture trigger input (TMP2). 5 V tolerant.	P96/A6/TOP21
TIP31	–	–	21		Capture trigger input (TMP3). 5 V tolerant.	P94/A4/TOP31
TIP40	–	–	34		External event count input/capture trigger input/external trigger input (TMP4). 5 V tolerant.	P93/A3/TOP40
TIP41	–	–	33		Capture trigger input (TMP4). 5 V tolerant.	P92/A2/TOP41
TIP50	34	41	53		External event count input/capture trigger input/external trigger input (TMP5). 5 V tolerant.	P915/A15/INTP6/TOP50
TIP51	33	40	52		Capture trigger input (TMP5). 5 V tolerant.	P914/A14/INTP5/TOP51
TIQ00	24	28	38	Input	External event count input/capture trigger input/external trigger input (TMQ0). 5 V tolerant.	P53/SIB2/KR3/TOQ00/RTP03 /DDO
TIQ01	21	28	38		Capture trigger input (TMQ0). 5 V tolerant.	P50/KR0/TOQ01/RTP00
TIQ02	22	26	36			P51/KR1/TOQ02/RTP01
TIQ03	23	27	37			P52/KR2/TOQ03/RTP02/DDI
TOP00	–	35	47	Output	Timer output (TMP0) N-ch open-drain output selectable. 5 V tolerant.	P32/ASCKA0/SCKB4/TIP00
TOP01	–	–	43			P33/TIP01
TOP10	–	–	44		Timer output (TMP1) N-ch open-drain output selectable. 5 V tolerant.	P34/TIP10
TOP11	–	–	23			P35/TIP11
TOP20	–	38	50		Timer output (TMP2) N-ch open-drain output selectable. 5 V tolerant.	P97/SIB1/TIP20
	32	–	–			P97/TIP20
TOP21	13	15	22			P96/TIP21
TOP31	–	–	21		Timer output (TMP3) N-ch open-drain output selectable. 5 V tolerant.	P94/TIP31
TOP40	–	–	34		Timer output (TMP4) N-ch open-drain output selectable. 5 V tolerant.	P93/TIP40
TOP41	–	–	33			P92/TIP41
TOP50	34	41	53		Timer output (TMP5) N-ch open-drain output selectable. 5 V tolerant.	P915/INTP6/TIP50
TOP51	33	40	52			P914/INTP5/TIP51

Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

(5/5)

Function	Pin No.			I/O	Description	Alternate Function
	JC3L (40)	JC3L (48)	JE3L			
TOQ00	24	28	38	Output	Timer output (TMQ0) N-ch open-drain output selectable. 5 V tolerant.	P53/SIB2/KR3/TIQ00/RTP03/DDO
TOQ01	21	25	35			P50/TIQ01/KR0/RTP00
TOQ02	22	26	36			P51/TIQ02/KR1/RTP01
TOQ03	23	27	37			P52/TIQ03/KR2/RTP02/DDI
TXDA0	–	33	45	Output	Serial transmit data output (UARTA0 to UARTA5) N-ch open-drain output selectable. 5 V tolerant.	P30/SOB4
	29	–	–			P30
TXDA1	19	23	31			P90/KR6/SDA02
TXDA2	–	18	26			P38/SDA00
V _{DD}	3	5	5	–	Positive power supply pin for internal circuits	–
V _{SS}	5	7	7	–	Ground potential for internal circuits	–
X1	6	8	8	Input	Connection of resonator for main clock	–
X2	7	9	9	–		–
XT1	9	11	11	Input	Connection of resonator for subclock	–
XT2	10	12	12	–		–

Remark JC3L (40): V850ES/JC3-L (40-pin products)

JC3L (48): V850ES/JC3-L (48-pin products)

JE3L: V850ES/JE3-L

2.2 Pin States

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

Table 2-2. Pin Operation States in Various Modes

Pin Name	When Power Is Turned On ^{Note 1}	During Reset (Except When Power Is Turned On)	HALT Mode ^{Note 2}	IDLE1, IDLE2, Sub-IDLE Mode ^{Note 2}	STOP Mode ^{Note 2}
P05/DRST	Pulled down	Pulled down ^{Note 3}	Held	Held	Held
P10/ANO0	Undefined	Hi-Z	Held	Held	Held
P53/DDO		Hi-Z ^{Note 4}	Held	Held	Held
Other port pins	Hi-Z	Hi-Z	Held	Held	Held

- Notes 1.** Duration until 1 ms elapses after the supply voltage reaches the operating supply voltage range (lower limit) when the power is turned on.
- 2.** Operates while an alternate function is operating. (For details about the operating status of alternate functions, see **CHAPTER 21 STANDBY FUNCTION**.)
- 3** Pulled down during external reset. During internal reset by the watchdog timer, clock monitor, etc., the state of this pin differs according to the OCDM.OCDM0 bit setting.
- 4** DDO output is specified in the on-chip debug mode.

Remark Hi-Z: High impedance
Held: The state during the level of pins is held.

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

(1/2)

Pin	Alternate Function	Pin No.			I/O Circuit Type	Recommended Connection of Unused Pin
		JC3L (40)	JC3L (48)	JE3L		
P02	NMI	11	13	17	10-D	Input: Independently connect to EV _{DD} or EV _{SS} via a resistor. Output: Leave open.
P03	INTP0/ADTRG/RTC1HZ	12	14	18		
P04	INTP1/RTCDIV/RTCCL	–	–	19		
P05	INTP2/ $\overline{\text{DRST}}$	27	31	41	10-N	Input: Independently connect to EV _{SS} via a resistor. Fixing to V _{DD} level is prohibited. Output: Leave open. Internally pull-down after reset by $\overline{\text{RESET}}$ pin.
P06	INTP3	–	–	20	10-D	Input: Independently connect to EV _{DD} or EV _{SS} via a resistor. Output: Leave open.
P10	ANO0	–	3	3	12-D	Input: Independently connect to AV _{REF1} or AV _{SS} via a resistor. Output: Leave open.
P30	TXDA0/SOB4	–	33	45	10-G	Input: Independently connect to EV _{DD} or EV _{SS} via a resistor. Output: Leave open.
	TXDA0	29	–	–		
P31	RXDA0/INTP7/SIB4	–	34	46	10-D	
	RXDA0/INTP7	30	–	–		
P32	ASCKA0/ $\overline{\text{SCKB4}}$ /TIP00	–	35	47	10-D	
P33	TIP01/TOP01	–	–	43		
P34	TIP10/TOP10	–	–	44		
P35	TIP11/TOP11	–	–	23		
P38	TXDA2/SDA00	–	18	26		
P39	RXDA2/SCL00	–	19	27		
P40	SIB0/SDA01	16	20	28		
P41	SOB0/SCL01	17	21	29		
P42	$\overline{\text{SCKB0}}$	18	22	30		
P50	TIQ01/KR0/TOQ01/RTP00	21	25	35		
P51	TIQ02/KR1/TOQ02/RTP01	22	26	36		
P52	TIQ03/KR2/TOQ03/RTP02/DDI	23	27	37		
P53	SIB2/KR3/TIQ00/TOQ00 /RTP03/DDO	24	28	38		
P54	SOB2/KR4/RTP04/DCK	25	29	39		
P55	$\overline{\text{SCKB2}}$ /KR5/RTP05/DMS	26	30	40		

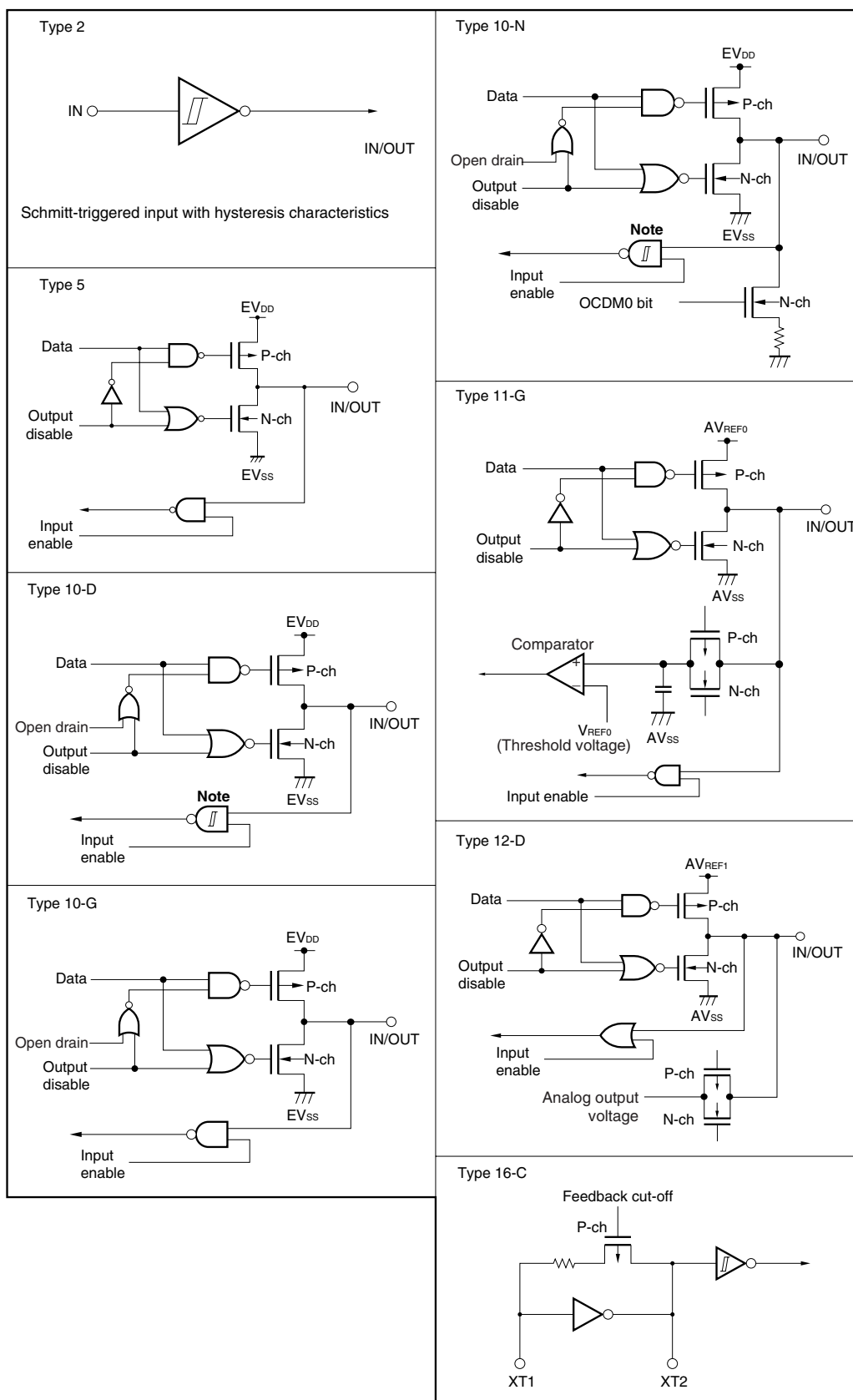
Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

(2/2)

Pin	Alternate Function	Pin No.			I/O Circuit Type	Recommended Connection of Unused Pin	
		JC3L (40)	JC3L (48)	JE3L			
P70 to P74	ANI0 to ANI4	40 to 36	48 to 44	64 to 60	11-G	Input: Independently connect to AV _{REF0} or AV _{SS} via a resistor. Output: Leave open.	
P75	ANI5	–	43	59			
P76 to P79	ANI6 to ANI9	–	–	58 to 55			
P90	KR6/TDXA1/SDA02	19	23	31	10-D	Input: Independently connect to EV _{DD} or EV _{SS} via a resistor. Output: Leave open.	
P91	KR7/RXDA1/SCL02	20	24	32			
P92	TIP41/TOP41/TXDA4	–	–	33			
P93	TIP40/TOP40/RXDA4	–	–	34			
P94	TIP31/TOP31/TXDA5	–	–	21			
P96	TIP21/TOP21	13	15	22			
P97	SIB1/TIP20/TOP20	–	38	50			
	TIP20/TOP20	32	–	–			
P98	SOB1	–	39	51			10-G
P99	SCKB1	–	36	48			10-D
P910	SIB3	–	–	16			
P911	SOB3	–	–	15	10-G		
P912	SCKB3	–	–	14	10-D		
P913	INTP4	–	–	13			
P914	INTP5/TIP51/TOP51	33	40	52			
P915	INTP6/TIP50/TOP50	34	41	53	5		
PCM0	–	35	42	54			
PDL5	FLMD1	31	37	49			
AV _{REF0}	–	1	1	1	–	Directly connect to V _{DD} and always supply power.	
AV _{REF1}	–	–	4	4	–		
AV _{SS}	–	2	2	2	–	Directly connect to V _{SS} and always supply power.	
EV _{DD}	–	14	16	24	–	Directly connect to V _{DD} and always supply power.	
EV _{SS}	–	15	17	25	–	Directly connect to V _{SS} and always supply power.	
FLMD0	–	28	32	42	–	Directly connect to V _{SS} in a mode other than the flash memory programming mode.	
REGC	–	4	6	6	–	Connection of regulator output stabilization capacitance (4.7 μF (recommended value))	
RESET	–	8	10	10	2	–	
V _{DD}	–	3	5	5	–	–	
V _{SS}	–	5	7	7	–	–	
X1	–	6	8	8	–	–	
X2	–	7	9	9	–	–	
XT1	–	9	11	11	16-C	Connect to V _{SS} .	
XT2	–	10	12	12	16-C	Leave open.	

Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

Figure 2-1. Pin I/O Circuits



Note Hysteresis characteristics are not available in port mode.

2.4 Cautions

When the power is turned on, the following pins may output an undefined level temporarily even during reset.

- P10/ANO0 pin
- P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO pin

CHAPTER 3 CPU FUNCTION

The CPU of the V850ES/JC3-L and V850ES/JE3-L are based on RISC architecture and executes almost all instructions in one clock cycle by using a 5-stage pipeline.

3.1 Features

- Variable length instructions (16 bits/32 bits)
- Minimum instruction execution time: 50 ns (operating on main clock (f_{XX}) of 20 MHz: $V_{DD} = 2.7$ to 3.6 V)
200 ns (operating on main clock (f_{XX}) of 5 MHz: $V_{DD} = 2.2$ to 3.6 V)
30.5 μ s (operating on subclock (f_{XT}) of 32.768 kHz)
- Memory space Program space: 64 MB linear
 Data space: 4 GB linear
- General-purpose registers: 32 bits \times 32 registers
- Internal 32-bit architecture
- 5-stage pipeline control
- Multiplication/division instruction
- Saturation operation instruction
- 32-bit shift instruction: 1 clock
- Load/store instruction with long/short format
- Four types of bit manipulation instructions
 - SET1
 - CLR1
 - NOT1
 - TST1

3.2 CPU Register Set

The registers of the V850ES/JC3-L and V850ES/JE3-L can be classified into two types: general-purpose program registers and dedicated system registers. All the registers are 32 bits wide.

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

(1) Program register set		(2) System register set	
31	General-purpose registers	31	0
r0	(Zero register)	EIPC	(Interrupt status saving register)
r1	(Assembler-reserved register)	EIPSW	(Interrupt status saving register)
r2			
r3	(Stack pointer (SP))	FEPC	(NMI status saving register)
r4	(Global pointer (GP))	FEPSW	(NMI status saving register)
r5	(Text pointer (TP))		
r6		ECR	(Interrupt source register)
r7			
r8		PSW	(Program status word)
r9			
r10		CTPC	(CALLT execution status saving register)
r11		CTPSW	(CALLT execution status saving register)
r12			
r13		DBPC	(Exception/debug trap status saving register)
r14		DBPSW	(Exception/debug trap status saving register)
r15			
r16			
r17		CTBP	(CALLT base pointer)
r18			
r19			
r20			
r21			
r22			
r23			
r24			
r25			
r26			
r27			
r28			
r29			
r30	(Element pointer (EP))		
r31	(Link pointer (LP))		
31			
PC	(Program counter)		
			0

3.2.1 Program register set

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

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

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

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

Table 3-1. Program Registers

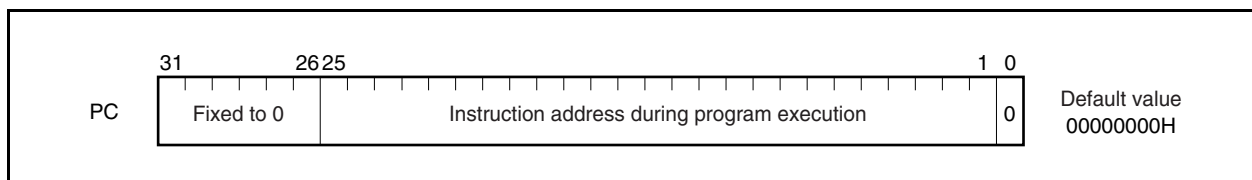
Name	Usage	Operation
r0	Zero register	Always holds 0.
r1	Assembler-reserved register	Used as working register to create 32-bit immediate data
r2	Register for address/data variable (if real-time OS does not use r2)	
r3	Stack pointer	Used to create a stack frame when a function is called
r4	Global pointer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

Remark For further details on the r1, r3 to r5, and r31 that are used in the assembler and C compiler, refer to the **CA850 (C Compiler Package) Assembly Language User's Manual**.

(2) Program counter (PC)

The program counter holds the instruction address during program execution. The lower 32 bits of this register are valid. Bits 31 to 26 are fixed to 0. A carry from bit 25 to 26 is ignored even if it occurs.

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



3.2.2 System register set

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

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

Table 3-2. System Register Numbers

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

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

2. These registers can be accessed only during the interval between the execution of the DBTRAP instruction or illegal opcode and DBRET instruction execution.

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

Remark √: Can be accessed

×: Access prohibited

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

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

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

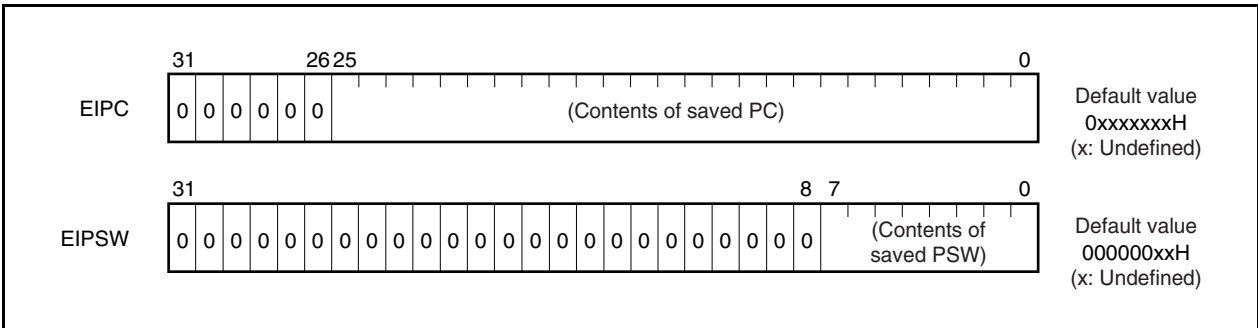
The address of the instruction next to the instruction under execution, except some instructions (see 19.8 Periods in Which Interrupts Are Not Acknowledged by CPU), is saved to EIPC when a software exception or a maskable interrupt occurs.

The current contents of the PSW are saved to EIPSW.

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

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

The value of EIPC is restored to the PC and the value of EIPSW to the PSW by the RETI instruction.



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

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

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

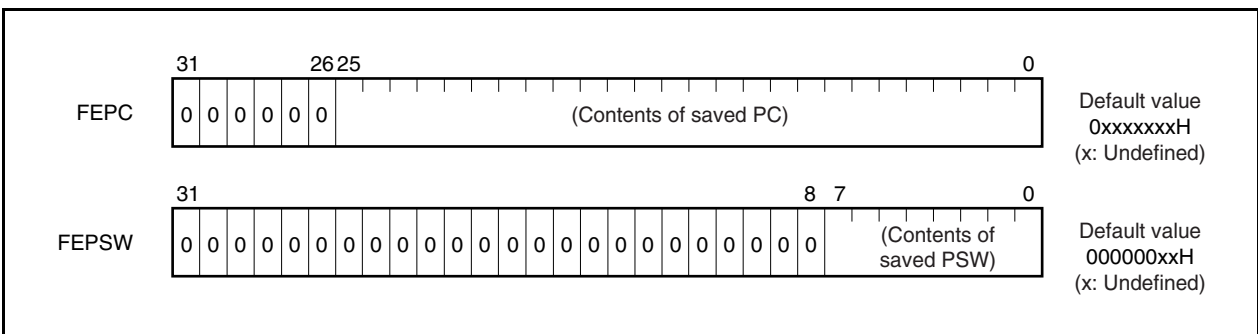
The address of the instruction next to the one of the instruction under execution, except some instructions, is saved to FEPC when an NMI occurs.

The current contents of the PSW are saved to FEPSW.

Because only one set of NMI status saving registers is available, the contents of these registers must be saved by program when multiple interrupts are enabled (for multiple interrupt servicing using the NMI pin and the INTWDT2 interrupt request signal).

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

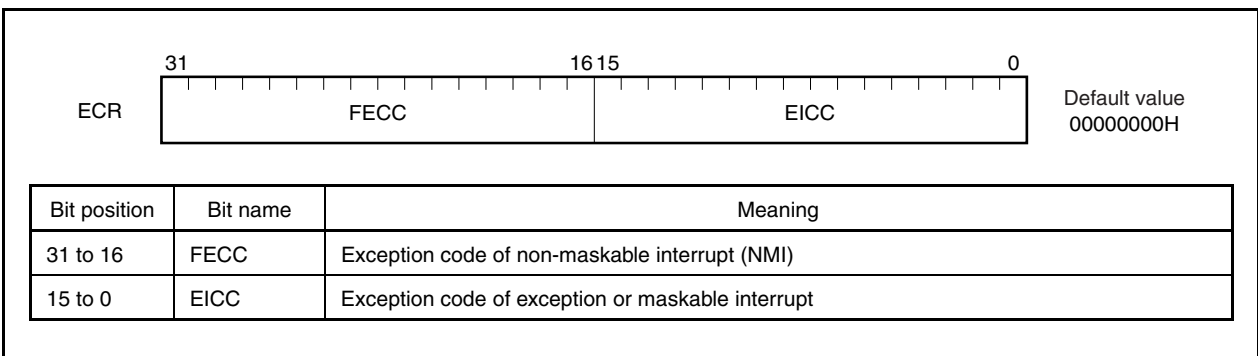
The value of FEPC is restored to the PC and the value of FEPSW to the PSW by the RETI instruction.



(3) Interrupt source register (ECR)

The interrupt source register (ECR) holds the source of an exception or interrupt if an exception or interrupt occurs.

This register holds the exception code of each interrupt source. Because this register is a read-only register, data cannot be written to this register using the LDSR instruction.



(4) Program status word (PSW)

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

If the contents of a bit of this register are changed by using the LDSR instruction, the new contents are validated immediately after completion of LDSR instruction execution. However if the ID flag is set to 1, interrupt requests will not be acknowledged while the LDSR instruction is being executed.

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

(1/2)

	31		8 7 6 5 4 3 2 1 0	
PSW	0 0	NP EP ID SAT CY OV S Z		Default value 0000020H

Bit position	Flag name	Meaning
7	NP	Indicates that a non-maskable interrupt (NMI) is being serviced. This bit is set to 1 when an NMI request is acknowledged, disabling multiple interrupts. 0: NMI is not being serviced. 1: NMI is being serviced.
6	EP	Indicates that an exception is being processed. This bit is set to 1 when an exception occurs. Even if this bit is set, interrupt requests are acknowledged. 0: Exception is not being processed. 1: Exception is being processed.
5	ID	Indicates whether a maskable interrupt can be acknowledged. 0: Interrupt enabled 1: Interrupt disabled
4	SAT ^{Note}	Indicates that the result of a saturation operation has overflowed and is saturated. Because this is a cumulative flag, it is set to 1 when the result of a saturation operation instruction is saturated, and is not cleared to 0 even if the subsequent operation result is not saturated. Use the LDSR instruction to clear this bit. This flag is neither set to 1 nor cleared to 0 by execution of an arithmetic operation instruction. 0: Not saturated 1: Saturated
3	CY	Indicates whether a carry or a borrow occurs as a result of an operation. 0: Carry or borrow does not occur. 1: Carry or borrow occurs.
2	OV ^{Note}	Indicates whether an overflow occurs during operation. 0: Overflow does not occur. 1: Overflow occurs.
1	S ^{Note}	Indicates whether the result of an operation is negative. 0: The result is positive or 0. 1: The result is negative.
0	Z	Indicates whether the result of an operation is 0. 0: The result is not 0. 1: The result is 0.

Remark Also read **Note** on the next page.

(2/2)

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

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

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

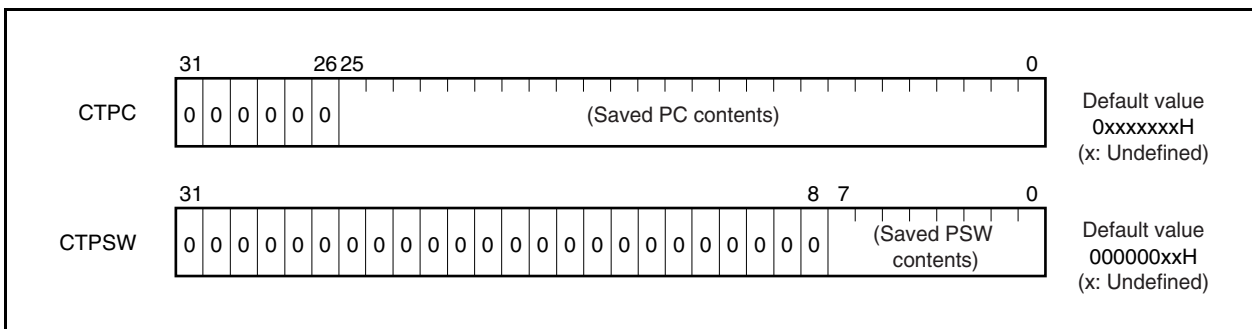
CTPC and CTPSW are CALLT execution status saving registers.

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

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

The current contents of the PSW are saved to CTPSW.

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



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

DBPC and DBPSW are exception/debug trap status registers.

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

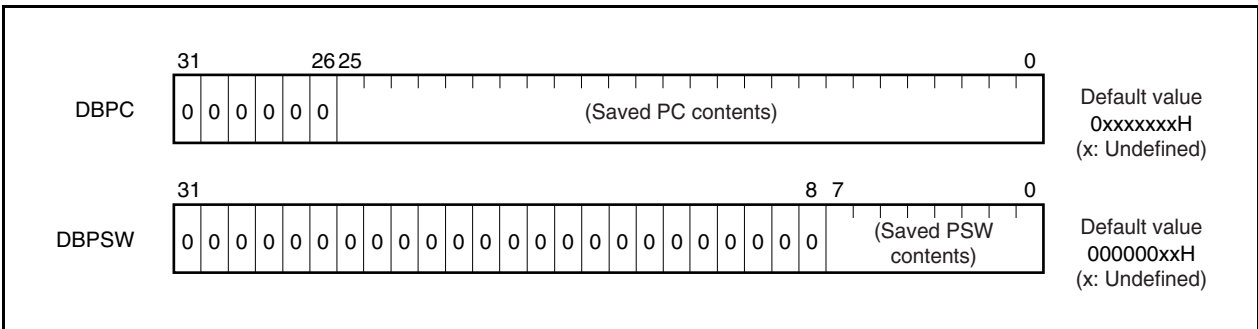
The contents to be saved to DBPC are the address of the instruction next to the one that is being executed when an exception trap or debug trap occurs.

The current contents of the PSW are saved to DBPSW.

This register can be read or written only during the interval between the execution of the DBTRAP instruction or illegal opcode and the DBRET instruction.

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

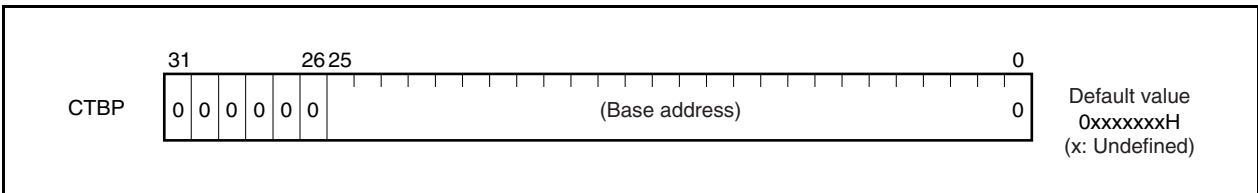
The value of DBPC is restored to the PC and the value of DBPSW to the PSW by the DBRET instruction.



(7) CALLT base pointer (CTBP)

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

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



3.3 Operation Modes

The V850ES/JC3-L and V850ES/JE3-L have the following operation modes.

- Normal operation mode
- Flash memory programming mode
- Self programming mode
- On-chip debug mode

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

To specify the normal operation mode, input a low level to the FLMD0 pin during the reset period.

A high level is input to the FLMD0 pin by the flash memory programmer in the flash memory programming mode if a flash programmer is connected. In the self-programming mode, input a high level to this pin from an external circuit.

Fix the specification of these pins in the application system and do not change the setting of these pins during operation.

FLMD0	FLMD1	Operation Mode
L	×	Normal operation mode
H	L	Flash memory programming mode
H	H	Setting prohibited

Remark H: High level
L: Low level
×: don't care

(1) Normal operation mode

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

(2) Flash memory programming mode

When this mode is specified, the internal flash memory can be programmed by using a flash programmer.

(3) Self programming mode

Data can be erased and written from/to the flash memory by using a user application program. For details, see **CHAPTER 28 FLASH MEMORY**.

(4) On-chip debug mode

The V850ES/JC3-L, V850ES/JE3-L is provided with an on-chip debug function that employs the JTAG (Joint Test Action Group) communication specifications.

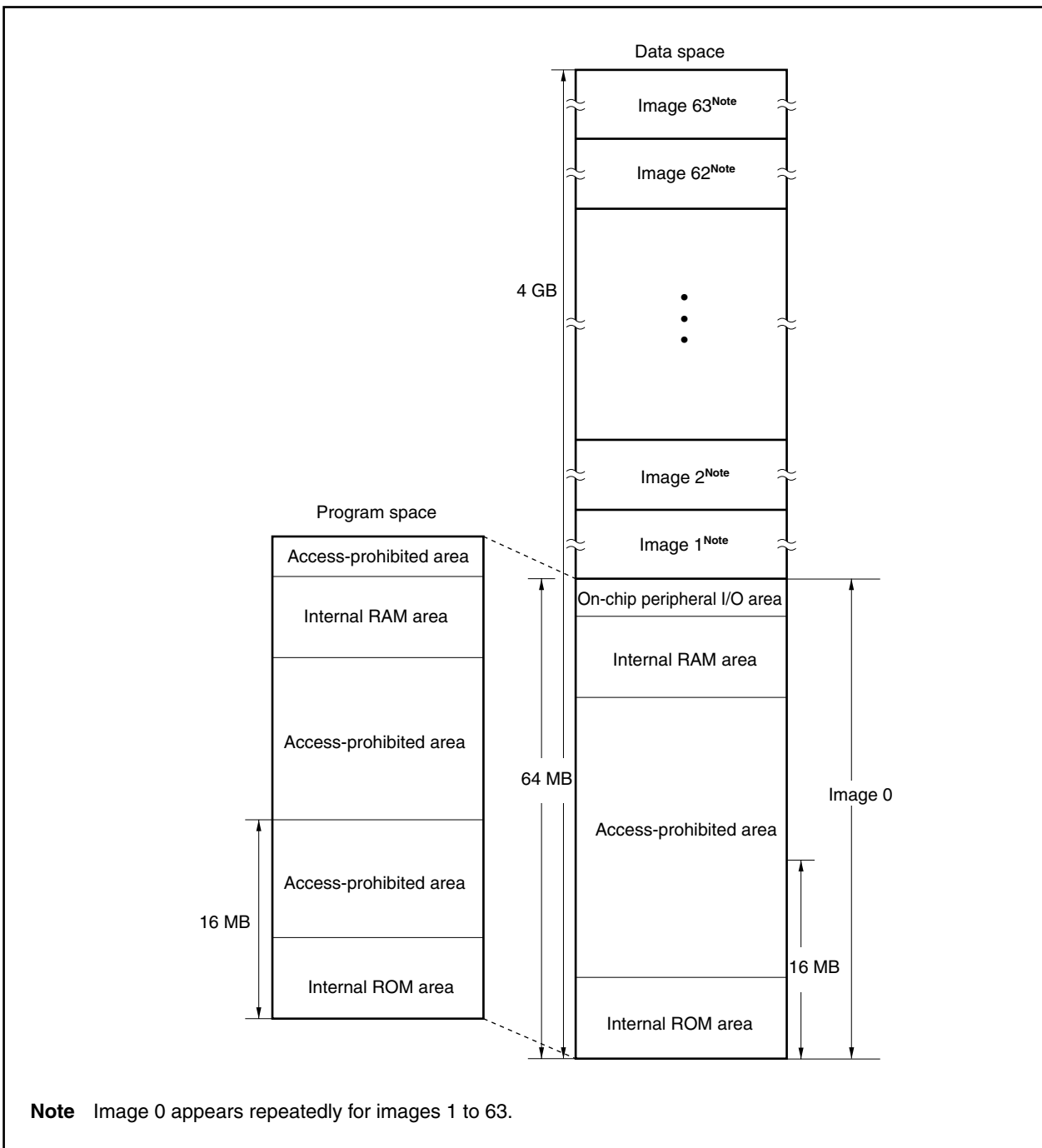
For details, see **CHAPTER 29 ON-CHIP DEBUG FUNCTION**.

3.4 Address Space

3.4.1 CPU address space

For instruction addressing, up to a combined total of 16 MB of external memory area and internal ROM area, plus an internal RAM area, are supported in a linear address space (program space) of up to 64 MB. For operand addressing (data access), up to 4 GB of a linear address space (data space) is supported. The 4 GB address space, however, is viewed as 64 images of a 64 MB physical address space. This means that the same 64 MB physical address space is accessed regardless of the value of bits 31 to 26.

Figure 3-1. Address Space Image



3.4.2 Memory map

The areas shown below are reserved in the V850ES/JC3-L, V850ES/JE3-L.

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

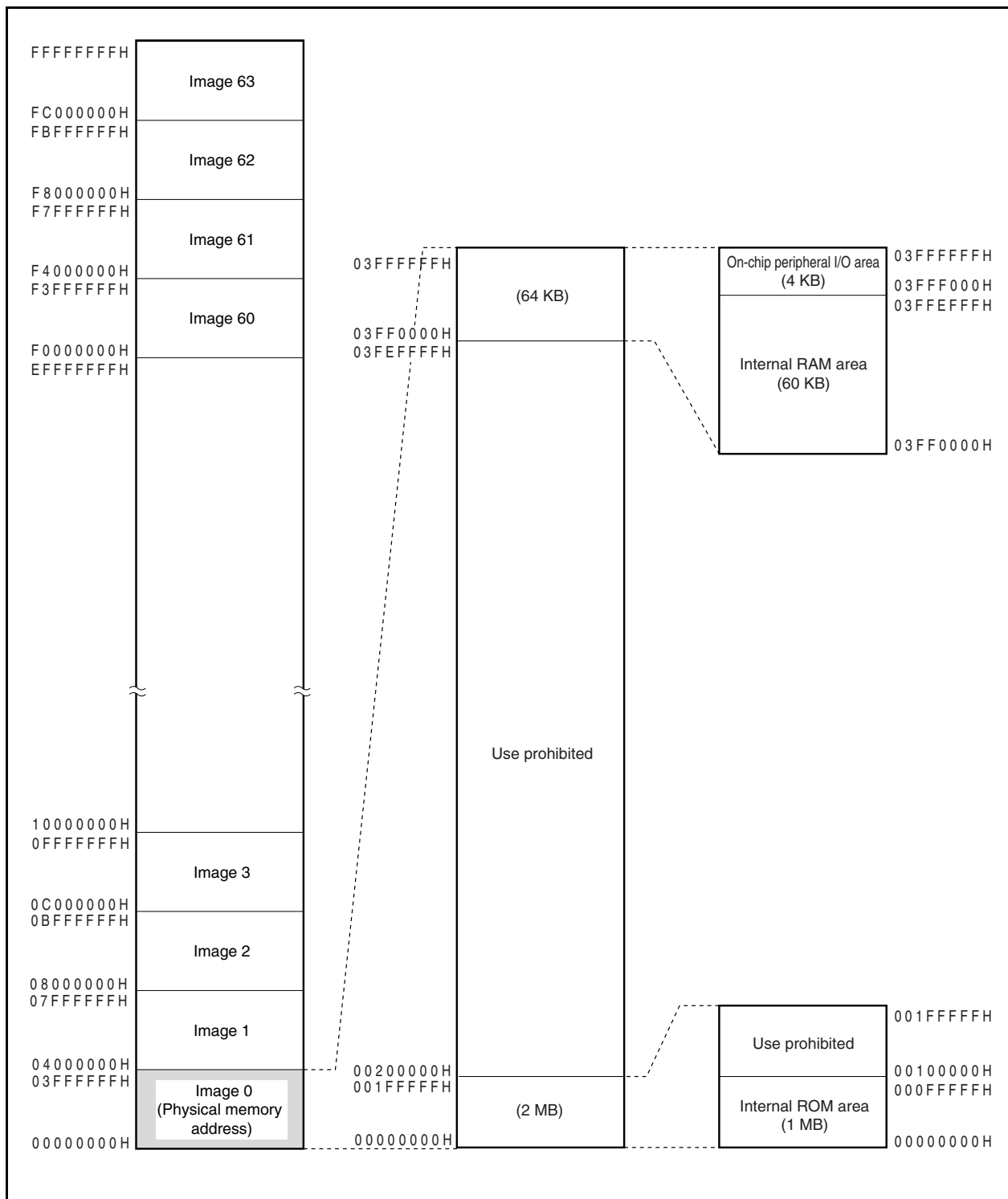
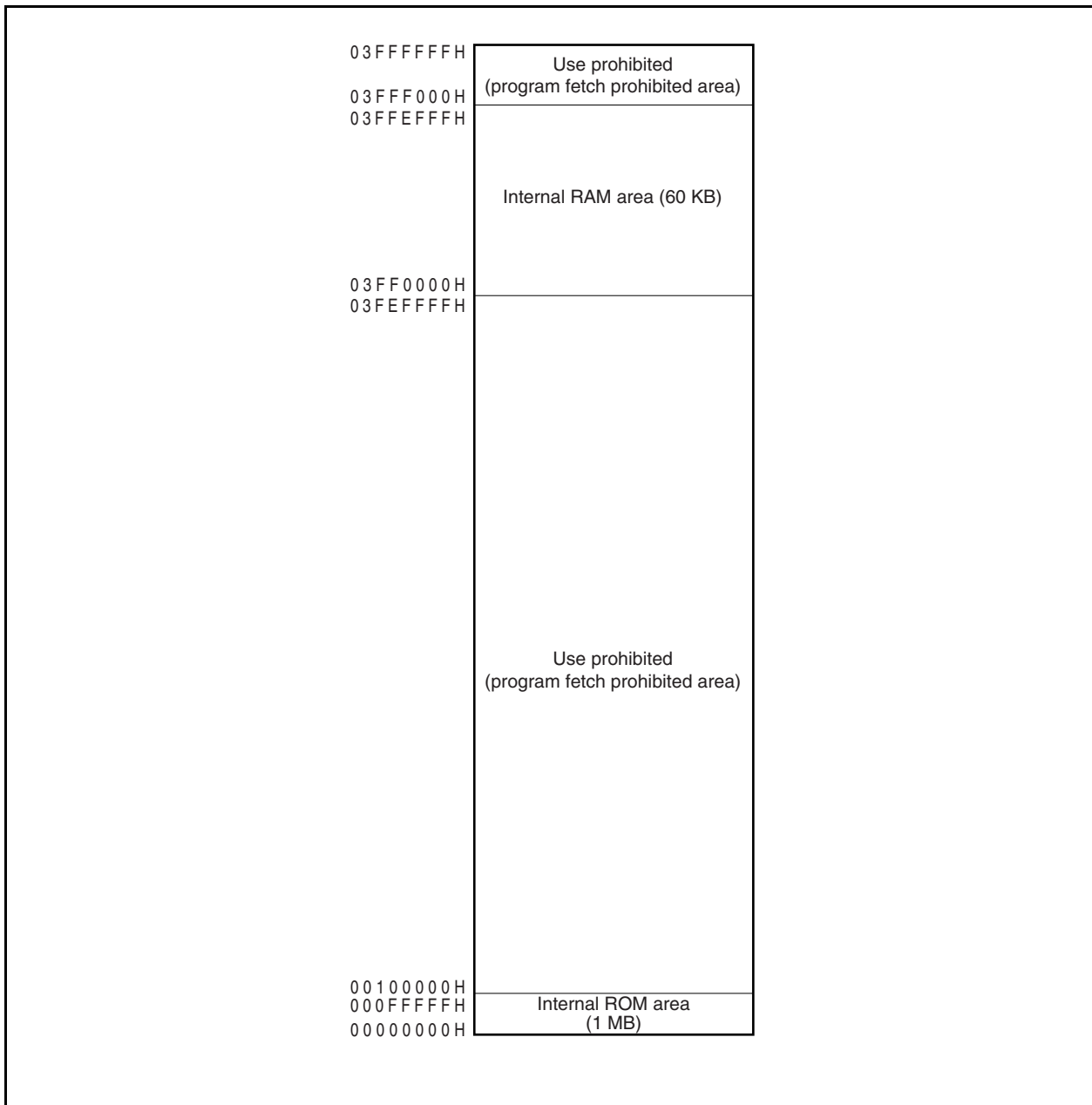


Figure 3-3. Program Memory Map



3.4.3 Areas

(1) Internal ROM area

Up to 1 MB is reserved as an internal ROM area.

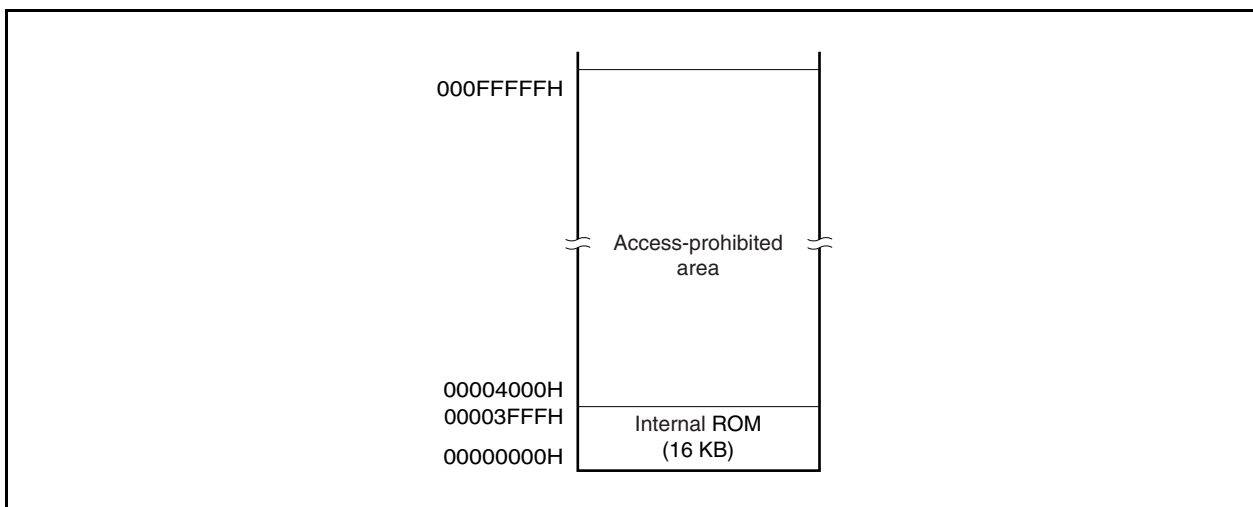
(a) Internal ROM (16 KB)

16 KB are allocated to addresses 00000000H to 00003FFFH in the following versions.

Accessing addresses 00004000H to 000FFFFFFH is prohibited.

- μ PD70F3797, 70F3801, 70F3805

Figure 3-4. Internal ROM Area (16 KB)



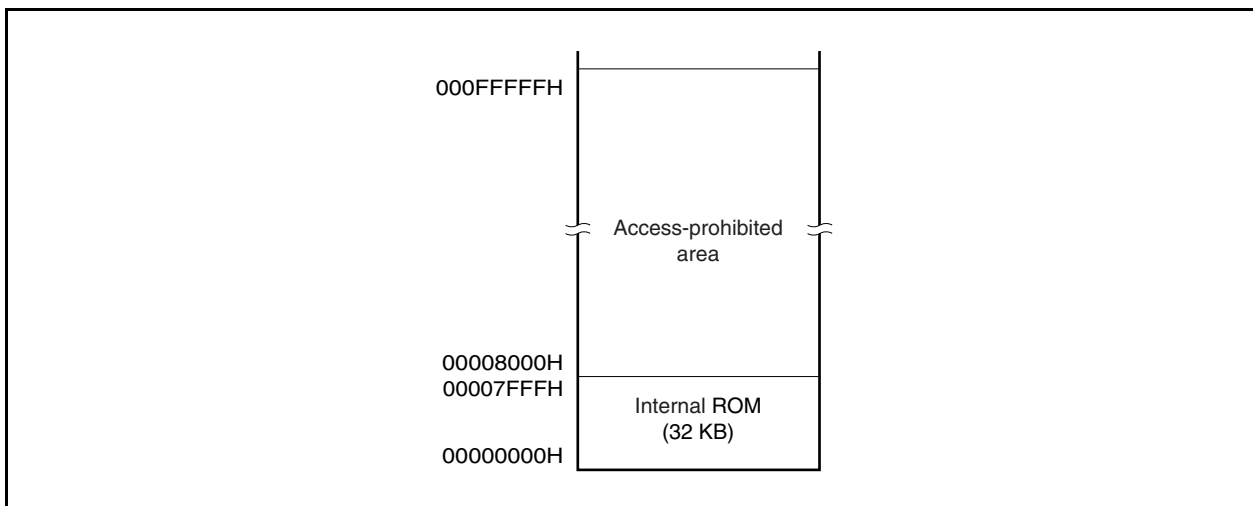
(b) Internal ROM (32 KB)

32 KB are allocated to addresses 00000000H to 00007FFFH in the following versions.

Accessing addresses 00008000H to 000FFFFFFH is prohibited.

- μ PD70F3798, 70F3802, 70F3806

Figure 3-5. Internal ROM Area (32 KB)

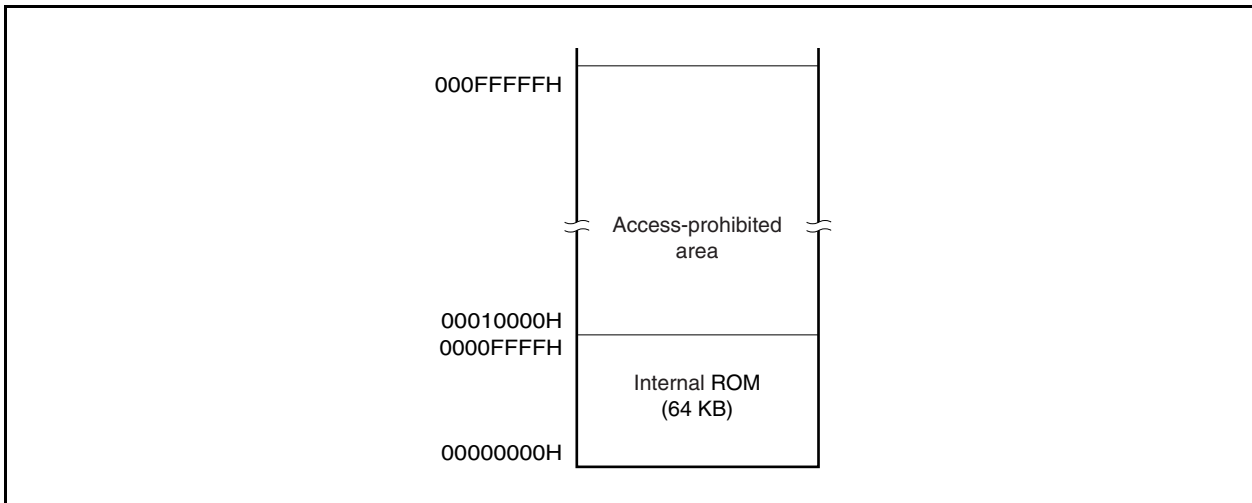


(c) Internal ROM (64 KB)

64 KB are allocated to addresses 00000000H to 0000FFFFH in the following versions.
Accessing addresses 00010000H to 000FFFFFH is prohibited.

- μ PD70F3799, 70F3803, 70F3807

Figure 3-6. Internal ROM Area (64 KB)

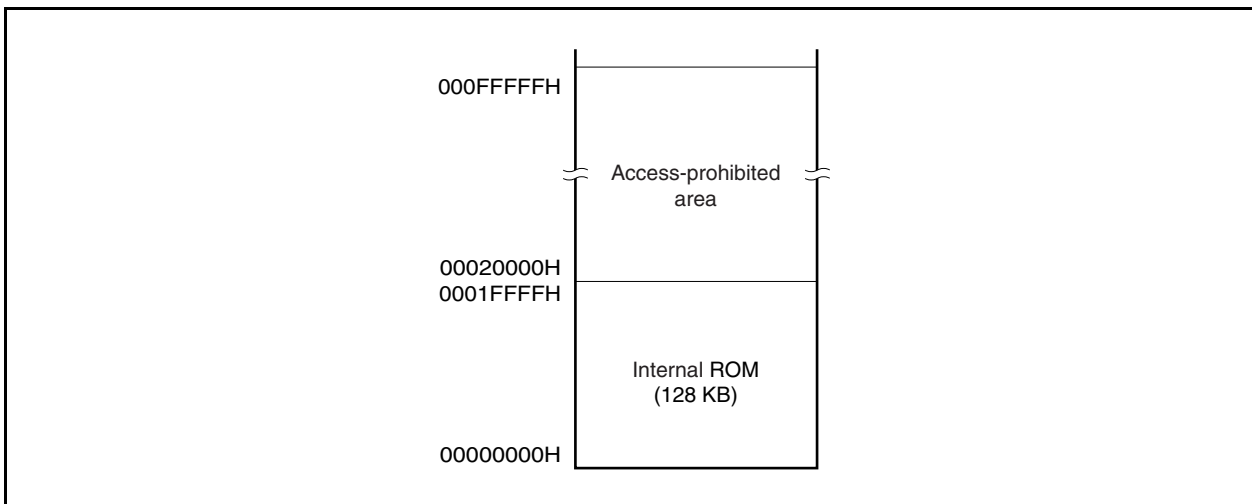


(d) Internal ROM (128 KB)

128 KB are allocated to addresses 00000000H to 0001FFFFH in the following versions.
Accessing addresses 00020000H to 000FFFFFH is prohibited.

- μ PD70F3800, 70F3804, 70F3808

Figure 3-7. Internal ROM Area (128 KB)

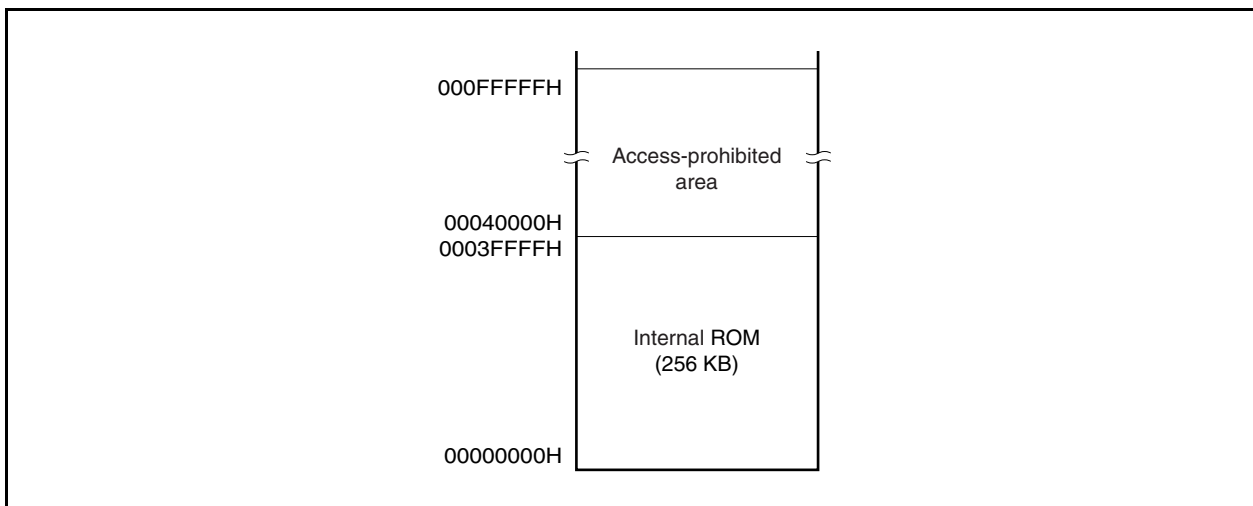


(e) Internal ROM (256 KB)

256 KB are allocated to addresses 00000000H to 0003FFFFH in the following versions.
 Accessing addresses 00040000H to 000FFFFFFH is prohibited.

- μ PD70F3838, 70F3839, 70F3840

Figure 3-8. Internal ROM Area (256 KB)



(2) Internal RAM area

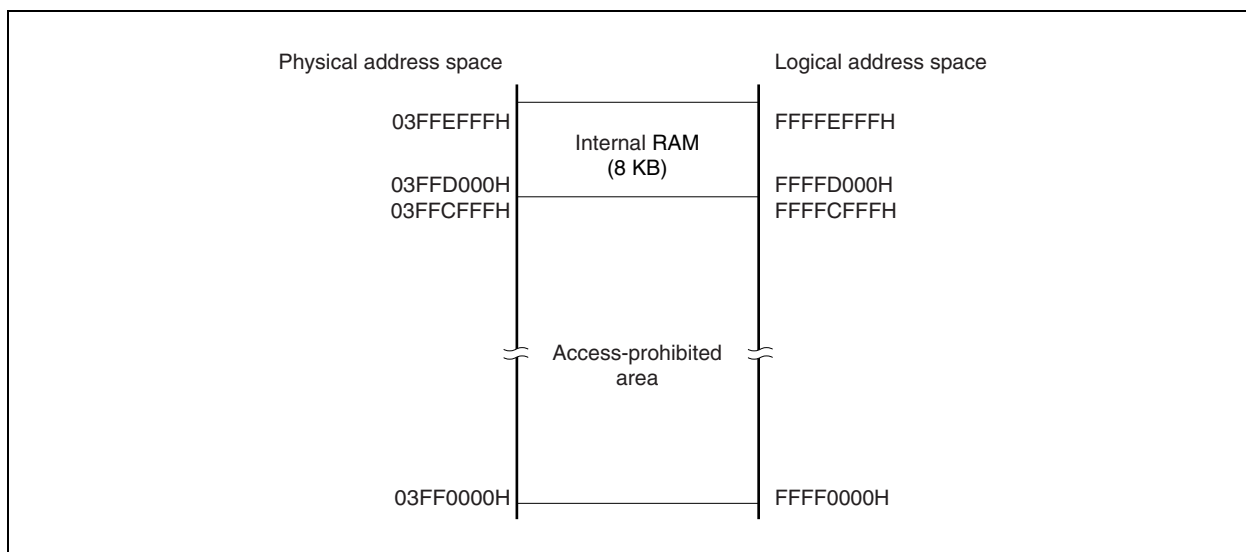
Up to 60 KB allocated to physical addresses 03FF0000H to 03FFFEFFFH are reserved as the internal RAM area.

(a) Internal RAM (8 KB)

8 KB are allocated to addresses 03FFD000H to 03FFFEFFFH in the following versions.
 Accessing addresses 03FF0000H to 03FFCFFFH is prohibited.

- μ PD70F3797, 70F3798, 70F3799, 70F3800, 70F3801, 70F3802, 70F3803, 70F3804, 70F3805, 70F3806, 70F3807, 70F3808

Figure 3-9. Internal RAM Area (8 KB)



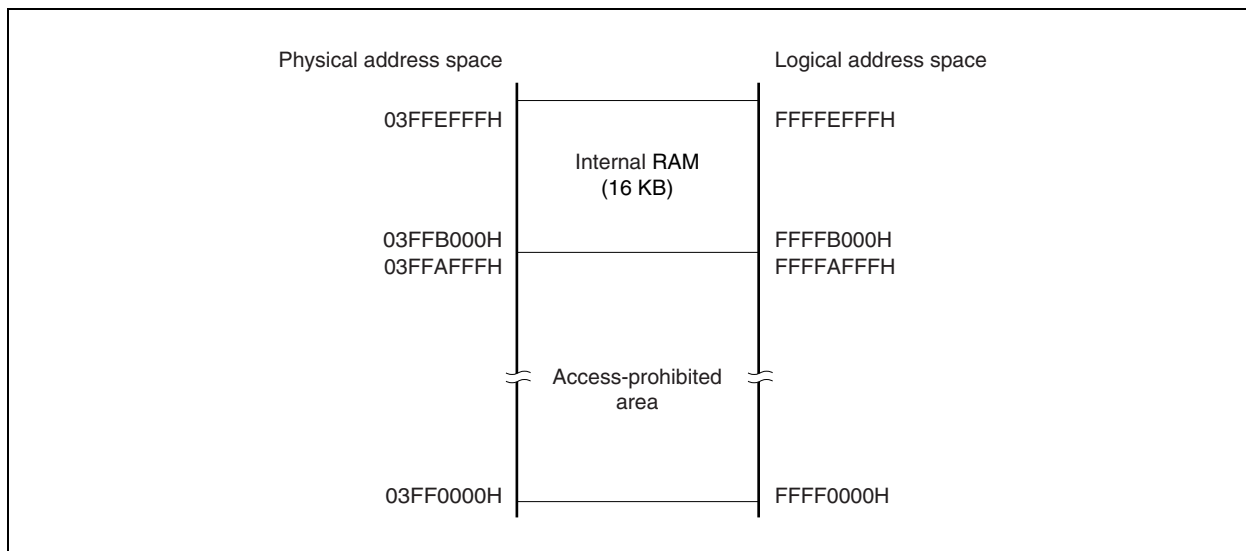
(b) Internal RAM (16 KB)

16 KB are allocated to addresses 03FFB000H to 03FFEFFFH in the following versions.

Accessing addresses 03FF0000H to 03FFAFFFH is prohibited.

- μ PD70F3838, 70F3839, 70F3840

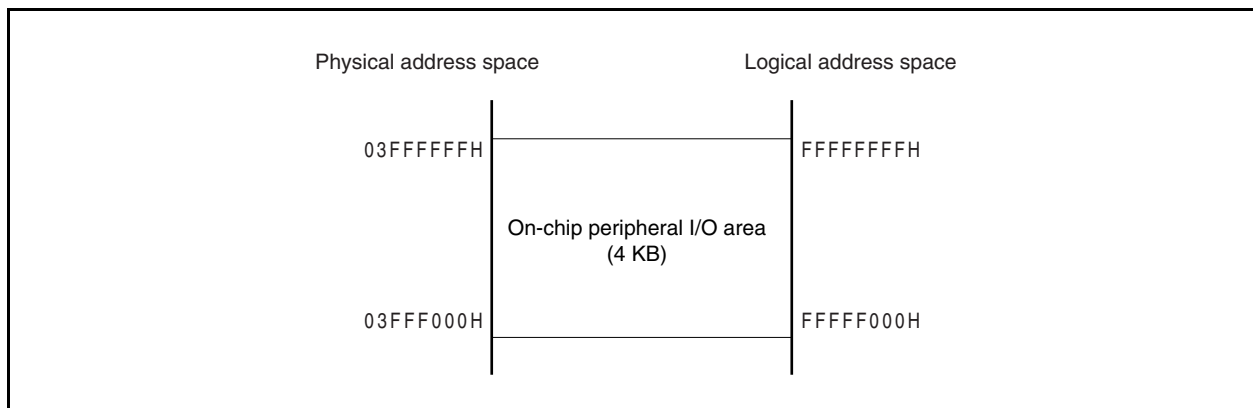
Figure 3-10. Internal RAM Area (16 KB)



(3) On-chip peripheral I/O area

4 KB allocated to physical addresses 03FFF000H to 03FFFFFFH are reserved as the on-chip peripheral I/O area.

Figure 3-11. On-Chip Peripheral I/O Area



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

- Cautions**
1. When a peripheral I/O register is accessed in word units, a word area is accessed twice in halfword units in the order of lower area then higher area, with the lower 2 bits of the address ignored.
 2. If a peripheral I/O register that can be accessed in byte units is accessed in halfword units, the lower 8 bits are valid. The higher 8 bits are undefined when the register is read and are invalid when the register is written.
 3. Addresses not defined as registers are reserved for future expansion. The operation is undefined and not guaranteed when these addresses are accessed.
 4. The internal ROM/RAM area and on-chip peripheral I/O area are assigned to successive addresses.

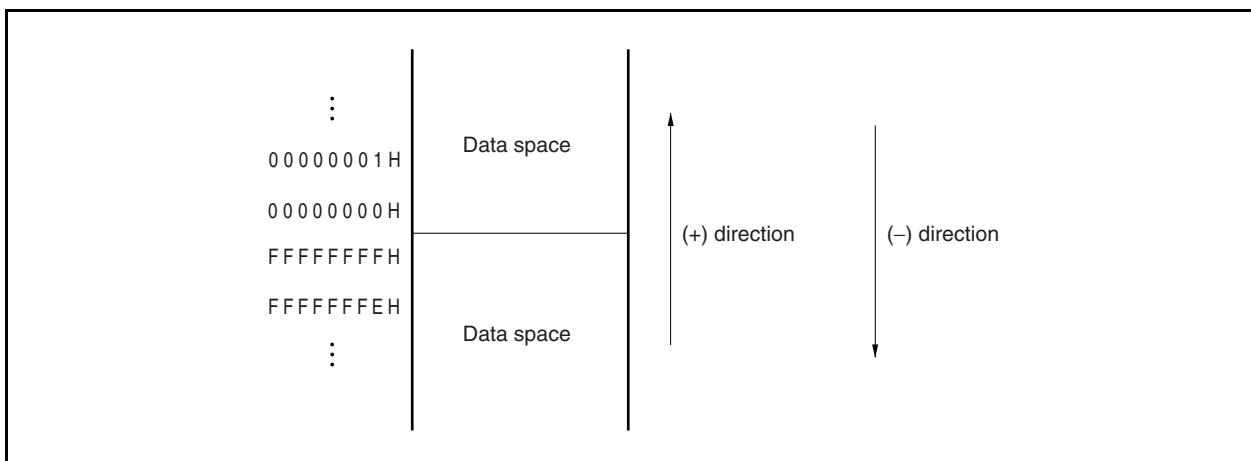
When accessing the internal ROM/RAM area by incrementing or decrementing addresses using a pointer operation for example, be careful not to access the on-chip peripheral I/O area by mistakenly extending over the internal ROM/RAM area boundary.

3.4.4 Wraparound of data space

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

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

Figure 3-12. Wraparound of Data Space



3.4.5 Recommended use of address space

The architecture of the V850ES/JC3-L, V850ES/JE3-L requires that a register that serves as a pointer be secured for address generation when operand data in the data space is accessed. The address stored in this pointer ± 32 KB can be directly accessed by an instruction for operand data. Because the number of general-purpose registers that can be used as a pointer is limited, however, by keeping the performance from dropping during address calculation when a pointer value is changed, as many general-purpose registers as possible can be secured for variables, and the program size can be reduced.

(1) Program space

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

To use the internal RAM area as the program space, access the following addresses.

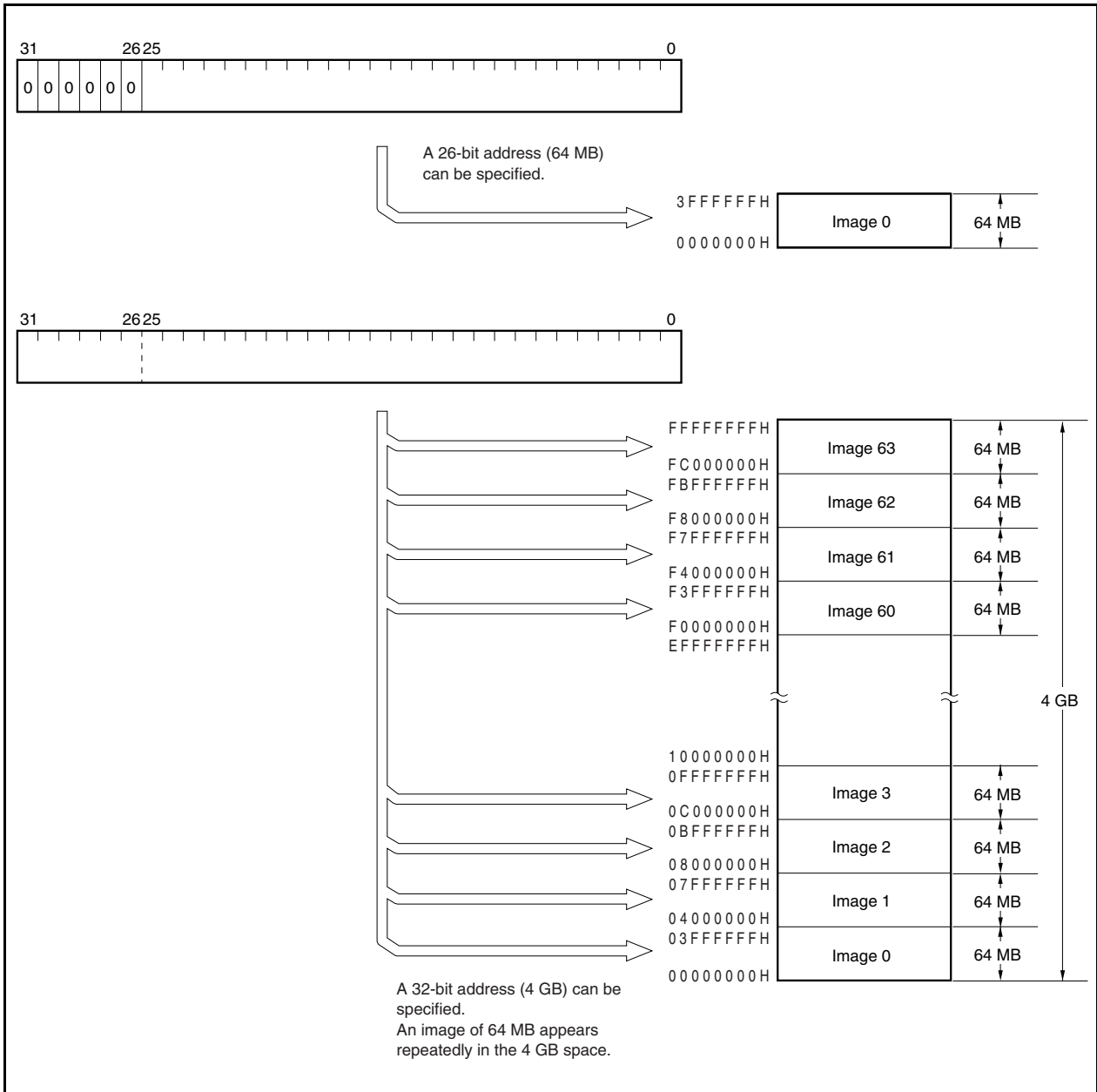
Caution If a branch instruction is at the upper limit of the internal RAM area, a prefetch operation (invalid fetch) straddling the on-chip peripheral I/O area does not occur.

Product Name	RAM Size	Access Address
μ PD70F3797, 70F3798, 70F3799, 70F3800, 70F3801, 70F3802, 70F3803, 70F3804, 70F3805, 70F3806, 70F3807, 70F3808	8 KB	03FFD000H to 03FFFFFFFFH
μ PD70F3838, 70F3839, 70F3840	16 KB	03FFB000H to 03FFFFFFFFH

(2) Data space

With the V850ES/JC3-L, V850ES/JE3-L, it seems that there are sixty-four 64 MB (26-bit address) physical address spaces on the 4 GB (32-bit address) CPU address space. Therefore, the most significant bit (bit 25) of a 26-bit address of these 64 MB spaces is sign-extended to 32 bits and allocated as an address.

Figure 3-13. Sign Extension in Data Space



(a) Application example of wraparound

If R = r0 (zero register) is specified for the LD/ST disp16 [R] instruction, a range of addresses 0000000H ±32 KB can be addressed by sign-extended disp16. All the resources, including the internal hardware, can be addressed by one pointer.

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

Figure 3-14. Example of Data Space Usage in μPD70F3840

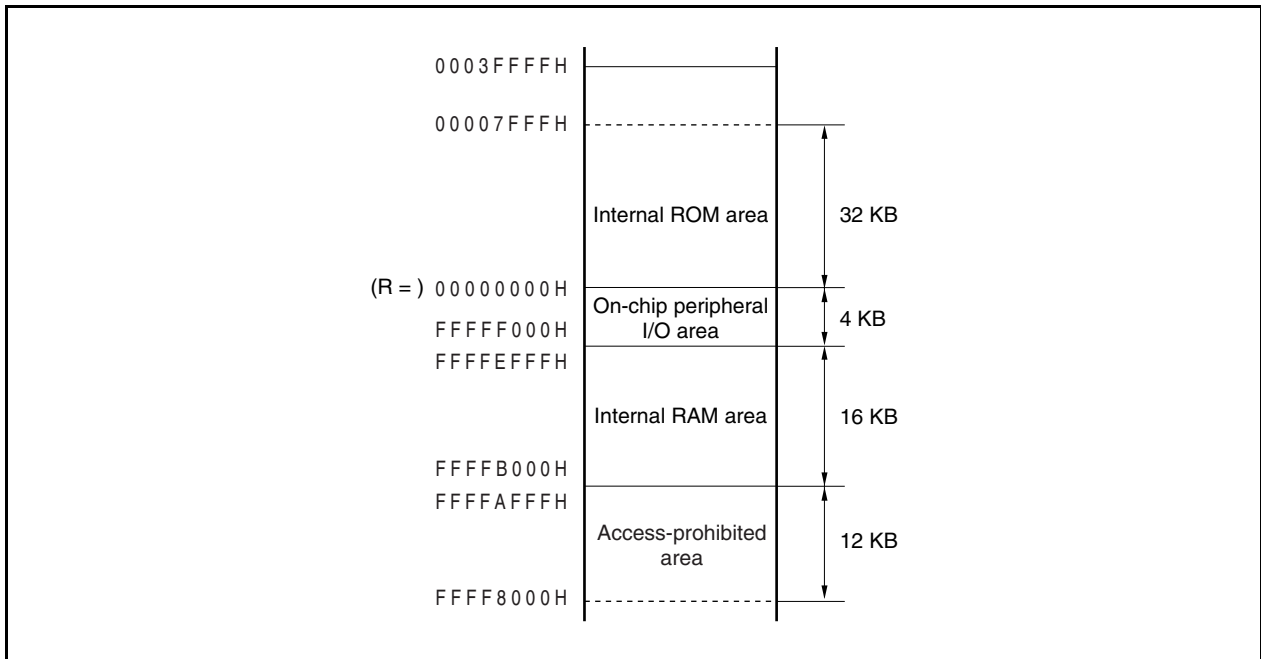
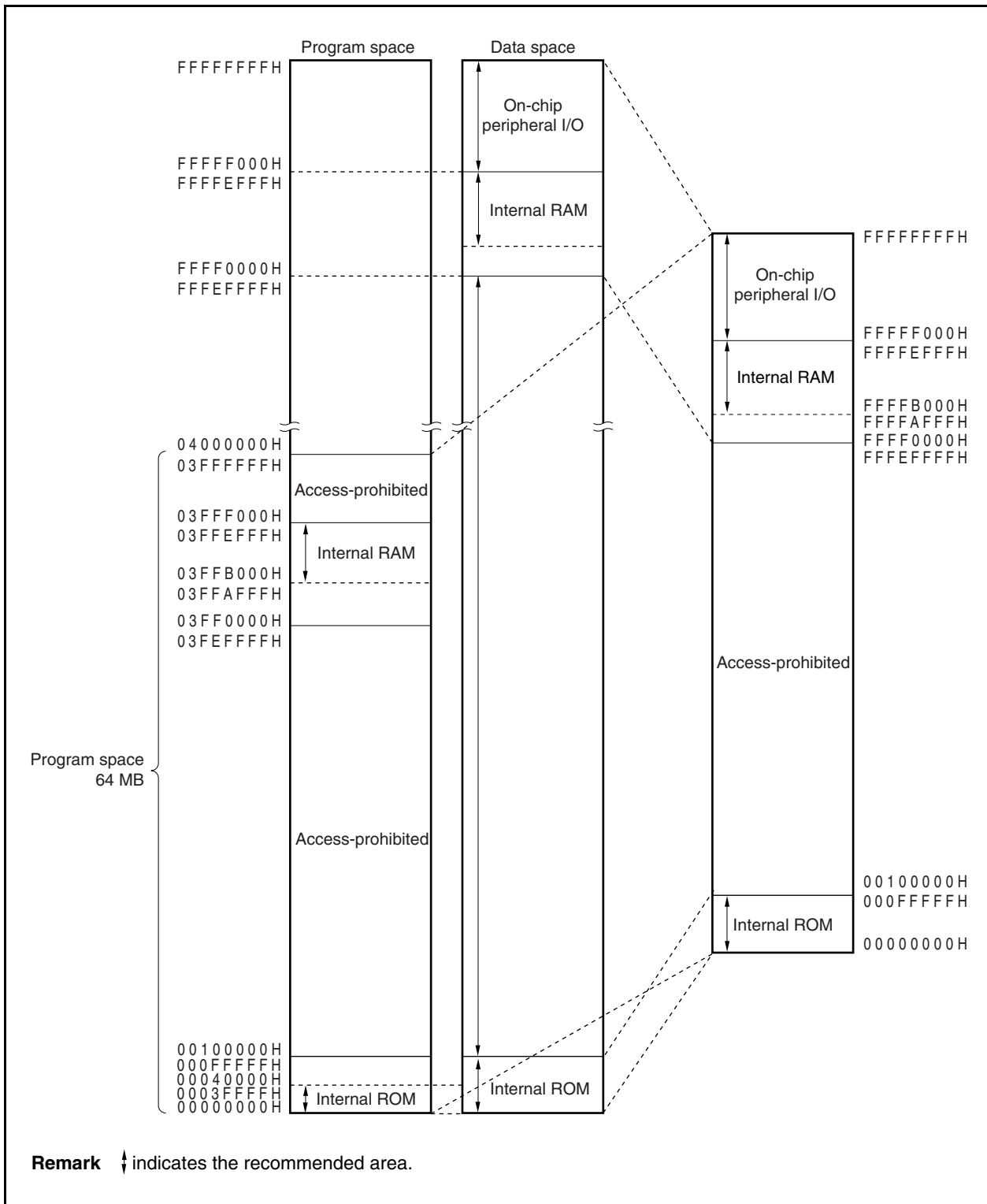


Figure 3-12. Recommended Memory Map (μ PD70F3840)



3.4.6 Peripheral I/O registers

(1/10)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF004H	Port DL register	PDL	R/W			√	0000H ^{Note}
FFFFF004H	Port DL register L	PDLL		√	√		00H ^{Note}
FFFFF00CH	Port CM register	PCM		√	√		00H ^{Note}
FFFFF024H	Port DL mode register	PMDL				√	FFFFH
FFFFF024H	Port DL mode register L	PMDLL		√	√		FFH
FFFFF02CH	Port CM mode register	PMCM		√	√		FFH
FFFFF06EH	System wait control register	VSWC			√		77H
FFFFF080H	DMA source address register 0L	DSA0L				√	Undefined
FFFFF082H	DMA source address register 0H	DSA0H				√	Undefined
FFFFF084H	DMA destination address register 0L	DDA0L				√	Undefined
FFFFF086H	DMA destination address register 0H	DDA0H				√	Undefined
FFFFF088H	DMA source address register 1L	DSA1L				√	Undefined
FFFFF08AH	DMA source address register 1H	DSA1H				√	Undefined
FFFFF08CH	DMA destination address register 1L	DDA1L				√	Undefined
FFFFF08EH	DMA destination address register 1H	DDA1H				√	Undefined
FFFFF090H	DMA source address register 2L	DSA2L				√	Undefined
FFFFF092H	DMA source address register 2H	DSA2H				√	Undefined
FFFFF094H	DMA destination address register 2L	DDA2L				√	Undefined
FFFFF096H	DMA destination address register 2H	DDA2H				√	Undefined
FFFFF098H	DMA source address register 3L	DSA3L				√	Undefined
FFFFF09AH	DMA source address register 3H	DSA3H				√	Undefined
FFFFF09CH	DMA destination address register 3L	DDA3L				√	Undefined
FFFFF09EH	DMA destination address register 3H	DDA3H				√	Undefined
FFFFF0C0H	DMA transfer count register 0	DBC0				√	Undefined
FFFFF0C2H	DMA transfer count register 1	DBC1				√	Undefined
FFFFF0C4H	DMA transfer count register 2	DBC2				√	Undefined
FFFFF0C6H	DMA transfer count register 3	DBC3				√	Undefined
FFFFF0D0H	DMA addressing control register 0	DADC0				√	0000H
FFFFF0D2H	DMA addressing control register 1	DADC1				√	0000H
FFFFF0D4H	DMA addressing control register 2	DADC2				√	0000H
FFFFF0D6H	DMA addressing control register 3	DADC3				√	0000H
FFFFF0E0H	DMA channel control register 0	DCHC0		√	√		00H
FFFFF0E2H	DMA channel control register 1	DCHC1		√	√		00H
FFFFF0E4H	DMA channel control register 2	DCHC2	√	√		00H	
FFFFF0E6H	DMA channel control register 3	DCHC3	√	√		00H	
FFFFF100H	Interrupt mask register 0	IMR0			√	FFFFH	
FFFFF100H	Interrupt mask register 0L	IMR0L	√	√		FFH	
FFFFF101H	Interrupt mask register 0H	IMR0H	√	√		FFH	

Note The output latch is 00H or 0000H. When these registers are in the input mode, the pin statuses are read.

(2/10)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFF102H	Interrupt mask register 1	IMR1	R/W			√	FFFFH
FFFFFF102H	Interrupt mask register 1L	IMR1L		√	√		FFH
FFFFFF103H	Interrupt mask register 1H	IMR1H		√	√		FFH
FFFFFF104H	Interrupt mask register 2	IMR2			√	FFFFH	
FFFFFF104H	Interrupt mask register 2L	IMR2L	√	√		FFH	
FFFFFF105H	Interrupt mask register 2H	IMR2H	√	√		FFH	
FFFFFF106H	Interrupt mask register 3	IMR3			√	FFFFH	
FFFFFF106H	Interrupt mask register 3L	IMR3L	√	√		FFH	
FFFFFF110H	Interrupt control register (INTLVI)	LVIIC	√	√		47H	
FFFFFF112H	Interrupt control register (INTP0)	PIC0	√	√		47H	
FFFFFF114H	Interrupt control register (INTP1)	PIC1	√	√		47H	
FFFFFF116H	Interrupt control register (INTP2)	PIC2	√	√		47H	
FFFFFF118H	Interrupt control register (INTP3)	PIC3	√	√		47H	
FFFFFF11AH	Interrupt control register (INTP4)	PIC4	√	√		47H	
FFFFFF11CH	Interrupt control register (INTP5)	PIC5	√	√		47H	
FFFFFF11EH	Interrupt control register (INTP6)	PIC6	√	√		47H	
FFFFFF120H	Interrupt control register (INTP7)	PIC7	√	√		47H	
FFFFFF122H	Interrupt control register (INTTQ0OV)	TQ0OVIC	√	√		47H	
FFFFFF124H	Interrupt control register (INTTQ0CC0)	TQ0CCIC0	√	√		47H	
FFFFFF126H	Interrupt control register (INTTQ0CC1)	TQ0CCIC1	√	√		47H	
FFFFFF128H	Interrupt control register (INTTQ0CC2)	TQ0CCIC2	√	√		47H	
FFFFFF12AH	Interrupt control register (INTTQ0CC3)	TQ0CCIC3	√	√		47H	
FFFFFF12CH	Interrupt control register (INTTP0OV)	TP0OVIC	√	√		47H	
FFFFFF12EH	Interrupt control register (INTTP0CC0)	TP0CCIC0	√	√		47H	
FFFFFF130H	Interrupt control register (INTTP0CC1)	TP0CCIC1	√	√		47H	
FFFFFF132H	Interrupt control register (INTTP1OV)	TP1OVIC	√	√		47H	
FFFFFF134H	Interrupt control register (INTTP1CC0)	TP1CCIC0	√	√		47H	
FFFFFF136H	Interrupt control register (INTTP1CC1)	TP1CCIC1	√	√		47H	
FFFFFF138H	Interrupt control register (INTTP2OV)	TP2OVIC	√	√		47H	
FFFFFF13AH	Interrupt control register (INTTP2CC0)	TP2CCIC0	√	√		47H	
FFFFFF13CH	Interrupt control register (INTTP2CC1)	TP2CCIC1	√	√		47H	
FFFFFF13EH	Interrupt control register (INTTP3OV)	TP3OVIC	√	√		47H	
FFFFFF140H	Interrupt control register (INTTP3CC0)	TP3CCIC0	√	√		47H	
FFFFFF142H	Interrupt control register (INTTP3CC1)	TP3CCIC1	√	√		47H	
FFFFFF144H	Interrupt control register (INTTP4OV)	TP4OVIC	√	√		47H	
FFFFFF146H	Interrupt control register (INTTP4CC0)	TP4CCIC0	√	√		47H	
FFFFFF148H	Interrupt control register (INTTP4CC1)	TP4CCIC1	√	√		47H	
FFFFFF14AH	Interrupt control register (INTTP5OV)	TP5OVIC	√	√		47H	
FFFFFF14CH	Interrupt control register (INTTP5CC0)	TP5CCIC0	√	√		47H	
FFFFFF14EH	Interrupt control register (INTTP5CC1)	TP5CCIC1	√	√		47H	
FFFFFF150H	Interrupt control register (INTTM0EQ0)	TM0EQIC0	√	√		47H	

(3/10)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value	
				1	8	16		
FFFFF152H	Interrupt control register (INTCB0R/INTIIC1)	CB0RIC/IICIC1	R/W	√	√		47H	
FFFFF154H	Interrupt control register (INTCB0T)	CB0TIC		√	√		47H	
FFFFF156H	Interrupt control register (INTCB1R)	CB1RIC		√	√		47H	
FFFFF158H	Interrupt control register (INTCB1T)	CB1TIC		√	√		47H	
FFFFF15AH	Interrupt control register (INTCB2R)	CB2RIC		√	√		47H	
FFFFF15CH	Interrupt control register (INTCB2T)	CB2TIC		√	√		47H	
FFFFF15EH	Interrupt control register (INTCB3R)	CB3RIC		√	√		47H	
FFFFF160H	Interrupt control register (INTCB3T)	CB3TIC		√	√		47H	
FFFFF162H	Interrupt control register (INTUA0R/INTCB4R)	UA0RIC/CB4RIC		√	√		47H	
FFFFF164H	Interrupt control register (INTUA0T/INTCB4T)	UA0TIC/CB4TIC		√	√		47H	
FFFFF166H	Interrupt control register (INTUA1R/INTIIC2)	UA1RIC/IICIC2		√	√		47H	
FFFFF168H	Interrupt control register (INTUA1T)	UA1TIC		√	√		47H	
FFFFF16AH	Interrupt control register (INTUA2R/INTIIC0)	UA2RIC/IICIC0		√	√		47H	
FFFFF16CH	Interrupt control register (INTUA2T)	UA2TIC		√	√		47H	
FFFFF16EH	Interrupt control register (INTAD)	ADIC		√	√		47H	
FFFFF170H	Interrupt control register (INTDMA0)	DMAIC0		√	√		47H	
FFFFF172H	Interrupt control register (INTDMA1)	DMAIC1		√	√		47H	
FFFFF174H	Interrupt control register (INTDMA2)	DMAIC2		√	√		47H	
FFFFF176H	Interrupt control register (INTDMA3)	DMAIC3		√	√		47H	
FFFFF178H	Interrupt control register (INTKR)	KRIC		√	√		47H	
FFFFF17AH	Interrupt control register (INTWTI/INTRTC2)	WTIIC/RTC2IC		√	√		47H	
FFFFF17CH	Interrupt control register (INTWT/INTRTC0)	WTIC/RTC0IC		√	√		47H	
FFFFF17EH	Interrupt control register (INTRTC1)	RTC1C		√	√		47H	
FFFFF1FAH	In-service priority register	ISPR		R	√	√		00H
FFFFF1FCH	Command register	PRCMD	W		√		Undefined	
FFFFF1FEH	Power save control register	PSC ^{Note}	R/W	√	√		00H	
FFFFF200H	A/D converter mode register 0	ADA0M0		√	√		00H	
FFFFF201H	A/D converter mode register 1	ADA0M1		√	√		00H	
FFFFF202H	A/D converter channel specification register	ADA0S		√	√		00H	
FFFFF203H	A/D converter mode register 2	ADA0M2		√	√		00H	
FFFFF204H	Power-fail compare mode register	ADA0PFM		√	√		00H	
FFFFF205H	Power-fail compare threshold value register	ADA0PFT		√	√		00H	
FFFFF210H	A/D conversion result register 0	ADA0CR0		R			√	Undefined
FFFFF211H	A/D conversion result register 0H	ADA0CR0H				√		Undefined
FFFFF212H	A/D conversion result register 1	ADA0CR1					√	Undefined
FFFFF213H	A/D conversion result register 1H	ADA0CR1H			√		Undefined	
FFFFF214H	A/D conversion result register 2	ADA0CR2				√	Undefined	
FFFFF215H	A/D conversion result register 2H	ADA0CR2H			√		Undefined	
FFFFF216H	A/D conversion result register 3	ADA0CR3				√	Undefined	
FFFFF217H	A/D conversion result register 3H	ADA0CR3H			√		Undefined	

Note This is a special register.

(4/10)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value	
				1	8	16		
FFFFF218H	A/D conversion result register 4	ADA0CR4	R			√	Undefined	
FFFFF219H	A/D conversion result register 4H	ADA0CR4H			√		Undefined	
FFFFF21AH	A/D conversion result register 5	ADA0CR5				√	Undefined	
FFFFF21BH	A/D conversion result register 5H	ADA0CR5H			√		Undefined	
FFFFF21CH	A/D conversion result register 6	ADA0CR6				√	Undefined	
FFFFF21DH	A/D conversion result register 6H	ADA0CR6H			√		Undefined	
FFFFF21EH	A/D conversion result register 7	ADA0CR7				√	Undefined	
FFFFF21FH	A/D conversion result register 7H	ADA0CR7H			√		Undefined	
FFFFF220H	A/D conversion result register 8	ADA0CR8				√	Undefined	
FFFFF221H	A/D conversion result register 8H	ADA0CR8H			√		Undefined	
FFFFF222H	A/D conversion result register 9	ADA0CR9				√	Undefined	
FFFFF223H	A/D conversion result register 9H	ADA0CR9H			√		Undefined	
FFFFF280H	D/A conversion value setting register 0	DA0CS0		R/W		√		00H
FFFFF281H	D/A conversion value setting register 1	DA0CS1				√		00H
FFFFF282H	D/A converter mode register	DA0M			√	√		00H
FFFFF300H	Key return mode register	KRM			√	√		00H
FFFFF308H	Selector operation control register 0	SELCNT0	√		√		00H	
FFFFF310H	CRC input register	CRCIN			√		00H	
FFFFF312H	CRC data register	CRCDC				√	0000H	
FFFFF318H	Noise elimination control register	NFC			√		00H	
FFFFF320H	Prescaler mode register 1	PRSM1	√		√		00H	
FFFFF321H	Prescaler compare register 1	PRSCM1			√		00H	
FFFFF324H	Prescaler mode register 2	PRSM2	√		√		00H	
FFFFF325H	Prescaler compare register 2	PRSCM2			√		00H	
FFFFF328H	Prescaler mode register 3	PRSM3	√		√		00H	
FFFFF329H	Prescaler compare register 3	PRSCM3			√		00H	
FFFFF331H	Regulator protection register	REGPR			√		00H	
FFFFF332H	Regulator output voltage level control register	REGOVL0			√		00H	
FFFFF340H	IIC division clock select register	OCKS0			√		00H	
FFFFF344H	IIC division clock select register	OCKS1			√		00H	
FFFFF400H	Port 0 register	P0	√		√		00H ^{Note}	
FFFFF402H	Port 1 register	P1	√		√		00H ^{Note}	
FFFFF406H	Port 3 register	P3				√	0000H ^{Note}	
FFFFF406H	Port 3 register L	P3L	√		√		00H ^{Note}	
FFFFF407H	Port 3 register H	P3H	√		√		00H ^{Note}	
FFFFF408H	Port 4 register	P4	√		√		00H ^{Note}	
FFFFF40AH	Port 5 register	P5	√		√		00H ^{Note}	
FFFFF40EH	Port 7 register L	P7L	√		√		00H ^{Note}	
FFFFF40FH	Port 7 register H	P7H	√		√		00H ^{Note}	
FFFFF412H	Port 9 register	P9				√	0000H ^{Note}	
FFFFF412H	Port 9 register L	P9L	√		√		00H ^{Note}	
FFFFF413H	Port 9 register H	P9H	√		√		00H ^{Note}	
FFFFF420H	Port 0 mode register	PM0	√		√		FFH	

Note The output latch is 00H or 0000H. When these registers are input, the pin statuses are read.

(5/10)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF422H	Port 1 mode register	PM1	R/W	√	√		FFH
FFFFF426H	Port 3 mode register	PM3				√	FFFFH
FFFFF426H	Port 3 mode register L	PM3L		√	√		FFH
FFFFF427H	Port 3 mode register H	PM3H		√	√		FFH
FFFFF428H	Port 4 mode register	PM4		√	√		FFH
FFFFF42AH	Port 5 mode register	PM5		√	√		FFH
FFFFF42EH	Port 7 mode register L	PM7L		√	√		FFH
FFFFF42FH	Port 7 mode register H	PM7H		√	√		FFH
FFFFF432H	Port 9 mode register	PM9				√	FFFFH
FFFFF432H	Port 9 mode register L	PM9L		√	√		FFH
FFFFF433H	Port 9 mode register H	PM9H		√	√		FFH
FFFFF440H	Port 0 mode control register	PMC0		√	√		00H
FFFFF446H	Port 3 mode control register	PMC3				√	0000H
FFFFF446H	Port 3 mode control register L	PMC3L		√	√		00H
FFFFF447H	Port 3 mode control register H	PMC3H		√	√		00H
FFFFF448H	Port 4 mode control register	PMC4		√	√		00H
FFFFF44AH	Port 5 mode control register	PMC5		√	√		00H
FFFFF452H	Port 9 mode control register	PMC9				√	0000H
FFFFF452H	Port 9 mode control register L	PMC9L		√	√		00H
FFFFF453H	Port 9 mode control register H	PMC9H		√	√		00H
FFFFF460H	Port 0 function control register	PFC0		√	√		00H
FFFFF466H	Port 3 function control register	PFC3				√	0000H
FFFFF466H	Port 3 function control register L	PFC3L		√	√		00H
FFFFF467H	Port 3 function control register H	PFC3H		√	√		00H
FFFFF468H	Port 4 function control register	PFC4		√	√		00H
FFFFF46AH	Port 5 function control register	PFC5		√	√		00H
FFFFF472H	Port 9 function control register	PFC9				√	0000H
FFFFF472H	Port 9 function control register L	PFC9L	√	√		00H	
FFFFF473H	Port 9 function control register H	PFC9H	√	√		00H	
FFFFF540H	TMQ0 control register 0	TQ0CTL0	√	√		00H	
FFFFF541H	TMQ0 control register 1	TQ0CTL1	√	√		00H	
FFFFF542H	TMQ0 I/O control register 0	TQ0IOC0	√	√		00H	
FFFFF543H	TMQ0 I/O control register 1	TQ0IOC1	√	√		00H	
FFFFF544H	TMQ0 I/O control register 2	TQ0IOC2	√	√		00H	
FFFFF545H	TMQ0 option register 0	TQ0OPT0	√	√		00H	
FFFFF546H	TMQ0 capture/compare register 0	TQ0CCR0			√	0000H	
FFFFF548H	TMQ0 capture/compare register 1	TQ0CCR1			√	0000H	
FFFFF54AH	TMQ0 capture/compare register 2	TQ0CCR2			√	0000H	
FFFFF54CH	TMQ0 capture/compare register 3	TQ0CCR3			√	0000H	
FFFFF54EH	TMQ0 counter read buffer register	TQ0CNT	R		√	0000H	

Note The output latch is 00H or 0000H. When these registers are input, the pin statuses are read.

(6/10)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF590H	TMP0 control register 0	TP0CTL0	R/W	√	√		00H
FFFFF591H	TMP0 control register 1	TP0CTL1		√	√		00H
FFFFF592H	TMP0 I/O control register 0	TP0IOC0		√	√		00H
FFFFF593H	TMP0 I/O control register 1	TP0IOC1		√	√		00H
FFFFF594H	TMP0 I/O control register 2	TP0IOC2		√	√		00H
FFFFF595H	TMP0 option register 0	TP0OPT0		√	√		00H
FFFFF596H	TMP0 capture/compare register 0	TP0CCR0				√	0000H
FFFFF598H	TMP0 capture/compare register 1	TP0CCR1				√	0000H
FFFFF59AH	TMP0 counter read buffer register	TP0CNT	R			√	0000H
FFFFF5A0H	TMP1 control register 0	TP1CTL0	R/W	√	√		00H
FFFFF5A1H	TMP1 control register 1	TP1CTL1		√	√		00H
FFFFF5A2H	TMP1 I/O control register 0	TP1IOC0		√	√		00H
FFFFF5A3H	TMP1 I/O control register 1	TP1IOC1		√	√		00H
FFFFF5A4H	TMP1 I/O control register 2	TP1IOC2		√	√		00H
FFFFF5A5H	TMP1 option register 0	TP1OPT0		√	√		00H
FFFFF5A6H	TMP1 capture/compare register 0	TP1CCR0				√	0000H
FFFFF5A8H	TMP1 capture/compare register 1	TP1CCR1				√	0000H
FFFFF5AAH	TMP1 counter read buffer register	TP1CNT	R			√	0000H
FFFFF5B0H	TMP2 control register 0	TP2CTL0	R/W	√	√		00H
FFFFF5B1H	TMP2 control register 1	TP2CTL1		√	√		00H
FFFFF5B2H	TMP2 I/O control register 0	TP2IOC0		√	√		00H
FFFFF5B3H	TMP2 I/O control register 1	TP2IOC1		√	√		00H
FFFFF5B4H	TMP2 I/O control register 2	TP2IOC2		√	√		00H
FFFFF5B5H	TMP2 option register 0	TP2OPT0		√	√		00H
FFFFF5B6H	TMP2 capture/compare register 0	TP2CCR0				√	0000H
FFFFF5B8H	TMP2 capture/compare register 1	TP2CCR1				√	0000H
FFFFF5BAH	TMP2 counter read buffer register	TP2CNT	R			√	0000H
FFFFF5C0H	TMP3 control register 0	TP3CTL0	R/W	√	√		00H
FFFFF5C1H	TMP3 control register 1	TP3CTL1		√	√		00H
FFFFF5C2H	TMP3 I/O control register 0	TP3IOC0		√	√		00H
FFFFF5C3H	TMP3 I/O control register 1	TP3IOC1		√	√		00H
FFFFF5C5H	TMP3 option register 0	TP3OPT0		√	√		00H
FFFFF5C6H	TMP3 capture/compare register 0	TP3CCR0				√	0000H
FFFFF5C8H	TMP3 capture/compare register 1	TP3CCR1				√	0000H
FFFFF5CAH	TMP3 counter read buffer register	TP3CNT		R			√
FFFFF5D0H	TMP4 control register 0	TP4CTL0	R/W	√	√		00H
FFFFF5D1H	TMP4 control register 1	TP4CTL1		√	√		00H
FFFFF5D2H	TMP4 I/O control register 0	TP4IOC0		√	√		00H
FFFFF5D3H	TMP4 I/O control register 1	TP4IOC1		√	√		00H
FFFFF5D4H	TMP4 I/O control register 2	TP4IOC2		√	√		00H
FFFFF5D5H	TMP4 option register 0	TP4OPT0		√	√		00H

(7/10)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF5D6H	TMP4 capture/compare register 0	TP4CCR0	R/W			√	0000H
FFFFF5D8H	TMP4 capture/compare register 1	TP4CCR1				√	0000H
FFFFF5DAH	TMP4 counter read buffer register	TP4CNT	R			√	0000H
FFFFF5E0H	TMP5 control register 0	TP5CTL0	R/W	√	√		00H
FFFFF5E1H	TMP5 control register 1	TP5CTL1		√	√		00H
FFFFF5E2H	TMP5 I/O control register 0	TP5IOC0		√	√		00H
FFFFF5E3H	TMP5 I/O control register 1	TP5IOC1		√	√		00H
FFFFF5E4H	TMP5 I/O control register 2	TP5IOC2		√	√		00H
FFFFF5E5H	TMP5 option register 0	TP5OPT0		√	√		00H
FFFFF5E6H	TMP5 capture/compare register 0	TP5CCR0				√	0000H
FFFFF5E8H	TMP5 capture/compare register 1	TP5CCR1				√	0000H
FFFFF5EAH	TMP5 counter read buffer register	TP5CNT	R			√	0000H
FFFFF680H	Watch timer operation mode register	WTM	R/W	√	√		00H
FFFFF690H	TMM0 control register 0	TM0CTL0		√	√		00H
FFFFF694H	TMM0 compare register 0	TM0CMP0				√	0000H
FFFFF6C0H	Oscillation stabilization time select register	OSTS			√		06H
FFFFF6C1H	PLL lockup time specification register	PLLS			√		03H
FFFFF6D0H	Watchdog timer mode register 2	WDTM2			√		67H
FFFFF6D1H	Watchdog timer enable register	WDTE			√		9AH
FFFFF6E0H	Real-time output buffer register 0L	RTBL0		√	√		00H
FFFFF6E2H	Real-time output buffer register 0H	RTBH0		√	√		00H
FFFFF6E4H	Real-time output port mode register 0	RTPM0		√	√		00H
FFFFF6E5H	Real-time output port control register 0	RTPC0		√	√		00H
FFFFF700H	Port 0 function control expansion register	PFCE0		√	√		00H
FFFFF706H	Port 3 function control expansion register L	PFCE3L		√	√		00H
FFFFF70AH	Port 5 function control expansion register	PFCE5		√	√		00H
FFFFF712H	Port 9 function control expansion register	PFCE9				√	0000H
FFFFF712H	Port 9 function control expansion register L	PFCE9L		√	√		00H
FFFFF713H	Port 9 function control expansion register H	PFCE9H		√	√		00H
FFFFF802H	System status register	SYS		√	√		00H
FFFFF80CH	Internal oscillation mode register	RCM		√	√		00H
FFFFF810H	DMA trigger factor register 0	DTFR0		√	√		00H
FFFFF812H	DMA trigger factor register 1	DTFR1		√	√		00H
FFFFF814H	DMA trigger factor register 2	DTFR2		√	√		00H
FFFFF816H	DMA trigger factor register 3	DTFR3		√	√		00H
FFFFF820H	Power save mode register	PSMR		√	√		00H
FFFFF822H	Clock control register	CKC		√	√		0AH
FFFFF824H	Lock register	LOCKR	R	√	√		00H
FFFFF828H	Processor clock control register	PCC ^{Note}	R/W	√	√		03H
FFFFF82CH	PLL control register	PLLCTL		√	√		01H
FFFFF82EH	CPU operation clock status register	CCLS	R	√	√		00H

Note This is a special register.

(8/10)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFF870H	Clock monitor mode register	CLM	R/W	√	√		00H
FFFFF888H	Reset source flag register	RESF		√	√		00H
FFFFF890H	Low-voltage detection register	LVIM		√	√		00H
FFFFF891H	Low-voltage detection level select register	LVIS			√		00H
FFFFF8B0H	Prescaler mode register 0	PRSM0		√	√		00H
FFFFF8B1H	Prescaler compare register 0	PRSCM0			√		00H
FFFFF9FCH	On-chip debug mode register	OCDM ^{Note}		√	√		01H
FFFFFA00H	UARTA0 control register 0	UA0CTL0		√	√		10H
FFFFFA01H	UARTA0 control register 1	UA0CTL1			√		00H
FFFFFA02H	UARTA0 control register 2	UA0CTL2			√		FFH
FFFFFA03H	UARTA0 option control register 0	UA0OPT0		√	√		14H
FFFFFA04H	UARTA0 status register	UA0STR		√	√		00H
FFFFFA06H	UARTA0 receive data register	UA0RX	R		√		FFH
FFFFFA07H	UARTA0 transmit data register	UA0TX	R/W		√		FFH
FFFFFA10H	UARTA1 control register 0	UA1CTL0		√	√		10H
FFFFFA11H	UARTA1 control register 1	UA1CTL1			√		00H
FFFFFA12H	UARTA1 control register 2	UA1CTL2			√		FFH
FFFFFA13H	UARTA1 option control register 0	UA1OPT0		√	√		14H
FFFFFA14H	UARTA1 status register	UA1STR		√	√		00H
FFFFFA16H	UARTA1 receive data register	UA1RX	R		√		FFH
FFFFFA17H	UARTA1 transmit data register	UA1TX	R/W		√		FFH
FFFFFA20H	UARTA2 control register 0	UA2CTL0		√	√		10H
FFFFFA21H	UARTA2 control register 1	UA2CTL1			√		00H
FFFFFA22H	UARTA2 control register 2	UA2CTL2			√		FFH
FFFFFA23H	UARTA2 option control register 0	UA2OPT0		√	√		14H
FFFFFA24H	UARTA2 status register	UA2STR		√	√		00H
FFFFFA26H	UARTA2 receive data register	UA2RX	R		√		FFH
FFFFFA27H	UARTA2 transmit data register	UA2TX	R/W		√		FFH
FFFFFAD0H	Sub-count register	RC1SUBC	R			√	0000H
FFFFFAD2H	Second count register	RC1SEC	R/W		√		00H
FFFFFAD3H	Minute count register	RC1MIN			√		00H
FFFFFAD4H	Hour count register	RC1HOUR			√		12H
FFFFFAD5H	Week count register	RC1WEEK			√		00H
FFFFFAD6H	Day count register	RC1DAY			√		01H
FFFFFAD7H	Month count register	RC1MONTH			√		01H
FFFFFAD8H	Year count register	RC1YEAR			√		00H
FFFFFAD9H	Time error correction register	RC1SUBU			√		00H
FFFFFADAH	Alarm minute set register	RC1ALM			√		00H
FFFFFADBH	Alarm time set register	RC1ALH			√		12H
FFFFFADCH	Alarm week set register	RC1ALW		√	√		00H

Note This is a special register.

(9/10)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFADDH	RTC control register 0	RC1CC0	R/W	√	√		00H
FFFFFADEH	RTC control register 1	RC1CC1		√	√		00H
FFFFFADFH	RTC control register 2	RC1CC2		√	√		00H
FFFFFAE0H	RTC control register 3	RC1CC3		√	√		00H
FFFFFC00H	External interrupt falling edge specification register 0	INTF0		√	√		00H
FFFFFC06H	External interrupt falling edge specification register 3	INTF3		√	√		00H
FFFFFC13H	External interrupt falling edge specification register 9H	INTF9H		√	√		00H
FFFFFC20H	External interrupt rising edge specification register 0	INTR0		√	√		00H
FFFFFC26H	External interrupt rising edge specification register 3	INTR3		√	√		00H
FFFFFC33H	External interrupt rising edge specification register 9H	INTR9H		√	√		00H
FFFFFC60H	Port 0 function register	PF0		√	√		00H
FFFFFC66H	Port 3 function register	PF3			√	0000H	
FFFFFC66H	Port 3 function register L	PF3L	√	√		00H	
FFFFFC67H	Port 3 function register H	PF3H	√	√		00H	
FFFFFC68H	Port 4 function register	PF4	√	√		00H	
FFFFFC6AH	Port 5 function register	PF5	√	√		00H	
FFFFFC72H	Port 9 function register	PF9			√	0000H	
FFFFFC72H	Port 9 function register L	PF9L	√	√		00H	
FFFFFC73H	Port 9 function register H	PF9H	√	√		00H	
FFFFFD00H	CSIB0 control register 0	CB0CTL0	√	√		01H	
FFFFFD01H	CSIB0 control register 1	CB0CTL1	√	√		00H	
FFFFFD02H	CSIB0 control register 2	CB0CTL2		√		00H	
FFFFFD03H	CSIB0 status register	CB0STR	√	√		00H	
FFFFFD04H	CSIB0 receive data register	CB0RX	R		√	0000H	
FFFFFD04H	CSIB0 receive data register L	CB0RXL		√		00H	
FFFFFD06H	CSIB0 transmit data register	CB0TX	R/W		√	0000H	
FFFFFD06H	CSIB0 transmit data register L	CB0TXL			√		00H
FFFFFD10H	CSIB1 control register 0	CB1CTL0	√	√		01H	
FFFFFD11H	CSIB1 control register 1	CB1CTL1	√	√		00H	
FFFFFD12H	CSIB1 control register 2	CB1CTL2		√		00H	
FFFFFD13H	CSIB1 status register	CB1STR	√	√		00H	
FFFFFD14H	CSIB1 receive data register	CB1RX	R		√	0000H	
FFFFFD14H	CSIB1 receive data register L	CB1RXL		√		00H	
FFFFFD16H	CSIB1 transmit data register	CB1TX	R/W		√	0000H	
FFFFFD16H	CSIB1 transmit data register L	CB1TXL			√		00H
FFFFFD20H	CSIB2 control register 0	CB2CTL0	√	√		01H	
FFFFFD21H	CSIB2 control register 1	CB2CTL1	√	√		00H	
FFFFFD22H	CSIB2 control register 2	CB2CTL2		√		00H	
FFFFFD23H	CSIB2 status register	CB2STR	√	√		00H	
FFFFFD24H	CSIB2 receive data register	CB2RX	R		√	0000H	
FFFFFD24H	CSIB2 receive data register L	CB2RXL		√		00H	

(10/10)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value
				1	8	16	
FFFFFD26H	CSIB2 transmit data register	CB2TX	R/W			√	0000H
FFFFFD26H	CSIB2 transmit data register L	CB2TXL			√		00H
FFFFFD30H	CSIB3 control register 0	CB3CTL0		√	√		01H
FFFFFD31H	CSIB3 control register 1	CB3CTL1		√	√		00H
FFFFFD32H	CSIB3 control register 2	CB3CTL2			√		00H
FFFFFD33H	CSIB3 status register	CB3STR		√	√		00H
FFFFFD34H	CSIB3 receive data register	CB3RX	R			√	0000H
FFFFFD34H	CSIB3 receive data register L	CB3RXL			√		00H
FFFFFD36H	CSIB3 transmit data register	CB3TX	R/W			√	0000H
FFFFFD36H	CSIB3 transmit data register L	CB3TXL			√		00H
FFFFFD40H	CSIB4 control register 0	CB4CTL0		√	√		01H
FFFFFD41H	CSIB4 control register 1	CB4CTL1		√	√		00H
FFFFFD42H	CSIB4 control register 2	CB4CTL2			√		00H
FFFFFD43H	CSIB4 status register	CB4STR		√	√		00H
FFFFFD44H	CSIB4 receive data register	CB4RX	R			√	0000H
FFFFFD44H	CSIB4 receive data register L	CB4RXL			√		00H
FFFFFD46H	CSIB4 transmit data register	CB4TX	R/W			√	0000H
FFFFFD46H	CSIB4 transmit data register L	CB4TXL			√		00H
FFFFFD80H	IIC shift register 0	IIC0				√	00H
FFFFFD82H	IIC control register 0	IICC0		√	√		00H
FFFFFD83H	Slave address register 0	SVA0			√		00H
FFFFFD84H	IIC clock select register 0	IICCL0		√	√		00H
FFFFFD85H	IIC function expansion register 0	IICX0	√	√		00H	
FFFFFD86H	IIC status register 0	IICS0	R	√	√		00H
FFFFFD8AH	IIC flag register 0	IICF0	R/W	√	√		00H
FFFFFD90H	IIC shift register 1	IIC1				√	00H
FFFFFD92H	IIC control register 1	IICC1		√	√		00H
FFFFFD93H	Slave address register 1	SVA1			√		00H
FFFFFD94H	IIC clock select register 1	IICCL1		√	√		00H
FFFFFD95H	IIC function expansion register 1	IICX1		√	√		00H
FFFFFD96H	IIC status register 1	IICS1	R	√	√		00H
FFFFFD9AH	IIC flag register 1	IICF1	R/W	√	√		00H
FFFFFDA0H	IIC shift register 2	IIC2				√	00H
FFFFFDA2H	IIC control register 2	IICC2		√	√		00H
FFFFFDA3H	Slave address register 2	SVA2			√		00H
FFFFFDA4H	IIC clock select register 2	IICCL2		√	√		00H
FFFFFDA5H	IIC function expansion register 2	IICX2		√	√		00H
FFFFFDA6H	IIC status register 2	IICS2	R	√	√		00H
FFFFDAAH	IIC flag register 2	IICF2	R/W	√	√		00H

3.4.7 Special registers

Special registers are registers that are protected from being written with illegal data due to a program hang-up. The V850ES/JC3-L and V850ES/JE3-L have the following seven special registers.

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

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

(1) Setting data to special registers

Set data to the special registers in the following sequence.

- <1> Disable DMA operation.
- <2> Prepare data to be set to the special register in a general-purpose register.
- <3> Write the data prepared in <2> to the PRCMD register.
- <4> Write the setting data to the special register (by using the following instructions).
 - Store instruction (ST/SST instruction)
 - Bit manipulation instruction (SET1/CLR1/NOT1 instruction)
- (<5> to <9> Insert NOP instructions (5 instructions).)^{Note}
- <10> Enable DMA operation if necessary.

Note When switching to the IDLE mode or the STOP mode (PSC.STP bit = 1), 5 NOP instructions must be inserted immediately after switching is performed.

Caution To resume the DMA operation in the status before the DMA operation was disabled after a special sequence, the DCHCn register status must be stored before the DMA operation is disabled. After the DCHCn register status is stored, the DCHCn.TCn bit must be checked before the DMA operation is resumed and the following processing must be executed according to the TCn bit status, because completion of DMA transfer may occur before the DMA operation is disabled.

- When the TCn bit is 0 (DMA transfer not completed), the contents of the DCHCn register stored before the DMA operation was disabled are written to the DCHCn register again.
- When the TCn bit is 1 (DMA transfer completed), DMA transfer completion processing is executed.

Remark n = 0 to 3

[Example] PSC register (setting standby mode)

```

    ST.B r11, PSMR[r0] ; Set PSMR register (setting IDLE1, IDLE2, and STOP modes).
<1>CLR1 0, DCHCn[r0] ; Disable DMA operation. n = 0 to 3
<2>MOV0x02, r10
<3>ST.B r10, PRCMD[r0] ; Write PRCMD register.
<4>ST.B r10, PSC[r0] ; Set PSC register.
<5>NOPNote ; Dummy instruction
<6>NOPNote ; Dummy instruction
<7>NOPNote ; Dummy instruction
<8>NOPNote ; Dummy instruction
<9>NOPNote ; Dummy instruction
<10>SET1 0, DCHCn[r0] ; Enable DMA operation. n = 0 to 3
(next instruction)

```

There is no special sequence required to read a special register.

Note Five NOP instructions or more must be inserted immediately after setting the IDLE1 mode, IDLE2 mode, or STOP mode (by setting the PSC.STP bit to 1).

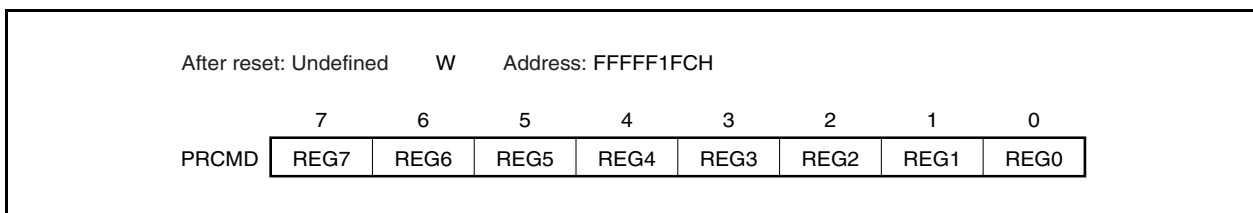
Caution When a store instruction is executed to store data in the command register, interrupts are not acknowledged. This is because it is assumed that steps <3> and <4> above are performed by successive store instructions. If another instruction is placed between <3> and <4>, and if an interrupt is acknowledged by that instruction, the above sequence may not be established, causing malfunction.

Remark Although dummy data is written to the PRCMD register, use the same general-purpose register used to set the special register (<4> in the example) to write data to the PRCMD register (<3> in the example). The same applies when a general-purpose register is used for addressing.

(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 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).



(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 sets this register to 00H.

After reset: 00H	R/W	Address: FFFFF802H							
		7	6	5	4	3	2	1	<0>
SYS		0	0	0	0	0	0	0	PRERR
	PRERR	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 <4> is executed without executing <3> in **3.4.7 (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 <4> in **3.4.7 (1) Setting data to special registers** is not the setting of a special register)

Remark Even if an on-chip peripheral I/O register is read (except by a bit manipulation instruction) or the internal RAM is accessed between an operation to write the PRCMD register and an operation to write a special register, the PRERR flag is not set, and the set 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.

3.4.8 Registers to be set first

Be sure to set the following registers first when using the V850ES/JC3-L, V850ES/JE3-L.

- System wait control register (VSWC)
- On-chip debug mode register (OCDM)
- Watchdog timer mode register 2 (WDTM2)

After setting the VSWC, OCDM, and WDTM2 registers, set the other registers as necessary.

(a) System wait control register (VSWC)

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

Three clock cycles are required to access an on-chip peripheral I/O register (without a wait cycle). The V850ES/JC3-L, V850ES/JE3-L require wait cycles according to the operating frequency. Set the following value to the VSWC register in accordance with the frequency used.

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

Reset sets this register to 77H (number of waits: 14).

After reset: 77H		R/W	Address: FFFFF06EH					
	7	6	5	4	3	2	1	0
VSWC								
Operating Frequency (f_{CLK})		Set Value of VSWC			Number of Waits			
$32 \text{ kHz} \leq f_{CLK} < 16.6 \text{ MHz}$		00H			0 (no waits)			
$16.6 \text{ MHz} \leq f_{CLK} \leq 20 \text{ MHz}$		01H			1			

(b) On-chip debug mode register (OCDM)

For details, see **CHAPTER 29 ON-CHIP DEBUG FUNCTION**.

(c) Watchdog timer mode register 2 (WDTM2)

The WDTM2 register sets the overflow time and the operation clock of watchdog timer 2.

Watchdog timer 2 automatically starts in the reset mode after reset is released. To specify the operation of watchdog timer 2, write to the WDTM2 register after reset is released.

For details, see **CHAPTER 11 WATCHDOG TIMER 2**.

3.4.9 Cautions

(1) Accessing special on-chip peripheral I/O registers

This product has two types of internal system buses.

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

The clock of the CPU bus and the clock of the peripheral bus are asynchronous. If an access to the CPU and an access to the peripheral hardware conflict, therefore, unexpected illegal data may be transferred. If there is a possibility of a conflict, the number of cycles for accessing the CPU changes when the peripheral hardware is accessed, so that correct data is transferred. As a result, the CPU does not start processing of the next instruction but enters the wait status. If this wait status occurs, the number of clocks required to execute an instruction increases by the number of wait clocks shown below.

This must be taken into consideration if real-time processing is required.

When special on-chip peripheral I/O registers are accessed, more wait states may be required in addition to the wait states set by the VSWC register.

The access conditions and how to calculate the number of wait states to be inserted (number of CPU clocks) at this time are shown below.

Table 3-3. Registers That Requires Waits

Peripheral Function	Register Name	Access	k
16-bit timer/event counter P (TMP) (n = 0 to 5)	TPnCNT	Read	1 or 2
	TPnCCR0, TPnCCR1	Write	<ul style="list-style-type: none"> • 1st access: No wait • Continuous write: 3 or 4
		Read	1 or 2
16-bit timer/event counter Q (TMQ)	TQ0CNT	Read	1 or 2
	TQ0CCR0 to TQ0CCR3	Write	<ul style="list-style-type: none"> • 1st access: No wait • Continuous write: 3 or 4
		Read	1 or 2
Watchdog timer 2 (WDT2)	WDTM2	Write (when WDT2 operating)	3
Real-time output function (RTO)	RTBL0, RTBH0	Write (RTPC0.RTPOE0 bit = 0)	1
A/D converter	ADA0M0	Read	1 or 2
	ADA0CR0 to ADA0CR9	Read	1 or 2
	ADA0CR0H to ADA0CR9H	Read	1 or 2
I ² C00 to I ² C02	IICS0 to IICS2	Read	1
CRC	CRCD	Write	1

Number of clocks necessary for access = $3 + i + j + (2 + j) \times k$

Caution Accessing the above registers is prohibited in the following statuses. If a wait cycle is generated, it can only be cleared by a reset.

- When the CPU operates on the subclock and main clock oscillation is stopped
- When the CPU operates on the internal oscillator clock

Remark i: Value (0) of higher 4 bits of VSWC register
j: Value (0 or 1) of lower 4 bits of VSWC register

(2) 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
 •
 •
 •

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.

<ii> mov r10, r28
 <iii> sld.w 0x28, r10

(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

4.1.1 V850ES/JC3-L (40-pin)

- I/O port pins: 27
 - N-ch open-drain output selectable: 20 (5 V tolerant: 17)
- Input/output specifiable in 1-bit units

4.1.2 V850ES/JC3-L (48-pin)

- I/O port pins: 34
 - N-ch open-drain output selectable: 25 (5 V tolerant: 20)
- Input/output specifiable in 1-bit units

4.1.3 V850ES/JE3-L

- I/O port pins: 50
 - N-ch open-drain output selectable: 37 (5 V tolerant: 28)
- Input/output specifiable in 1-bit units

4.2 Basic Port Configuration

4.2.1 V850ES/JC3-L (40-pin)

The V850ES/JC3-L (40-pin) features a total of 27 I/O port pins organized as ports 0, 3 to 5, 7, 9, CM, and DL. The port configuration is shown below.

Figure 4-1. Port Configuration (V850ES/JC3-L (40-pin))

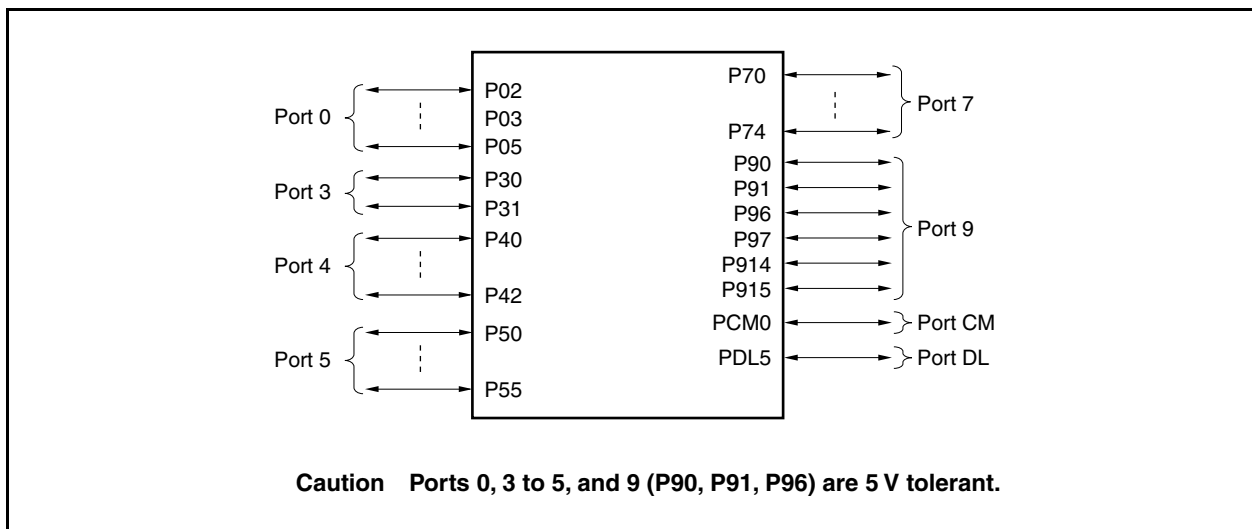


Table 4-1. I/O Buffer Power Supplies for Pins (V850ES/JC3-L (40-pin))

Power Supply	Corresponding Pins
AV _{REF0}	Port 7
EV _{DD}	$\overline{\text{RESET}}$, ports 0, 3 to 5, 9, CM, DL

4.2.2 V850ES/JC3-L (48-pin)

The V850ES/JC3-L (48-pin) features a total of 34 I/O port pins organized as ports 0, 1, 3 to 5, 7, 9, CM, and DL. The port configuration is shown below.

Figure 4-2. Port Configuration (V850ES/JC3-L (48-pin))

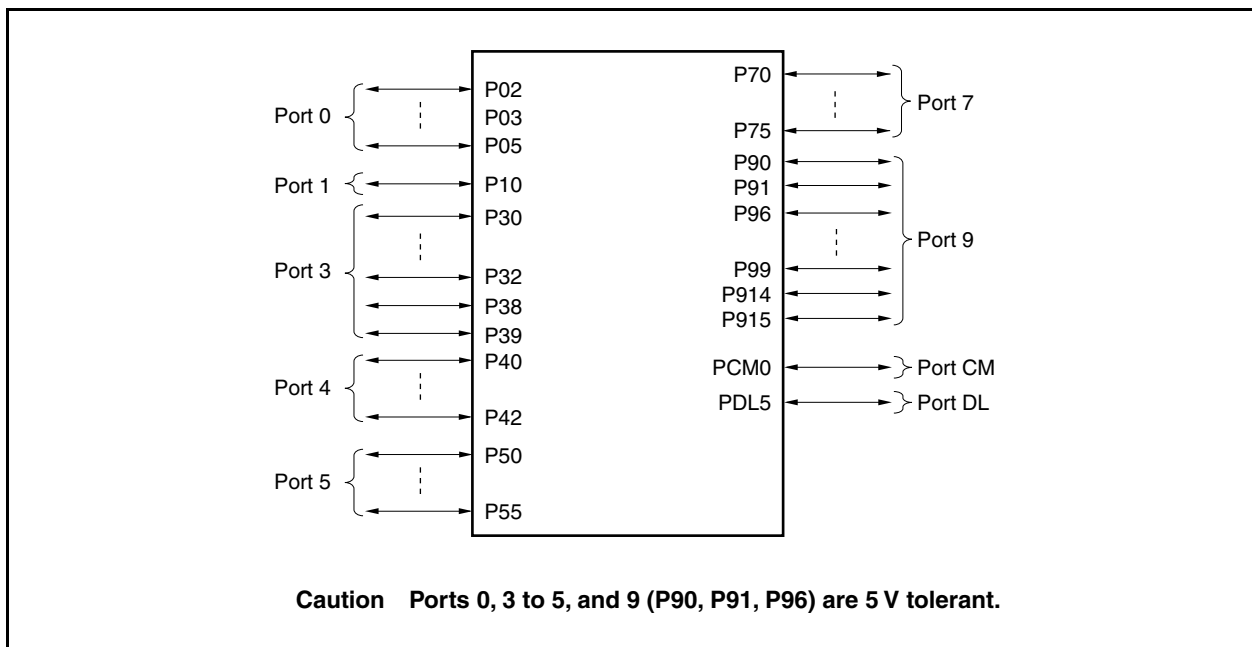


Table 4-2. I/O Buffer Power Supplies for Pins (V850ES/JC3-L (48-pin))

Power Supply	Corresponding Pins
AV _{REF0}	Port 7
AV _{REF1}	Port 1
EV _{DD}	$\overline{\text{RESET}}$, ports 0, 3 to 5, 9, CM, DL

4.2.3 V850ES/JE3-L

The V850ES/JE3-L features a total of 50 I/O port pins organized as ports 0, 1, 3 to 5, 7, 9, CM, and DL. The port configuration is shown below.

Figure 4-3. Port Configuration (V850ES/JE3-L)

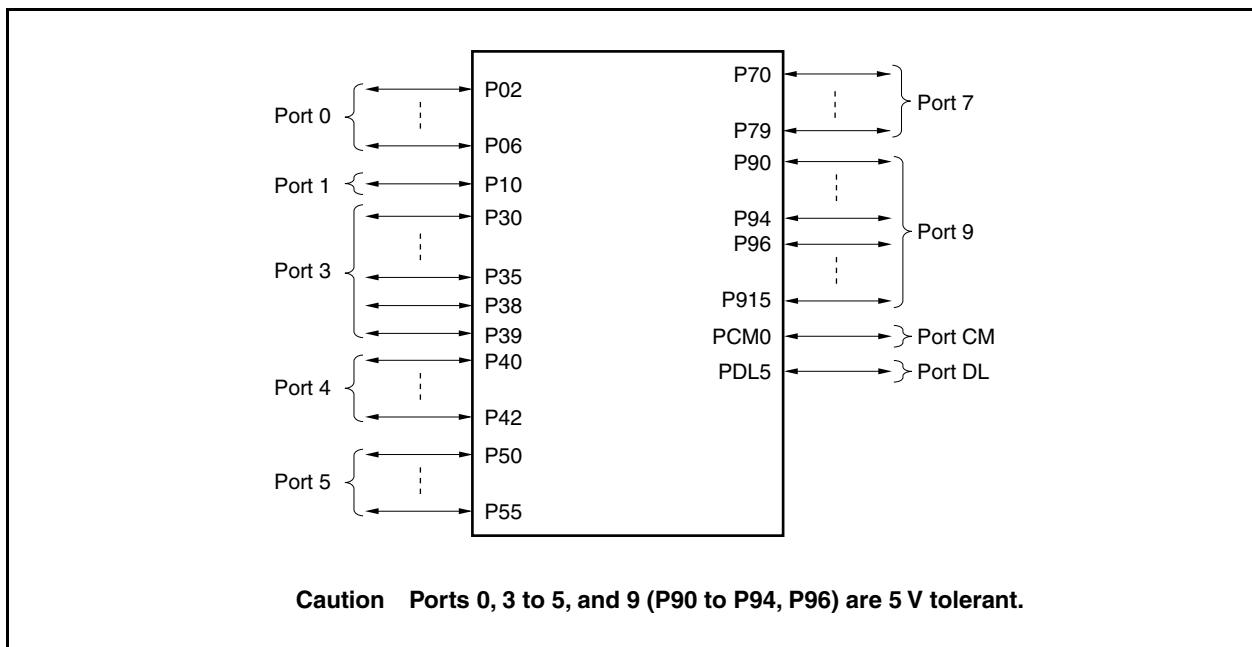


Table 4-3. I/O Buffer Power Supplies for Pins (V850ES/JE3-L)

Power Supply	Corresponding Pins
AV _{REF0}	Port 7
AV _{REF1}	Port 1
EV _{DD}	$\overline{\text{RESET}}$, ports 0, 3 to 5, 9, CM, DL

4.3 Port Configuration

The ports consist of the following hardware.

Table 4-4. Port Configuration (V850ES/JC3-L (40-pin))

Item	Configuration
Control registers	Port n mode register (PMn: n = 0, 3 to 5, 7, 9, CM, DL) Port n mode control register (PMCn: n = 0, 3 to 5, 9) Port n function control register (PFCn: n = 0, 3 to 5, 9) Port n function control expansion register (PFCEn: n = 0, 5, 9) Port n function register (PFn: n = 0, 3 to 5, 9)
Port pins	I/O: 34

Table 4-5. Port Configuration (V850ES/JC3-L (48-pin))

Item	Configuration
Control registers	Port n mode register (PMn: n = 0, 1, 3 to 5, 7, 9, CM, DL) Port n mode control register (PMCn: n = 0, 3 to 5, 9) Port n function control register (PFCn: n = 0, 3 to 5, 9) Port n function control expansion register (PFCEn: n = 0, 3, 5, 9) Port n function register (PFn: n = 0, 3 to 5, 9)
Port pins	I/O: 34

Table 4-6. Port Configuration (V850ES/JE3-L)

Item	Configuration
Control registers	Port n mode register (PMn: n = 0, 1, 3 to 5, 7, 9, CM, DL) Port n mode control register (PMCn: n = 0, 3 to 5, 9) Port n function control register (PFCn: n = 0, 3 to 5, 9) Port n function control expansion register (PFCEn: n = 0, 3, 5, 9) Port n function register (PFn: n = 0, 3 to 5, 9)
Port pins	I/O: 50

(1) Port n register (Pn)

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

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: 00H ^{Note} (output latch)		R/W																
Pn	<table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>Pn7</td><td>Pn6</td><td>Pn5</td><td>Pn4</td><td>Pn3</td><td>Pn2</td><td>Pn1</td><td>Pn0</td> </tr> </table>	7	6	5	4	3	2	1	0	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0	
7	6	5	4	3	2	1	0											
Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0											
Pnm	Control of output data (in output mode)																	
0	0 is output.																	
1	1 is output.																	

Note This value is undefined for input-only ports.

The operation when writing or reading the Pn register differs depending on the specified mode.

Table 4-7. Reading and Writing Pn Register

PMCn Register Setting	PMn Register Setting	Writing Pn Register	Reading Pn Register
Port mode (PMCnm bit = 0)	Output mode (PMnm bit = 0)	Write to the output latch ^{Note} . The contents of the output latch are output from the pin.	The value of the output latch is read.
	Input mode (PMnm bit = 1)	Write to the output latch ^{Note} . The status of the pin is not affected.	The pin status is read.
Alternate-function mode (PMCnm bit = 1)	Output mode (PMnm bit = 0)	Write to the output latch ^{Note} . The status of the pin is not affected. The pin operates as an alternate-function pin.	The value of the output latch is read.
	Input mode (PMnm bit = 1)	Write to the output latch ^{Note} . The status of the pin is not affected. The pin operates as an alternate-function pin.	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.
The output latch value is cleared by a reset.

(2) Port n mode register (PMn)

PMn specifies the input mode or output mode of the port.

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

After reset: FFH		R/W						
	7	6	5	4	3	2	1	0
PMn	PMn7	PMn6	PMn5	PMn4	PMn3	PMn2	PMn1	PMn0
	PMnm	Control of I/O mode						
	0	Output mode						
	1	Input mode						

(3) Port n mode control register (PMCn)

If the port function and the alternate function need to be switched, specify the port mode or the alternate function mode by using this register.

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

After reset: 00H		R/W						
	7	6	5	4	3	2	1	0
PMCn	PMCn7	PMCn6	PMCn5	PMCn4	PMCn3	PMCn2	PMCn1	PMCn0
	PMCnm	Specification of operation mode						
	0	Port mode						
	1	Alternate function mode						

(4) Port n function control register (PFCn)

PFCn is a register that specifies the alternate function to be used when one pin has two or more alternate functions. Each bit of the PFCn register corresponds to one pin of port n and can be specified in 1-bit units.

After reset: 00H		R/W						
	7	6	5	4	3	2	1	0
PFCn	PFCn7	PFCn6	PFCn5	PFCn4	PFCn3	PFCn2	PFCn1	PFCn0
PFCnm	Specification of alternate function							
0	Alternate function 1							
1	Alternate function 2							

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

The PFCEn register specifies the alternate function of a port pin in combination with the PFCn register if the pin has three or more alternate functions.

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

After reset: 00H		R/W						
	7	6	5	4	3	2	1	0
PFCEn	PFCEn7	PFCEn6	PFCEn5	PFCEn4	PFCEn3	PFCEn2	PFCEn1	PFCEn0
	7	6	5	4	3	2	1	0
PFCn	PFCn7	PFCn6	PFCn5	PFCn4	PFCn3	PFCn2	PFCn1	PFCn0
PFCEnm	PFCnm	Specification of alternate function						
0	0	Alternate function 1						
0	1	Alternate function 2						
1	0	Alternate function 3						
1	1	Alternate function 4						

(6) Port n function register (PFn)

PFn is a register that specifies normal output (CMOS output) or N-ch open-drain output.

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

After reset: 00H R/W

	7	6	5	4	3	2	1	0
PFn	PFn7	PFn6	PFn5	PFn4	PFn3	PFn2	PFn1	PFn0

PFnm ^{Note}	Specification of normal output (CMOS output)/N-ch open-drain output
0	Normal output (CMOS output)
1	N-ch open-drain output

Note Regardless of the settings of the PMCn register, the PFnm bit is valid only if the PMn.PMnm bit is set to 0 (output mode). If the PMnm bit is set to 1 (input mode), the values specified for the PFn register are invalid.

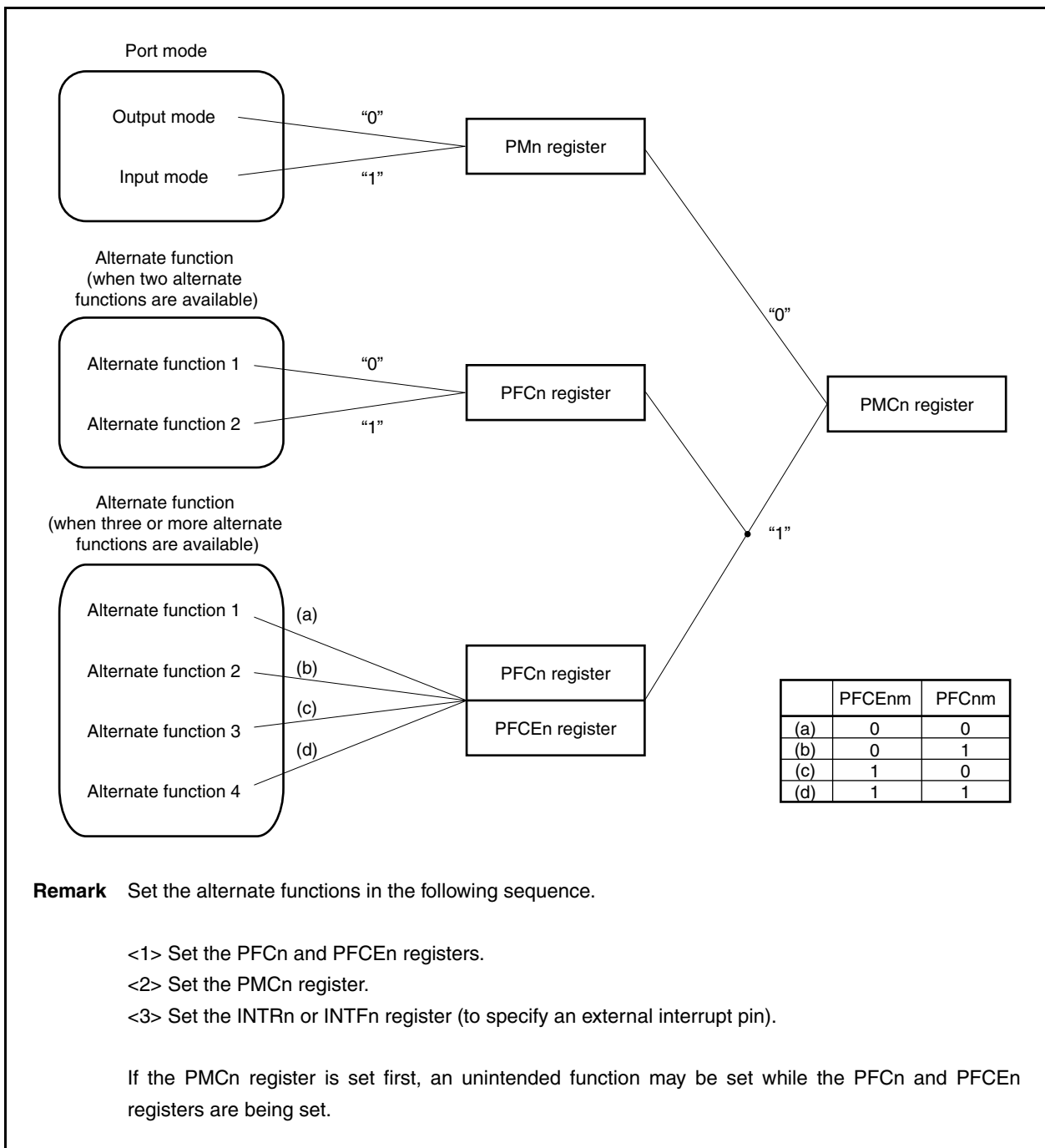
Example <1> The PFn register values are valid when:
 PFnm bit = 1 ... N-ch open drain output is specified.
 PMnm bit = 0 ... Output mode is specified.
 PMCnm bit = Any value

<2> The PFn register values are invalid when:
 PFnm bit = 1 ... N-ch open drain output is specified.
 PMnm bit = 1 ... Input mode is specified.
 PMCnm bit = Any value

(7) Port setting

Set a port as illustrated below.

Figure 4-4. Setting of Each Register and Pin Function



4.3.1 Port 0

Port 0 is a 5-bit port (V850ES/JE3-L) or a 3-bit port (V850ES/JC3-L (40-pin), V850ES/JC3-L (48-pin)) for which I/O settings can be controlled in 1-bit units.

Port 0 includes the following alternate-function pins.

Table 4-8. Port 0 Alternate-Function Pins

Pin No.			Function Name	Alternate Function		Remark	Block Type
JC3L (40)	JC3L (48)	JE3L		Name	I/O		
11	13	17	P02	NMI	Input	Selectable as N-ch open-drain output	L-1
12	14	18	P03	INTP0/ADTRG/RTC1HZ	Input/Output		U-15
–	–	19	P04	INTP1/RTCDIV/RTCCL	Input/Output		N-2
27	31	41	P05	INTP2/ $\overline{\text{DRST}}$ ^{Note}	Input		AA-1
–	–	20	P06	INTP3	Input		L-1

Note The $\overline{\text{DRST}}$ pin is used for on-chip debugging.

If on-chip debugging is not used, fix the P05/INTP2/ $\overline{\text{DRST}}$ pin to low level between when the reset signal of the $\overline{\text{RESET}}$ pin is released and when the OCDM.OCDM0 bit is cleared (0).

For details, see **4.6.3 Cautions on on-chip debug pins**.

Caution The P02 to P06 pins have hysteresis characteristics in the input mode of the alternate function, but do not have hysteresis characteristics in the port mode.

Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

(1) Port 0 register (P0)

(a) V850ES/JC3-L (40-pin), V850ES/JC3-L (48-pin)

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

	7	6	5	4	3	2	1	0
P0	0	0	P05	0	P03	P02	0	0

P0n	Output data control (in output mode) (n = 2, 3, 5)
0	Outputs 0
1	Outputs 1

(b) V850ES/JE3-L

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

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

P0n	Output data control (in output mode) (n = 2 to 6)
0	Outputs 0
1	Outputs 1

(2) Port 0 mode register (PM0)

(a) V850ES/JC3-L (40-pin), V850ES/JC3-L (48-pin)

After reset: FFH R/W Address: FFFFF420H

	7	6	5	4	3	2	1	0
PM0	1	1	PM05	1	PM03	PM02	1	1

PM0n	I/O mode control (n = 2, 3, 5)
0	Output mode
1	Input mode

(b) V850ES/JE3-L

After reset: FFH R/W Address: FFFFF420H

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

PM0n	I/O mode control (n = 2 to 6)
0	Output mode
1	Input mode

(3) Port 0 mode control register (PMC0)**(a) V850ES/JC3-L (40-pin), V850ES/JC3-L (48-pin)**

After reset: 00H R/W Address: FFFFF440H

	7	6	5	4	3	2	1	0
PMC0	0	0	PMC05	0	PMC03	PMC02	0	0
PMC05	Specification of pin operation							
0	I/O port (P05)							
1	INTP2 input							
PMC03	Specification of pin operation							
0	I/O port (P03)							
1	INTP0 input/ADTRG input/RTC1HZ output							
PMC02	Specification of pin operation							
0	I/O port (P02)							
1	NMI input							

(b) V850ES/JE3-L

After reset: 00H R/W Address: FFFFF440H

	7	6	5	4	3	2	1	0
PMC0	0	PMC06	PMC05	PMC04	PMC03	PMC02	0	0
PMC06	Specification of pin operation							
0	I/O port (P06)							
1	INTP3 input							
PMC05	Specification of pin operation							
0	I/O port (P05)							
1	INTP2 input							
PMC04	Specification of pin operation							
0	I/O port (P04)							
1	INTP1 input/RTCDIV output/RTCCL output							
PMC03	Specification of pin operation							
0	I/O port (P03)							
1	INTP0 input/ADTRG input/RTC1HZ output							
PMC02	Specification of pin operation							
0	I/O port (P02)							
1	NMI input							

Caution The P05/INTP2/ $\overline{\text{DRST}}$ pin becomes the $\overline{\text{DRST}}$ pin regardless of the value of the PMC05 bit when the OCDM.OCDM0 bit is 1.

(4) Port 0 function control register (PFC0)**(a) V850ES/JC3-L (40-pin), V850ES/JC3-L (48-pin)**

After reset: 00H R/W Address: FFFFF460H

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

(b) V850ES/JE3-L

After reset: 00H R/W Address: FFFFF460H

	7	6	5	4	3	2	1	0
PFC0	0	0	0	PFC04	PFC03	0	0	0

Remark For details of alternate function specification, see **4.3.1 (6) Port 0 alternate function specifications.**

(5) Port 0 function control expansion register (PFCE0)**(a) V850ES/JC3-L (40-pin), V850ES/JC3-L (48-pin)**

After reset: 00H R/W Address: FFFFF700H

	7	6	5	4	3	2	1	0
PFCE0	0	0	0	0	PFCE03	0	0	0

(b) V850ES/JE3-L

After reset: 00H R/W Address: FFFFF700H

	7	6	5	4	3	2	1	0
PFCE0	0	0	0	PFCE04	PFCE03	0	0	0

Remark For details of alternate function specification, see **4.3.1 (6) Port 0 alternate function specifications.**

(6) Port 0 alternate function specifications

PFCE04 ^{Note}	PFC04 ^{Note}	Specification of P04 pin alternate function ^{Note}
0	0	INTP1 input
0	1	RTCDIV output
1	0	RTCCCL output
1	1	Setting prohibited

Note V850ES/JE3-L only

PFCE03	PFC03	Specification of P03 pin alternate function
0	0	INTP0 input
0	1	ADTRG input
1	0	Setting prohibited
1	1	RTC1HZ output

(7) Port 0 function register (PF0)

(a) V850ES/JC3-L (40-pin), V850ES/JC3-L (48-pin)

After reset: 00H R/W Address: FFFFFFFC60H

	7	6	5	4	3	2	1	0
PF0	0	0	PF05	0	PF03	PF02	0	0

PF0n	Specification of normal output (CMOS output) or N-ch open-drain output (n = 2, 3, 5)
0	Normal output (CMOS output)
1	N-ch open-drain output

(b) V850ES/JE3-L

After reset: 00H R/W Address: FFFFFFFC60H

	7	6	5	4	3	2	1	0
PF0	0	PF06	PF05	PF04	PF03	PF02	0	0

PF0n	Specification of normal output (CMOS output) or N-ch open-drain output (n = 2 to 6)
0	Normal output (CMOS output)
1	N-ch open-drain output

Caution When an output pin is pulled up to EV_{DD} or higher, be sure to set the PF0n bit to 1.

4.3.2 Port 1 (V850ES/JC3-L (48-pin), V850ES/JE3-L)

Port 1 is a 1-bit port for which I/O settings can be controlled in 1-bit units.

Port 1 includes the following alternate-function pin.

Table 4-9. Port 1 Alternate-Function Pins

Pin No.			Function Name	Alternate Function		Remark	Block Type
JC3L (40)	JC3L (48)	JE3L		Name	I/O		
-	3	3	P10	ANO0	Output	-	A-2

Caution When the power is turned on, the P10 pin may output an undefined level temporarily even during reset.

Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

(1) Port 1 register (P1)

After reset: 00H (output latch)		R/W	Address: FFFFF402H					
	7	6	5	4	3	2	1	0
P1	0	0	0	0	0	0	0	P10
P10	Output data control (in output mode)							
0	Outputs 0							
1	Outputs 1							

Caution Do not read or write the P1 register during D/A conversion (see 14.4.3 Cautions).

(2) Port 1 mode register (PM1)

After reset: FFH		R/W	Address: FFFFF422H					
	7	6	5	4	3	2	1	0
PM1	1	1	1	1	1	1	1	PM10
PM10	I/O mode control							
0	Output mode							
1	Input mode							

Caution When using P10 as the alternate function (ANO0 pin output), specify the input mode (PM10 bit = 1)..

4.3.3 Port 3

Port 3 is a 8-bit port (V850ES/JE3-L) or a 5-bit port (V850ES/JC3-L (48-pin)), a 2-bit port (V850ES/JC3-L (40-pin)) for which I/O settings can be controlled in 1-bit units.

Port 3 includes the following alternate-function pins.

Table 4-10. Port 3 Alternate-Function Pins

Pin No.			Function Name	Alternate Function		Remark	Block Type
JC3L (40)	JC3L (48)	JE3L		Name	I/O		
–	33	45	P30	TXDA0/SOB4	Output	Selectable as N-ch open-drain output	G-3
29	–	–		TXDA0	Output		G-13
–	34	46	P31	RXDA0/INTP7/SIB4	Input		N-3
30	–	–		RXDA0/INTP7	Input		N-4
–	35	47	P32	ASCKA0/ $\overline{\text{SCKB4}}$ /TIP00/TOP00	I/O		U-1
–	–	43	P33	TIP01/TOP01	I/O		G-1
–	–	44	P34	TIP10/TOP10	I/O		G-1
–	–	23	P35	TIP11/TOP11	I/O		G-1
–	18	26	P38	TXDA2/SDA00	I/O		G-12
–	19	27	P39	RXDA2/SCL00	I/O		G-6

Caution The P31 to P35, P38, P39 pins have hysteresis characteristics in the input mode of the alternate-function pin, but do not have hysteresis characteristics in the port mode.

Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

(1) Port 3 register (P3)

(a) V850ES/JC3-L (40-pin)

After reset: 0000H (output latch) R/W Address: FFFFF406H

	7	6	5	4	3	2	1	0
P3L	0	0	0	0	0	0	P31	P30

P3n	Output data control (in output mode) (n = 0, 1)
0	Outputs 0
1	Outputs 1

(b) V850ES/JC3-L (48-pin)

After reset: 0000H (output latch) R/W Address: P3 FFFFF406H,
P3L FFFFF406H, P3H FFFFF407H

	15	14	13	12	11	10	9	8
P3 (P3H)	0	0	0	0	0	0	P39	P38
	7	6	5	4	3	2	1	0
(P3L)	0	0	0	0	0	P32	P31	P30

P3n	Output data control (in output mode) (n = 0 to 2, 8, 9)
0	Outputs 0
1	Outputs 1

(c) V850ES/JE3-L

After reset: 0000H (output latch) R/W Address: P3 FFFFF406H,
P3L FFFFF406H, P3H FFFFF407H

	15	14	13	12	11	10	9	8
P3 (P3H)	0	0	0	0	0	0	P39	P38
	7	6	5	4	3	2	1	0
(P3L)	0	0	P35	P34	P33	P32	P31	P30

P3n	Output data control (in output mode) (n = 0 to 5, 8, 9)
0	Outputs 0
1	Outputs 1

Remarks 1. The P3 register can be read or written in 16-bit units.

However, when using the higher 8 bits of the P3 register as the P3H register and the lower 8 bits as the P3L register, P3 can be read or written in 8-bit or 1-bit units.

2. To read/write bits 8 to 15 of the P3 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the P3H register.

(2) Port 3 mode register (PM3)**(a) V850ES/JC3-L (40-pin)**

After reset: FFFFH R/W Address: FFFFF426H

	7	6	5	4	3	2	1	0
PM3L	1	1	1	1	1	1	PM31	PM30

PM3n	I/O mode control (n = 0, 1)	
0	Output mode	
1	Input mode	

(b) V850ES/JC3-L (48-pin)After reset: FFFFH R/W Address: PM3 FFFFF426H,
PM3L FFFFF426H, PM3H FFFFF427H

	15	14	13	12	11	10	9	8
PM3 (PM3H)	1	1	1	1	1	1	PM39	PM38

	7	6	5	4	3	2	1	0
(PM3L)	1	1	1	1	1	PM32	PM31	PM30

PM3n	I/O mode control (n = 0 to 2, 8, 9)	
0	Output mode	
1	Input mode	

(c) V850ES/JE3-LAfter reset: FFFFH R/W Address: PM3 FFFFF426H,
PM3L FFFFF426H, PM3H FFFFF427H

	15	14	13	12	11	10	9	8
PM3 (PM3H)	1	1	1	1	1	1	PM39	PM38

	7	6	5	4	3	2	1	0
(PM3L)	1	1	PM35	PM34	PM33	PM32	PM31	PM30

PM3n	I/O mode control (n = 0 to 5, 8, 9)	
0	Output mode	
1	Input mode	

- Remarks**
- The PM3 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PM3 register as the PM3H register and the lower 8 bits as the PM3L register, PM3 can be read or written in 8-bit or 1-bit units.
 - To read/write bits 8 to 15 of the PM3 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PM3H register.

(3) Port 3 mode control register (PMC3)

(1/2)

(a) V850ES/JC3-L (40-pin)

After reset: 0000H R/W Address: FFFFF446H

	7	6	5	4	3	2	1	0
PMC3L	0	0	0	0	0	0	PMC31	PMC30

PMC31	Specification of pin operation	
0	I/O port (P31)	
1	RXDA0 input/INTP7 input	

PMC30	Specification of pin operation	
0	I/O port (P30)	
1	TXDA0 output	

(b) V850ES/JC3-L (48-pin)

After reset: 0000H R/W Address: PMC3 FFFFF446H,
PMC3L FFFFF446H, PMC3H FFFFF447H

	15	14	13	12	11	10	9	8
PMC3 (PMC3H)	0	0	0	0	0	0	PMC39	PMC38

	7	6	5	4	3	2	1	0
(PMC3L)	0	0	0	0	0	PMC32	PMC31	PMC30

PMC39	Specification of pin operation	
0	I/O port (P39)	
1	RXDA2 input/SCL00 I/O	

PMC38	Specification of pin operation	
0	I/O port (P38)	
1	TXDA2 output/SDA00 I/O	

PMC32	Specification of pin operation	
0	I/O port (P32)	
1	ASCKA0 input/SCKB4 I/O/TIP00 input/TOP00 output	

PMC31	Specification of pin operation	
0	I/O port (P31)	
1	RXDA0 input/SIB4 input/INTP7 input	

PMC30	Specification of pin operation	
0	I/O port (P30)	
1	TXDA0 output/SOB4 output	

(c) V850ES/JE3-L

After reset: 0000H R/W Address: PMC3 FFFF446H,
PMC3L FFFF446H, PMC3H FFFF447H

	15	14	13	12	11	10	9	8
PMC3 (PMC3H)	0	0	0	0	0	0	PMC39	PMC38

	7	6	5	4	3	2	1	0
(PMC3L)	0	0	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

PMC39	Specification of pin operation
0	I/O port (P39)
1	RXDA2 input/SCL00 I/O

PMC38	Specification of pin operation
0	I/O port (P38)
1	TXDA2 output/SDA00 I/O

PMC35	Specification of pin operation
0	I/O port (P35)
1	TIP11 input/TOP11 output

PMC34	Specification of pin operation
0	I/O port (P34)
1	TIP10 input/TOP10 output

PMC33	Specification of pin operation
0	I/O port (P33)
1	TIP01 input/TOP01 output

PMC32	Specification of pin operation
0	I/O port (P32)
1	ASCKA0 input/SCKB4 I/O/TIP00 input/TOP00 output

PMC31	Specification of pin operation
0	I/O port (P31)
1	RXDA0 input/SIB4 input/INTP7 input

PMC30	Specification of pin operation
0	I/O port (P30)
1	TXDA0 output/SOB4 output

- Remarks**
- The PMC3 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PMC3 register as the PMC3H register and the lower 8 bits as the PMC3L register, PMC3 can be read or written in 8-bit or 1-bit units.
 - To read/write bits 8 to 15 of the PMC3 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PMC3H register.

(4) Port 3 function control register (PFC3)**(a) V850ES/JC3-L (48-pin)**

After reset: 0000H R/W Address: PFC3 FFFFF466H,
PFC3L FFFFF466H, PFC3H FFFFF467H

	15	14	13	12	11	10	9	8
PFC3 (PFC3H)	0	0	0	0	0	0	PFC39	PFC38
	7	6	5	4	3	2	1	0
(PFC3L)	0	0	0	0	0	PFC32	PFC31	PFC30

(b) V850ES/JE3-L

After reset: 0000H R/W Address: PFC3 FFFFF466H,
PFC3L FFFFF466H, PFC3H FFFFF467H

	15	14	13	12	11	10	9	8
PFC3 (PFC3H)	0	0	0	0	0	0	PFC39	PFC38
	7	6	5	4	3	2	1	0
(PFC3L)	0	0	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

Remarks 1. For details of alternate function specification, see **4.3.3 (6) Port 3 alternate function specifications.**

2. The PFC3 register can be read or written in 16-bit units.

However, when using the higher 8 bits of the PFC3 register as the PFC3H register and the lower 8 bits as the PFC3L register, PFC3 can be read or written in 8-bit and 1-bit units.

3. To read/write bits 8 to 15 of the PFC3 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PFC3H register.

(5) Port 3 function control expansion register L (PFCE3L) (V850ES/JC3-L (48-pin), V850ES/JE3-L)

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3L	0	0	0	0	0	PFCE32	0	0

Remark For details of alternate function specification, see **4.3.3 (6) Port 3 alternate function specifications.**

(6) Port 3 alternate function specifications

PFC39 ^{Note1}	Specification of P39 pin alternate function
0	RXDA2 input
1	SCL00 I/O

PFC38 ^{Note1}	Specification of P38 pin alternate function
0	TXDA2 output
1	SDA00 I/O

PFC35 ^{Note2}	Specification of P35 pin alternate function
0	TIP11 input
1	TOP11 output

PFC34 ^{Note2}	Specification of P34 pin alternate function
0	TIP10 input
1	TOP10 output

PFC33 ^{Note2}	Specification of P33 pin alternate function
0	TIP01 input
1	TOP01 output

PFC32 ^{Note1}	PFC32 ^{Note1}	Specification of P32 pin alternate function
0	0	ASCKA0 input
0	1	$\overline{\text{SCKB4}}$ I/O
1	0	TIP00 input
1	1	TOP00 output

PFC31	Specification of P31 pin alternate function
0	RXDA0 input/INTP7 ^{Note3} input
1	SIB4 input ^{Note1}

PFC30	Specification of P30 pin alternate function
0	TXDA0 output
1	SOB4 output ^{Note1}

- Notes1.** V850ES/JC3-L (48-pin), V850/JE3-L only
- 2.** V850ES/JE3-L only
- 3.** INTP7 and RXDA0 are alternate functions. When using the pin for RXDA0, disable edge detection for INTP7 (clear the INTF3.INTF31 bit and the INTR3.INTR31 bit to 0). When using the pin for INTP7, stop UARTA0 reception (clear the UA0CTL0.UA0RXE bit to 0).

(7) Port 3 function register (PF3)**(a) V850ES/JC3-L (40-pin)**

After reset: 0000H R/W Address: FFFFFFFC66H

	7	6	5	4	3	2	1	0
PF3L	0	0	0	0	0	0	PF31	PF30

PF3n	Specification of normal output (CMOS output) or N-ch open-drain output (n = 0, 1)
0	Normal output (CMOS output)
1	N-ch open-drain output

(b) V850ES/JC3-L (48-pin)After reset: 0000H R/W Address: PF3 FFFFFFFC66H,
PF3L FFFFFFFC66H, PF3H FFFFFFFC67H

	15	14	13	12	11	10	9	8
PF3 (PF3H)	0	0	0	0	0	0	PF39	PF38

	7	6	5	4	3	2	1	0
(PF3L)	0	0	0	0	0	PF32	PF31	PF30

PF3n	Specification of normal output (CMOS output) or N-ch open-drain output (n = 0 to 2, 8, 9)
0	Normal output (CMOS output)
1	N-ch open-drain output

(c) V850ES/JE3-LAfter reset: 0000H R/W Address: PF3 FFFFFFFC66H,
PF3L FFFFFFFC66H, PF3H FFFFFFFC67H

	15	14	13	12	11	10	9	8
PF3 (PF3H)	0	0	0	0	0	0	PF39	PF38

	7	6	5	4	3	2	1	0
(PF3L)	0	0	PF35	PF34	PF33	PF32	PF31	PF30

PF3n	Specification of normal output (CMOS output) or N-ch open-drain output (n = 0 to 5, 8, 9)
0	Normal output (CMOS output)
1	N-ch open-drain output

Cautions 1. When an output pin is pulled up to EV_{DD} or higher, be sure to set the PF3n bit to 1.

2. Be sure to clear bits 15 to 10, 5 to 3 to "0".

Remarks 1. The PF3 register can be read or written in 16-bit units.

However, when using the higher 8 bits of the PF3 register as the PF3H register and the lower 8 bits as the PF3L register, PF3 can be read or written in 8-bit or 1-bit units.

2. To read/write bits 8 to 15 of the PF3 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PF3H register.

4.3.4 Port 4

Port 4 is a 3-bit port that controls I/O in 1-bit units.

Port 4 includes the following alternate-function pins.

Table 4-11. Port 4 Alternate-Function Pins

Pin No.			Function Name	Alternate Function		Remark	Block Type
JC3L (40)	JC3L (48)	JE3L		Name	I/O		
16	20	28	P40	SIB0/SDA01	I/O	Selectable as N-ch open-drain output	G-6
17	21	29	P41	SOB0/SCL01	I/O		G-12
18	22	30	P42	SCKB0	I/O		E-3

Caution The P40 to P42 pins have hysteresis characteristics in the input mode of the alternate-function pin, but do not have hysteresis characteristics in the port mode.

Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

(1) Port 4 register (P4)

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

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

P4n	Output data control (in output mode) (n = 0 to 2)
0	Outputs 0
1	Outputs 1

(2) Port 4 mode register (PM4)

After reset: FFH R/W Address: FFFFF428H

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

PM4n	I/O mode control (n = 0 to 2)
0	Output mode
1	Input mode

(3) Port 4 mode control register (PMC4)

After reset: 00H		R/W	Address: FFFFF448H					
	7	6	5	4	3	2	1	0
PMC4	0	0	0	0	0	PMC42	PMC41	PMC40
	Specification of pin operation							
	0	I/O port (P42)						
	1	SCKB0 I/O						
	Specification of pin operation							
	0	I/O port (P41)						
	1	SOB0 output/SCL01 I/O						
	Specification of pin operation							
	0	I/O port (P40)						
	1	SIB0 input/SDA01 I/O						

(4) Port 4 function control register (PFC4)

After reset: 00H		R/W	Address: FFFFF468H					
	7	6	5	4	3	2	1	0
PFC4	0	0	0	0	0	0	PFC41	PFC40
	Specification of P41 pin alternate function							
	0	SOB0 output						
	1	SCL01 I/O						
	Specification of P40 pin alternate function							
	0	SIB0 input						
	1	SDA01 I/O						

(5) Port 4 function register (PF4)

After reset: 00H		R/W	Address: FFFFFC68H					
	7	6	5	4	3	2	1	0
PF4	0	0	0	0	0	PF42	PF41	PF40
	Specification of normal output (CMOS output) or N-ch open-drain output (n = 0 to 2)							
	0	Normal output (CMOS output)						
	1	N-ch open-drain output						

Caution When an output pin is pulled up to EV_{DD} or higher, be sure to set the PF4n bit to 1.

4.3.5 Port 5

Port 5 is a 6-bit port that controls I/O in 1-bit units.

Port 5 includes the following alternate-function pins.

Table 4-12. Port 5 Alternate-Function Pins

Pin No.			Function Name	Alternate Function		Remark	Block Type
JC3L (40)	JC3L (48)	JE3L		Name	I/O		
21	25	35	P50	TIQ01/KR0/TOQ01/RTP00	I/O	Selectable as N-ch open-drain output	U-5
22	26	36	P51	TIQ02/KR1/TOQ02/RTP01	I/O		U-5
23	27	37	P52	TIQ03/KR2/TOQ03/RTP02/DDI ^{Note}	I/O		U-6
24	28	38	P53	SIB2/KR3/TIQ00/TOQ00/RTP03/DDO ^{Note}	I/O		U-7
25	29	39	P54	SOB2/KR4/RTP04/DCK ^{Note}	I/O		U-8
26	30	40	P55	SCKB2/KR5/RTP05/DMS ^{Note}	I/O		U-9

Note The DDI, DDO, DCK, and DMS pins are used for on-chip debugging.

- Cautions**
1. When the power is turned on, the P53 pin may output an undefined level temporarily even during reset.
 2. The P50 to P55 pins have hysteresis characteristics in the input mode of the alternate function, but do not have hysteresis characteristics in the port mode.

Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

(1) Port 5 register (P5)

After reset: 00H (output latch) R/W Address: FFFFF40AH								
	7	6	5	4	3	2	1	0
P5	0	0	P55	P54	P53	P52	P51	P50
P5n	Output data control (in output mode) (n = 0 to 5)							
0	Outputs 0							
1	Outputs 1							

(2) Port 5 mode register (PM5)

After reset: FFH R/W Address: FFFFF42AH

	7	6	5	4	3	2	1	0
PM5	1	1	PM55	PM54	PM53	PM52	PM51	PM50

PM5n	I/O mode control (n = 0 to 5)
0	Output mode
1	Input mode

(3) Port 5 mode control register (PMC5)

After reset: 00H R/W Address: FFFFF44AH

	7	6	5	4	3	2	1	0
PMC5	0	0	PMC55	PMC54	PMC53	PMC52	PMC51	PMC50

PMC55	Specification of pin operation
0	I/O port (P55)
1	SCKB2 I/O/KR5 input/RTP05 output

PMC54	Specification of pin operation
0	I/O port (P54)
1	SOB2 output/KR4 input/RTP04 output

PMC53	Specification of pin operation
0	I/O port (P53)
1	SIB2 input/KR3 input/TIQ00 input/TOQ00 output/RTP03 output

PMC52	Specification of pin operation
0	I/O port (P52)
1	TIQ03 input/KR2 input/TOQ03 output/RTP02 output

PMC51	Specification of pin operation
0	I/O port (P51)
1	TIQ02 input/KR1 input/TOQ02 output/RTP01 output

PMC50	Specification of pin operation
0	I/O port (P50)
1	TIQ01 input/KR0 input/TOQ01 output/RTP00 output

(4) Port 5 function control register (PFC5)

After reset: 00H R/W Address: FFFFF46AH

	7	6	5	4	3	2	1	0
PFC5	0	0	PFC55	PFC54	PFC53	PFC52	PFC51	PFC50

Remark For details of alternate function specification, see **4.3.5 (6) Port 5 alternate function specifications**.

(5) 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	PFCE55	PFCE54	PFCE53	PFCE52	PFCE51	PFCE50

Remark For details of alternate function specification, see **4.3.5 (6) Port 5 alternate function specifications**.

(6) Port 5 alternate function specifications

PFCE55	PFC55	Specification of P55 pin alternate function
0	0	$\overline{\text{SCKB2}}$ I/O
0	1	KR5 input
1	0	Setting prohibited
1	1	RTP05 output

PFCE54	PFC54	Specification of P54 pin alternate function
0	0	SOB2 output
0	1	KR4 input
1	0	Setting prohibited
1	1	RTP04 output

PFCE53	PFC53	Specification of P53 pin alternate function
0	0	SIB2 input
0	1	TIQ00 input/KR3 ^{Note} input
1	0	TOQ00 output
1	1	RTP03 output

PFCE52	PFC52	Specification of P52 pin alternate function
0	0	Setting prohibited
0	1	TIQ03 input/KR2 ^{Note} input
1	0	TOQ03 input
1	1	RTP02 output

PFCE51	PFC51	Specification of P51 pin alternate function
0	0	Setting prohibited
0	1	TIQ02 input/KR1 ^{Note} input
1	0	TOQ02 output
1	1	RTP01 output

PFCE50	PFC50	Specification of P50 pin alternate function
0	0	Setting prohibited
0	1	TIQ01 input/KR0 ^{Note} input
1	0	TOQ01 output
1	1	RTP00 output

Note KRn and TIQ0m are alternate functions. When using the pin for TIQ0m, disable key return detection for KRn (clear the KRM.KRMn bit to 0). When using the pin for KRn, disable edge detection for TIQ0m (n = 0 to 3, m = 0 to 3).

Alternate Function Name	Use as TIQ0m Function	Use as KRn Function
KR0/TIQ01	KRM.KRM0 bit = 0	TQ0IOC1.TQ0TIG2, TQ0IOC1.TQ0TIG3 bits = 0
KR1/TIQ02	KRM.KRM1 bit = 0	TQ0IOC1.TQ0TIG4, TQ0IOC1.TQ0TIG5 bits = 0
KR2/TIQ03	KRM.KRM2 bit = 0	TQ0IOC1.TQ0TIG6, TQ0IOC1.TQ0TIG7 bits = 0
KR3/TIQ00	KRM.KRM3 bit = 0	TQ0IOC1.TQ0TIG0, TQ0IOC1.TQ0TIG1 bits = 0 TQ0IOC2.TQ0EES0, TQ0IOC2.TQ0EES1 bits = 0 TQ0IOC2.TQ0ETS0, TQ0IOC2.TQ0ETS1 bits = 0

(7) Port 5 function register (PF5)

After reset: 00H R/W Address: FFFFC6AH

	7	6	5	4	3	2	1	0
PF5	0	0	PF55	PF54	PF53	PF52	PF51	PF50

PF5n	Specification of normal output (CMOS output) or N-ch open-drain output (n = 0 to 5)
0	Normal output (CMOS output)
1	N-ch open-drain output

- Cautions**
1. When an output pin is pulled up to EV_{DD} or higher, be sure to set the PF5n bit to 1.
 2. Bits 6 and 7 of the PF5 register must always be set to 0.

4.3.6 Port 7

Port 7 is a 10-bit port (V850ES/JE3-L) or a 6-bit port (V850ES/JC3-L (48-pin)), a 5-bit port (V850ES/JC3-L (40-pin)) for which I/O settings can be controlled in 1-bit units.

Port 7 includes the following alternate-function pins.

Table 4-13. Port 7 Alternate-Function Pins

Pin No.			Function Name	Alternate Function		Remark	Block Type
JC3L (40)	JC3L (48)	JE3L		Name	I/O		
40	48	64	P70	ANI0	Input	-	A-1
39	47	63	P71	ANI1	Input		A-1
38	46	62	P72	ANI2	Input		A-1
37	45	61	P73	ANI3	Input		A-1
36	44	60	P74	ANI4	Input		A-1
-	43	59	P75	ANI5	Input		A-1
-	-	58	P76	ANI6	Input		A-1
-	-	57	P77	ANI7	Input		A-1
-	-	56	P78	ANI8	Input		A-1
-	-	55	P79	ANI9	Input		A-1

Remark JC3L (40): V850ES/JC3-L (40-pin products)

JC3L (48): V850ES/JC3-L (48-pin products)

JE3L: V850ES/JE3-L

(1) Port 7 register H, port 7 register L (P7H, P7L)

(a) V850ES/JC3-L (40-pin)

After reset: 00H (output latch) R/W Address: FFFFF40EH

	7	6	5	4	3	2	1	0
P7L	0	0	0	P74	P73	P72	P71	P70

P7n	Output data control (in output mode) (n = 0 to 4)
0	Outputs 0
1	Outputs 1

(b) V850ES/JC3-L (48-pin)

After reset: 00H (output latch) R/W Address: FFFFF40EH

	7	6	5	4	3	2	1	0
P7L	0	0	P75	P74	P73	P72	P71	P70

P7n	Output data control (in output mode) (n = 0 to 5)
0	Outputs 0
1	Outputs 1

(c) V850ES/JE3-L

After reset: 00H (output latch) R/W Address: P7L FFFFF40EH, P7H FFFFF40FH

	7	6	5	4	3	2	1	0
P7H	0	0	0	0	0	0	P79	P78

	7	6	5	4	3	2	1	0
P7L	P77	P76	P75	P74	P73	P72	P71	P70

P7n	Output data control (in output mode) (n = 0 to 9)
0	Outputs 0
1	Outputs 1

Caution Do not read or write the P7H and P7L registers during A/D conversion (see 13.6 (4) Alternate I/O).

Remark These registers cannot be accessed in 16-bit units as the P7 register. They can be read or written in 8-bit or 1-bit units as the P7H and P7L registers.

(2) Port 7 mode register H, port 7 mode register L (PM7H, PM7L)

(a) V850ES/JC3-L (40-pin)

After reset: FFH R/W Address: FFFFF42EH

	7	6	5	4	3	2	1	0
PM7L	1	1	1	PM74	PM73	PM72	PM71	PM70

PM7n	I/O mode control (n = 0 to 4)
0	Output mode
1	Input mode

(b) V850ES/JC3-L (48-pin)

After reset: FFH R/W Address: FFFFF42EH

	7	6	5	4	3	2	1	0
PM7L	1	1	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	I/O mode control (n = 0 to 5)
0	Output mode
1	Input mode

(c) V850ES/JE3-L

After reset: FFH R/W Address: PM7L FFFFF42EH, PM7H FFFFF42FH

	7	6	5	4	3	2	1	0
PM7H	1	1	1	1	1	1	PM79	PM78

	7	6	5	4	3	2	1	0
PM7L	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	I/O mode control (n = 0 to 9)
0	Output mode
1	Input mode

Caution When using the P7n pin as its alternate function (ANIn pin), set the PM7n bit to 1.

Remark These registers cannot be accessed in 16-bit units as the PM7 register. They can be read or written in 8-bit or 1-bit units as the PM7H and PM7L registers.

4.3.7 Port 9

Port 9 is a 15-bit port (V850ES/JE3-L) or a 6-bit port (V850ES/JC3-L (48-pin)), a 5-bit port (V850ES/JC3-L (40-pin)) for which I/O settings can be controlled in 1-bit units.

Port 9 includes the following alternate-function pins.

Table 4-14. Port 9 Alternate-Function Pins

Pin No.			Function Name	Alternate Function		Remark	Block Type
JC3L (40)	JC3L (48)	JE3L		Name	I/O		
19	23	31	P90	KR6/TXDA1/SDA02	I/O	Selectable as N-ch open-drain output	U-18
20	24	32	P91	KR7/RXDA1/SCL02	I/O		U-19
–	–	33	P92	TIP41/TOP41	I/O		U-20
–	–	34	P93	TIP40/TOP40	I/O		U-20
–	–	21	P94	TIP31/TOP31	I/O		U-20
13	15	22	P96	TIP21/TOP21	I/O		U-21
–	38	50	P97	SIB1/TIP20/TOP20	I/O		U-22
32	–	–		TIP20/TOP20	I/O		U-23
–	39	51	P98	SOB1	Output		G-14
–	36	48	P99	$\overline{\text{SCKB1}}$	I/O		G-15
–	–	16	P910	SIB3	I/O		G-16
–	–	15	P911	SOB3	Output		G-14
–	–	14	P912	$\overline{\text{SCKB3}}$	I/O		G-15
–	–	13	P913	INTP4	I/O		N-5
33	40	52	P914	INTP5/TIP51/TOP51	I/O		U-24
34	41	53	P915	INTP6/TIP50/TOP50	I/O	U-24	

Caution The P90 to P94, P97, P99, P910, and P912 to P915 pins have hysteresis characteristics in the input mode of the alternate-function pin, but do not have hysteresis characteristics in the port mode.

Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

(1) Port 9 register (P9)**(a) V850ES/JC3-L (40-pin)**

After reset: 0000H (output latch) R/W Address: P9 FFFFF412H,
P9L FFFFF412H, P9H FFFFF413H

	15	14	13	12	11	10	9	8
P9 (P9H)	P915	P914	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
(P9L)	P97	P96	0	0	0	0	P91	P90

P9n	Output data control (in output mode) (n = 0, 1, 6, 7, 14, 15)
0	Outputs 0
1	Outputs 1

(b) V850ES/JC3-L (48-pin)

After reset: 0000H (output latch) R/W Address: P9 FFFFF412H,
P9L FFFFF412H, P9H FFFFF413H

	15	14	13	12	11	10	9	8
P9 (P9H)	P915	P914	0	0	0	0	P99	P98

	7	6	5	4	3	2	1	0
(P9L)	P97	P96	0	0	0	0	P91	P90

P9n	Output data control (in output mode) (n = 0, 1, 6 to 9, 14, 15)
0	Outputs 0
1	Outputs 1

(c) V850ES/JE3-L

After reset: 0000H (output latch) R/W Address: P9 FFFFF412H,
P9L FFFFF412H, P9H FFFFF413H

	15	14	13	12	11	10	9	8
P9 (P9H)	P915	P914	P913	P912	P911	P910	P99	P98

	7	6	5	4	3	2	1	0
(P9L)	P97	P96	0	P94	P93	P92	P91	P90

P9n	Output data control (in output mode) (n = 0 to 4, 6 to 15)
0	Outputs 0
1	Outputs 1

Remarks 1. The P9 register can be read or written in 16-bit units.

However, when using the higher 8 bits of the P9 register as the P9H register and the lower 8 bits as the P9L register, P9 can be read or written in 8-bit or 1-bit units.

2. To read/write bits 8 to 15 of the P9 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the P9H register.

(2) Port 9 mode register (PM9)**(a) V850ES/JC3-L (40-pin)**

After reset: FFFFH R/W Address: PM9 FFFFF432H,
PM9L FFFFF432H, PM9H FFFFF433H

	15	14	13	12	11	10	9	8
PM9 (PM9H)	PM915	PM914	1	1	1	1	1	1
	7	6	5	4	3	2	1	0
(PM9L)	PM97	PM96	1	1	1	1	PM91	PM90
PM9n	I/O mode control (n = 0, 1, 6, 7, 14, 15)							
0	Output mode							
1	Input mode							

(b) V850ES/JC3-L (48-pin)

After reset: FFFFH R/W Address: PM9 FFFFF432H,
PM9L FFFFF432H, PM9H FFFFF433H

	15	14	13	12	11	10	9	8
PM9 (PM9H)	PM915	PM914	1	1	1	1	PM99	PM98
	7	6	5	4	3	2	1	0
(PM9L)	PM97	PM96	1	1	1	1	PM91	PM90
PM9n	I/O mode control (n = 0, 1, 6 to 9, 14, 15)							
0	Output mode							
1	Input mode							

(c) V850ES/JE3-L

After reset: FFFFH R/W Address: PM9 FFFFF432H,
PM9L FFFFF432H, PM9H FFFFF433H

	15	14	13	12	11	10	9	8
PM9 (PM9H)	PM915	PM914	PM913	PM912	PM911	PM910	PM99	PM98
	7	6	5	4	3	2	1	0
(PM9L)	PM97	PM96	1	PM94	PM93	PM92	PM91	PM90
PM9n	I/O mode control (n = 0 to 4, 6 to 15)							
0	Output mode							
1	Input mode							

- Remarks**
- The PM9 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PM9 register as the PM9H register and the lower 8 bits as the PM9L register, PM9 can be read or written in 8-bit and 1-bit units.
 - To read/write bits 8 to 15 of the PM9 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PM9H register.

(3) Port 9 mode control register (PMC9)

(1/4)

(a) V850ES/JC3-L (40-pin)

After reset: 0000H R/W Address: PMC9 FFFFF452H,
PMC9L FFFFF452H, PMC9H FFFFF453H

	15	14	13	12	11	10	9	8
PMC9 (PMC9H)	PMC915	PMC914	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
(PMC9L)	PMC97	PMC96	0	0	0	0	PMC91	PMC90

PMC915	Specification of pin operation
0	I/O port (P915)
1	INTP6 input/TIP50 input/TOP50 output

PMC914	Specification of pin operation
0	I/O port (P914)
1	INTP5 input/TIP51 input/TOP51 output

PMC97	Specification of pin operation
0	I/O port (P97)
1	TIP20 input/TOP20 output

PMC96	Specification of pin operation
0	I/O port (P96)
1	TIP21 input/TOP21 output

PMC91	Specification of pin operation
0	I/O port (P91)
1	KR7 input/RXDA1 input/SCL02 I/O

PMC90	Specification of pin operation
0	I/O port (P90)
1	KR6 input/TXDA1 output/SDA02 I/O

(2/4)

(b) V850ES/JC3-L (48-pin)

After reset: 0000H

R/W

Address: PMC9 FFFFF452H,
PMC9L FFFFF452H, PMC9H FFFFF453H

	15	14	13	12	11	10	9	8
PMC9 (PMC9H)	PMC915	PMC914	0	0	0	0	PMC99	PMC98

	7	6	5	4	3	2	1	0
(PMC9L)	PMC97	PMC96	0	0	0	0	PMC91	PMC90

PMC915	Specification of pin operation
0	I/O port (P915)
1	INTP6 input/TIP50 input/TOP50 output

PMC914	Specification of pin operation
0	I/O port (P914)
1	INTP5 input/TIP51 input/TOP51 output

PMC99	Specification of pin operation
0	I/O port (P99)
1	SCKB1 I/O

PMC98	Specification of pin operation
0	I/O port (P98)
1	SOB1 output

PMC97	Specification of pin operation
0	I/O port (P97)
1	SIB1 input/TIP20 input/TOP20 output

PMC96	Specification of pin operation
0	I/O port (P96)
1	TIP21 input/TOP21 output

PMC91	Specification of pin operation
0	I/O port (P91)
1	KR7 input/RXDA1 input/SCL02 I/O

PMC90	Specification of pin operation
0	I/O port (P90)
1	KR6 input/TXDA1 output/SDA02 I/O

(c) V850ES/JE3-L (1/2)

After reset: 0000H R/W Address: PMC9 FFFFF452H,
 PMC9L FFFFF452H, PMC9H FFFFF453H

	15	14	13	12	11	10	9	8
PMC9 (PMC9H)	PMC915	PMC914	PMC913	PMC912	PMC911	PMC910	PMC99	PMC98

	7	6	5	4	3	2	1	0
(PMC9L)	PMC97	PMC96	0	PMC94	PMC93	PMC92	PMC91	PMC90

PMC915	Specification of pin operation
0	I/O port (P915)
1	INTP6 input/TIP50 input/TOP50 output
PMC914	Specification of pin operation
0	I/O port (P914)
1	INTP5 input/TIP51 input/TOP51 output
PMC913	Specification of pin operation
0	I/O port (P913)
1	INTP4 input
PMC912	Specification of pin operation
0	I/O port (P912)
1	SCKB3 I/O
PMC911	Specification of pin operation
0	I/O port (P911)
1	SOB3 output
PMC910	Specification of pin operation
0	I/O port (P910)
1	SIB3 input
PMC99	Specification of pin operation
0	I/O port (P99)
1	SCKB1 I/O
PMC98	Specification of pin operation
0	I/O port (P98)
1	SOB1 output

(c) V850ES/JE3-L (2/2)

PMC97	Specification of pin operation
0	I/O port (P97)
1	SIB1 input/TIP20 input/TOP20 output
PMC96	Specification of pin operation
0	I/O port (P96)
1	TIP21 input/TOP21 output
PMC94	Specification of pin operation
0	I/O port (P94)
1	TIP31 input/TOP31 output
PMC93	Specification of pin operation
0	I/O port (P93)
1	TIP40 input/TOP40 output
PMC92	Specification of pin operation
0	I/O port (P92)
1	TIP41 input/TOP41 output
PMC91	Specification of pin operation
0	I/O port (P91)
1	KR7 input/RXDA1 input/SCL02 I/O
PMC90	Specification of pin operation
0	I/O port (P90)
1	KR6 input/TXDA1 output/SDA02 I/O

- Remarks**
1. The PMC9 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PMC9 register as the PMC9H register and the lower 8 bits as the PMC9L register, PMC9 can be read or written in 8-bit or 1-bit units.
 2. To read/write bits 8 to 15 of the PMC9 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PMC9H register.

(4) Port 9 function control register (PFC9)

Caution When performing separate address bus output (A0 to A15), set the PMC9 register to FFFFH for all 16 bits at once after clearing the PFC9 and PFCE9 registers to 0000H.

(a) V850ES/JC3-L (40-pin)

After reset: 0000H R/W Address: PFC9 FFFFF472H,
PFC9L FFFFF472H, PFC9H FFFFF473H

	15	14	13	12	11	10	9	8
PFC9 (PFC9H)	PFC915	PFC914	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
(PFC9L)	PFC97	PFC96	0	0	0	0	PFC91	PFC90

(b) V850ES/JC3-L (48-pin)

After reset: 0000H R/W Address: PFC9 FFFFF472H,
PFC9L FFFFF472H, PFC9H FFFFF473H

	15	14	13	12	11	10	9	8
PFC9 (PFC9H)	PFC915	PFC914	0	0	0	0	PFC99	PFC98
	7	6	5	4	3	2	1	0
(PFC9L)	PFC97	PFC96	0	0	0	0	PFC91	PFC90

(c) V850ES/JE3-L

After reset: 0000H R/W Address: PFC9 FFFFF472H,
PFC9L FFFFF472H, PFC9H FFFFF473H

	15	14	13	12	11	10	9	8
PFC9 (PFC9H)	PFC915	PFC914	PFC913	PFC912	PFC911	PFC910	PFC99	PFC98
	7	6	5	4	3	2	1	0
(PFC9L)	PFC97	PFC96	0	PFC94	PFC93	PFC92	PFC91	PFC90

Remarks 1. For details of alternate function specification, see **4.3.7 (6) Port 9 alternate function specifications.**

2. The PFC9 register can be read or written in 16-bit units.

However, when using the higher 8 bits of the PFC9 register as the PFC9H register and the lower 8 bits as the PFC9L register, PFC9 can be read or written in 8-bit or 1-bit units.

3. To read/write bits 8 to 15 of the PFC9 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PFC9H register.

(5) Port 9 function control expansion register (PFCE9)

Caution When performing separate address bus output (A0 to A15), set the PMC9 register to FFFFH for all 16 bits at once after clearing the PFC9 and PFCE9 registers to 0000H.

(a) V850ES/JC3-L (40-pin), V850ES/JC3-L (48-pin)

After reset: 0000H	R/W	Address: PFCE9 FFFFF712H, PFCE9L FFFFF712H, PFCE9H FFFFF713H							
		15	14	13	12	11	10	9	8
PFCE9 (PFCE9H)		PFCE915	PFCE914	0	0	0	0	0	0
		7	6	5	4	3	2	1	0
(PFCE9L)		PFCE97	PFCE96	0	0	0	0	PFCE91	PFCE90

(b) V850ES/JE3-L

After reset: 0000H	R/W	Address: PFCE9 FFFFF712H, PFCE9L FFFFF712H, PFCE9H FFFFF713H							
		15	14	13	12	11	10	9	8
PFCE9 (PFCE9H)		PFCE915	PFCE914	0	0	0	0	0	0
		7	6	5	4	3	2	1	0
(PFCE9L)		PFCE97	PFCE96	0	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90

- Remarks**
- For details of alternate function specification, see **4.3.7 (6) Port 9 alternate function specifications**.
 - The PFCE9 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PFCE9 register as the PFCE9H register and the lower 8 bits as the PFCE9L register, PFCE9 can be read or written in 8-bit or 1-bit units.
 - To read/write bits 8 to 15 of the PFCE9 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PFCE9H register.

(6) Port 9 alternate function specifications

PFCE915	PFC915	Specification of P915 pin alternate function
0	0	Setting prohibited
0	1	INTP6 input
1	0	TIP50 input
1	1	TOP50 output

PFCE914	PFC914	Specification of P914 pin alternate function
0	0	Setting prohibited
0	1	INTP5 input
1	0	TIP51 input
1	1	TOP51 output

PFC913 ^{Note1}	Specification of P913 pin alternate function
0	Setting prohibited
1	INTP4 input

PFC912 ^{Note1}	Specification of P912 pin alternate function
0	Setting prohibited
1	SCKB3 I/O

PFC911 ^{Note1}	Specification of P911 pin alternate function
0	Setting prohibited
1	SOB3 output

PFC910 ^{Note1}	Specification of P910 pin alternate function
0	Setting prohibited
1	SIB3 input

PFC99 ^{Note2}	Specification of P99 pin alternate function
0	Setting prohibited
1	SCKB1 I/O

PFC98 ^{Note2}	Specification of P98 pin alternate function
0	Setting prohibited
1	SOB1 output

PFCE97	PFC97	Specification of P97 pin alternate function
0	0	Setting prohibited
0	1	SIB1 input ^{Note2}
1	0	TIP20 input
1	1	TOP20 output

- Notes**
1. V850ES/JE3-L only
 2. V850ES/JC3-L (48-pin), V850ES/JE3-L only

PFCE96	PFC96	Specification of P96 pin alternate function
0	0	Setting prohibited
0	1	Setting prohibited
1	0	TIP21 input
1	1	TOP21 output

PFCE94 ^{Note1}	PFC94 ^{Note1}	Specification of P94 pin alternate function
0	0	Setting prohibited
0	1	TIP31 input
1	0	TOP31 output
1	1	Setting prohibited

PFCE93 ^{Note1}	PFC93 ^{Note1}	Specification of P93 pin alternate function
0	0	Setting prohibited
0	1	TIP40 input
1	0	TOP40 output
1	1	Setting prohibited

PFCE92 ^{Note1}	PFC92 ^{Note1}	Specification of P92 pin alternate function
0	0	Setting prohibited
0	1	TIP41 input
1	0	TOP41 output
1	1	Setting prohibited

PFCE91	PFC91	Specification of P91 pin alternate function
0	0	Setting prohibited
0	1	KR7 input
1	0	RXDA1 input/KR7 input ^{Note2}
1	1	SCL02 I/O

PFCE90	PFC90	Specification of P90 pin alternate function
0	0	Setting prohibited
0	1	KR6 input
1	0	TXDA1 output
1	1	SDA02 I/O

Notes1. V850ES/JE3-L only

- The RXDA1 and KR7 functions cannot be used at the same time. When using the pin for RXDA1, do not use the KR7 function. When using the pin for KR7, do not use the RXDA1 function. (It is recommended to set the PFC91 bit to 1 and clear the PFCE91 bit to 0.)

(7) Port 9 function register (PF9)

(a) V850ES/JC3-L (40-pin)

After reset: 0000H R/W Address: PF9 FFFFFFFC72H,
PF9L FFFFFFFC72H, PF9H FFFFFFFC73H

	15	14	13	12	11	10	9	8
PF9 (PF9H)	PF915	PF914	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
(PF9L)	PF97	PF96	0	0	0	0	PF91	PF90

PF9n	Specification of normal output (CMOS output) or N-ch open-drain output (n = 0, 1, 6, 7, 14, 15)
0	Normal output (CMOS output)
1	N-ch open-drain output

(b) V850ES/JC3-L (48-pin)

After reset: 0000H R/W Address: PF9 FFFFFFFC72H,
PF9L FFFFFFFC72H, PF9H FFFFFFFC73H

	15	14	13	12	11	10	9	8
PF9 (PF9H)	PF915	PF914	0	0	0	0	PF99	PF98

	7	6	5	4	3	2	1	0
(PF9L)	PF97	PF96	0	0	0	0	PF91	PF90

PF9n	Specification of normal output (CMOS output) or N-ch open-drain output (n = 0, 1, 6 to 9, 14, 15)
0	Normal output (CMOS output)
1	N-ch open-drain output

(c) V850ES/JE3-L

After reset: 0000H R/W Address: PF9 FFFFFFFC72H,
PF9L FFFFFFFC72H, PF9H FFFFFFFC73H

	15	14	13	12	11	10	9	8
PF9 (PF9H)	PF915	PF914	PF913	PF912	PF911	PF910	PF99	PF98

	7	6	5	4	3	2	1	0
(PF9L)	PF97	PF96	0	PF94	PF93	PF92	PF91	PF90

PF9n	Specification of normal output (CMOS output) or N-ch open-drain output (n = 0 to 4, 6 to 15)
0	Normal output (CMOS output)
1	N-ch open-drain output

Caution When output pins P90 to P96 are pulled up to EV_{DD} or higher, be sure to set the PF9n bit to 1. Pull up output pins P97 to P915 to the same potential as EV_{DD} , even when they are used as N-ch open-drain output pins.

- Remarks**
- The PF9 register can be read or written in 16-bit units. However, when using the higher 8 bits of the PF9 register as the PF9H register and the lower 8 bits as the PF9L register, PF9 can be read or written in 8-bit or 1-bit units.
 - To read/write bits 8 to 15 of the PF9 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PF9H register.

4.3.8 Port CM

Port CM is a 1-bit port for which I/O settings can be controlled in 1-bit units.

Port CM includes the following alternate-function pins.

Table 4-15. Port CM Alternate-Function Pins

Pin No.			Function Name	Alternate Function		Remark	Block Type
JC3L (40)	JC3L (48)	JE3L		Name	I/O		
35	42	54	PCM0	-	-	-	C-1

Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

(1) Port CM register (PCM)

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

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

PCM0	Output data control (in output mode)
0	Outputs 0
1	Outputs 1

(2) Port CM mode register (PMCM)

After reset: FFH R/W Address: FFFFF02CH

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

PMCM0	I/O mode control
0	Output mode
1	Input mode

4.3.9 Port DL

Port DL is a 1-bit port for which I/O settings can be controlled in 1-bit units.

Table 4-14. Port DL Alternate-Function Pins

Pin No.			Function Name	Alternate Function		Remark	Block Type
JC3L (40)	JC3L (48)	JE3L		Name	I/O		
31	37	49	PDL5	FLMD1 ^{Note}	Input	–	C-1

Note Since this pin is set in the flash memory programming mode, it does not need to be manipulated by using the port control register. For details, see **CHAPTER 28 FLASH MEMORY**.

Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

(1) Port DL register (PDL)

After reset: 0000H (output latch) R/W Address: PDL FFFF004H, PDL FFFF004H

	15	14	13	12	11	10	9	8
PDL	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
(PDL)	0	0	PDL5	0	0	0	0	0
PDL5	Output data control (in output mode)							
0	Outputs 0							
1	Outputs 1							

Remark The PDL register can be read or written in 16-bit units.
However, when using the lower 8 bits as the PDL register, PDL can be read or written in 8-bit or 1-bit units.

(2) Port DL mode register (PMDL)

After reset: FFFFH R/W Address: PMDL FFFF024H, PMDL FFFF024H

	15	14	13	12	11	10	9	8
PMDL	1	1	1	1	1	1	1	1
	7	6	5	4	3	2	1	0
(PMDL)	1	1	PMDL5	1	1	1	1	1
PMDL5	I/O mode control							
0	Output mode							
1	Input mode							

Remark The PMDL register can be read or written in 16-bit units.
However, when using the lower 8 bits as the PMDL register, PMDL can be read or written in 8-bit or 1-bit units.

4.4 Block Diagrams

Figure 4-5. Block Diagram of Type A-1

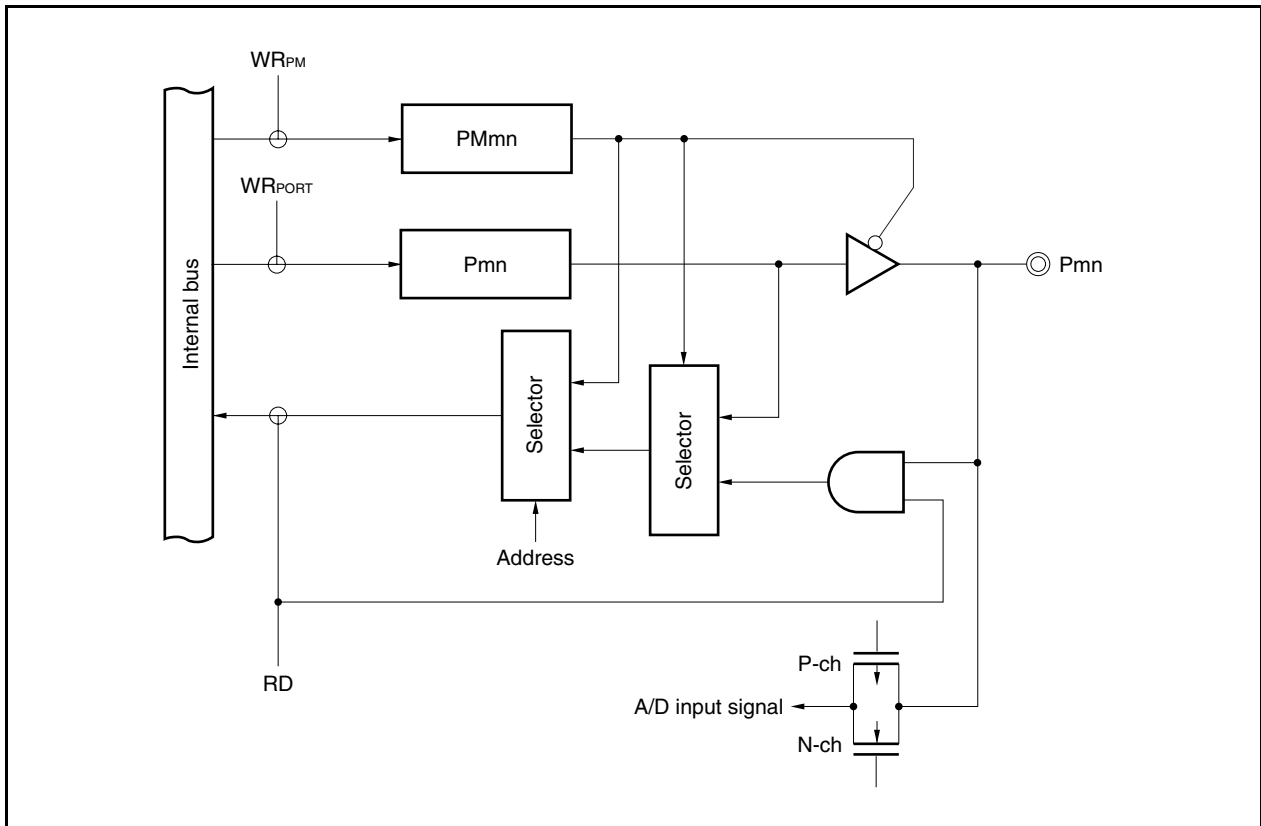


Figure 4-6. Block Diagram of Type A-2

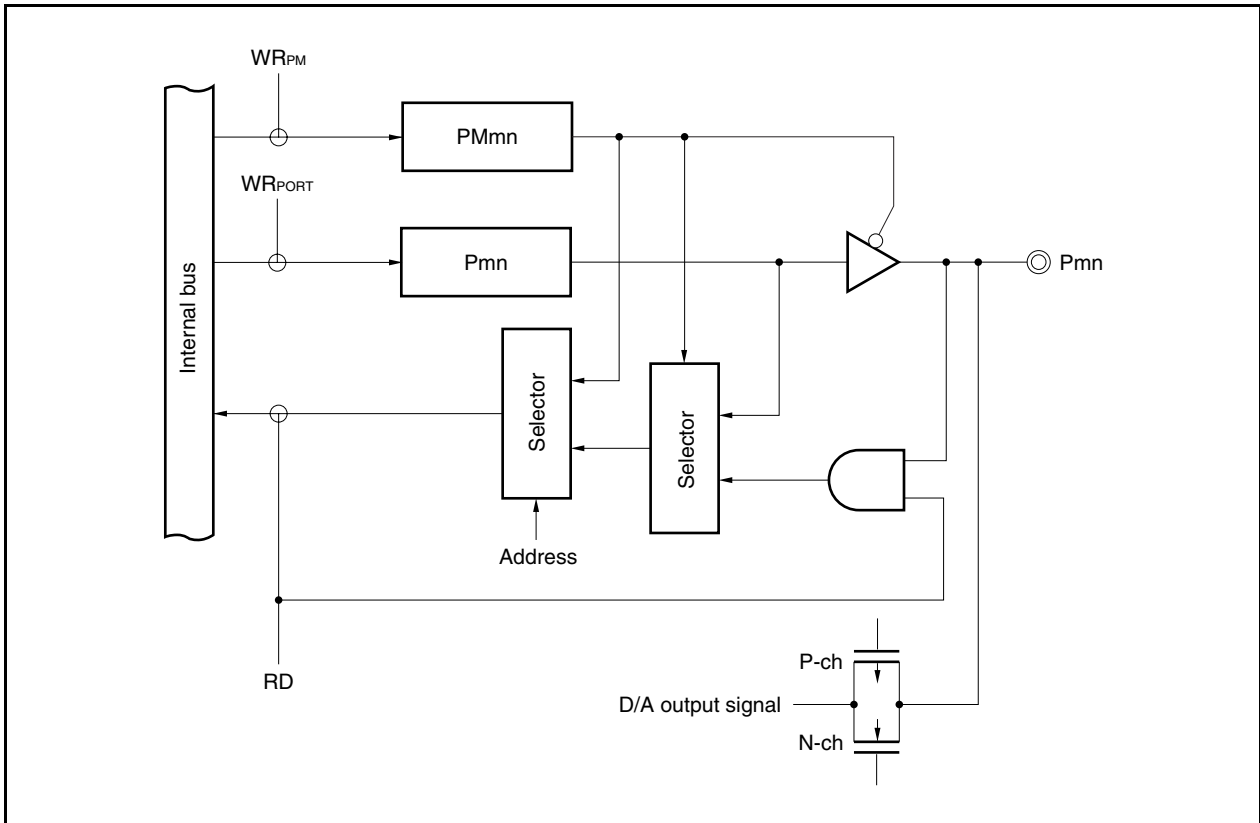


Figure 4-7. Block Diagram of Type C-1

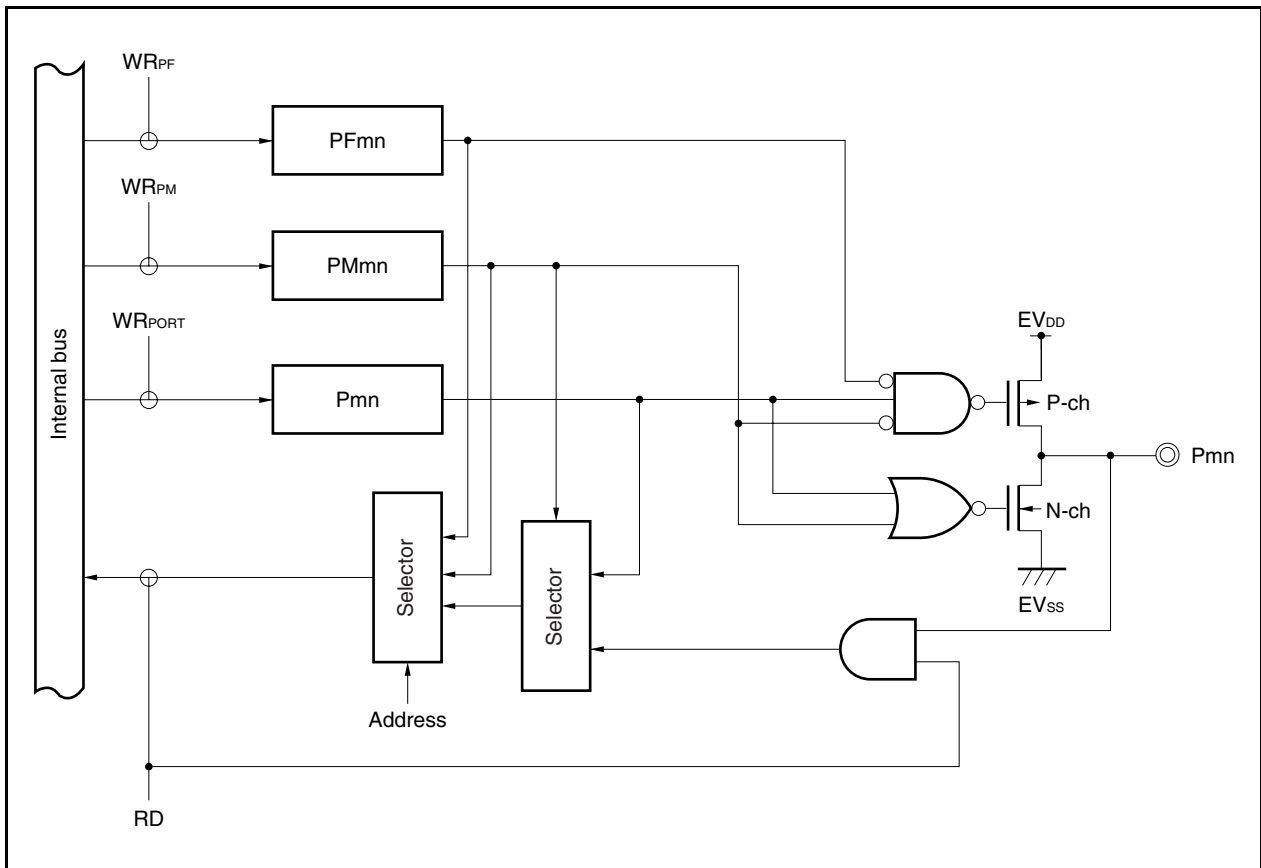
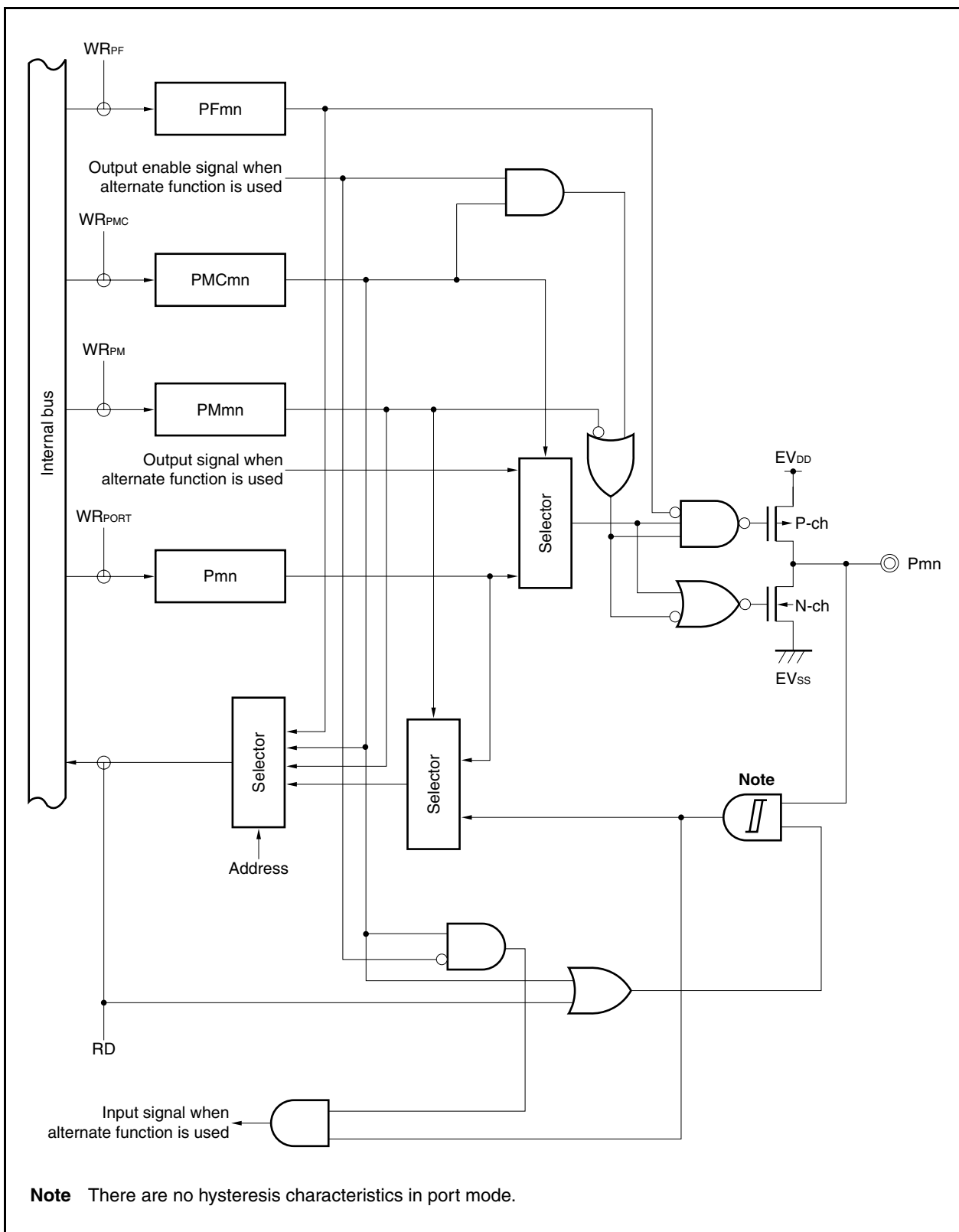


Figure 4-8. Block Diagram of Type E-3



Note There are no hysteresis characteristics in port mode.

Figure 4-9. Block Diagram of Type G-1

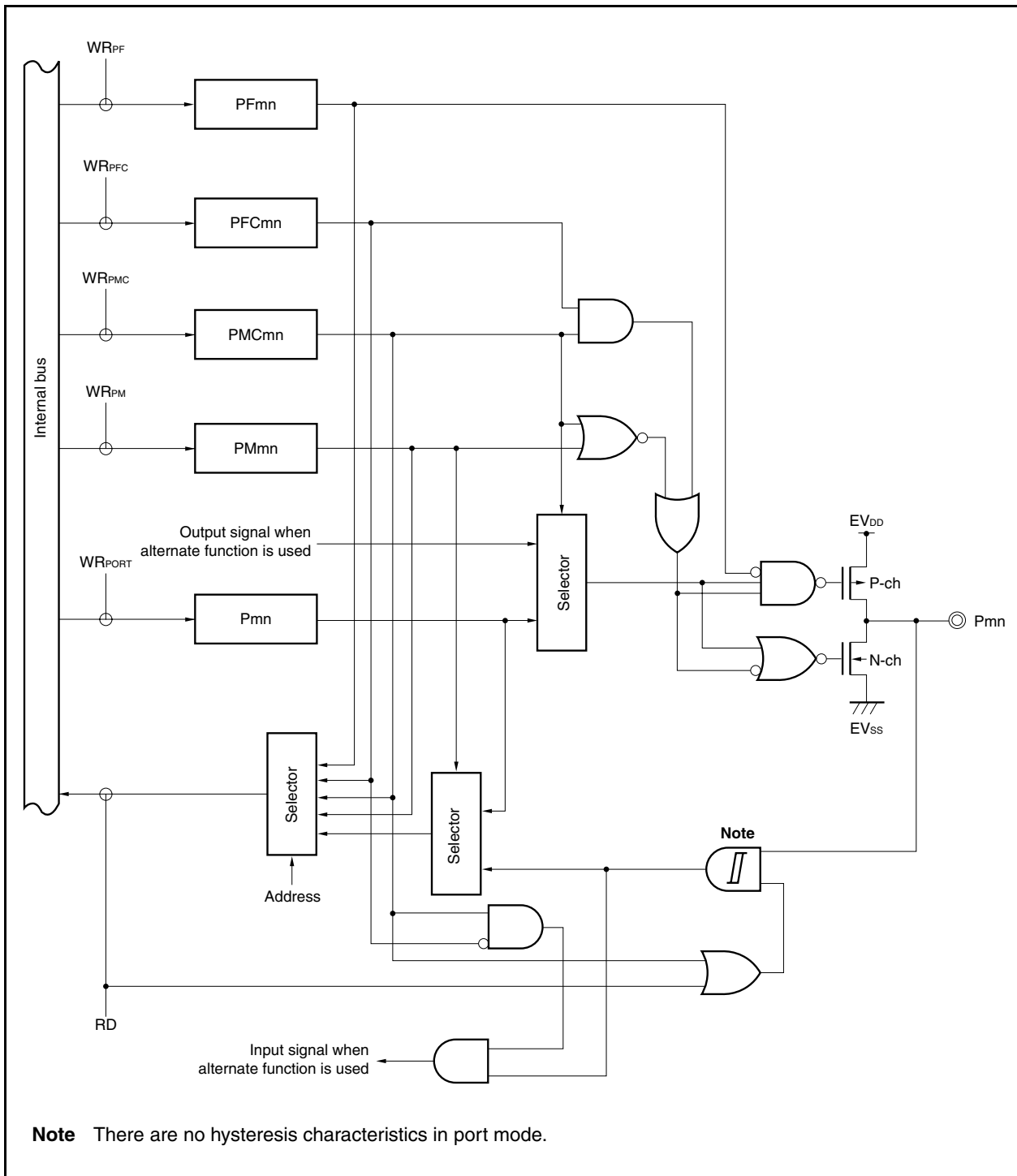


Figure 4-10. Block Diagram of Type G-3

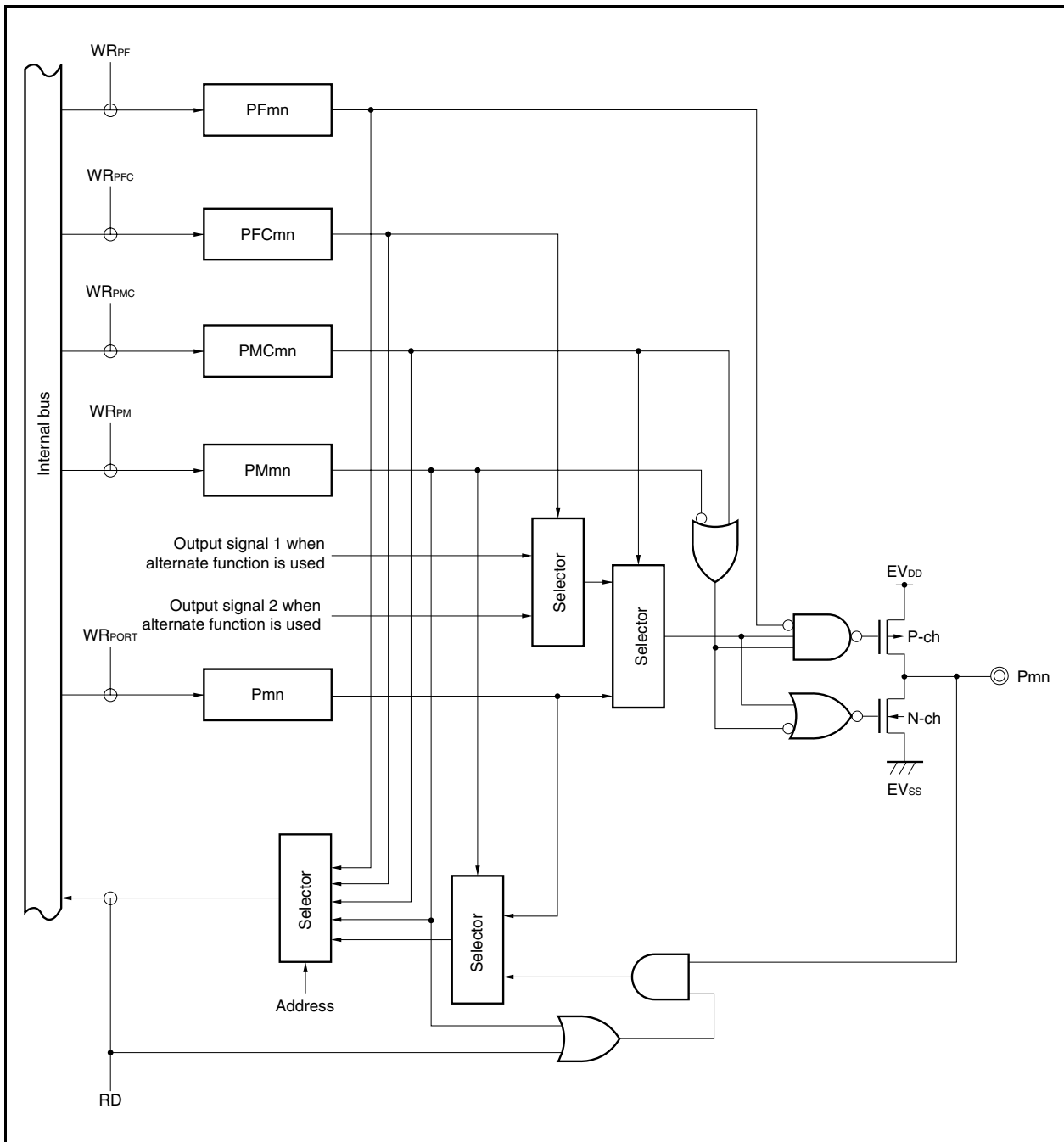


Figure 4-11. Block Diagram of Type G-6

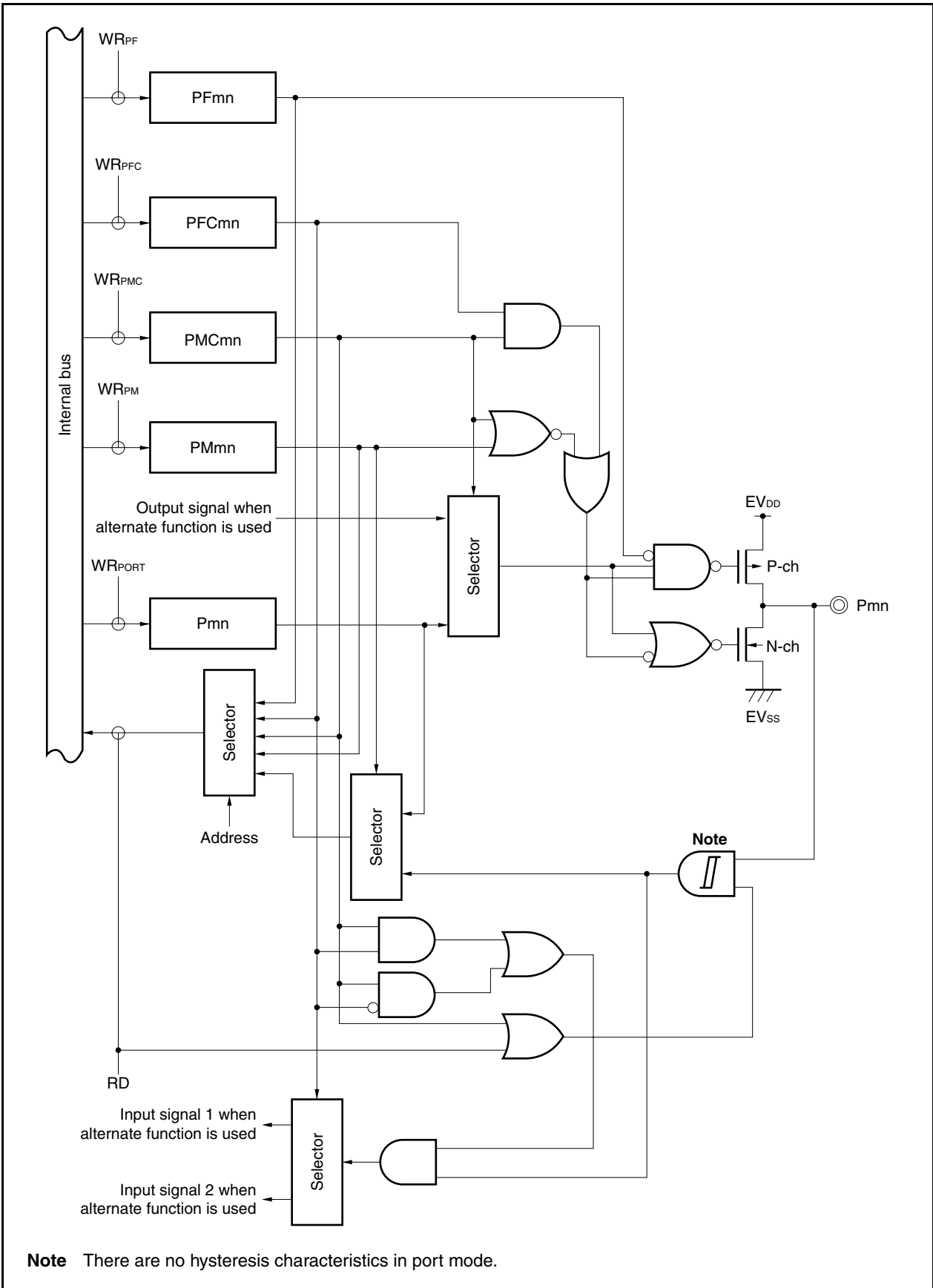


Figure 4-12. Block Diagram of Type G-12

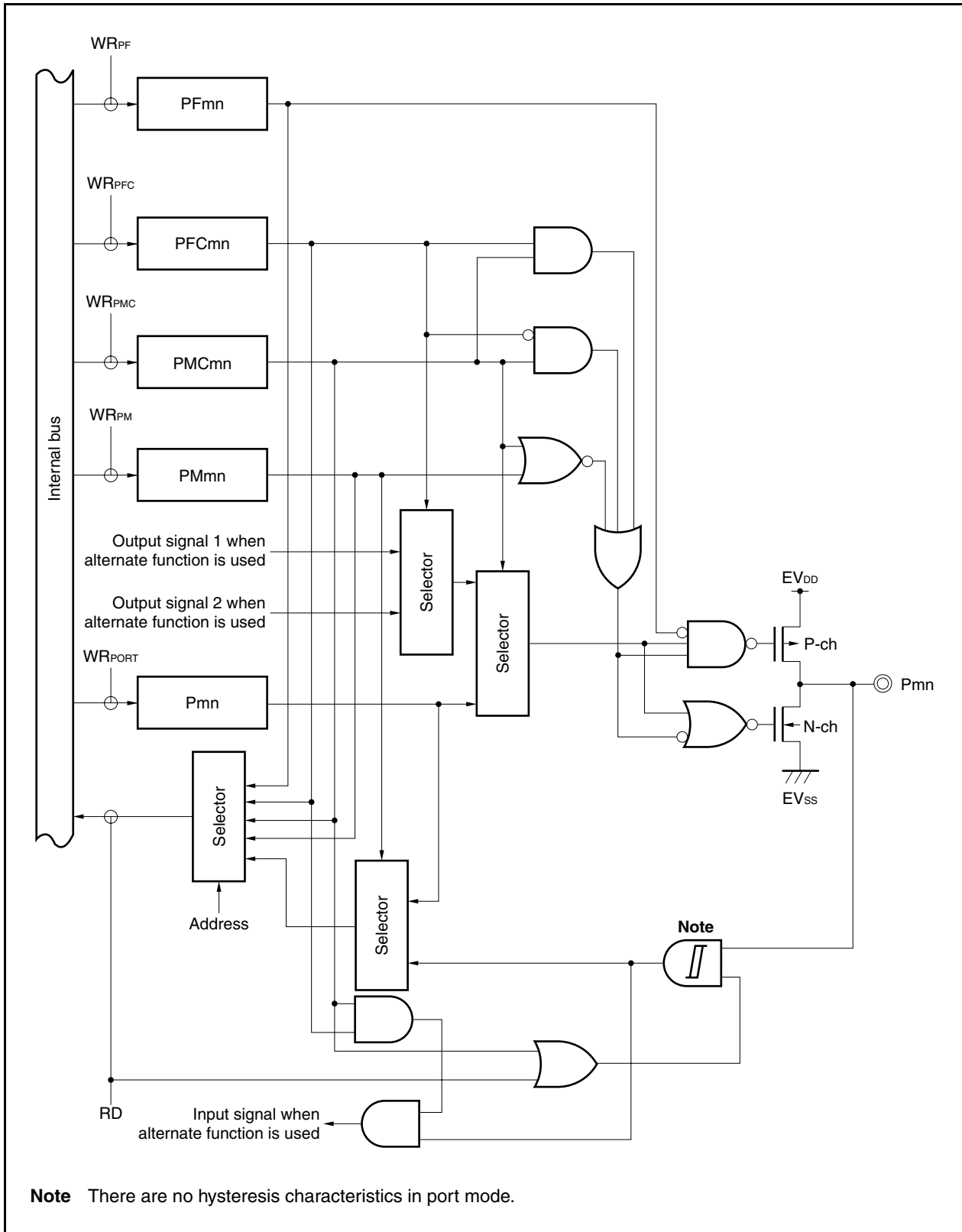


Figure 4-13. Block Diagram of Type G-13

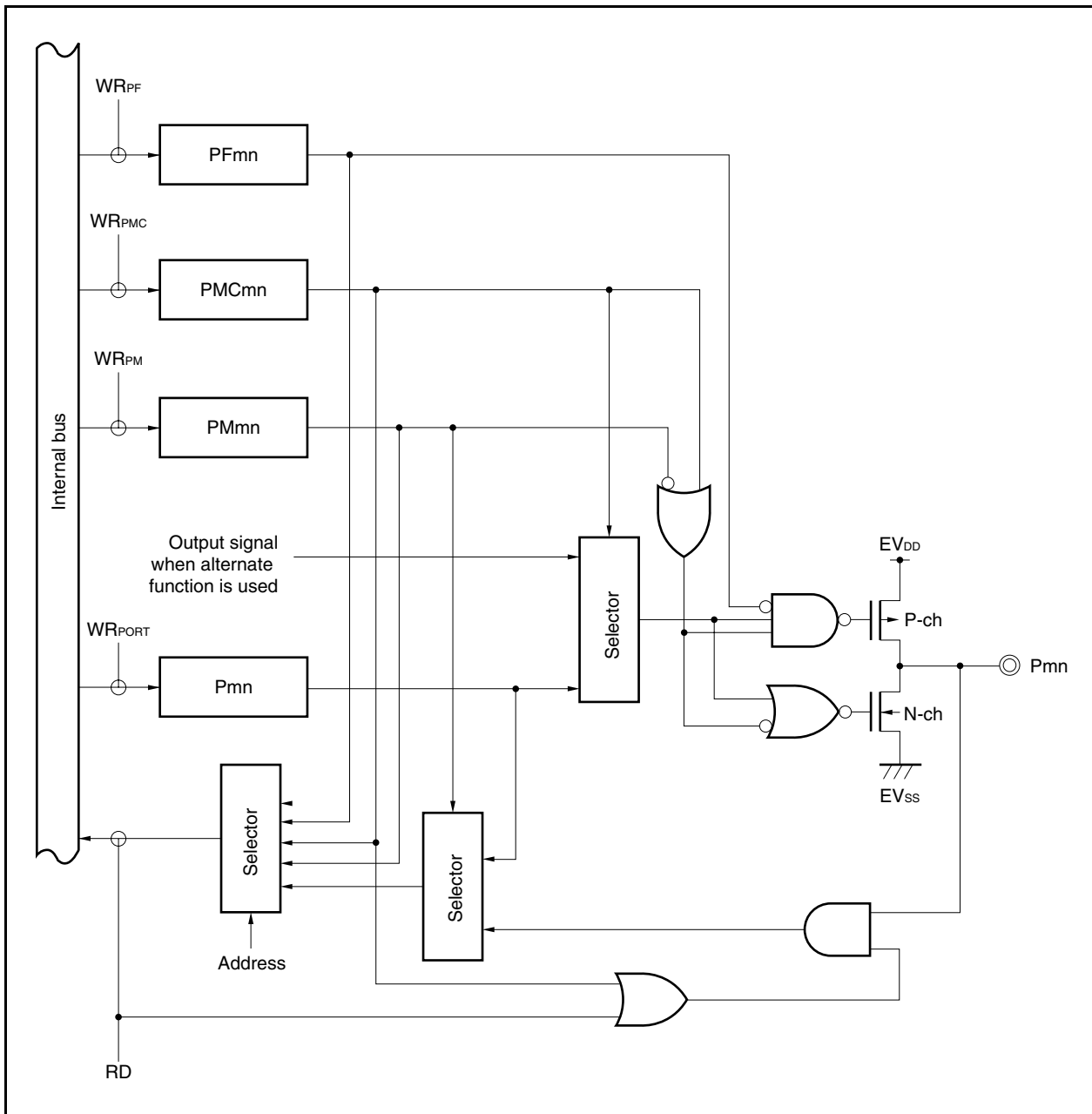


Figure 4-14. Block Diagram of Type G-14

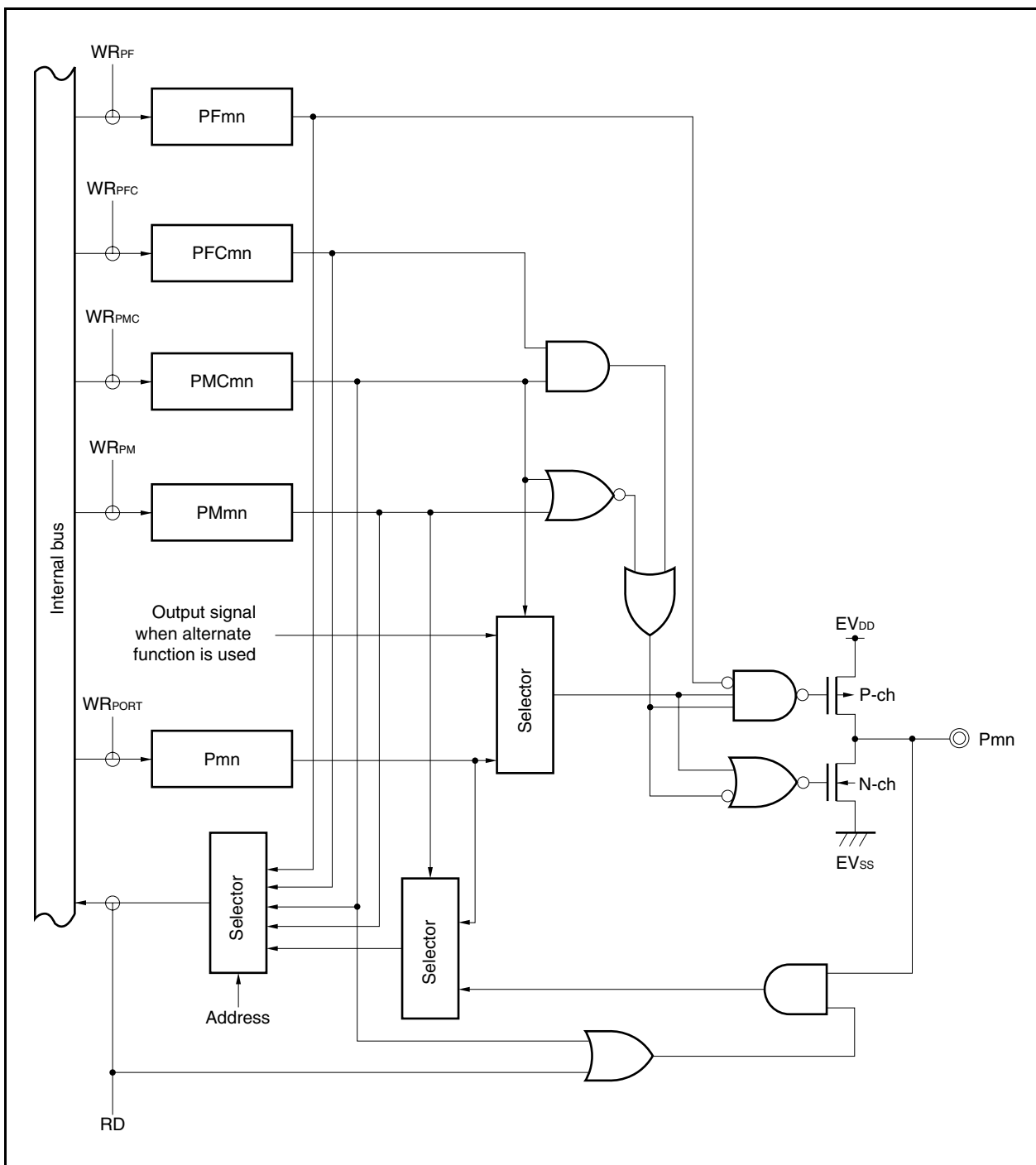


Figure 4-15. Block Diagram of Type G-15

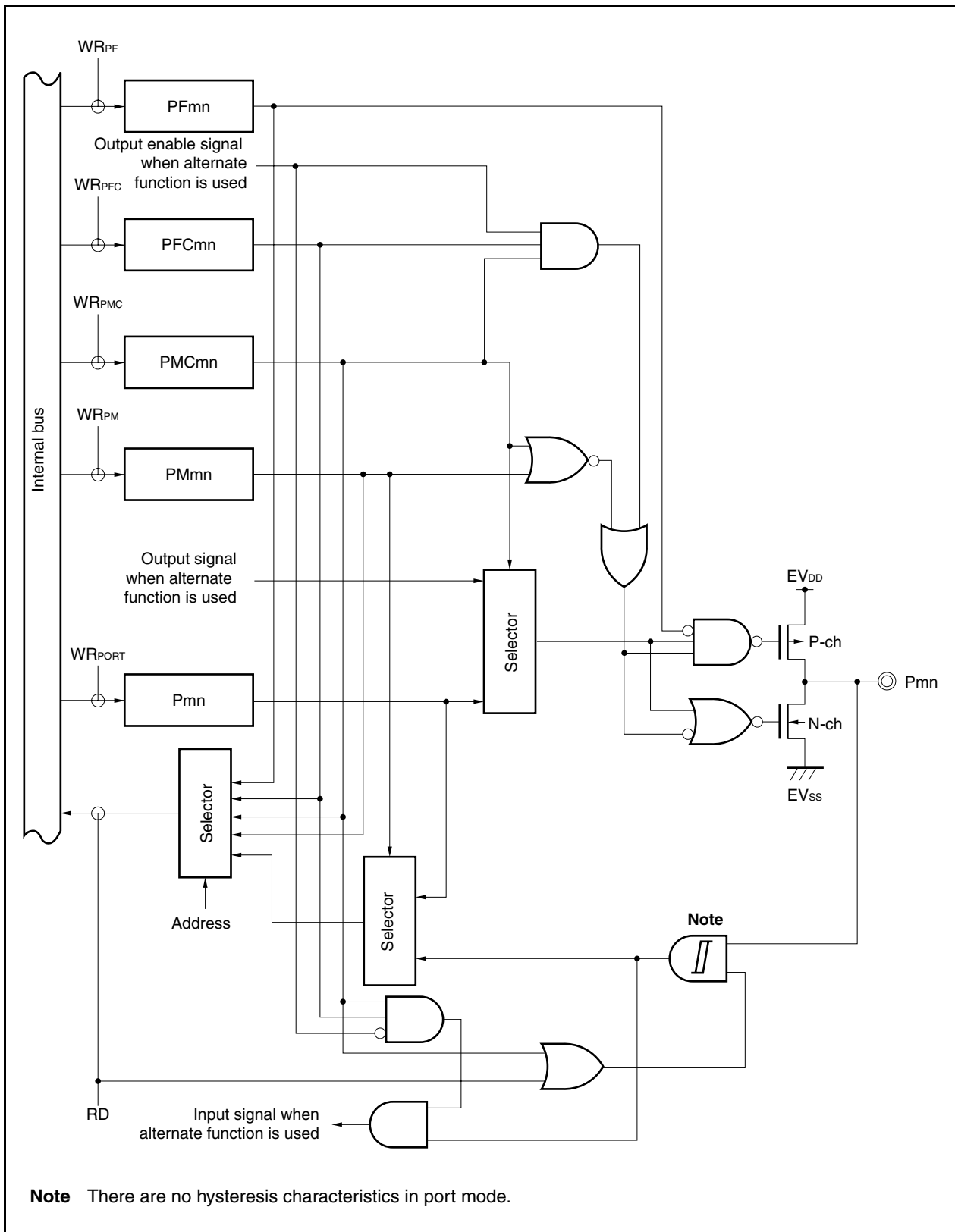


Figure 4-16. Block Diagram of Type G-16

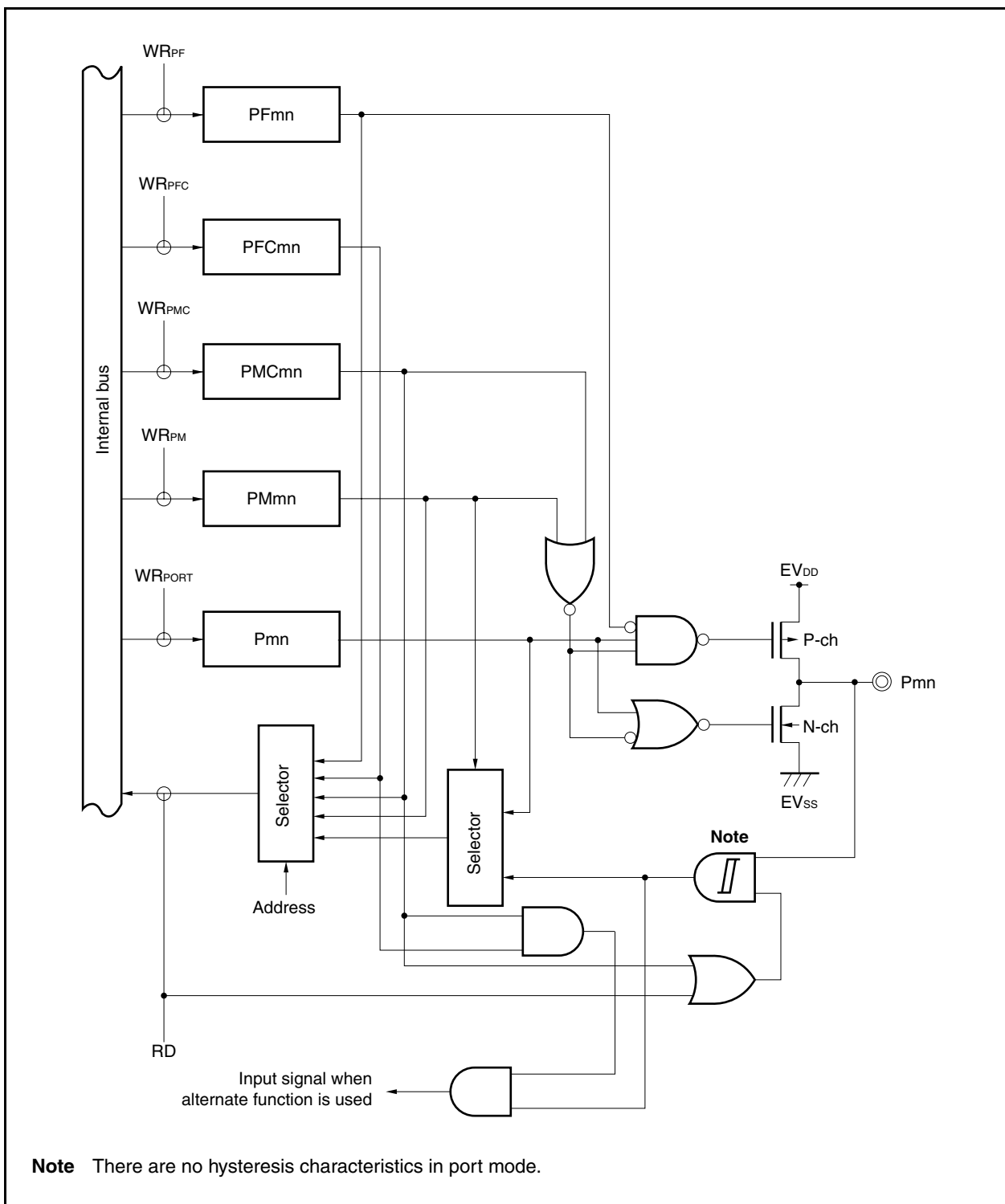


Figure 4-17. Block Diagram of Type L-1

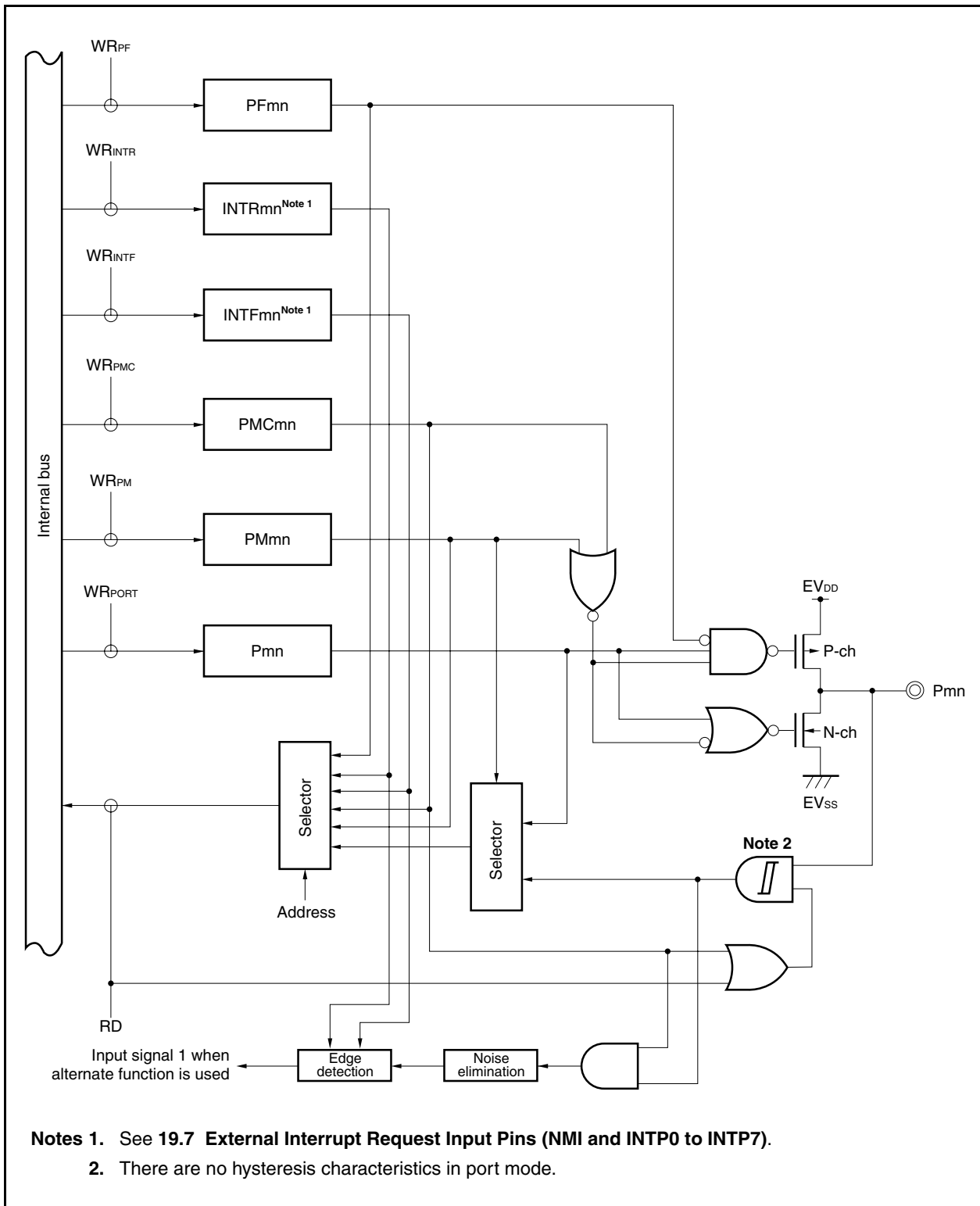


Figure 4-18. Block Diagram of Type N-2

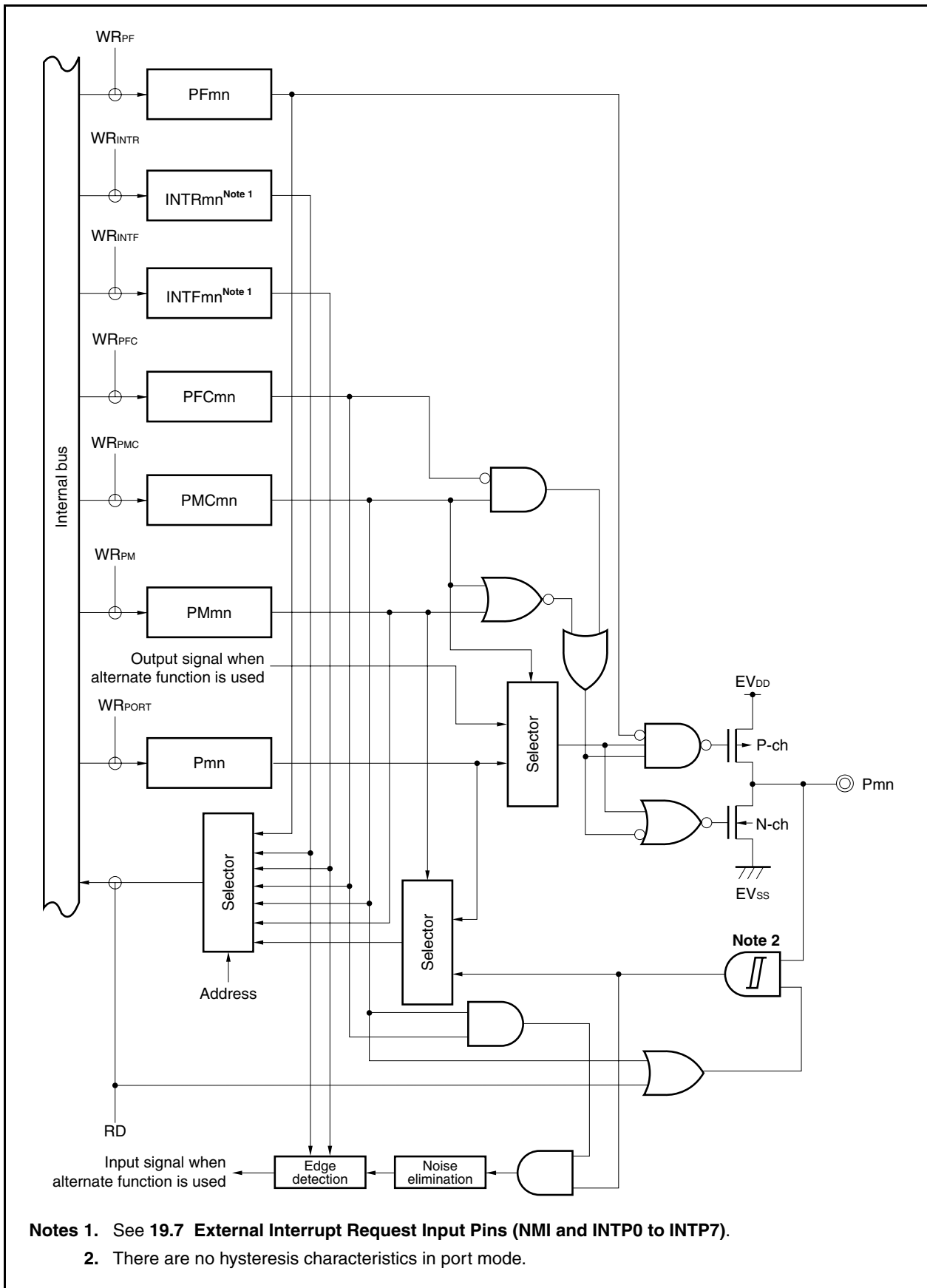
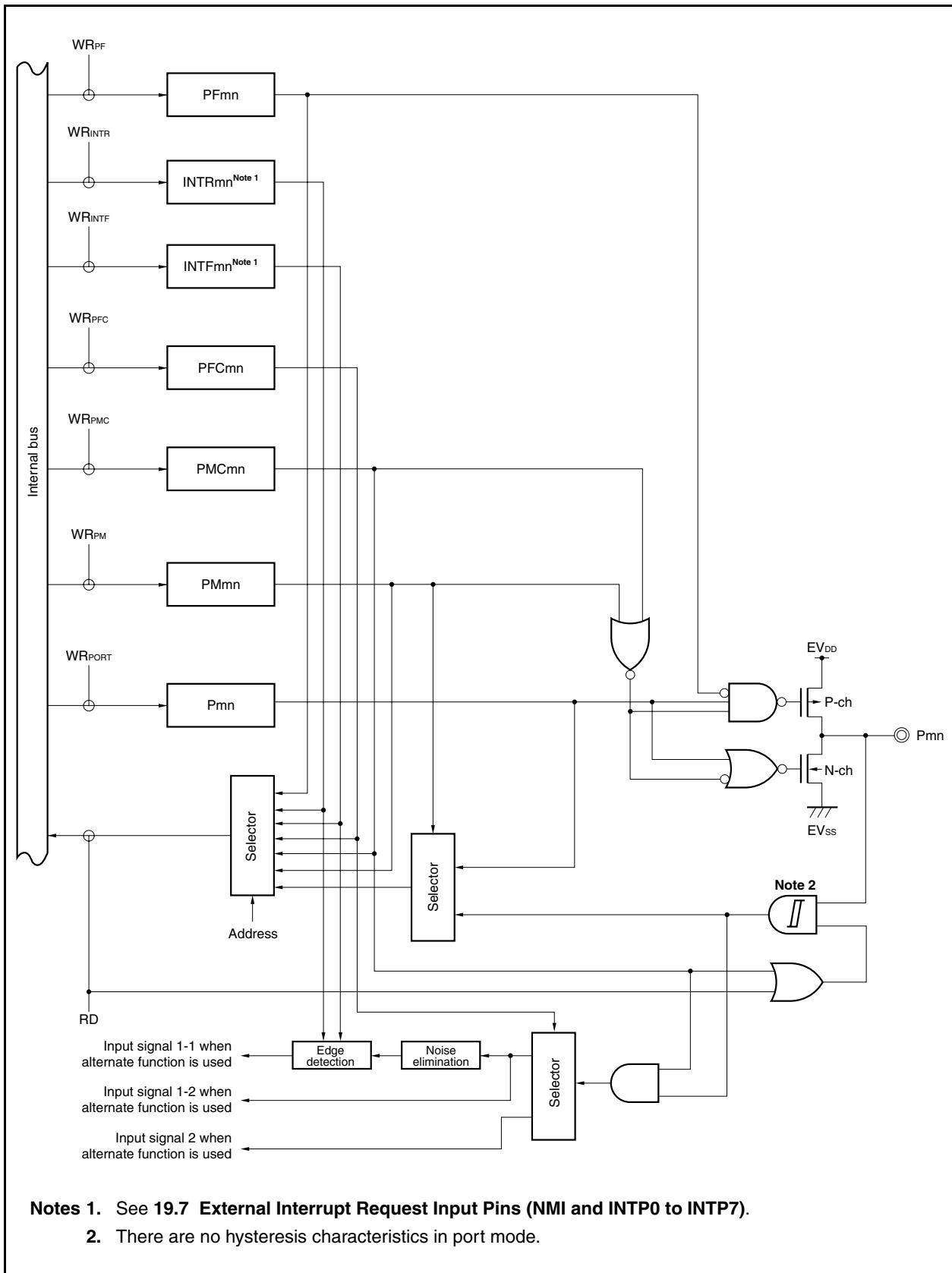


Figure 4-19. Block Diagram of Type N-3



- Notes 1.** See 19.7 External Interrupt Request Input Pins (NMI and INTP0 to INTP7).
- 2.** There are no hysteresis characteristics in port mode.

Figure 4-20. Block Diagram of Type N-4

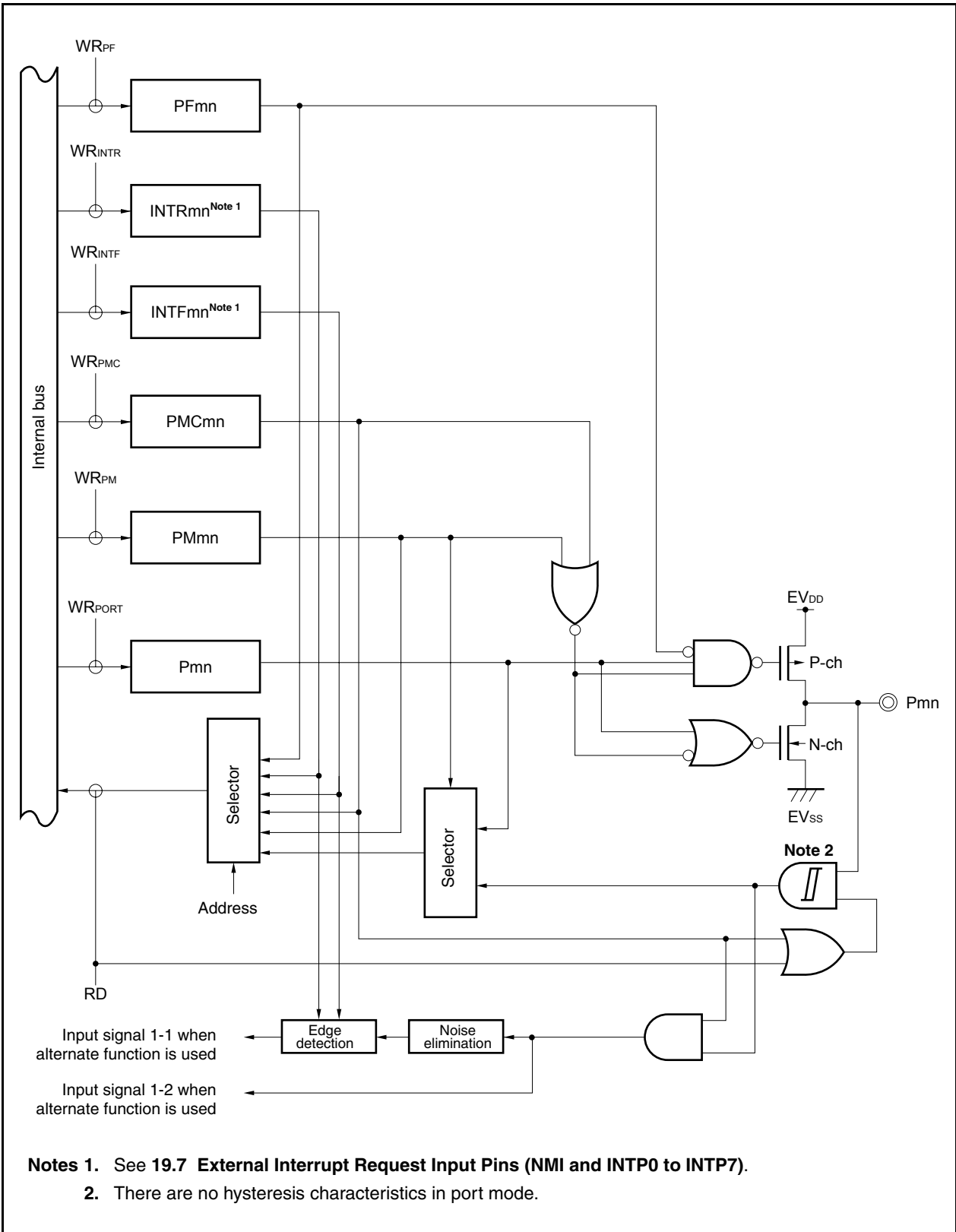


Figure 4-21. Block Diagram of Type N-5

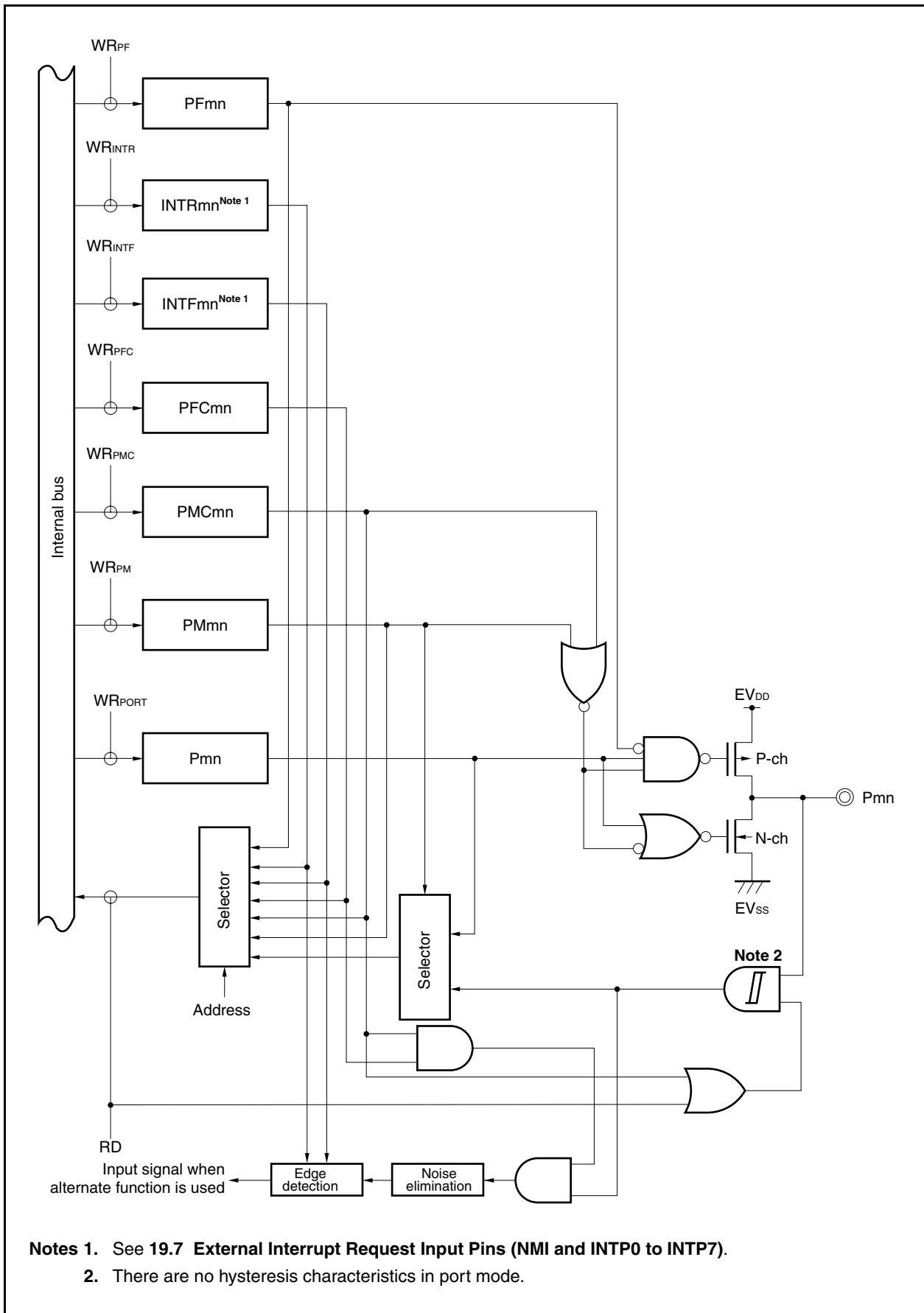


Figure 4-22. Block Diagram of Type U-1

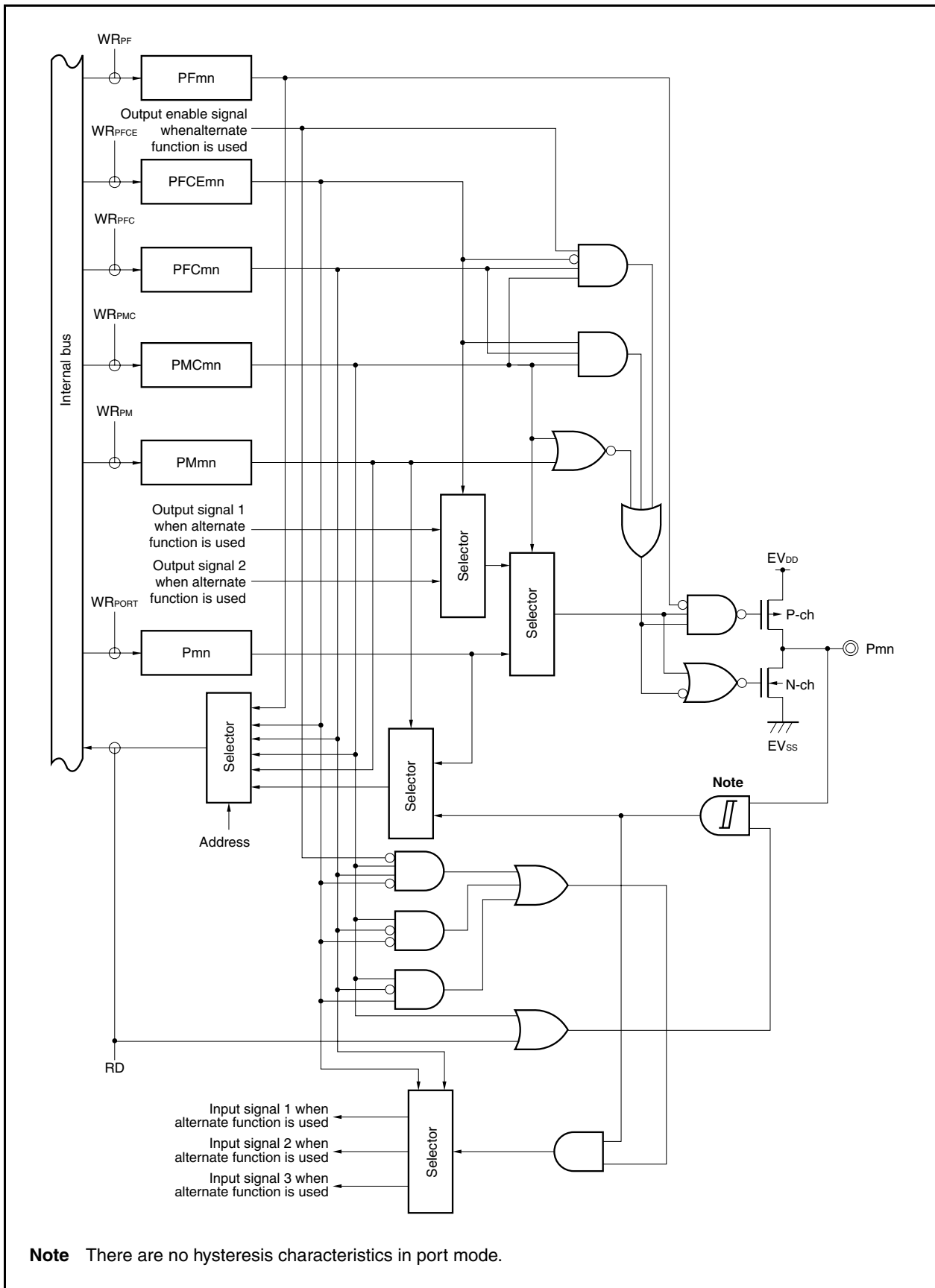


Figure 4-23. Block Diagram of Type U-5

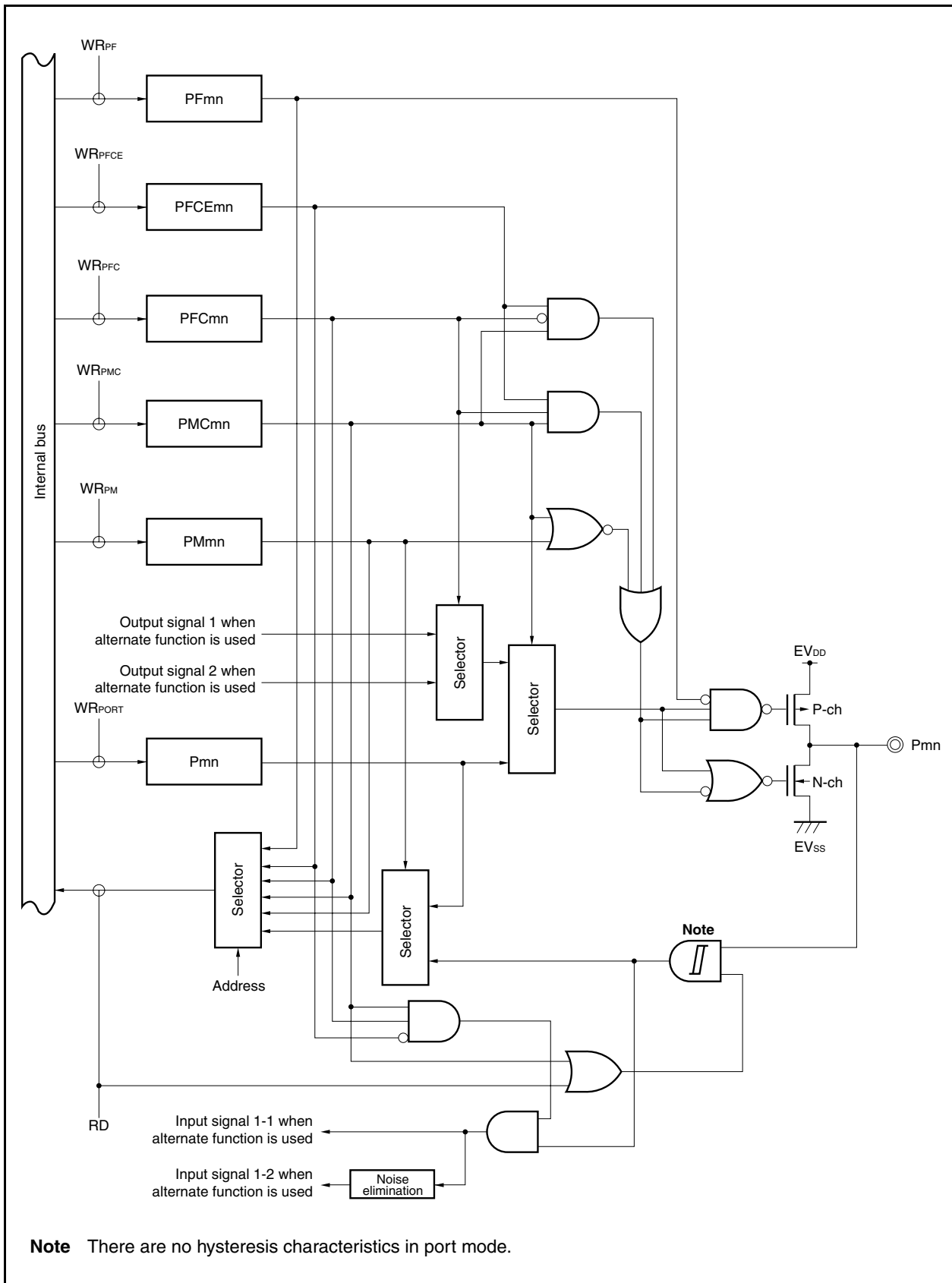


Figure 4-24. Block Diagram of Type U-6

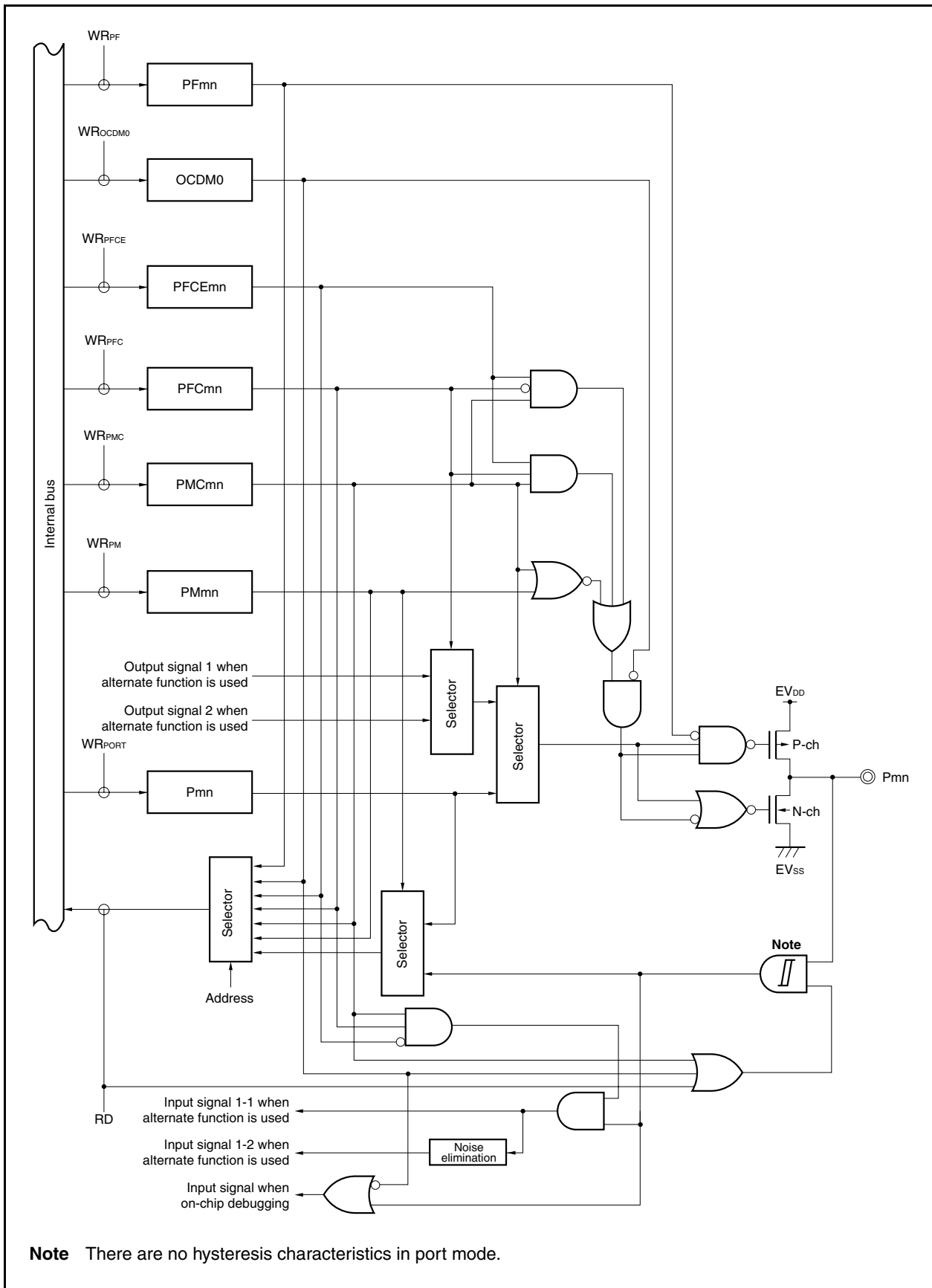


Figure 4-25. Block Diagram of Type U-7

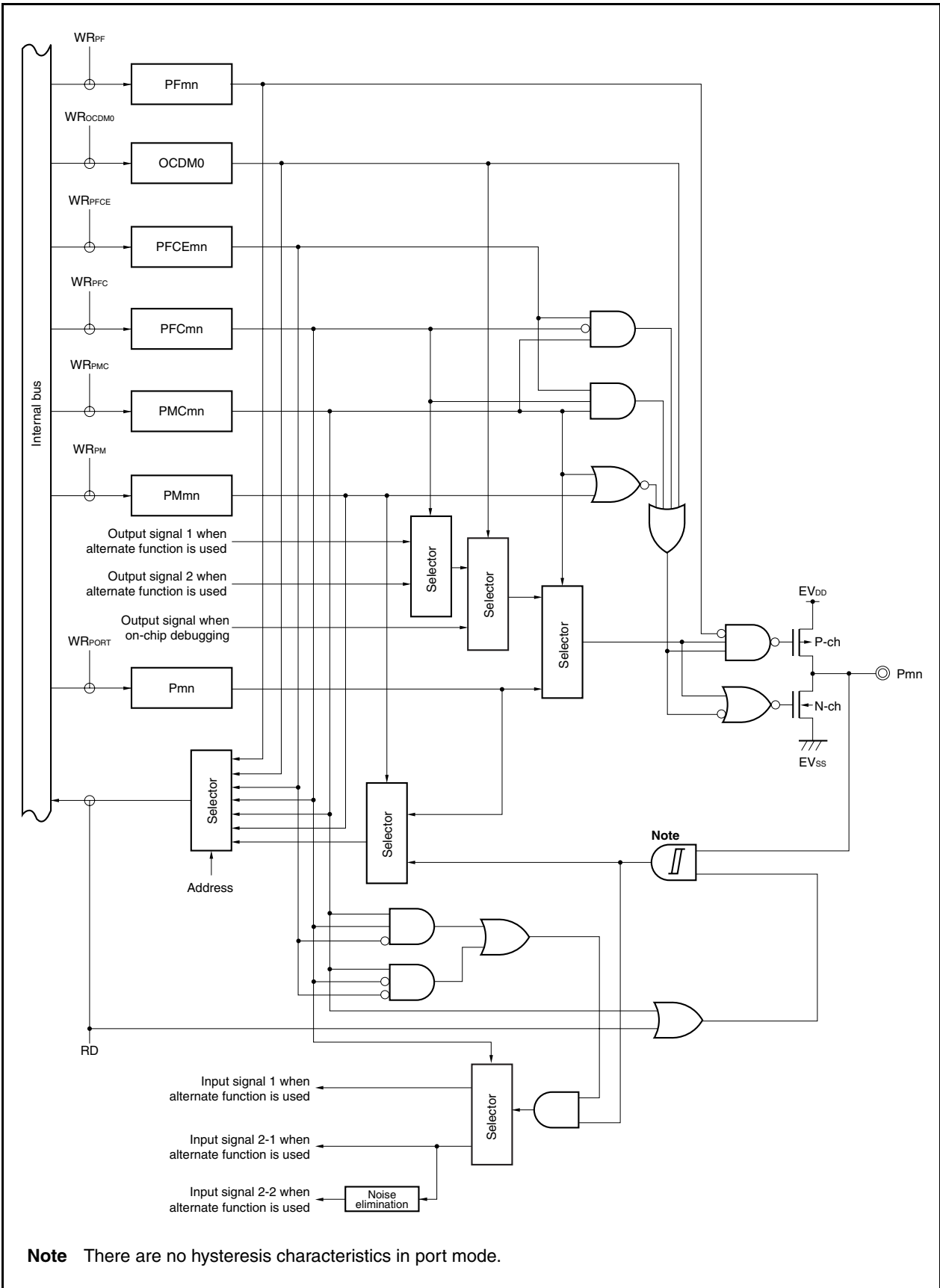


Figure 4-26. Block Diagram of Type U-8

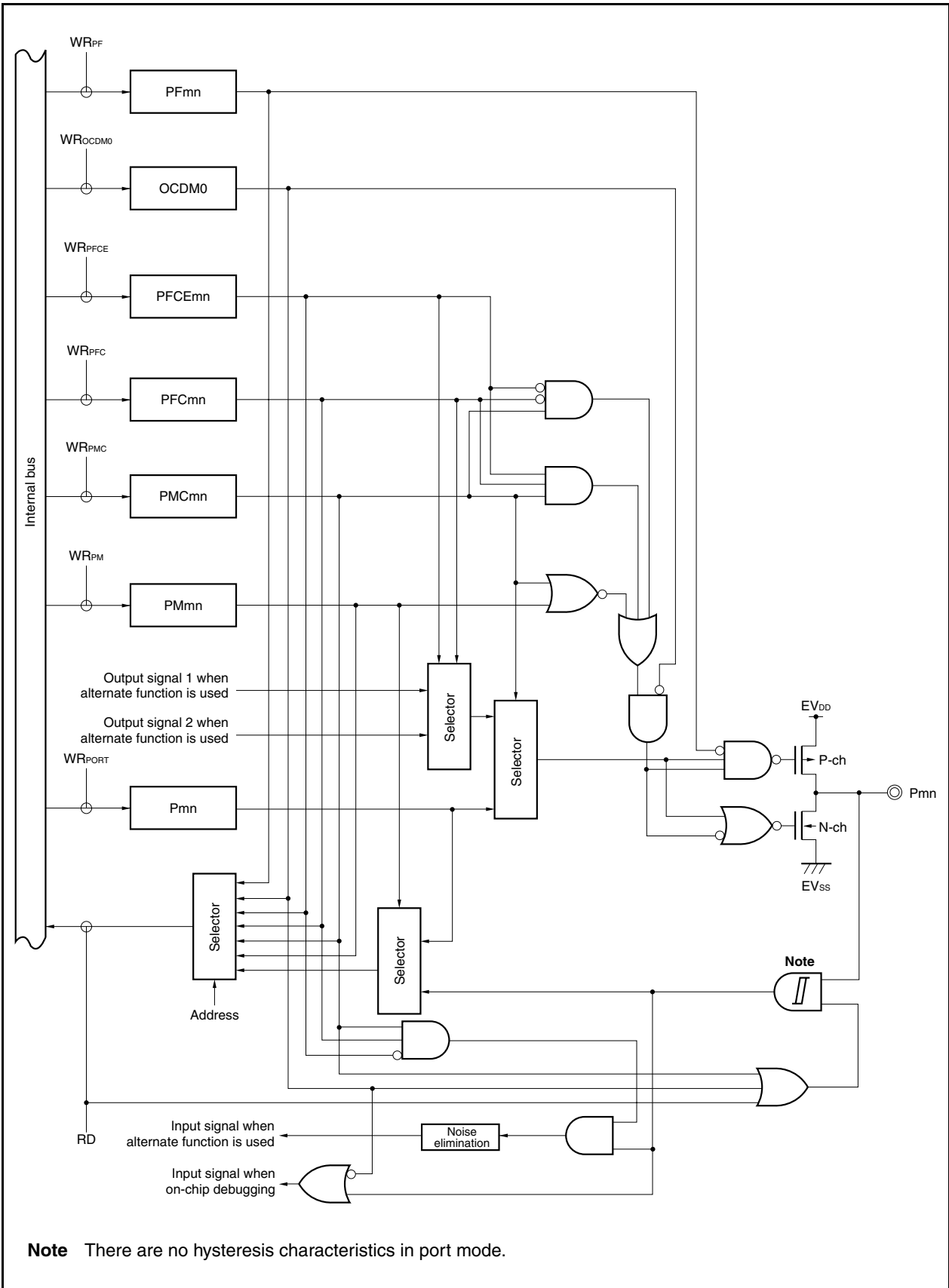
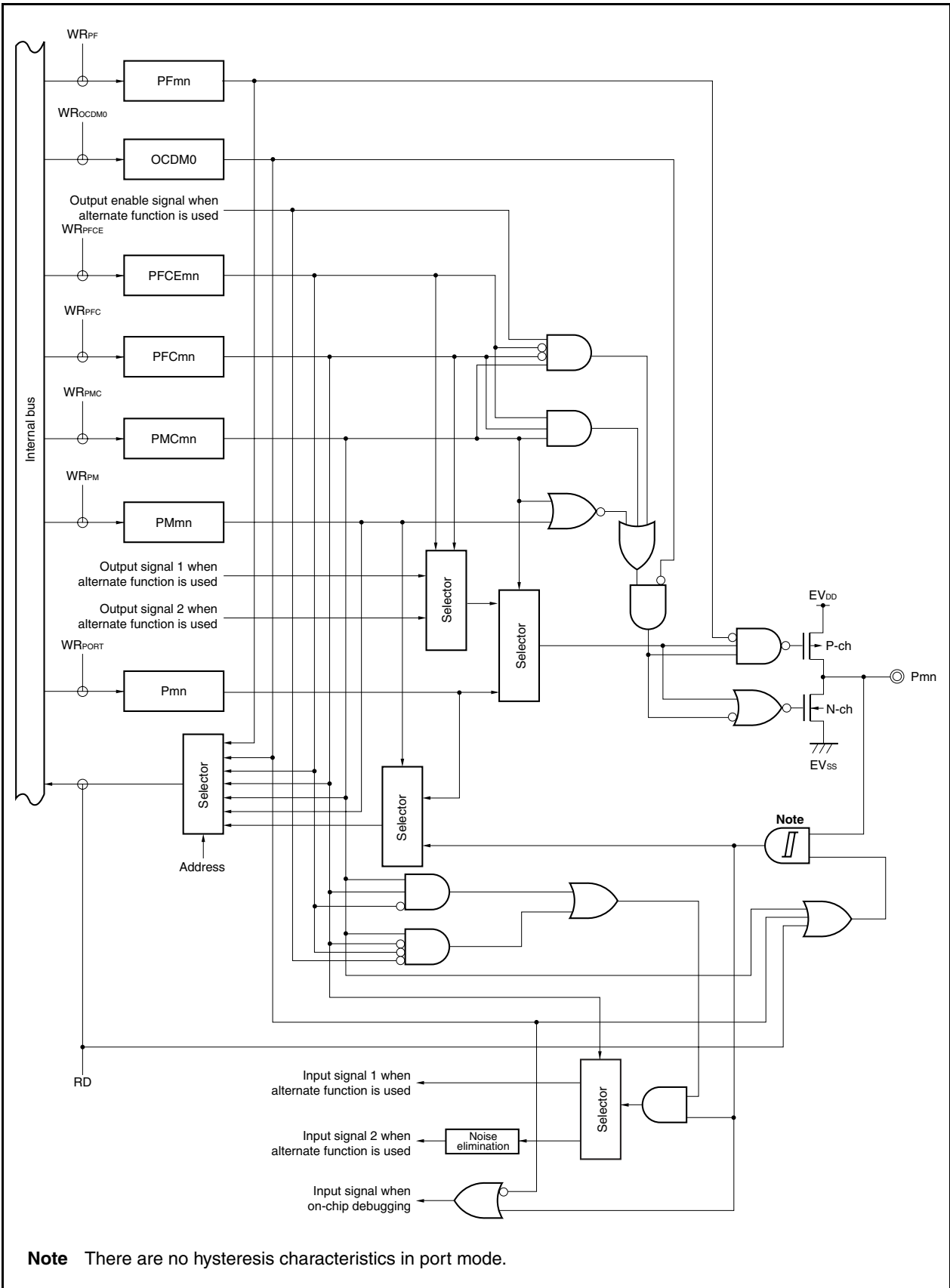


Figure 4-27. Block Diagram of Type U-9



Note There are no hysteresis characteristics in port mode.

Figure 4-28. Block Diagram of Type U-15

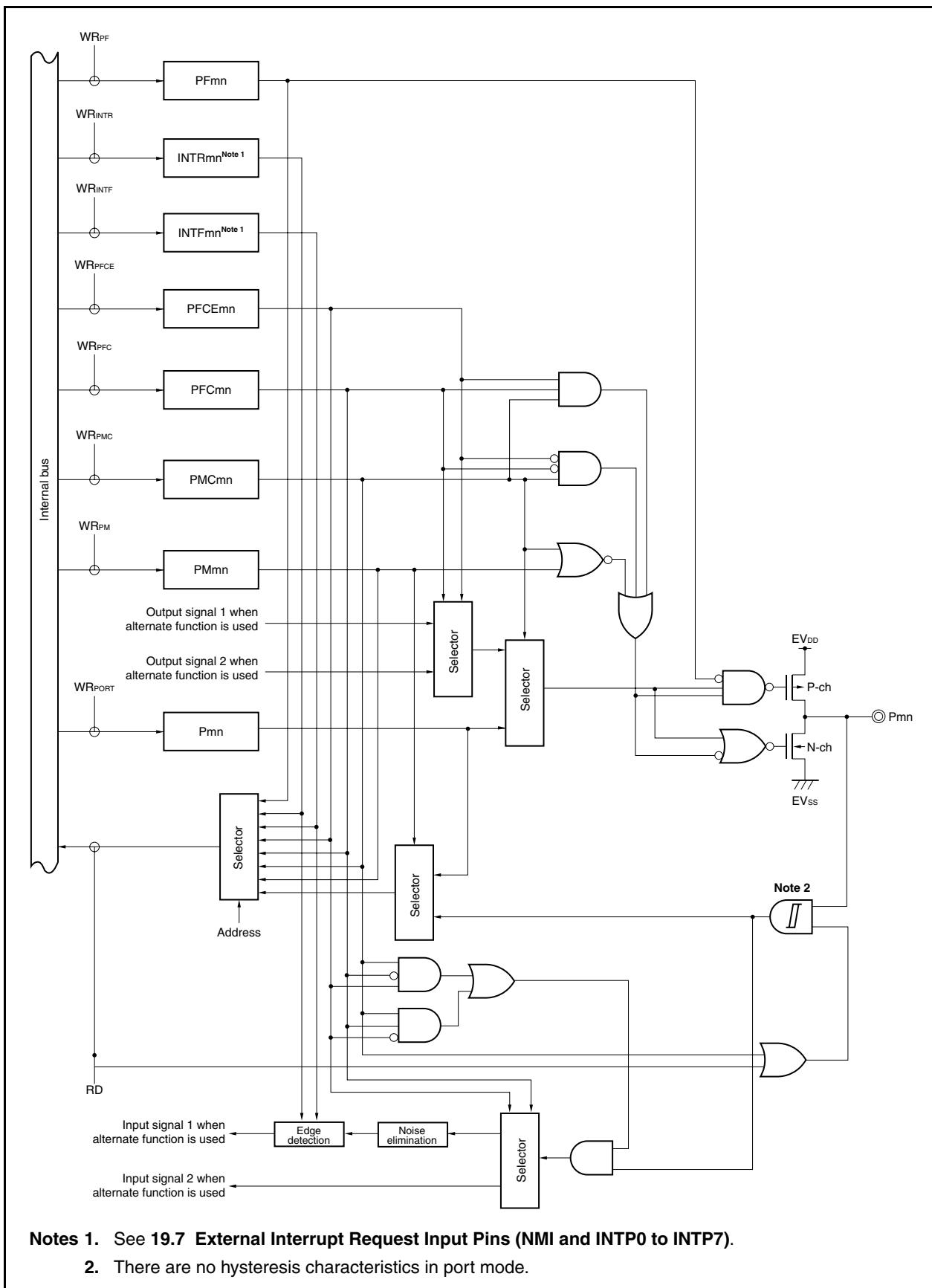


Figure 4-29. Block Diagram of Type U-18

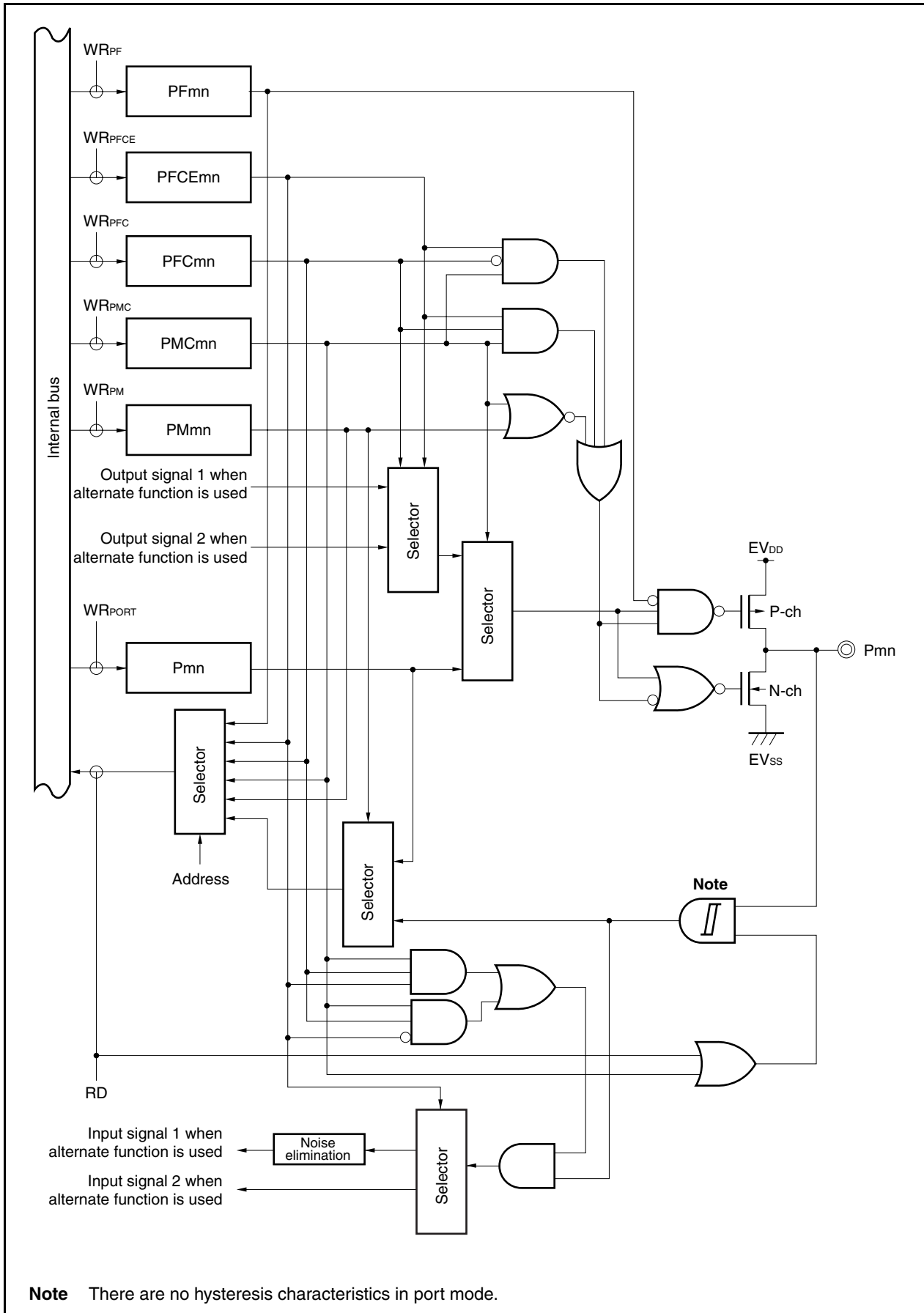


Figure 4-30. Block Diagram of Type U-19

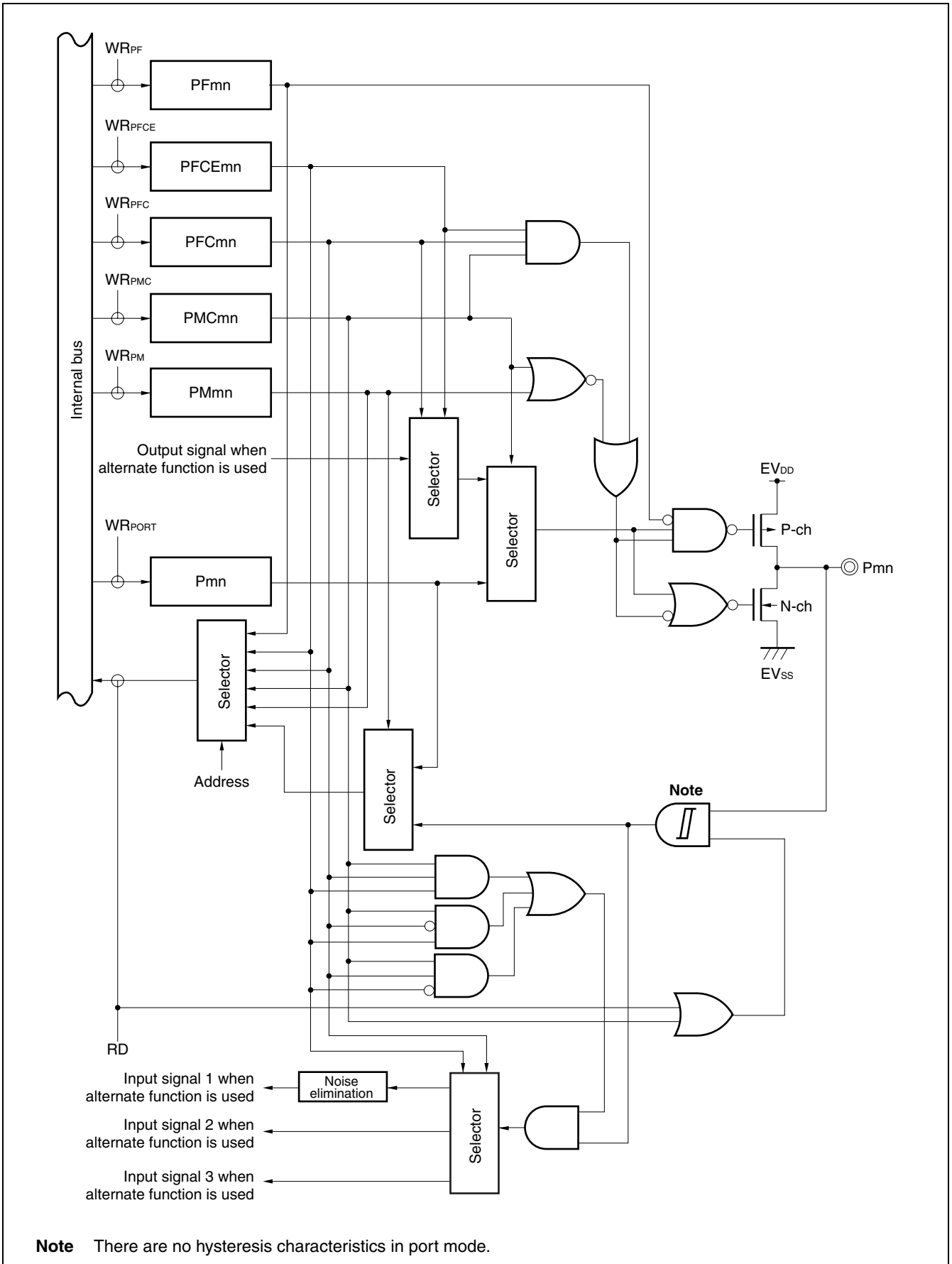


Figure 4-31. Block Diagram of Type U-20

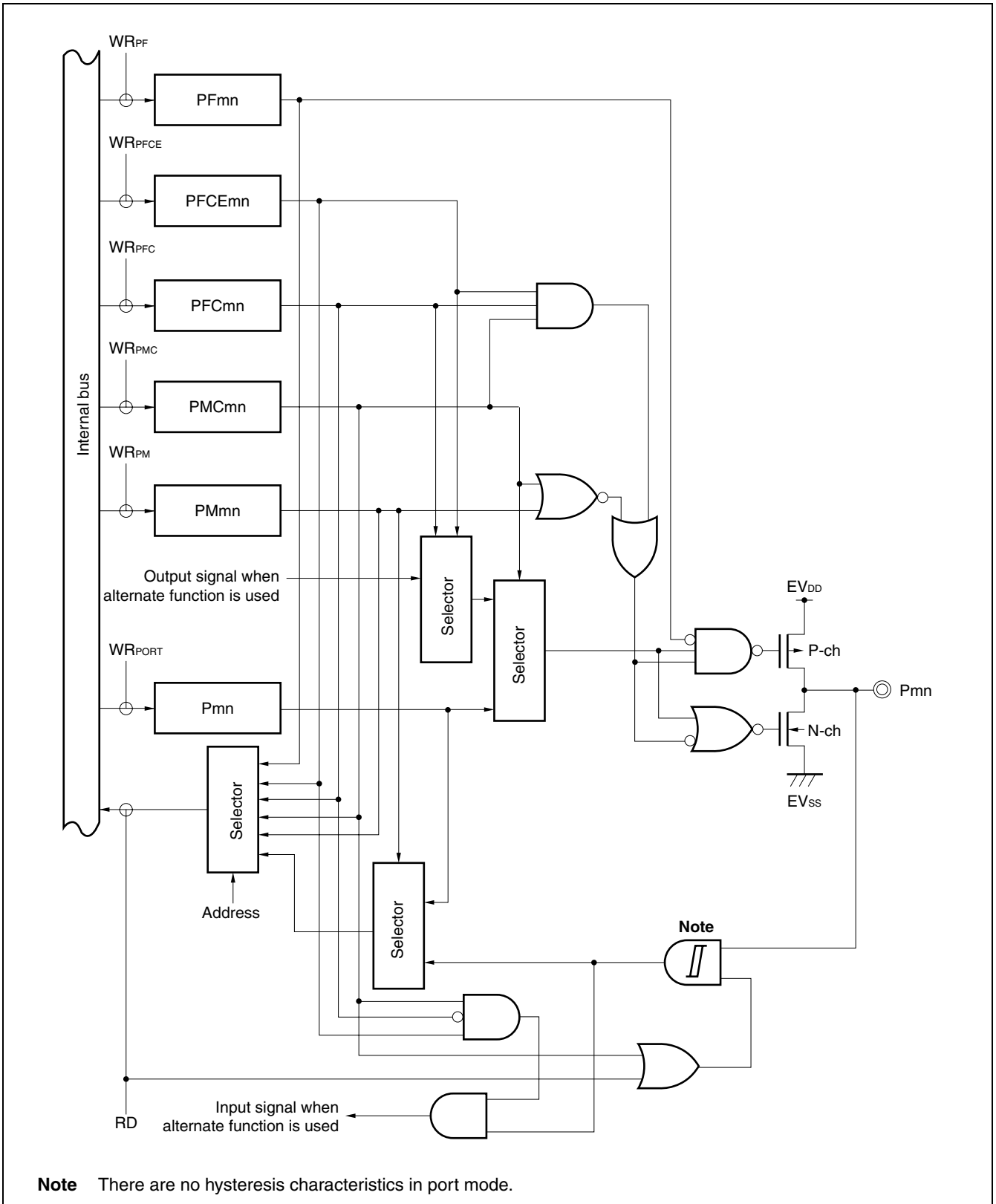


Figure 4-32. Block Diagram of Type U-21

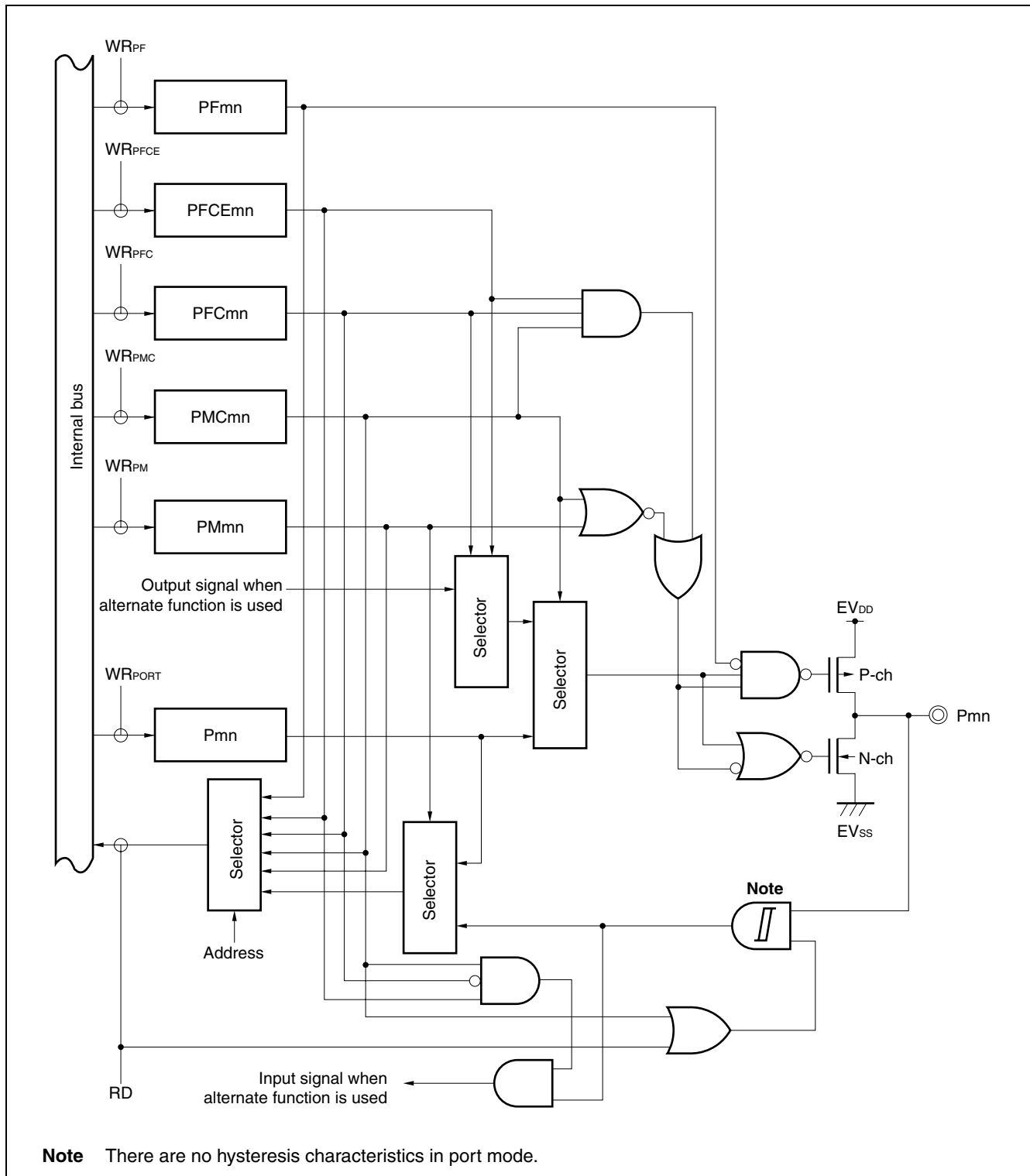


Figure 4-33. Block Diagram of Type U-22

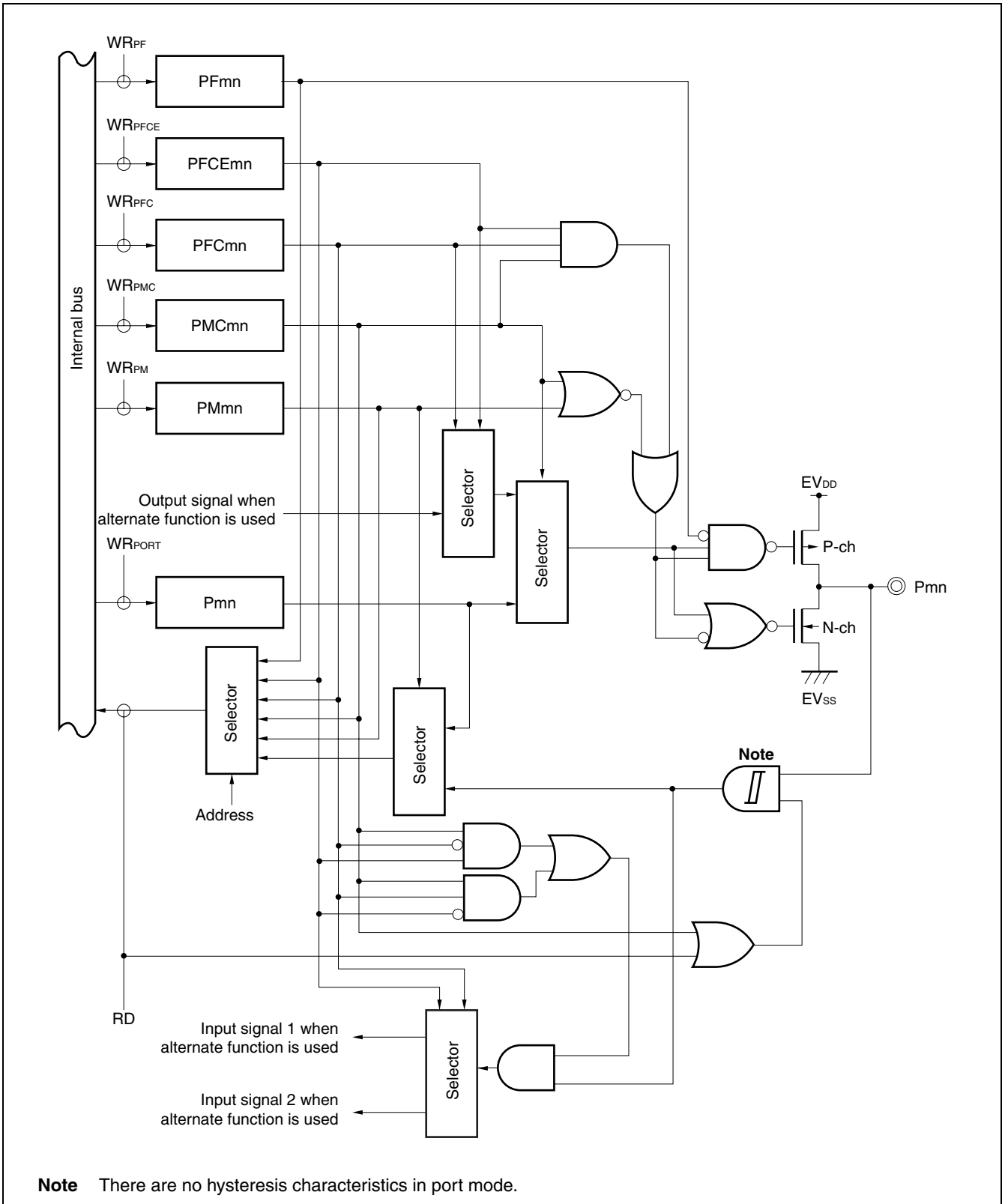
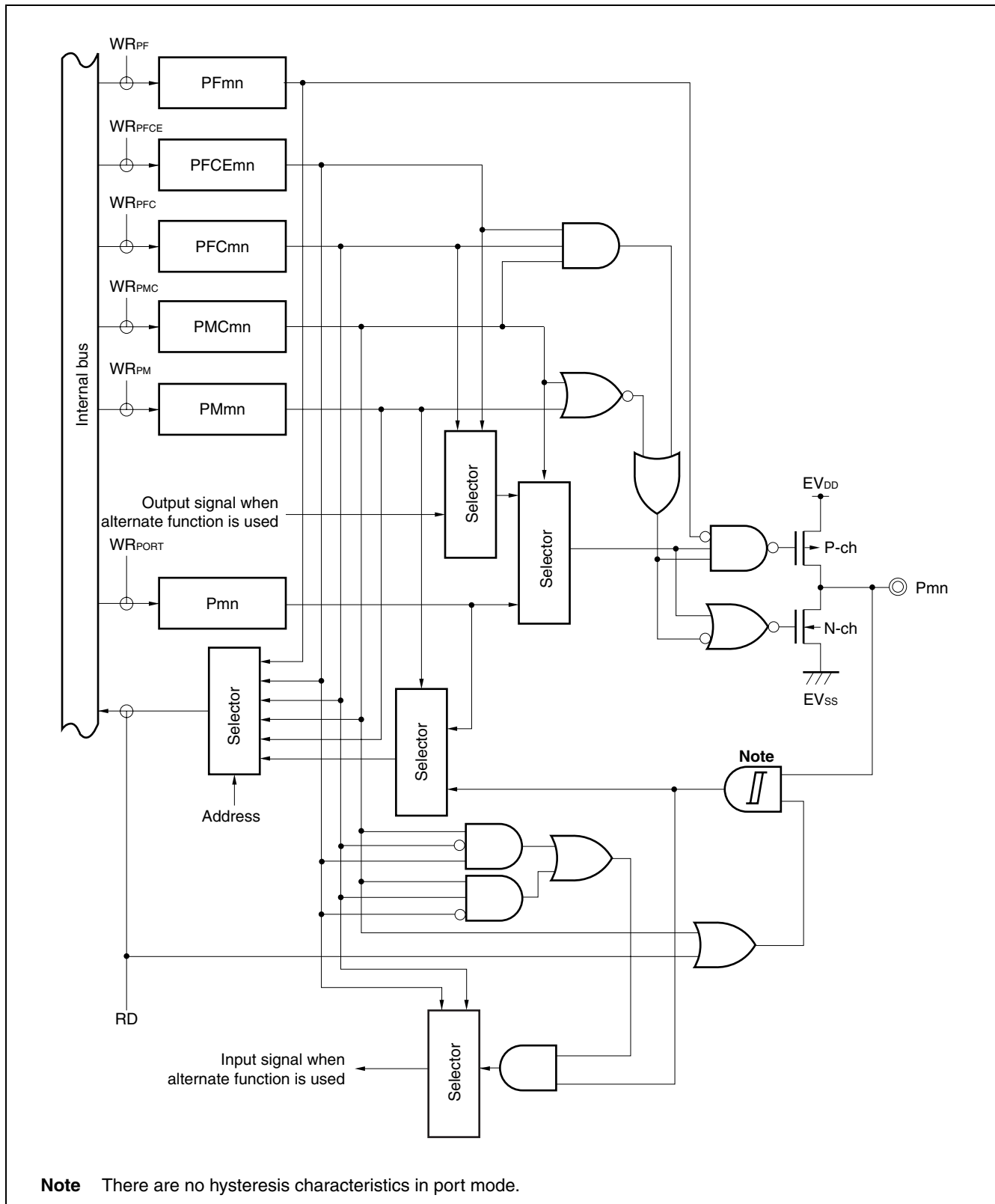


Figure 4-34. Block Diagram of Type U-23



Note There are no hysteresis characteristics in port mode.

Figure 4-35. Block Diagram of Type U-24

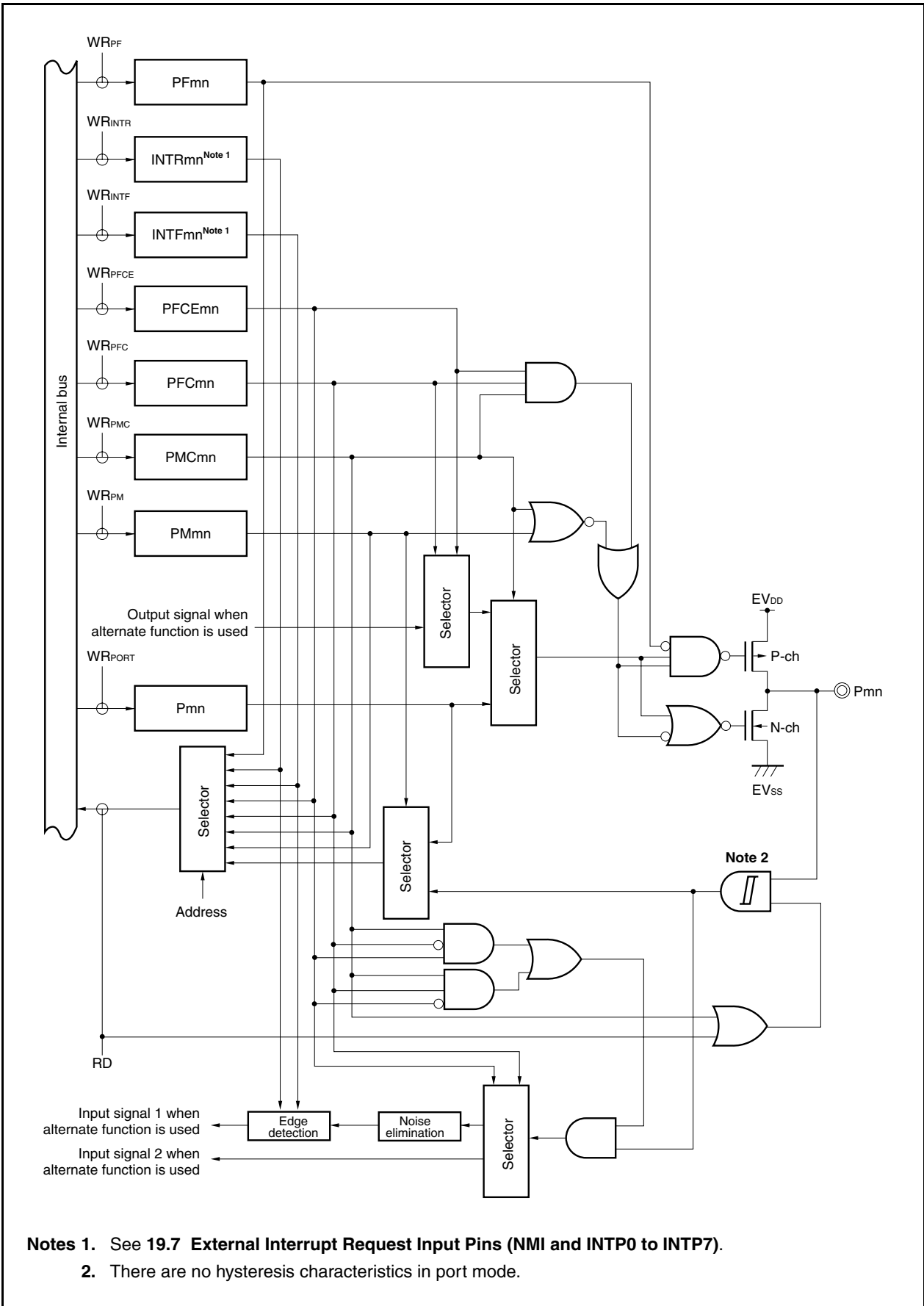
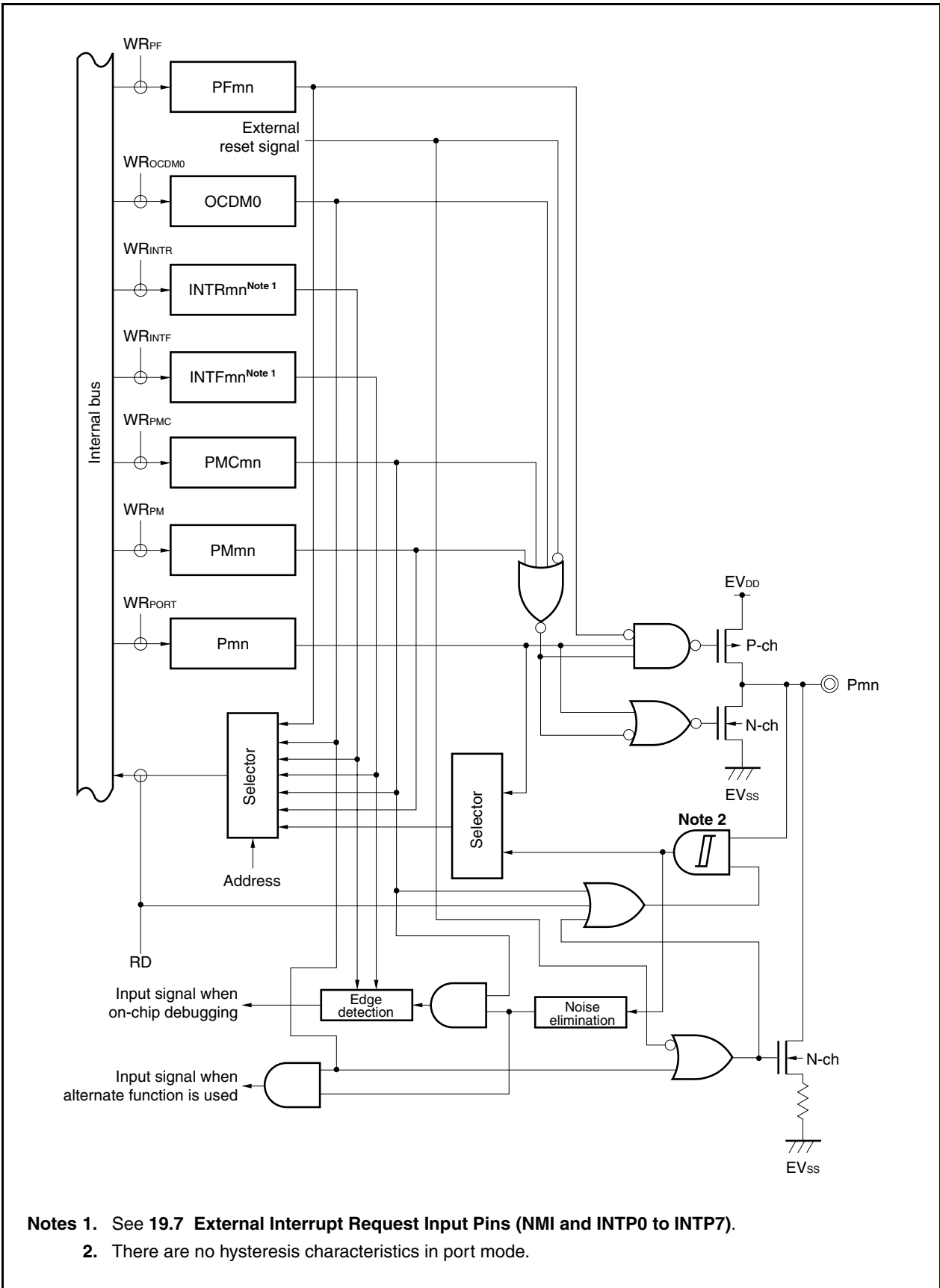


Figure 4-36. Block Diagram of Type AA-1



- Notes 1. See 19.7 External Interrupt Request Input Pins (NMI and INTP0 to INTP7).
- 2. There are no hysteresis characteristics in port mode.

4.5 Port Register Settings When Alternate Function Is Used

Table 4-17 shows the port register settings when each port pin is used for an alternate function. When using a port pin as an alternate-function pin, refer to the description of each pin.

Table 4-17. Settings When Pins Are Used for Alternate Functions (1/5)

Function 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						
P02	NMI	Input	P02 = Setting not required	PM02 = Setting not required	PMC02 = 1	-	-	
P03	INTP0	Input	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	PFCE03 = 0	PFC03 = 0	
	ADTRG	Input	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	PFCE03 = 0	PFC03 = 1	
	RTC1HZ	Input	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	PFCE03 = 1	PFC03 = 1	
P04	INTP1	Input	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	PFCE04 = 0	PFC04 = 0	
	RTCDIV	Output	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	PFCE04 = 0	PFC04 = 1	
	RTCCL	Output	P04 = Setting not required	PM04 = Setting not required	PMC04 = 1	PFCE04 = 1	PFC04 = 0	
P05	INTP2	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	-	-	
	DRST	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = Setting not required	-	-	OCDM0 (OCDM) = 1
P06	INTP3	Input	P06 = Setting not required	PM06 = Setting not required	PMC06 = 1	-	-	
P10	ANO0	Output	P10 = Setting not required	PM10 = 1	-	-	-	
P30	TXDA0	Output	P30 = Setting not required	PM30 = Setting not required	PMC30 = 1	-	PFC30 = 0	
	SOB4	Output	P30 = Setting not required	PM30 = Setting not required	PMC30 = 1	-	PFC30 = 1	
P31	RXDA0	Input	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	-	Note , PFC31 = 0	
	INTP7	Input	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	-	Note , PFC31 = 0	
	SIB4	Input	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	-	PFC31 = 1	
P32	ASCKA0	Input	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 0	PFC32 = 0	
	SCKB4	I/O	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 0	PFC32 = 1	
	TIP00	Input	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 1	PFC32 = 0	
	TOP00	Output	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 1	PFC32 = 1	
P33	TIP01	Input	P33 = Setting not required	PM33 = Setting not required	PMC33 = 1	-	PFC33 = 0	
	TOP01	Output	P33 = Setting not required	PM33 = Setting not required	PMC33 = 1	-	PFC33 = 1	
P34	TIP10	Input	P34 = Setting not required	PM34 = Setting not required	PMC34 = 1	-	PFC34 = 0	
	TOP10	Output	P34 = Setting not required	PM34 = Setting not required	PMC34 = 1	-	PFC34 = 1	
P35	TIP11	Input	P35 = Setting not required	PM35 = Setting not required	PMC35 = 1	-	PFC35 = 0	
	TOP11	Output	P35 = Setting not required	PM35 = Setting not required	PMC35 = 1	-	PFC35 = 1	

Note INTP7 and RXDA0 are alternate functions. When using the pin for RXDA0, disable edge detection for INTP7 (clear the INTF3.INTF31 bit and the INTR3.INTR31 bit to 0). When using the pin for INTP7, stop UARTA0 reception (clear the UA0CTL0.UA0RXE bit to 0).

Table 4-17. Settings When Pins Are Used for Alternate Functions (2/5)

Function 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						
P38	TXDA2	Output	P38 = Setting not required	PM38 = Setting not required	PMC38 = 1	–	PFC38 = 0	
	SDA00	I/O	P38 = Setting not required	PM38 = Setting not required	PMC38 = 1	–	PFC38 = 1	PF38 (PF3) = 1
P39	RXDA2	Input	P39 = Setting not required	PM39 = Setting not required	PMC39 = 1	–	PFC39 = 0	
	SCL00	I/O	P39 = Setting not required	PM39 = Setting not required	PMC39 = 1	–	PFC39 = 1	PF39 (PF3) = 1
P40	SIB0	Input	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	–	PFC40 = 0	
	SDA01	I/O	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	–	PFC40 = 1	PF40 (PF4) = 1
P41	SOB0	Output	P41 = Setting not required	PM41 = Setting not required	PMC41 = 1	–	PFC41 = 0	
	SCL01	I/O	P41 = Setting not required	PM41 = Setting not required	PMC41 = 1	–	PFC41 = 1	PF41 (PF4) = 1
P42	SCKB0	I/O	P42 = Setting not required	PM42 = Setting not required	PMC42 = 1	–	–	
P50	TIQ01	Input	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 0	PFC50 = 1	KRM0 (KRM) = 0
	KR0	Input	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 0	PFC50 = 1	TQ0TIG2,TQ0TIG3 (TQ0IOC1) = 0
	TOQ01	Output	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 1	PFC50 = 0	
	RTP00	Output	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 1	PFC50 = 1	
P51	TIQ02	Input	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 0	PFC51 = 1	KRM1 (KRM) = 0
	KR1	Input	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 0	PFC51 = 1	TQ0TIG4,TQ0TIG5 (TQ0IOC1) = 0
	TOQ02	Output	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 1	PFC51 = 0	
	RTP01	Output	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 1	PFC51 = 1	
P52	TIQ03	Input	P52 = Setting not required	PM52 = Setting not required	PMC52 = 1	PFCE52 = 0	PFC52 = 1	KRM2 (KRM) = 0
	KR2	Input	P52 = Setting not required	PM52 = Setting not required	PMC52 = 1	PFCE52 = 0	PFC52 = 1	TQ0TIG6,TQ0TIG7 (TQ0IOC1) = 0
	TOQ03	Output	P52 = Setting not required	PM52 = Setting not required	PMC52 = 1	PFCE52 = 1	PFC52 = 0	
	RTP02	Output	P52 = Setting not required	PM52 = Setting not required	PMC52 = 1	PFCE52 = 1	PFC52 = 1	
	DDI	Input	P52 = Setting not required	PM52 = Setting not required	PMC52 = Setting not required	PMCE52 = Setting not required	PFC52 = Setting not required	OCDM0 (OCDM) = 1

Table 4-17. Settings When Pins Are Used for Alternate Functions (3/5)

Function 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						
P53	SIB2	Input	P53 = Setting not required	PM53 = Setting not required	PMC53 = 1	PFCE53 = 0	PFC53 = 0	
	TIQ00	Input	P53 = Setting not required	PM53 = Setting not required	PMC53 = 1	PFCE53 = 0	PFC53 = 1	KRM3 (KRM) = 0
	KR3	Input	P53 = Setting not required	PM53 = Setting not required	PMC53 = 1	PFCE53 = 0	PFC53 = 1	TQ0TIG0, TQ0TIG1 (TQ0IOC1) = 0, TQ0EES0, TQ0EES1 (TQ0IOC2) = 0, TQ0ETS0, TQ0ETS1 (TQ0IOC2) = 0
	TOQ00	Output	P53 = Setting not required	PM53 = Setting not required	PMC53 = 1	PFCE53 = 1	PFC53 = 0	
	RTP03	Output	P53 = Setting not required	PM53 = Setting not required	PMC53 = 1	PFCE53 = 1	PFC53 = 1	
	DDO	Output	P53 = Setting not required	PM53 = Setting not required	PMC53 = Setting not required	PFCE53 = Setting not required	PFC53 = Setting not required	OCDM0 (OCDM) = 1
P54	SOB2	Output	P54 = Setting not required	PM54 = Setting not required	PMC54 = 1	PFCE54 = 0	PFC54 = 0	
	KR4	Input	P54 = Setting not required	PM54 = Setting not required	PMC54 = 1	PFCE54 = 0	PFC54 = 1	
	RTP04	Output	P54 = Setting not required	PM54 = Setting not required	PMC54 = 1	PFCE54 = 1	PFC54 = 1	
	DCK	Input	P54 = Setting not required	PM54 = Setting not required	PMC54 = Setting not required	PFCE54 = Setting not required	PFC54 = Setting not required	OCDM0 (OCDM) = 1
P55	SCKB2	I/O	P55 = Setting not required	PM55 = Setting not required	PMC55 = 1	PFCE55 = 0	PFC55 = 0	
	KR5	Input	P55 = Setting not required	PM55 = Setting not required	PMC55 = 1	PFCE55 = 0	PFC55 = 1	
	RTP05	Output	P55 = Setting not required	PM55 = Setting not required	PMC55 = 1	PFCE55 = 1	PFC55 = 1	
	DMS	Input	P55 = Setting not required	PM55 = Setting not required	PMC55 = Setting not required	PFCE55 = Setting not required	PFC55 = Setting not required	OCDM0 (OCDM) = 1

Table 4-17. Settings When Pins Are Used for Alternate Functions (4/5)

Function 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	PM70 = 1	–	–	–	
P71	ANI1	Input	P71 = Setting not required	PM71 = 1	–	–	–	
P72	ANI2	Input	P72 = Setting not required	PM72 = 1	–	–	–	
P73	ANI3	Input	P73 = Setting not required	PM73 = 1	–	–	–	
P74	ANI4	Input	P74 = Setting not required	PM74 = 1	–	–	–	
P75	ANI5	Input	P75 = Setting not required	PM75 = 1	–	–	–	
P76	ANI6	Input	P76 = Setting not required	PM76 = 1	–	–	–	
P77	ANI7	Input	P77 = Setting not required	PM77 = 1	–	–	–	
P78	ANI8	Input	P78 = Setting not required	PM78 = 1	–	–	–	
P79	ANI9	Input	P79 = Setting not required	PM79 = 1	–	–	–	
P90	KR6	Input	P90 = Setting not required	PM90 = Setting not required	PMC90 = 1	PFCE90 = 0	PFC90 = 1	
	TXDA1	Output	P90 = Setting not required	PM90 = Setting not required	PMC90 = 1	PFCE90 = 1	PFC90 = 0	
	SDA02	I/O	P90 = Setting not required	PM90 = Setting not required	PMC90 = 1	PFCE90 = 1	PFC90 = 1	PF90 (PF9) = 1
P91	KR7	Input	P91 = Setting not required	PM91 = Setting not required	PMC91 = 1	PFCE91 = 0	PFC91 = 1	
	RXDA1/KR7 ^{Note}	Input	P91 = Setting not required	PM91 = Setting not required	PMC91 = 1	PFCE91 = 1	PFC91 = 0	
	SCL02	I/O	P91 = Setting not required	PM91 = Setting not required	PMC91 = 1	PFCE91 = 1	PFC91 = 1	PF91 (PF9) = 1
P92	TIP41	Input	P92 = Setting not required	PM92 = Setting not required	PMC92 = 1	PFCE92 = 0	PFC92 = 1	
	TOP41	Output	P92 = Setting not required	PM92 = Setting not required	PMC92 = 1	PFCE92 = 1	PFC92 = 0	
P93	TIP40	Input	P93 = Setting not required	PM93 = Setting not required	PMC93 = 1	PFCE93 = 0	PFC93 = 1	
	TOP40	Output	P93 = Setting not required	PM93 = Setting not required	PMC93 = 1	PFCE93 = 1	PFC93 = 0	
P94	TIP31	Input	P94 = Setting not required	PM94 = Setting not required	PMC94 = 1	PFCE94 = 0	PFC94 = 1	
	TOP31	Output	P94 = Setting not required	PM94 = Setting not required	PMC94 = 1	PFCE94 = 1	PFC94 = 0	

Note. The RXDA1 and KR7 functions cannot be used at the same time. When using the pin for RXDA1, do not use the KR7 function. When using the pin for KR7, do not use the RXDA1 function. (It is recommended to set the PFC91 bit to 1 and clear the PFCE91 bit to 0.)

Table 4-17. Settings When Pins Are Used for Alternate Functions (5/5)

Function 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						
P96	TIP21	Input	P96 = Setting not required	PM96 = Setting not required	PMC96 = 1	PFCE96 = 1	PFC96 = 0	
	TOP21	Output	P96 = Setting not required	PM96 = Setting not required	PMC96 = 1	PFCE96 = 1	PFC96 = 1	
P97	TIP20	Input	P97 = Setting not required	PM97 = Setting not required	PMC97 = 1	PFCE97 = 1	PFC97 = 0	
	TOP20	Output	P97 = Setting not required	PM97 = Setting not required	PMC97 = 1	PFCE97 = 1	PFC97 = 1	
P98	SOB1	Output	P98 = Setting not required	PM98 = Setting not required	PMC98 = 1	–	PFC98 = 1	
P99	SCKB1	I/O	P99 = Setting not required	PM99 = Setting not required	PMC99 = 1	–	PFC99 = 1	
P910	SIB3	Input	P910 = Setting not required	PM910 = Setting not required	PMC910 = 1	–	PFC910 = 1	
P911	SOB3	Output	P911 = Setting not required	PM911 = Setting not required	PMC911 = 1	–	PFC911 = 1	
P912	SCKB3	I/O	P912 = Setting not required	PM912 = Setting not required	PMC912 = 1	–	PFC912 = 1	
P913	INTP4	Input	P913 = Setting not required	PM913 = Setting not required	PMC913 = 1	–	PFC913 = 1	
P914	INTP5	Input	P914 = Setting not required	PM914 = Setting not required	PMC914 = 1	PFCE914 = 0	PFC914 = 1	
	TIP51	Input	P914 = Setting not required	PM914 = Setting not required	PMC914 = 1	PFCE914 = 1	PFC914 = 0	
	TOP51	Output	P914 = Setting not required	PM914 = Setting not required	PMC914 = 1	PFCE914 = 1	PFC914 = 1	
P915	INTP6	Input	P915 = Setting not required	PM915 = Setting not required	PMC915 = 1	PFCE915 = 0	PFC915 = 1	
	TIP50	Input	P915 = Setting not required	PM915 = Setting not required	PMC915 = 1	PFCE915 = 1	PFC915 = 0	
	TOP50	Output	P915 = Setting not required	PM915 = Setting not required	PMC915 = 1	PFCE915 = 1	PFC915 = 1	
PDL5	FLMD1 ^{Note}	Input	PDL5 = Setting not required	PMDL5 = Setting not required	PMCDL5 = Setting not required	–	–	

Note Since this pin is set in the flash memory programming mode, it does not need to be manipulated by using the port control register. For details, see **CHAPTER 28 FLASH MEMORY**.

4.6 Cautions

4.6.1 Cautions on setting port pins

(1) In the V850ES/JC3-L and V850ES/JE3-L, general-purpose port pins are shared with several peripheral I/O functions. To switch between using a pin as a general-purpose port pin (port mode) and as a peripheral function I/O pin (alternate-function mode), use the PMCN register. Note the following when setting this register.

(a) Cautions when switching from port mode to alternate-function mode

Switch from the port mode to the alternate-function mode in the following order:

- | | |
|---|-----------------------------------|
| <1> Set the PFn register ^{Note 1} : | N-ch open-drain setting |
| <2> Set the PFCn and PFCEn registers: | Alternate-function selection |
| <3> Set the corresponding bit of the PMCN register to 1: | Switch to alternate-function mode |
| <4> Set the INTRn and INTFn registers ^{Note 2} : | External interrupt setting |

Note that if the PMCN register is set first, an unexpected operation may occur at the moment the register is set or when the pin states change in accordance with the setting of the PFn, PFCn, and PFCEn registers.

A specific example is shown below.

Notes 1. N-ch open-drain output pin only.

2. Only when the external interrupt function is selected.

Caution Regardless of the port mode/alternate-function mode setting, the Pn register is read and written as follows:

- Pn register read: The port output latch value is read (when PMn.PMnm bit = 0), or the pin state is read (PMn.PMnm bit = 1).
- Pn register write: The port output latch is written

[Example] SCL01 pin setting example

The SCL01 pin is used alternately as the P41/SOB0 pin. Select the desired pin function by using the PMC4, PFC4, and PF4 registers.

PMC41 Bit	PFC41 Bit	PF41 Bit	Pin Function
0	don't care	1	P41 (in output port mode, N-ch open-drain output)
1	0	1	SOB0 output (N-ch open-drain output)
	1	1	SCL01 I/O (N-ch open-drain output)

The setting order that may cause a malfunction when switching from the P41 pin function to the SCL01 pin function is shown below.

Setting Order	Setting	Pin State	Pin Level
<1>	Initial value (PMC41 bit = 0, PFC41 bit = 0, PF41 bit = 0)	Port mode (input)	Hi-Z
<2>	PMC41 bit ← 1	SOB0 output	Low level (may be high level depending on the CSIB0 setting)
<3>	PFC41 bit ← 1	SCL01 I/O	High level (CMOS output)
<4>	PF41 bit ← 1	SCL01 I/O	Hi-Z (N-ch open-drain output)

In <2>, I²C communication may be affected since the alternate-function SOB0 output is output to the pin. In the CMOS output period of <2> or <3>, an unnecessary current may be generated.

(b) Cautions on alternate-function mode (input)

The signal input to the alternate-function block is low level when the PMCn.PMCnm bit is 0 due to the ANDed output of the PMCn register set value and the pin level. Thus, depending on the port setting and alternate-function operation enable timing, an unexpected operation may occur. Therefore, switch between the port mode and alternate-function mode in the following sequence.

- Switching from port mode to alternate-function mode (input)
Set the pins to the alternate-function mode using the PMCn register and then enable the alternate-function operation.
- Switching from alternate-function mode (input) to port mode
Stop the alternate-function operation and then switch the pins to the port mode.

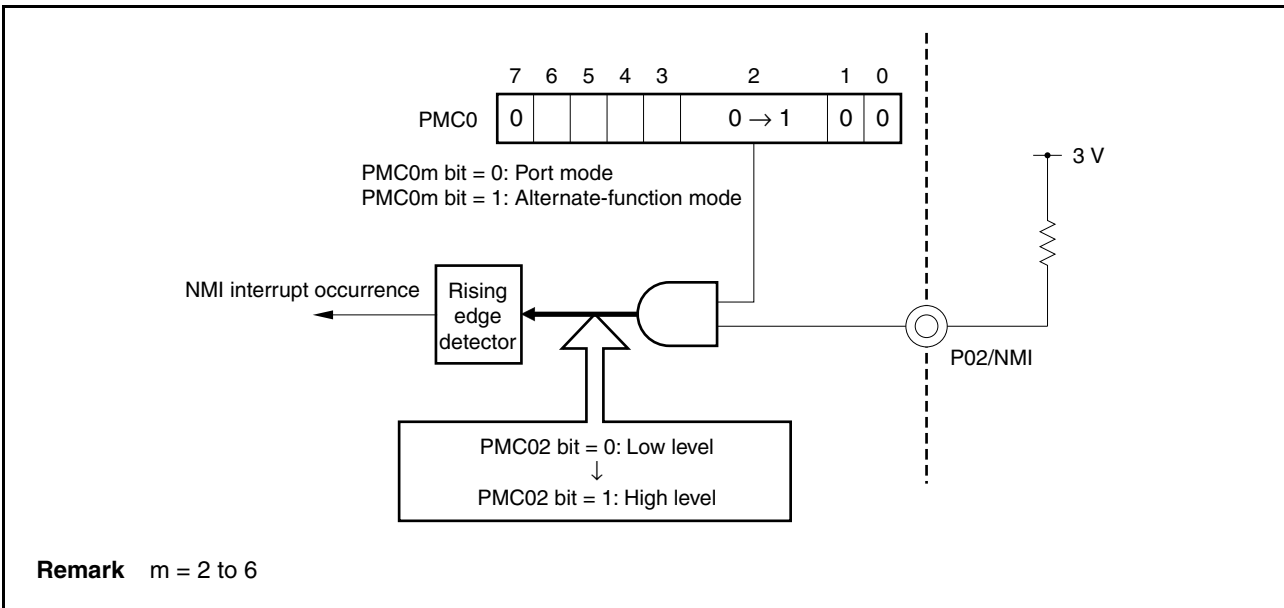
Specific examples are shown below.

[Example 1] Switching from general-purpose port pin (P02) to external interrupt pin (NMI)

When the P02/NMI pin is pulled up as shown in Figure 4-37 and the rising edge is specified by the NMI pin edge detection setting, even though a high level is input continuously to the NMI pin while switching from the P02 pin to the NMI pin (PMC02 bit = 0 → 1), this is detected as a rising edge, as if the low level changed to a high level, and an NMI interrupt occurs.

To avoid this, set the NMI pin's valid edge after switching from the P02 pin to the NMI pin.

Figure 4-37. Example of Switching from P02 to NMI (Incorrect)

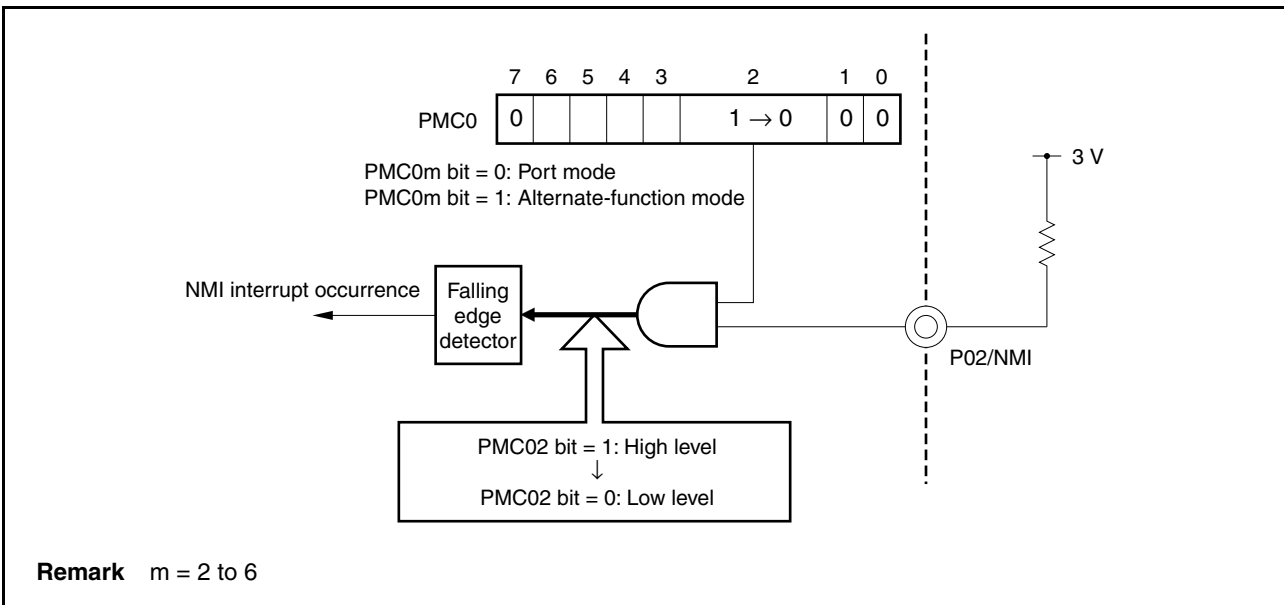


[Example 2] Switching from external pin (NMI) to general-purpose port pin (P02)

When the P02/NMI pin is pulled up as shown in Figure 4-38 and the falling edge is specified by the NMI pin edge detection setting, even though a high level is input continuously to the NMI pin when switching from the NMI pin to the P02 pin (PMC02 bit = 1 → 0), this is detected as a falling edge, as if the high level changed to a low level, and an NMI interrupt occurs.

To avoid this, set the NMI pin edge detection as “No edge detected” before switching to the P02 pin.

Figure 4-38. Example of Switching from NMI to P02 (Incorrect)



(2) In port mode, the PFn.PFn_m bit is valid only in the output mode (PMn.PMn_m bit = 0). In the input mode (PMn_m bit = 1), the value of the PFn_m bit is not reflected in the buffer.

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

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

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

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

Explanation: When writing to and reading from the Pn register of a port whose PMnm bit is 1, the output latch is written and the pin status is read.

A bit manipulation instruction is executed in the following order in the V850ES/JC3-L, V850ES/JE3-L.

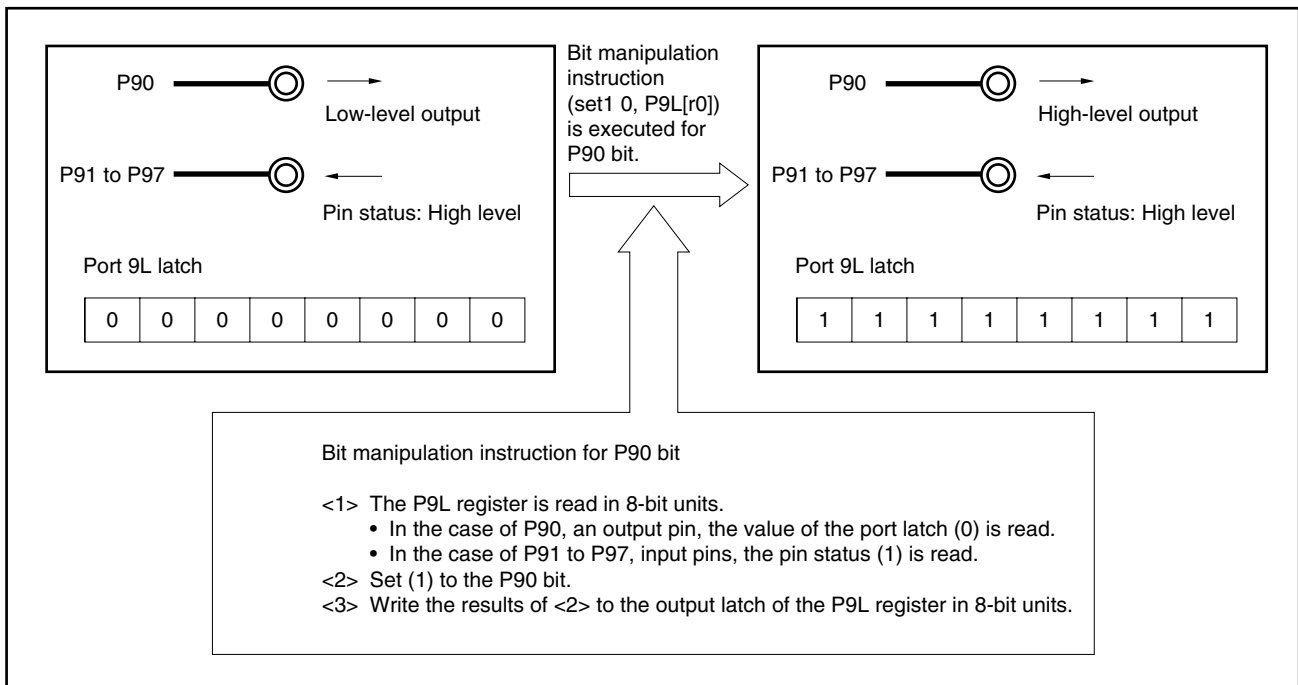
- <1> The Pn register is read in 8-bit units.
- <2> The targeted bit is manipulated.
- <3> The Pn register is written in 8-bit units.

In step <1>, the value of the output latch (0) of the P90 pin, which is an output pin, is read, while the pin statuses of the P91 to P97 pins, which are input pins, are read. If the pin statuses of the P91 to P97 pins are high level at this time, the value read 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 4-39. Bit Manipulation Instruction (P90 Pin)



4.6.3 Cautions on on-chip debug pins

The $\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO pins are on-chip debug pins.

After reset by the $\overline{\text{RESET}}$ pin, the P05/INTP2/ $\overline{\text{DRST}}$ pin is initialized to function as an on-chip debug pin ($\overline{\text{DRST}}$). If a high level is input to the $\overline{\text{DRST}}$ pin at this time, the on-chip debug mode is set, and the DCK, DMS, DDI, and DDO pins can be used.

The following action must be taken if on-chip debugging is not used.

- Clear the OCDM0 bit of the OCDM register (special register) (0)

At this time, fix the P05/INTP2/ $\overline{\text{DRST}}$ pin to low level from when reset by the $\overline{\text{RESET}}$ pin is released until the above action is taken.

If a high level is input to the $\overline{\text{DRST}}$ pin before the above action is taken, it may cause a malfunction (CPU deadlock). Handle the P05 pin with the utmost care.

Caution After reset by the WDT2RES signal, clock monitor (CLM), or low-voltage detector (LVI), the P05/INTP2/ $\overline{\text{DRST}}$ pin is not initialized to function as an on-chip debug pin ($\overline{\text{DRST}}$). The OCDM register holds the current value.

4.6.4 Cautions on P05/INTP2/ $\overline{\text{DRST}}$ pin

The P05/INTP2/ $\overline{\text{DRST}}$ pin has an internal pull-down resistor (30 k Ω TYP.). After a reset by the $\overline{\text{RESET}}$ pin, the pull-down resistor is connected. The pull-down resistor is disconnected when the OCDM0 bit is cleared (0).

4.6.5 Cautions on P10, P11, and P53 pins when power is turned on

When the power is turned on, the following pins may output an undefined level temporarily even during reset.

- P10/ANO0 pin
- P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO pin

4.6.6 Hysteresis characteristics

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

P02 to P06

P30 to P32, P38 to P39

P40 to P42

P50 to P55

P90 to P94, P96, P97, P99, P910, P912 to P915

CHAPTER 5 CLOCK GENERATOR

5.1 Overview

The clock generator generates the clock signals that are input to the CPU and peripherals. The clock generator includes a PLL circuit, which enables the clock frequency to be multiplied by four. The clock frequency can also be divided before clock signals are input to the CPU or on-chip peripherals. Clock oscillation can also be stopped to save power.

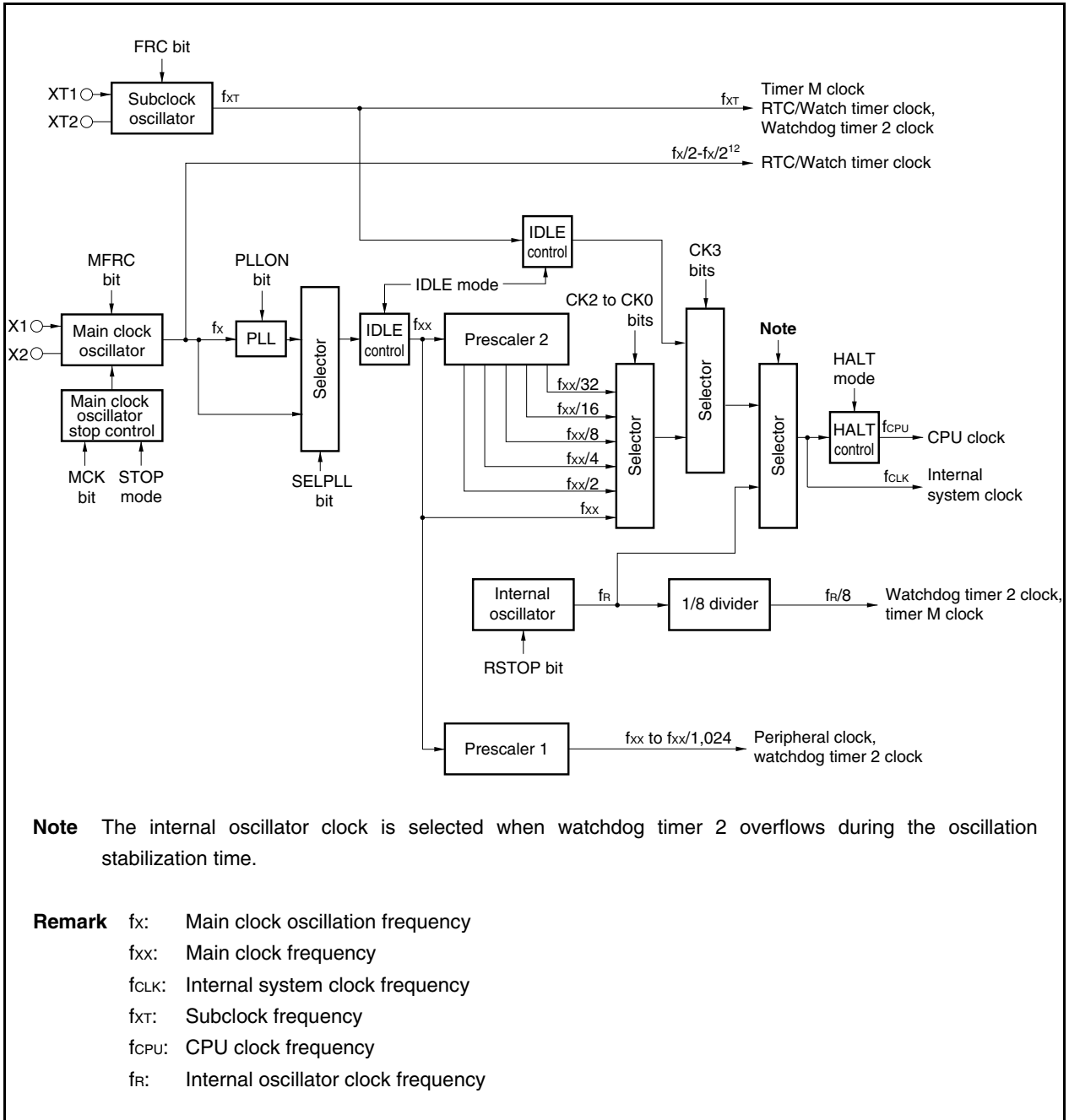
The clock generator has the following features:

- Main clock oscillator
 - In clock-through mode
 - $f_x = 2.5$ to 10 MHz ($f_{xx} = 2.5$ to 10 MHz)
 - In PLL mode
 - $f_x = 2.5$ to 5 MHz ($\times 4$: $f_{xx} = 10$ to 20 MHz)
- Subclock oscillator
 - $f_{XT} = 32.768$ kHz
- Internal oscillator
 - $f_R = 220$ kHz (TYP.)
- Multiplication ($\times 4$) function via PLL (Phase Locked Loop)
 - Clock-through mode/PLL mode selectable ($f_x = 2.5$ to 5 MHz)
- Internal system clock generation
 - 7 steps (f_{xx} , $f_{xx}/2$, $f_{xx}/4$, $f_{xx}/8$, $f_{xx}/16$, $f_{xx}/32$, f_{XT})
- Peripheral clock generation

Remark f_x : Main clock oscillation frequency
 f_{xx} : Main clock frequency
 f_{XT} : Subclock frequency
 f_R : Internal oscillator clock frequency

5.2 Configuration

Figure 5-1. Clock Generator



(1) Main clock oscillator

The main clock oscillator uses a ceramic/crystal resonator connected to X1 and X2 pins to oscillate the following frequencies (f_x).

- In clock-through mode
 $f_x = 2.5$ to 10 MHz ($f_{xx} = 2.5$ to 10 MHz)
- In PLL mode
 $f_x = 2.5$ to 5 MHz ($\times 4$: $f_{xx} = 10$ to 20 MHz)

An external clock of the following frequency is input to the X1 pin.

- In clock-through/PLL mode
 $f_x = 2.5$ to 5 MHz

(2) Subclock oscillator

The subclock oscillator oscillates a frequency of 32.768 kHz (f_{XT}).

This is in the RTC backup area and causes the subclock to continue oscillating even in the RTC backup mode.

(3) Main clock oscillator stop control

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

Oscillation of the main clock oscillator is stopped in the STOP mode or when the PCC.MCK bit is 1 (valid only when the PCC.CLS bit is 1).

(4) Internal oscillator

Oscillates a frequency (f_R) of 220 kHz (TYP.).

(5) Prescaler 1

This prescaler generates the clock (f_{xx} to $f_{xx}/1,024$) to be supplied to the following on-chip peripheral functions: TMP0 to TMP5, TMQ0, TMM0, CSIB0 to CSIB4, UARTA0 to UARTA2, I²C00 to I²C02, ADC, and WDT2

(6) Prescaler 2

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

The clock generated by prescaler 2 (f_{xx} to $f_{xx}/32$) 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, RAM, and DMA blocks.

(7) PLL

This circuit multiplies the clock generated by the main clock oscillator (f_x) by 4.

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 using the PLLCTL.SELPLL bit.

PLL is started or stopped by the PLLCTL.PLLON bit.

5.3 Registers

(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 3.4.7 **Special registers**).

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

Reset sets this register to 03H.

(1/2)

After reset: 03H R/W Address: FFFFF828H

	7	<6>	5	<4>	<3>	2	1	0
PCC	FRC ^{Note1}	MCK	MFRC	CLS ^{Note2}	CK3	CK2	CK1	CK0
FRC ^{Note1}	Use of subclock on-chip feedback resistor							
0	Used							
1	Not used							
MCK	Main clock oscillator control							
0	Oscillation enabled							
1	Oscillation stopped							
MFRC	Use of main clock on-chip feedback resistor							
0	Used (when ceramic/crystal resonator is used)							
1	Not used (when external clock is used)							
CLS ^{Note2}	Status of CPU clock (f _{CPU})							
0	Main clock operation							
1	Subclock operation							
CK3	CK2	CK1	CK0	Clock selection (f _{CLK} /f _{CPU})				
0	0	0	0	f _{xx}				
0	0	0	1	f _{xx} /2				
0	0	1	0	f _{xx} /4				
0	0	1	1	f _{xx} /8 (Initial value)				
0	1	0	0	f _{xx} /16				
0	1	0	1	f _{xx} /32				
0	1	1	×	Setting prohibited				
1	×	×	×	f _{XT}				

Notes1. When the FRC bit is set (to 1), the subclock stops oscillating

2. The CLS bit is a read-only bit.

(2/2)

- Cautions**
1. Use a bit manipulation instruction to manipulate the CK3 bit. When using an 8-bit manipulation instruction, do not change the set values of the CK2 to CK0 bits.
 2. When the external clock is used, set the MFRC bit to “1” so as not to use the internal feedback resistor.
 3. Even if the MCK bit is set (1) while the system is operating with the main clock as the CPU clock, the operation of the main clock does not stop. It stops after the CPU clock has been changed to the subclock.
 4. Before changing the MCK bit from 0 to 1, stop the on-chip peripheral functions operating on the main clock.
 5. When the main clock is stopped and the device is operating on the subclock, clear (0) the MCK bit and secure the oscillation stabilization time by software before switching the CPU clock to the main clock or operating the on-chip peripheral functions.

Remark ×: don't care

(a) Example of changing main clock operation to subclock operation

- <1> CK3 bit ← 1: Use of a bit manipulation instruction is recommended. Do not change the CK2 to CK0 bits.
- <2> Subclock operation: Read the CLS bit to check if subclock operation has started. It takes the following time after the CK3 bit is set until subclock operation is started.
Max.: $1/f_{XT}$ (1/subclock frequency)
- <3> MCK bit ← 1: Set the MCK bit to 1 only when stopping the main clock.

Cautions 1. When stopping the main clock, stop the PLL. Also stop the operations of the on-chip peripheral functions operating on the main clock.

- 2. If the following condition is not satisfied, change the CK2 to CK0 bits so that the condition is satisfied, then change to the subclock operation mode.**

$$\text{Internal system clock (f}_{CLK}\text{)} > \text{Subclock (f}_{XT}\text{: 32.768 kHz)} \times 4$$

Remark Internal system clock (f_{CLK}): Clock generated from the main clock (f_{XX}) by setting bits CK2 to CK0

[Description example]

```

_DMA_DISABLE:
clr1      0, DCHCn[r0]      -- DMA operation disabled. n = 0 to 3
<1> _SET_SUB_RUN :
st.b      r0, PRCMD[r0]
set1      3, PCC[r0]       -- CK3 bit ← 1
<2> _CHECK_CLS :
tst1      4, PCC[r0]       -- Wait until subclock operation starts.
bz        _CHECK_CLS
<3> _STOP_MAIN_CLOCK :
st.b      r0, PRCMD[r0]
set1      6, PCC[r0]       -- MCK bit ← 1, main clock is stopped.
_DMA_ENABLE:
set1      0, DCHCn[r0]     -- DMA operation enabled. n = 0 to 3

```

Remark The description above is simply an example. Note that in <2> above, the CLS bit is checked in a closed loop.

(b) Example of changing subclock operation to main clock operation

- <1> MCK bit ← 0: Main clock starts oscillating
- <2> Insert waits by program and wait until the oscillation stabilization time of the main clock has elapsed.
- <3> CK3 bit ← 0: Use of a bit manipulation instruction is recommended. Do not change the CK2 to CK0 bits.
- <4> Main clock operation: It takes the following time after the CK3 bit is set until main clock operation is started.
 Max.: $1/f_{XT}$ (1/subclock frequency)
 Therefore, insert one NOP instruction immediately after setting the CK3 bit to 0.

Caution Enable operation of the on-chip peripheral functions operating on the main clock only after the oscillation of the main clock stabilizes. If their operations are enabled before the lapse of the oscillation stabilization time, a malfunction may occur.

[Description example]

```

_DMA_DISABLE:
  clr1      0, DCHCn[r0]          -- DMA operation disabled. n = 0 to 3
<1> _START_MAIN_OSC :
  st.b      r0, PRCMD[r0]        -- Release of protection of special registers
  clr1      6, PCC[r0]           -- Main clock starts oscillating.
<2> movea    0x55, r0, r11        -- Wait for oscillation stabilization time.
  _WAIT_OST :
  nop
  nop
  nop
  addi      -1, r11, r11
  bnz       _WAIT_OST
<3> st.b     r0, PRCMD[r0]
  clr1      3, PCC[r0]           -- CK3 ← 0
<4> nop
_DMA_ENABLE:
  set1      0, DCHCn[r0]        -- DMA operation enabled. n = 0 to 3

```

(2) Internal oscillator mode register (RCM)

The RCM register is an 8-bit register that sets the operation mode of the internal oscillator.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF80CH

	7	6	5	4	3	2	1	<0>
RCM	0	0	0	0	0	0	0	RSTOP

RSTOP	Oscillation/stop of internal oscillator
0	Do not stop internal oscillator oscillation
1	Stop internal oscillator

- Cautions**
1. The internal oscillator cannot be stopped while the CPU is operating on the internal oscillator clock (CCLS.CCLS bit = 1). Do not set the RSTOP bit to 1.
 2. The internal oscillator oscillates if a watchdog timer overflow occurs while the oscillator signal is stabilizing after STOP mode has been canceled by the occurrence of an interrupt (that is, if the CCLS.CCLS bit is set to 1), even if the internal oscillator is stopped (the RSTOP bit is 1). At this time, RSTOP remains set to 1.

(3) CPU operation clock status register (CCLS)

The CCLS register indicates the status of the CPU operation clock.

This register is read-only, in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H^{Note} R Address: FFFFF82EH

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

CCLS	CPU operation clock status
0	Operating on main clock (f _x) or subclock (f _{xT}).
1	Operating on internal oscillator clock (f _R).

Note If a WDT overflow occurs during oscillation stabilization after a reset is released or STOP mode is released, the CCLS bit is set to 1 and the reset value is 01H.

5.4 Operations

5.4.1 Operation of each clock

The following table shows the operation status of each clock.

Table 5-1. Operation Status of Each Clock

Register Setting and Operation Status Target Clock	PCC Register								
	CLK Bit = 0, MCK Bit = 0					CLS Bit = 1, MCK Bit = 0		CLS Bit = 1, MCK Bit = 1	
	During Reset	During Oscillation Stabilization Time Count	HALT Mode	IDLE1 Mode, IDLE2 Mode	STOP Mode	Subclock Mode	Sub-IDLE Mode	Subclock Mode	Sub-IDLE Mode
Main clock oscillator (fx)	×	√	√	√	×	√	√	×	×
Subclock oscillator (fxT)	√	√	√	√	√	√	√	√	√
CPU clock (fCPU)	×	×	×	×	×	√	×	√	×
Internal system clock (fCLK)	×	×	√	×	×	√	×	√	×
Main clock (in PLL mode, fxx)	×	√ ^{Note1}	√	√	×	√	√	×	×
Peripheral clock (fxx to fxx/1,024)	×	×	√	×	×	√	×	×	×
WT clock (main)	×	√	√	√	×	√	√	×	×
WT clock (sub)	√	√	√	√	√	√	√	√	√
WDT2 clock (internal oscillation)	×	√	√	√	√	√	√	√	√
WDT2 clock (main)	×	×	√	×	×	√	×	×	×
WDT2 clock (sub)	√	√	√	√	√	√	√	√	√
RTC clock (main)	×	×	√	√	×	√	√	×	×
RTC clock (sub)	√	√	√	√	√	√	√	√	√

Notes1. Lockup time

2. Be sure to set the PLLCTL.PLLON to 0.

Remark √: Operating
 ×: Stopped
 -: Undefined

5.4.2 External clock signal input

An external clock signal can be directly input to the oscillator. Input the clock to the X1 pin and leave the X2 pin open. Set the PCC.MFRC bit to 1 (on-chip feedback resistor not used). Note, however, that time is required to stabilize the oscillator signal even when inputting an external clock signal.

5.5 PLL Function

5.5.1 Overview

In the V850ES/JC3-L, V850ES/JE3-L, an operating clock that is the oscillation frequency multiplied by 4 by the PLL function or an unmultiplied clock (clock-through mode) can be selected as the operating clock of the CPU and on-chip peripheral functions.

When PLL function is used: Input clock = 2.5 to 5 MHz (output: 10 to 20 MHz)

Clock-through mode: Input clock = 2.5 to 10 MHz (output: 2.5 to 10 MHz)

5.5.2 Registers

(1) PLL control register (PLLCTL)

The PLLCTL register is an 8-bit register that controls the PLL function.

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

Reset sets this register to 01H.

After reset: 01H		R/W	Address: FFFFF82CH					
PLLCTL	7	6	5	4	3	2	<1>	<0>
	0	0	0	0	0	0	SELPLL	PLLON
SELPLL		Selection of CPU operation clock mode						
0		Clock-through mode						
1		PLL mode						
PLLON		Control of PLL operation						
0		Disable PLL operation						
1		Enable PLL operation (After PLL operation starts, a lockup time is required for frequency stabilization.)						

- Cautions**
1. When the PLLON bit is cleared to 0, the SELPLL bit is automatically cleared to 0 (clock-through mode).
 2. The SELPLL bit can be set to 1 only when the PLL clock frequency is stabilized. If not (if the PLL is unlocked), "0" is written to the SELPLL bit whatever data is written to it.

(2) Clock control register (CKC)

The CKC register is a special register. Data can be written to this register only in a combination of specific sequences (see **3.4.7 Special registers**).

The CKC register controls the internal system clock in the PLL mode.

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

Reset sets this register to 0AH.

After reset: 0AH R/W Address: FFFFF822H

	7	6	5	4	3	2	1	0
CKC	0	0	0	0	1	0	1	CKDIV0

CKDIV0	Internal system clock (f_{xx}) in PLL mode
0	$f_{xx} = 4 \times f_x$ ($f_x = 2.5$ to 5.0 MHz)
1	Setting prohibited

- Cautions**
1. The PLL mode cannot be used when $f_x = 5.0$ to 10.0 MHz.
 2. Be sure to set the CKC register to 0AH. If a value other than 0AH is set, the operation is not guaranteed.

(3) Lock register (LOCKR)

The PLL locks the phase at a given frequency after the power is turned on or immediately after the STOP mode is canceled. The time required for the frequency to stabilize is the lockup time (frequency stabilization time). This state until the frequency stabilizes is called the lockup status, and the state in which the frequency is stabilized is called the locked status.

The LOCKR register includes a LOCK bit that reflects the PLL frequency stabilization status.

This register is read-only, in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R Address: FFFFF824H

	7	6	5	4	3	2	1	<0>
LOCKR	0	0	0	0	0	0	0	LOCK

LOCK	PLL lock status check
0	Locked status
1	Unlocked status

Caution The LOCK register does not reflect the lock status of the PLL in real time. The set/clear conditions are as follows.

[Set conditions]

- Upon system reset^{Note}
- In IDLE2 or STOP mode
- Upon setting of PLL stop (clearing of PLLCTL.PLLON bit to 0)
- Upon stopping main clock and using CPU on subclock (setting of PCC.CK3 bit to 1 and setting of PCC.MCK bit to 1)

Note This register is set to 01H by reset and cleared to 00H after the reset has been released and the oscillation stabilization time has elapsed.

[Clear conditions]

- Upon overflow of oscillation stabilization time following reset release (OSTS register default time (see **CHAPTER 27 OPTION BYTE**))
- Upon oscillation stabilization timer overflow (time set by OSTS register) following STOP mode release, when the STOP mode was set in the PLL operating status
- Upon PLL lockup time timer overflow (time set by PLLS register) when the PLLCTL.PLLON bit is changed from 0 to 1
- After the setup time inserted upon release of the IDLE2 mode (time set by the OSTS register) has elapsed when the IDLE2 mode is set during PLL operation.

(4) PLL lockup time specification register (PLLS)

The PLLS register is an 8-bit register used to select the PLL lockup time when the PLLCTL.PLLON bit is changed from 0 to 1.

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

Reset sets this register to 03H.

After reset: 03H R/W Address: FFFFF6C1H

	7	6	5	4	3	2	1	0
PLLS	0	0	0	0	0	0	PLLS1	PLLS0

PLLS1	PLLS0	Selection of PLL lockup time
0	0	$2^{10}/f_x$
0	1	$2^{11}/f_x$
1	0	$2^{12}/f_x$
1	1	$2^{13}/f_x$ (default value)

- Cautions**
1. Set so that the lockup time is at least 800 μ s.
 2. Do not change the PLLS register setting during the lockup period.

5.5.3 Usage

(1) When PLL is used

- After the reset signal has been released, the PLL operates (PLLCTL.PLLON bit = 1), but because the default mode is the clock-through mode (PLLCTL.SELPLL bit = 0), select the PLL mode (SELPLL bit = 1).
- To enable PLL operation, first set the PLLON bit to 1, and then set the SELPLL bit to 1 after the LOCKR.LOCK bit becomes 0. To stop the PLL, first select the clock-through mode (SELPLL bit = 0), wait for 8 clocks or more, and then stop the PLL (PLLON bit = 0).
- The PLL stops during transition to the IDLE2 or STOP mode regardless of the setting and is restored from the IDLE2 or STOP mode to the status before transition. The time required for restoration is as follows.

(a) When transitioning to the IDLE2 or STOP mode from the clock through mode

- STOP mode: Set the OSTS register so that the oscillation stabilization time is at least 1 ms.
- IDLE2 mode: Set the OSTS register so that the setup time is at least 350 μ s.

(b) When transitioning to the IDLE 2 or STOP mode while remaining in the PLL operation mode

- STOP mode: Set the OSTS register so that the oscillation stabilization time is at least 1 ms.
- IDLE2 mode: Set the OSTS register so that the setup time is at least 800 μ s.

When transitioning to the IDLE1 mode, the PLL does not stop. Stop the PLL if necessary.

(2) When PLL is not used

- The clock-through mode (SELPLL bit = 0) is selected after the reset signal has been released, but the PLL is operating (PLLON bit = 1) and must therefore be stopped (PLLON bit = 0).

The time required for restoration from the IDLE2 and STOP modes is as follows.

- STOP mode: Set the OSTS register so that the oscillation stabilization time is at least 1 ms.
- IDLE2 mode: Set the OSTS register so that the setup time is at least 350 μ s.

5.6 How to Connect a Resonator

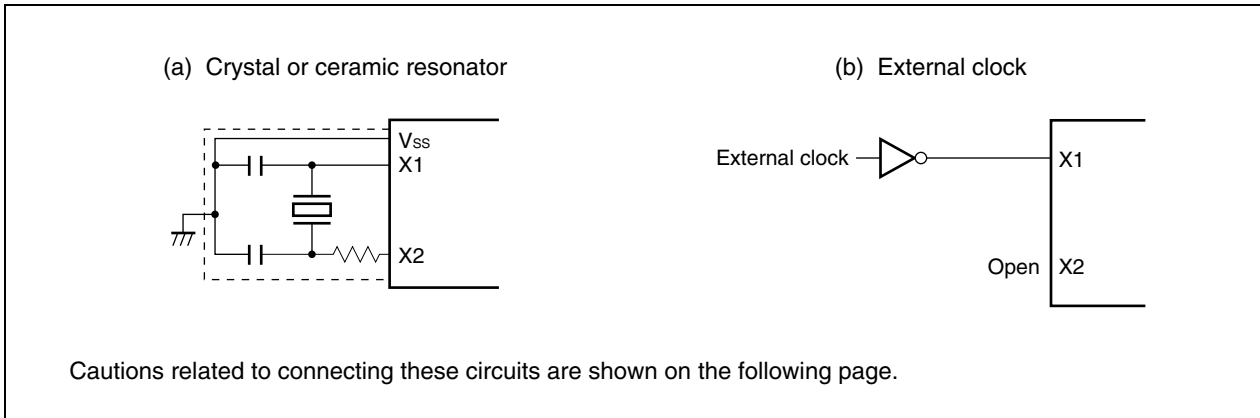
5.6.1 Main clock oscillator

The signal input to the main clock oscillator is oscillated by a ceramic or crystal resonator connected to the X1 and X2 pins. The frequency of the resonator is 2.5 to 10 MHz.

An external clock signal can also be input to the main clock oscillator.

Figure 5-2 shows an example of the circuit connected to the main clock oscillator.

Figure 5-2. Example of Circuit Connected to Main Clock Oscillator

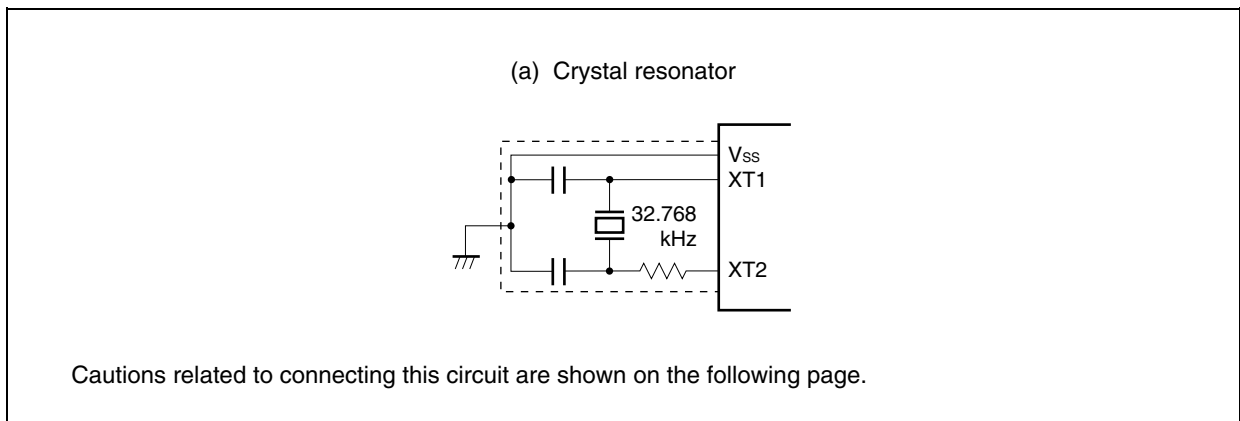


5.6.2 Subclock oscillator

The signal input to the subclock oscillator is oscillated by a crystal resonator connected to the XT1 and XT2 pins. The frequency of the resonator is 32.768 kHz (standard).

Figure 5-3 shows an example of the circuit connected to the subclock oscillator.

Figure 5-3. Example of Circuit Connected to Subclock Oscillator



Caution 1. When using the main clock or subclock oscillator, wire as follows in the area enclosed by the broken lines in Figures 5-2 and 5-3 to avoid an adverse effect from wiring capacitance.

- Keep the wiring length as short as possible.
- Do not cross the wiring with the other signal lines.
- Do not route the wiring near a signal line through which a high fluctuating current flows.
- Always make the ground point of the oscillator capacitor the same potential as V_{SS} .
- Do not ground the capacitor to a ground pattern through which a high current flows.
- Do not fetch signals from the oscillator.

The subclock oscillator is designed as a low-amplitude circuit for reducing power consumption. Particular care is therefore required with the wiring method when the subclock is used.

Figure 5-4 shows examples of incorrect resonator connections.

Figure 5-4. Examples of Incorrect Resonator Connections (1/2)

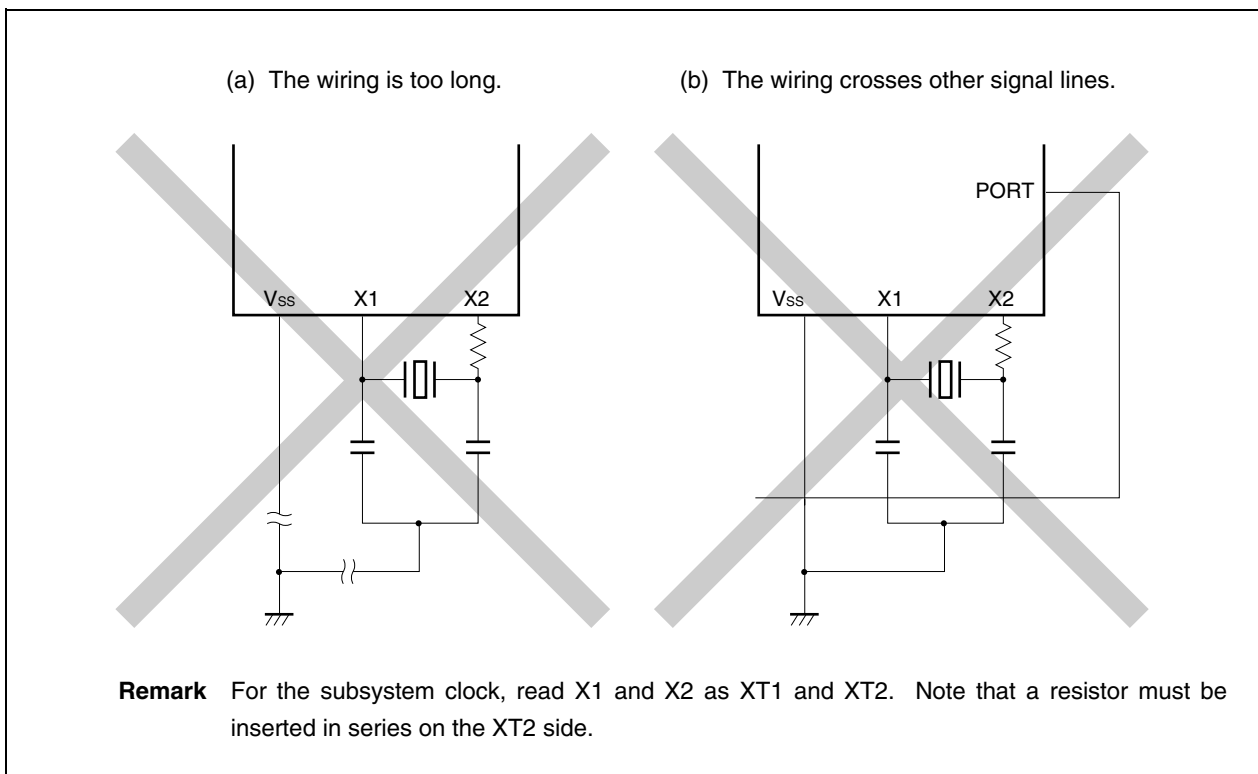
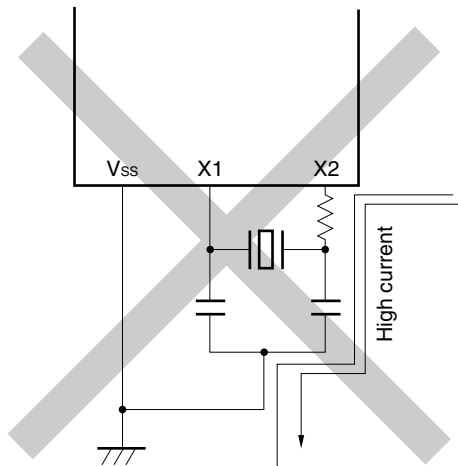
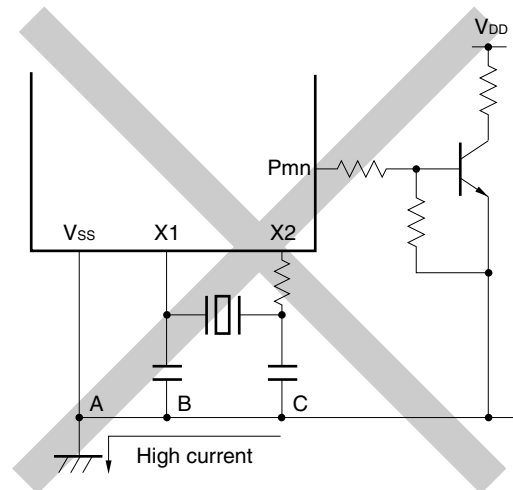


Figure 5-4. Examples of Incorrect Resonator Connections (2/2)

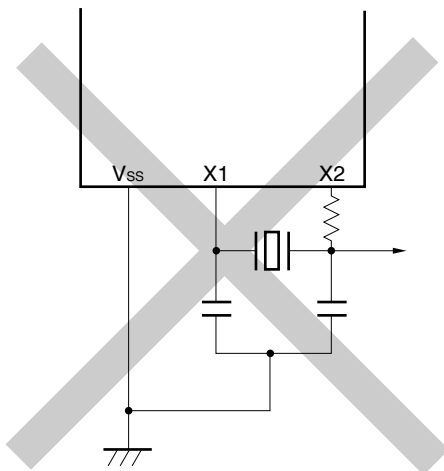
- (c) The wiring is routed near a signal line through which a high fluctuating current flows.



- (d) A current with a higher potential than that of the ground line of the oscillator block is flowing. (The potential differs at points A, B, and C.)



- (e) Signals are being fetched.



Remark For the subsystem clock, read X1 and X2 as XT1 and XT2. Note that a resistor must be inserted in series on the XT2 side.

Caution If X2 and XT1 are wired in parallel, crosstalk noise from XT1 may have a synergistic effect on X2, causing a malfunction.

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

Timer P (TMP) is a 16-bit timer/event counter.

The V850ES/JC3-L, V850ES/JE3-L have six timer/event counter channels, TMP0 to TMP5.

6.1 Overview

TMPn has the following features.

- (1) Interval timer
TMPn generates an interrupt at a preset interval and can output a square wave.
- (2) External event counter
TMPn counts the number of externally input signal pulses.
- (3) External trigger pulse output
TMPn starts counting and outputs a pulse when the specified external signal is input.
- (4) One-shot pulse output
TMPn outputs a one-shot pulse with an output width that can be freely specified.
- (5) PWM output
TMPn outputs a pulse with a constant cycle whose active width can be changed.
The pulse duty can also be changed freely even while the timer is operating.
- (6) Free-running timer
TMPn increments from 0000H to FFFFH and then resets.
- (7) Pulse width measurement
TMPn can be used to measure the pulses of a signal input externally.

Remark n = 0 to 5

6.2 Configuration

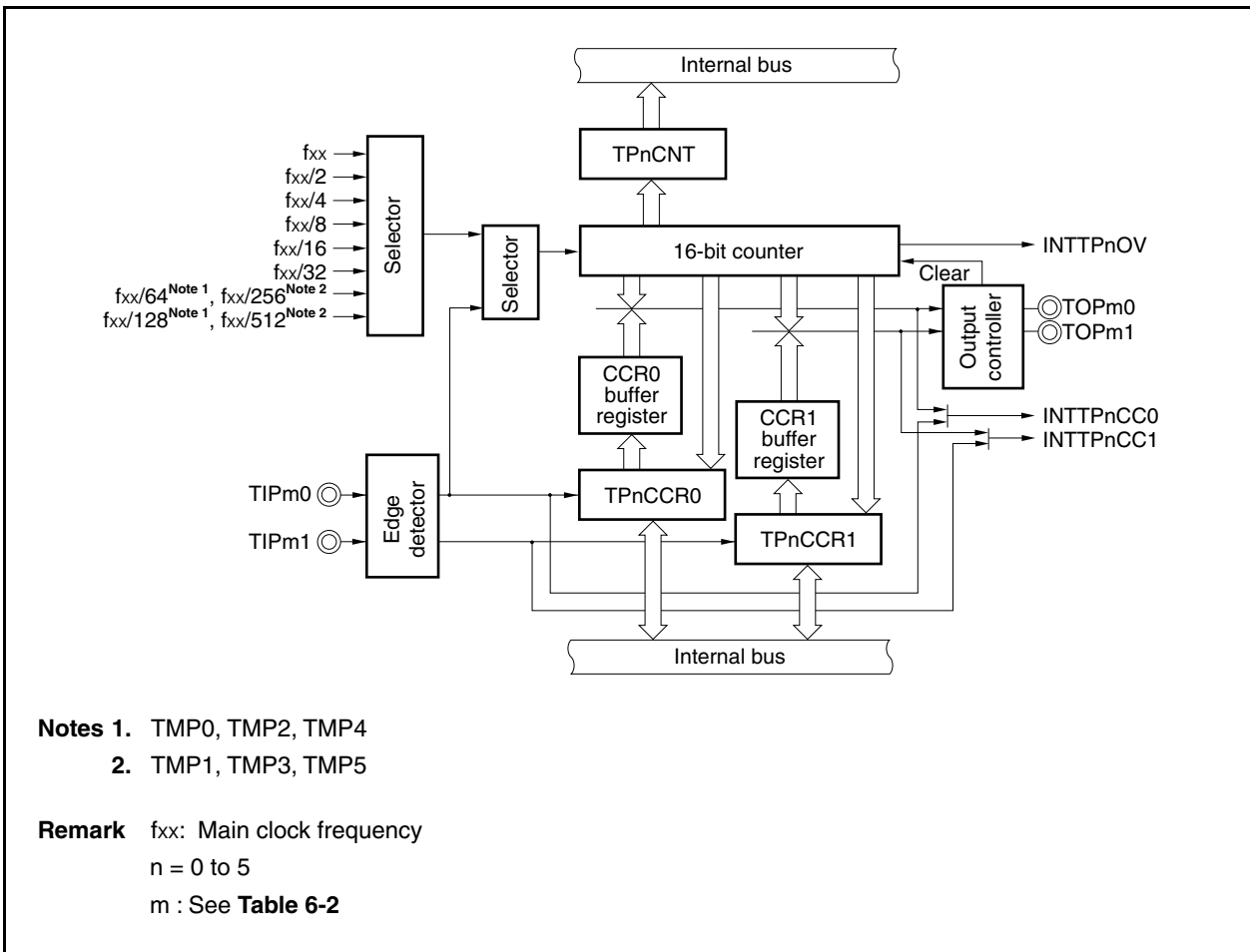
TMPn includes the following hardware.

Table 6-1. Configuration of TMPn

Item	Configuration
Registers	16-bit counter TMPn counter read buffer register (TPnCNT) TMPn capture/compare registers 0, 1 (TPnCCR0, TPnCCR1) CCR0, CCR1 buffer registers TMPn control registers 0, 1 (TPnCTL0, TPnCTL1) TMPn I/O control registers 0 to 2 (TPnIOC0 to TPnIOC2) TMPn option register 0 (TPnOPT0)
Timer inputs	2 (TIPm0, TIPm1 pins)
Timer outputs	2 (TOPm0, TOPm1 pins)

- Remarks1.** n = 0 to 5
2. m : see **Table 6-2**

Figure 6-1. Block Diagram of TMPn



(1) 16-bit counter

This is a 16-bit counter that counts internal clocks and external events.

This counter can be read by using the TPnCNT register.

When the TPnCTL0.TPnCE bit is 0 and the counter is stopped, the counter value is FFFFH. If the TPnCNT register is read at this time, 0000H is read.

Reset sets the TPnCE bit to 0, stopping the counter, and setting its value to FFFFH.

(2) TMPn counter read buffer register (TPnCNT)

This is a read buffer register from which the value of the 16-bit counter can be read.

(3) TMPn capture/compare registers 0 and 1 (TPnCCR0 and TPnCCR1)

These registers can be used as either capture registers or compare registers, in accordance with the specified mode.

(4) CCR0 buffer register

This is a 16-bit compare register that compares the 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. If the value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTPnCC0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is cleared to 0000H after reset because the TPnCCR0 register is cleared to 0000H.

(5) CCR1 buffer register

This is a 16-bit compare register that compares the 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. If the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTPnCC1) is generated.

The CCR1 buffer register cannot be read or written directly.

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

(6) TMPn control registers 0 and 1 (TPnCTL0 and TPnCTL1)

These are 8-bit registers that control the operations of TMPn.

(7) TMPn I/O control registers 0 to 2 (TPnIOC0 to TPnIOC2)

These are 8-bit registers that control the input and output of TMPn.

(8) TMPn option register 0 (TPnOPT0)

This is an 8-bit register that controls the specification of settings such as capture and compare.

(9) Edge detector

This circuit detects the valid edges input to the TIPn0 and TIPn1 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.

(10) Output controller

This circuit controls the output of the TOPn0 and TOPn1 pins. The output controller is controlled by the TPnIOC0 register.

(11) Selector

The selector selects the count clock for the 16-bit counter. One of eight internal clocks or the input of an external event can be selected as the count clock.

6.2.1 Pins used by TMPn

The input and output pins used by TMPn are shown in Table 6-2 below. When using these pins for TMPn, first set them to port mode. For details, see **Table 4-17 Settings When Pins Are Used for Alternate Functions**.

Table 6-2. Pins Used by TMPn

Timer Channel	Pin No.			Port	TMP Input	TMP Output	Alternate Function
	JC3L (40)	JC3L (48)	JE3-L				
TMP0	–	35	47	P32	TIP00 ^{Note}	TOP00	ASCKA0/SCKB4
	–	–	43	P33	TIP01	TOP01	–
TMP1	–	–	44	P97	TIP10 ^{Note}	TOP10	–
	–	–	23	P96	TIP11	TOP11	–
TMP2	–	38	50	P97	TIP20 ^{Note}	TOP20	SIB1
	32	–	–				–
	13	15	22	P96	TIP21	TOP21	–
TMP3	–	–	21	P94	TIP31	TOP31	–
TMP4	–	–	34	P93	TIP40 ^{Note}	TOP40	–
	–	–	33	P92	TIP41	TOP41	–
TMP5	34	41	53	P915	TIP50 ^{Note}	TOP50	INTP6
	33	40	52	P914	TIP51	TOP51	INTP5

Note The TIPn0 pin functions as a capture trigger input, as an external event input, and as an external trigger input (n = 0 to 2, 4, 5).

Remark JC3L (40): V850ES/JC3-L (40-pin products)

JC3L (48): V850ES/JC3-L (48-pin products)

JE3L: V850ES/JE3-L

6.2.2 Register configuraiton

For the V850ES/JC3-L and V850ES/JE3-L, the registers and their bit assignment differ for each product.

The register configuration for each product is shown in the following tables

(1) V850ES/JC3-L (40-pin)

Channel	Register name	Bit position							
		7	6	5	4	3	2	1	0
TMP0	TP0CTL0	TP0CE	0	0	0	0	TP0CKS2	TP0CKS1	TP0CKS0
	TP0CTL1	0	0	0	0	0	TP0MD2	TP0MD1	TP0MD0
TMP1	TP1CTL0	TP1CE	0	0	0	0	TP1CKS2	TP1CKS1	TP1CKS0
	TP1CTL1	0			0	0	TP1MD2	TP1MD1	TP1MD0
TMP2	TP2CTL0	TP2CE	0	0	0	0	TP2CKS2	TP2CKS1	TP2CKS0
	TP2CTL1	0	TP2EST	TP2EEE	0	0	TP2MD2	TP2MD1	TP2MD0
	TP2IOC0	0	0	0	0	TP2OL1	TP2OE1	TP2OL0	TP2OE0
	TP2IOC1	0	0	0	0	TP2IS3	TP2IS2	TP2IS1	TP2IS0
	TP2IOC2	0	0	0	0	TP2EES1	TP2EES0	TP2ETS1	TP2ETS0
	TP2OPT0	0	0	TP2CCS1	TP2CCS0	0	0	0	TP2OVF
TMP3	TP3CTL0	TP3CE	0	0	0	0	TP3CKS2	TP3CKS1	TP3CKS0
	TP3CTL1	0	0	0	0	0	TP3MD2	TP3MD1	TP3MD0
TMP4	TP4CTL0	TP4CE	0	0	0	0	TP4CKS2	TP4CKS1	TP4CKS0
	TP4CTL1	0	0	0	0	0	TP4MD2	TP4MD1	TP4MD0
TMP5	TP5CTL0	TP5CE	0	0	0	0	TP5CKS2	TP5CKS1	TP5CKS0
	TP5CTL1	0	TP5EST	TP5EEE	0	0	TP5MD2	TP5MD1	TP5MD0
	TP5IOC0	0	0	0	0	TP5OL1	TP5OE1	TP5OL0	TP5OE0
	TP5IOC1	0	0	0	0	TP5IS3	TP5IS2	TP5IS1	TP5IS0
	TP5IOC2	0	0	0	0	TP5EES1	TP5EES0	TP5ETS1	TP5ETS0
	TP5OPT0	0	0	TP5CCS1	TP5CCS0	0	0	0	TP5OVF

Remark The TPnCCR0, TPnCCR1, and TPnCNT registers are available for all channels.

(2) V850ES/JC3-L (48-pin)

Channel	Register name	Bit position							
		7	6	5	4	3	2	1	0
TMP0	TP0CTL0	TP0CE	0	0	0	0	TP0CKS2	TP0CKS1	TP0CKS0
	TP0CTL1	0	TP0EST	TP0EEE	0	0	TP0MD2	TP0MD1	TP0MD0
	TP0IOC0	0	0	0	0	0	0	TP0OL0	TP0OE0
	TP0IOC1	0	0	0	0	0	0	TP0IS1	TP0IS0
	TP0IOC2	0	0	0	0	TP0EES1	TP0EES0	TP0ETS1	TP0ETS0
	TP0OPT0	0	0	TP0CCS1	TP0CCS0	0	0	0	TP0OVF
TMP1	TP1CTL0	TP1CE	0	0	0	0	TP1CKS2	TP1CKS1	TP1CKS0
	TP1CTL1	0			0	0	TP1MD2	TP1MD1	TP1MD0
TMP2	TP2CTL0	TP2CE	0	0	0	0	TP2CKS2	TP2CKS1	TP2CKS0
	TP2CTL1	0	TP2EST	TP2EEE	0	0	TP2MD2	TP2MD1	TP2MD0
	TP2IOC0	0	0	0	0	TP2OL1	TP2OE1	TP2OL0	TP2OE0
	TP2IOC1	0	0	0	0	TP2IS3	TP2IS2	TP2IS1	TP2IS0
	TP2IOC2	0	0	0	0	TP2EES1	TP2EES0	TP2ETS1	TP2ETS0
	TP2OPT0	0	0	TP2CCS1	TP2CCS0	0	0	0	TP2OVF
TMP3	TP3CTL0	TP3CE	0	0	0	0	TP3CKS2	TP3CKS1	TP3CKS0
	TP3CTL1	0	0	0	0	0	TP3MD2	TP3MD1	TP3MD0
TMP4	TP4CTL0	TP4CE	0	0	0	0	TP4CKS2	TP4CKS1	TP4CKS0
	TP4CTL1	0	0	0	0	0	TP4MD2	TP4MD1	TP4MD0
TMP5	TP5CTL0	TP5CE	0	0	0	0	TP5CKS2	TP5CKS1	TP5CKS0
	TP5CTL1	0	TP5EST	TP5EEE	0	0	TP5MD2	TP5MD1	TP5MD0
	TP5IOC0	0	0	0	0	TP5OL1	TP5OE1	TP5OL0	TP5OE0
	TP5IOC1	0	0	0	0	TP5IS3	TP5IS2	TP5IS1	TP5IS0
	TP5IOC2	0	0	0	0	TP5EES1	TP5EES0	TP5ETS1	TP5ETS0
	TP5OPT0	0	0	TP5CCS1	TP5CCS0	0	0	0	TP5OVF

Remark The TPnCCR0, TPnCCR1, and TPnCNT registers are available for all channels.

(3) V850ES/JE3-L

Channel	Register name	Bit position							
		7	6	5	4	3	2	1	0
TMP0	TP0CTL0	TP0CE	0	0	0	0	TP0CK2	TP0CK1	TP0CK0
	TP0CTL1	0	TP0EST	TP0EEE	0	0	TP0MD2	TP0MD1	TP0MD0
	TP0IOC0	0	0	0	0	TP0OL1	TP0OE1	TP0OL0	TP0OE0
	TP0IOC1	0	0	0	0	TP0IS3	TP0IS2	TP0IS1	TP0IS0
	TP0IOC2	0	0	0	0	TP0EES1	TP0EES0	TP0ETS1	TP0ETS0
	TP0OPT0	0	0	TP0CCS1	TP0CCS0	0	0	0	TP0OVF
TMP1	TP1CTL0	TP1CE	0	0	0	0	TP1CK2	TP1CK1	TP1CK0
	TP1CTL1	0	TP1EST	TP1EEE	0	0	TP1MD2	TP1MD1	TP1MD0
	TP1IOC0	0	0	0	0	TP1OL1	TP1OE1	TP1OL0	TP1OE0
	TP1IOC1	0	0	0	0	TP1IS3	TP1IS2	TP1IS1	TP1IS0
	TP1IOC2	0	0	0	0	TP1EES1	TP1EES0	TP1ETS1	TP1ETS0
	TP1OPT0	0	0	TP1CCS1	TP1CCS0	0	0	0	TP1OVF
TMP2	TP2CTL0	TP2CE	0	0	0	0	TP2CK2	TP2CK1	TP2CK0
	TP2CTL1	0	TP2EST	TP2EEE	0	0	TP2MD2	TP2MD1	TP2MD0
	TP2IOC0	0	0	0	0	TP2OL1	TP2OE1	TP2OL0	TP2OE0
	TP2IOC1	0	0	0	0	TP2IS3	TP2IS2	TP2IS1	TP2IS0
	TP2IOC2	0	0	0	0	TP2EES1	TP2EES0	TP2ETS1	TP2ETS0
	TP2OPT0	0	0	TP2CCS1	TP2CCS0	0	0	0	TP2OVF
TMP3	TP3CTL0	TP3CE	0	0	0	0	TP3CK2	TP3CK1	TP3CK0
	TP3CTL1	0	TP3EST	0	0	0	TP3MD2	TP3MD1	TP3MD0
	TP3IOC0	0	0	0	0	TP3OL1	TP3OE1	0	0
	TP3IOC1	0	0	0	0	TP3IS3	TP3IS2	0	0
	TP3OPT0	0	0	TP3CCS1	TP3CCS0	0	0	0	TP3OVF
TMP4	TP4CTL0	TP4CE	0	0	0	0	TP4CK2	TP4CK1	TP4CK0
	TP4CTL1	0	TP4EST	TP4EEE	0	0	TP4MD2	TP4MD1	TP4MD0
	TP4IOC0	0	0	0	0	TP4OL1	TP4OE1	TP4OL0	TP4OE0
	TP4IOC1	0	0	0	0	TP4IS3	TP4IS2	TP4IS1	TP4IS0
	TP4IOC2	0	0	0	0	TP4EES1	TP4EES0	TP4ETS1	TP4ETS0
	TP4OPT0	0	0	TP4CCS1	TP4CCS0	0	0	0	TP4OVF
TMP5	TP5CTL0	TP5CE	0	0	0	0	TP5CK2	TP5CK1	TP5CK0
	TP5CTL1	0	TP5EST	TP5EEE	0	0	TP5MD2	TP5MD1	TP5MD0
	TP5IOC0	0	0	0	0	TP5OL1	TP5OE1	TP5OL0	TP5OE0
	TP5IOC1	0	0	0	0	TP5IS3	TP5IS2	TP5IS1	TP5IS0
	TP5IOC2	0	0	0	0	TP5EES1	TP5EES0	TP5ETS1	TP5ETS0
	TP5OPT0	0	0	TP5CCS1	TP5CCS0	0	0	0	TP5OVF

Remark The TPnCCR0, TPnCCR1, and TPnCNT registers are available for all channels.

6.2.3 Interrupts

The following three types of interrupt signals are used by TMPn:

(1) INTTPnCC0

This signal is generated when the value of the 16-bit counter matches the value of the CCR0 buffer register, or when a capture signal is input from the TIPn0 pin.

(2) INTTPnCC1

This signal is generated when the value of the 16-bit counter matches the value of the CCR1 buffer register, or when a capture signal is input from the TIPn1 pin.

(3) INTTPnOV

This signal is generated when the 16-bit counter overflows after incrementing up to FFFFH.

6.3 Registers

The registers that control TMPn are as follows.

- TMPn control register 0 (TPnCTL0)
- TMPn control register 1 (TPnCTL1)
- TMPn I/O control register 0 (TPnIOC0)
- TMPn I/O control register 1 (TPnIOC1)
- TMPn I/O control register 2 (TPnIOC2)
- TMPn option register 0 (TPnOPT0)
- TMPn capture/compare register 0 (TPnCCR0)
- TMPn capture/compare register 1 (TPnCCR1)
- TMPn counter read buffer register (TPnCNT)

Remark When using the functions of the TIPn0, TIPn1, TOPn0, and TOPn1 pins, see **Table 4-17 Settings When Pins Are Used for Alternate Functions.**

(1) **TMPn control register 0 (TPnCTL0)**

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

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

Reset sets this register to 00H.

The same value can always be written to the TPnCTL0 register by software.

After reset: 00H R/W Address: TP0CTL0 FFFFF590H, TP1CTL0 FFFFF5A0H,
 TP2CTL0 FFFFF5B0H, TP3CTL0 FFFFF5C0H,
 TP4CTL0 FFFFF5D0H, TP5CTL0 FFFFF5E0H

		<7>	6	5	4	3	2	1	0
TPnCTL0	TPnCE	0	0	0	0	TPnCKS2	TPnCKS1	TPnCKS0	

(n = 0 to 5)

TPnCE	TMPn operation control
0	TMPn operation disabled (TMPn reset asynchronously ^{Note}).
1	TMPn operation enabled. TMPn operation started.

TPnCKS2	TPnCKS1	TPnCKS0	Internal count clock selection	
			n = 0, 2, 4	n = 1, 3, 5
0	0	0	f _{xx}	
0	0	1	f _{xx} /2	
0	1	0	f _{xx} /4	
0	1	1	f _{xx} /8	
1	0	0	f _{xx} /16	
1	0	1	f _{xx} /32	
1	1	0	f _{xx} /64	f _{xx} /256
1	1	1	f _{xx} /128	f _{xx} /512

Note TPnOPT0.TPnOVF bit, 16-bit counter, timer output (TOPn0, TOPn1 pins)

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

Remark f_{xx}: Main clock frequency

(2) TMPn control register 1 (TPnCTL1)

The TPnCTL1 register is an 8-bit register that controls the operation of TMPn.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TP0CTL1 FFFFF591H, TP1CTL1 FFFFF5A1H,
TP2CTL1 FFFFF5B1H, TP3CTL1 FFFFF5C1H,
TP4CTL1 FFFFF5D1H, TP5CTL1 FFFFF5E1H

	7	<6>	<5>	4	3	2	1	0
TPnCTL1 (n = 0 to 5)	0	TPnEST	TPnEEE	0	0	TPnMD2	TPnMD1	TPnMD0

TPnEST	Software trigger control
0	–
1	Generate a valid signal for external trigger input. <ul style="list-style-type: none"> • 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.

TPnEEE	Count clock selection
0	Disable operation with external event count input. (Perform counting with the internal count clock selected by the TPnCTL0.TPnCK0 to TPnCK2 bits.)
1	Enable operation with external event count input. (Perform counting at the valid edge of the external event count input signal.)

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 the 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.

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

4. Be sure to clear bits 3, 4, and 7 to “0”.

(3) TMPn I/O control register 0 (TPnIOC0)

The TPnIOC0 register is an 8-bit register that controls the operation of timer output (TOPn0, TOPn1 pins). This register can be read or written in 8-bit or 1-bit units. Reset sets this register to 00H.

After reset: 00H R/W Address: TP0IOC0 FFFFF592H, TP1IOC0 FFFFF5A2H,
TP2IOC0 FFFFF5B2H, TP3IOC0 FFFFF5C2H,
TP4IOC0 FFFFF5D2H, TP5IOC0 FFFFF5E2H

	7	6	5	4	3	<2>	1	<0>
TPnIOC0 (n = 0 to 5)	0	0	0	0	TPnOL1	TPnOE1	TPnOL0	TPnOE0

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 disabled • When TPnOL1 bit = 0: Low level is output from the TOPn1 pin • When TPnOL1 bit = 1: High level is output from the TOPn1 pin
1	Timer output enabled (a pulse is output from the TOPn1 pin).

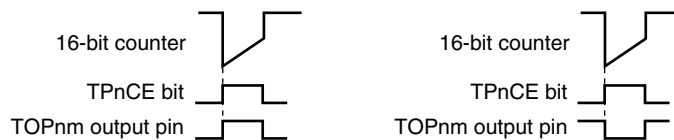
TPnOL0	TOPn0 pin output level setting ^{Note}
0	TOPn0 pin starts output at high level
1	TOPn0 pin starts output at low level

TPnOE0	TOPn0 pin output setting
0	Timer output disabled • When TPnOL0 bit = 0: Low level is output from the TOPn0 pin • When TPnOL0 bit = 1: High level is output from the TOPn0 pin
1	Timer output enabled (a pulse is output from the TOPn0 pin).

Note The output level of the timer output pin (TOPnm) specified by the TPnOLm bit is shown below (m = 0, 1).

• When TPnOLm bit = 0

• When TPnOLm bit = 1



Cautions 1. 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.

2. Even if the TPnOLm bit is manipulated when the TPnCE and TPnOEm bits are 0, the TOPnm pin output level varies (m = 0, 1).

(4) TMPn I/O control register 1 (TPnIOC1)

The TPnIOC1 register is an 8-bit register that controls the valid edge of the capture trigger input signals (TIPn0, TIPn1 pins).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TP0IOC1 FFFFF593H, TP1IOC1 FFFFF5A3H,
TP2IOC1 FFFFF5B3H, TP3IOC1 FFFFF5C3H,
TP4IOC1 FFFFF5D3H, TP5IOC1 FFFFF5E3H

	7	6	5	4	3	2	1	0
TPnIOC1	0	0	0	0	TPnIS3	TPnIS2	TPnIS1	TPnIS0

(n = 0 to 5)

TPnIS3	TPnIS2	Capture trigger input signal (TIPn1 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 (TIPn0 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.
 2. The TPnIS3 to TPnIS0 bits are valid only in the free-running timer mode 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 of the external event count input signal (TIPn0 pin) and external trigger input signal (TIPn0 pin).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TP0IOC2 FFFFF594H, TP1IOC2 FFFFF5A4H,
TP2IOC2 FFFFF5B4H, TP4IOC2 FFFFF5D4H,
TP5IOC2 FFFFF5E4H

	7	6	5	4	3	2	1	0
TPnIOC2 (n = 0 to 2, 4, 5)	0	0	0	0	TPnEES1	TPnEES0	TPnETS1	TPnETS0

TPnEES1	TPnEES0	External event count input signal (TIPn0 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 (TIPn0 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 (TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 bits = 001) has been set.
 3. The TPnETS1 and TPnETS0 bits are valid only when the external trigger pulse output mode (TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 bits = 010) or the one-shot pulse output mode (TPnCTL1.TPnMD2 to TPnCTL1.TPnMD0 = 011) is set.

(6) TMPn option register 0 (TPnOPT0)

The TPnOPT0 register is an 8-bit register used to set the capture/compare operation and indicate the detection of an overflow.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TP0OPT0 FFFFF595H, TP1OPT0 FFFFF5A5H,
TP2OPT0 FFFFF5B5H, TP3OPT0 FFFFF5C5H,
TP4OPT0 FFFFF5D5H, TP5OPT0 FFFFF5E5H

	7	6	5	4	3	2	1	<0>
TPnOPT0 (n = 0 to 5)	0	0	TPnCCS1	TPnCCS0	0	0	0	TPnOVF

TPnCCS1	TPnCCR1 register capture/compare selection
0	Compare register selected
1	Capture register selected
The TPnCCS1 bit setting is valid only in the free-running timer mode.	

TPnCCS0	TPnCCR0 register capture/compare selection
0	Compare register selected
1	Capture register selected
The TPnCCS0 bit setting is valid only in the free-running timer mode.	

TPnOVF	TMPn overflow detection flag
Set (1)	Overflow occurred
Reset (0)	TPnOVF bit 0 written or TPnCTL0.TPnCE bit = 0
<ul style="list-style-type: none"> The TPnOVF bit is set 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 interrupt request signal (INTTPnOV) is generated at the same time that the TPnOVF bit is set to 1. The INTTPnOV signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode. The TPnOVF bit is not cleared even when the TPnOVF bit or the TPnOPT0 register are read when the TPnOVF bit = 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 influence 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 clear bits 1 to 3, 6, and 7 to "0".**

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

The TPnCCR0 register can be used as a capture register or a compare register depending on the mode.

This register can be selected as a capture register or a compare register only in the free-running timer mode, according to 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.

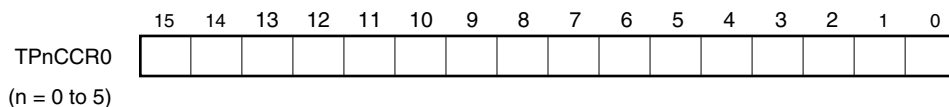
This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

Caution Accessing the TPnCCR0 register is prohibited in the following statuses. Moreover, if the system is in the wait status, the only way to cancel the wait status is to execute a reset. For details, see 3.4.9 (1) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates on the subclock and main clock oscillation is stopped
- When the CPU operates on the internal oscillator clock

After reset: 0000H R/W Address: TP0CCR0 FFFFF596H, TP1CCR0 FFFFF5A6H,
 TP2CCR0 FFFFF5B6H, TP3CCR0 FFFFF5C6H,
 TP4CCR0 FFFFF5D6H, TP5CCR0 FFFFF5E6H



(a) Function as compare register

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

The set value of the TPnCCR0 register is transferred to the CCR0 buffer register. When the value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTPnCC0) is generated. If TOPn0 pin output is enabled at this time, the output of the TOPn0 pin is inverted (For details, see the descriptions of each operating mode.).

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.

(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 (TIPn0 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 (TIPn0) is detected.

Even if the capture operation and reading the TPnCCR0 register conflict, the correct value of the TPnCCR0 register can be read.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 6-3. 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
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	—

Remark For details about anytime write and batch write, see **6.4 (2) Anytime write and batch write**.

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

The TPnCCR1 register can be used as a capture register or a compare register depending on the mode.

This register can be selected 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.

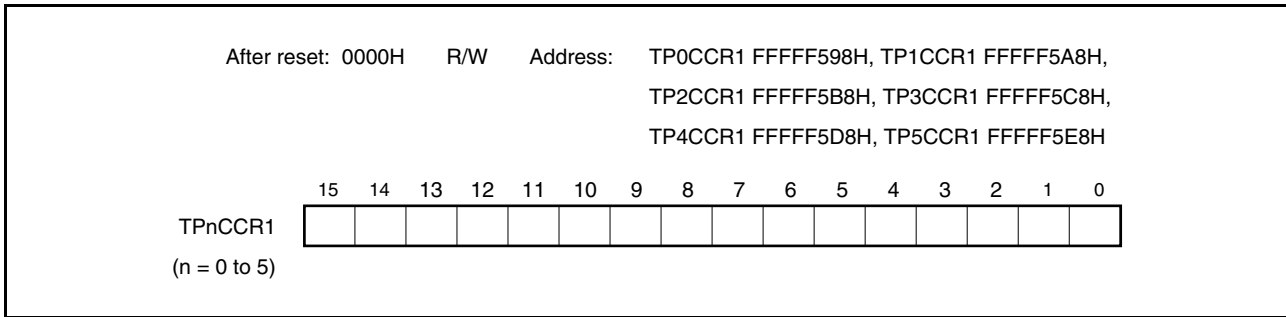
The TPnCCR1 register can be read or written during operation.

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

Reset sets this register to 0000H.

Caution Accessing the TPnCCR1 register is prohibited in the following statuses. Moreover, if the system is in the wait status, the only way to cancel the wait status is to execute a reset. For details, see 3.4.9 (1) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates on the subclock and main clock oscillation is stopped
- When the CPU operates on the internal oscillator clock



(a) Function as compare register

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

The set value of the TPnCCR1 register is transferred to the CCR1 buffer register. When the value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTPnCC1) is generated. If TOPn1 pin output is enabled at this time, the output of the TOPn1 pin is inverted (For details, see the descriptions of each operating mode.).

(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 (TIPn1 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 (TIPn1) is detected.

Even if the capture operation and reading the TPnCCR1 register conflict, the correct value of the TPnCCR1 register can be read.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 6-4. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	—

Remark For details about anytime write and batch write, see **6.4 (2) Anytime write and batch write**.

(9) TMPn counter read buffer register (TPnCNT)

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

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

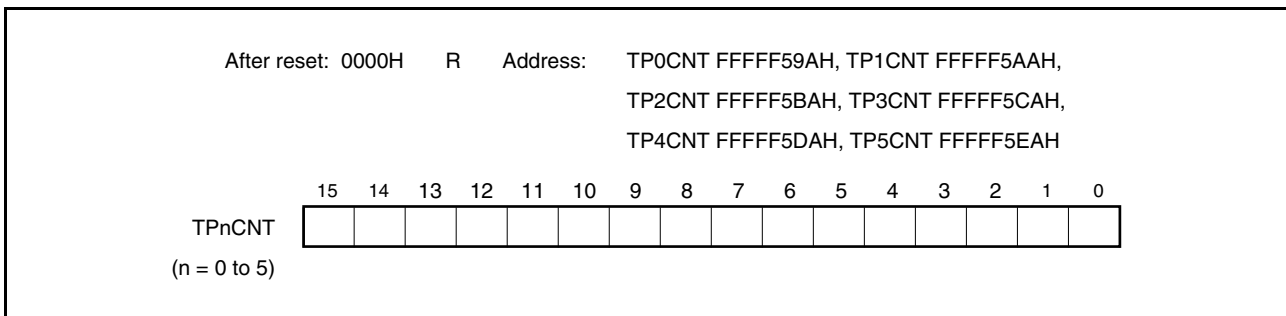
This register is read-only, in 16-bit units.

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.

Because the TPnCE bit is cleared to 0, the value of the TPnCNT register is cleared to 0000H after reset.

Caution Accessing the TPnCNT register is prohibited in the following statuses. Moreover, if the system is in the wait status, the only way to cancel the wait status is to execute a reset. For details, see 3.4.9 (1) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates on the subclock and main clock oscillation is stopped
- When the CPU operates on the internal oscillator clock



6.4 Operations

For the V850ES/JC3-L, V850ES/JE3-L, the modes that can be enabled depend on each channel. The following table shows the modes that can be enabled for each channel.

Table 6-5. TMPn Modes and Channels That Can Use These Modes (V850ES/JC3-L (40-pin))

Operating Mode	TMP0	TMP1	TMP2	TMP3	TMP4	TMP5
Interval timer mode	Conditionally available ^{Note}	Conditionally available ^{Note}	Available	Conditionally available ^{Note}	Not available	Available
External event count mode	Not available	Not available	Available	Not available	Not available	Available
External trigger pulse output mode	Not available	Not available	Available	Not available	Not available	Available
One-shot pulse output mode	Not available	Not available	Available	Not available	Not available	Available
PWM output mode	Not available	Not available	Available	Not available	Not available	Available
Free-running timer mode	Not available	Not available	Available	Not available	Not available	Available
Pulse width measurement mode	Not available	Not available	Available	Not available	Not available	Available

Note Because TIP00, TIP01, TOP00, TOP01, TIP10, TIP11, TOP10, TOP11, TIP30, TIP31, TOP30, TOP31, TIP40, TIP41, TOP40, and TOP41 do not exist, modes enabled by using these pins are not available.

Table 6-6. TMPn Modes and Channels That Can Use These Modes (V850ES/JC3-L (48-pin))

Operating Mode	TMP0	TMP1	TMP2	TMP3	TMP4	TMP5
Interval timer mode	Conditionally available ^{Note}	Conditionally available ^{Note}	Available	Conditionally available ^{Note}	Not available	Available
External event count mode	Conditionally available ^{Note}	Not available	Available	Not available	Not available	Available
External trigger pulse output mode	Conditionally available ^{Note}	Not available	Available	Not available	Not available	Available
One-shot pulse output mode	Conditionally available ^{Note}	Not available	Available	Not available	Not available	Available
PWM output mode	Conditionally available ^{Note}	Not available	Available	Not available	Not available	Available
Free-running timer mode	Conditionally available ^{Note}	Not available	Available	Not available	Not available	Available
Pulse width measurement mode	Conditionally available ^{Note}	Not available	Available	Not available	Not available	Available

Note Because TIP01, TOP01, TIP10, TIP11, TOP10, TOP11, TIP30, TIP31, TOP30, TOP31, TIP40, TIP41, TOP40, and TOP41 do not exist, modes enabled by using these pins are not available.

Table 6-7. TMPn Modes and Channels That Can Use These Modes (V850ES/JE3-L)

Operating Mode	TMP0	TMP1	TMP2	TMP3	TMP4	TMP5
Interval timer mode	Available	Available	Available	Conditionally available ^{Note}	Available	Available
External event count mode	Available	Available	Available	Not available	Available	Available
External trigger pulse output mode	Available	Available	Available	Conditionally available ^{Note}	Available	Available
One-shot pulse output mode	Available	Available	Available	Conditionally available ^{Note}	Available	Available
PWM output mode	Available	Available	Available	Conditionally available ^{Note}	Available	Available
Free-running timer mode	Available	Available	Available	Conditionally available ^{Note}	Available	Available
Pulse width measurement mode	Available	Available	Available	Conditionally available ^{Note}	Available	Available

Note Because TIP30 and TOP30 do not exist, modes enabled by using these pins are not available.

TMPn can execute the following operations:

Table 6-8. TMPn Operating Modes

Operating Mode	TPnCTL1.TPnEST Bit (Software Trigger Bit)	TIPn0 Pin (External Trigger Input)	Capture/Compare Register Setting	Compare Register Write	Count Clock
Interval timer mode	Invalid	Invalid	Compare only	Anytime write	Internal/external
External event count mode ^{Note 1}	Invalid	Invalid	Compare only	Anytime write	External
External trigger pulse output mode ^{Note 2}	Valid	Valid	Compare only	Batch write	Internal
One-shot pulse output mode ^{Note 2}	Valid	Valid	Compare only	Anytime write	Internal
PWM output mode	Invalid	Invalid	Compare only	Batch write	Internal/external
Free-running timer mode	Invalid	Invalid	Can be switched	Anytime write	Internal/external
Pulse width measurement mode ^{Note 2}	Invalid	Invalid	Capture only	Not applicable	Internal

Notes 1. When using the external event count mode, specify that the valid edge of the TIPn0 pin capture trigger input is not detected (by clearing the TPnIOC1.TPnIS1 and TPnIOC1.TPnIS0 bits to 0).

2. When using the external trigger pulse output mode, one-shot pulse output mode, and pulse width measurement mode, select the internal clock as the count clock (by clearing the TPnCTL1.TPnEEE bit to 0).

(1) Basic counter operation

The basic operation of the 16-bit counter is described below. For more details, see the descriptions of each operating mode.

(a) Starting counting

TMPn starts counting from FFFFH in all operating modes, and increments as follows: FFFFH, 0000H, 0001H, 0002H, 0003H....

(b) Clearing TMPn

TMPn is cleared to 0000H when its value matches the value of the compare register or when the value of TMPn is captured upon the input of a valid capture trigger signal.

Note that when TMPn increments from FFFFH to 0000H immediately after it starts counting and following an overflow, it does not mean that TMPn has been cleared. Consequently, the INTTPnCC0 and INTTPnCC1 interrupts are not generated in this case.

(c) Overflow

TMPn overflows after it increments from FFFFH to 0000H in free-running timer mode and pulse width measurement mode. An overflow sets the TPnOPT0.TPnOVF bit to 1 and generates an interrupt request signal (INTTPnOV). Note that INTTPnOV will not be generated in the following cases:

- When TMPn has just started counting.
- When the compare value at which TMPn is cleared is specified as FFFFH.
- In pulse width measurement mode, when TMPn increments from FFFFH to 0000H after being cleared when its value of FFFFH was captured.

Caution After the INTTPnOV overflow interrupt request signal occurs, be sure to confirm that the overflow flag (TPnOVF) is set to 1.

(d) Reading TMPn while it is incrementing

TMPn can be read while it is incrementing by using the TPnCNT register.

Specifically, the value of TMPn can be read by reading the TPnCNT register while the TPnCLT0.TPnCE bit is 1. Note, however, that when the TPnCLT0.TPnCE bit is 0, the value of TMPn is always FFFFH and the value of the TPnCNT register is always 0000H.

(2) Anytime write and batch write

The TPnCCR0 and TPnCCR1 registers can be written even while TMPn is operating (that is, while the TPnCTL0.TPnCE bit is 1), but the way the CCR0 and CCR1 registers are written differs depending on the mode. The two writing methods are anytime write and batch write.

(a) Anytime write

This writing method is used to transfer data from the TPnCCR0 and TPnCCR1 registers to the CCR0 and CCR1 buffer registers any time while TMPn is operating.

Figure 6-2. Flowchart Showing Basic Anytime Write Operation

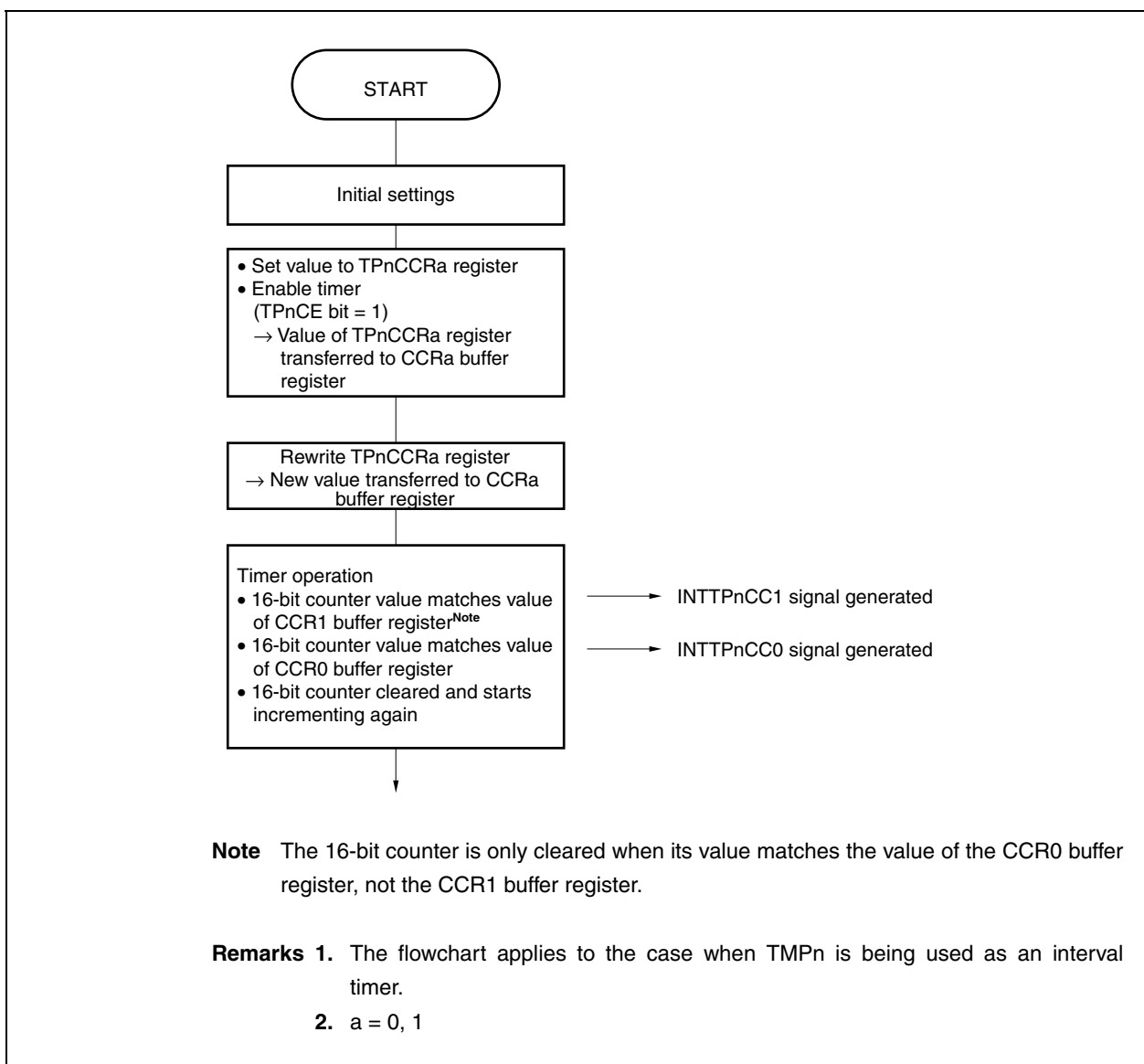
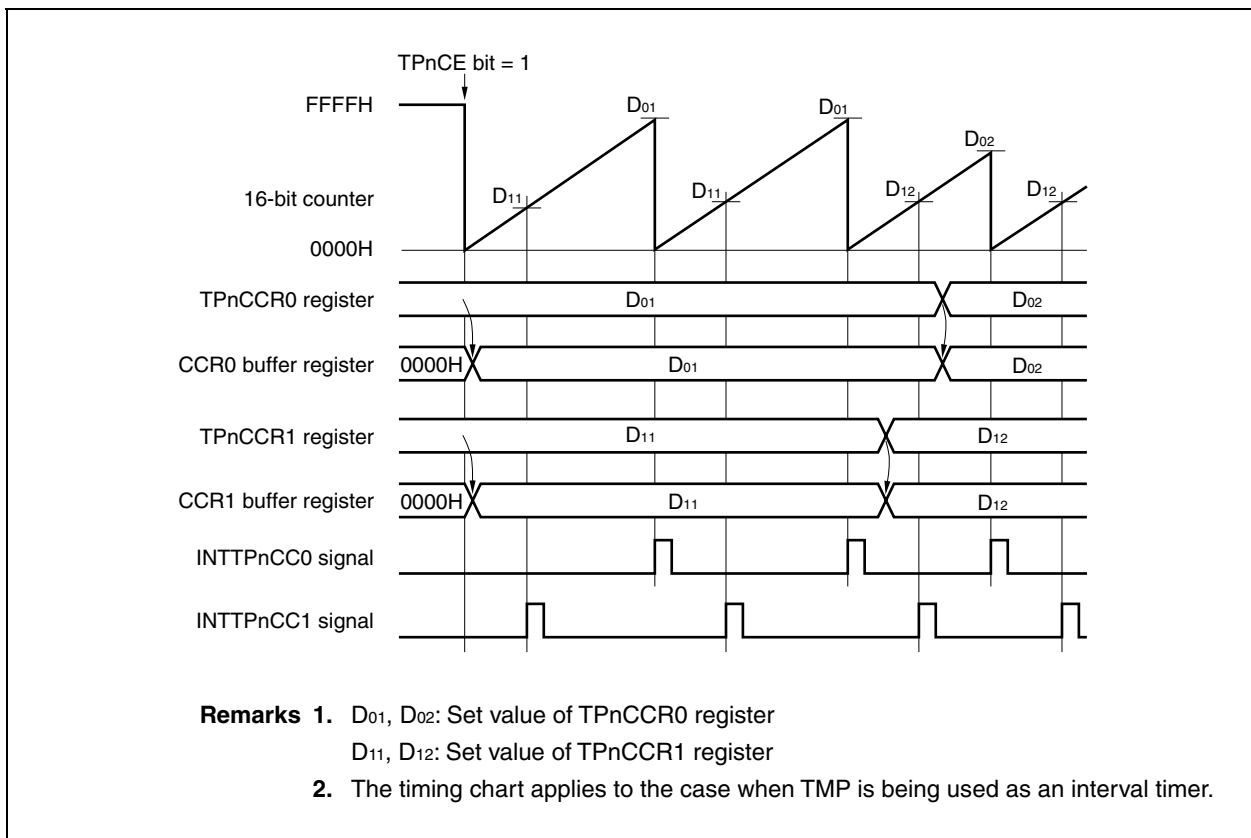


Figure 6-3. Anytime Write Timing



(b) Batch write

This writing method is used to transfer data from the TPnCCR0 and TPnCCR1 registers to the CCR0 and CCR1 buffer registers all at once while TMPn is operating. The data is transferred when the value of the 16-bit counter matches the value of the CCR0 buffer register. Transfer is enabled by writing to the TPnCCR1 register. Whether transfer of the next data is enabled or not depends on whether the TPnCCR1 register has been written.

To specify the value of the rewritten TPnCCR0 and TPnCCR1 registers as the 16-bit counter compare value (that is, the value to be transferred to the CCR0 and CCR1 buffer registers), the TPnCCR0 register must be rewritten before the value of the 16-bit counter matches the value of the CCR0 buffer register, and then the TPnCCR1 register must be written. The value of the TPnCCR0 and TPnCCR1 registers is then transferred to the CCR0 and CCR1 buffer registers when the value of the 16-bit counter matches the value of the CCR0 buffer register. Note that even if you wish to rewrite only the TPnCCR0 register value, you must also write the same value to the TPnCCR1 register (that is, the same value as the value already specified for the TPnCCR1 register).

Figure 6-4. Flowchart Showing Basic Batch Write Operation

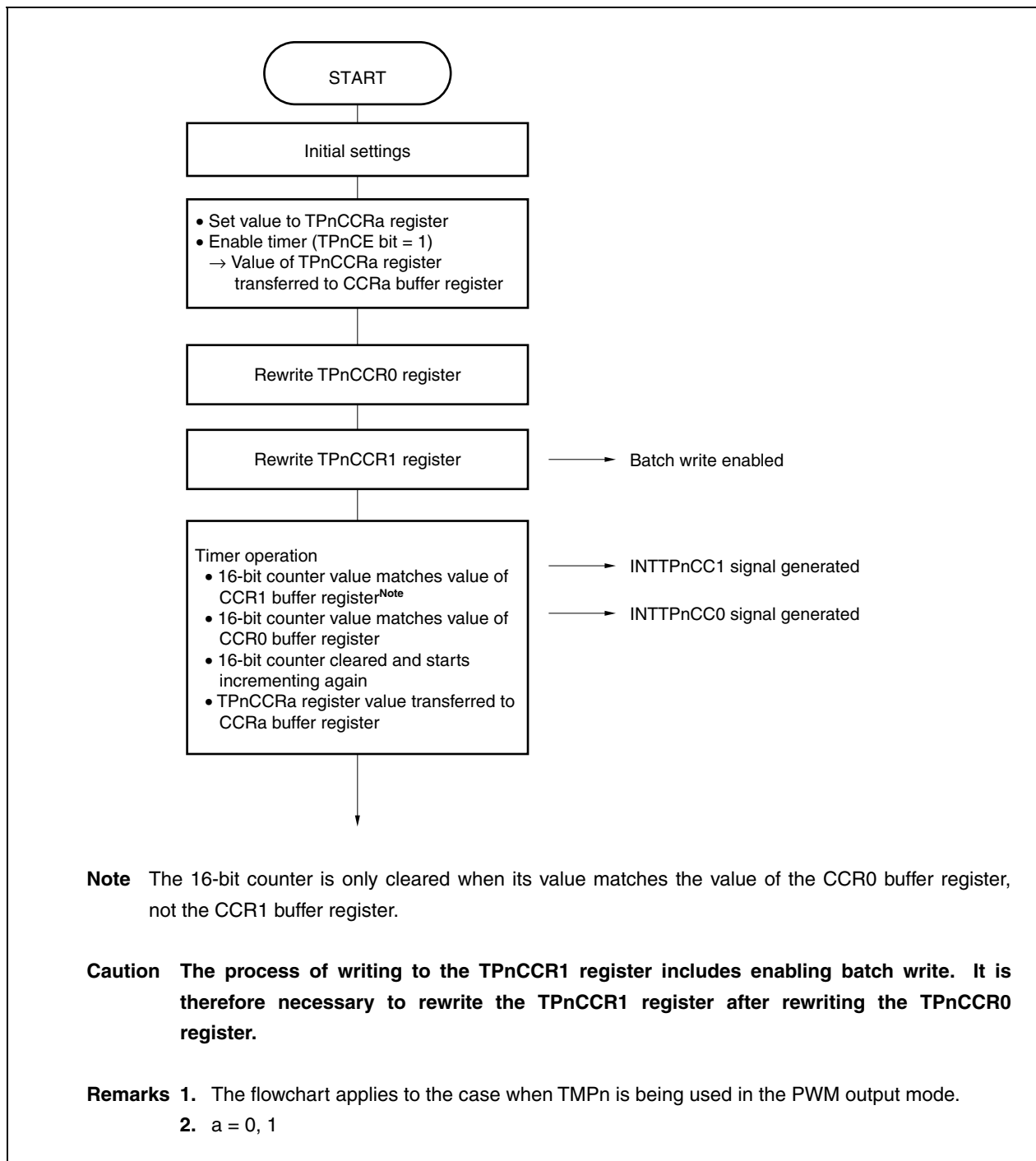
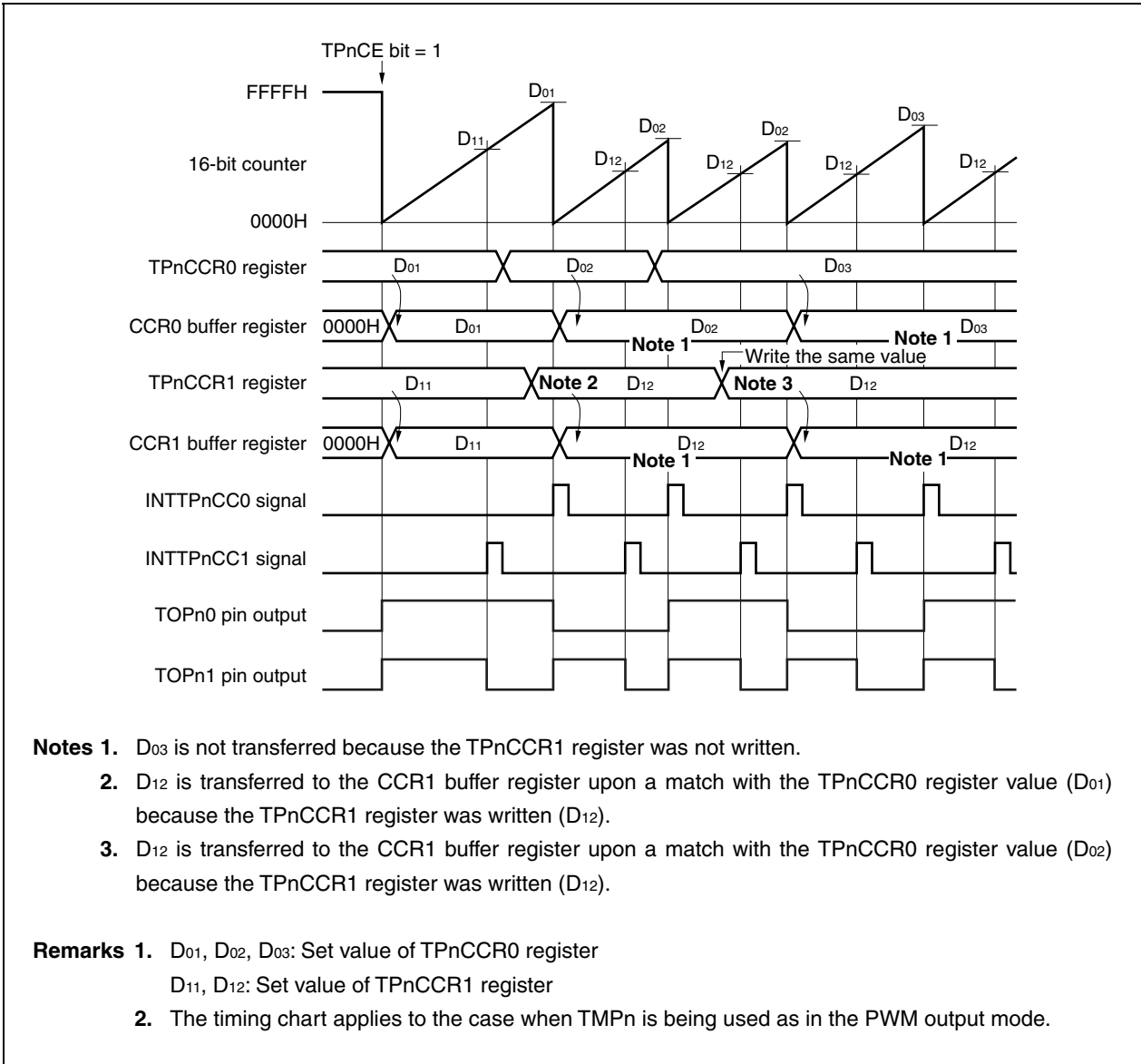


Figure 6-5. Batch Write Timing



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

In the interval timer mode, setting the TPnCTL0.TPnCE bit to 1 generates an interrupt request signal (INTTPnCC0) at a specified interval. Setting the TPnCE bit to 1 can also start the timer, which then outputs a square wave whose half cycle is equal to the interval from the TOPn0 pin.

Usually, the TPnCCR1 register is not used in the interval timer mode. Mask interrupts from this register by setting the interrupt mask flag (TPnCCMK1).

- Remarks 1.** For how to set the TOPn0 pin, see **Table 6-2 Pins Used by TMPn** and **Table 4-17 Settings When Pins Are Used for Alternate Functions**.
- 2.** For how to enable the INTTPnCC0 interrupt signal, see **CHAPTER 19 INTERRUPT SERVICING/ EXCEPTION PROCESSING FUNCTION**.

Figure 6-6. Configuration of Interval Timer

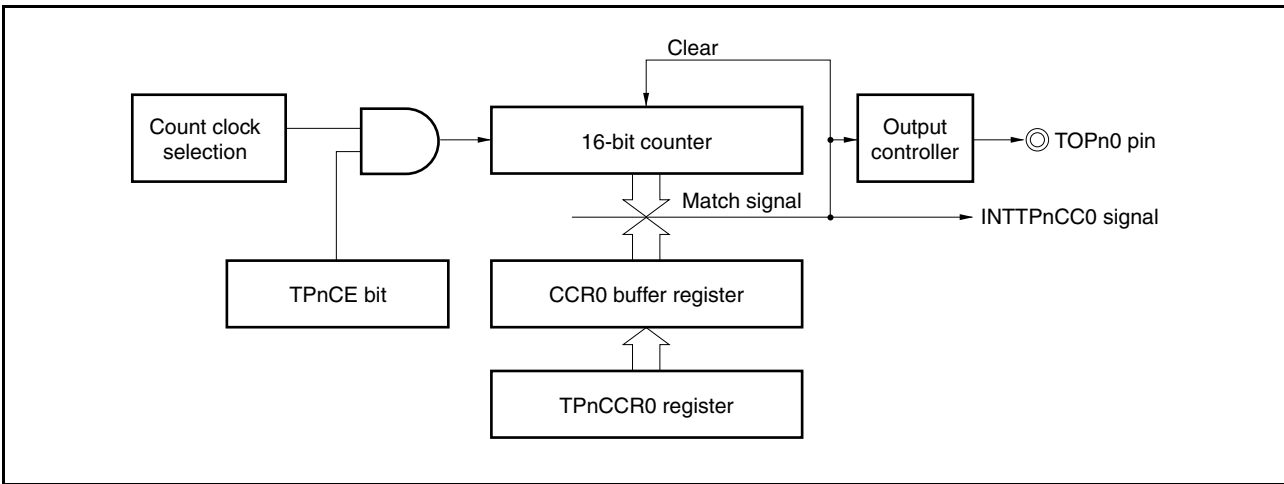
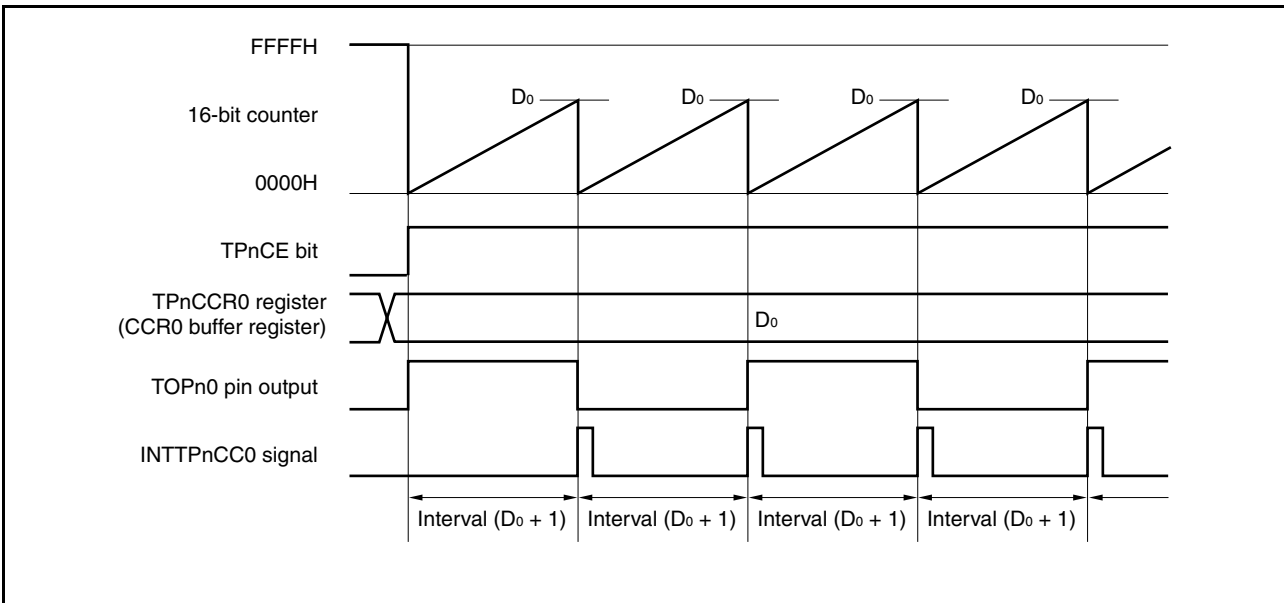


Figure 6-7. Basic Timing of Operation in Interval Timer Mode



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 incrementing. At this time, the output of the TOPn0 pin is inverted and 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, the 16-bit counter is cleared to 0000H, the output of the TOPn0 pin is inverted, and a compare match interrupt request signal (INTTPnCC0) is generated.

The interval can be calculated by using the following expression:

$$\text{Interval} = (\text{Set value of TPnCCR0 register} + 1) \times \text{Count clock cycle}$$

An example of the register settings when the interval timer mode is used is shown in the figure below.

Figure 6-8. Register Settings in Interval Timer Mode (1/2)

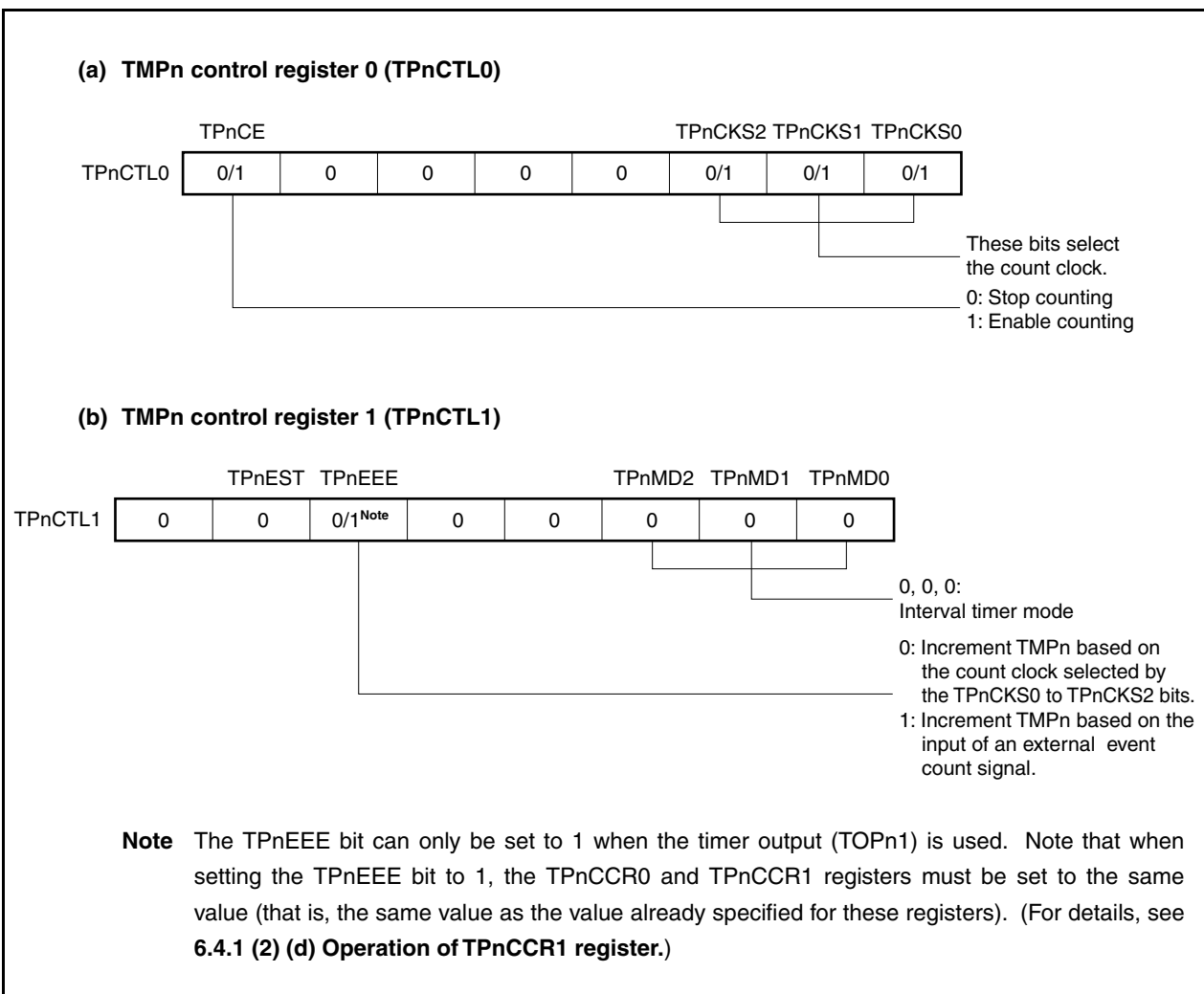
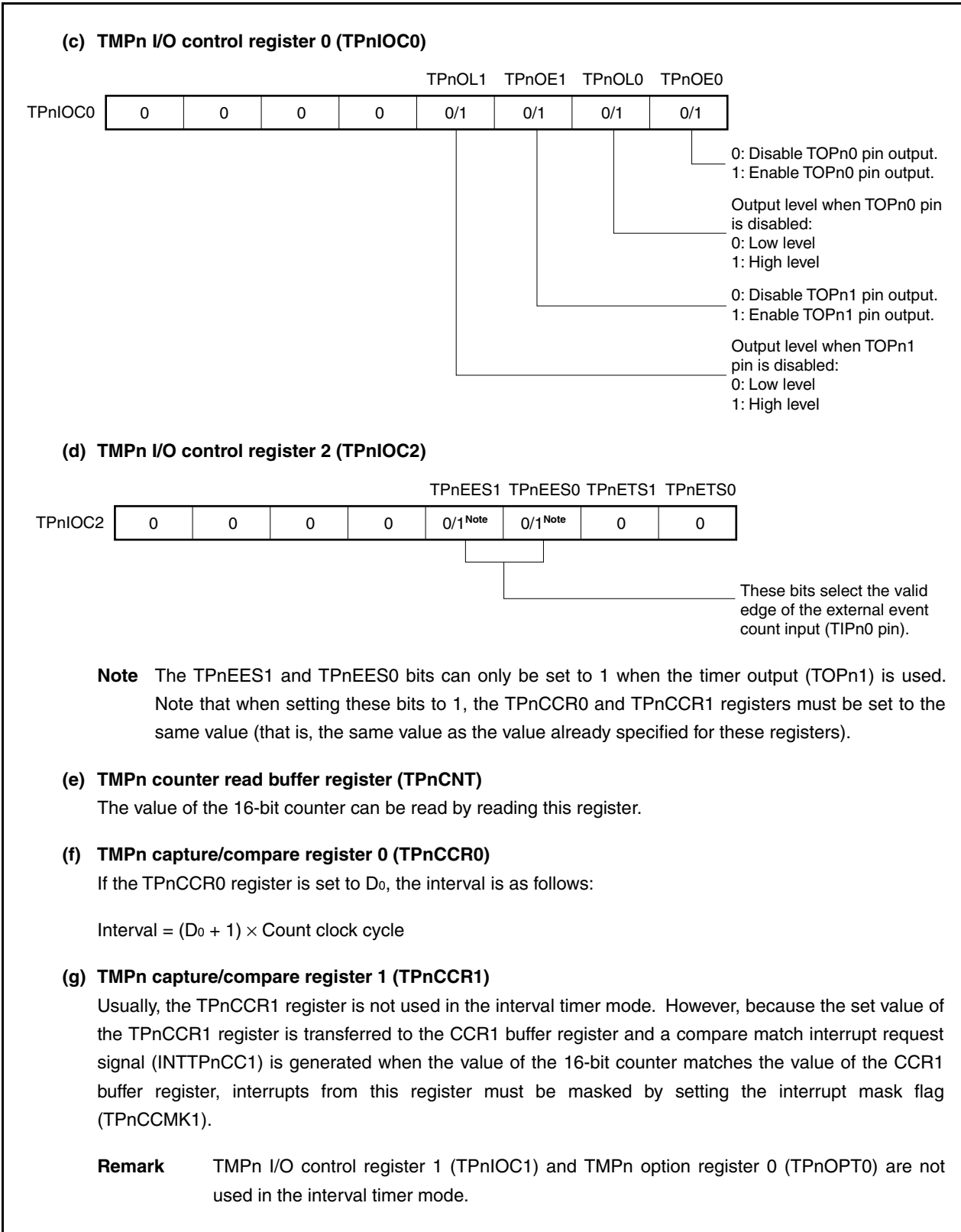
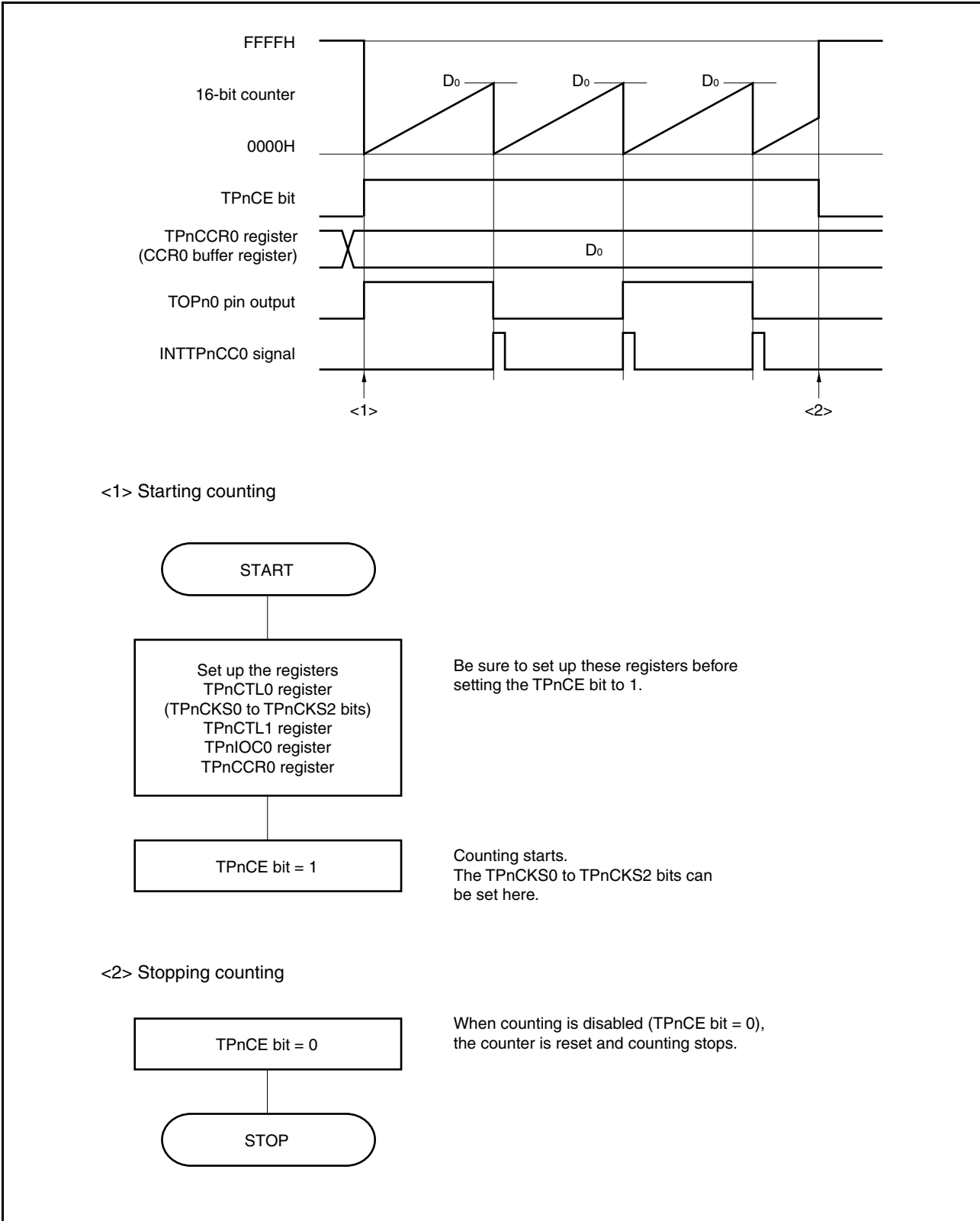


Figure 6-8. Register Settings in Interval Timer Mode (2/2)



(1) Operations in interval timer mode

Figure 6-9. Timing and Processing of Operations in Interval Timer Mode



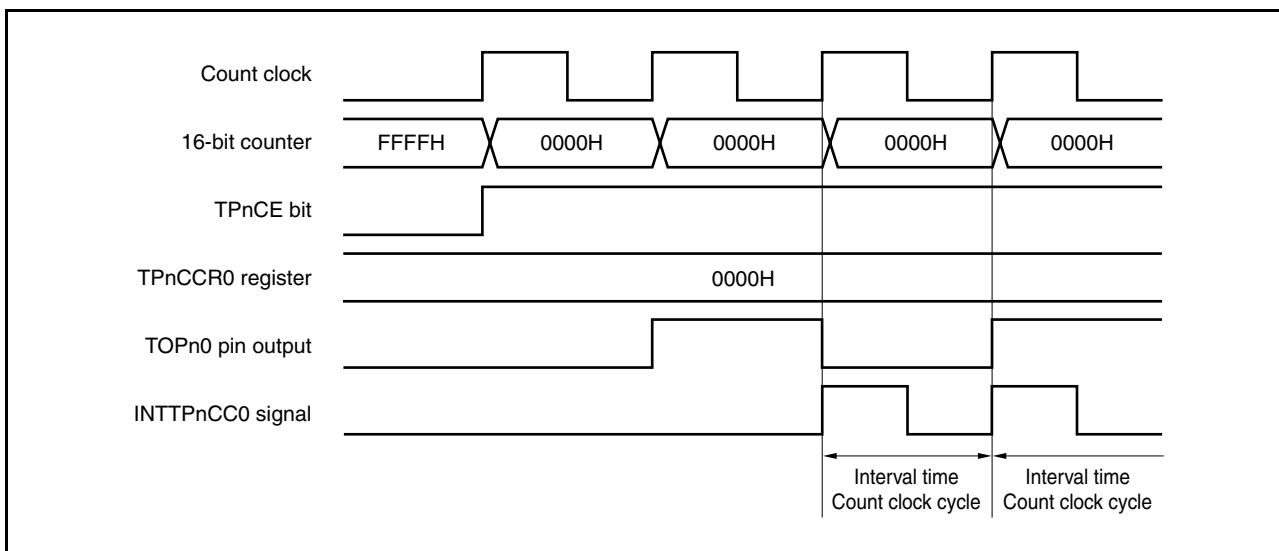
(2) Using interval timer mode

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

When the TPnCCR0 register is set to 0000H, the INTTPnCC0 signal is generated each count clock cycle from the second clock cycle, and the output of the TOPn0 pin is inverted.

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

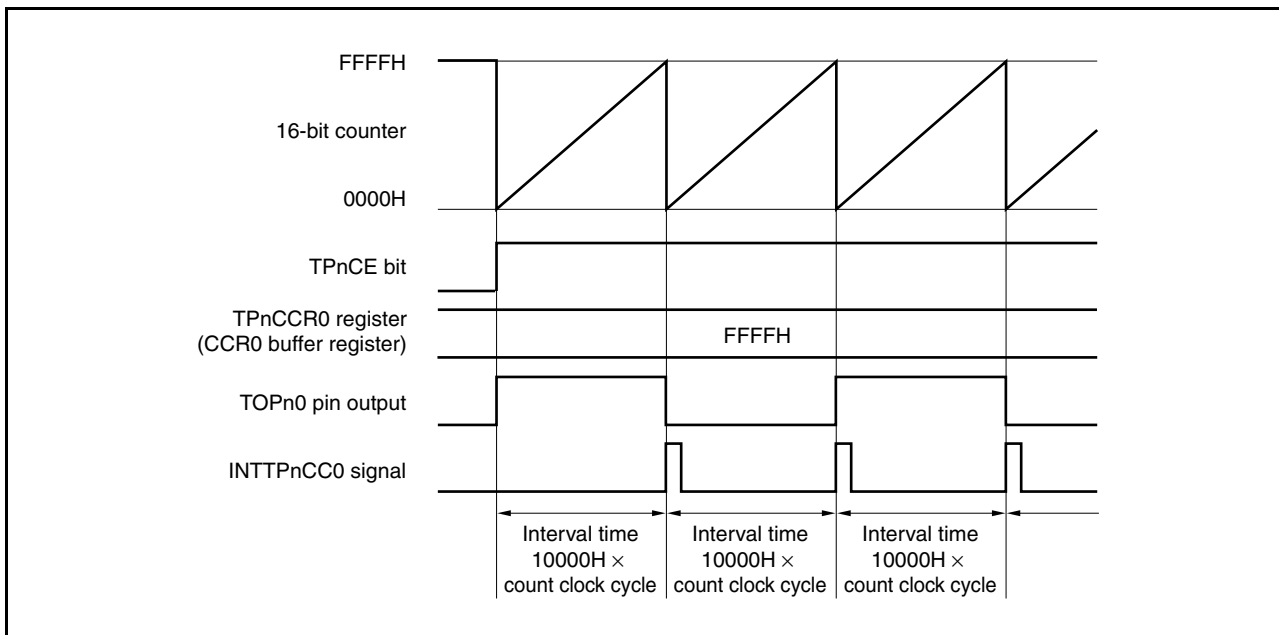
Figure 6-10. Operation of Interval Timer When TPnCCR0 Register Is Set to 0000H



(b) Operation when TPnCCR0 register is set to FFFFH

When the TPnCCR0 register is set to FFFFH, the 16-bit counter increments up to FFFFH and is reset to 0000H in synchronization with the next increment timing. The INTTPnCC0 signal is then generated and the output of the TOPn0 pin is inverted. At this time, an overflow interrupt request signal (INTTPnOV) is not generated, nor is the overflow flag (TPnOPT0.TPnOVF bit) set to 1.

Figure 6-11. Operation of Interval Timer When TPnCCR0 Register Is Set to FFFFH

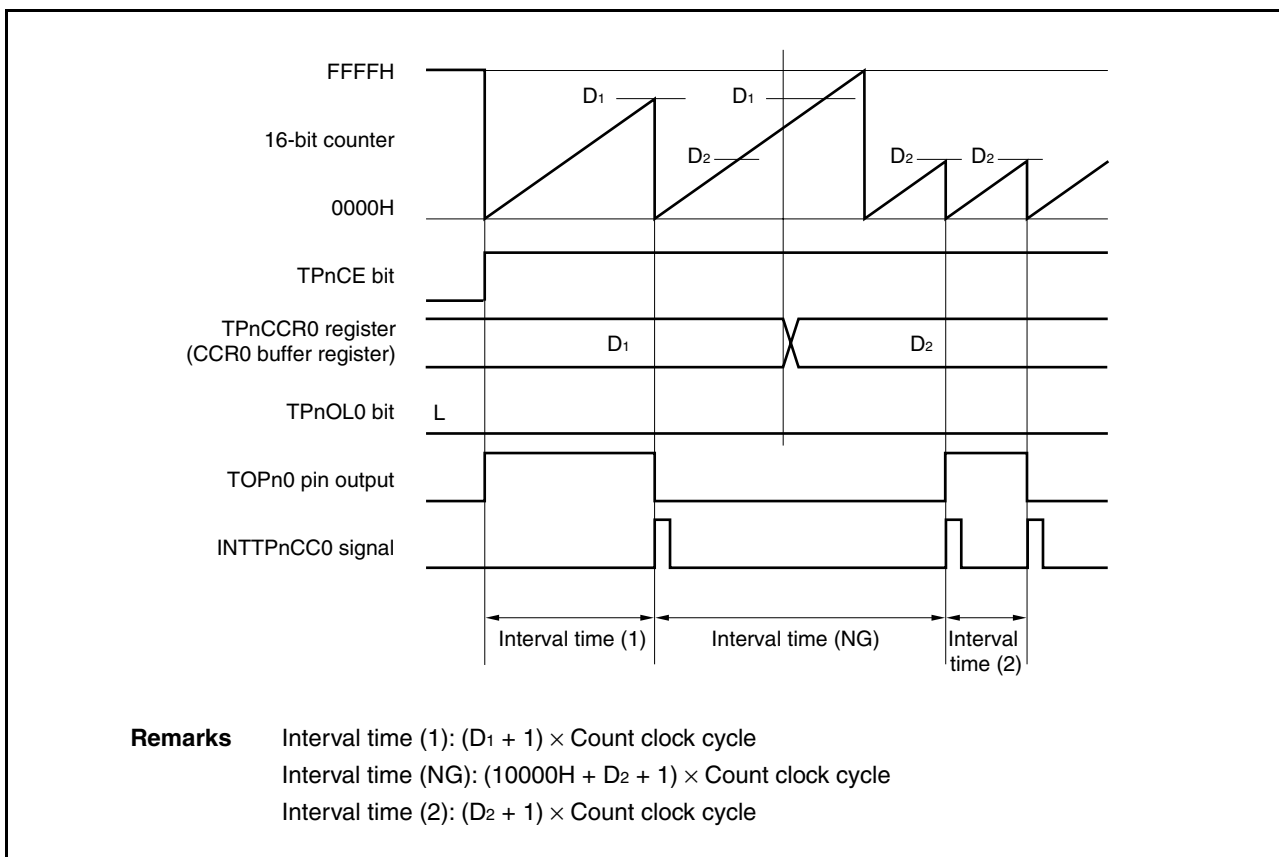


(c) Notes on rewriting TPnCCR0 register

When rewriting the value of the TPnCCR0 register to a smaller value, stop counting first and then change the set value.

If the value of the TPnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.

Figure 6-12. Rewriting TPnCCR0 Register



If the value of the TPnCCR0 register is changed from D_1 to D_2 while the counter value is greater than D_2 but less than D_1 , the TPnCCR0 register value is transferred to the CCR0 buffer register as soon as the register has been rewritten. Consequently, the value that is compared with the 16-bit counter value is D_2 .

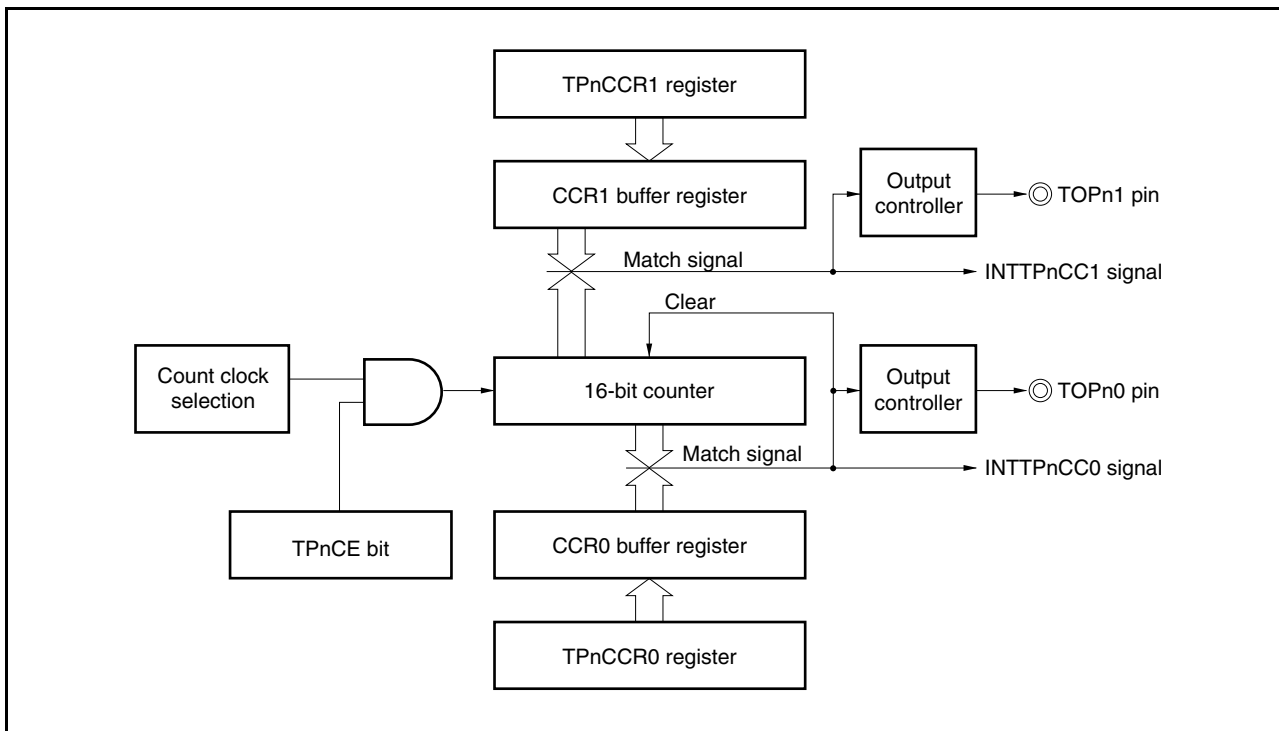
Because the counter value has already exceeded D_2 , however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the counter value matches D_2 , the INTTPnCC0 signal is generated and the output of the TOPn0 pin is inverted.

Therefore, the INTTPnCC0 signal may not be generated at the interval " $(D_1 + 1) \times \text{Count clock cycle}$ " or " $(D_2 + 1) \times \text{Count clock cycle}$ " as originally expected, but instead may be generated at an interval of " $(10000H + D_2 + 1) \times \text{Count clock cycle}$ ".

(d) Operation of TPnCCR1 register

The TPnCCR1 register is configured as follows in the interval timer mode.

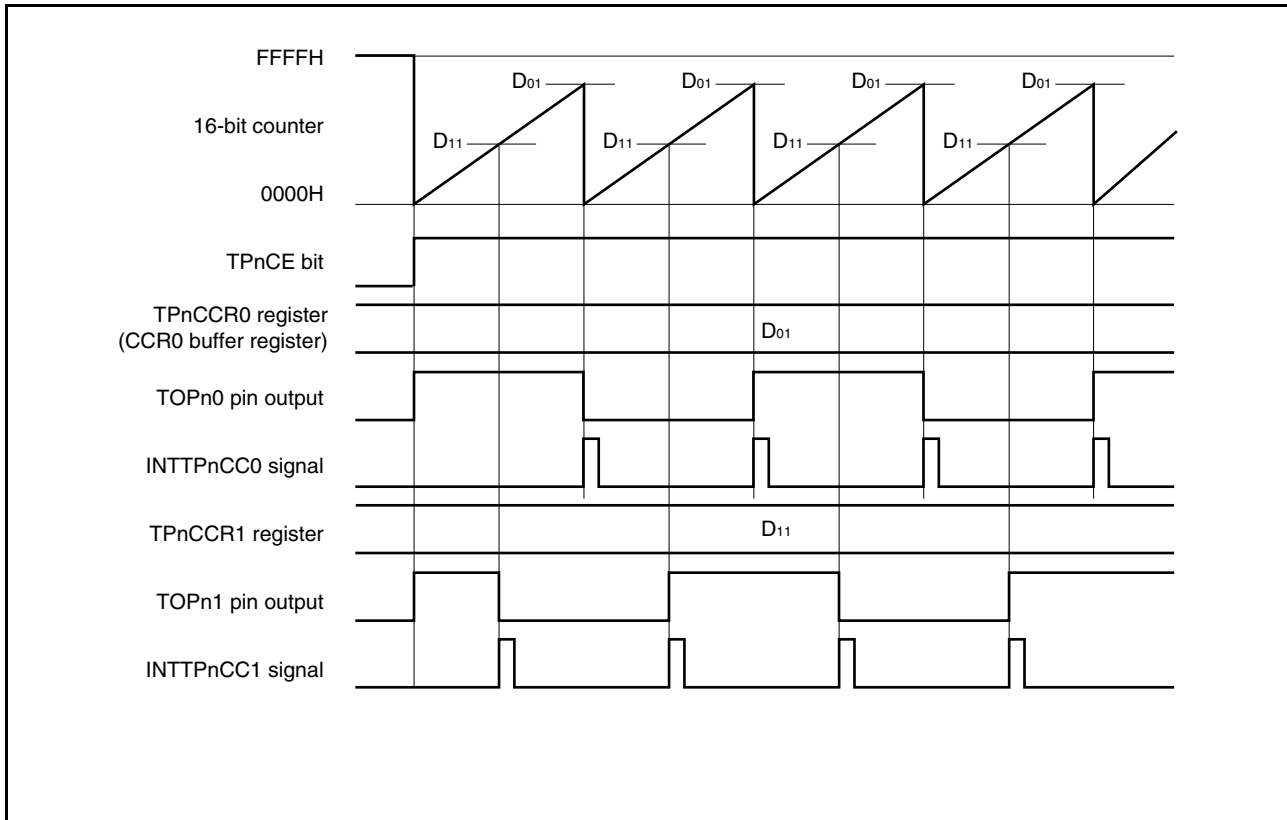
Figure 6-13. Configuration of TPnCCR1 Register



If the value of the TPnCCR1 register is less than or equal to the value of the TPnCCR0 register, the INTTPnCC1 signal is generated once per cycle. At the same time, the output of the TOPn1 pin is inverted. The TOPn1 pin outputs a square wave with the same cycle as that output by the TOPn0 pin but with a different phase.

A chart showing the timing of operations when the value of the TPnCCR1 register (D_{11}) is less than or equal to the value of the TPnCCR0 register (D_{01}) is shown below.

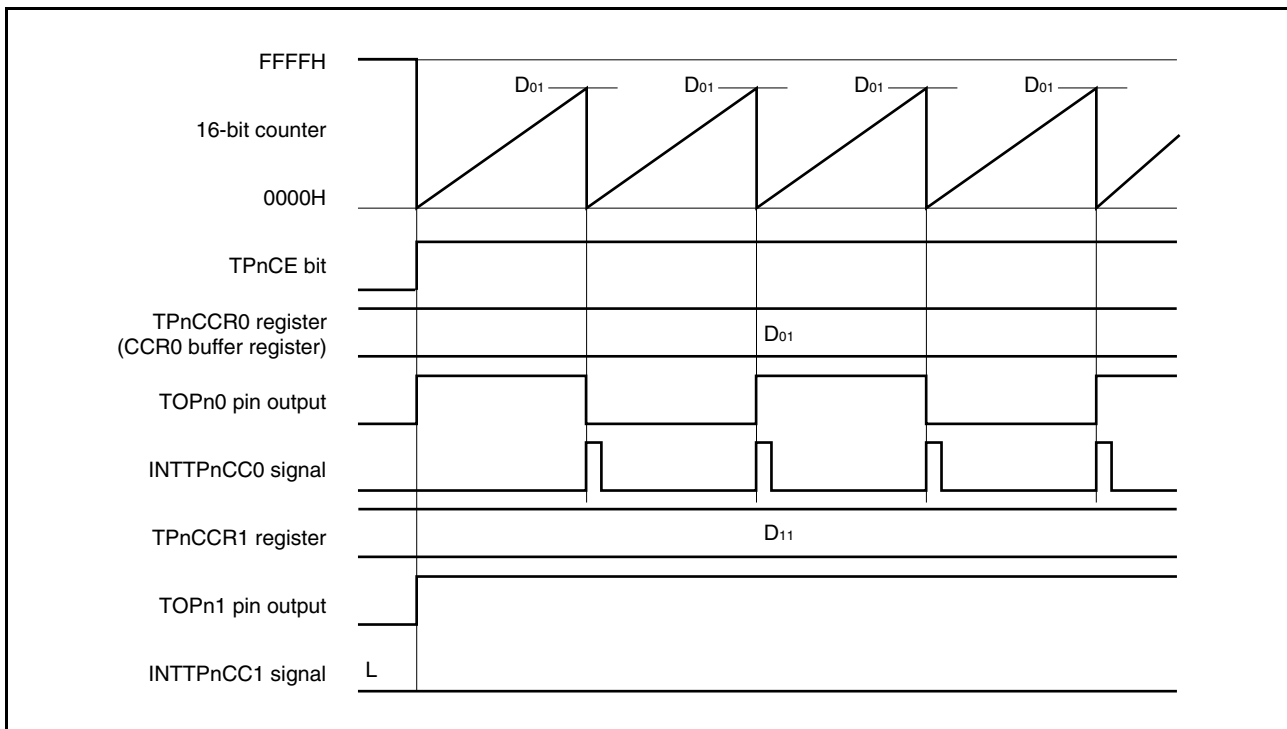
Figure 6-14. Timing of Operations When $D_{01} \geq D_{11}$



If the value of the TPnCCR1 register is greater than the value of the TPnCCR0 register, the value of the 16-bit counter will not match the value of the TPnCCR1 register. Consequently, the INTTPnCC1 signal is not generated, nor is the output of the TOPn1 pin changed.

A chart showing the timing of operations when the value of the TPnCCR1 register (D_{11}) is greater than the value of the TPnCCR0 register (D_{01}) is shown below.

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



(3) Operation of interval timer based on input of external event count

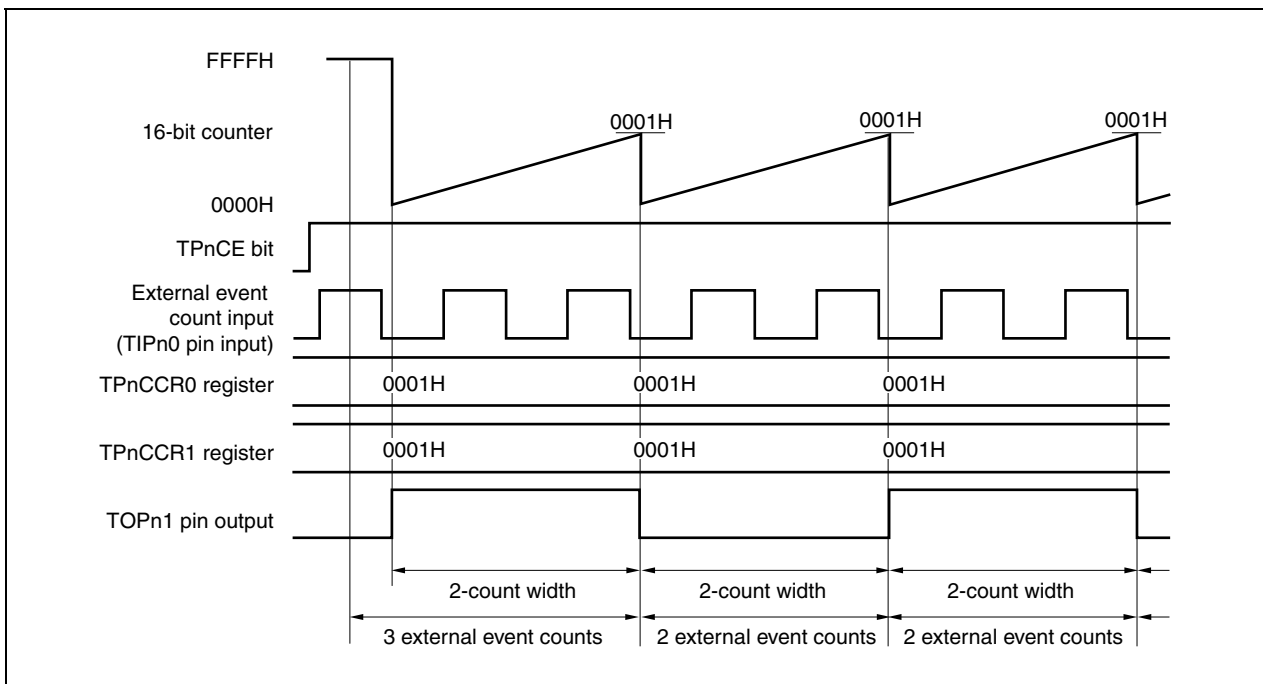
(a) Operation

When the 16-bit counter is incrementing based on the valid edge of the external count input (TIPn0 pin) in the interval timer mode, one external event count valid edge must be input immediately after the TPnCE bit changes from 0 to 1 to start the counter incrementing after the 16-bit counter is cleared from FFFFH to 0000H.

Once the TPnCCR0 and TPnCCR1 registers are set to 0001H (that is, the same value as was previously set), the TOPn1 pin output is inverted every two counts of the 16-bit counter.

Note that the TPnCTL1.TPnEEE bit can only be set to 1 when timer output (TOPn1) is used based on the input of an external event count.

Figure 6-16. Operation of Interval Timer Based on Input of External Event Count (TIPn0)



6.4.2 External event count mode (TPnMD2 to TPnMD0 bits = 001)

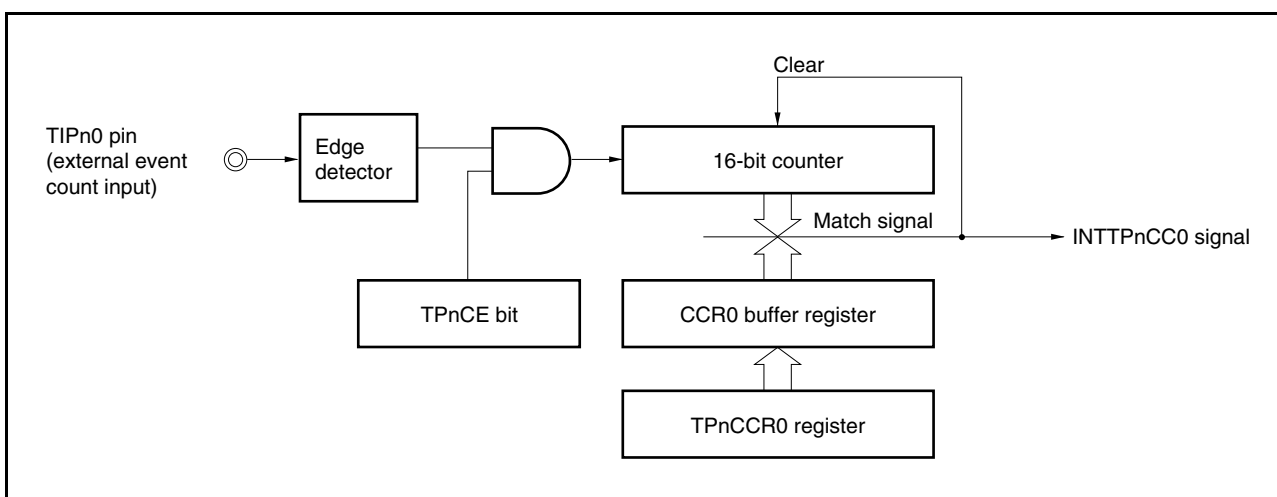
In the external event count mode, the valid edge of the external event count input is counted when the TPnCTL0.TPnCE bit is set to 1, and an interrupt request signal (INTTPnCC0) is generated each time the specified number of edges have been counted. The timer output pins (TOPn0 and TOPn1) cannot be used. To use the TOPn1 pin in the external event count mode, first set the TPnCTL1.TPnEEE bit to 1 in the interval timer mode (see 6.4.1 (3) **Operation of interval timer based on input of external event count**).

Usually, the TPnCCR1 register is not used in the external event count mode.

Remarks 1. For how to set the TIPn0 pin, see **Table 6-2 Pins Used by TMPn** and **Table 4-15 Settings When Pins Are Used for Alternate Functions**.

2. For how to enable the INTTPnCC0 interrupt signal, see **CHAPTER 19 INTERRUPT SERVICING/ EXCEPTION PROCESSING FUNCTION**.

Figure 6-17. Configuration of Interval Timer in External Event Count Mode

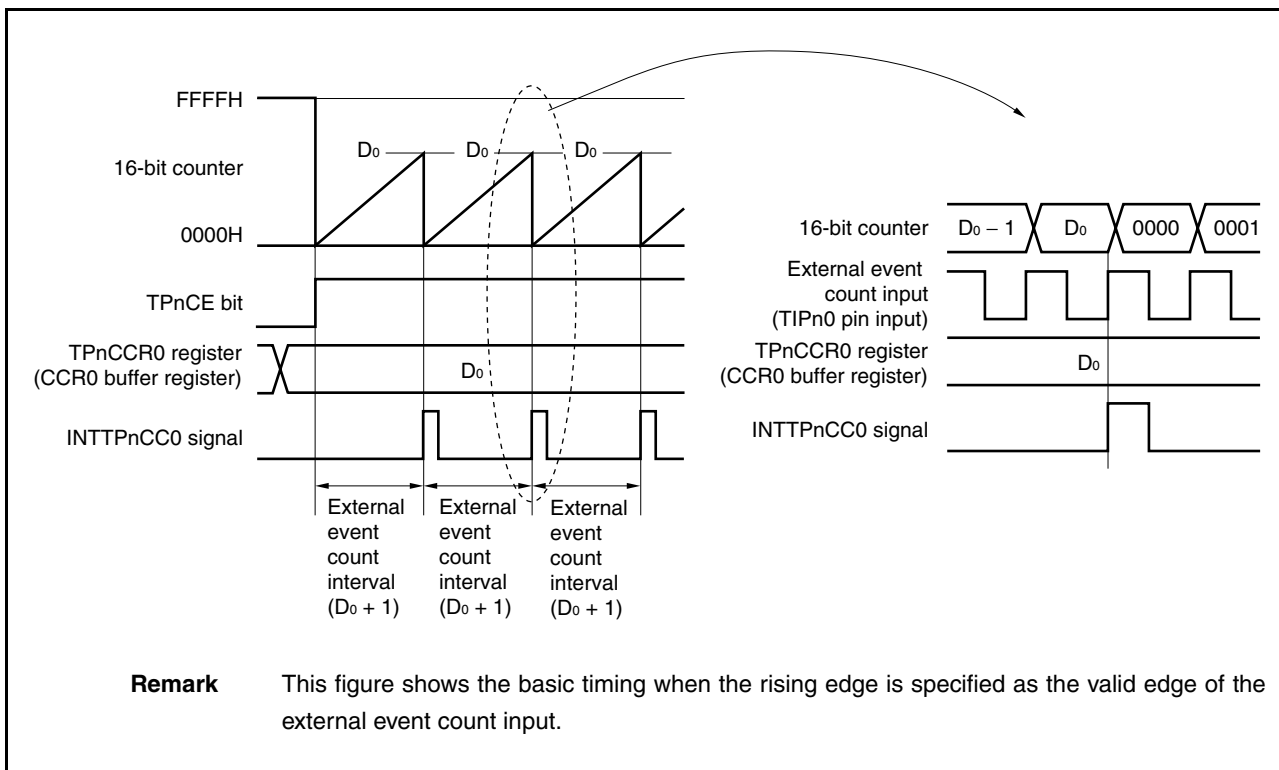


When the TPnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H. The counter increments each time the valid edge of the external event count input is detected, and the value of the TPnCCR0 register is transferred to the CCR0 buffer register.

When the value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTPnCC0) is generated.

The INTTPnCC0 signal is generated each time the valid edge of the external event count input has been detected the specified number of times (that is, the value of the TPnCCR0 register + 1).

Figure 6-18. Basic Timing of Operations in External Event Count Mode



An example of the register settings when the external event count mode is used is shown in the figure below.

Figure 6-19. Register Settings in External Event Count Mode (1/2)

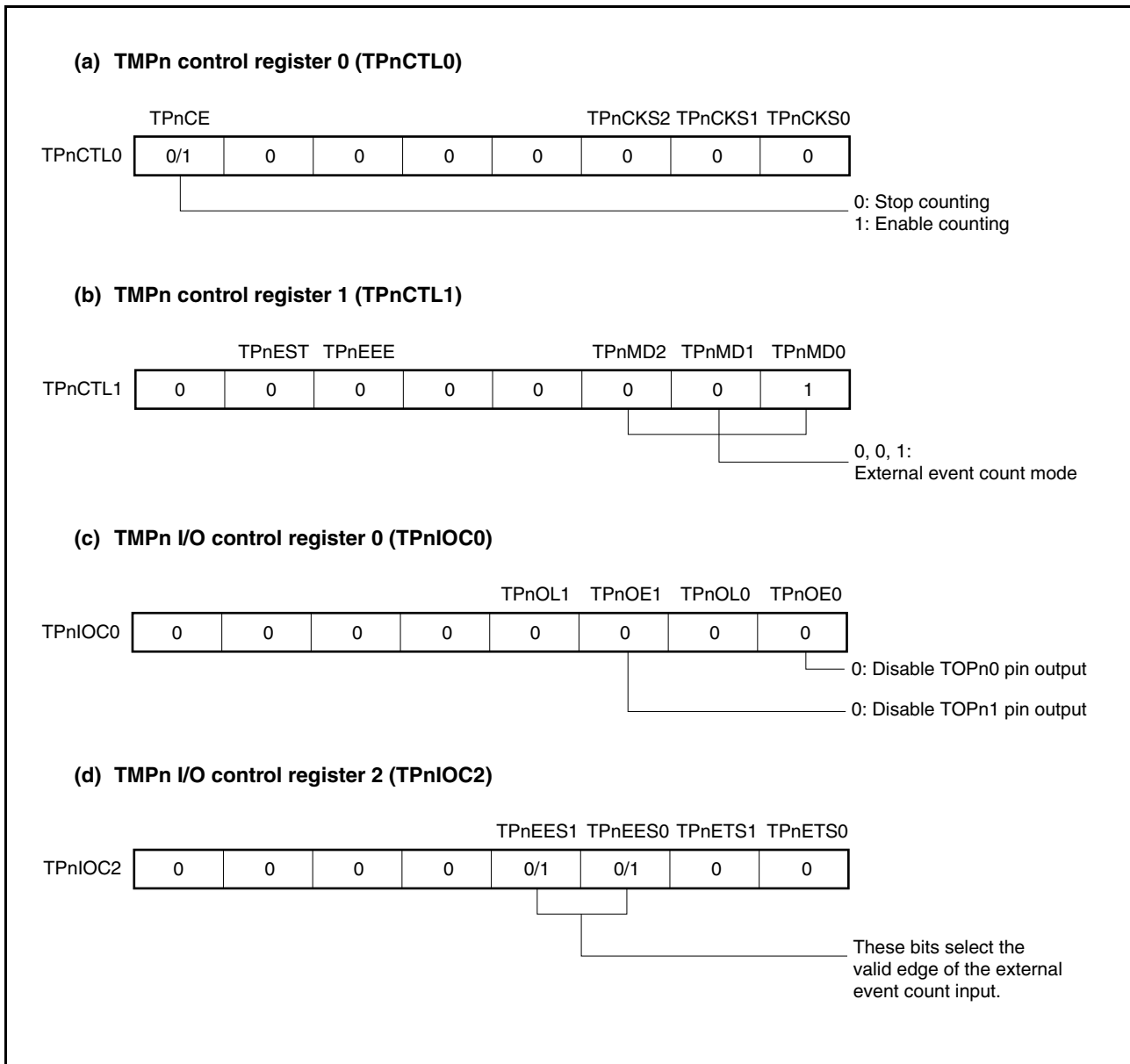


Figure 6-19. Register Settings in External Event Count Mode (2/2)**(e) TMPn counter read buffer register (TPnCNT)**

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

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

When the TPnCCR0 register is set to D_0 , the counter is cleared and a compare match interrupt request signal (INTTPnCC0) is generated when the number of external events reaches $(D_0 + 1)$.

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

The TPnCCR1 register is not usually used in the external event count mode. However, because the set value of the TPnCCR1 register is transferred to the CCR1 buffer register and a compare match interrupt request signal (INTTPnCC1) is generated when the value of the 16-bit counter matches the value of the CCR1 buffer register, interrupts from this register must be masked by setting the interrupt mask flag (TPnCCMK1).

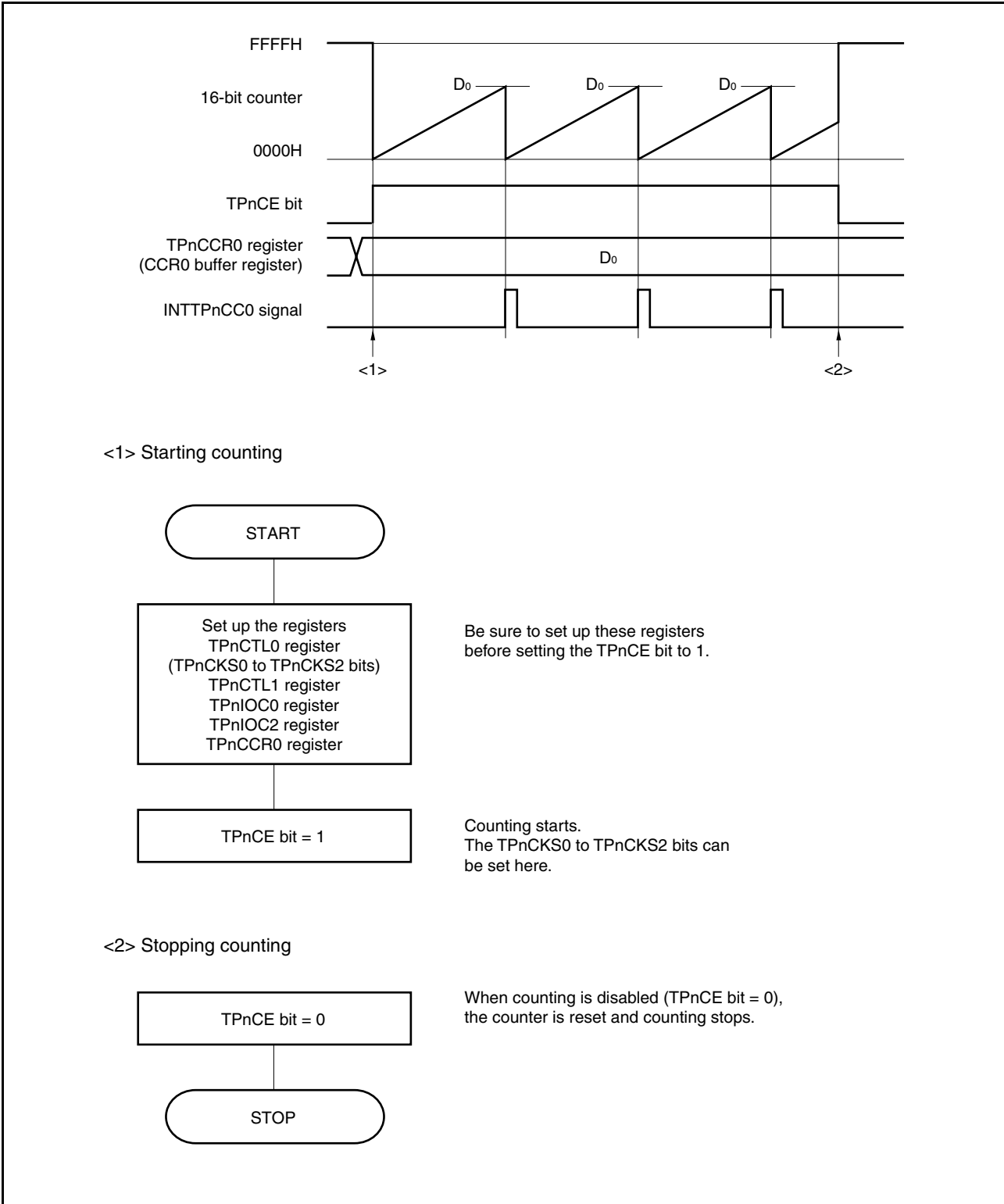
Cautions 1. Do not set the TPnCCR0 register to 0000H in the external event count mode.

- 2. Timer output cannot be used in the external event count mode. When using the timer output based on the input of an external event count, first set the operating mode to interval mode, and then specify "operation enabled" for the external event count input (by setting the TPnCTL1.TPnMD2 to TPnMD0 bits to 0, 0, 0 and setting the TPnCTL1.TPnEEE bit to 1). For details, see 6.4.1 (3) Operation of interval timer based on input of external event count.**

Remarks TMPn I/O control register 1 (TPnIOC1) and TMPn option register 0 (TPnOPT0) are not used in the external event count mode.

(1) Operations in external event count mode

Figure 6-20. Timing and Processing of Operations in External Event Count Mode

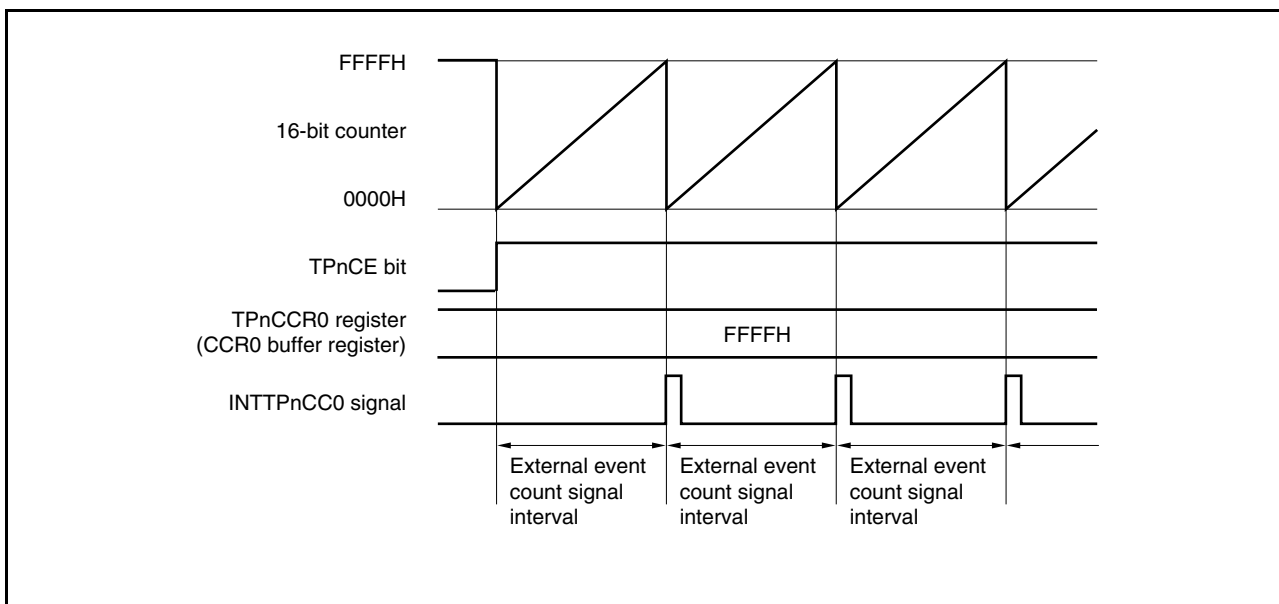


(2) Using external event count mode

(a) Operation when TPnCCR0 register is set to FFFFH

When the TPnCCR0 register is set to FFFFH, the 16-bit counter increments up to FFFFH upon detection of the valid edge of the external event count signal and is reset to 0000H in synchronization with the next increment timing. The INTTPnCC0 signal is then generated. At this time, the TPnOPT0.TPnOVF bit is not set to 1.

Figure 6-21. Operation When TPnCCR0 Register Is Set to FFFFH

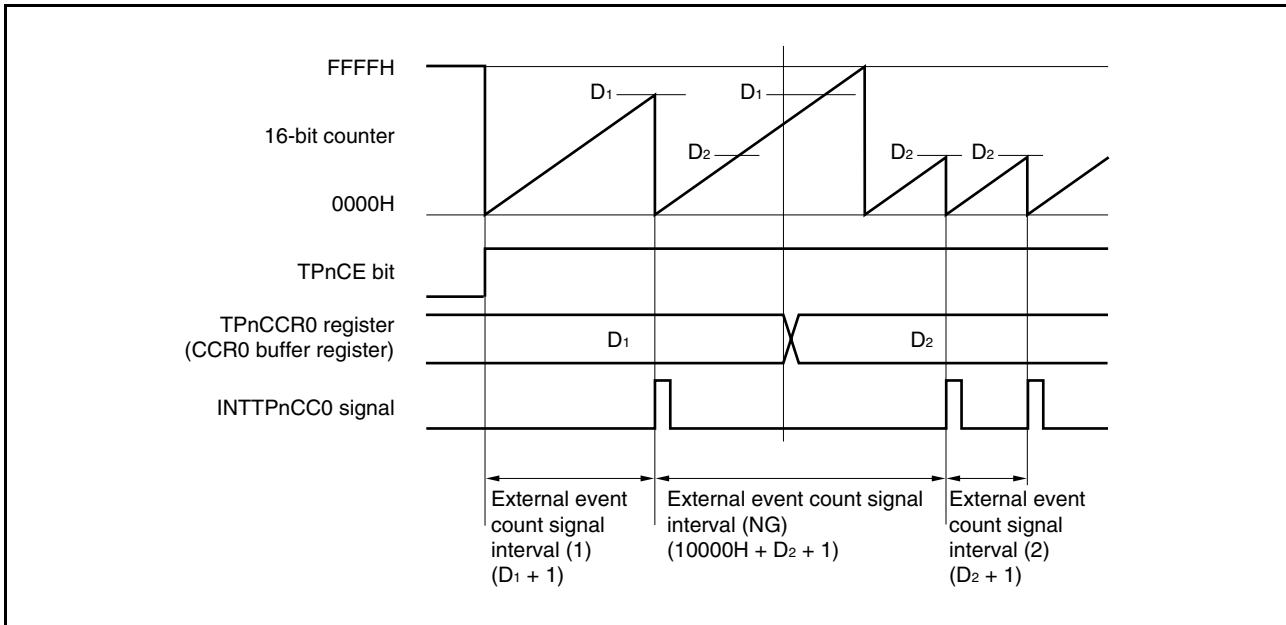


(b) Notes on rewriting TPnCCR0 register

When rewriting the value of the TPnCCR0 register to a smaller value, stop counting first and then change the set value.

If the value of the TPnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.

Figure 6-22. Rewriting TPnCCR0 Register



If the value of the TPnCCR0 register is changed from D1 to D2 while the counter value is greater than D2 but less than D1, the TPnCCR0 register value is transferred to the CCR0 buffer register as soon as the register has been rewritten. Consequently, the value that is compared with the 16-bit counter value is D2.

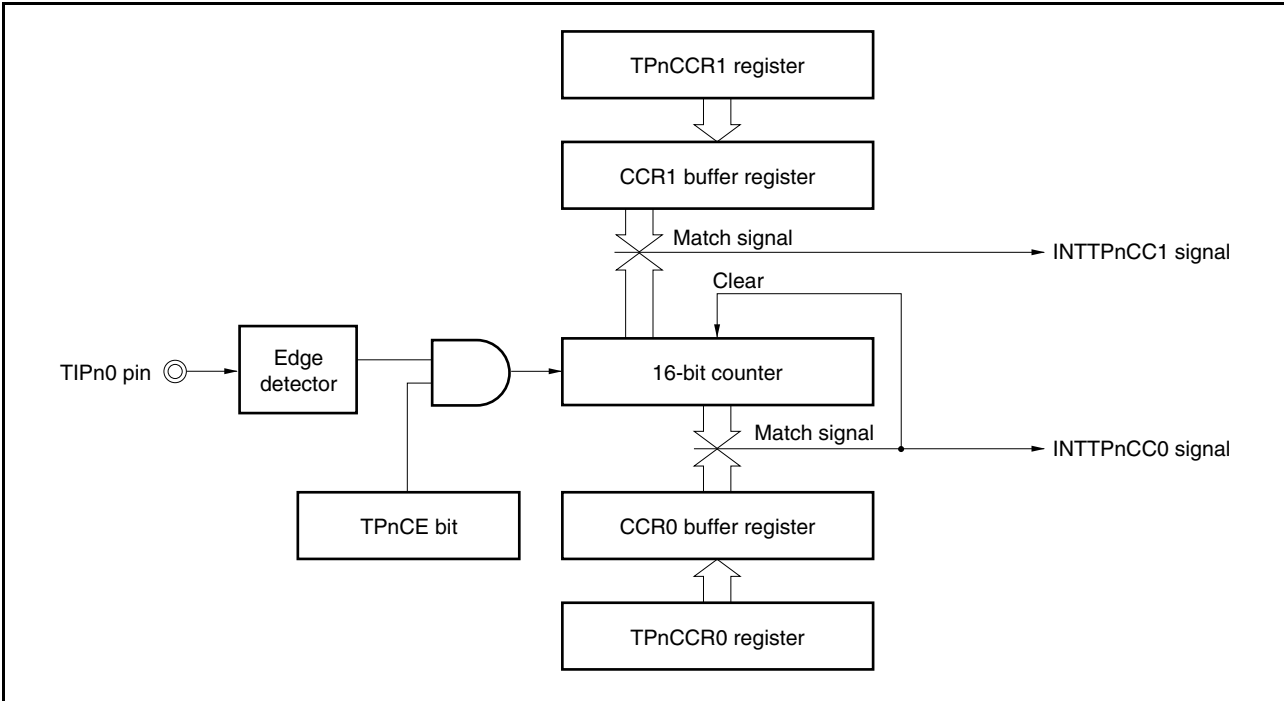
Because the counter value has already exceeded D2, however, the 16-bit counter increments up to FFFFH, overflows, and then increments up again from 0000H. When the counter value matches D2, the INTTPnCC0 signal is generated.

Therefore, the INTTPnCC0 signal may not be generated at the valid edge of the external event count signal when the external event count is “(D1 + 1)” or “(D2 + 1)” as originally expected, but instead may be generated at the valid edge of the external event count signal when the external event count is “(10000H + D2 + 1)”.

(c) Operation of TPnCCR1 register

The TPnCCR1 register is configured as follows in the external event count mode.

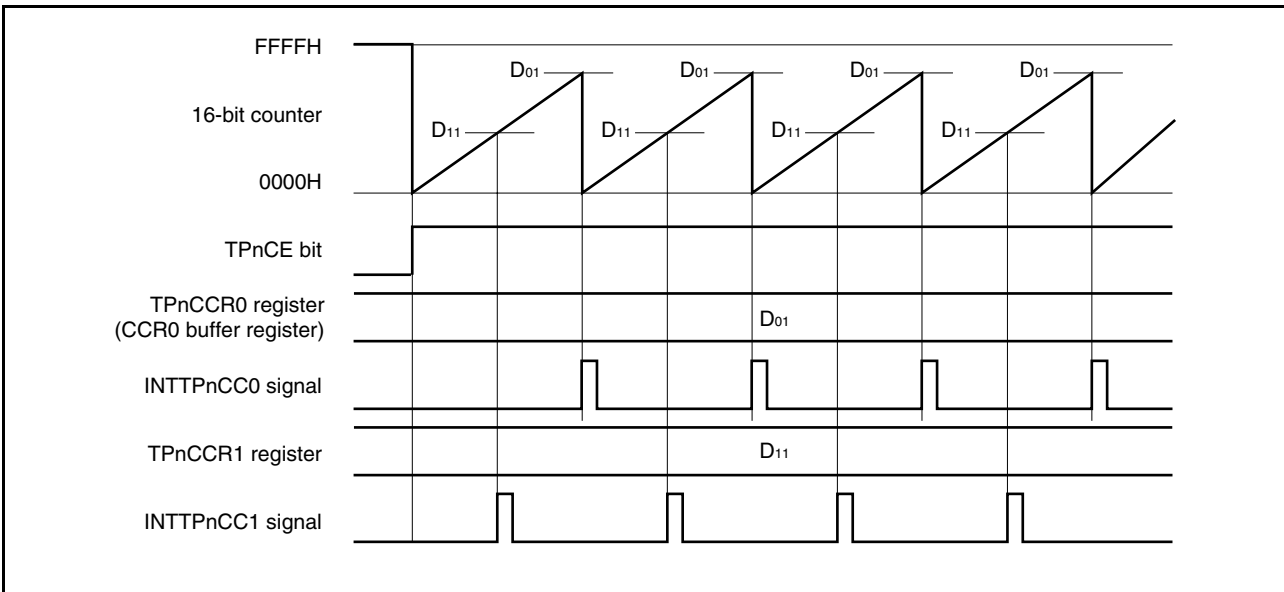
Figure 6-23. Configuration of TPnCCR1 Register



If the value of the TPnCCR1 register is less than or equal to the value of the TPnCCR0 register, the INTTPnCC1 signal is generated once per cycle.

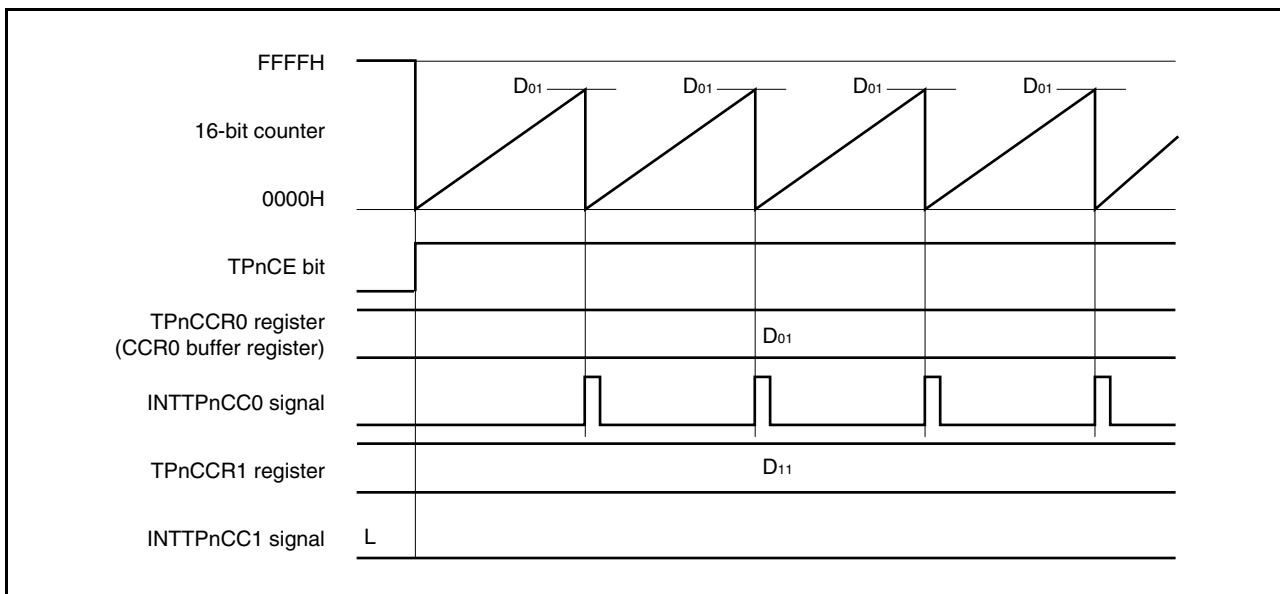
A chart showing the timing of operations when the value of the TPnCCR1 register (D_{11}) is less than or equal to the value of the TPnCCR0 register (D_{01}) is shown below.

Figure 6-24. Timing of Operations When $D_{01} \geq D_{11}$



If the value of the TPnCCR1 register is greater than the value of the TPnCCR0 register, the value of the 16-bit counter will not match the value of the TPnCCR1 register and the INTTPnCC1 signal will not be generated. A chart showing the timing of operations when the value of the TPnCCR1 register (D_{11}) is greater than the value of the TPnCCR0 register (D_{01}) is shown below.

Figure 6-25. Timing of Operations When $D_{01} < D_{11}$



6.4.3 External trigger pulse output mode (TPnMD2 to TPnMD0 bits = 010)

In the external trigger pulse output mode, when the TPnCTL0.TPnCE bit is set to 1, TMPn waits for a trigger, which is the valid edge of the external trigger input signal, and starts incrementing when this trigger is detected. TMPn then 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 instead of the external trigger, a square wave that has one cycle of the PWM waveform as half its cycle can also be output from the TOPn0 pin.

- Remarks 1.** For how to set the TIPn0, TOPn0, and TOPn1 pins, see **Table 6-2 Pins Used by TMPn** and **Table 4-17 Settings When Pins Are Used for Alternate Functions**.
- 2.** For how to enable the INTTPnCC0 and INTTPnCC1 interrupt signals, see **CHAPTER 19 INTERRUPT SERVICING/EXCEPTION PROCESSING FUNCTION**.

Figure 6-26. Configuration of TMP in External Trigger Pulse Output Mode

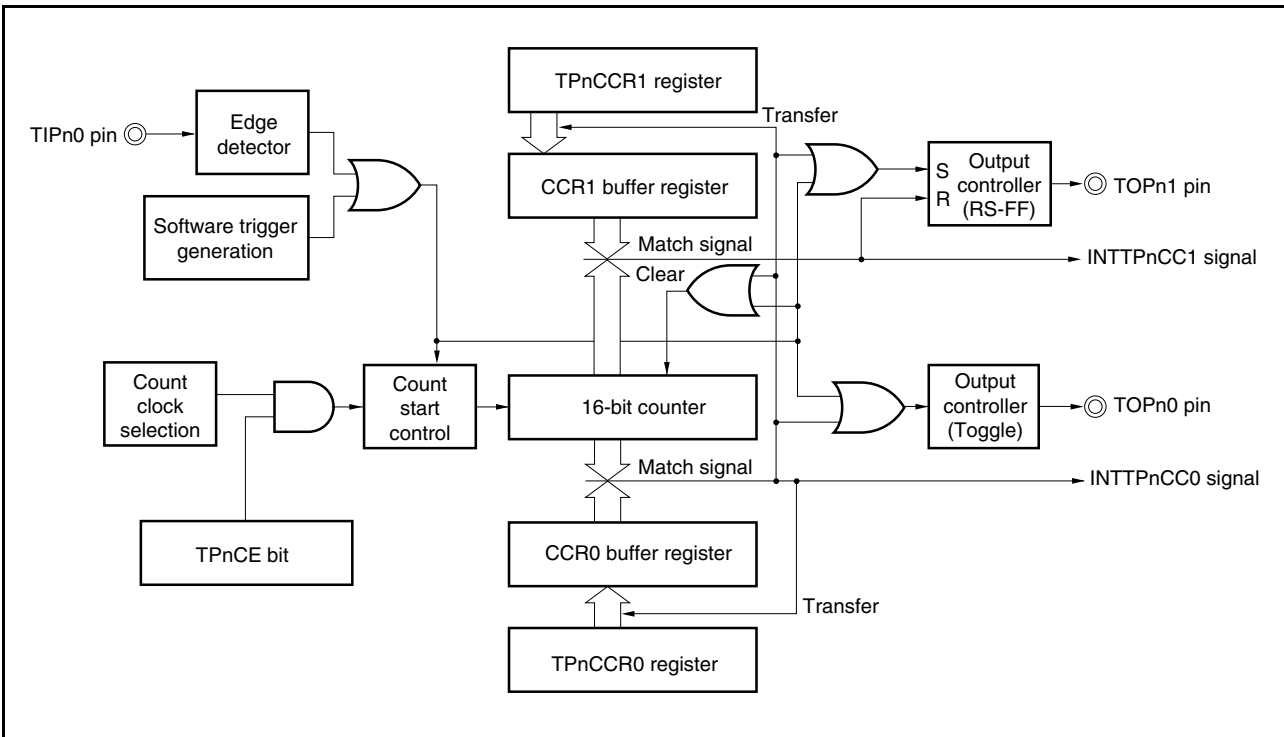
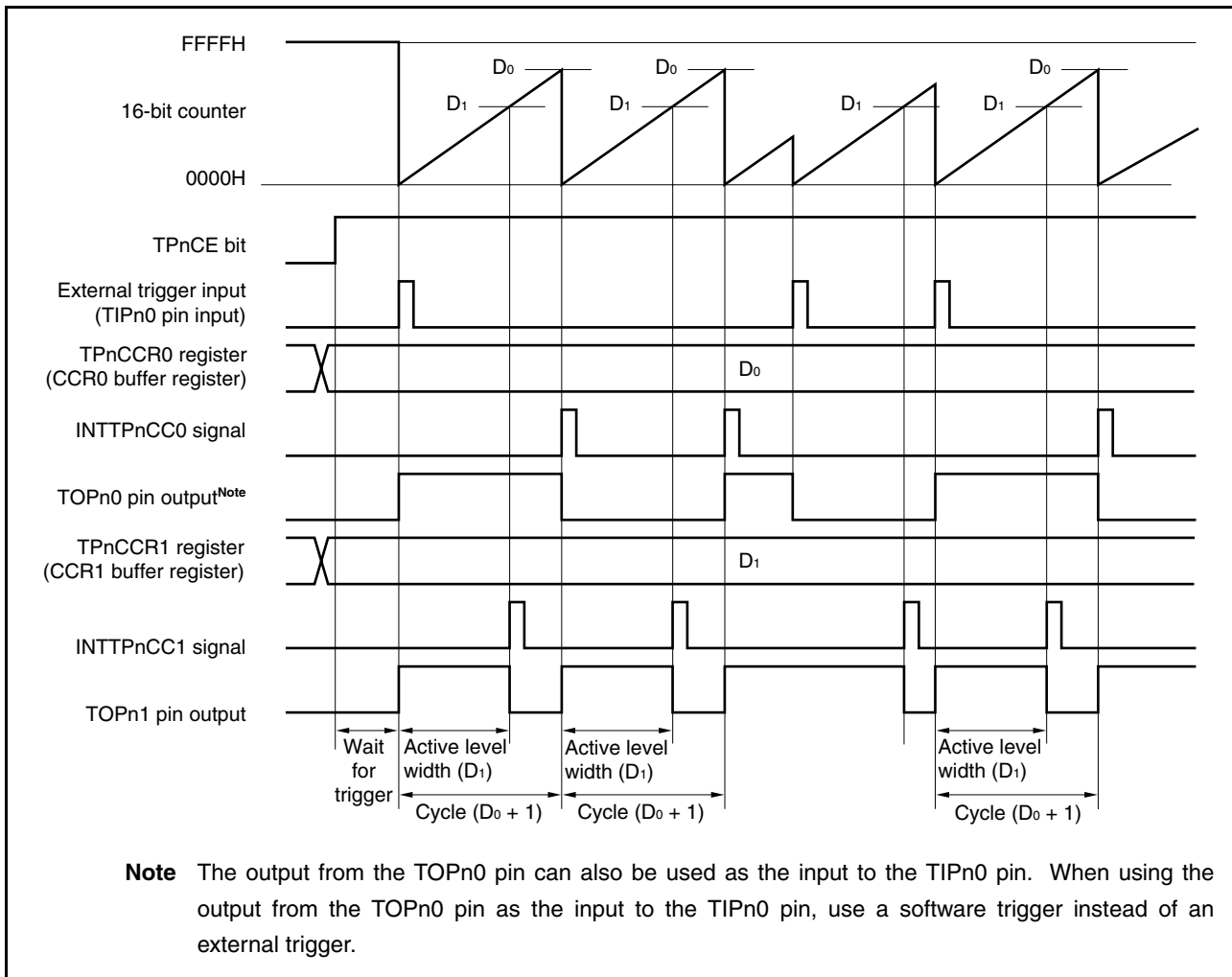


Figure 6-27. Basic Timing of Operations in External Trigger Pulse Output Mode



When the TPnCE bit is set to 1, TMPn waits for a trigger. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts incrementing, and outputs a PWM waveform from the TOPn1 pin. If the trigger is generated again while the counter is incrementing, the counter is cleared to 0000H and restarts incrementing, and the output of the TOPn0 pin is inverted. (The TOPn1 pin outputs a high level signal 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.

$$\text{Active level width} = (\text{Set value of TPnCCR1 register}) \times \text{Count clock cycle}$$

$$\text{Cycle} = (\text{Set value of TPnCCR0 register} + 1) \times \text{Count clock cycle}$$

$$\text{Duty factor} = (\text{Set value of TPnCCR1 register}) / (\text{Set value of TPnCCR0 register} + 1)$$

The INTTPnCC0 compare match interrupt request signal is generated when the 16-bit counter increments next time after its value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The INTTPnCC1 compare match interrupt request signal is generated when the value of the 16-bit counter matches the value of the CCR1 buffer register.

Either the valid edge of the external trigger input signal or setting the software trigger (TPnCTL1.TPnEST bit) to 1 is used as the trigger.

Figure 6-28. Register Settings in External Trigger Pulse Output Mode (1/2)

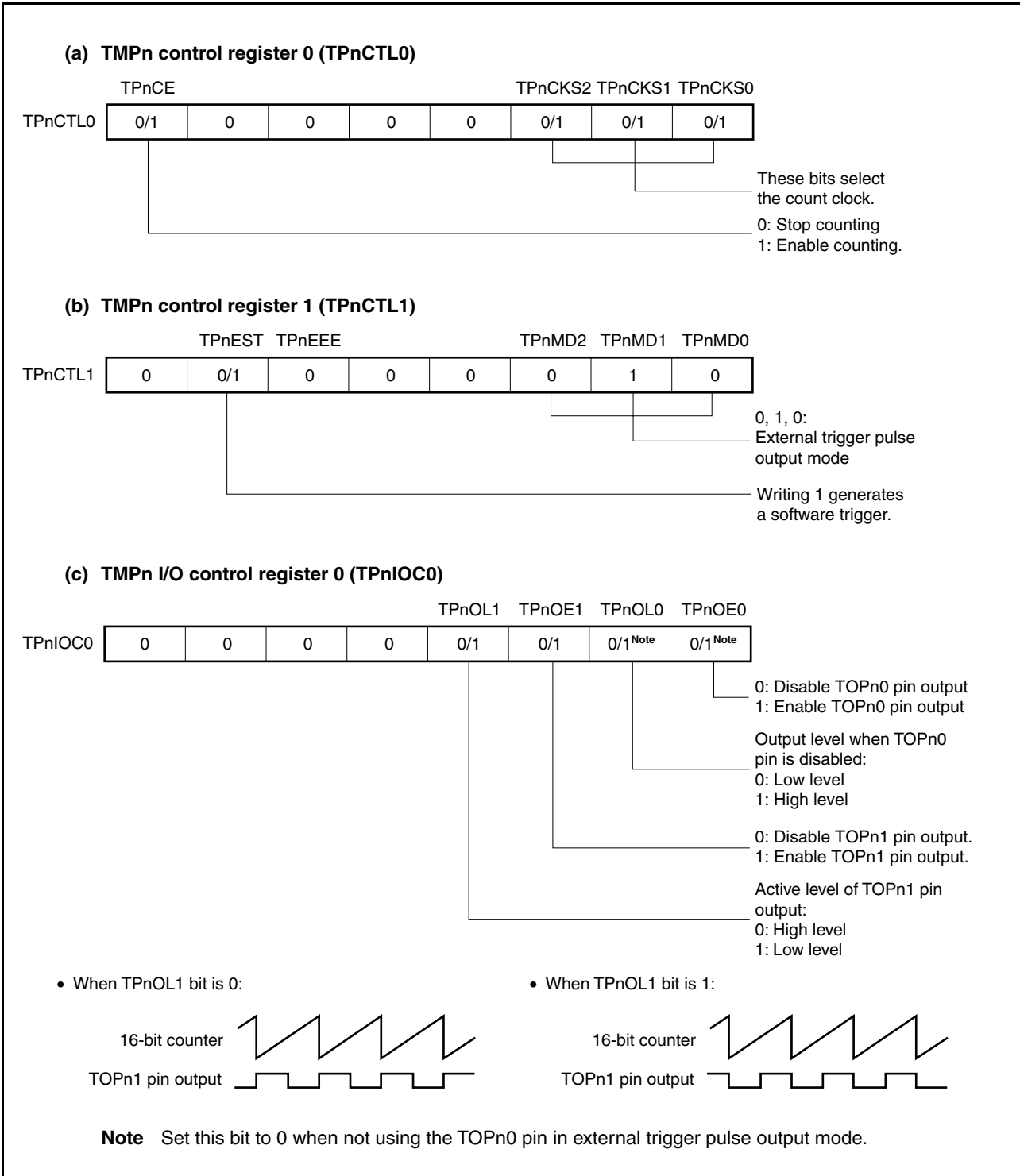
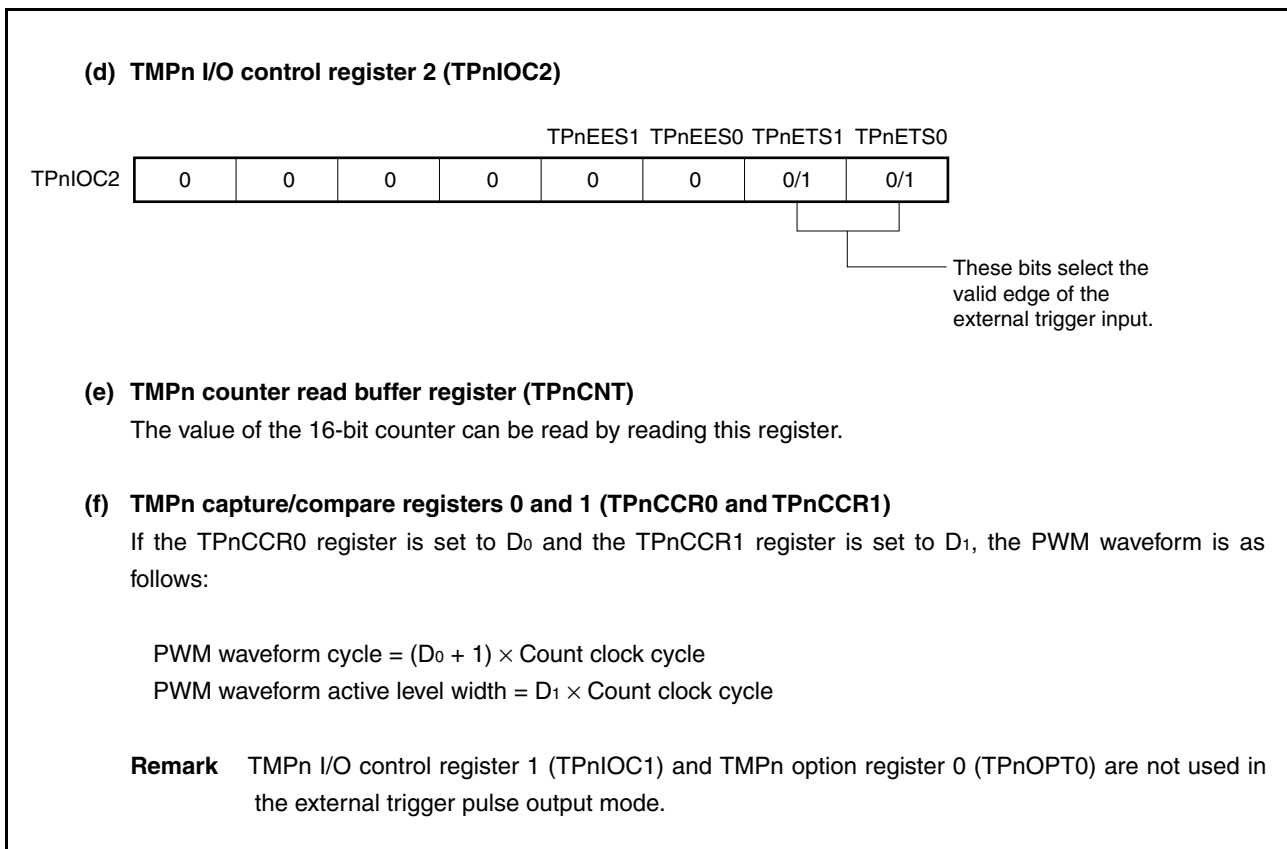


Figure 6-28. Register Settings in External Trigger Pulse Output Mode (2/2)



(1) Operations in external trigger pulse output mode

Figure 6-29. Timing and Processing of Operations in External Trigger Pulse Output Mode (1/2)

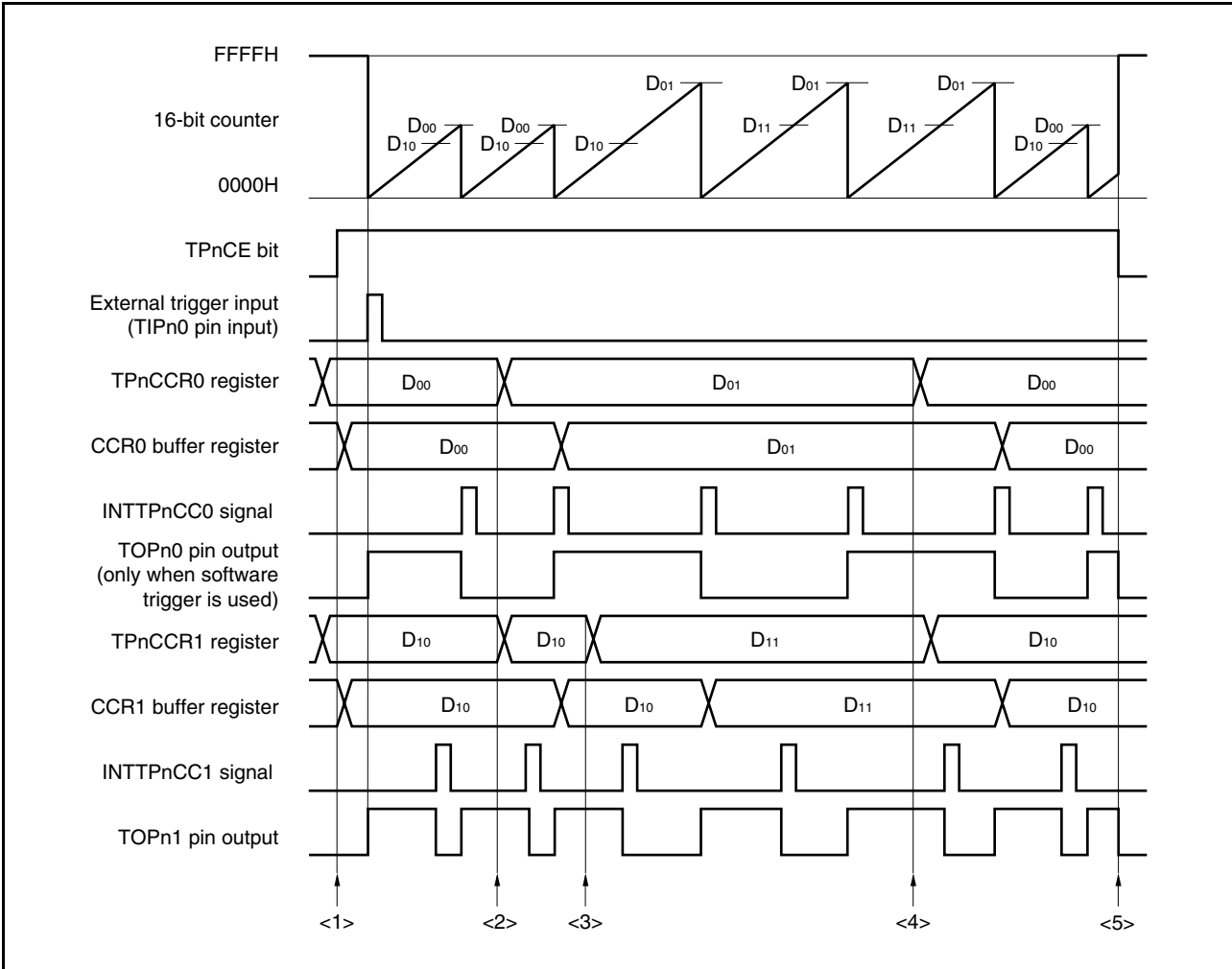
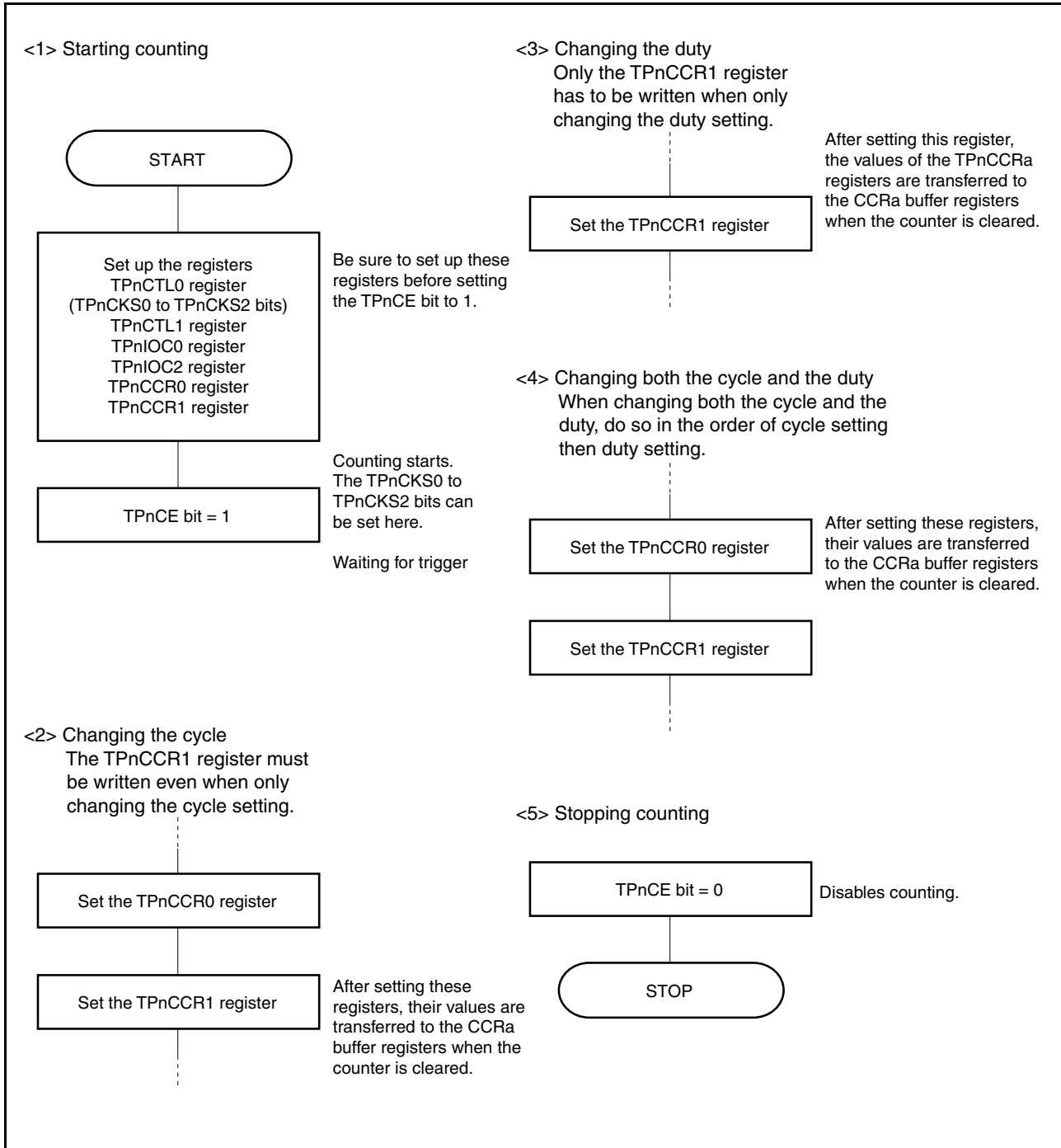


Figure 6-29. Timing and Processing of Operations in External Trigger Pulse Output Mode (2/2)



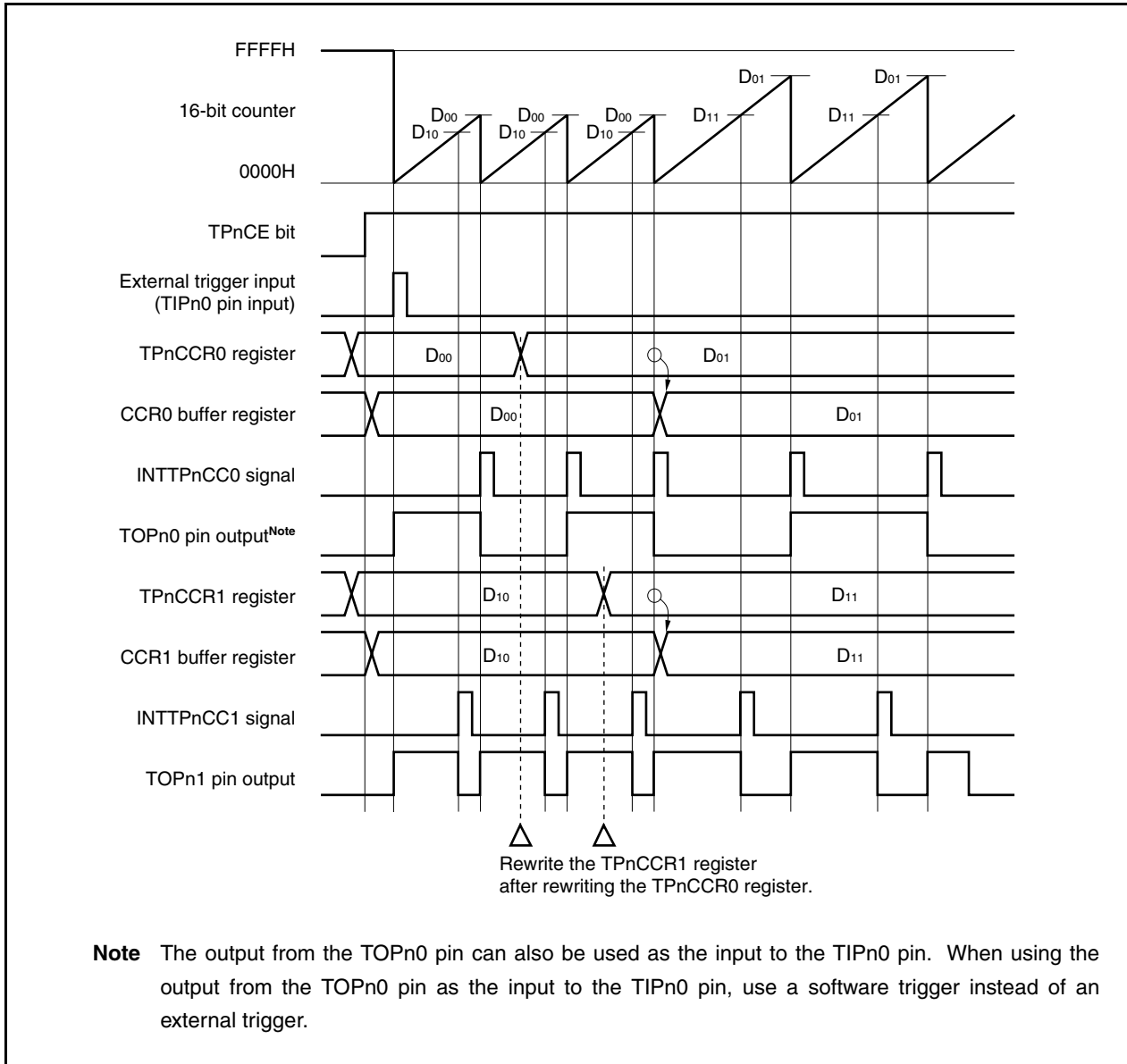
(2) Using external trigger pulse output mode

How to change the PWM waveform in the external trigger pulse output mode is described below.

(a) Changing the PWM waveform while the counter is incrementing

To change the PWM waveform while the counter is incrementing, write to the TPnCCR1 register after changing the waveform setting. When rewriting the TPnCCRa register after writing to the TPnCCR1 register, do so after the INTTPnCC0 signal has been detected.

Figure 6-30. Changing PWM Waveform While Counter Is Incrementing



In order to transfer data from the TPnCCRa register to the CCRa buffer register, the TPnCCR1 register must be written.

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 to be compared with the 16-bit counter value.

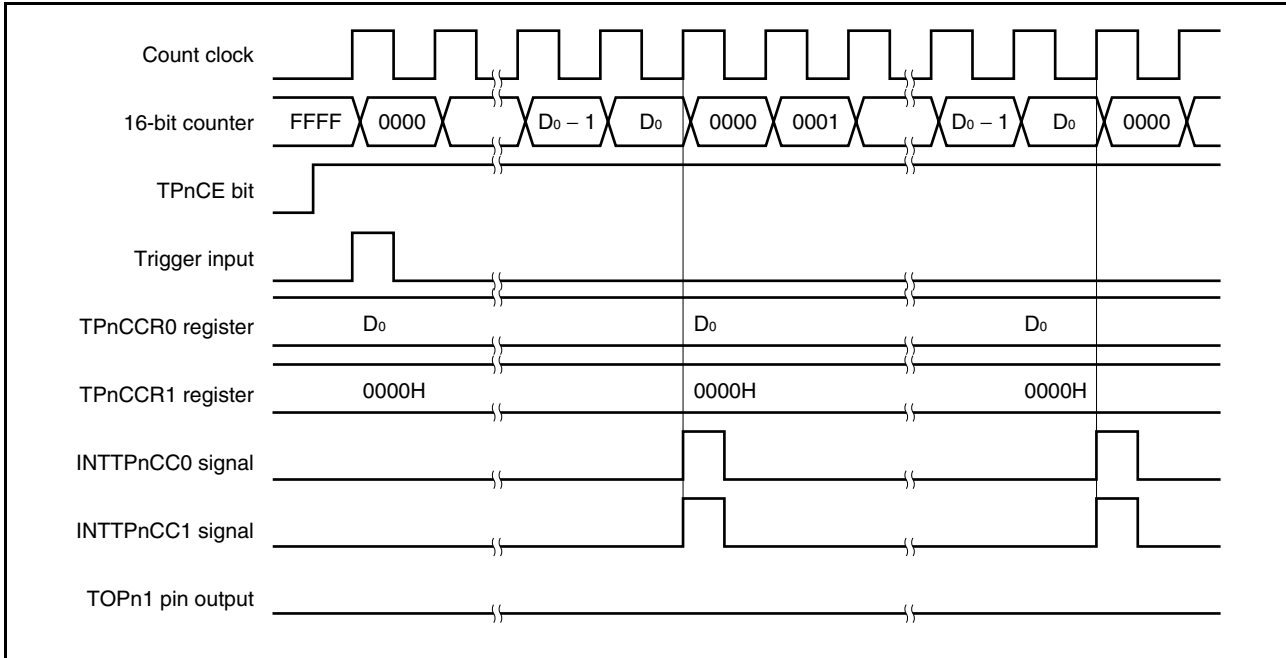
- <1> To change both the cycle and active level width of the PWM waveform, first set the cycle to the TPnCCR0 register and then set the active level width to the TPnCCR1 register.
- <2> To change only the cycle of the PWM waveform, first set the cycle to the TPnCCR0 register, and then write the same value to the TPnCCR1 register (that is, the same value as the value already specified for the TPnCCR1 register).
- <3> To change only the active level width (duty factor) of the PWM waveform, only the TPnCCR1 register has to be set.

Caution To rewrite the TPnCCR0 or TPnCCR1 register after writing the TPnCCR1 register, do so after the INTTPnCC0 signal has been 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.

(b) Outputting a 0% or 100% PWM waveform

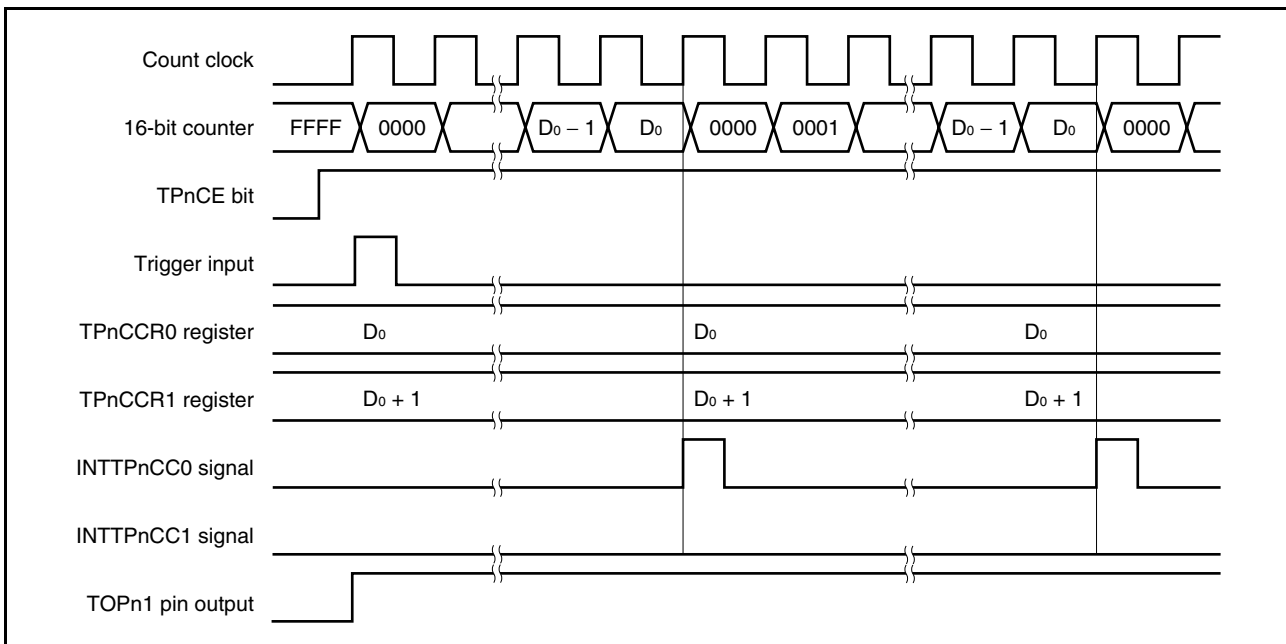
To output a 0% waveform, clear the TPnCCR1 register to 0000H. If the set value of the TPnCCR0 register is FFFFH, the INTTPnCC1 signal is generated periodically.

Figure 6-31. Outputting 0% PWM Waveform



To output a 100% waveform, set the value of TPnCCR0 register + 1 to the TPnCCR1 register. If the value of the TPnCCR0 register is FFFFH, a 100% waveform cannot be output.

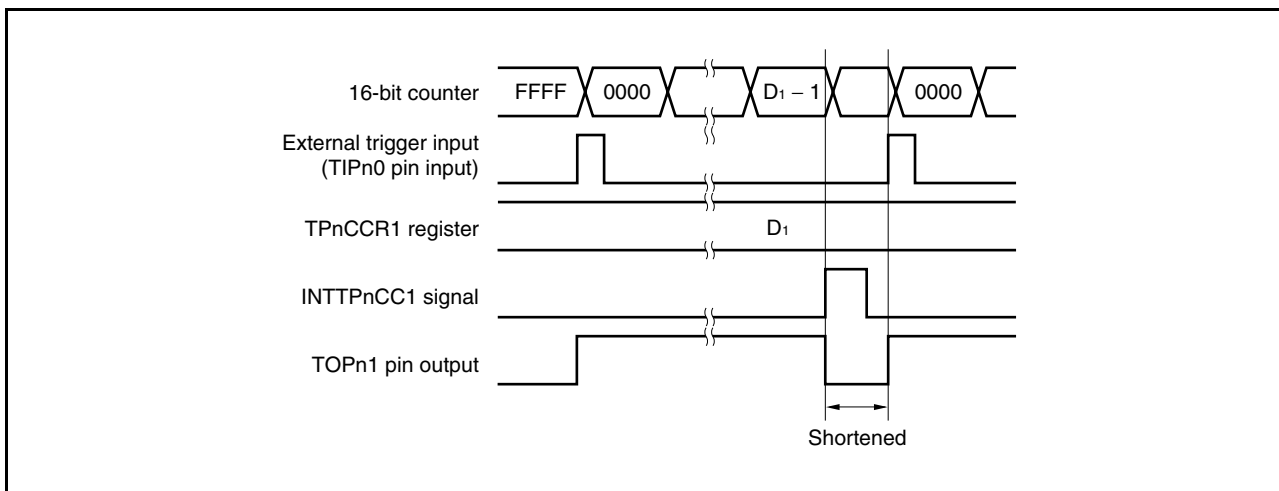
Figure 6-32. Outputting 100% PWM Waveform



(c) Detection of trigger immediately before or after INTTPnCC1 generation

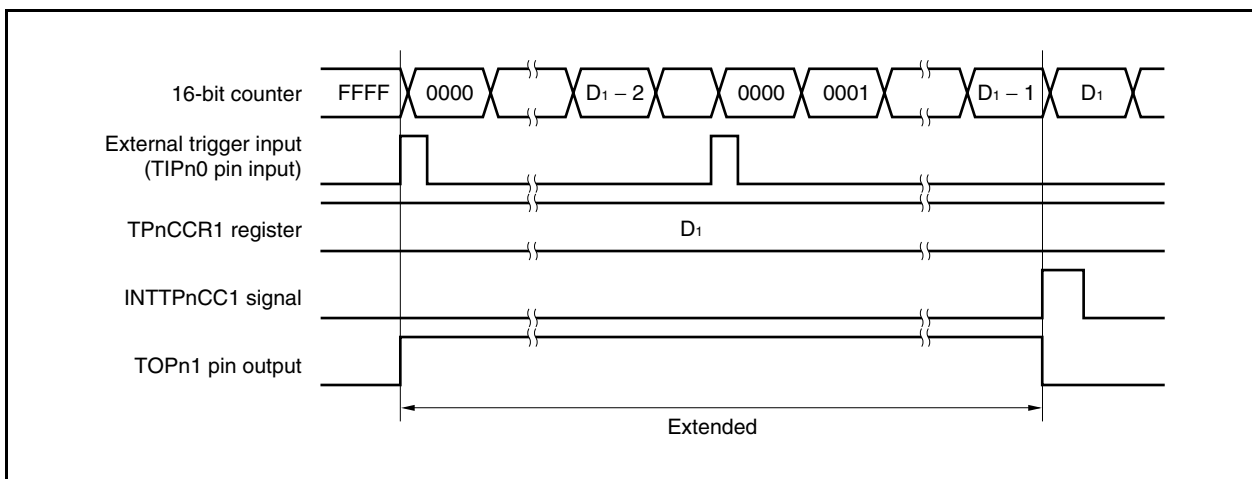
If the trigger is detected immediately after the INTTPnCC1 signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TOPn1 pin is set to the active level, and the counter continues incrementing. Consequently, the inactive period of the PWM waveform is shortened.

Figure 6-33. Detection of Trigger Immediately After INTTPnCC1 Signal Was Generated



If the trigger is detected immediately before the INTTPnCC1 signal is generated, the INTTPnCC1 signal is not generated, and the 16-bit counter is cleared to 0000H and continues incrementing. The output signal of the TOPn1 pin remains active. Consequently, the active period of the PWM waveform is extended.

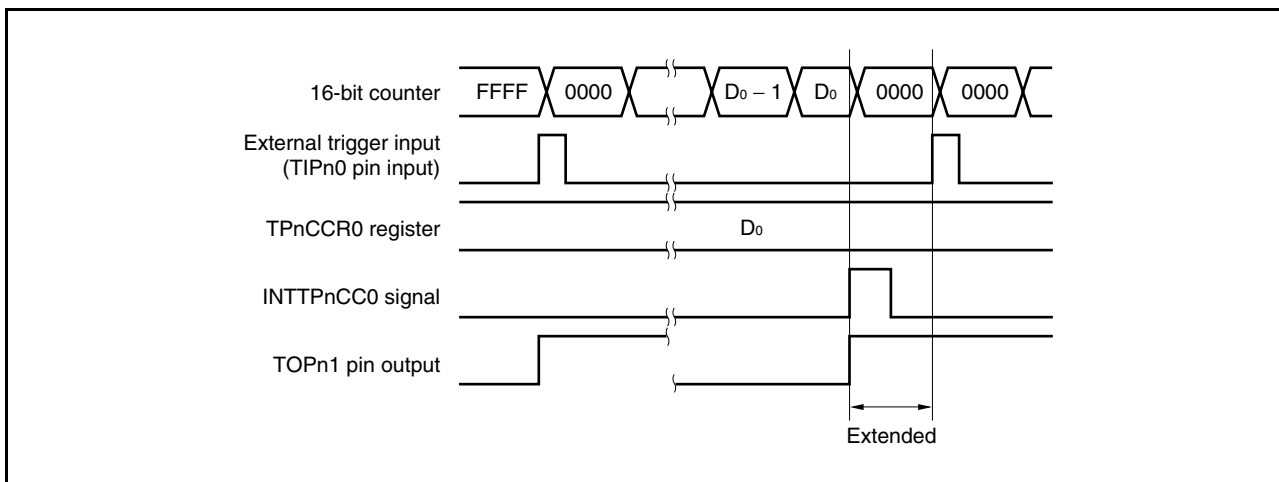
Figure 6-34. Detection of Trigger Immediately Before INTTPnCC1 Signal Is Generated



(d) Detection of trigger immediately before or after INTTPnCC0 generation

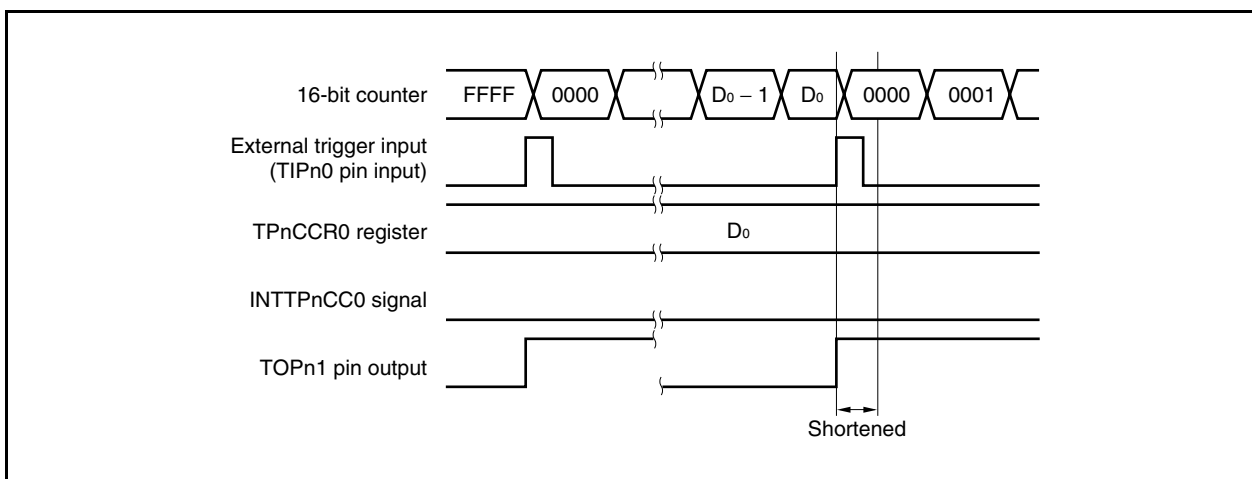
If the trigger is detected immediately after the INTTPnCC0 signal is generated, the 16-bit counter is cleared to 0000H and continues incrementing. Therefore, the active period of the TOPn1 pin is extended by the amount of time between the generation of the INTTPnCC0 signal and the detection of the trigger.

Figure 6-35. Detection of Trigger Immediately After INTTPnCC0 Signal Was Generated



If the trigger is detected immediately before the INTTPnCC0 signal is generated, the INTTPnCC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOPn1 pin is set to the active level, and the counter continues incrementing. Consequently, the inactive period of the PWM waveform is shortened.

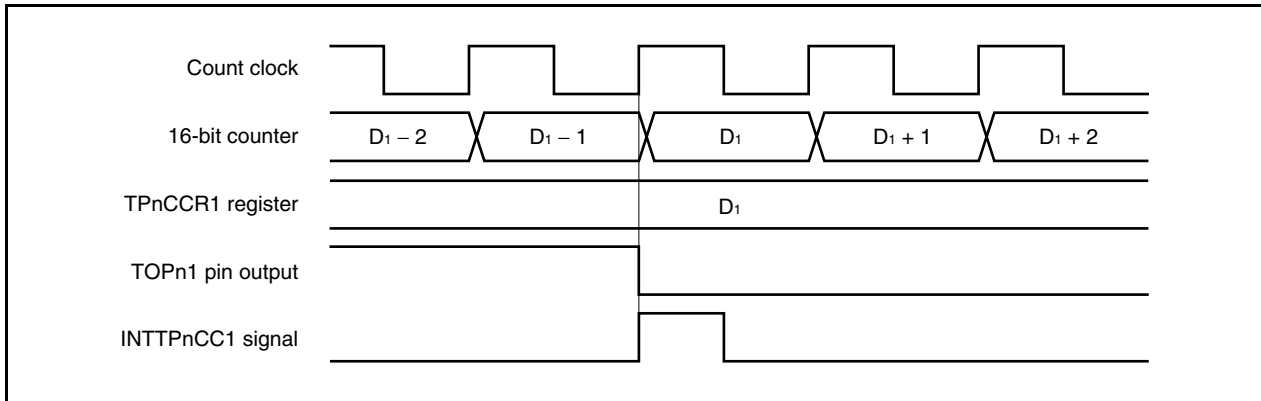
Figure 6-36. Detection of Trigger Immediately Before INTTPnCC0 Signal Is Generated



(e) Timing of generating the compare match interrupt request signal (INTTPnCC1)

In the external trigger pulse output mode, the INTTPnCC1 signal is generated when the value of the 16-bit counter matches the value of the TPnCCR1 register.

Figure 6-37. Timing of Generating Compare Match Interrupt Signal (INTTPnCC1)



6.4.4 One-shot pulse output mode (TPnMD2 to TPnMD0 bits = 011)

In the one-shot pulse output mode, when the TPnCTL0.TPnCE bit is set to 1, TMPn waits for a trigger, which is the valid edge of the external trigger input, and starts incrementing when this trigger is detected. TMPn then 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 signal while the 16-bit counter is incrementing, and the inactive level signal when the counter is stopped (waiting for a trigger).

- Remarks 1.** For how to set the TIPn0, TOPn0, and TOPn1 pins, see **Table 6-2 Pins Used by TMPn** and **Table 4-17 Settings When Pins Are Used for Alternate Functions**.
- 2.** For how to enable the INTTPnCC0 and INTTPnCC1 interrupt signals, see **CHAPTER 19 INTERRUPT SERVICING/EXCEPTION PROCESSING FUNCTION**.

Figure 6-38. Configuration of TMPn in One-Shot Pulse Output Mode

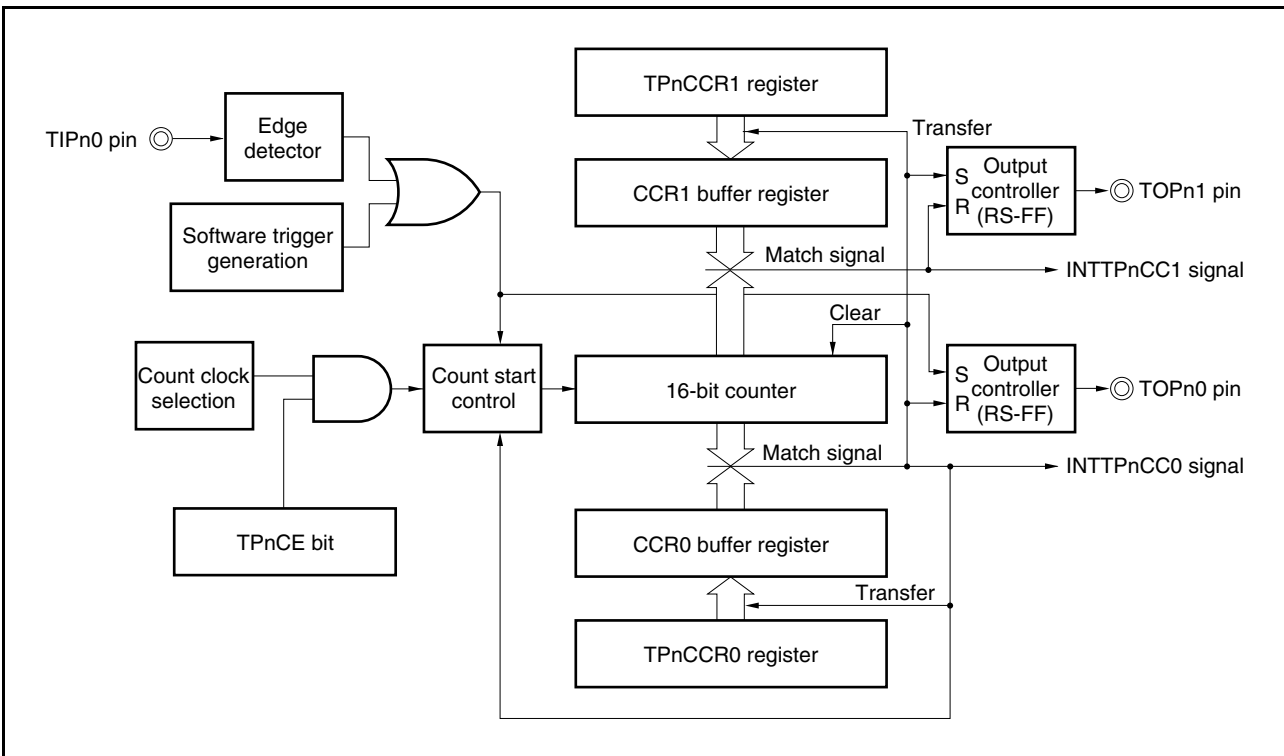
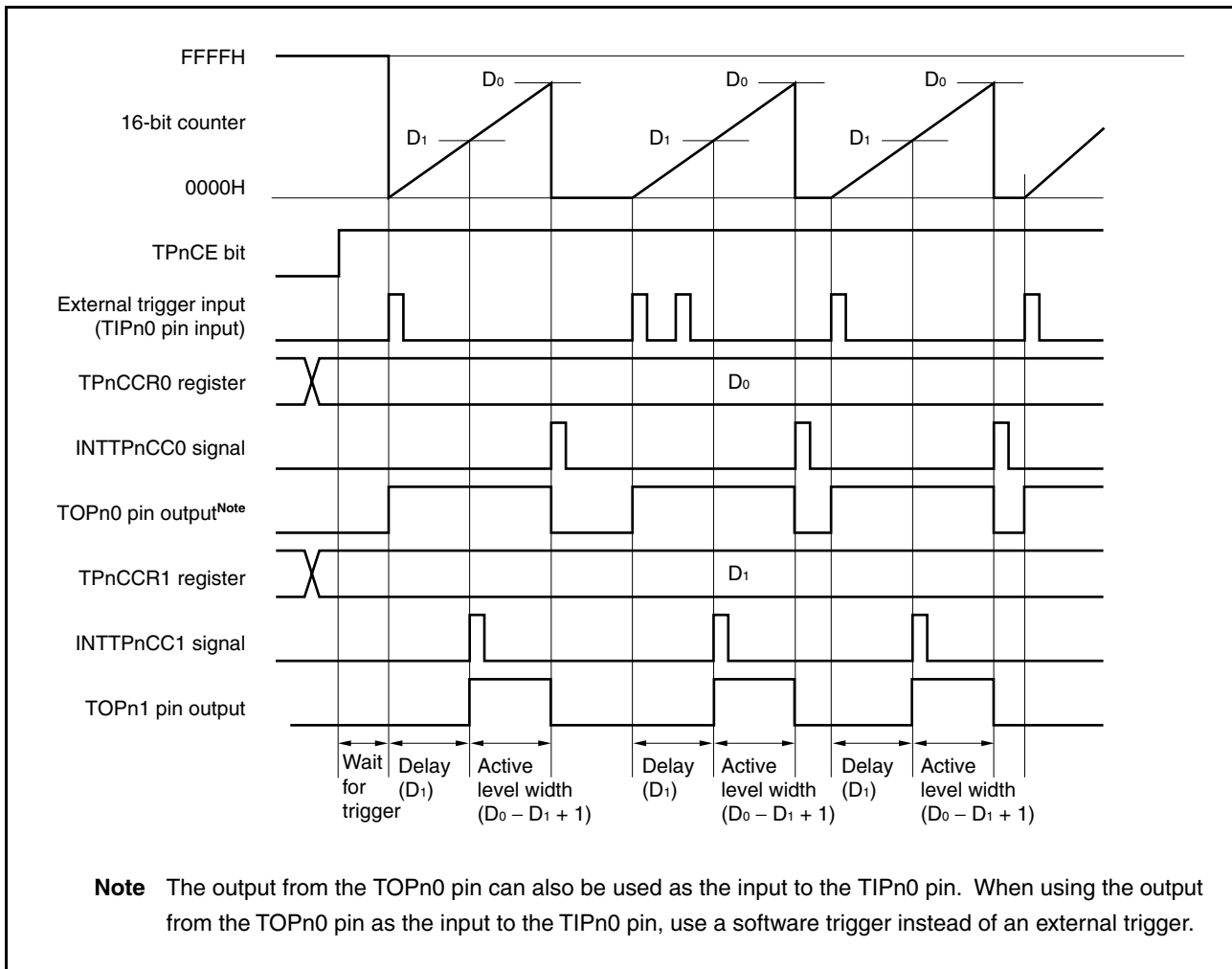


Figure 6-39. Basic Timing of Operations in One-Shot Pulse Output Mode



When the TPnCE bit is set to 1, TMPn waits for a trigger. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts incrementing, and outputs a one-shot pulse from the TOPn1 pin. After the one-shot pulse is output, the 16-bit counter is set to 0000H, stops incrementing, and waits for a trigger. 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:

$$\text{Output delay period} = (\text{Set value of TPnCCR1 register}) \times \text{Count clock cycle}$$

$$\text{Active level width} = (\text{Set value of TPnCCR0 register} - \text{Set value of TPnCCR1 register} + 1) \times \text{Count clock cycle}$$

The INTTPnCC0 compare match interrupt request signal is generated when the 16-bit counter increments next time after its value matches the value of the CCR0 buffer register. The INTTPnCC1 compare match interrupt request signal is generated when the value of the 16-bit counter matches the value of the CCR1 buffer register.

Either the valid edge of the external trigger input signal or setting the software trigger (TPnCTL1.TPnEST bit) to 1 is used as the trigger.

Figure 6-40. Register Settings in One-Shot Pulse Output Mode (1/2)

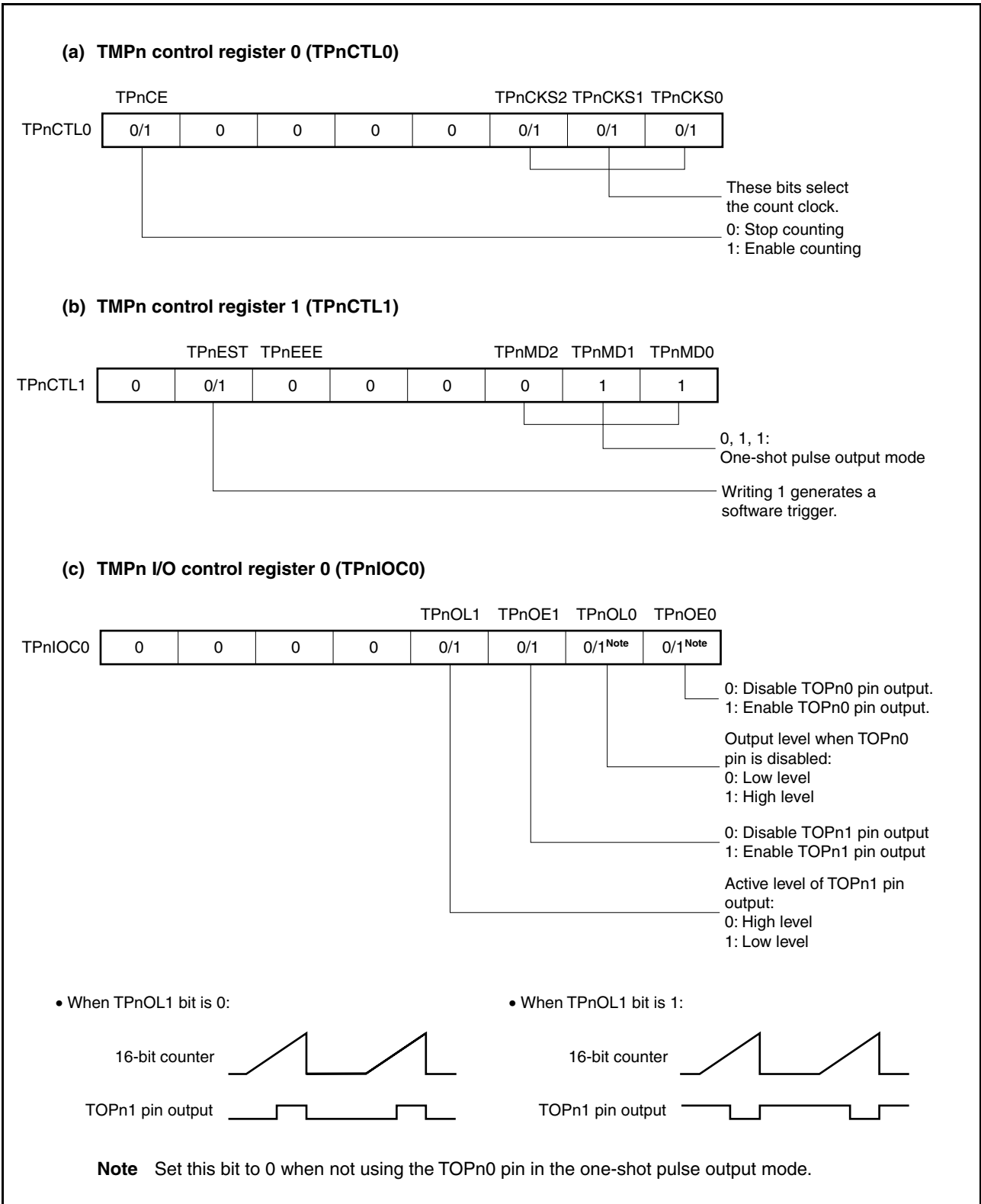
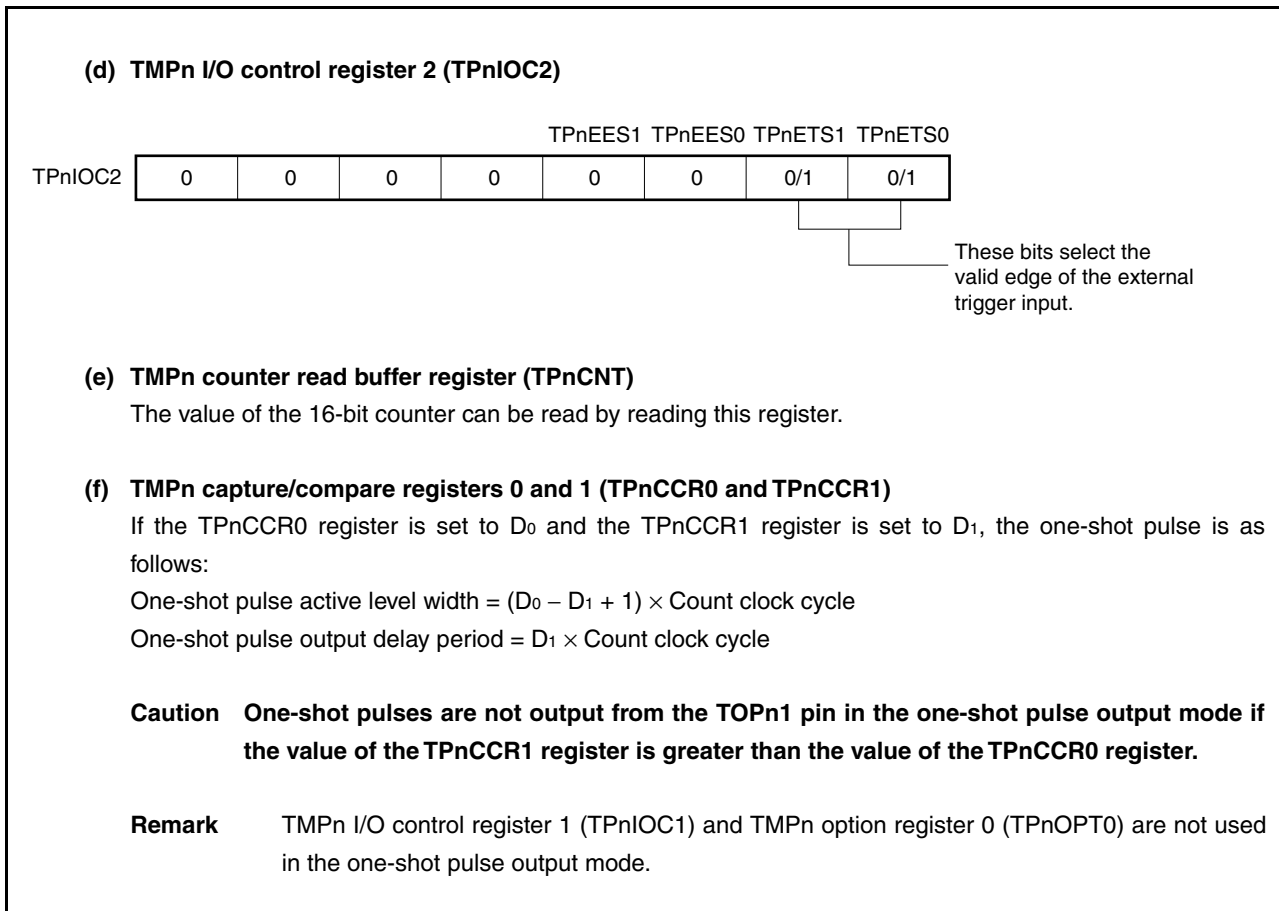
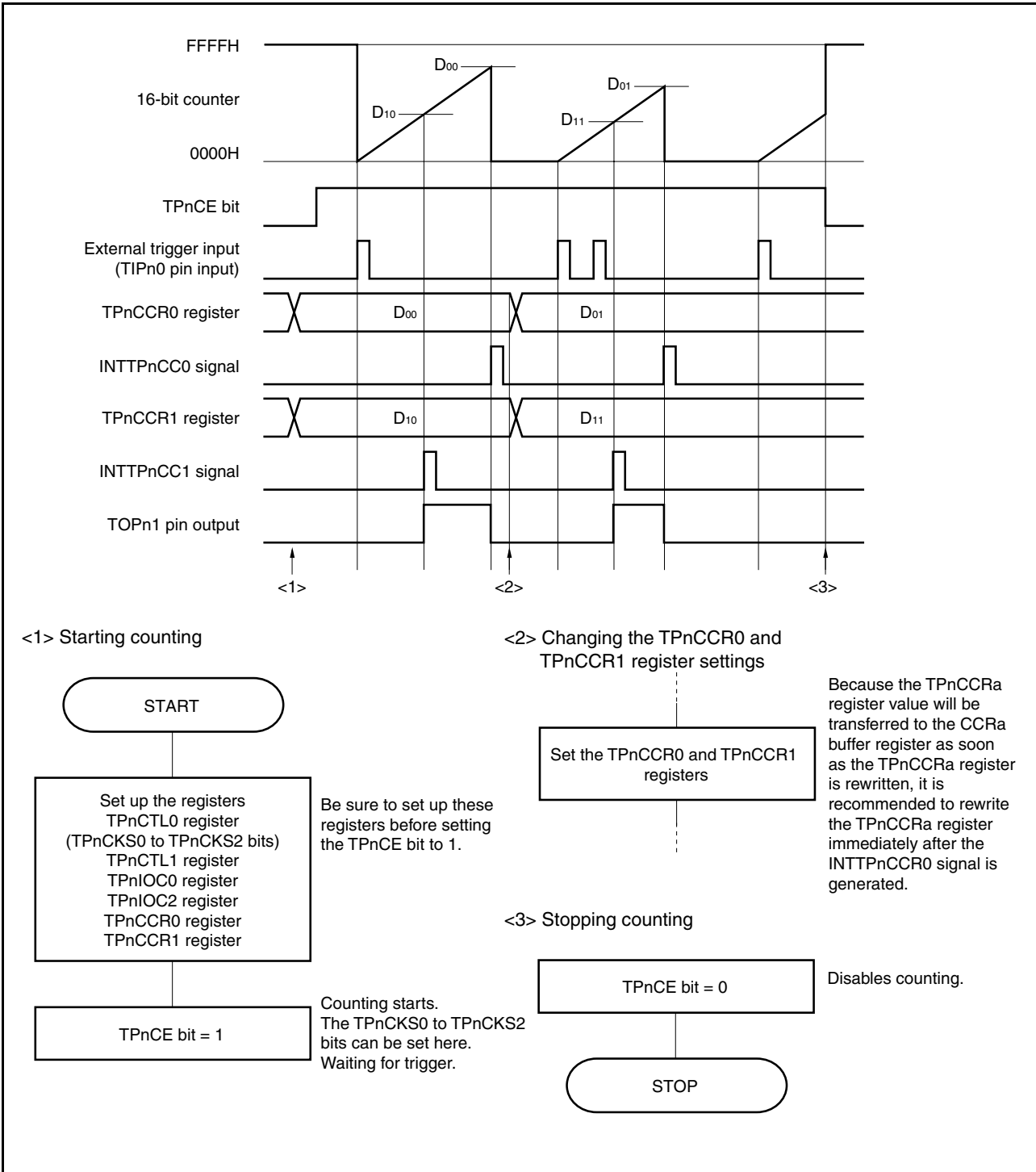


Figure 6-40. Register Settings in One-Shot Pulse Output Mode (2/2)



(1) Operations in one-shot pulse output mode

Figure 6-41. Timing and Processing of Operations in One-Shot Pulse Output Mode



(2) Using one-shot pulse mode

(a) Rewriting the TPnCCRa register

When rewriting the value of the TPnCCRa register to a smaller value, stop counting first and then change the set value.

When changing the value of the TPnCCR0 register from D_{00} to D_{01} and the value of the TPnCCR1 register from D_{10} to D_{11} , if the registers are rewritten under any of the following conditions, a one-shot pulse will not be output as expected.

Condition 1 When rewriting the TPnCCR0 register, if:

$D_{00} > D_{01}$ or,

$D_{00} < 16\text{-bit counter value} < D_{01}$

In the case of condition 1, the 16-bit counter will not be cleared and will overflow in the cycle in which the new value is being written. The counter will be cleared for the first time at the newly written value (D_{01}).

Condition 2 When rewriting the TPnCCR1 register, if:

$D_{10} > D_{11}$ or,

$D_{10} < 16\text{-bit counter value} < D_{11}$

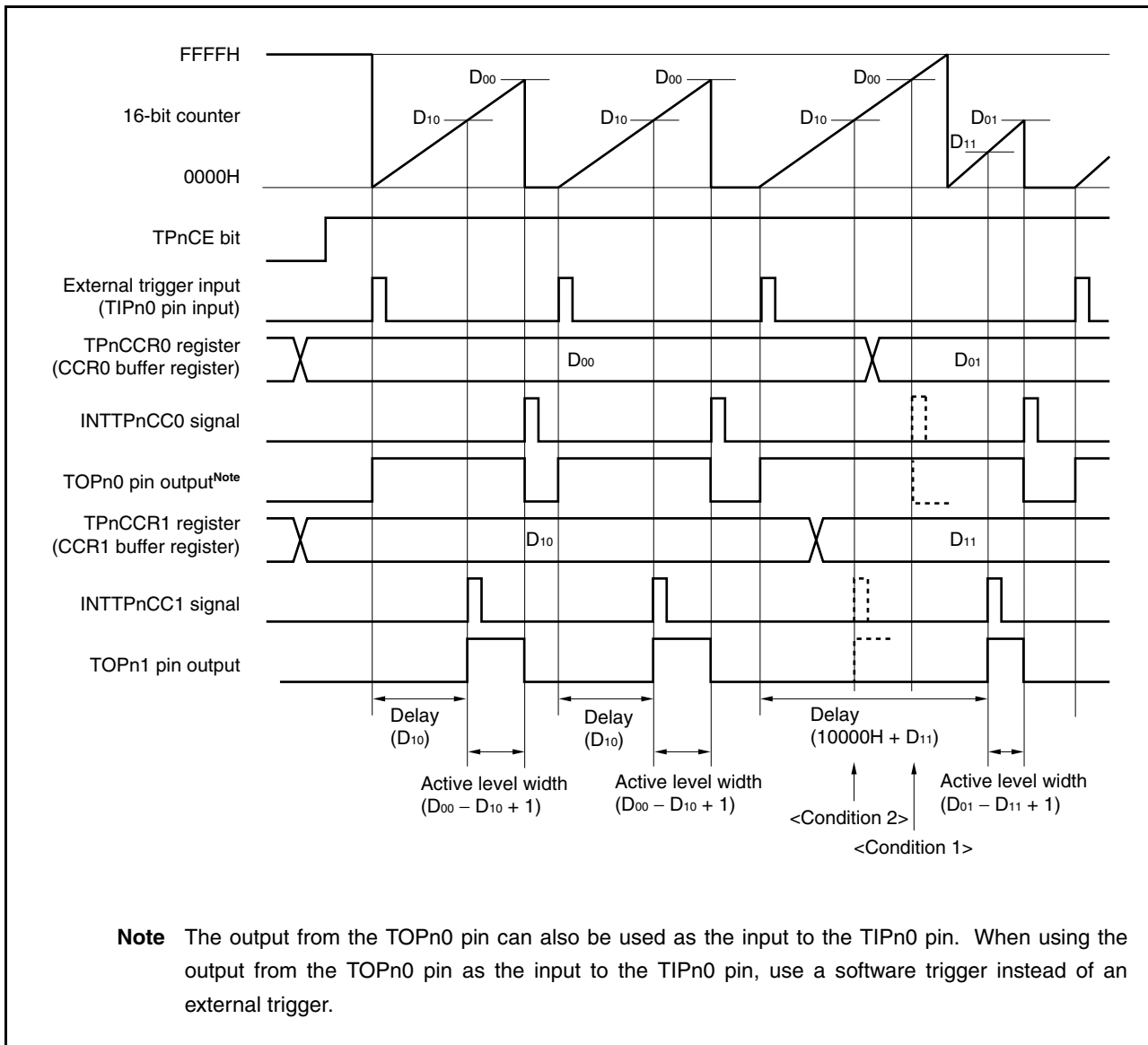
In the case of condition 2, the TOPn1 pin output cannot be inverted to the active level in the cycle in which the new value is being written.

An example of what happens when condition 1 and condition 2 are satisfied in the same cycle is shown in Figure 6-42.

The 16-bit counter increments up to FFFFH, overflows, and starts incrementing again from 0000H.

When the 16-bit counter value matches D_{11} , the INTTPnCC1 signal is generated and the TOPn1 pin output is set to the active level. Subsequently, when the 16-bit counter value matches D_{01} , the INTTPnCC0 signal is generated, the TOPn1 pin output is set to the inactive level, and the counter stops incrementing.

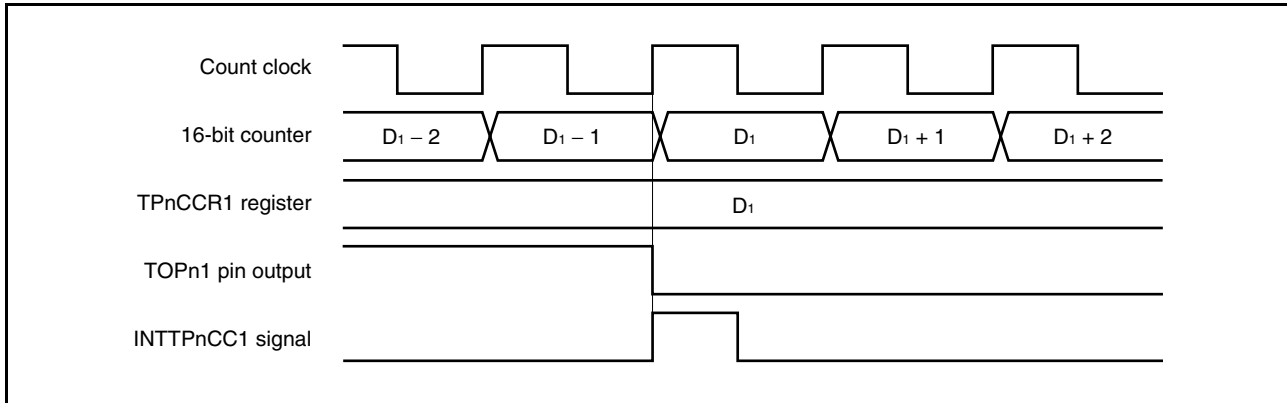
Figure 6-42. Rewriting TPnCCRa Register



(b) Timing of generating the compare match interrupt request signal (INTTPnCC1)

In the one-shot pulse output mode, the INTTPnCC1 signal is generated when the value of the 16-bit counter matches the value of the TPnCCR1 register.

Figure 6-43. Timing of Generating Compare Match Interrupt Signal (INTTPnCC1)



6.4.5 PWM output mode (TPnMD2 to TPnMD0 bits = 100)

In the PWM output mode, when the TPnCTL0.TPnCE bit is set to 1, TMPn outputs a PWM waveform from the TOPn1 pin.

A pulse that has one cycle of the PWM waveform as half its cycle can also be output from the TOPn0 pin.

- Remarks 1.** For how to set the TIPn0, TOPn0, and TOPn1 pins, see **Table 6-2 Pins Used by TMPn** and **Table 4-17 Settings When Pins Are Used for Alternate Functions**.
- 2.** For how to enable the INTTPnCC0 and INTTPnCC1 interrupt signals, see **CHAPTER 19 INTERRUPT SERVICING/EXCEPTION PROCESSING FUNCTION**.

Figure 6-44. Configuration of TMPn in PWM Output Mode

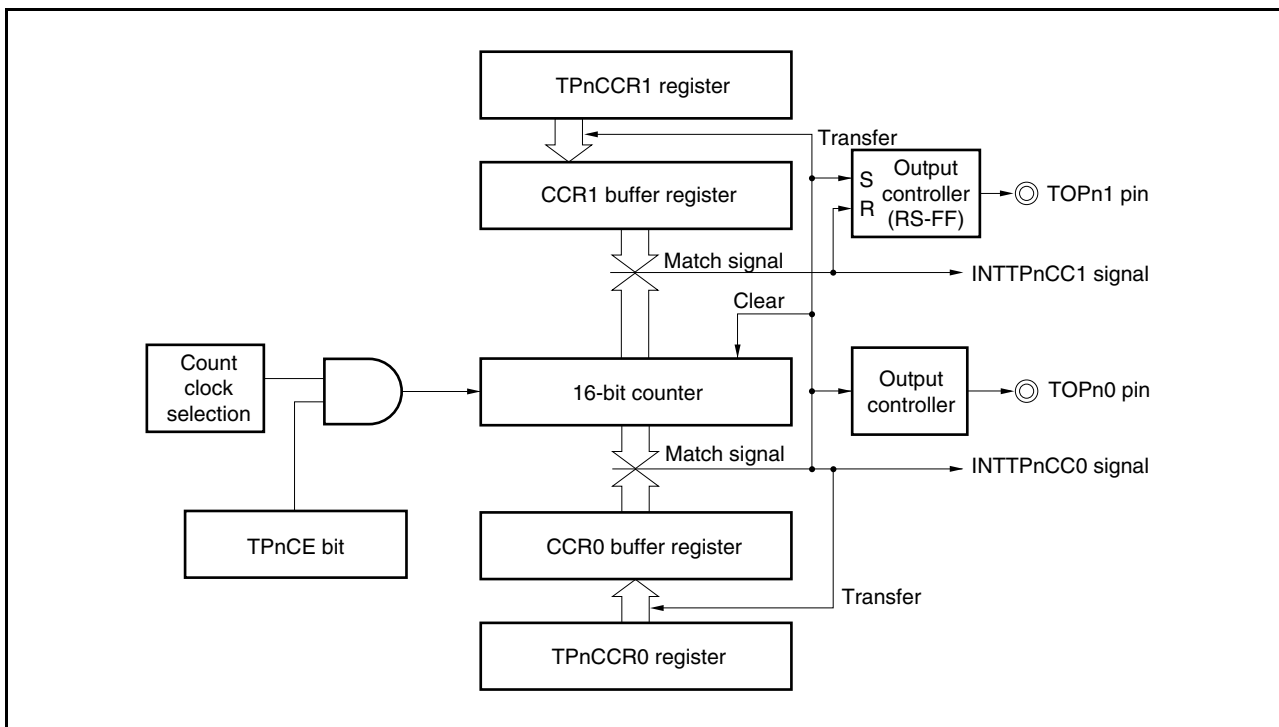
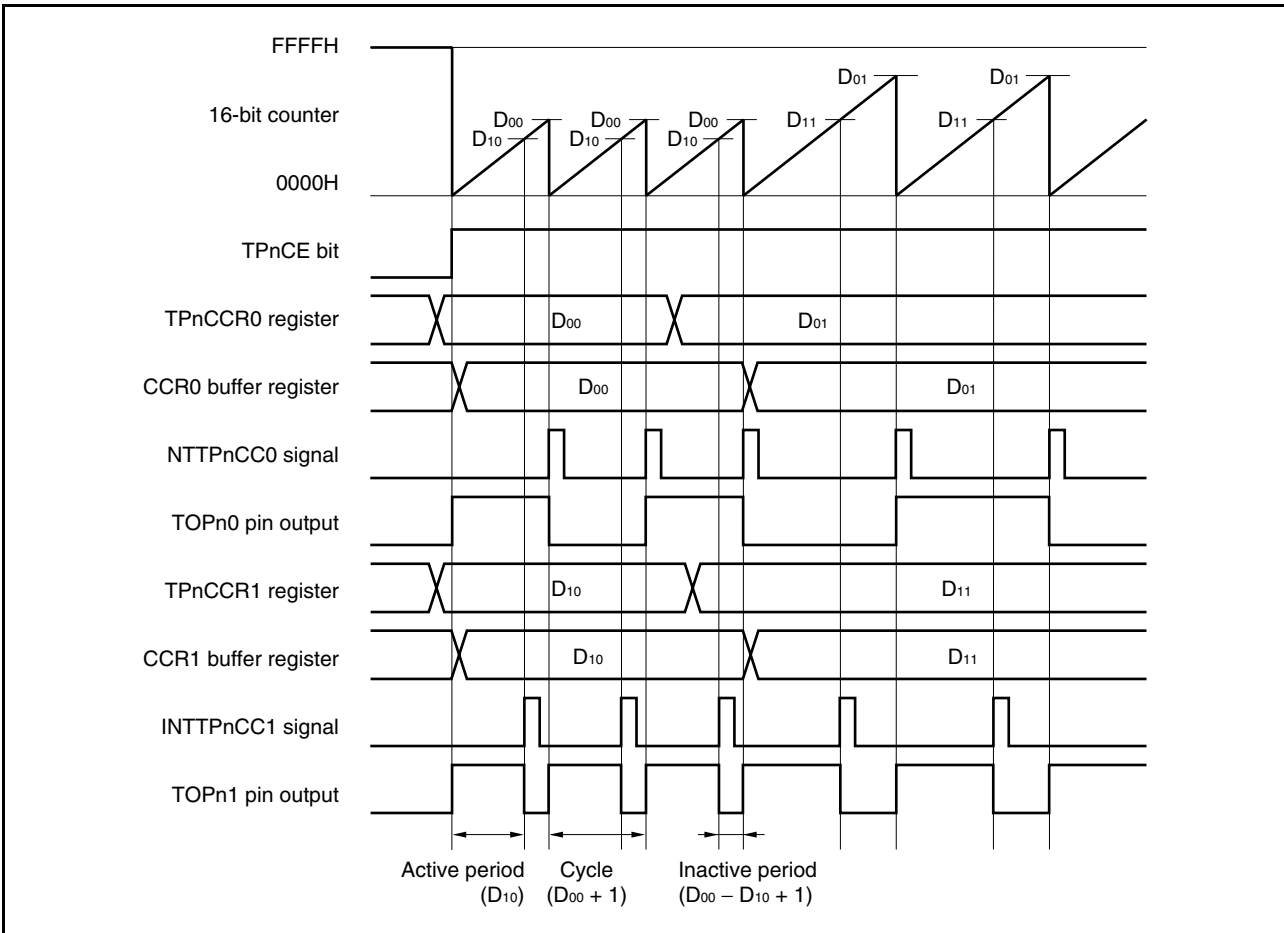


Figure 6-45. Basic Timing of Operations in PWM Output Mode



When the TPnCE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts incrementing, 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:

$$\text{Active level width} = (\text{Set value of TPnCCR1 register}) \times \text{Count clock cycle}$$

$$\text{Cycle} = (\text{Set value of TPnCCR0 register} + 1) \times \text{Count clock cycle}$$

$$\text{Duty factor} = (\text{Set value of TPnCCR1 register}) / (\text{Set value of TPnCCR0 register} + 1)$$

The PWM waveform can be changed by rewriting the TPnCCRa register while the counter is incrementing. The newly written value is reflected when the value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

The INTTPnCC0 compare match interrupt request signal is generated when the 16-bit counter increments next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The INTTPnCC1 compare match interrupt request signal is generated when the value of the 16-bit counter matches the value of the CCR1 buffer register.

Figure 6-46. Register Settings in PWM Output Mode (1/2)

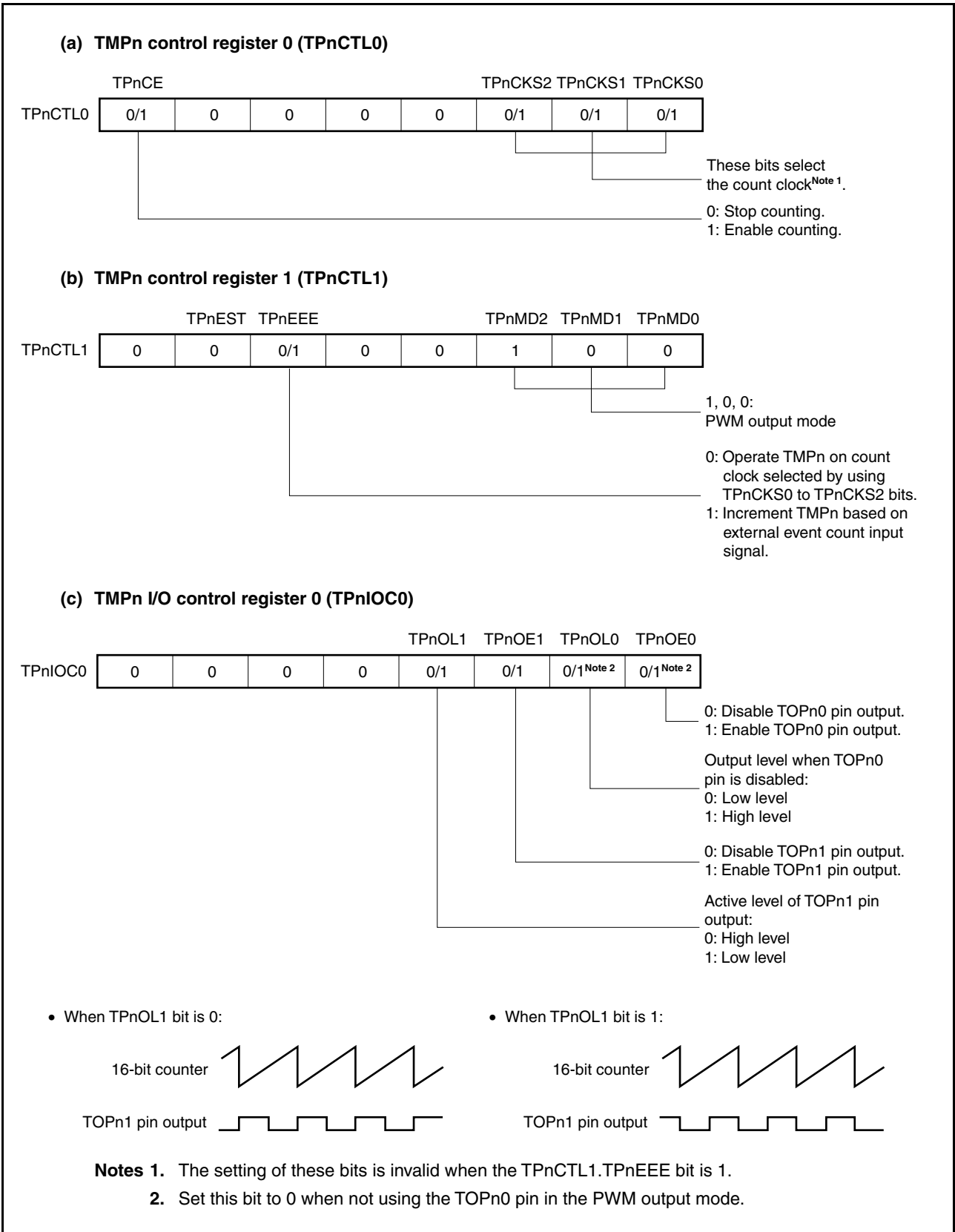
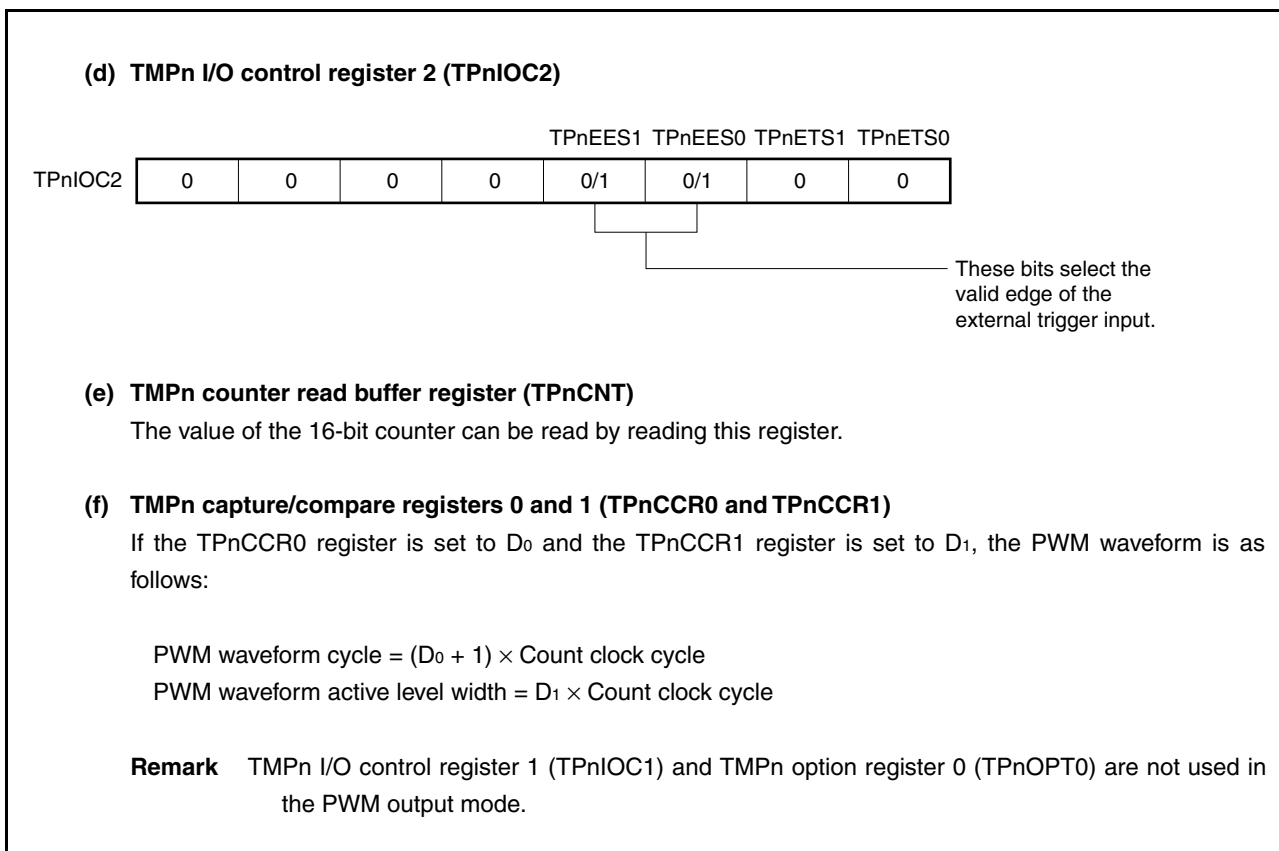


Figure 6-46. Register Settings in PWM Output Mode (2/2)



(1) Operations in PWM output mode

Figure 6-47. Timing and Processing of Operations in PWM Output Mode (1/2)

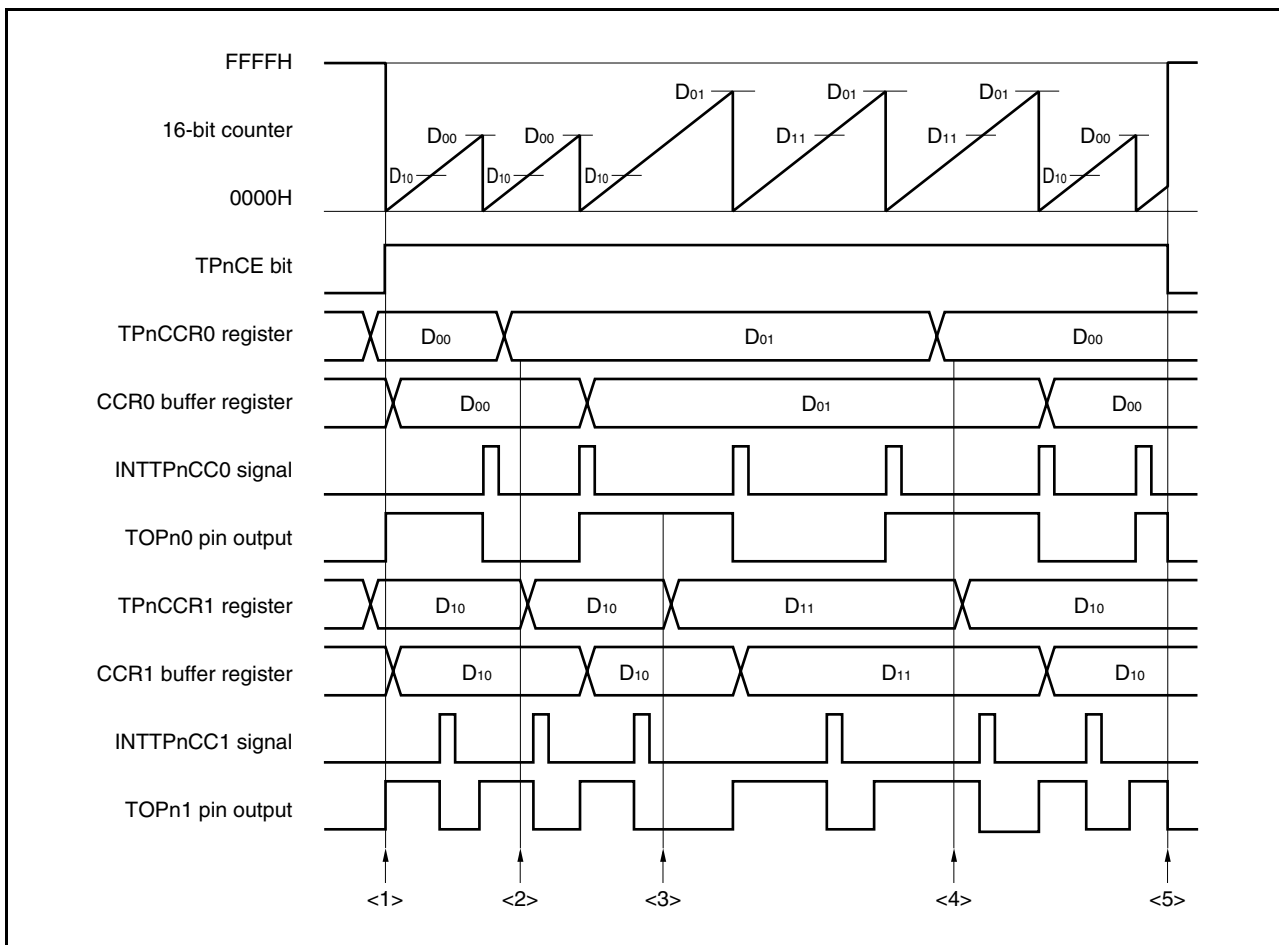
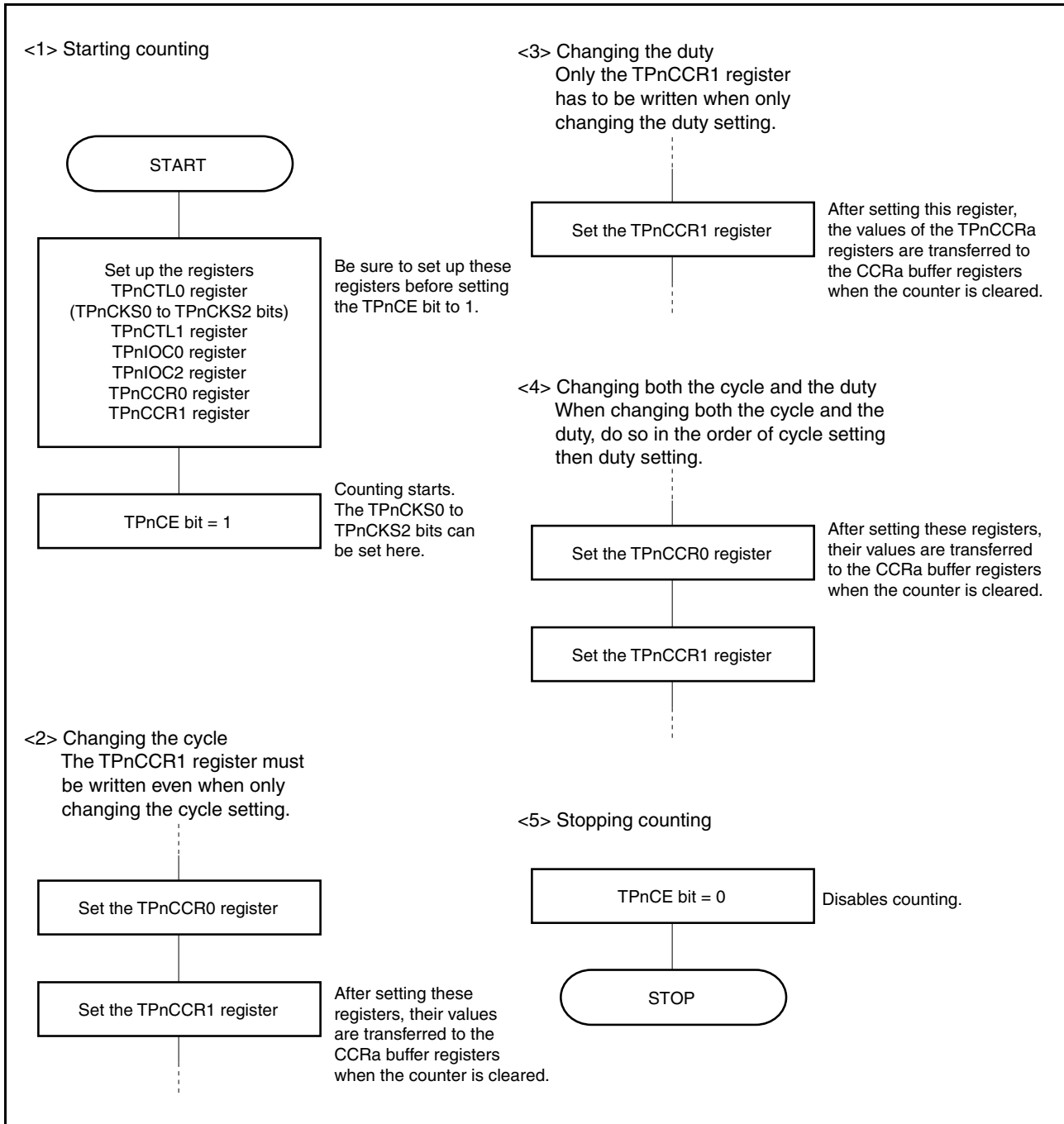
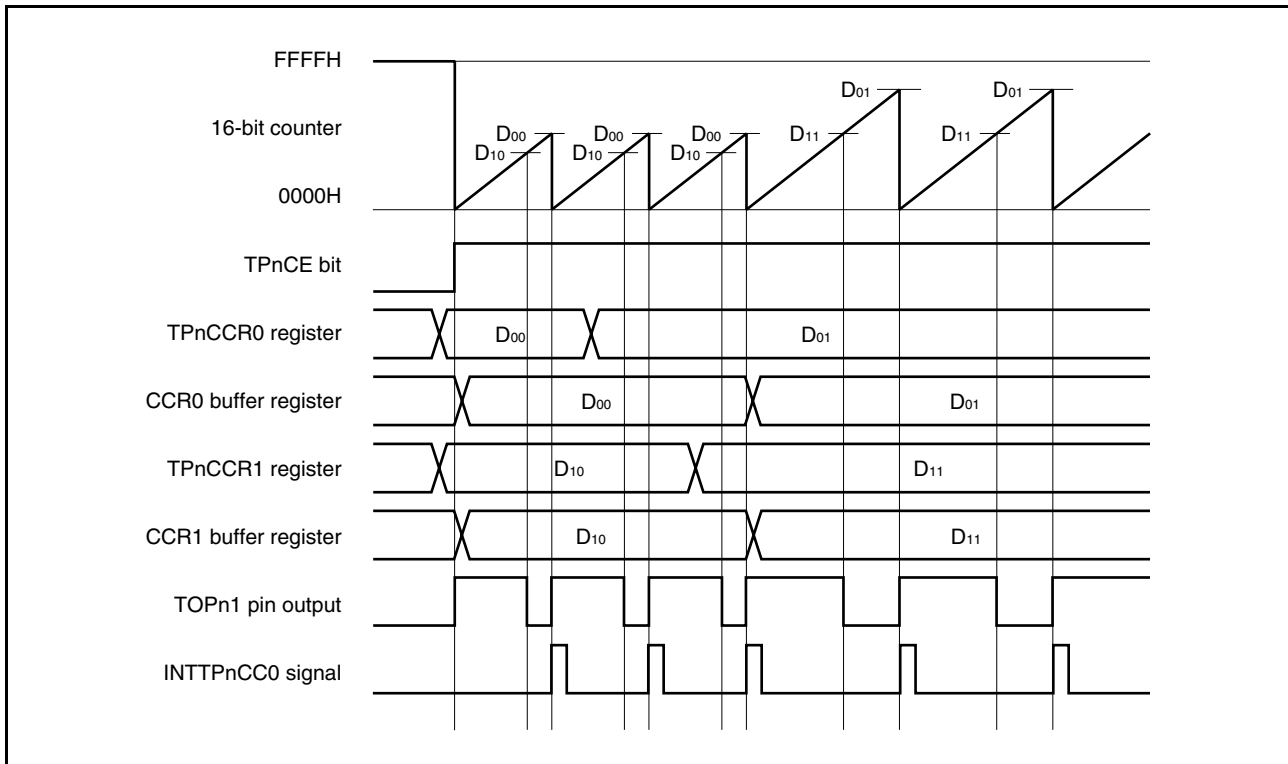


Figure 6-47. Timing and Processing of Operations in PWM Output Mode (2/2)



(2) Using PWM output mode**(a) Changing the PWM waveform while the counter is incrementing**

To change the PWM waveform while the counter is incrementing, write to the TPnCCR1 register after changing the waveform setting. When rewriting the TPnCCRa register after writing to the TPnCCR1 register, do so after the INTTPnCC0 signal has been detected.

Figure 6-48. Changing PWM Waveform While Counter Is Incrementing

In order to transfer data from the TPnCCRa register to the CCRa buffer register, the TPnCCR1 register must be written.

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 to be compared with the 16-bit counter value.

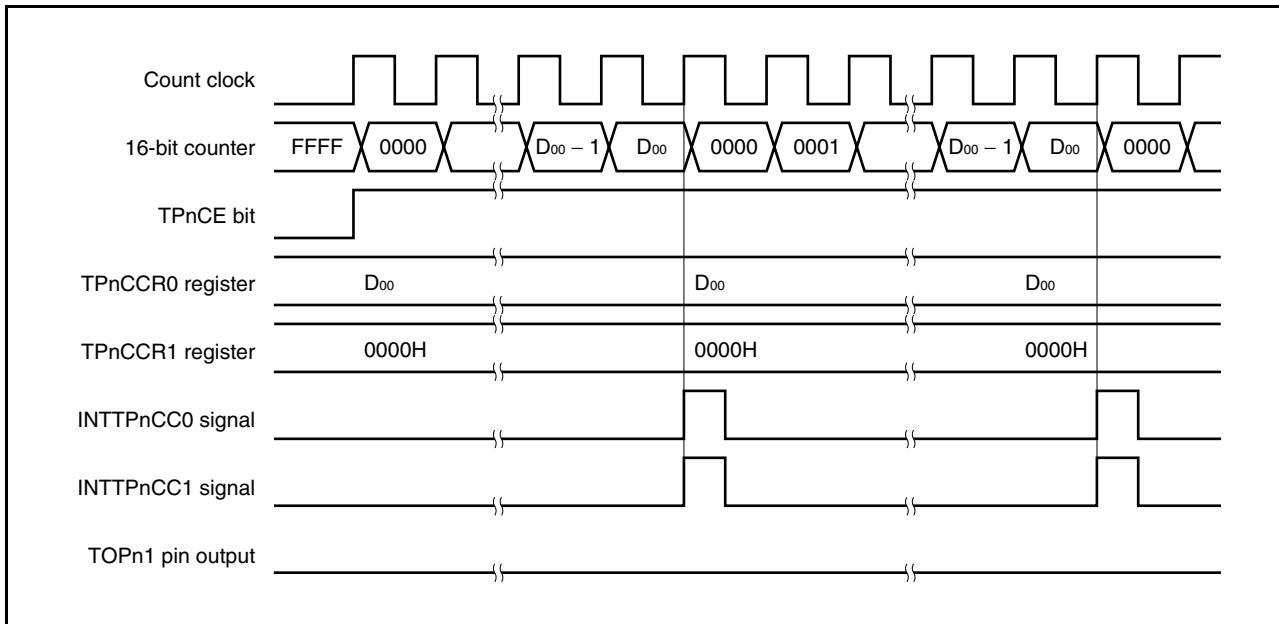
- <1> To change both the cycle and active level width of the PWM waveform, first set the cycle to the TPnCCR0 register and then set the active level width to the TPnCCR1 register.
- <2> To change only the cycle of the PWM waveform, first set the cycle to the TPnCCR0 register, and then write the same value to the TPnCCR1 register (that is, the same value as the value already specified for the TPnCCR1 register).
- <3> To change only the active level width (duty factor) of the PWM waveform, only the TPnCCR1 register has to be set.

Caution To rewrite the TPnCCR0 or TPnCCR1 register after writing the TPnCCR1 register, do so after the INTTPnCC0 signal has been 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.

(b) Outputting a 0% or 100% PWM waveform

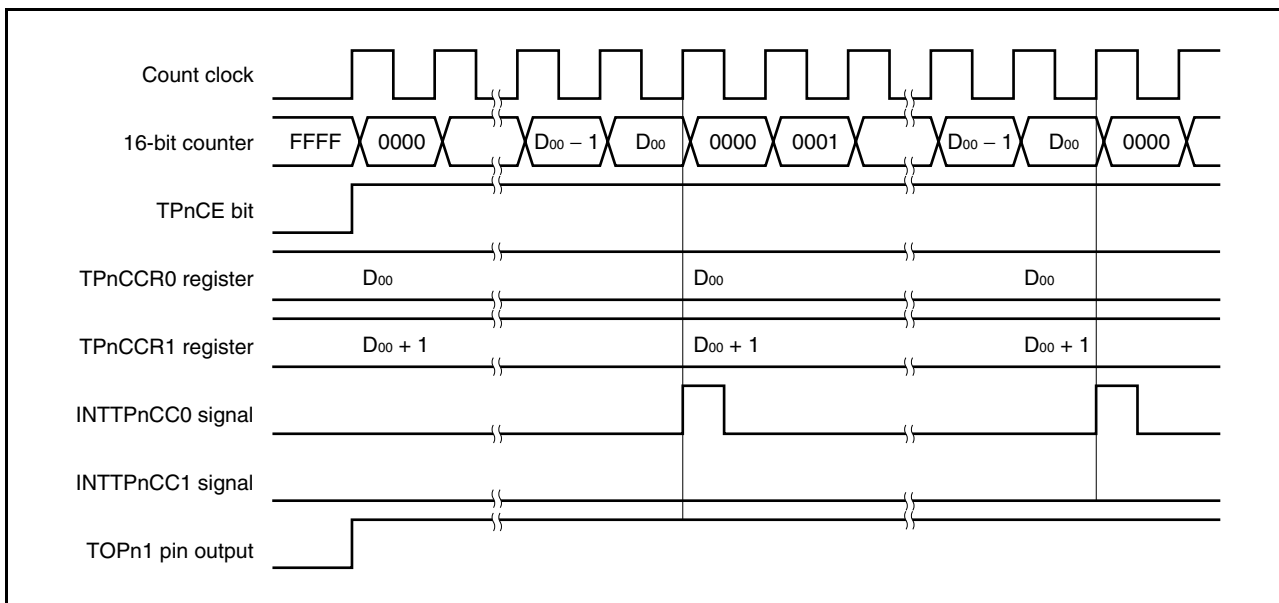
To output a 0% waveform, clear the TPnCCR1 register to 0000H.

Figure 6-49. Outputting 0% PWM Waveform



To output a 100% waveform, set the value of TPnCCR0 register + 1 to the TPnCCR1 register. If the value of the TPnCCR0 register is FFFFH, a 100% waveform cannot be output.

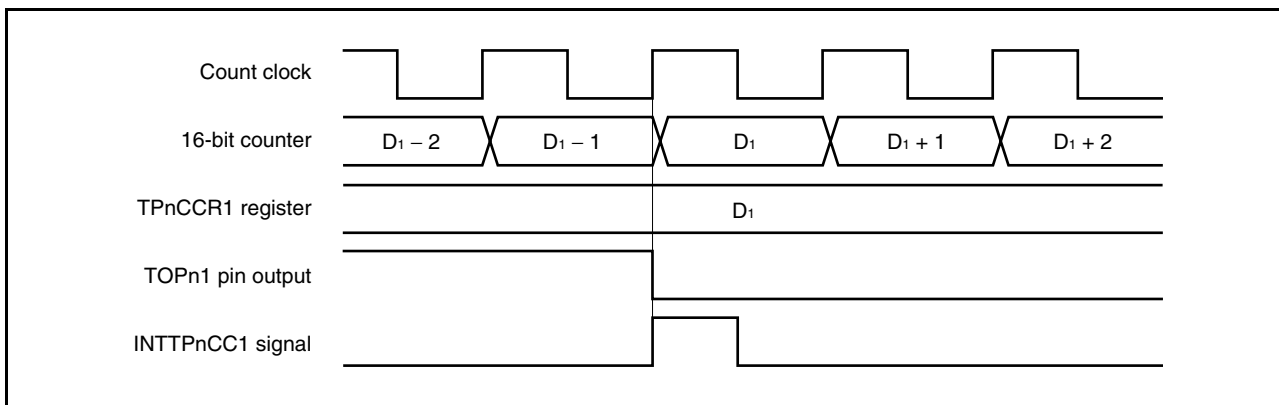
Figure 6-50. Outputting 100% PWM Waveform



(c) Timing of generating the compare match interrupt request signal (INTTPnCC1)

In the PWM output mode, the INTTPnCC1 signal is generated when the value of the 16-bit counter matches the value of the TPnCCR1 register.

Figure 6-51. Timing of Generating Compare Match Interrupt Signal (INTTPnCC1)

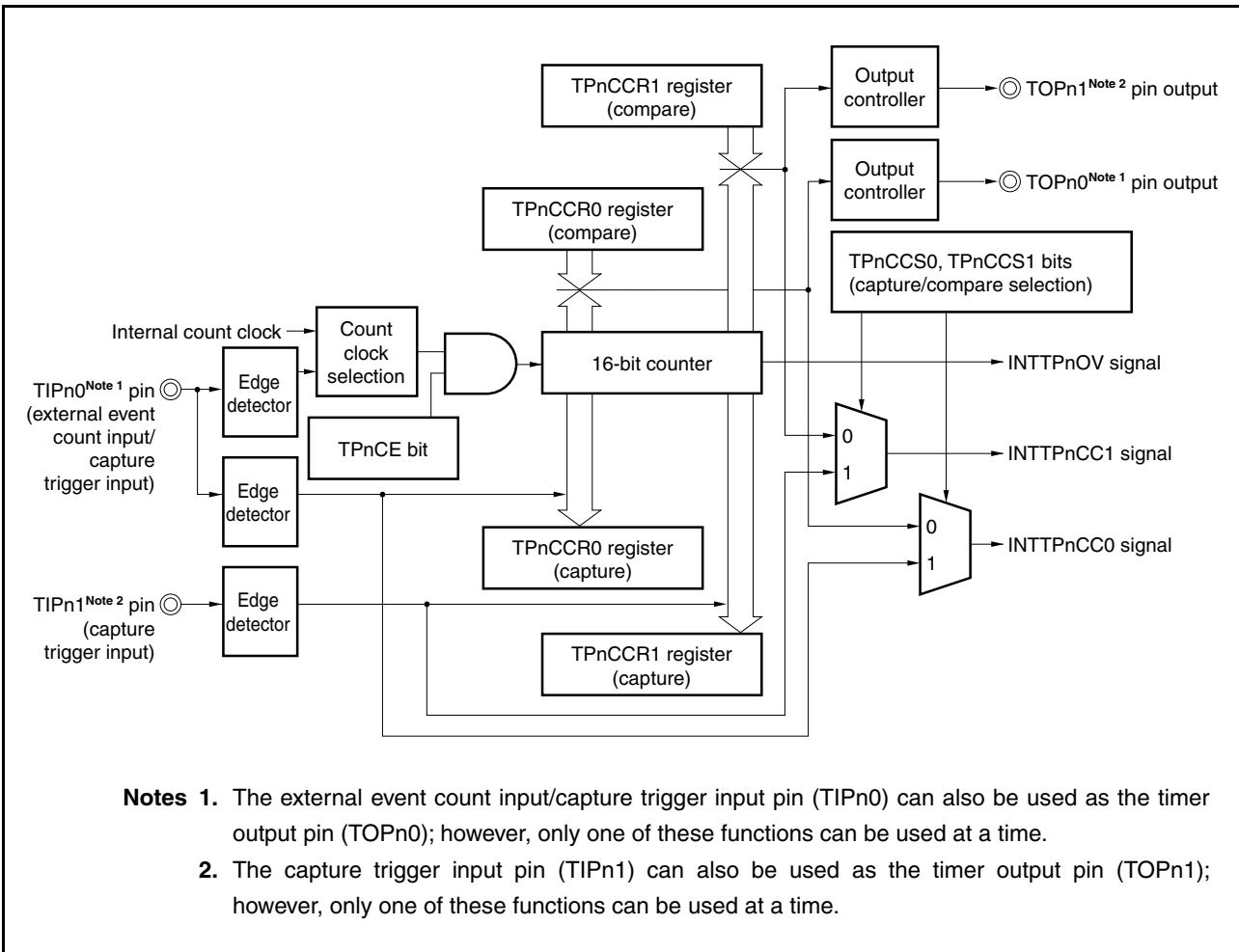


6.4.6 Free-running timer mode (TPnMD2 to TPnMD0 bits = 101)

In the free-running timer mode, TMPn starts incrementing when the TPnCTL0.TPnCE bit is set to 1. At this time, the TPnCCRa register can be used as a compare register or a capture register, according to the setting of the TPnOPT0.TPnCCS0 and TPnOPT0.TPnCCS1 bits.

- Remarks 1.** For how to set the TIPn0, TIPn1, TOPn0, and TOPn1 pins, see **Table 6-2 Pins Used by TMPn** and **Table 4-17 Settings When Pins Are Used for Alternate Functions**.
- 2.** For how to enable the INTTPnCC0 and INTTPnCC1 interrupt signals, see **CHAPTER 19 INTERRUPT SERVICING/EXCEPTION PROCESSING FUNCTION**.

Figure 6-52. Configuration of TMPn in Free-Running Timer Mode



- Notes 1.** The external event count input/capture trigger input pin (TIPn0) can also be used as the timer output pin (TOPn0); however, only one of these functions can be used at a time.
- 2.** The capture trigger input pin (TIPn1) can also be used as the timer output pin (TOPn1); however, only one of these functions can be used at a time.

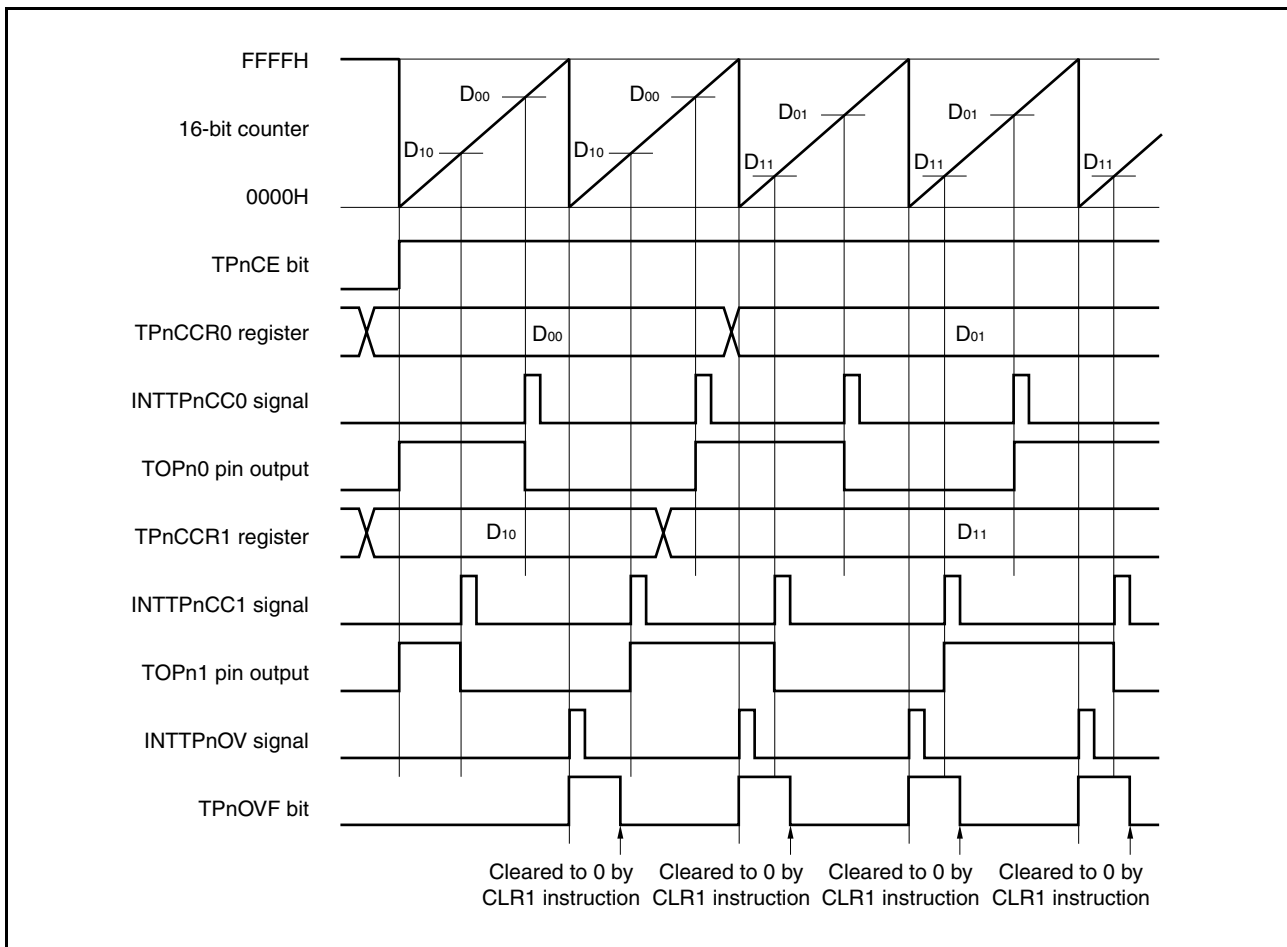
- Compare operation

When the TPnCE bit is set to 1, TMPn starts incrementing, and the output signals of the TOPn0 and TOPn1 pins are inverted. When the value of the 16-bit counter later matches the set value of the TPnCCRa register, a compare match interrupt request signal (INTTPnCCa) is generated, and the output signal of the TOPna pin is inverted.

The 16-bit counter continues incrementing in synchronization with the count clock. Once the counter reaches FFFFH, it generates an overflow interrupt request signal (INTTPnOV) at the next clock, is cleared to 0000H, and continues incrementing. At this time, the overflow flag (the TPnOPT0.TPnOVF bit) is also set to 1. The overflow flag must be cleared to 0 by executing a CLR1 software instruction.

The TPnCCRa register can be rewritten while the counter is incrementing. If it is rewritten, the new value is immediately applied, and compared with the count value.

Figure 6-53. Basic Timing of Operations in Free-Running Timer Mode (Compare Function)



- Capture operation

When the TPnCE bit is set to 1, the 16-bit counter starts incrementing. When it is detected that a valid edge as been input to the TIPna pin, the value of the 16-bit counter is stored in the TPnCCRa register, and a capture interrupt request signal (INTTPnCCa) is generated.

The 16-bit counter continues incrementing in synchronization with the count clock. When the counter reaches FFFFH, it generates an overflow interrupt request signal (INTTPnOV) at the next clock, is cleared to 0000H, and continues incrementing. At this time, the overflow flag (the TPnOPT0.TPnOVF bit) is also set to 1. The overflow flag must be cleared to 0 by executing a CLR1 software instruction.

Figure 6-54. Basic Timing of Operations in Free-Running Timer Mode (Capture Function)

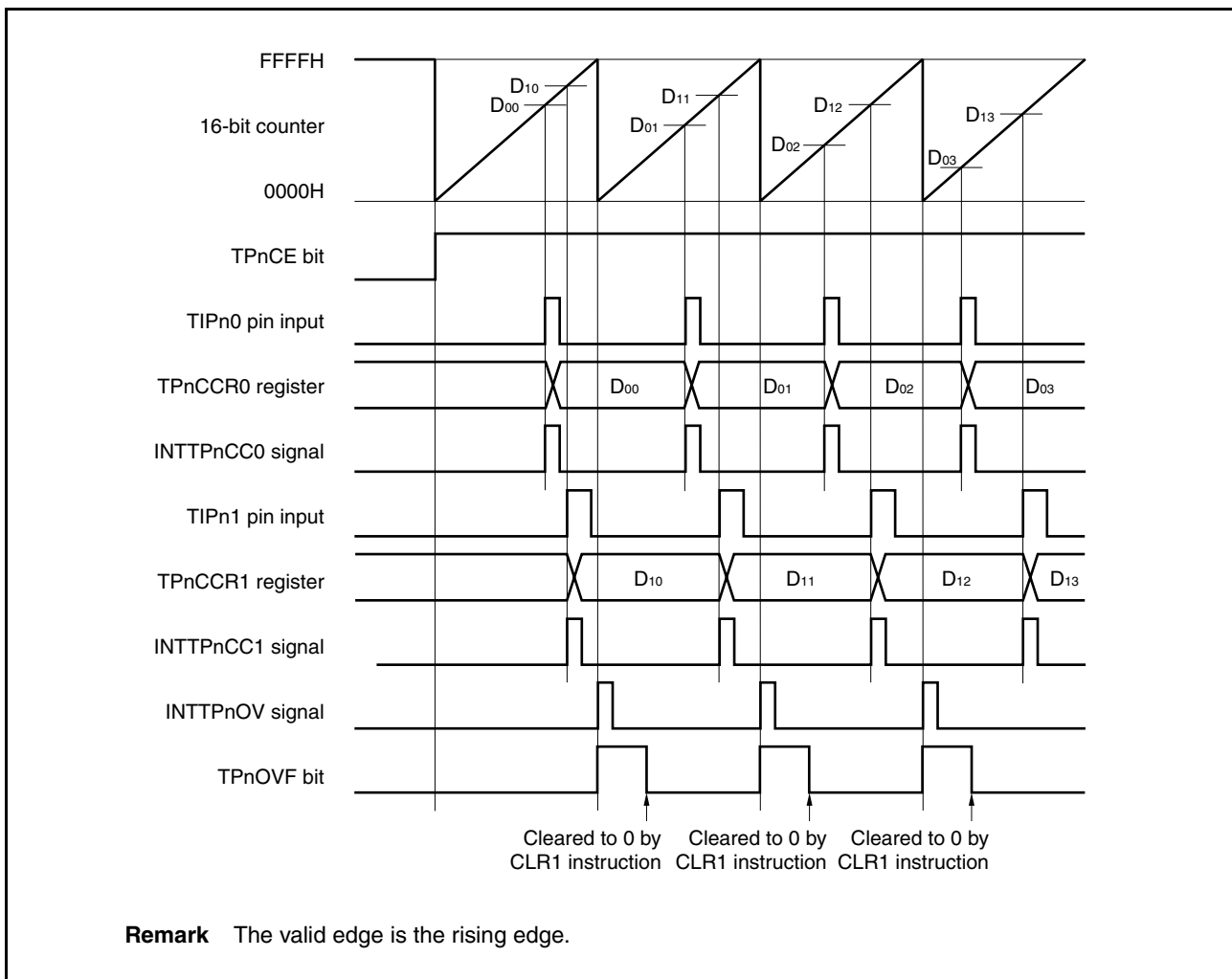


Figure 6-55. Register Settings in Free-Running Timer Mode (1/2)

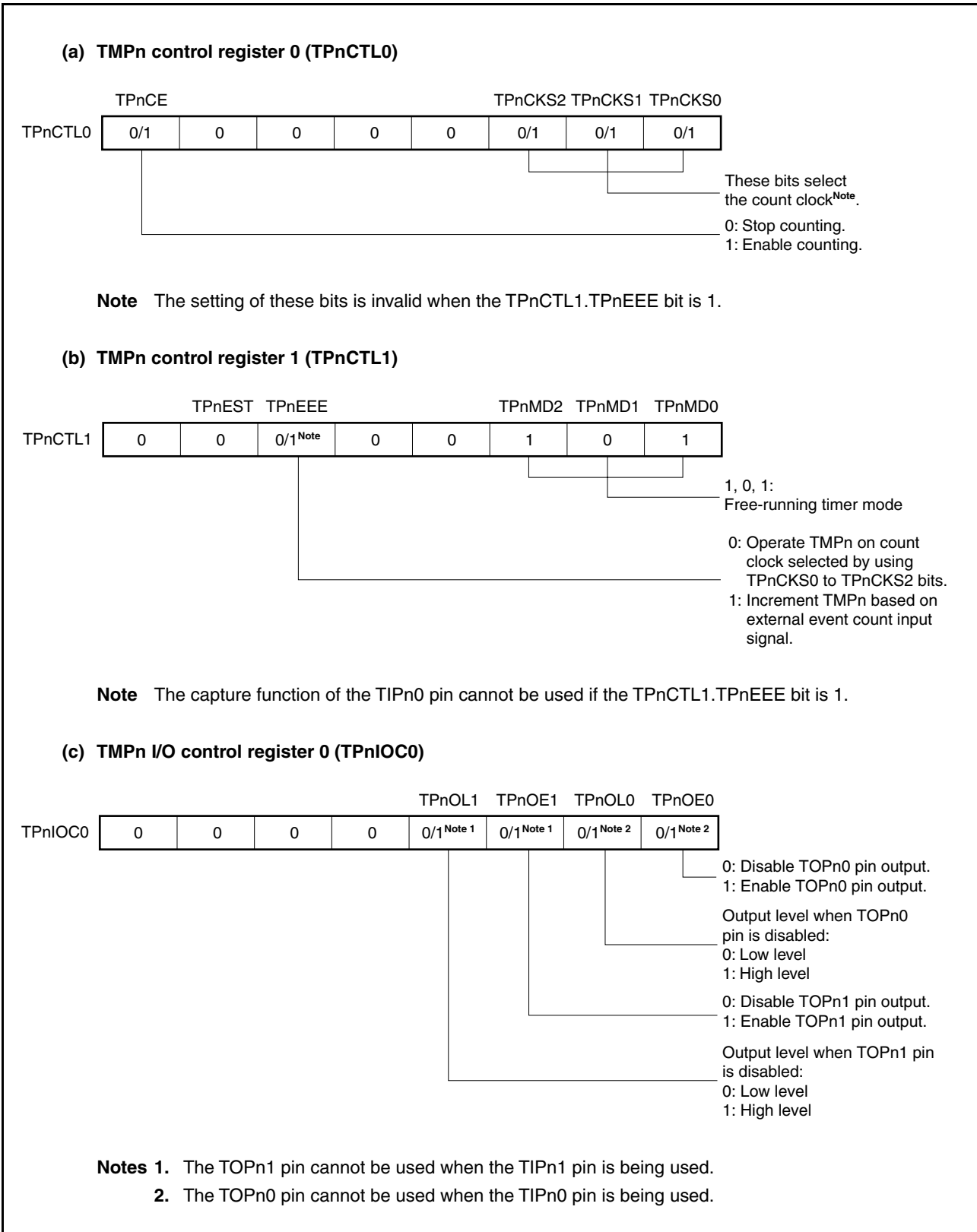
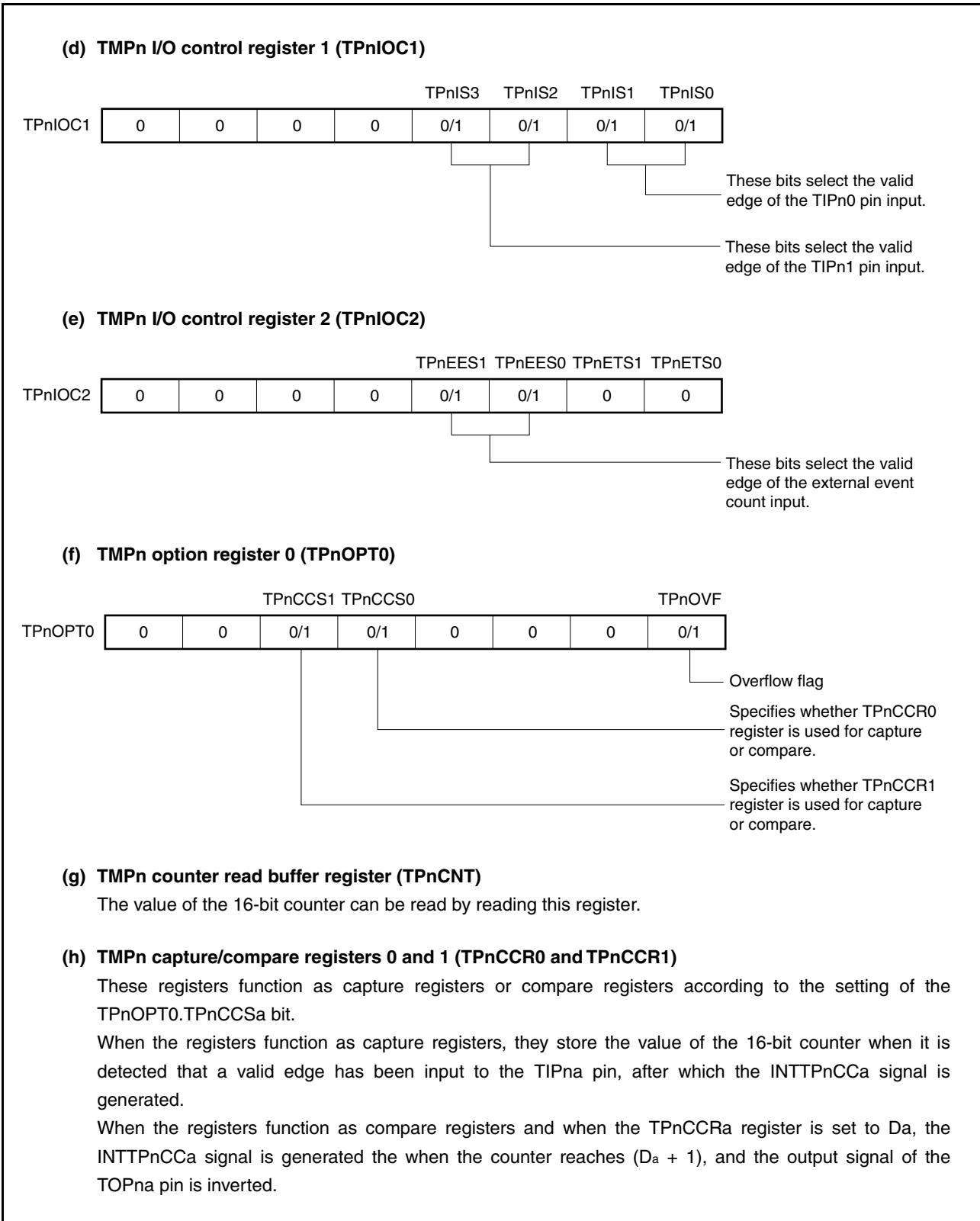


Figure 6-55. Register Settings in Free-Running Timer Mode (2/2)



(1) Operations in free-running timer mode

The following two operations occur in the free-running timer mode:

- Capture operations
- Compare operations

(a) Using a capture/compare register as a compare register

Figure 6-56. Timing and Processing of Operations in Free-Running Timer Mode (Compare Function) (1/2)

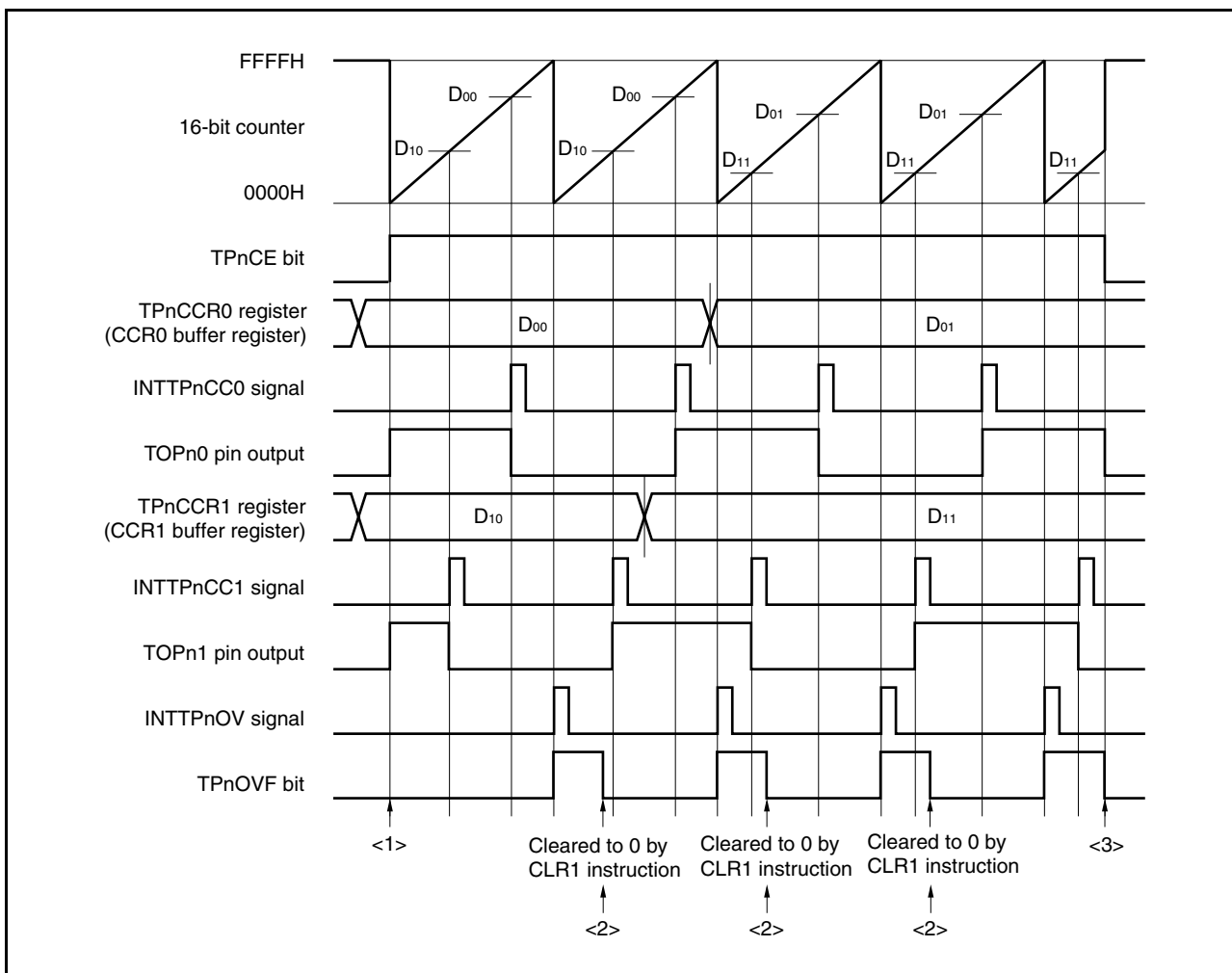
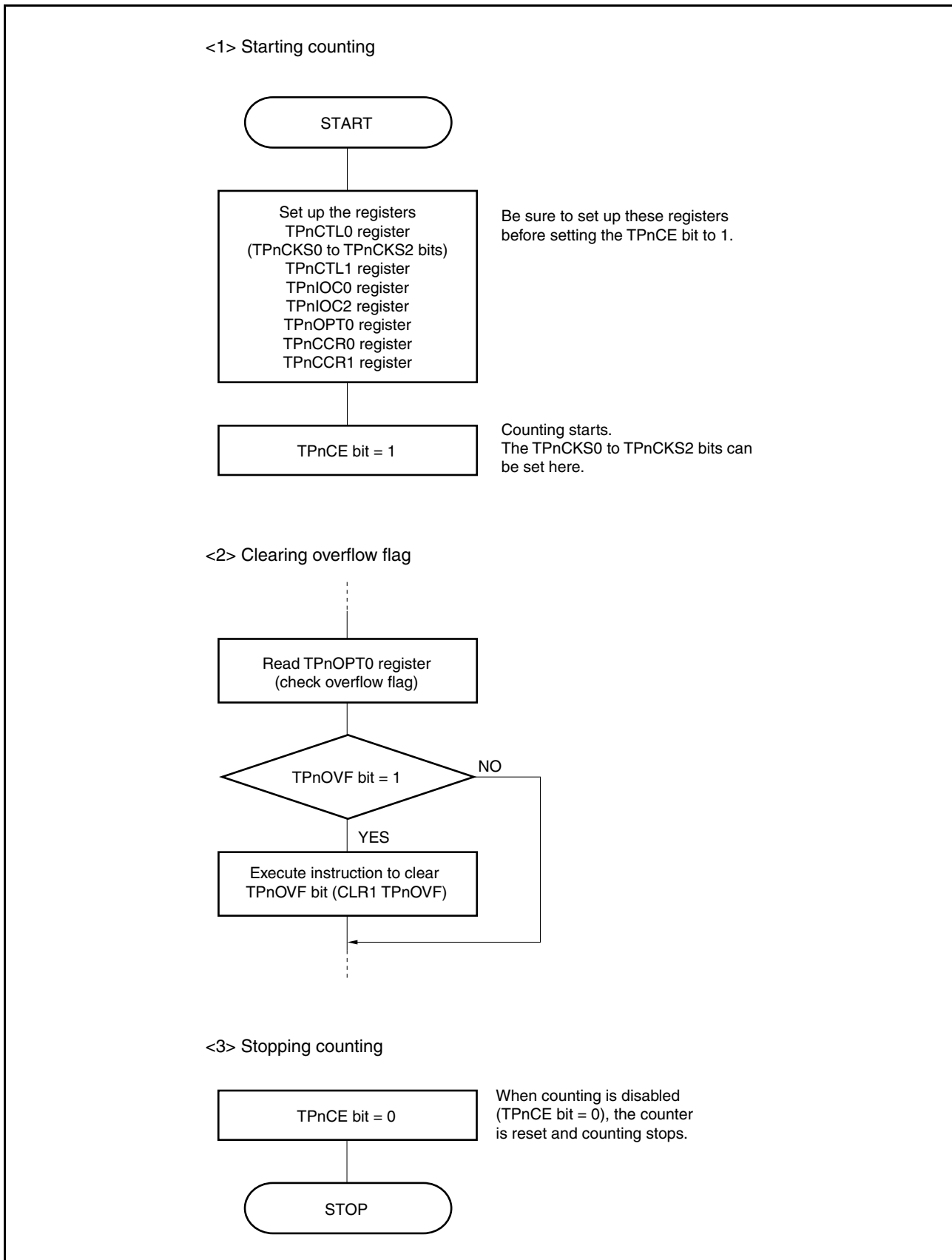


Figure 6-56. Timing and Processing of Operations in Free-Running Timer Mode (Compare Function) (2/2)



(b) Using a capture/compare register as a capture register

Figure 6-57. Timing and Processing of Operations in Free-Running Timer Mode (Capture Function) (1/2)

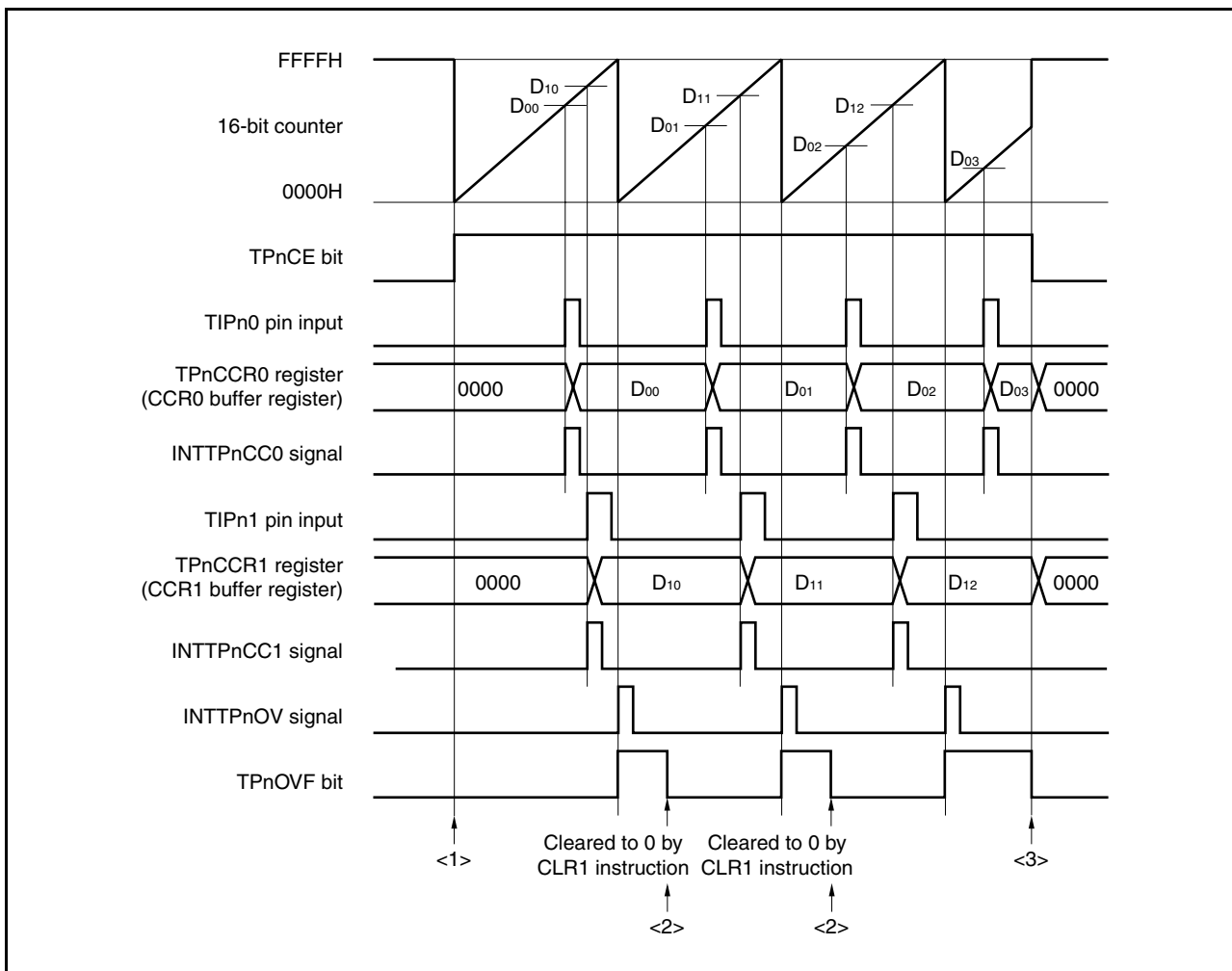
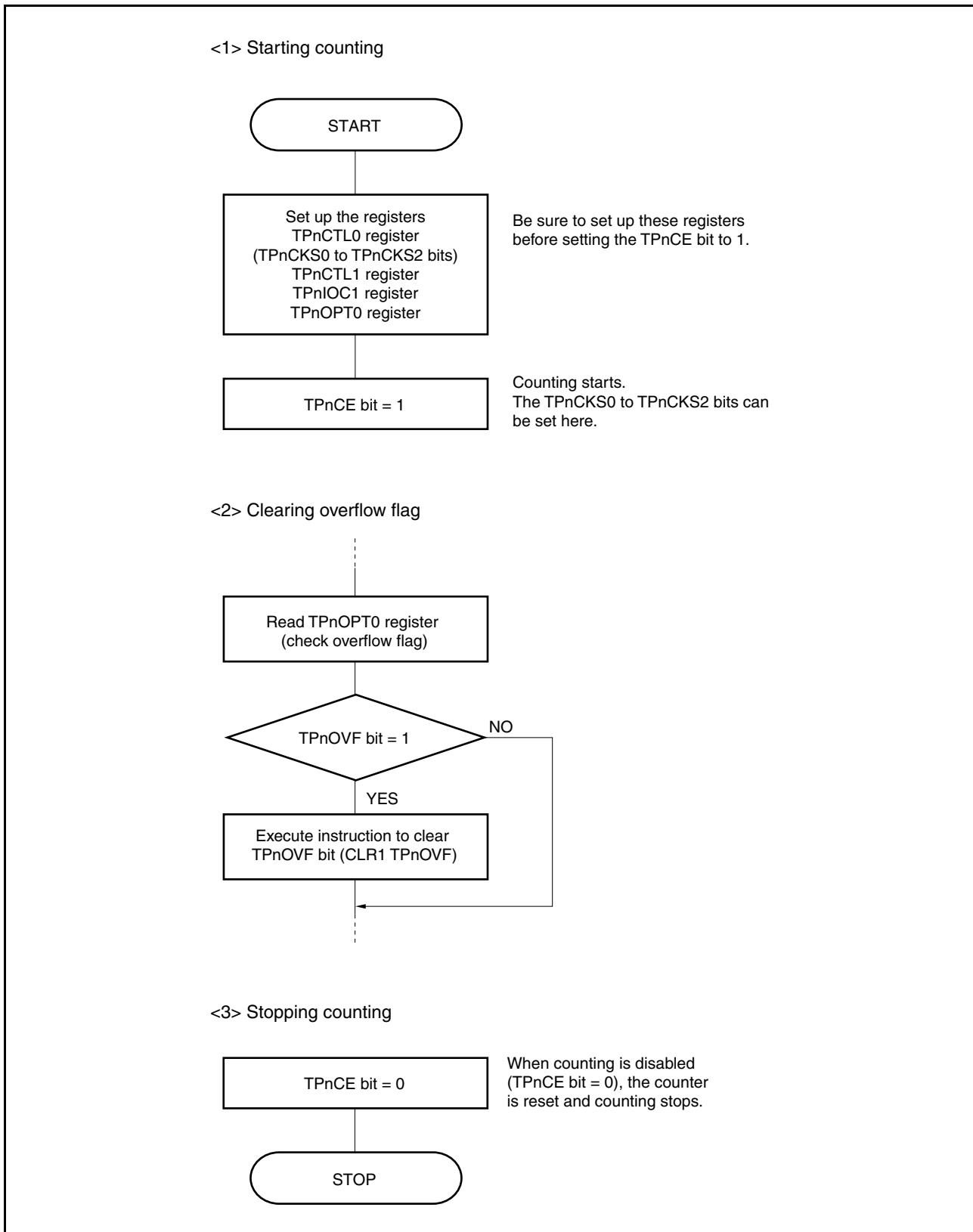


Figure 6-57. Timing and Processing of Operations in Free-Running Timer Mode (Capture Function) (2/2)

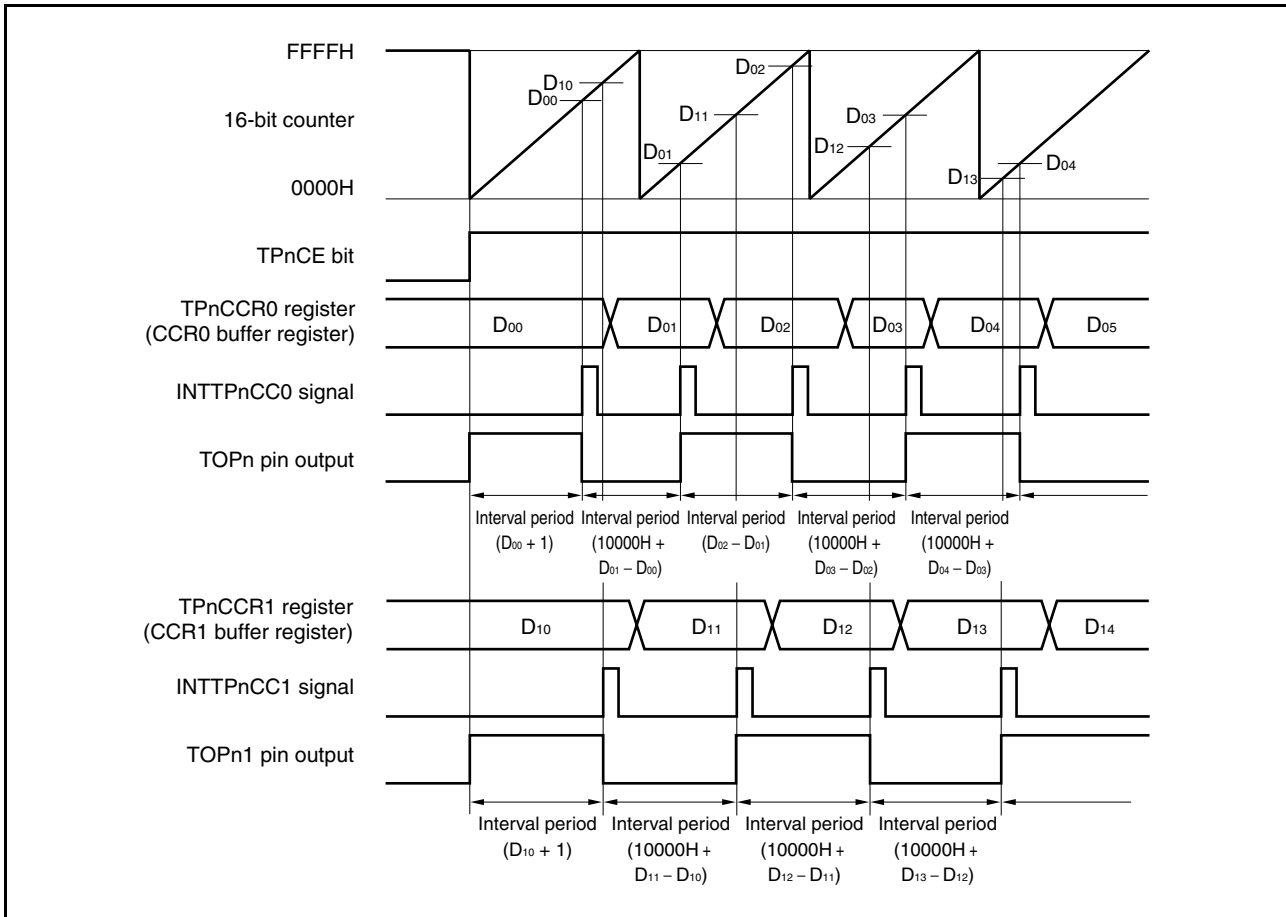


(2) Using free-running timer mode

(a) Interval operation using the TPnCCRa register as a compare register

When TMPn is used as an interval timer with the TPnCCRa register used as a compare register, the comparison value at which the next interrupt request signal is generated each time the INTTPnCCa signal has been detected must be set by software.

Figure 6-58. Interval Operation of TMPn in Free-Running Timer Mode



When performing an interval operation in the free-running timer mode, two intervals can be set for one channel. To perform the interval operation, the value of the corresponding TPnCCRa register must be set again in the interrupt servicing that is executed when the INTTPnCCa signal is detected. The value to be set in this case can be calculated by the following expression, where “Da” is the interval period.

Compare register default value: $D_a - 1$

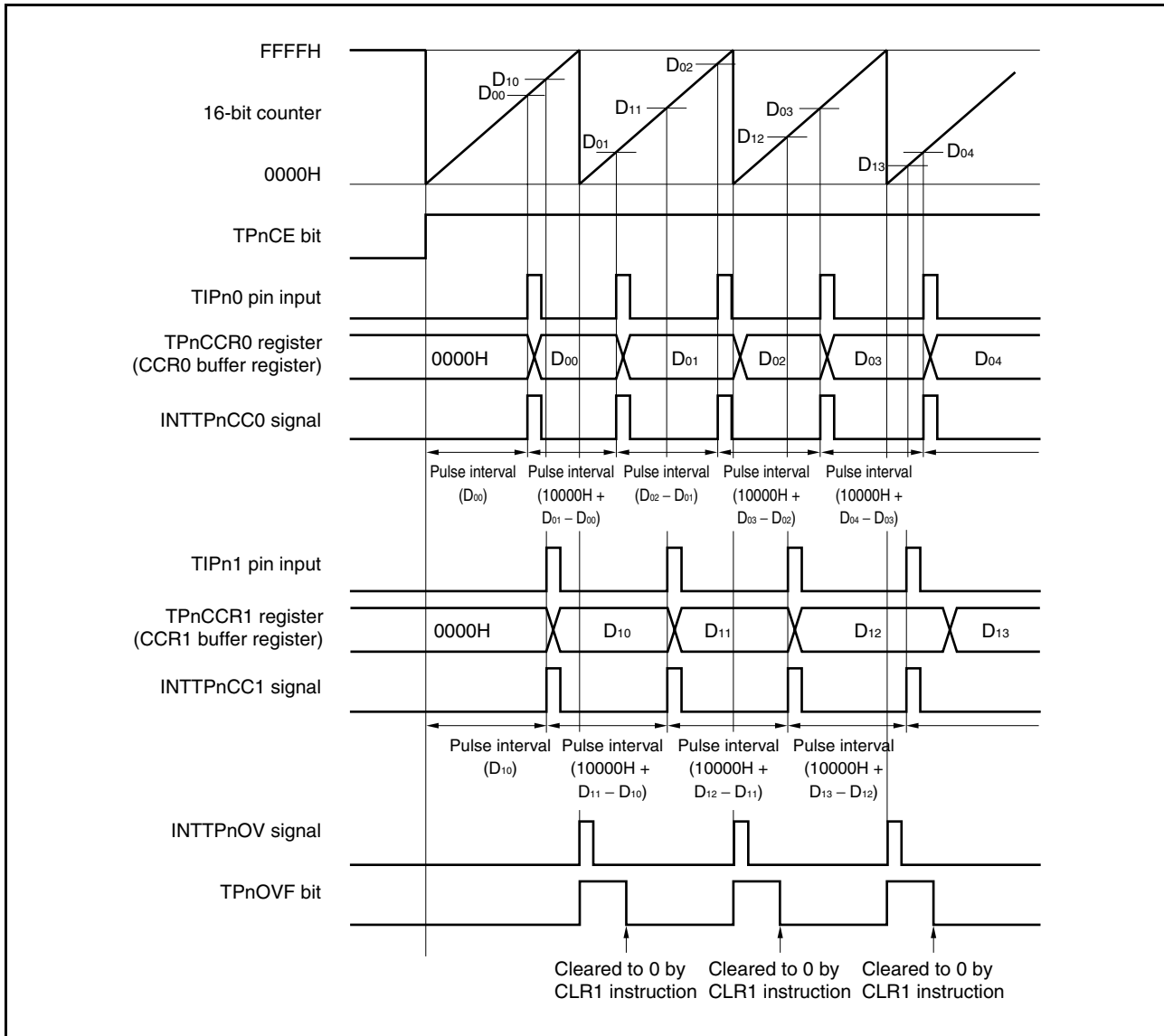
Value set to compare register second and subsequent time: Previous set value + D_a

(If the calculation result is greater than FFFFH, subtract 10000H from the result and set the register to this value.)

(b) Pulse width measurement using the TPnCCRa register as a capture register

When pulse width measurement is performed with the TPnCCRa register used as a capture register, each time the INTTPnCCa signal has been detected, the capture register must be read and the interval must be calculated by software.

Figure 6-59. Pulse Width Measurement by TMPn in Free-Running Timer Mode



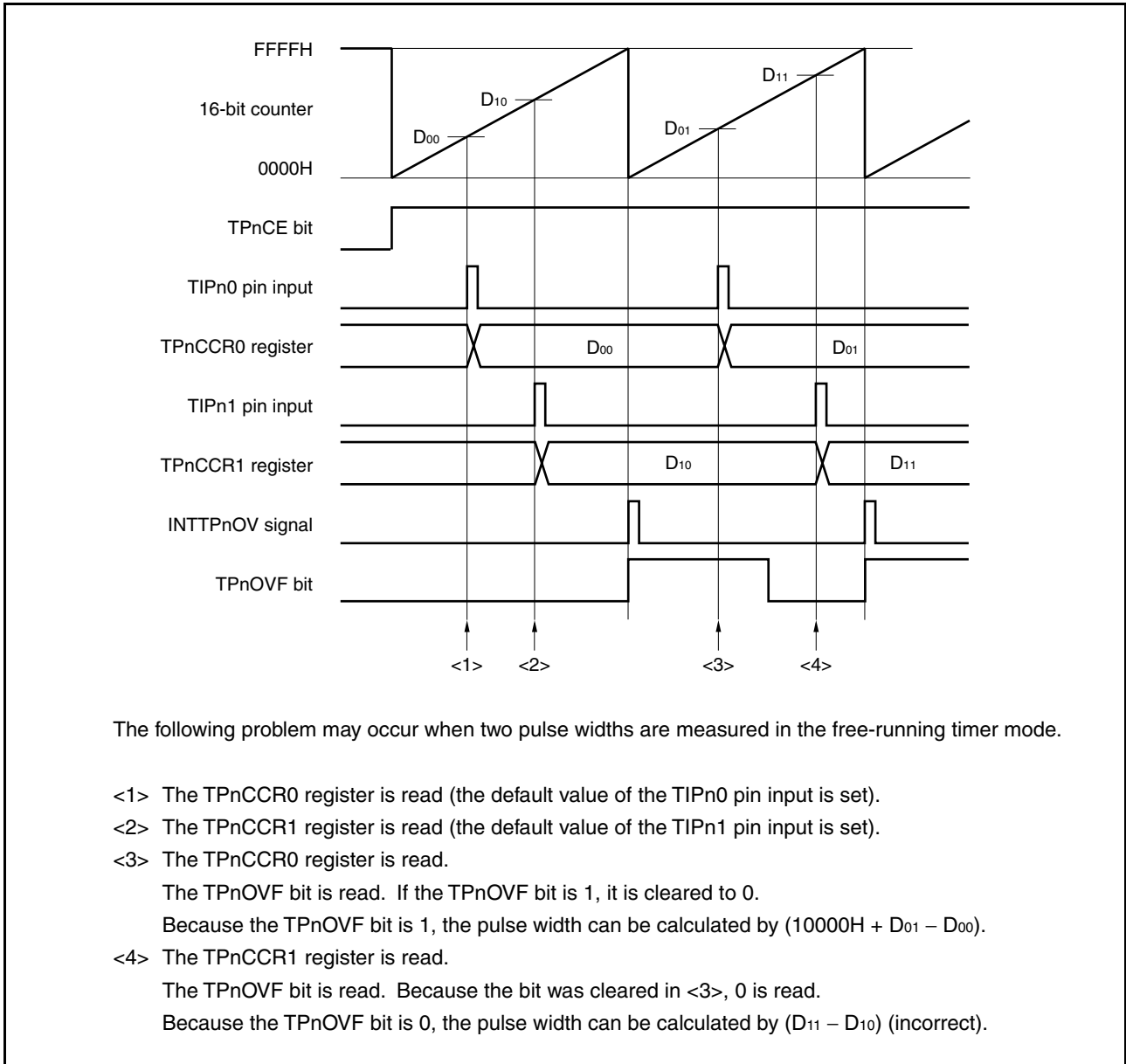
When executing pulse width measurement in the free-running timer mode, two pulse widths can be measured for one channel.

When measuring a pulse width, the pulse width can be calculated by reading the value of the TPnCCRa register in synchronization with the INTTPnCCa signal, and calculating the difference between that value and the previously read value.

(c) Processing an 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.

Figure 6-60. 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> The TPNCCR0 register is read (the default value of the TIPn0 pin input is set).

<2> The TPNCCR1 register is read (the default value of the TIPn1 pin input is set).

<3> The TPNCCR0 register is read.

The TPNOVF bit is read. If the TPNOVF bit is 1, it is cleared to 0.

Because the TPNOVF bit is 1, the pulse width can be calculated by $(10000H + D_{01} - D_{00})$.

<4> The TPNCCR1 register is read.

The TPNOVF bit is read. Because the bit was cleared in <3>, 0 is read.

Because the TPNOVF bit 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.

This problem can be resolved by using software, as shown in the example below.

Figure 6-61. Example of Resolving Problem When Two Capture Registers Are Used By Using Overflow Interrupt

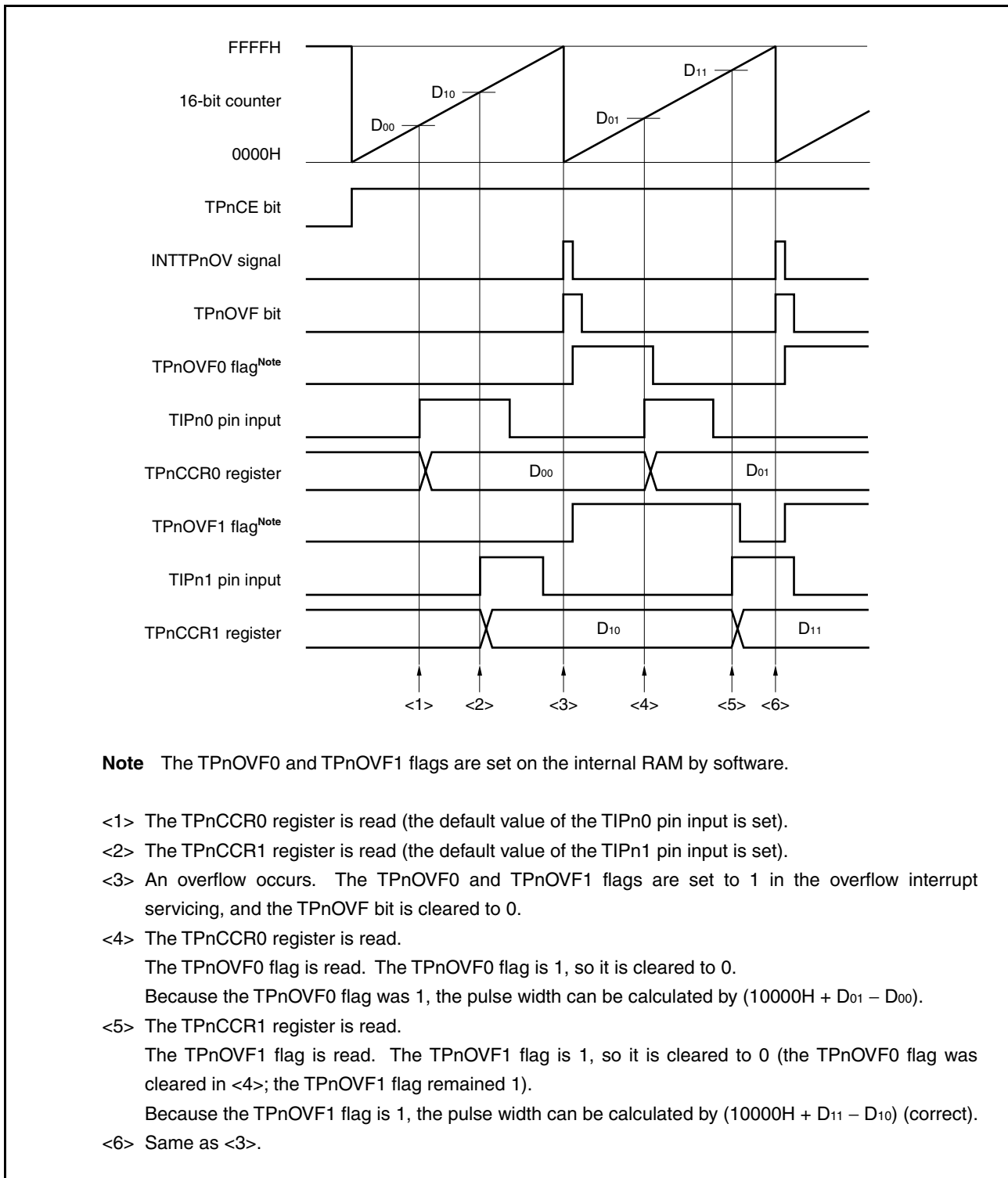
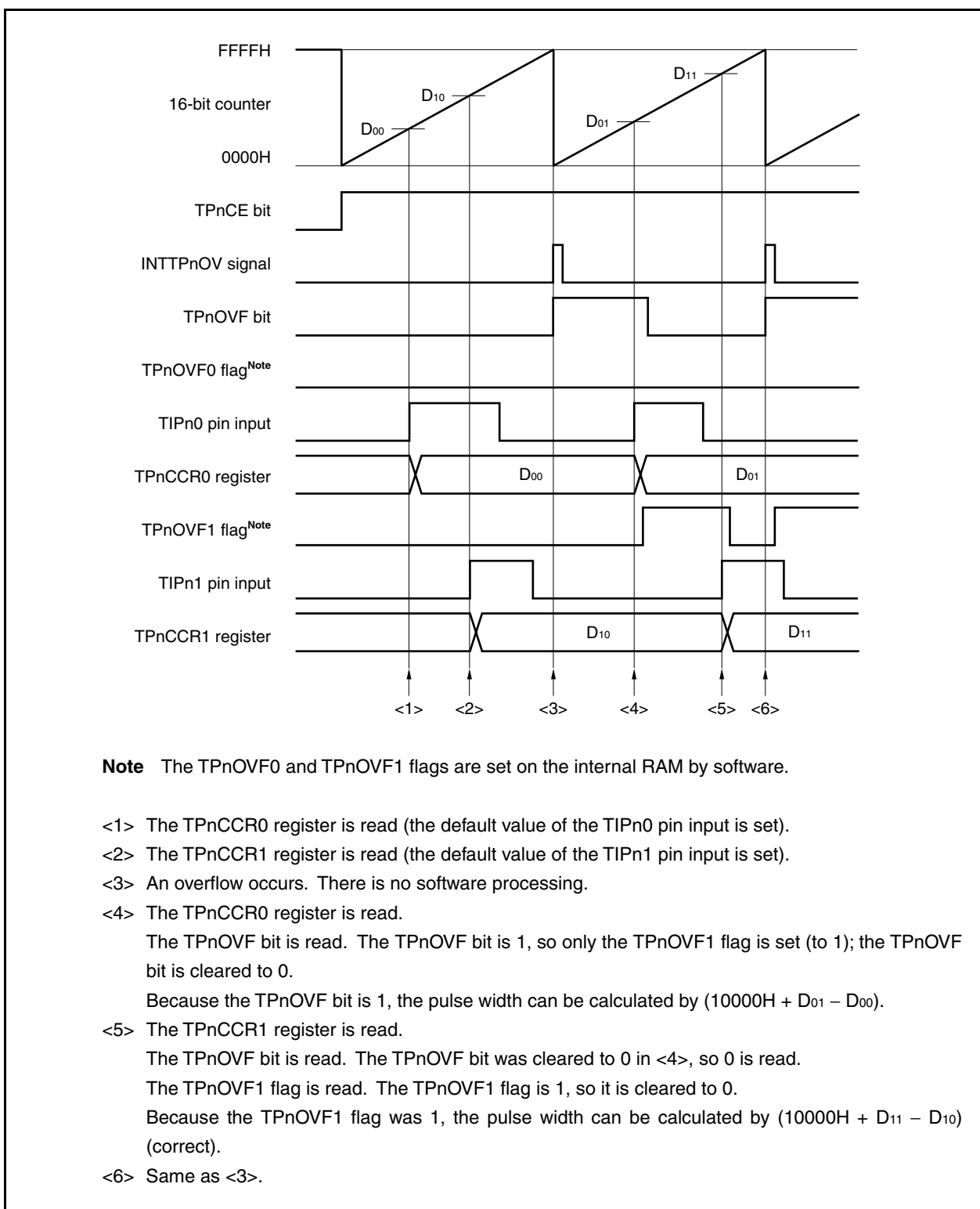


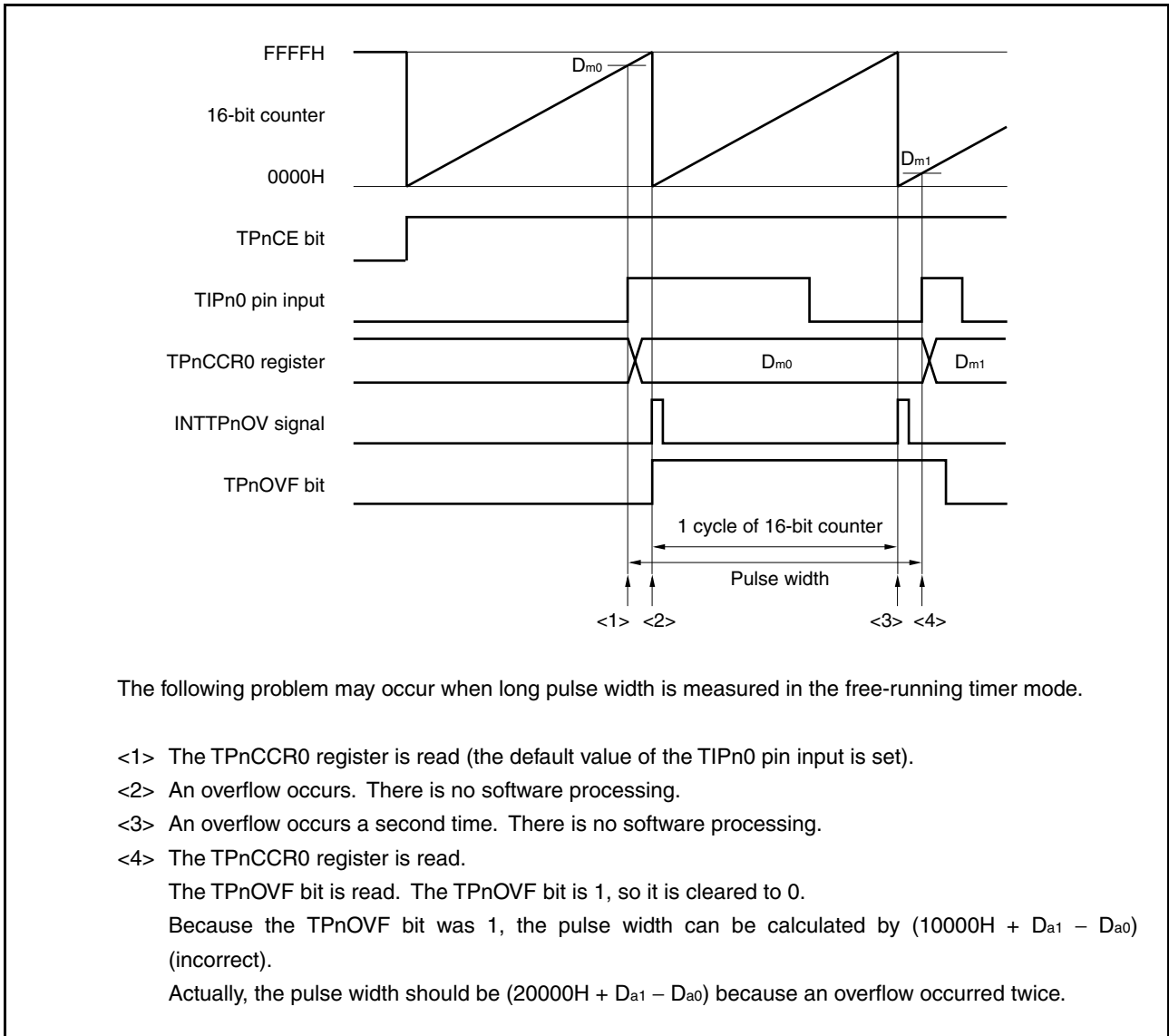
Figure 6-62. Example of Resolving Problem When Two Capture Registers Are Used Without Using Overflow Interrupt



(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 between the first capture trigger and the next. First, an example of incorrect processing is shown below.

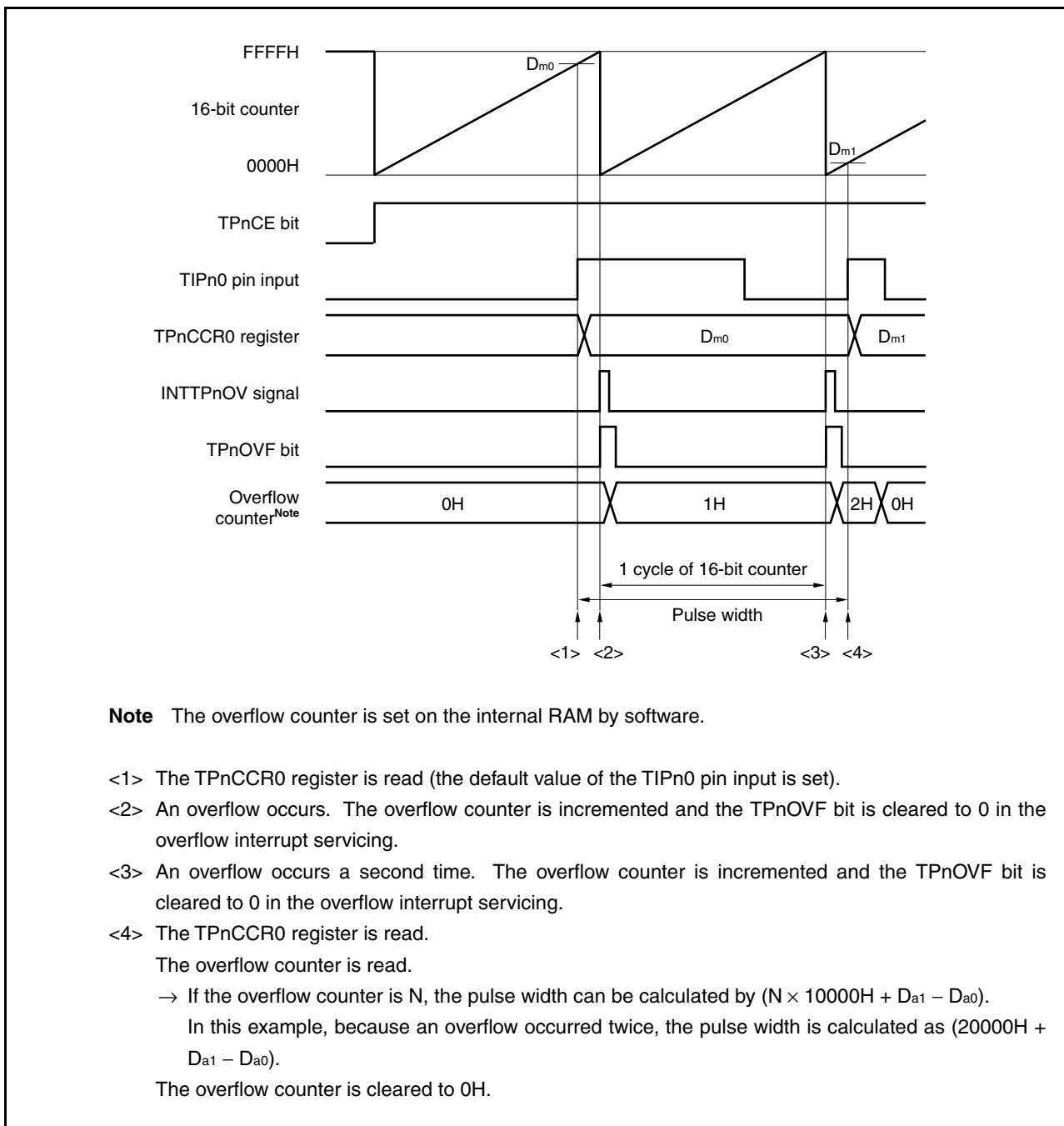
Figure 6-63. Example of Incorrect Processing When Capture Trigger Interval Is Long (When Using TIPn0)



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 to resolve the problem. An example of how to use software to resolve the problem is shown below.

Figure 6-64. Example of Using Software Processing to Resolve Problem When Capture Trigger Interval Is Long (When Using TIPn0)



(e) Clearing the overflow flag (TPnOVF)

The overflow flag (TPnOVF) can be cleared to 0 by reading the TPnOVF bit and, if its value is 1, either clearing the bit to 0 by using the CLR1 instruction or by writing 8-bit data (with bit 0 as 0) to the TPnOPT0 register.

6.4.7 Pulse width measurement mode (TPnMD2 to TPnMD0 bits = 110)

In the pulse width measurement mode, TMPn starts incrementing when the TPnCTL0.TPnCE bit is set to 1. Each time it is detected that a valid edge has been input to the TIPna pin, the 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 (INTTPnCCa) occurs.

Select either the TIPn0 or TIPn1 pin as the capture trigger input pin. Specify “No edge detected” by using the TPnIOC1 register for the unused pins.

- Remarks 1.** For how to set the TIPn0 and TIPn1 pins, see **Table 6-2 Pins Used by TMPn** and **Table 4-17 Settings When Pins Are Used for Alternate Functions**.
- 2.** For how to enable the INTTPnCC0 and INTTPnCC1 interrupt signals, see **CHAPTER 19 INTERRUPT SERVICING/EXCEPTION PROCESSING FUNCTION**.

Figure 6-65. Configuration of TMPn in Pulse Width Measurement Mode

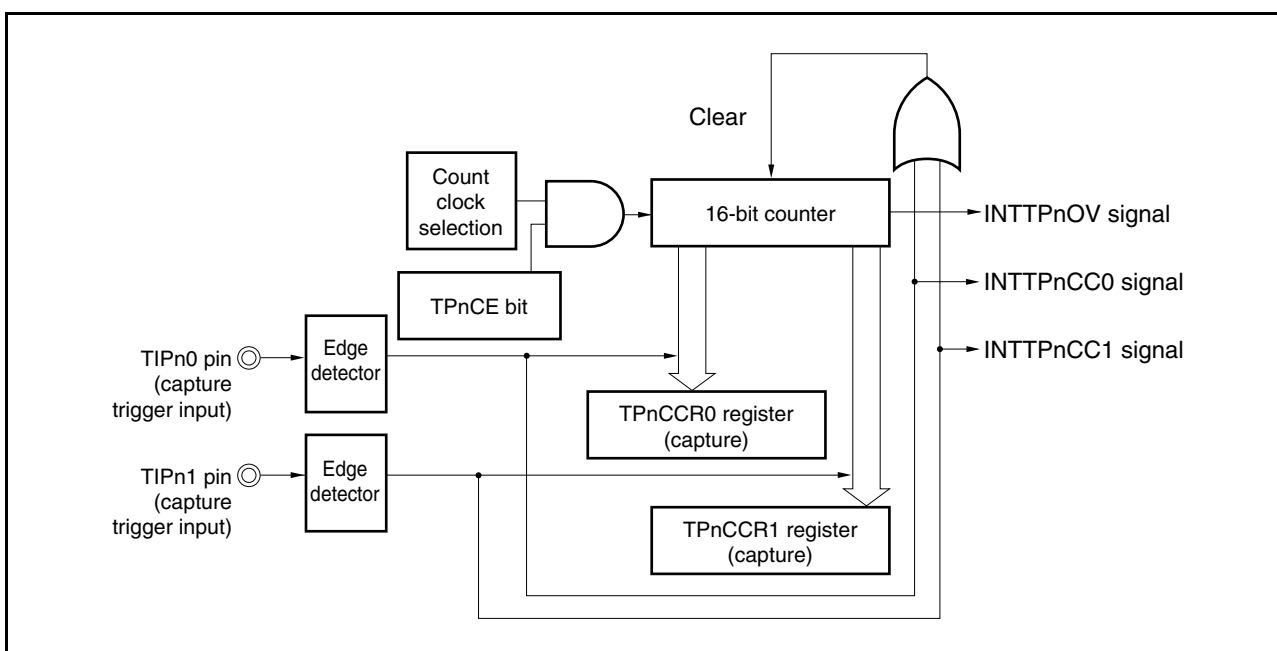
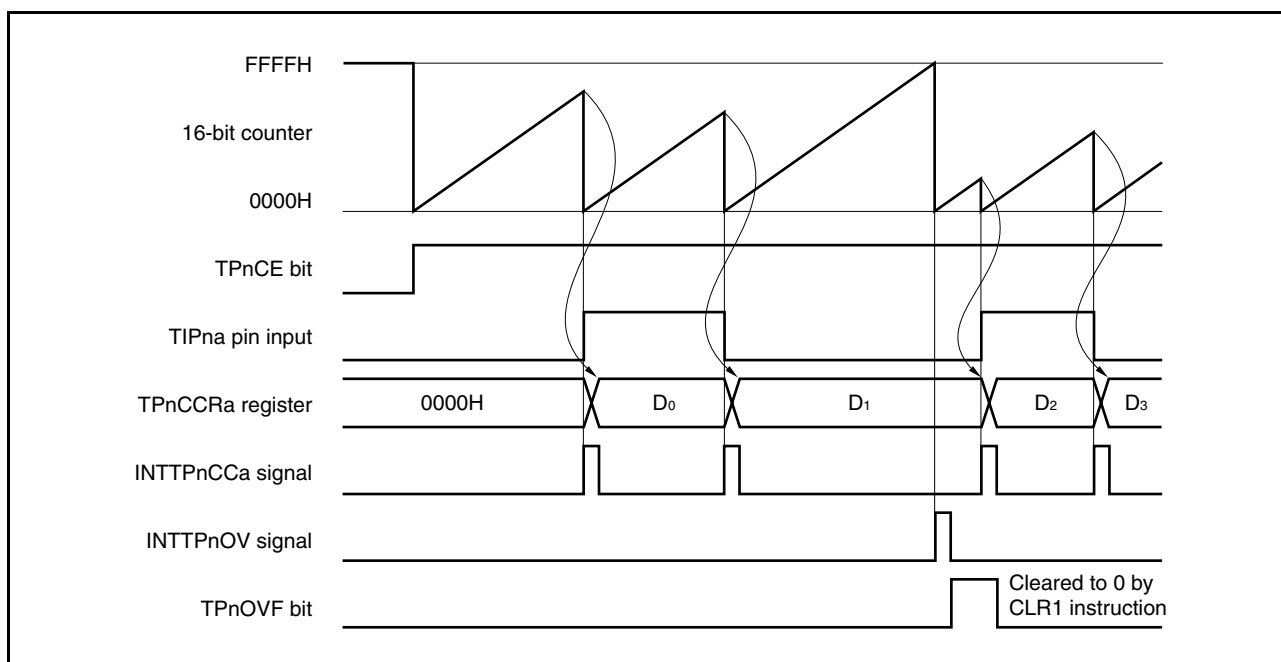


Figure 6-66. Basic Timing of Operations in Pulse Width Measurement Mode



When the TPnCE bit is set to 1, the 16-bit counter starts incrementing. When it is subsequently detected that a valid edge has been input to the TIPna pin, the 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 (INTTPnCCa) is generated.

The pulse width is calculated as follows.

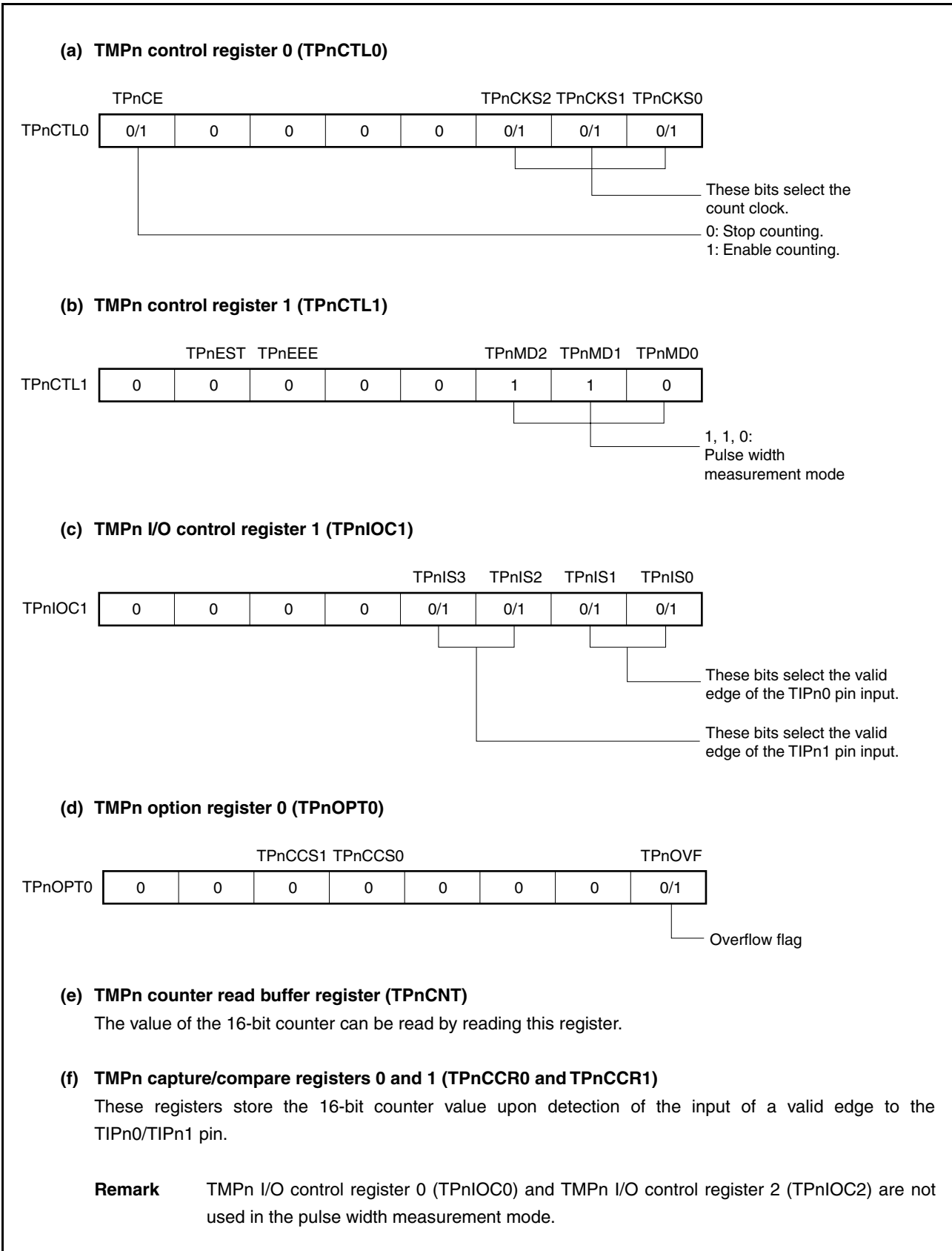
$$\text{Pulse width} = \text{Captured value} \times \text{Count clock cycle}$$

If a valid edge has not been input to the TIPna pin by the time the 16-bit counter has incremented up to FFFFH, an overflow interrupt request signal (INTTPnOV) is generated at the next count clock, and the counter is cleared to 0000H and continues incrementing. At this time, the overflow flag (TPnOPT0.TPnOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the software instruction CLR1.

If the overflow flag is set to 1, the pulse width can be calculated as follows.

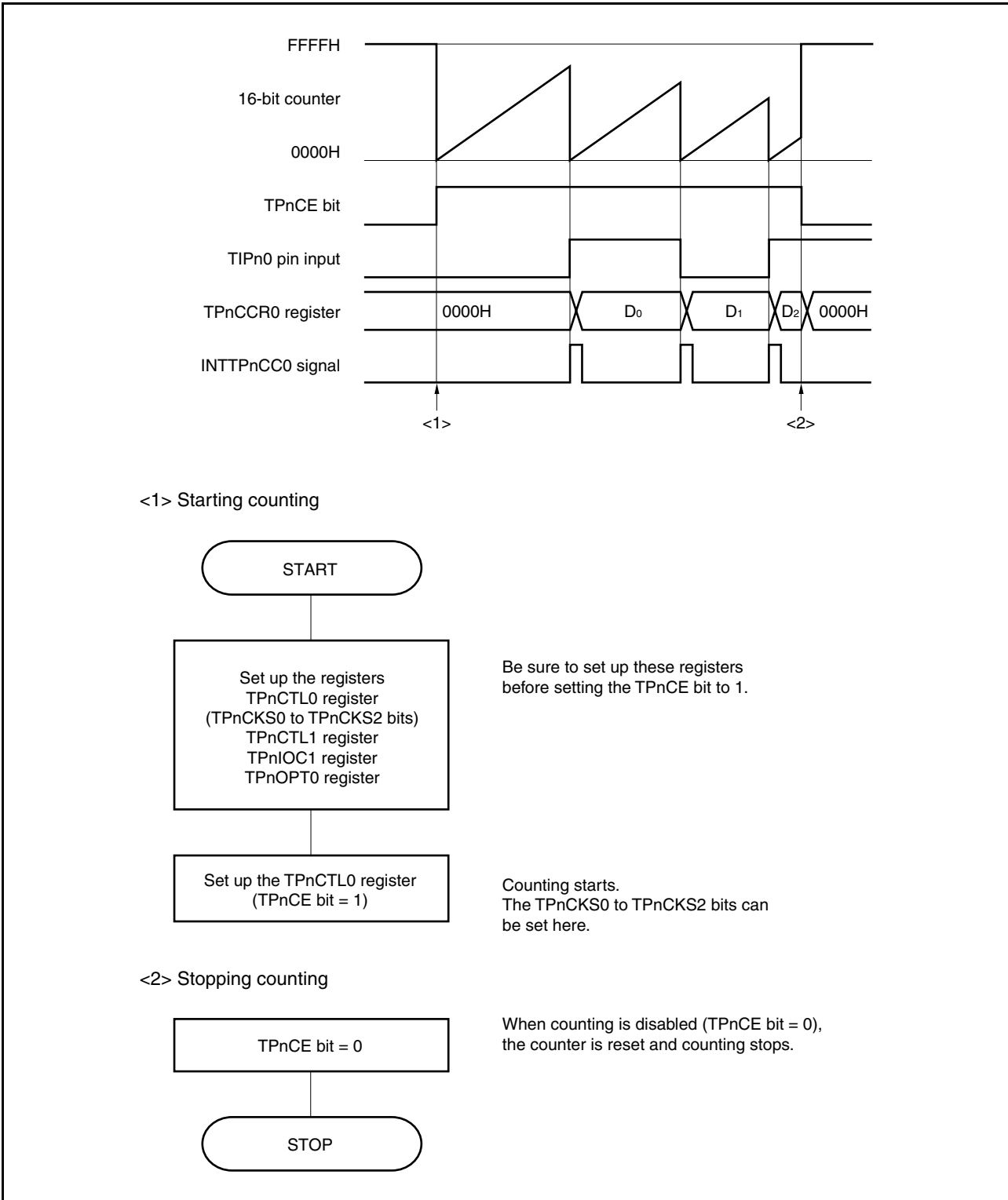
$$\text{Pulse width} = (10000\text{H} \times \text{Number of times the TPnOVF bit was set (1)} + \text{Captured value}) \times \text{Count clock cycle}$$

Figure 6-67. Register Settings in Pulse Width Measurement Mode



(1) Operations in pulse width measurement mode

Figure 6-68. Timing and Processing of Operations in Pulse Width Measurement Mode



(2) Using pulse width measurement mode**(a) Clearing the overflow flag (TPnOVF)**

The overflow flag (TPnOVF) can be cleared to 0 by reading the TPnOVF bit and, if its value is 1, either clearing the bit to 0 by using the CLR1 instruction or by writing 8-bit data (with bit 0 as 0) to the TPnOPT0 register.

6.4.8 Timer output operations

The following table shows the operations and output levels of the TOPn0 and TOPn1 pins.

Table 6-6. Timer Output Control in Each Mode

Operation Mode	TOPn1 Pin	TOPn0 Pin
Interval timer mode	Square wave output	
External event count mode	-	
External trigger pulse output mode	External trigger pulse output	Square wave output
One-shot pulse output mode	One-shot pulse output	
PWM output mode	PWM output	
Free-running timer mode	Square wave output (only when compare function is used)	
Pulse width measurement mode	-	

Table 6-7. 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

6.5 Selector (V850ES/JE3-L only)

In the V850ES/JE3-L, the selector can be used to specify the capture trigger input for TMP as either a signal input to a port/timer alternate-function pin or peripheral I/O (TMP/UARTA) signal.

By using the selector, the following is possible:

- The TIP10 and TIP11 input signals of TMP1 can be selected as either the port/timer alternate-function pins (TIP10 and TIP11 pins) or the UARTA reception alternate-function pins (RXDA0 and RXDA1).
 → When the RXDA0 or RXDA1 signal of UART0 or UART1 is selected, the baud rate error in LIN reception transfer of UARTA can be calculated.

- Cautions**
1. When using the selector, set the capture trigger input of TMP before connecting the timer.
 2. When setting the selector, first disable the peripheral I/O to be connected (TMP or UARTA).

The capture input for the selector is specified by the following register.

(1) Selector operation control register 0 (SELCNT0)

The SELCNT0 register is an 8-bit register that selects the capture trigger for TMP1.

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

Reset sets this register to 00H.

After reset: 00H	R/W	Address: FFFFF308H						
SELCNT0	7	6	5	<4>	<3>	2	1	0
	0	0	0	ISEL4	ISEL3	0	0	0

ISEL4	Selection of TIP11 input signal (TMP1)
0	TIP11 pin input
1	RXDA1 pin input

ISEL3	Selection of TIP10 input signal (TMP1)
0	TIP10 pin input
1	RXDA0 pin input

Cautions

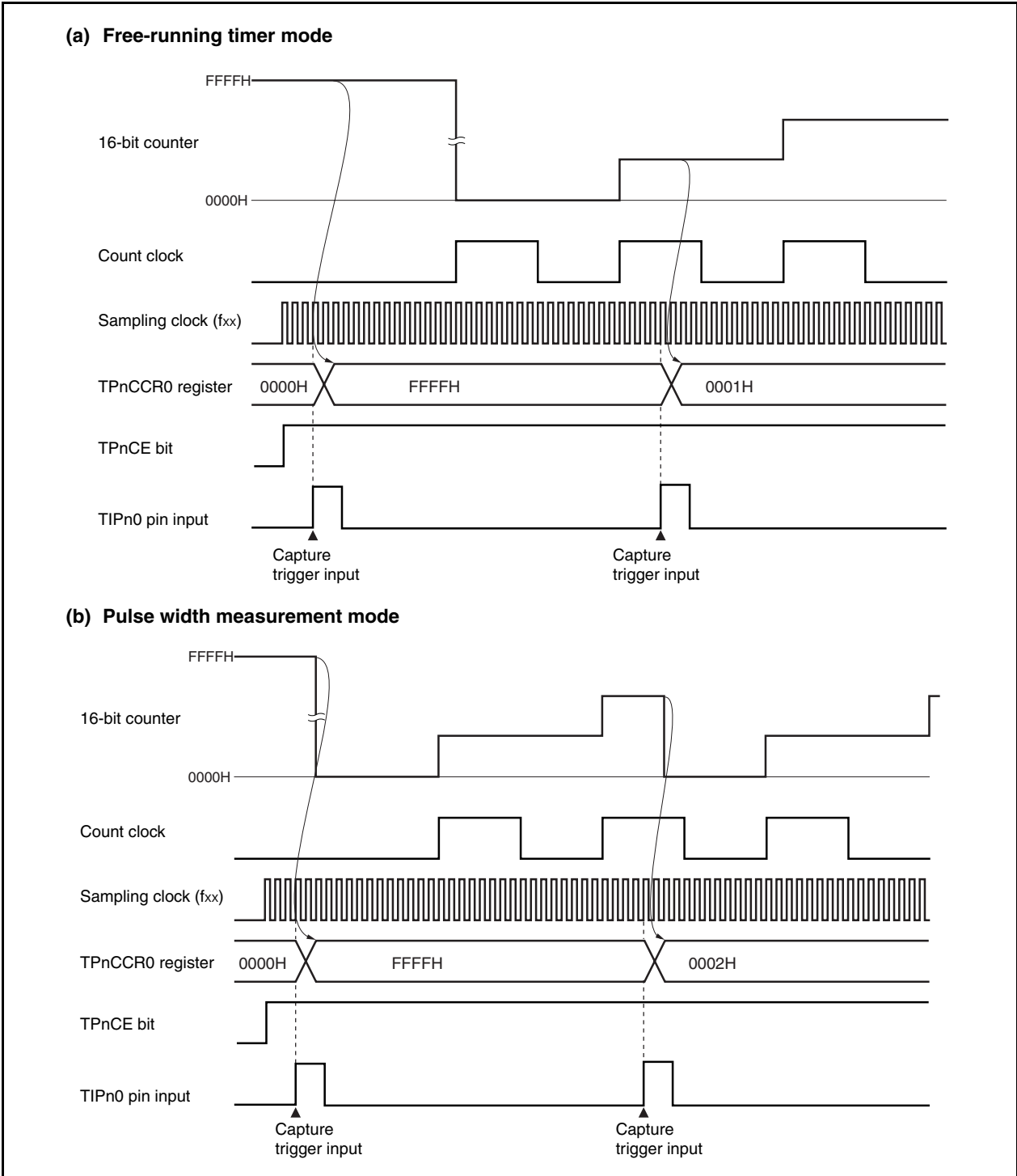
1. When setting the ISEL3 and ISEL4 bits to 1, set the corresponding pin to the capture input mode.
2. Be sure to clear bits 7 to 5 and 2 to 0 to "0".

6.6 Cautions

(1) Capture operation

When the capture operation is used and $f_{xx}/8$, $f_{xx}/16$, $f_{xx}/32$, $f_{xx}/64$, $f_{xx}/128$, $f_{xx}/256$, or $f_{xx}/512$ is selected as the count clock, FFFFH, not 0000H, may be captured in the TPnCCR0 and TPnCCR1 registers, or the capture operation may not be performed at all (the capture interrupt does not occur) if the capture trigger is input immediately after the TPnCE bit is set to 1.

This also occurs during the period in which no external event counts are input while the capture operation is being used and an external event count input is being used as the count clock.



CHAPTER 7 16-BIT TIMER/EVENT COUNTER Q (TMQ)

Timer Q (TMQ) is a 16-bit timer/event counter.

The V850ES/JC3-L, V850ES/JE3-L incorporate one TMQ timer/counter, TMQ0.

7.1 Functions

TMQ0 has the following features:

- (1) Interval timer
TMQ0 generates an interrupt at a preset interval and can output a square wave.
- (2) External event counter
TMQ0 counts the number of externally input signal pulses.
- (3) External trigger pulse output
TMQ0 starts counting and outputs a pulse when the specified external signal is input.
- (4) One-shot pulse output
TMQ0 outputs a one-shot pulse with an output width that can be freely specified.
- (5) PWM output
TMQ0 outputs a pulse with a constant cycle whose active width can be changed.
The pulse duty can also be changed freely even while the timer is operating.
- (6) Free-running timer
The 16-bit counter increments from 0000H to FFFFH and then resets.
- (7) Pulse width measurement
TMQ0 can be used to measure the pulses of a signal input externally.

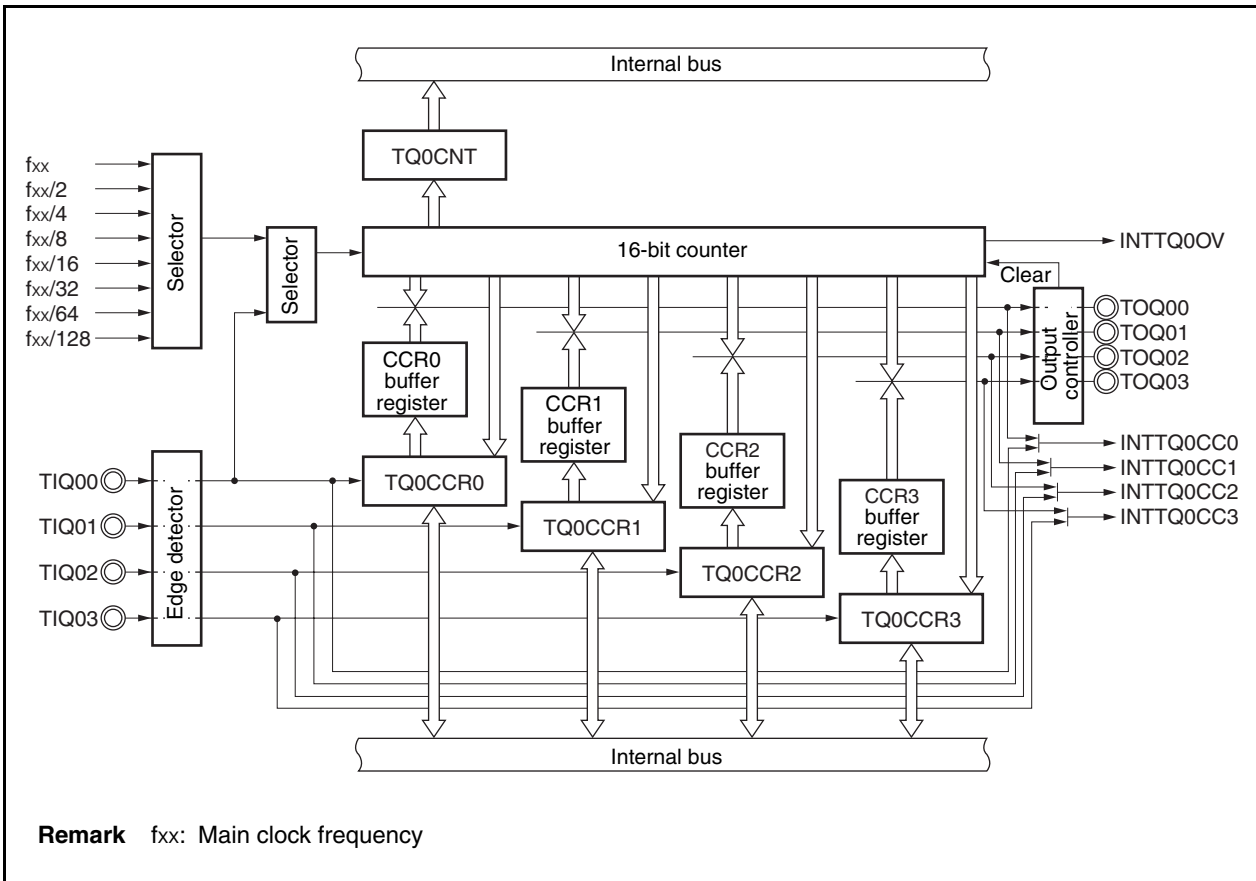
7.2 Configuration

TMQ0 includes the following hardware.

Table 7-1. Configuration of TMQ0

Item	Configuration
Registers	16-bit counter TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3) TMQ0 counter read buffer register (TQ0CNT) CCR0 to CCR3 buffer registers TMQ0 control registers 0, 1 (TQ0CTL0, TQ0CTL1) TMQ0 I/O control registers 0 to 2 (TQ0IOC0 to TQ0IOC2) TMQ0 option register 0 (TQ0OPT0)
Timer inputs	4 (TIQ00 to TIQ03 pins)
Timer outputs	4 (TOQ00 to TOQ03 pins)

Figure 7-1. Block Diagram of TMQ0



(1) 16-bit counter

This is a 16-bit counter that counts internal clocks and external events.

This counter can be read by using the TQ0CNT register.

When the TQ0CTL0.TQ0CE bit is 0 and the counter is stopped, the counter value is FFFFH. If the TQ0CNT register is read at this time, 0000H is read.

Reset sets the TQ0CE bit to 0, stopping the counter, and setting its value to FFFFH.

(2) CCR0 buffer register

This is a 16-bit compare register that compares the 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. If the value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTQ0CC0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is cleared to 0000H after reset because the TQ0CCR0 register is cleared to 0000H.

(3) CCR1 buffer register

This is a 16-bit compare register that compares the 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. If the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTQ0CC1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is cleared to 0000H after reset because the TQ0CCR1 register is cleared to 0000H.

(4) CCR2 buffer register

This is a 16-bit compare register that compares the 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. If the count value of the 16-bit counter matches the value of the CCR2 buffer register, a compare match interrupt request signal (INTTQ0CC2) is generated.

The CCR2 buffer register cannot be read or written directly.

The CCR2 buffer register is cleared to 0000H after reset because the TQ0CCR2 register is cleared to 0000H.

(5) CCR3 buffer register

This is a 16-bit compare register that compares the 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. If the count value of the 16-bit counter matches the value of the CCR3 buffer register, a compare match interrupt request signal (INTTQ0CC3) is generated.

The CCR3 buffer register cannot be read or written directly.

The CCR3 buffer register is cleared to 0000H after reset because the TQ0CCR3 register is cleared to 0000H.

(6) Edge detector

This circuit detects the valid edges input to the TIQ00 to TIQ03 pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TQ0IOC1 and TQ0IOC2 registers.

(7) Output controller

This circuit controls the output of the TOQ00 to TOQ03 pins. The output controller is controlled by the TQ0IOC0 register.

(8) Selector

The selector selects the count clock for the 16-bit counter. One of eight internal clocks or the input of an external event can be selected as the count clock.

7.2.1 Pins used by TMQ0

The input and output pins used by TMQ0 are shown in Table 7-2 below. When using these pins for TMQ0, first set them to port mode. For details, see **Table 4-17 Settings When Pins Are Used for Alternate Functions**.

Table 7-2. Pins Used by TMQ0

Pin No.			Port	TMQ0 Input	TMQ0 Output	Alternate Function
JC3L (40)	JC3L (48)	JE3L				
24	28	38	P53	TIQ00 ^{Note}	TOQ00	SIB2/KR3/RTP03/DDO
21	25	35	P50	TIQ01	TOQ01	KR0/RTP00
22	26	36	P51	TIQ02	TOQ02	KR1/RTP01
23	23	37	P52	TIQ03	TOQ03	KR2/RTP02/DDI

Note The TIQ00 pin functions as a capture trigger input, as an external event input, and as an external trigger input.

7.2.2 Interrupts

The following five types of interrupt signals are used by TMQ0:

(1) INTTQ0CC0

This signal is generated when the value of the 16-bit counter matches the value of the CCR0 buffer register, or when a capture signal is input from the TIQ00 pin.

(2) INTTQ0CC1

This signal is generated when the value of the 16-bit counter matches the value of the CCR1 buffer register, or when a capture signal is input from the TIQ01 pin.

(3) INTTQ0CC2

This signal is generated when the value of the 16-bit counter matches the value of the CCR2 buffer register, or when a capture signal is input from the TIQ02 pin.

(4) INTTQ0CC3

This signal is generated when the value of the 16-bit counter matches the value of the CCR3 buffer register, or when a capture signal is input from the TIQ03 pin.

(5) INTTQ0OV

This signal is generated when the 16-bit counter overflows after incrementing up to FFFFH.

7.3 Registers

The registers that control TMQ0 are as follows:

- TMQ0 control register 0 (TQ0CTL0)
- TMQ0 control register 1 (TQ0CTL1)
- TMQ0 I/O control register 0 (TQ0IOC0)
- TMQ0 I/O control register 1 (TQ0IOC1)
- TMQ0 I/O control register 2 (TQ0IOC2)
- TMQ0 option register 0 (TQ0OPT0)
- TMQ0 capture/compare register 0 (TQ0CCR0)
- TMQ0 capture/compare register 1 (TQ0CCR1)
- TMQ0 capture/compare register 2 (TQ0CCR2)
- TMQ0 capture/compare register 3 (TQ0CCR3)
- TMQ0 counter read buffer register (TQ0CNT)

Remark When using the functions of the TIQ00 to TIQ03 and TOQ00 to TOQ03 pins, see **Table 4-17 Settings When Pins Are Used for Alternate Functions.**

(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 sets this register to 00H.

The same value can always be written to the TQ0CTL0 register by software.

After reset: 00H		R/W	Address: FFFFF540H					
	<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. Operating clock supply stopped. (TMQ0 reset asynchronously ^{Note.})						
1		TMQ0 operation enabled. Operating clock supply started. (TMQ0 operation started.)						
TQ0CKS2		TQ0CKS1	TQ0CKS0	Internal count clock selection				
0		0	0	f _{xx}				
0		0	1	f _{xx} /2				
0		1	0	f _{xx} /4				
0		1	1	f _{xx} /8				
1		0	0	f _{xx} /16				
1		0	1	f _{xx} /32				
1		1	0	f _{xx} /64				
1		1	1	f _{xx} /128				

Note The TQ0OPT0.TQ0OVF bit and 16-bit counter are reset at the same time. In addition, the timer output pins (TOQ00 to TOQ03 pins) are reset to the status set by the TQ0IOC0 register when the 16-bit counter is reset.

Cautions

1. Set the TQ0CKS2 to TQ0CKS0 bits when the TQ0CE bit = 0. The TQ0CKS2 to TQ0CKS0 bits can be set at the same time as changing the value of the TQ0CE bit from 0 to 1.
2. Be sure to clear bits 3 to 6 to "0".

Remark f_{xx}: Main clock frequency

(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 sets this register to 00H.

After reset: 00H		R/W	Address: FFFFF541H						
		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. <ul style="list-style-type: none"> • 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. 							
TQ0EEE		Count clock selection							
0		Disable operation with external event count input. (Perform counting using the internal count clock selected by the TQ0CLT0.TQ0CK0 to TQ0CK2 bits.)							
1		Enable operation with external event count input. (Perform counting at the valid edge of the external event count input signal.)							
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	Setting prohibited						

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 after stopping the timer (by setting the TQ0CTL0.TQ0CE bit to 0). (However, if the same value is being written, this can be done while the TQ0CE bit is 1.) The operation is not guaranteed if the TQ0EEE and TQ0MD2 to TQ0MD0 bits are rewritten while the TQ0CE bit is 1. If the TQ0EEE and TQ0MD2 to TQ0MD0 bits were mistakenly rewritten while the TQ0CE bit was 1, clear the TQ0CE bit to 0 and then write the bits again.
4. Be sure to clear 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 (TOQ00 to TOQ03 pins).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF542H

	7	<6>	5	<4>	3	<2>	1	<0>
TQ0IOC0	TQ0OL3	TQ0OE3	TQ0OL2	TQ0OE2	TQ0OL1	TQ0OE1	TQ0OL0	TQ0OE0

TQ0OLm	TOQ0m pin output level setting (m = 0 to 3) ^{Note}
0	TOQ0m pin starts output at high level
1	TOQ0m pin starts output at low level

TQ0OE _m	TOQ0m pin output setting (m = 0 to 3)
0	Timer output disabled • When TQ0OLm bit = 0: Low level is output from the TOQ0m pin • When TQ0OLm bit = 1: High level is output from the TOQ0m pin
1	Timer output enabled (a pulse is output from the TOQ0m pin)

Note The output level of the timer output pin (TOQ0m) specified by the TQ0OLm bit is shown below.

• When TQ0OLm bit = 0

• When TQ0OLm bit = 1

Cautions

1. Rewrite the TQ0OLm and TQ0OE_m 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. Even if the TQ0OLm bit is manipulated when the TQ0CE and TQ0OE_m bits are 0, the TOQ0m pin output level varies.

Remark m = 0 to 3

(4) TMQ0 I/O control register 1 (TQ0IOC1)

The TQ0IOC1 register is an 8-bit register that controls the specification of the valid edge of the capture trigger input signals (TIQ00 to TIQ03 pins).

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

Reset sets this register to 00H.

After reset: 00H		R/W	Address: FFFF543H							
			7	6	5	4	3	2	1	0
TQ0IOC1			TQ0IS7	TQ0IS6	TQ0IS5	TQ0IS4	TQ0IS3	TQ0IS2	TQ0IS1	TQ0IS0
	TQ0IS7	TQ0IS6	Capture trigger input signal (TIQ03 pin) valid edge setting							
	0	0	No edge detection (capture operation invalid)							
	0	1	Detection of rising edge							
	1	0	Detection of falling edge							
	1	1	Detection of both edges							
	TQ0IS5	TQ0IS4	Capture trigger input signal (TIQ02 pin) valid edge 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 (TIQ01 pin) valid edge setting							
	0	0	No edge detection (capture operation invalid)							
	0	1	Detection of rising edge							
	1	0	Detection of falling edge							
	1	1	Detection of both edges							
	TQ0IS1	TQ0IS0	Capture trigger input signal (TIQ00 pin) valid edge setting							
	0	0	No edge detection (capture operation invalid)							
	0	1	Detection of rising edge							
	1	0	Detection of falling edge							
	1	1	Detection of both edges							

- Cautions**
1. Rewrite the TQ0IS7 to TQ0IS0 bits when the TQ0CTL0.TQ0CE bit = 0. (The same value can be written when the TQ0CE bit = 1.) If rewriting was mistakenly performed, clear the TQ0CE bit to 0 and then set the bits again.
 2. The TQ0IS7 to TQ0IS0 bits are valid only in the free-running timer mode 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 specification of the valid edge of the external event count input signal (TIQ00 pin) and external trigger input signal (TIQ00 pin).

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

Reset sets this register to 00H.

After reset: 00H	R/W	Address: FFFF544H								
			7	6	5	4	3	2	1	0
TQ0IOC2			0	0	0	0	TQ0EES1	TQ0EES0	TQ0ETS1	TQ0ETS0
			TQ0EES1	TQ0EES0	External event count input signal (TIQ00 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 (TIQ00 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 when the external trigger pulse output mode (TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 bits = 010) or the one-shot pulse output mode (TQ0CTL1.TQ0MD2 to TQ0CTL1.TQ0MD0 = 011) is set.

(6) TMQ0 option register 0 (TQ0OPT0)

The TQ0OPT0 register is an 8-bit register that specifies the capture/compare operation and indicates the detection of an overflow.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF545H								
	7	6	5	4	3	2	1	<0>
TQ0OPT0	TQ0CCS3	TQ0CCS2	TQ0CCS1	TQ0CCS0	0	0	0	TQ0OVF

TQ0CCSm	TQ0CCRm register capture/compare selection
0	Compare register selected
1	Capture register selected
The TQ0CCSm bit setting is valid only in the free-running timer mode.	

TQ0OVF	TMQ0 overflow detection
Set (1)	Overflow occurred
Reset (0)	TQ0OVF bit 0 written or TQ0CTL0.TQ0CE bit = 0
<ul style="list-style-type: none"> • The TQ0OVF bit is set 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 interrupt request signal (INTTQ0OV) is generated at the same time that the TQ0OVF bit is set to 1. The INTTQ0OV 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 even when the TQ0OVF bit or the TQ0OPT0 register are read when the TQ0OVF bit = 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. 	

Cautions

1. Rewrite the TQ0CCS3 to TQ0CCS0 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. Be sure to clear bits 1 to 3 to "0".

Remark m = 0 to 3

(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 selected as a capture register or a compare register only in the free-running timer mode, according to the setting of the TQ0OPT0.TQ0CCS0 bit. In any other mode, this register can be used only as a compare register.

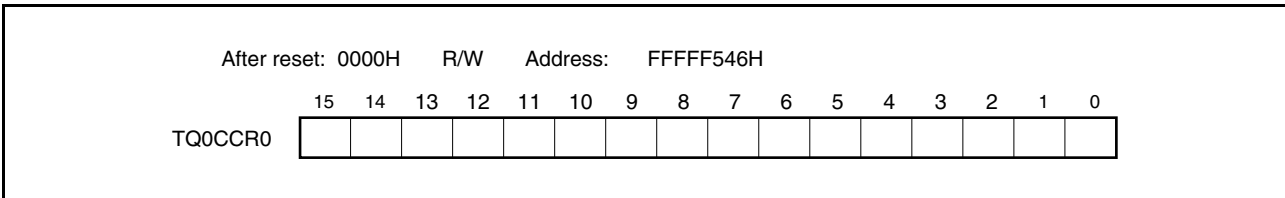
The TQ0CCR0 register can be read or written during operation.

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

Reset sets this register to 0000H.

Caution Accessing the TQ0CCR0 register is prohibited in the following statuses. Moreover, if the system is in the wait status, the only way to cancel the wait status is to execute a reset. For details, see 3.4.9 (1) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates on the subclock and main clock oscillation is stopped
- When the CPU operates on the internal oscillator clock



(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 (INTTQ0CC0) is generated. If TOQ00 pin output is enabled at this time, the output of the TOQ00 pin is inverted (For details, see the descriptions of each operating mode.).

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.

(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 (TIQ00 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TQ0CCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIQ00 pin) is detected.

Even if the capture operation and reading the TQ0CCR0 register conflict, the correct value of the TQ0CCR0 register can be read.

The following table shows the functions of the capture/compare register in each operation mode, and how to write data to the compare register.

Table 7-3. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	—

Remark For details about anytime write and batch write, see 7.4 (2) **Anytime write and batch write**.

(8) TMQ0 capture/compare register 1 (TQ0CCR1)

The TQ0CCR1 register can be used as a capture register or a compare register depending on the mode.

This register can be selected as a capture register or a compare register only in the free-running timer mode, according to 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.

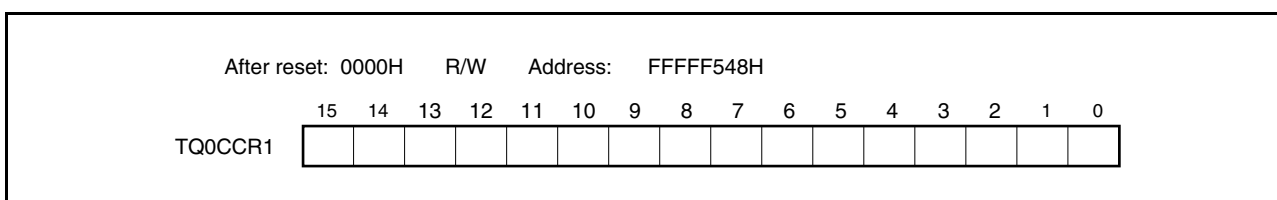
The TQ0CCR1 register can be read or written during operation.

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

Reset sets this register to 0000H.

Caution Accessing the TQ0CCR1 register is prohibited in the following statuses. Moreover, if the system is in the wait status, the only way to cancel the wait status is to execute a reset. For details, see 3.4.9 (1) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates on the subclock and main clock oscillation is stopped
- When the CPU operates on the internal oscillator clock

**(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 (INTTQ0CC1) is generated. If TOQ01 pin output is enabled at this time, the output of the TOQ01 pin is inverted (For details, see the descriptions of each operating mode.).

(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 (TIQ01 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TQ0CCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIQ01 pin) is detected.

Even if the capture operation and reading the TQ0CCR1 register conflict, the correct value of the TQ0CCR1 register can be read.

The following table shows the functions of the capture/compare register in each operation mode, and how to write data to the compare register.

Table 7-4. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	—

Remark For details about anytime write and batch write, see 7.4 (2) **Anytime write and batch write**.

(9) TMQ0 capture/compare register 2 (TQ0CCR2)

The TQ0CCR2 register can be used as a capture register or a compare register depending on the mode.

This register can be selected as a capture register or a compare register only in the free-running timer mode, according to 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.

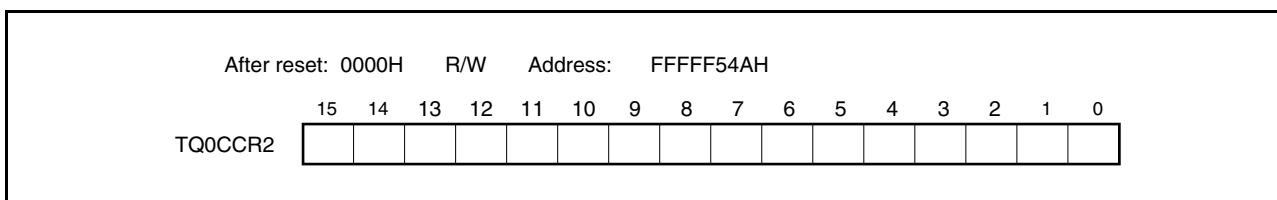
The TQ0CCR2 register can be read or written during operation.

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

Reset sets this register to 0000H.

Caution Accessing the TQ0CCR2 register is prohibited in the following statuses. Moreover, if the system is in the wait status, the only way to cancel the wait status is to execute a reset. For details, see 3.4.9 (1) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates on the subclock and main clock oscillation is stopped
- When the CPU operates on the internal oscillator clock

**(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 (INTTQ0CC2) is generated. If TOQ02 pin output is enabled at this time, the output of the TOQ02 pin is inverted (For details, see the descriptions of each operating mode.).

(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 (TIQ02 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TQ0CCR2 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIQ02 pin) is detected.

Even if the capture operation and reading the TQ0CCR2 register conflict, the correct value of the TQ0CCR2 register can be read.

The following table shows the functions of the capture/compare register in each operation mode, and how to write data to the compare register.

Table 7-5. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	—

Remark For details about anytime write and batch write, see 7.4 (2) **Anytime write and batch write**.

(10)TMQ0 capture/compare register 3 (TQ0CCR3)

The TQ0CCR3 register can be used as a capture register or a compare register depending on the mode.

This register can be selected as a capture register or a compare register only in the free-running timer mode, according to 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.

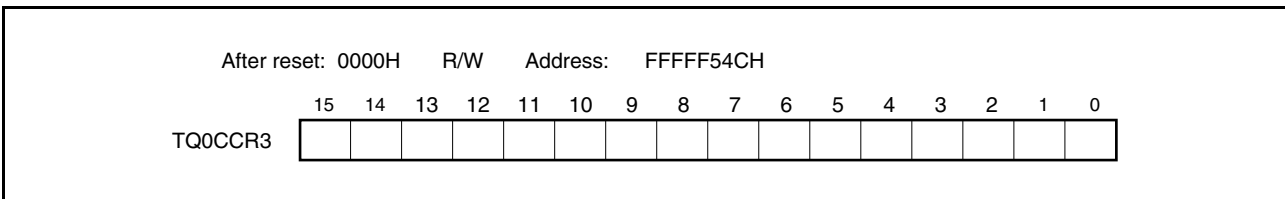
The TQ0CCR3 register can be read or written during operation.

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

Reset sets this register to 0000H.

Caution Accessing the TQ0CCR3 register is prohibited in the following statuses. Moreover, if the system is in the wait status, the only way to cancel the wait status is to execute a reset. For details, see 3.4.9 (1) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates on the subclock and main clock oscillation is stopped
- When the CPU operates on the internal oscillator clock



(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 (INTTQ0CC3) is generated. If TOQ03 pin output is enabled at this time, the output of the TOQ03 pin is inverted (For details, see the descriptions of each operating mode.).

(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 (TIQ03 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TQ0CCR3 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIQ03 pin) is detected.

Even if the capture operation and reading the TQ0CCR3 register conflict, the correct value of the TQ0CCR3 register can be read.

The following table shows the functions of the capture/compare register in each operation mode, and how to write data to the compare register.

Table 7-6. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write
One-shot pulse output	Compare register	Anytime write
PWM output	Compare register	Batch write
Free-running timer	Capture/compare register	Anytime write
Pulse width measurement	Capture register	—

Remark For details about anytime write and batch write, see 7.4 (2) **Anytime write and batch write**.

(11) TMQ0 counter read buffer register (TQ0CNT)

The TQ0CNT register is a read buffer register from which the value of the 16-bit counter can be read.

If this register is read when the TQ0CTL0.TQ0CE bit = 1, the count value of the 16-bit timer can be read.

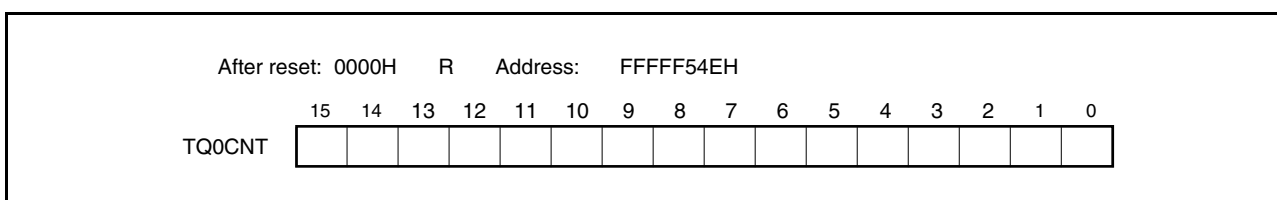
This register is read-only, in 16-bit units.

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.

Because the TQ0CE bit is cleared to 0, the value of the TQ0CNT register is cleared to 0000H after reset.

Caution Accessing the TQ0CNT register is prohibited in the following statuses. Moreover, if the system is in the wait status, the only way to cancel the wait status is to execute a reset. For details, see 3.4.9 (1) **Accessing specific on-chip peripheral I/O registers**.

- When the CPU is operating on the subclock and main clock oscillation is stopped
- When the CPU is operating on the internal clock



7.4 Operations

TMQ0 can execute the following operations:

Table 7-7. TMQ0 Operating Modes

Operating Mode	TQ0CTL1.TQ0EST Bit (Software Trigger Bit)	TIQ00 Pin (External Trigger Input)	Capture/Compare Register Setting	Compare Register Write	Count Clock
Interval timer mode	Invalid	Invalid	Compare only	Anytime write	Internal/external
External event count mode ^{Note 1}	Invalid	Invalid	Compare only	Anytime write	External
External trigger pulse output mode ^{Note 2}	Valid	Valid	Compare only	Batch write	Internal
One-shot pulse output mode ^{Note 2}	Valid	Valid	Compare only	Anytime write	Internal
PWM output mode	Invalid	Invalid	Compare only	Batch write	Internal/external
Free-running timer mode	Invalid	Invalid	Can be switched	Anytime write	Internal/external
Pulse width measurement mode ^{Note 2}	Invalid	Invalid	Capture only	Not applicable	Internal

- Notes**
1. When using the external event count mode, specify that the valid edge of the TIQ00 pin capture trigger input is not detected (by clearing the TQ0IOC1.TQ0IS1 and TQ0IOC1.TQ0IS0 bits to 0).
 2. When using the external trigger pulse output mode, one-shot pulse output mode, and pulse width measurement mode, select the internal clock as the count clock (by clearing the TQ0CTL1.TQ0EEE bit to 0).

(1) Basic counter operation

The basic operation of the 16-bit counter is described below. For more details, see the descriptions of each operating mode.

(a) Starting counting

TMQ0 starts counting from FFFFH in all operating modes, and increments as follows: FFFFH, 0000H, 0001H, 0002H, 0003H....

(b) Clearing TMQ0

TMQ0 is cleared to 0000H when its value matches the value of the compare register or when the value of TMQ0 is captured upon the input of a valid capture trigger signal.

Note that when TMQ0 increments from FFFFH to 0000H after it starts counting and immediately following an overflow, it does not mean that TMQ0 has been cleared. Consequently, the INTTQ0CCm interrupt is not generated in this case (m = 0 to 3).

(c) Overflow

TMQ0 overflows after it increments from FFFFH to 0000H in free-running timer mode and pulse width measurement mode. An overflow sets the TQ0OPT0.TQ0OVF bit to 1 and generates an interrupt request signal (INTTQ0OV). Note that INTTQ0OV will not be generated in the following cases:

- When TMQ0 has just started counting.
- When the compare value at which TMQ0 is cleared is specified as FFFFH.
- In pulse width measurement mode, when TMQ0 increments from FFFFH to 0000H after being cleared when its value of FFFFH was captured.

Caution After the INTTQ0OV overflow interrupt request signal occurs, be sure to confirm that the overflow flag (TQ0OVF) is set to 1.

(d) Reading TMQ0 while it is incrementing

TMQ0 can be read while it is incrementing by using the TQ0CNT register.

Specifically, the value of TMQ0 can be read by reading the TQ0CNT register while the TQ0CLT0.TQ0CE bit is 1. Note, however, that when the TQ0CLT0.TQ0CE bit is 0, the value of TMQ0 is always FFFFH and the value of the TQ0CNT register is always 0000H.

(2) Anytime write and batch write

The TQ0CCR0 to TQ0CCR3 registers can be written even while TMQ0 is operating (that is, while the TQ0CTL0.TQ0CE bit is 1), but the way the CCR0 to CCR3 buffer registers are written differs depending on the mode. The two writing methods are anytime write and batch write.

(a) Anytime write

This writing method is used to transfer data from the TQ0CCR0 to TQ0CCR3 registers to the CCR0 to CCR3 buffer registers any time while TMQ0 is operating.

Figure 7-2. Flowchart Showing Basic Anytime Write Operation

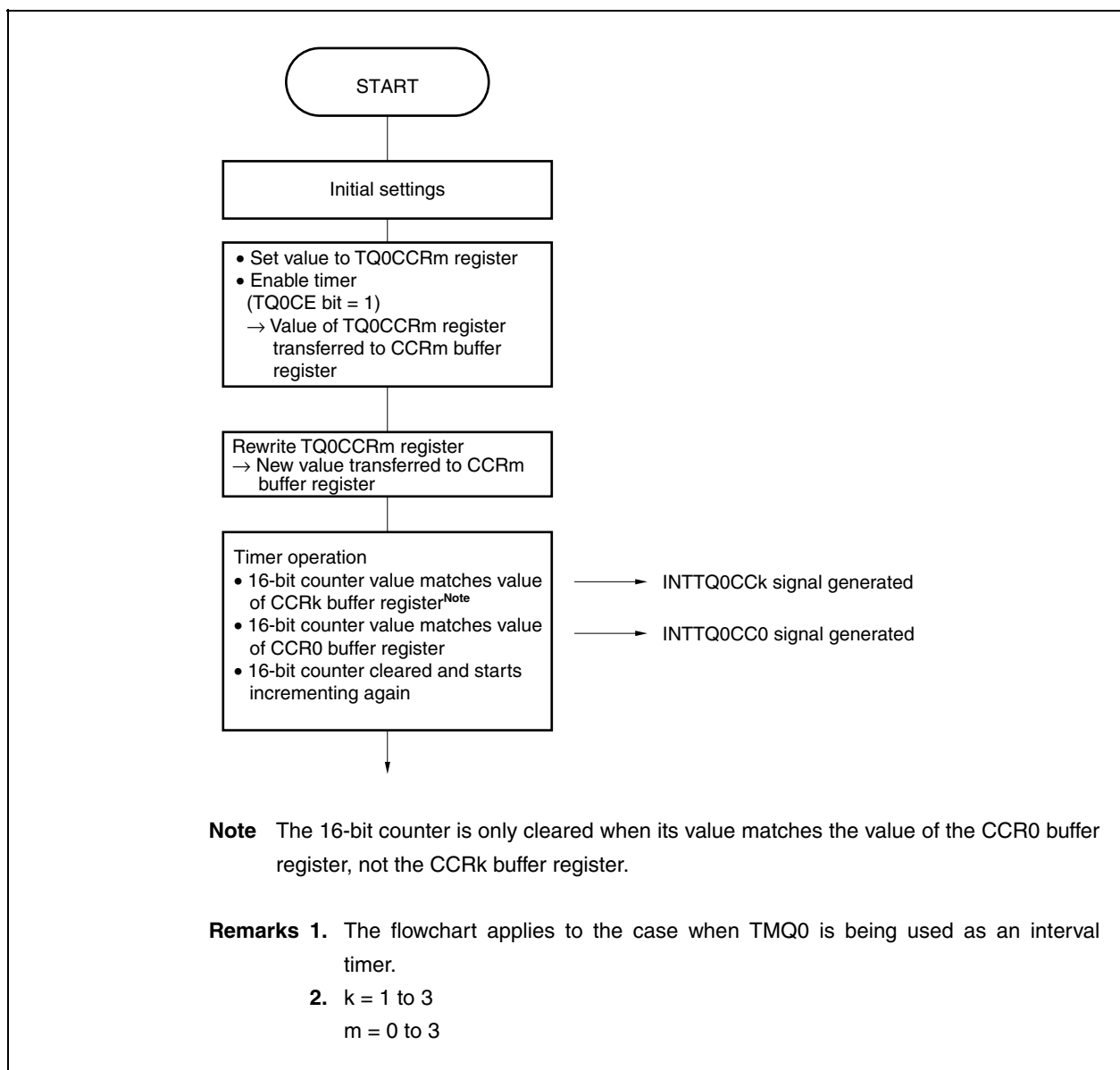
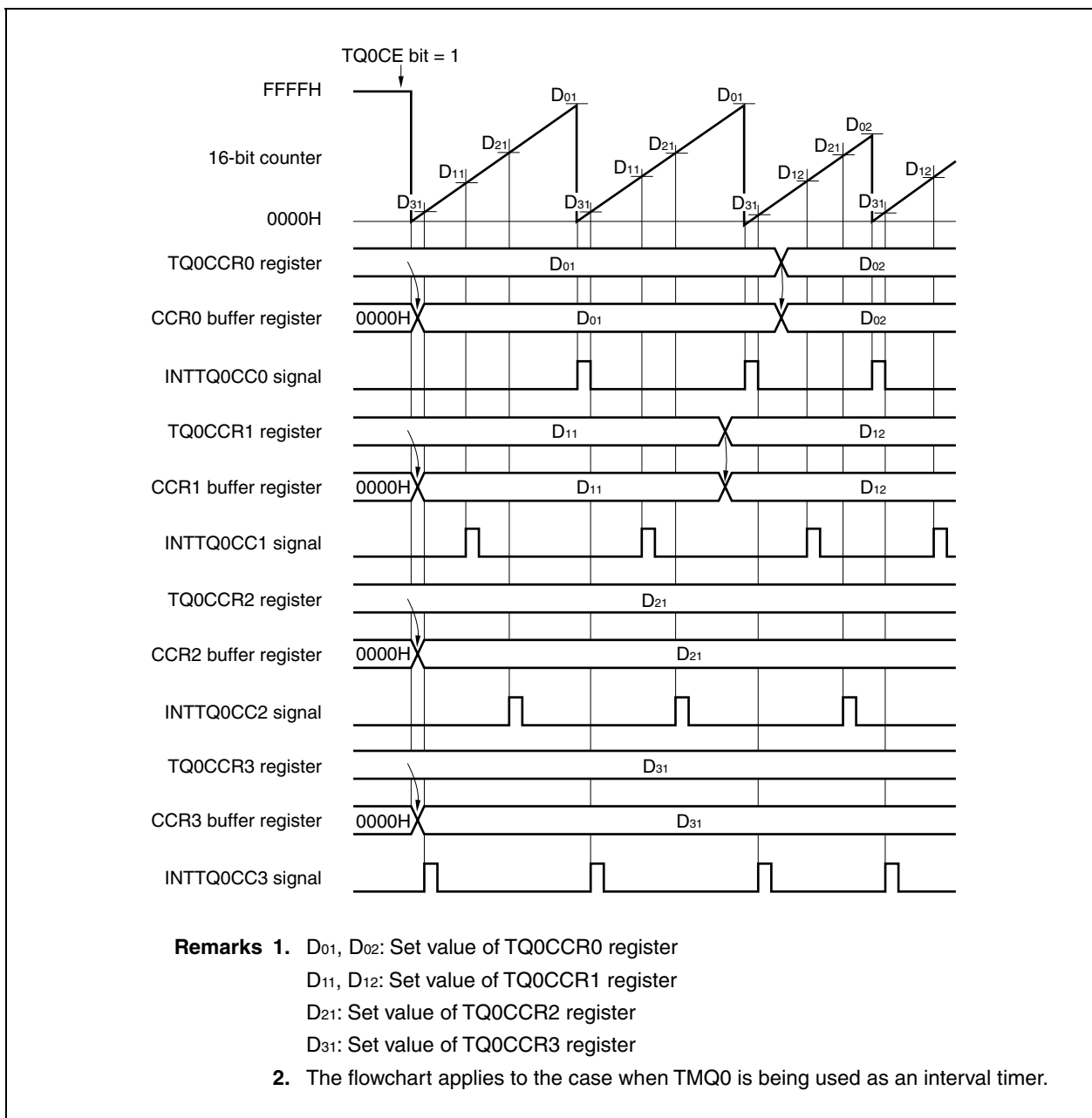


Figure 7-3. Anytime Write Timing



(b) Batch write

This writing method is used to transfer data from the TQ0CCR0 to TQ0CCR3 registers to the CCR0 to CCR3 buffer registers all at once while TMQ0 is operating. The data is transferred when the value of the 16-bit counter matches the value of the CCR0 buffer register. Transfer is enabled by writing to the TQ0CCR1 register. Whether transfer of the next data is enabled or not depends on whether the TQ0CCR1 register has been written.

To specify the value of the rewritten TQ0CCR0 to TQ0CCR3 registers as the 16-bit counter compare value (that is, the value to be transferred to the CCR0 to CCR3 buffer registers), the TQ0CCR0 register must be rewritten before the value of the 16-bit counter matches the value of the CCR0 buffer register, and then the TQ0CCR1 register must be written. The value of the TQ0CCR0 to TQ0CCR3 registers is then transferred to the CCR0 to CCR3 buffer registers when the value of the 16-bit counter matches the value of the CCR0 buffer register. Note that even if you wish to rewrite one of the TQ0CCR0, TQ0CCR2 and TQ0CCR3 register values, you must also write the same value to the TQ0CCR1 register (that is, the same value as the value already specified for the TQ0CCR1 register).

Figure 7-4. Flowchart Showing Basic Batch Write Operation

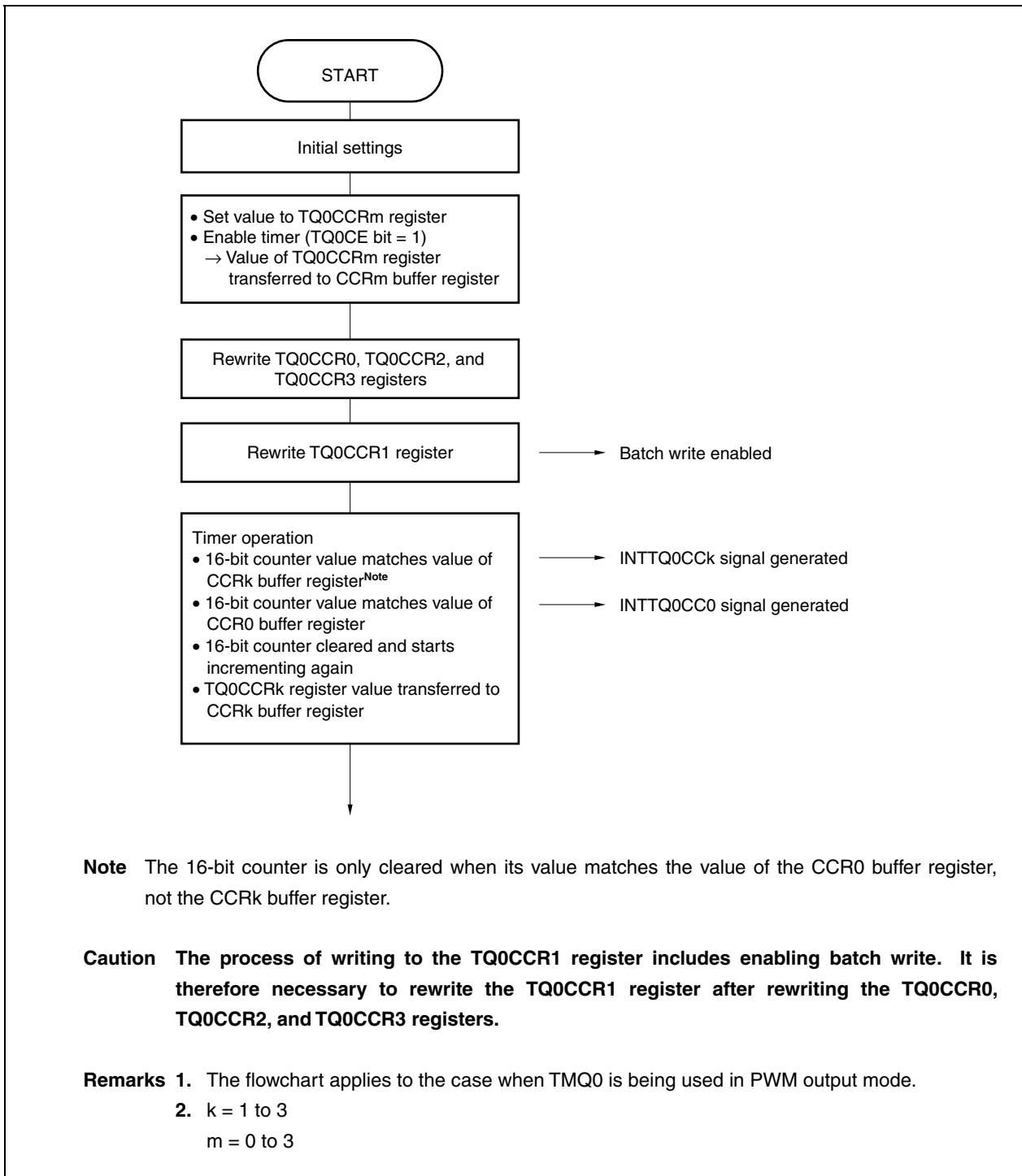
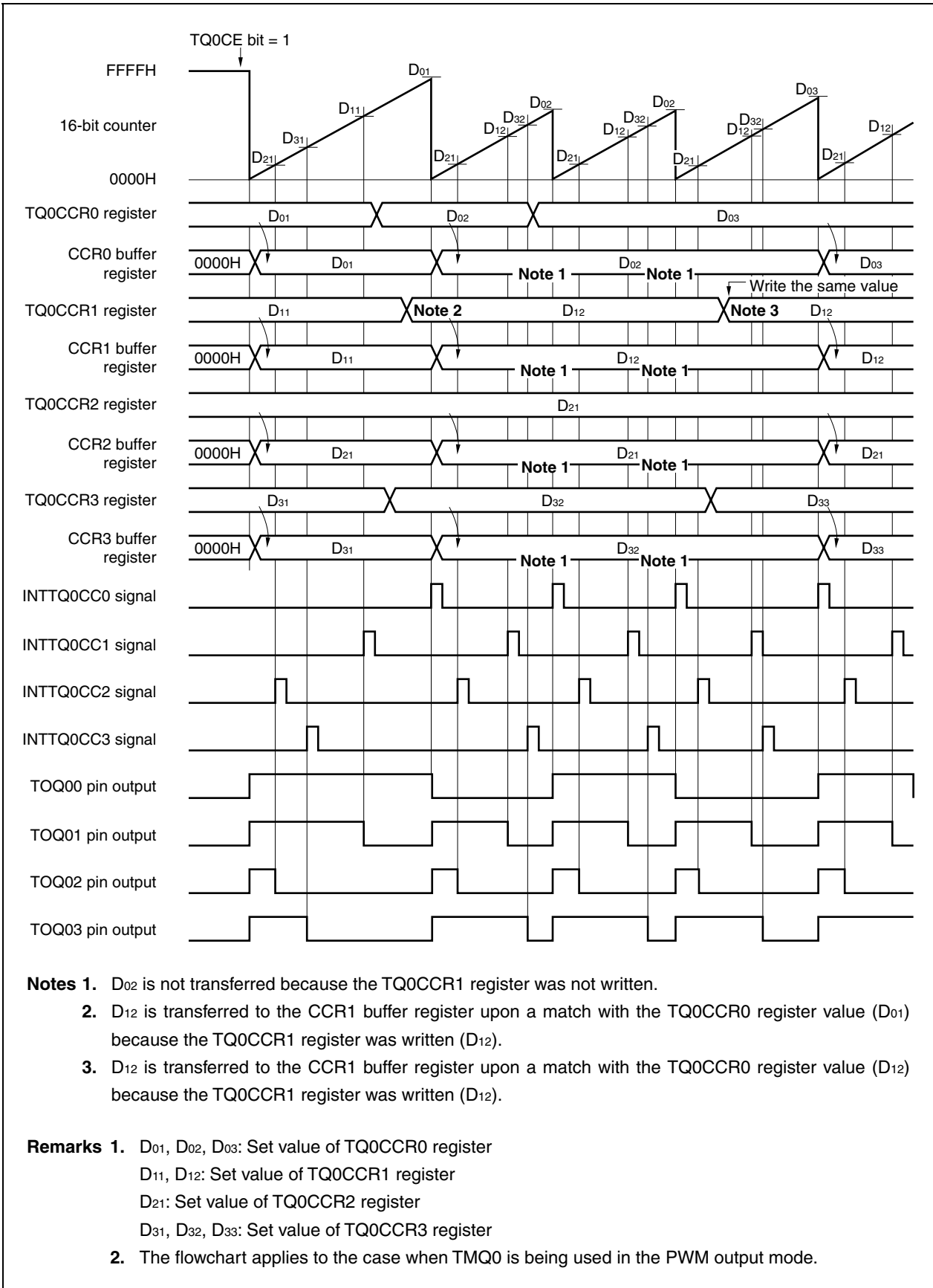


Figure 7-5. Batch Write Timing



7.4.1 Interval timer mode (TQ0MD2 to TQ0MD0 bits = 000)

In the interval timer mode, setting the TQ0CTL0.TQ0CE bit to 1 generates an interrupt request signal (INTTQ0CC0) at a specified interval. Setting the TQ0CE bit to 1 can also start the timer, which then outputs a square wave whose half cycle is equal to the interval from the TOQ00 pin.

Usually, the TQ0CCR1 to TQ0CCR3 registers are not used in the interval timer mode. Mask interrupts from these registers by setting the interrupt mask flags (TQ0CCMK1 to TQ0CCMK3).

- Remarks 1.** For how to set the TOQ00 pin, see **Table 7-2 Pins Used by TMQ0** and **Table 4-17 Settings When Pins Are Used for Alternate Functions**.
- 2.** For how to enable the INTTQ0CC0 interrupt signal, see **CHAPTER 19 INTERRUPT SERVICING/ EXCEPTION PROCESSING FUNCTION**.

Figure 7-6. Configuration of Interval Timer

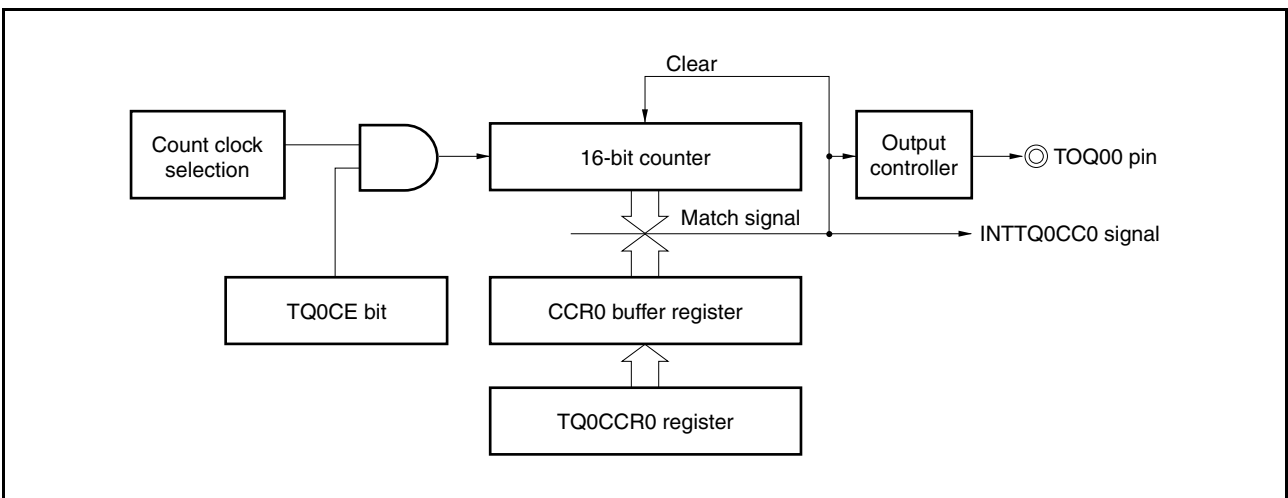
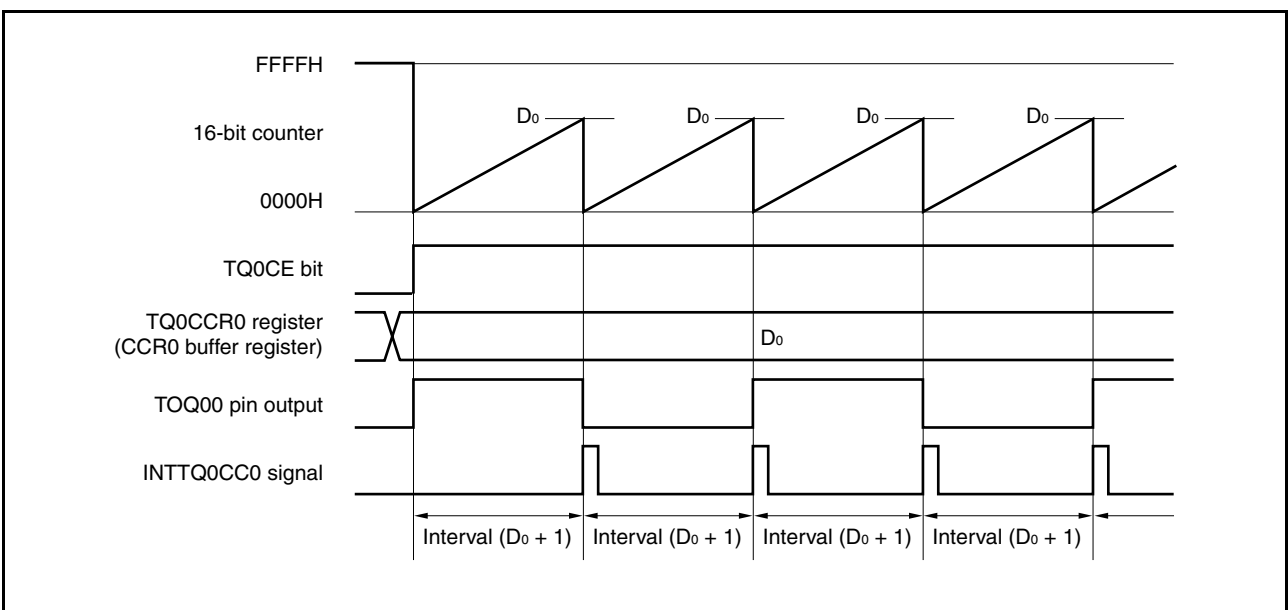


Figure 7-7. Basic Timing of Operation in Interval Timer Mode



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 incrementing. At this time, the output of the TOQ00 pin is inverted and 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, the 16-bit counter is cleared to 0000H, the output of the TOQ00 pin is inverted, and a compare match interrupt request signal (INTTQ0CC0) is generated.

The interval can be calculated by using the following expression:

$$\text{Interval} = (\text{Set value of TQ0CCR0 register} + 1) \times \text{Count clock cycle}$$

An example of the register settings when the interval timer mode is used is shown in the figure below.

Figure 7-8. Register Settings in Interval Timer Mode (1/3)

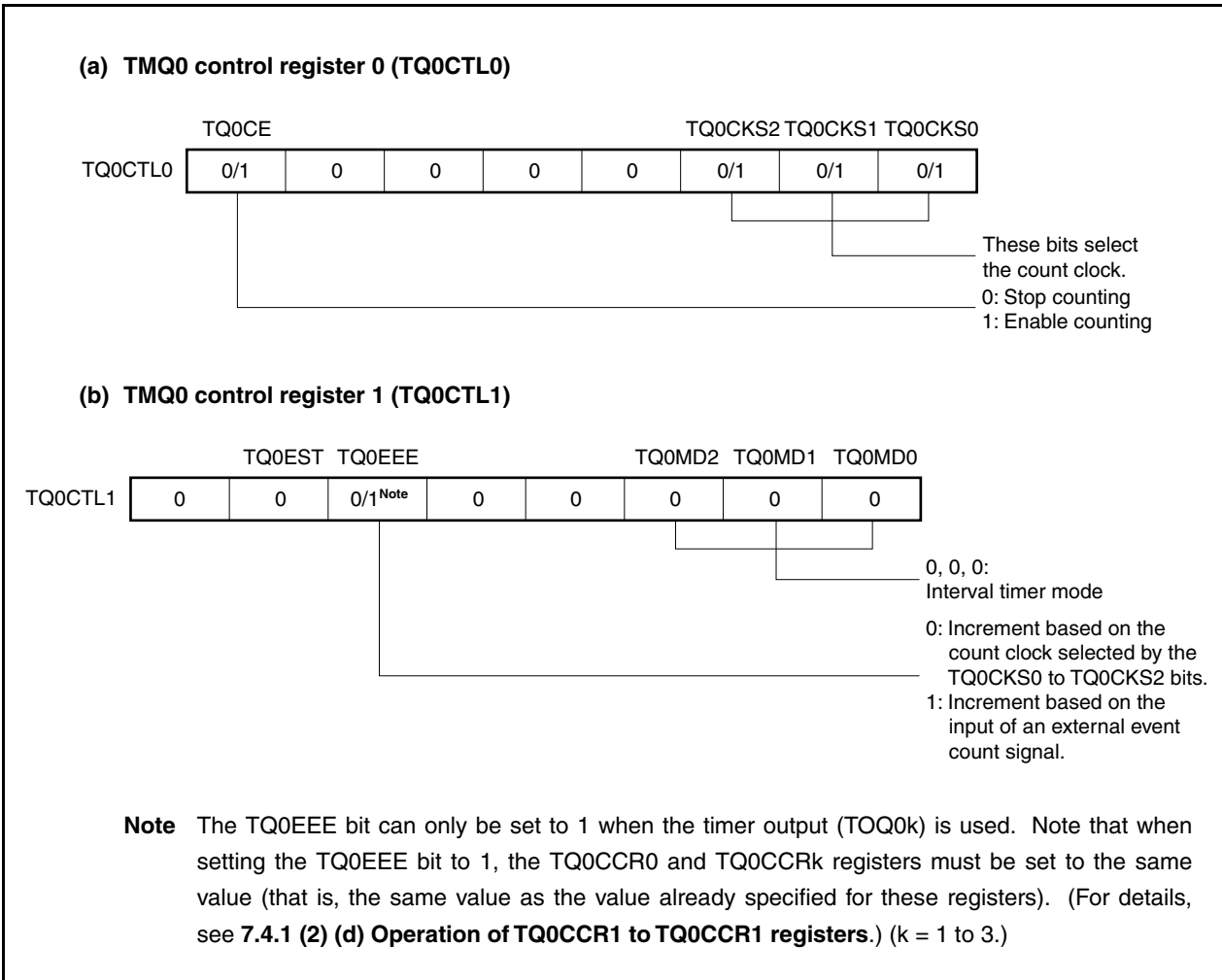


Figure 7-8. Register Settings in Interval Timer Mode (2/3)

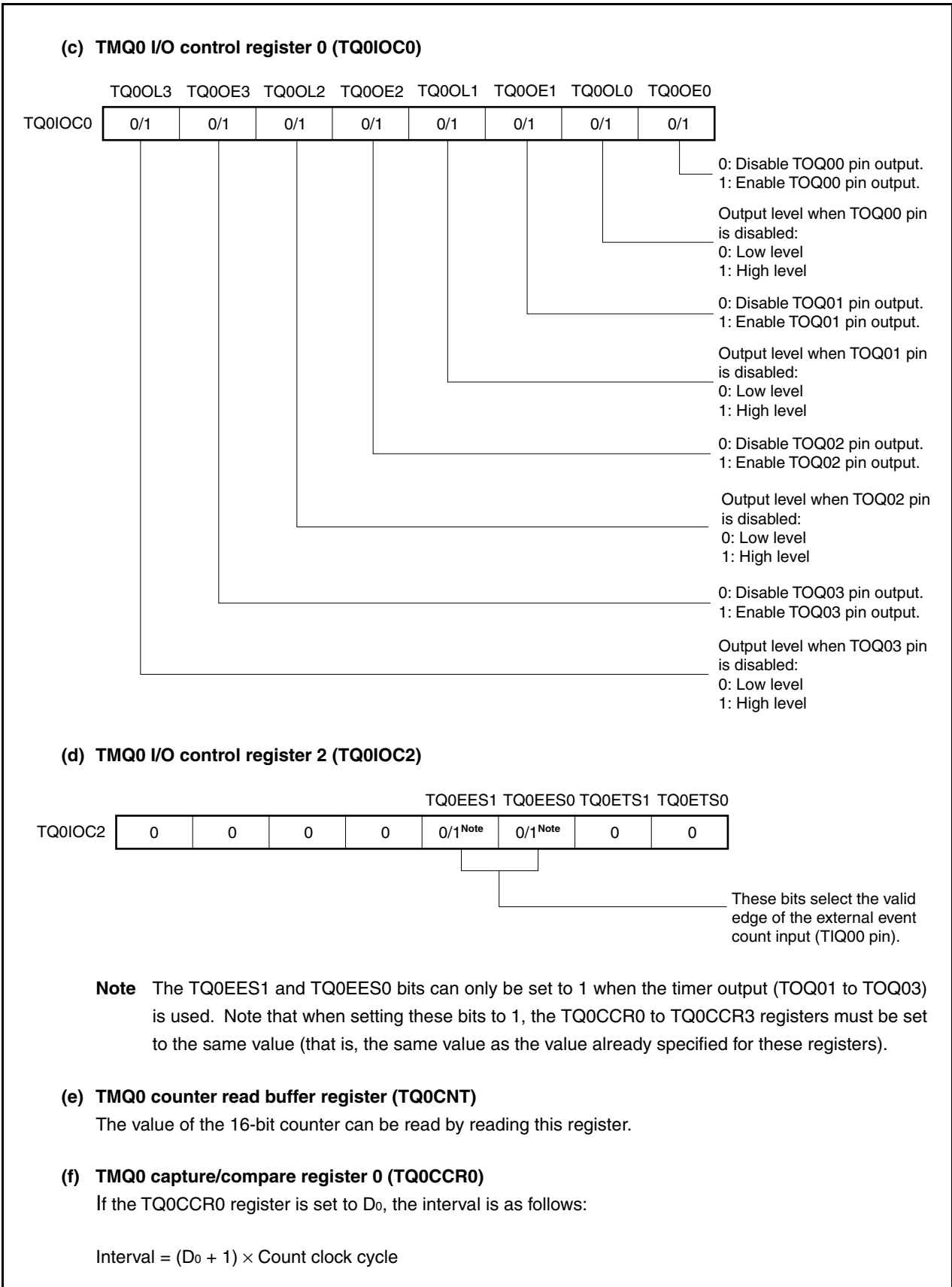


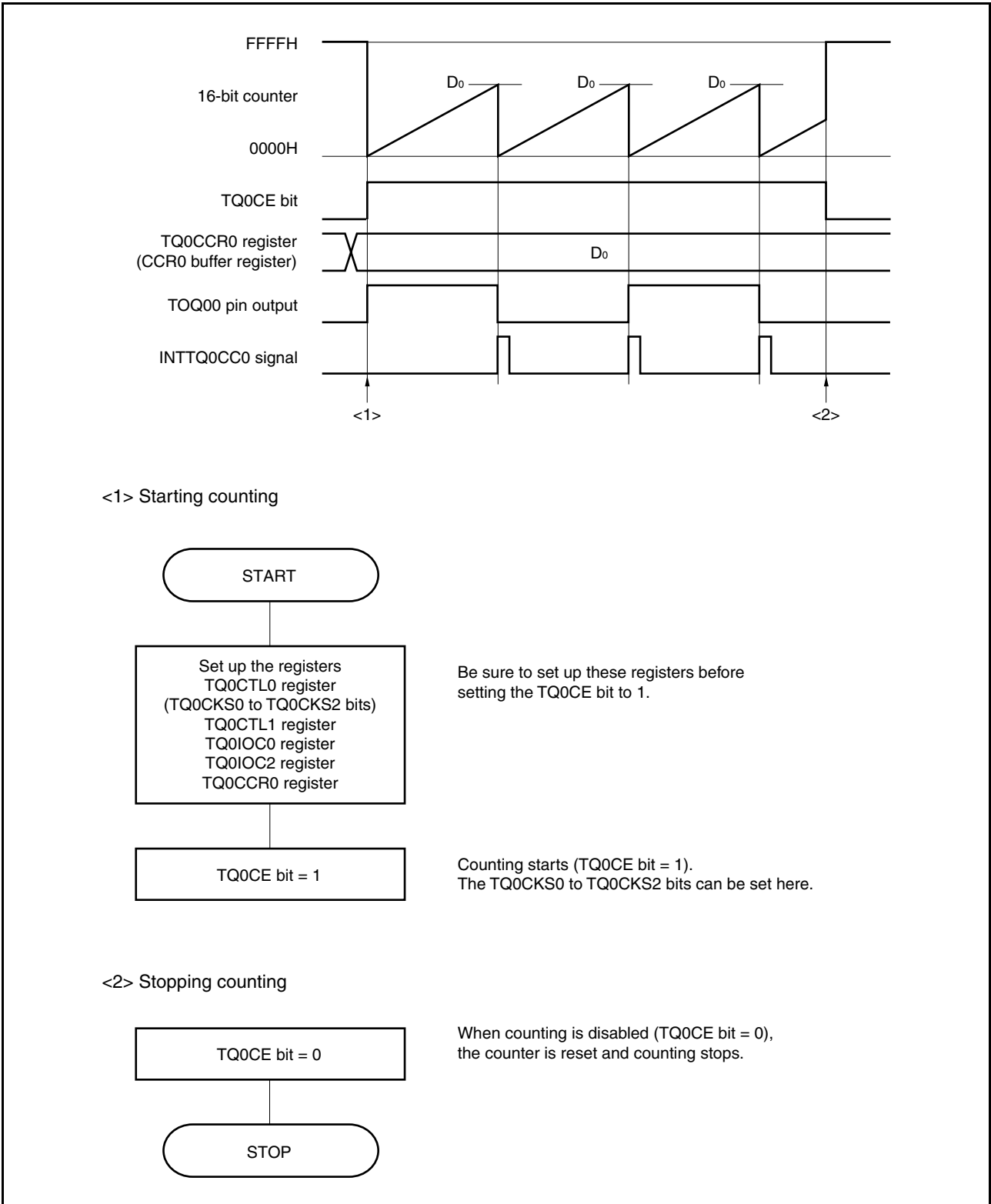
Figure 7-8. Register Settings in Interval Timer Mode (3/3)**(g) TMQ0 capture/compare registers 1 to 3 (TQ0CCR1 to TQ0CCR3)**

Usually, the TQ0CCR1 to TQ0CCR3 registers are not used in the interval timer mode. However, because the set values of the TQ0CCR1 to TQ0CCR3 registers are transferred to the CCR1 to CCR3 buffer registers and a compare match interrupt request signal (INTTQ0CC1 to INTTQ0CC3) is generated when the value of the 16-bit counter matches the value of the CCR1 to CCR3 buffer registers, interrupts from these registers must be masked by setting the interrupt mask flags (TQ0CCMK1 to TQ0CCMK3).

Remark TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the interval timer mode.

(1) Operations in interval timer mode

Figure 7-9. Timing and Processing of Operations in Interval Timer Mode



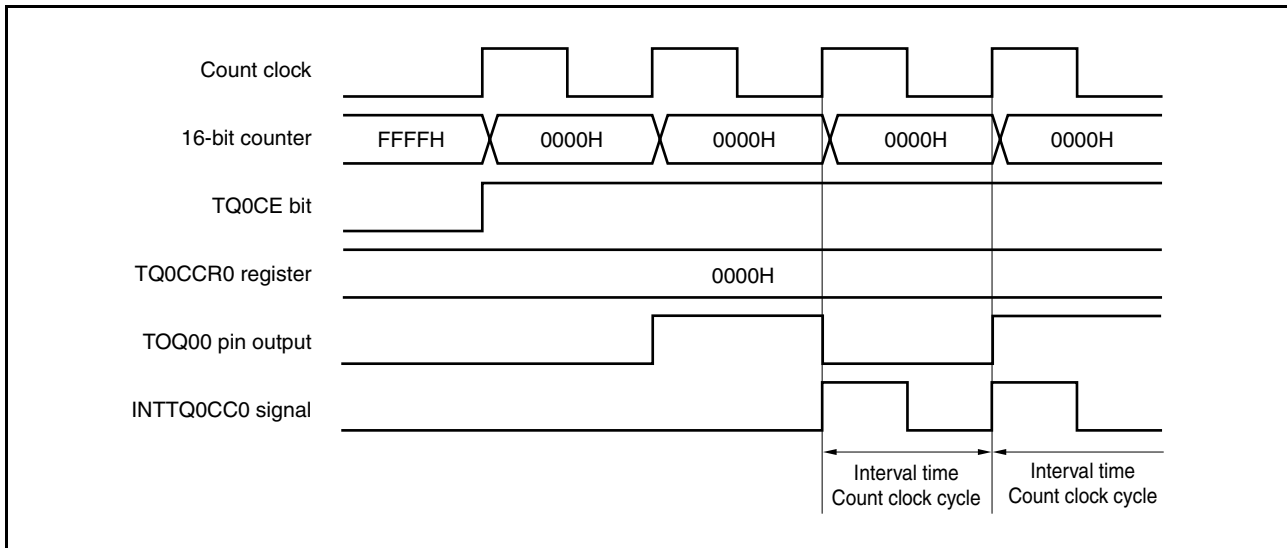
(2) Using interval timer mode

(a) Operation when TQ0CCR0 register is set to 0000H

When the TQ0CCR0 register is set to 0000H, the INTTQ0CC0 signal is generated each count clock cycle from the second clock cycle, and the output of the TOQ00 pin is inverted.

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

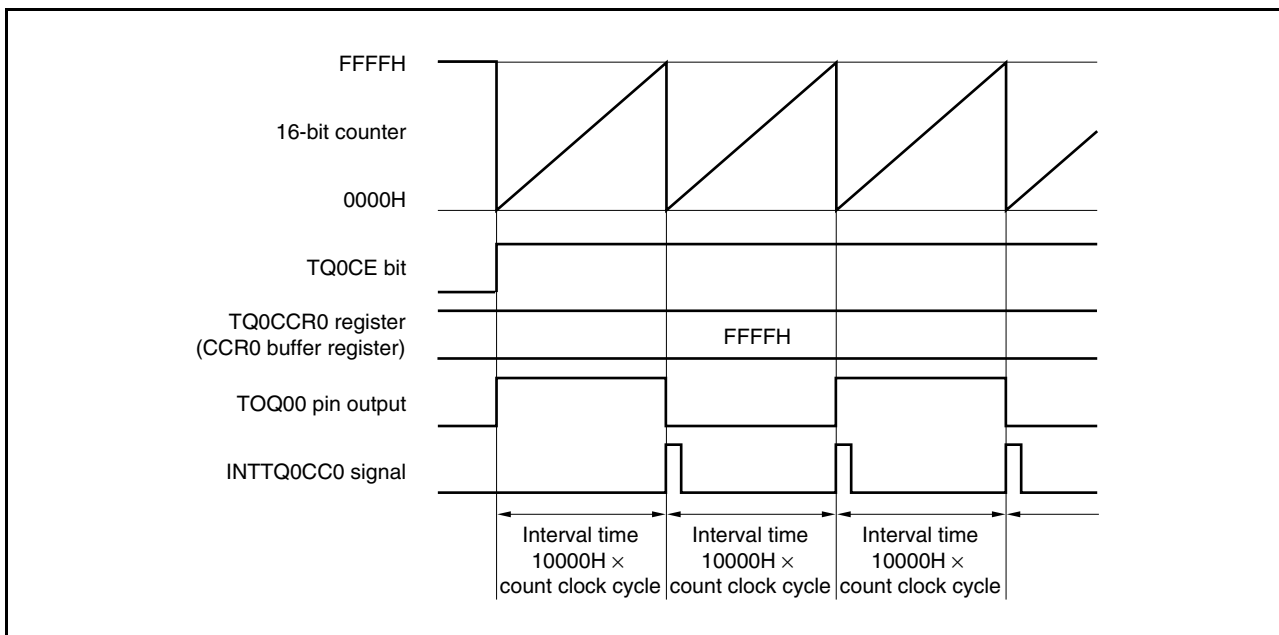
Figure 7-10. Operation of Interval Timer When TQ0CCR0 Register Is Set to 0000H



(b) Operation when TQ0CCR0 register is set to FFFFH

When the TQ0CCR0 register is set to FFFFH, the 16-bit counter increments up to FFFFH and is reset to 0000H in synchronization with the next increment timing. The INTTQ0CC0 signal is then generated and the output of the TOQ00 pin is inverted. At this time, an overflow interrupt request signal (INTTQ0OV) is not generated, nor is the overflow flag (TQ0OPT0.TQ0OVF bit) set to 1.

Figure 7-11. Operation of Interval Timer When TQ0CCR0 Register Is Set to FFFFH

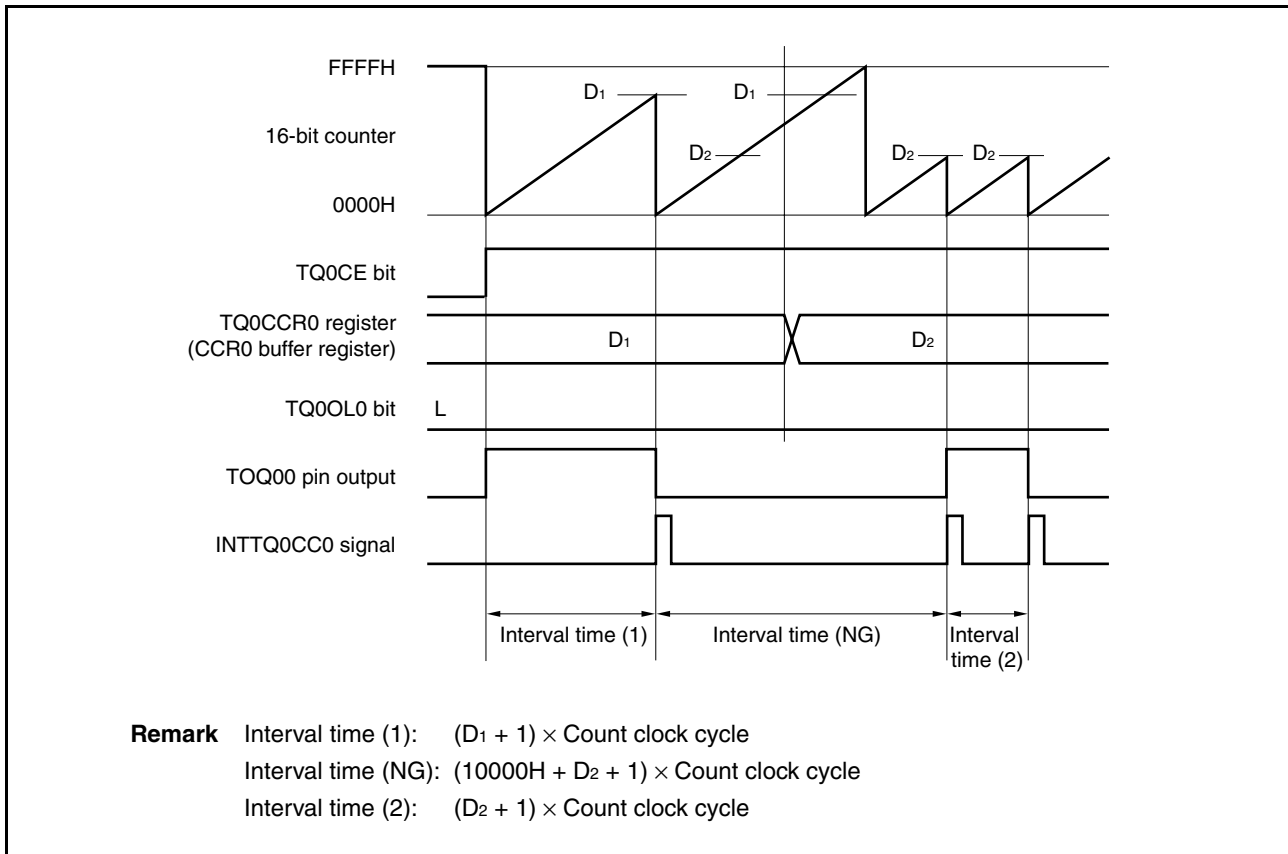


(c) Notes on rewriting TQ0CCR0 register

When rewriting the value of the TQ0CCR0 register to a smaller value, stop counting first and then change the set value.

If the value of the TQ0CCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.

Figure 7-12. Rewriting TQ0CCR0 Register



If the value of the TQ0CCR0 register is changed from D_1 to D_2 while the counter value is greater than D_2 but less than D_1 , the TQ0CCR0 register value is transferred to the CCR0 buffer register as soon as the register has been rewritten. Consequently, the value that is compared with the 16-bit counter value is D_2 .

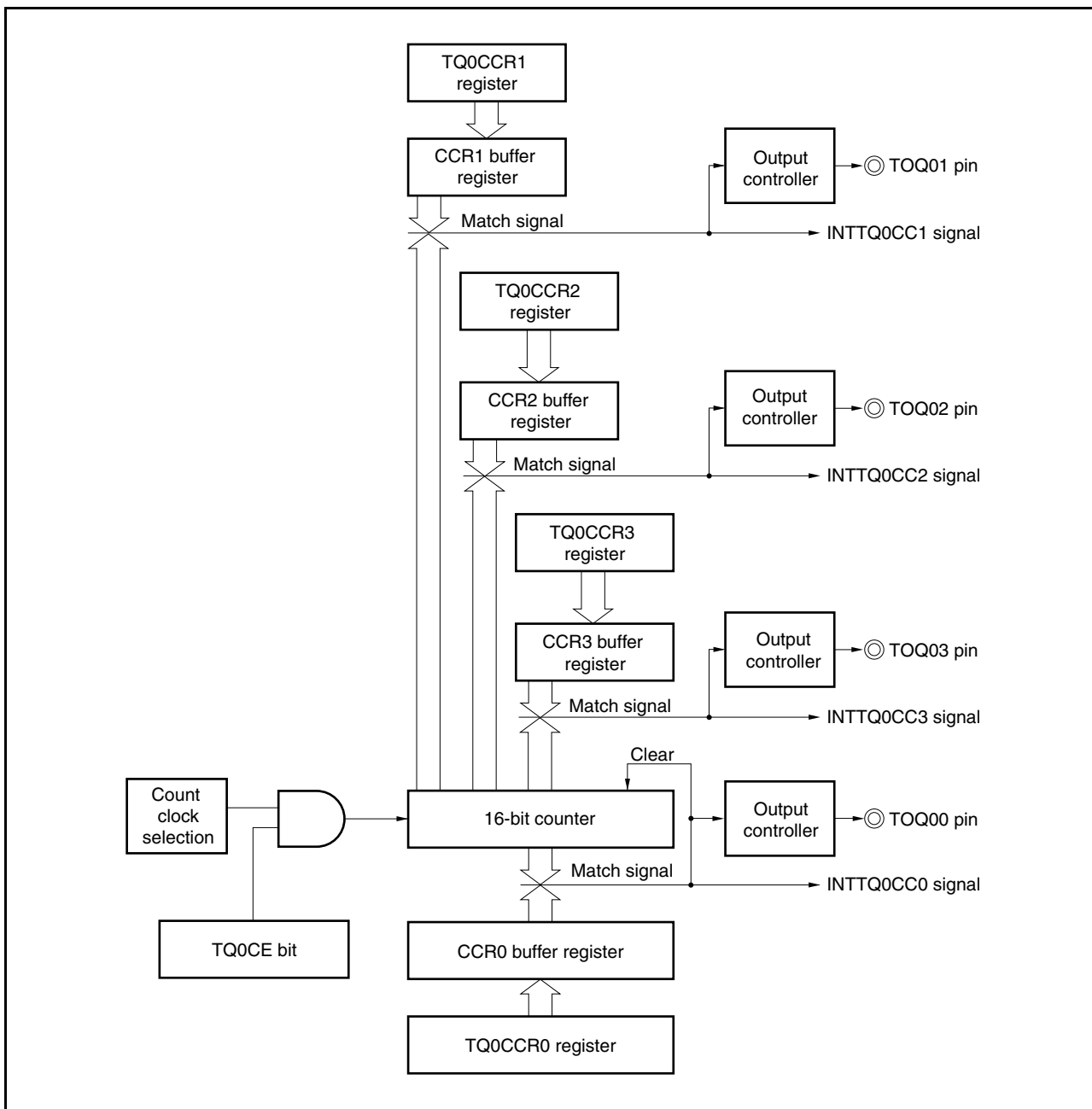
Because the counter value has already exceeded D_2 , however, the 16-bit counter increments to FFFFH, overflows, and then increments again from 0000H. When the counter value matches D_2 , the INTTQ0CC0 signal is generated and the output of the TOQ00 pin is inverted.

Therefore, the INTTQ0CC0 signal may not be generated at the interval " $(D_1 + 1) \times \text{Count clock cycle}$ " or " $(D_2 + 1) \times \text{Count clock cycle}$ " as originally expected, but instead may be generated at an interval of " $(10000H + D_2 + 1) \times \text{Count clock cycle}$ ".

(d) Operation of TQ0CCR1 to TQ0CCR3 registers

The TQ0CCR1 to TQ0CCR3 registers are configured as follows in the interval timer mode.

Figure 7-13. Configuration of TQ0CCR1 to TQ0CCR3 Registers

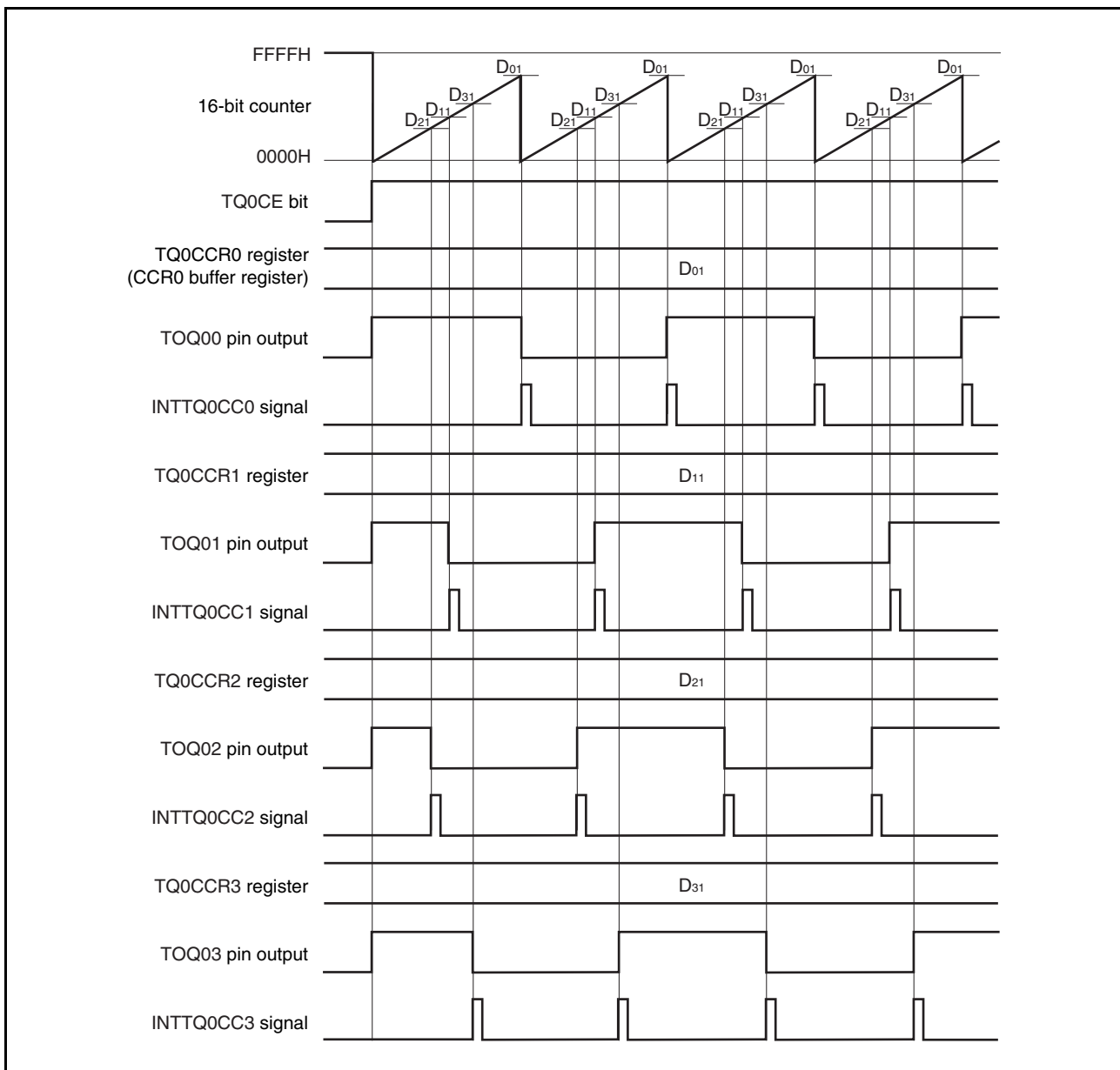


If the value of the TQ0CCRk register is less than or equal to the value of the TQ0CCR0 register, the INTTQ0CCk signal is generated once per cycle. At the same time, the output of the TOQ0k pin is inverted. The TOQ0k pin outputs a square wave with the same cycle as that output by the TOQ00 pin but with a different phase.

A chart showing the timing of operations when the value of the TQ0CCRk register (D_{k1}) is less than or equal to the value of the TQ0CCR0 register (D_{01}) is shown below.

Remark k = 1 to 3

Figure 7-14. Timing of Operations When $D_{01} \geq D_{k1}$

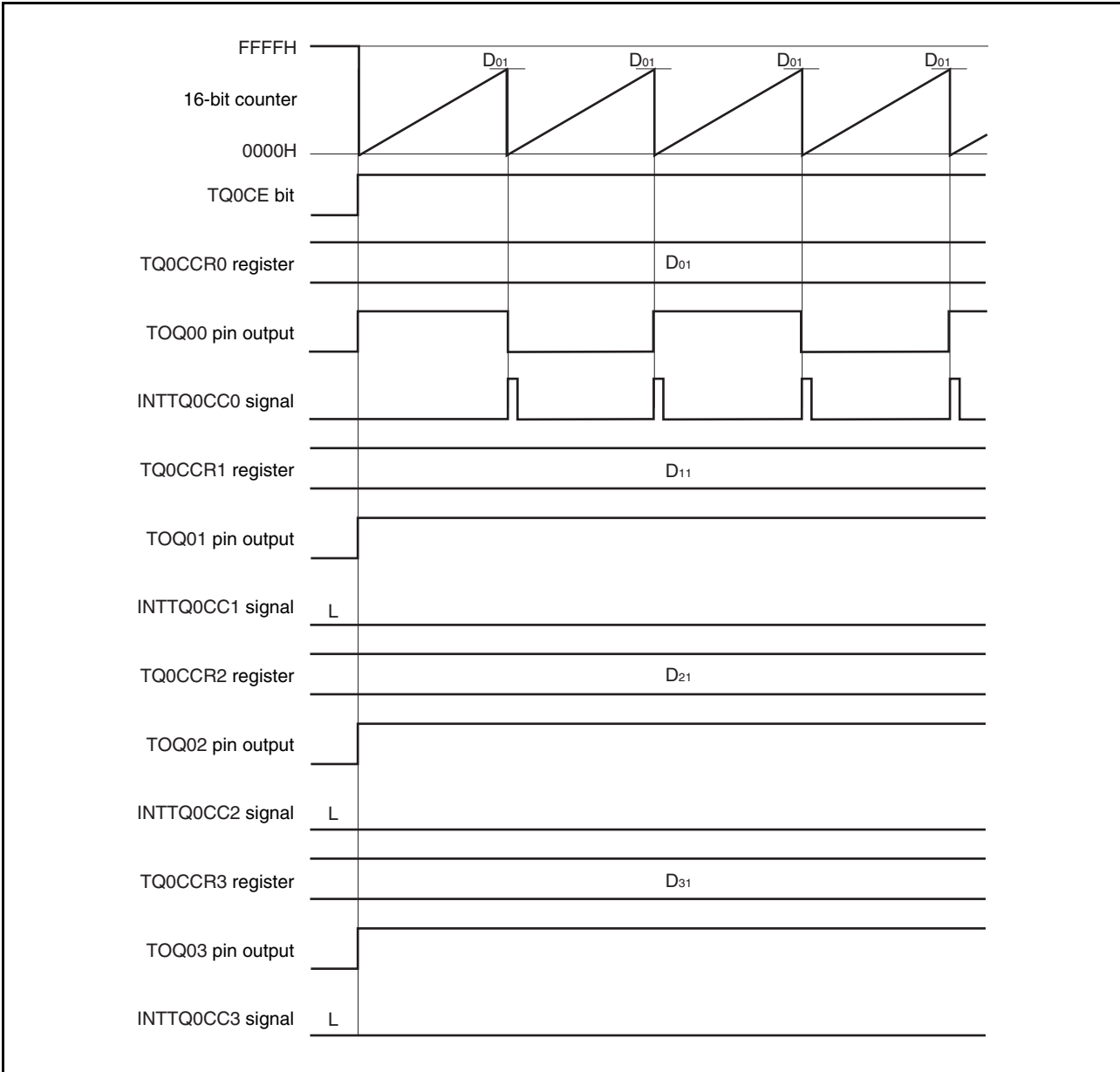


If the value of the TQ0CCRk register is greater than the value of the TQ0CCR0 register, the value of the 16-bit counter will not match the value of the TQ0CCRk register. Consequently, the INTTQ0CCk signal is not generated, nor is the output of the TOQ0k pin changed.

A chart showing the timing of operations when the value of the TQ0CCRk register (D_{k1}) is greater than the value of the TQ0CCR0 register (D_{01}) is shown below.

Remark k = 1 to 3

Figure 7-15. Timing of Operations When $D_{01} < D_{k1}$

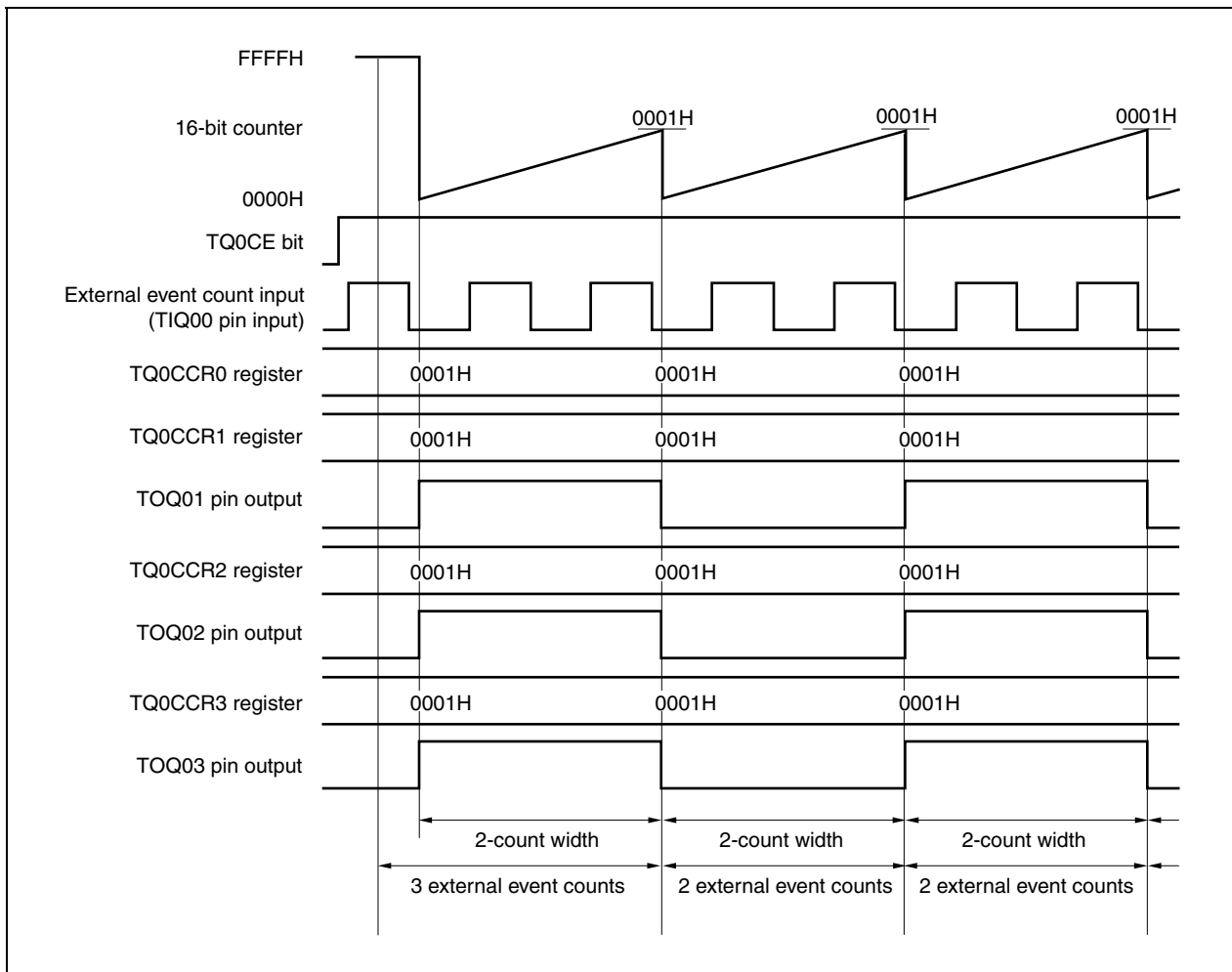


(3) Operation of interval timer based on input of external event count

(a) Operation

When the 16-bit counter is incrementing based on the valid edge of the external event count input (TIQ00 pin) in the interval timer mode, one external event count valid edge must be input immediately after the TQ0CE bit changes from 0 to 1 to start the counter incrementing after the 16-bit counter is cleared from FFFFH to 0000H. Once the TQ0CCR0 and TQ0CCRk registers are set to 0001H (that is, the same value as was previously set), the TOQ0k pin output is inverted every two counts of the 16-bit counter (k = 1 to 3). Note that the TQ0CTL1.TQ0EEE bit can only be set to 1 when timer output (TOQ0k) is used based on the input of an external event count.

Figure 7-16. Operation of Interval Timer Based on Input of External Event Count



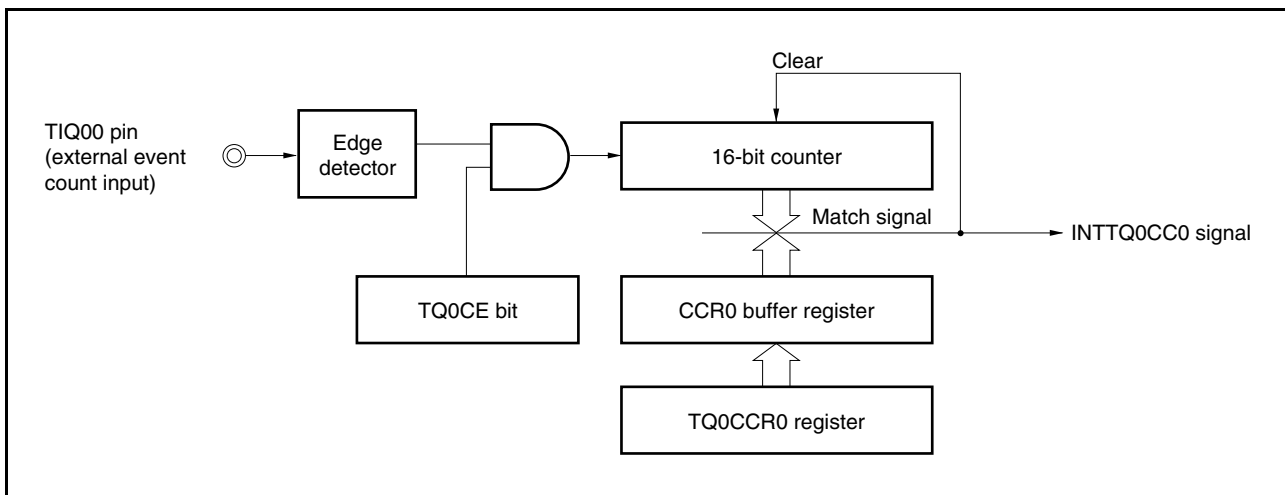
7.4.2 External event count mode (TQ0MD2 to TQ0MD0 bits = 001)

In the external event count mode, the valid edge of the external event count input is counted when the TQ0CTL0.TQ0CE bit is set to 1, and an interrupt request signal (INTTQ0CC0) is generated each time the specified number of edges have been counted. The timer output pins (TOQ00 to TOQ03) cannot be used. To use the TOQ01 to TOQ03 pins in the external event count mode, be sure to set the TQ0CTL1.TQ0EEE bit to 1 in the interval timer mode first. (For details, see 7.4.1 (3) Operation of interval timer based on input of external event count.)

Usually, the TQ0CCR1 to TQ0CCR3 registers are not used in the external event count mode.

- Remarks 1.** For how to set the TIQ00 pin, see **Table 7-2 Pins Used by TMQ0** and **Table 4-17 Settings When Pins Are Used for Alternate Functions**.
- 2.** For how to enable the INTTQ0CC0 interrupt signal, see **CHAPTER 19 INTERRUPT SERVICING/ EXCEPTION PROCESSING FUNCTION**.

Figure 7-17. Configuration of Interval Timer in External Event Count Mode

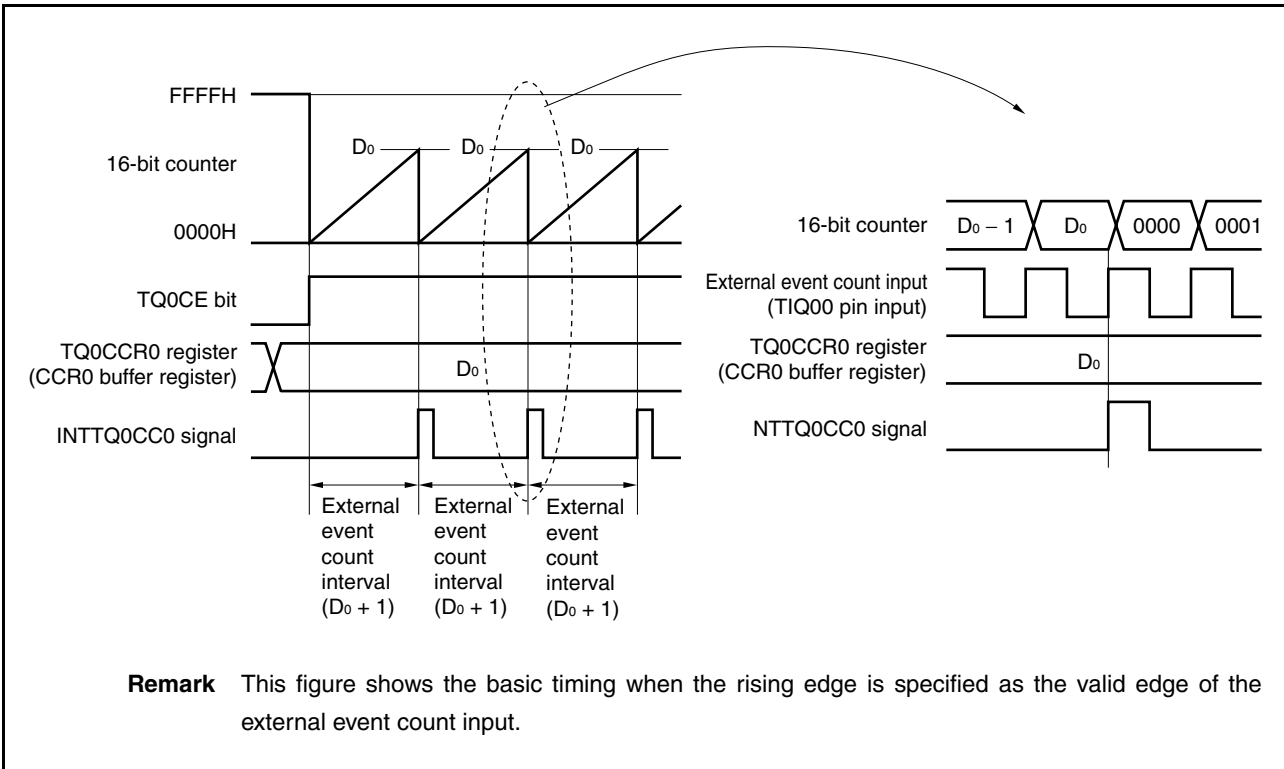


When the TQ0CE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H. The counter increments each time the valid edge of the external event count input is detected, and the value of the TQ0CCR0 register is transferred to the CCR0 buffer register.

When the value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTQ0CC0) is generated.

The INTTQ0CC0 signal is generated each time the valid edge of the external event count input has been detected the specified number of times (that is, the value of the TQ0CCR0 register + 1).

Figure 7-18. Basic Timing of Operations in External Event Count Mode



An example of the register settings when the external event count mode is used is shown in the figure below.

Figure 7-19. Register Settings in External Event Count Mode (1/2)

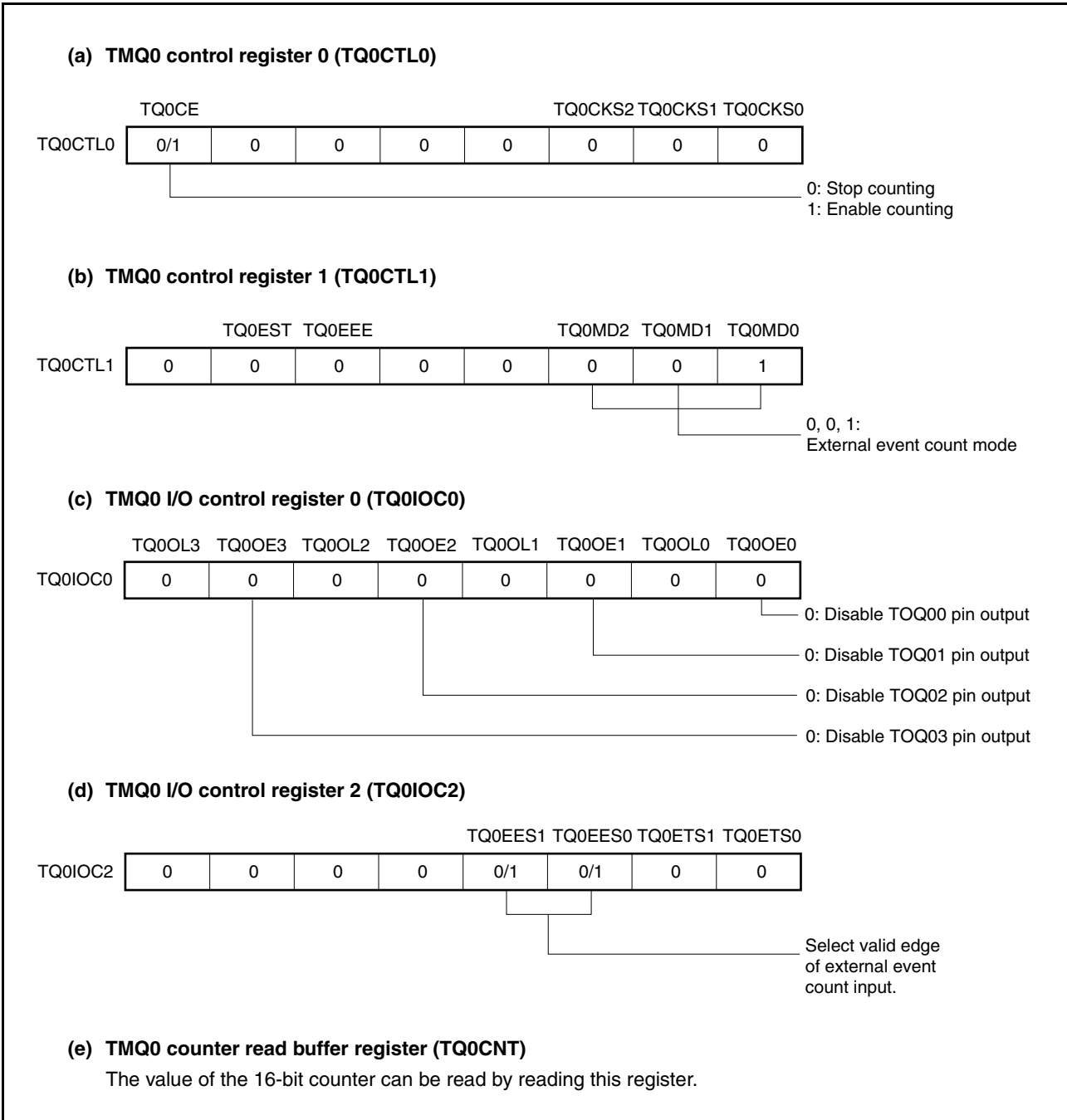


Figure 7-19. Register Settings in External Event Count Mode (2/2)

(f) TMQ0 capture/compare register 0 (TQ0CCR0)

When the TQ0CCR0 register is set to D_0 , the counter is cleared and a compare match interrupt request signal (INTTQ0CC0) is generated when the number of external events reaches $(D_0 + 1)$.

(g) TMQ0 capture/compare registers 1 to 3 (TQ0CCR1 to TQ0CCR3)

The TQ0CCR1 to TQ0CCR3 registers are not usually used in the external event count mode. However, because the set values of the TQ0CCR1 to TQ0CCR3 registers are transferred to the CCR1 to CCR3 buffer registers and a compare match interrupt request signal (INTTQ0CC1 to INTTQ0CC3) is generated when the value of the 16-bit counter matches the value of the CCR1 to CCR3 buffer registers, interrupts from these registers must be masked by setting the interrupt mask flags (TQ0CCMK1 to TQ0CCMK3).

Cautions 1. Do not set the TQ0CCR0 register to 0000H in the external event count mode.

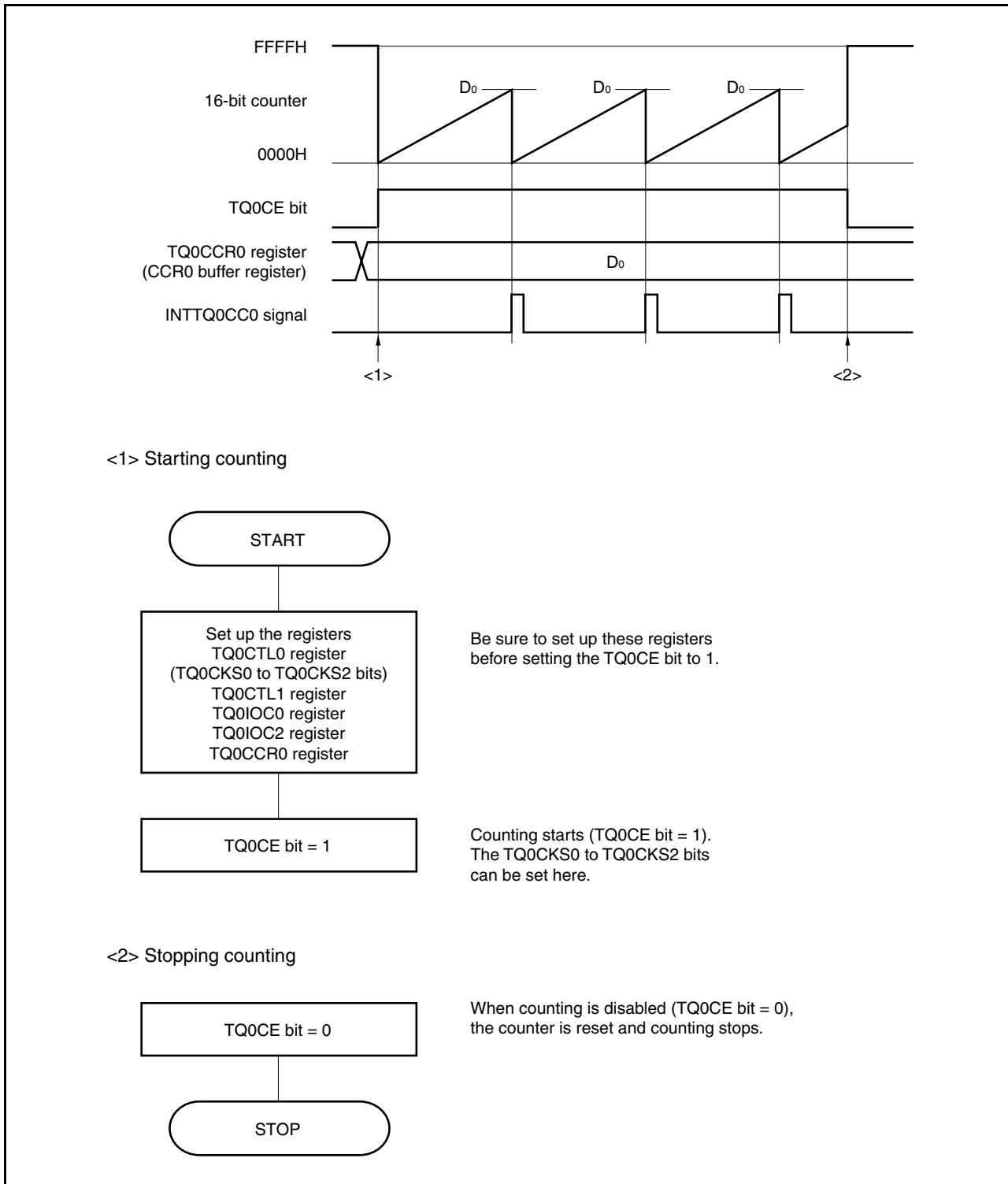
2. Timer output cannot be used in the external event count mode. When using the timer output based on the input of an external event count, first set the operating mode to interval mode, and then specify “operation enabled” for the external event count input (by setting the TQ0CTL1.TQ0MD2 to TQ0MD0 bits to 0, 0, 0 and setting the TQ0CTL1.TQ0EEE bit to 1). (For details, see 7.4.1 (3) Operation of interval timer based on input of external event count.)

3. When an external clock is used as the count clock, the external clock can be input only from the TIQ00 pin. At this time, set the TQ0IOC1.TQ0IS1 and TQ0IOC1.TQ0IS0 bits to 0, 0 (capture trigger input (TIQ00 pin): no edge detection).

Remark TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the external event count mode.

(1) Operations in external event count mode

Figure 7-20. Timing and Processing of Operations in External Event Count Mode

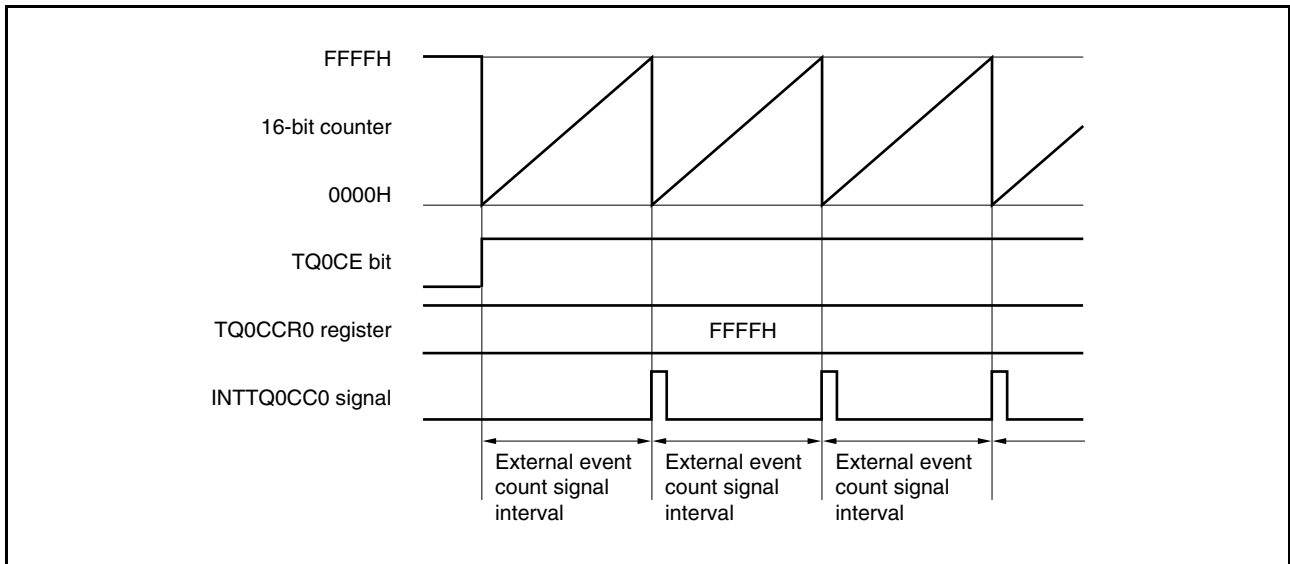


(2) Using external event count mode

(a) Operation when TQ0CCR0 register is set to FFFFH

When the TQ0CCR0 register is set to FFFFH, the 16-bit counter increments up to FFFFH upon detection of the valid edge of the external event count signal and is reset to 0000H in synchronization with the next increment timing. The INTTQ0CC0 signal is then generated. At this time, the TQ0OPT0.TQ0OVF bit is not set to 1.

Figure 7-21. Operation When TQ0CCR0 Register Is Set to FFFFH

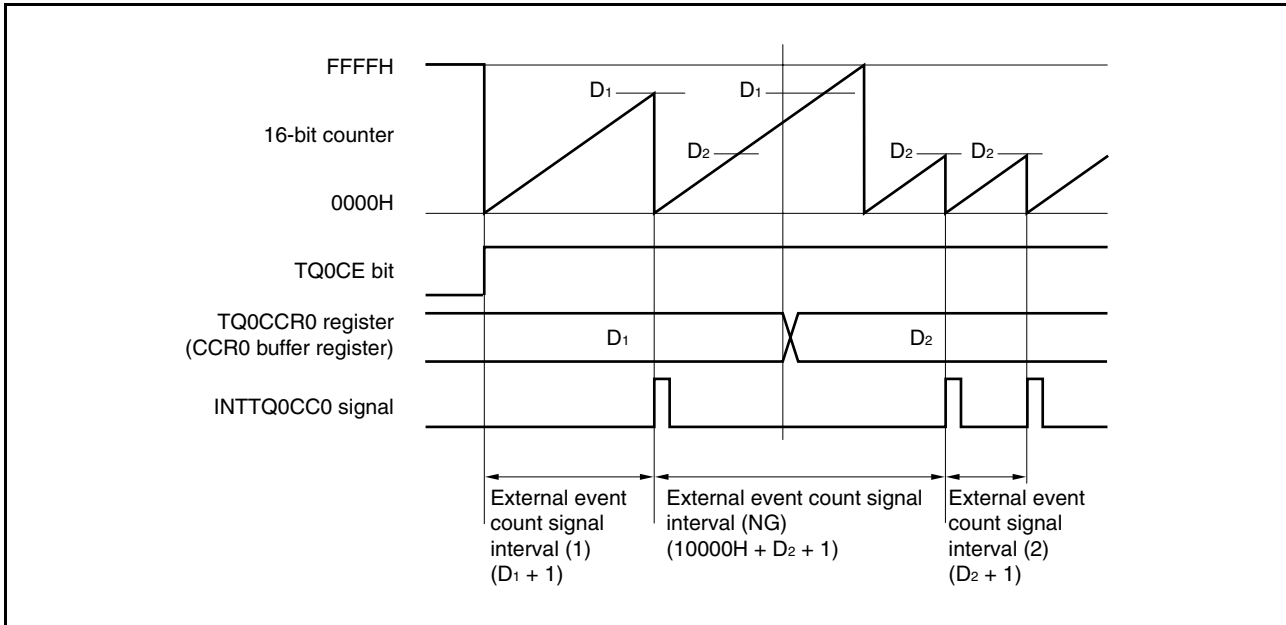


(b) Notes on rewriting TQ0CCR0 register

When rewriting the value of the TQ0CCR0 register to a smaller value, stop counting first and then change the set value.

If the value of the TQ0CCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.

Figure 7-22. Rewriting TQ0CCR0 Register



If the value of the TQ0CCR0 register is changed from D_1 to D_2 while the counter value is greater than D_2 but less than D_1 , the TQ0CCR0 register value is transferred to the CCR0 buffer register as soon as the register has been rewritten. Consequently, the value that is compared with the 16-bit counter value is D_2 .

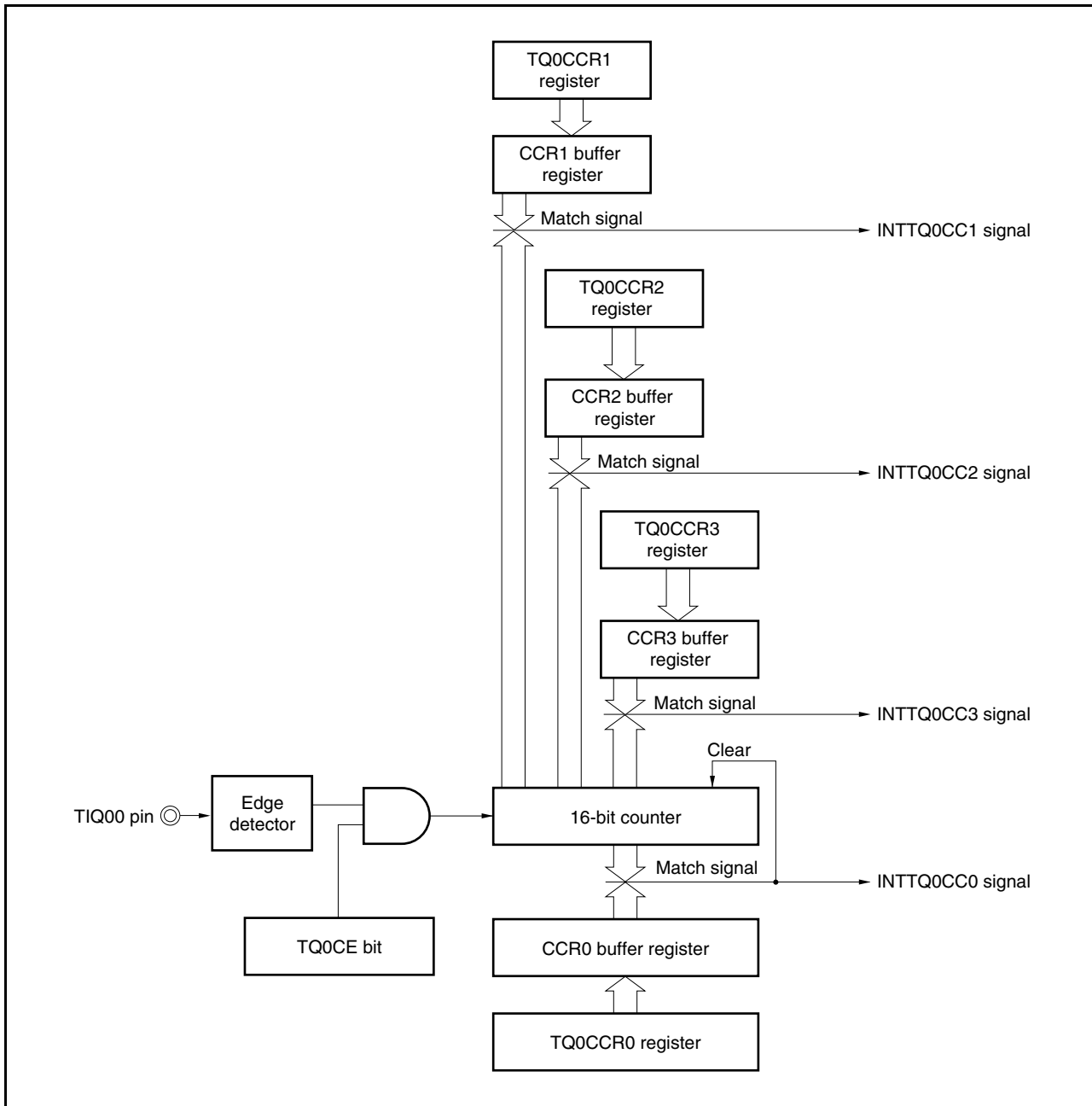
Because the counter value has already exceeded D_2 , however, the 16-bit counter increments up to FFFFH, overflows, and then increments up again from 0000H. When the counter value matches D_2 , the INTTQ0CC0 signal is generated.

Therefore, the INTTQ0CC0 signal may not be generated at the valid edge of the external event count signal when the external event count is " $(D_1 + 1)$ " or " $(D_2 + 1)$ " as originally expected, but instead may be generated at the valid edge of the external event count signal when the external event count is " $(10000H + D_2 + 1)$ ".

(c) Operation of TQ0CCR1 to TQ0CCR3 registers

The TQ0CCR1 to TQ0CCR3 registers are configured as follows in the external event count mode.

Figure 7-23. Configuration of TQ0CCR1 to TQ0CCR3 Registers

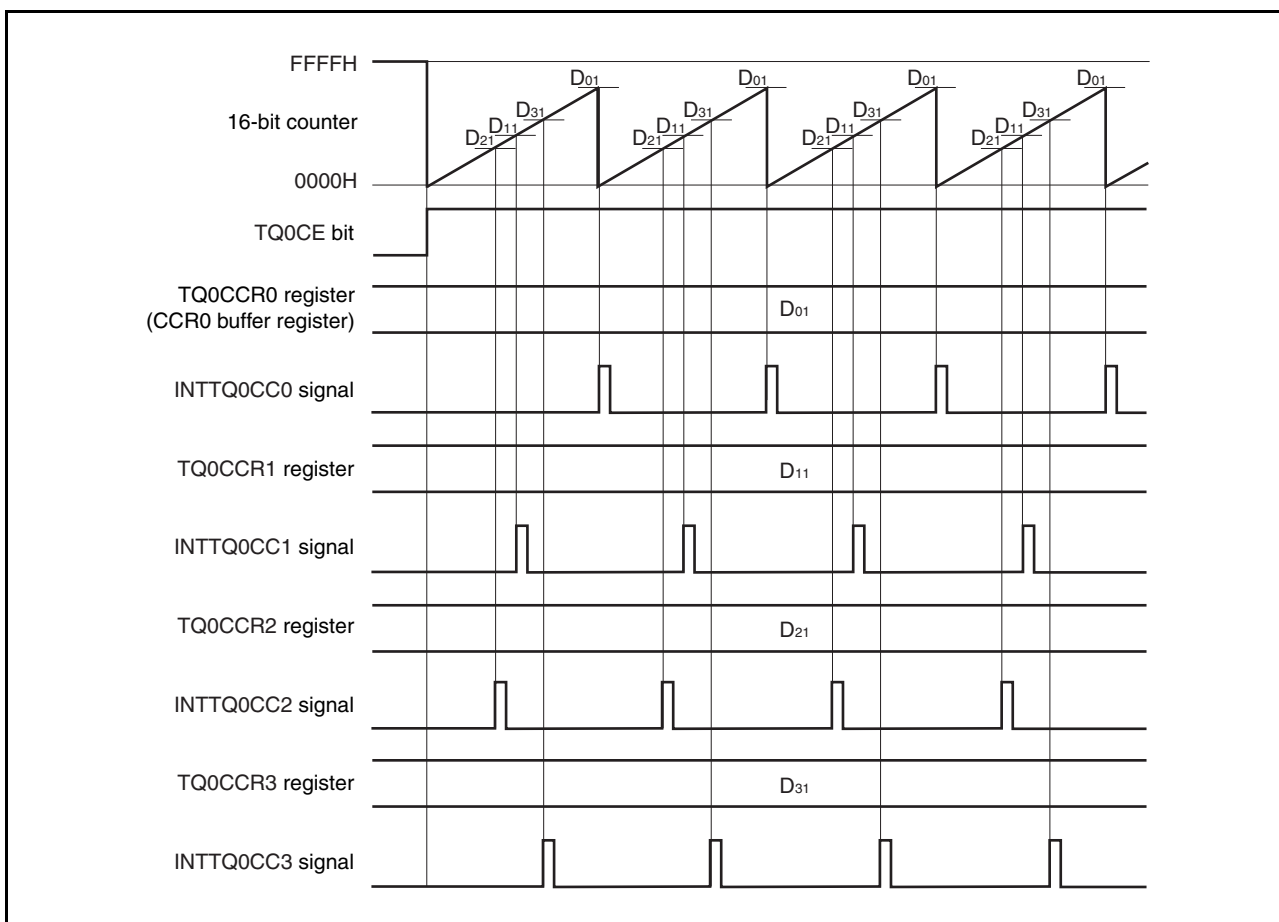


If the value of the TQ0CCRk register is less than or equal to the value of the TQ0CCR0 register, the INTTQ0CCk signal is generated once per cycle.

A chart showing the timing of operations when the value of the TQ0CCRk register (D_{k1}) is less than or equal to the value of the TQ0CCR0 register (D_{01}) is shown below.

Remark k = 1 to 3

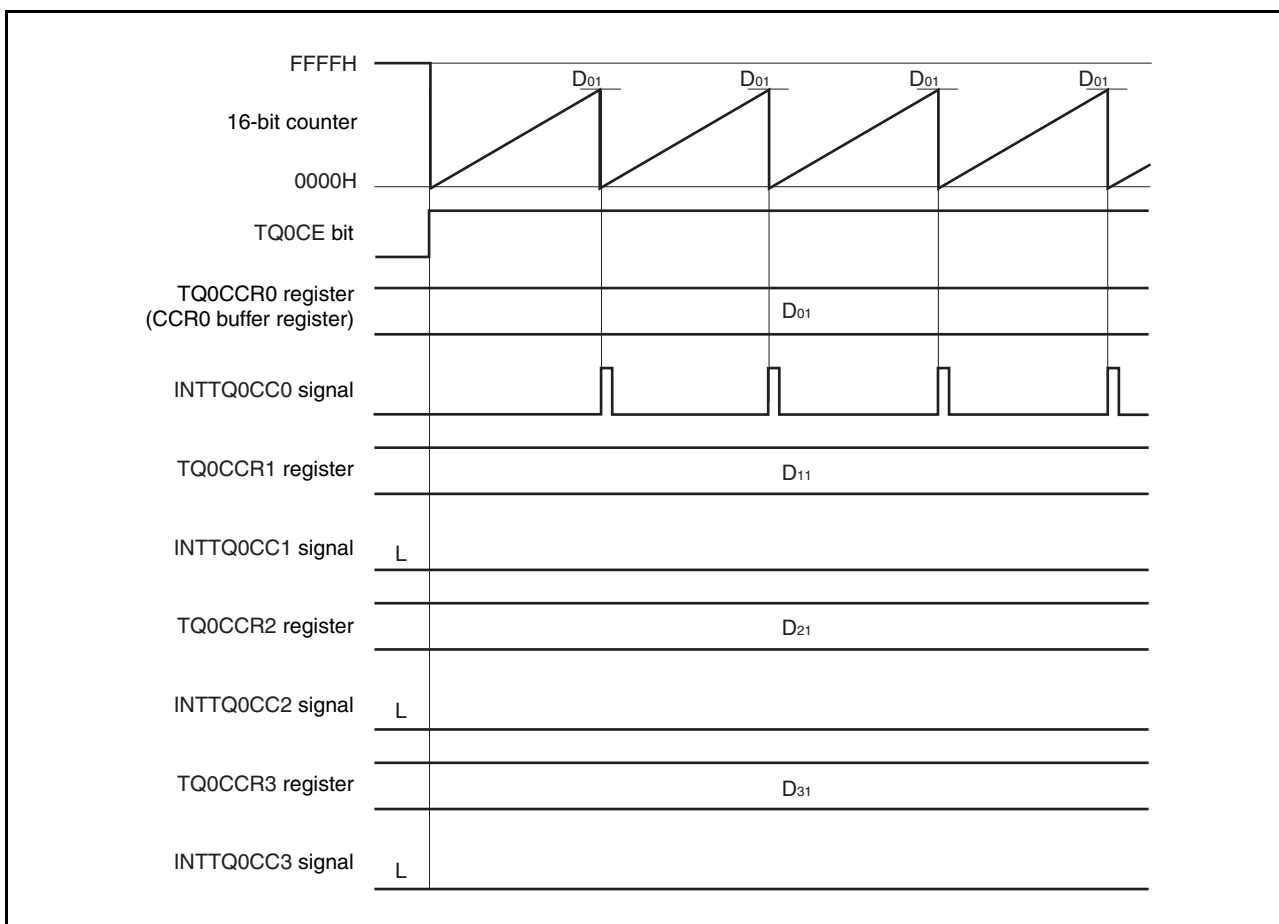
Figure 7-24. Timing of Operations When $D_{01} \geq D_{k1}$



If the value of the TQ0CCRk register is greater than the value of the TQ0CCR0 register, the value of the 16-bit counter will not match the value of the TQ0CCRk register and the INTTQ0CCk signal will not be generated. A chart showing the timing of operations when the value of the TQ0CCRk register (D_{k1}) is greater than the value of the TQ0CCR0 register (D_{01}) is shown below.

Remark k = 1 to 3

Figure 7-25. Timing of Operations When $D_{01} < D_{k1}$



7.4.3 External trigger pulse output mode (TQ0MD2 to TQ0MD0 bits = 010)

In the external trigger pulse output mode, when the TQ0CTL0.TQ0CE bit is set to 1, TMQ0 waits for a trigger, which is the valid edge of the external trigger input signal, and starts incrementing when this trigger is detected. TMQ0 then outputs a PWM waveform from the TOQ01 to TOQ03 pins.

Pulses can also be output by generating a software trigger instead of using the external trigger. When using a software trigger instead of the external trigger, a square wave that has one cycle of the PWM waveform as half its cycle can also be output from the TOQ00 pin.

- Remarks 1. For how to set the TIQ00 and TOQ00 to TOQ03 pins, see Table 7-2 Pins Used by TMQ0 and Table 4-17 Settings When Pins Are Used for Alternate Functions.
- 2. For how to enable the INTTQ0CC0 to INTTQ0CC3 interrupt signals, see CHAPTER 19 INTERRUPT SERVICING/EXCEPTION PROCESSING FUNCTION.

Figure 7-26. Configuration of TMQ0 in External Trigger Pulse Output Mode

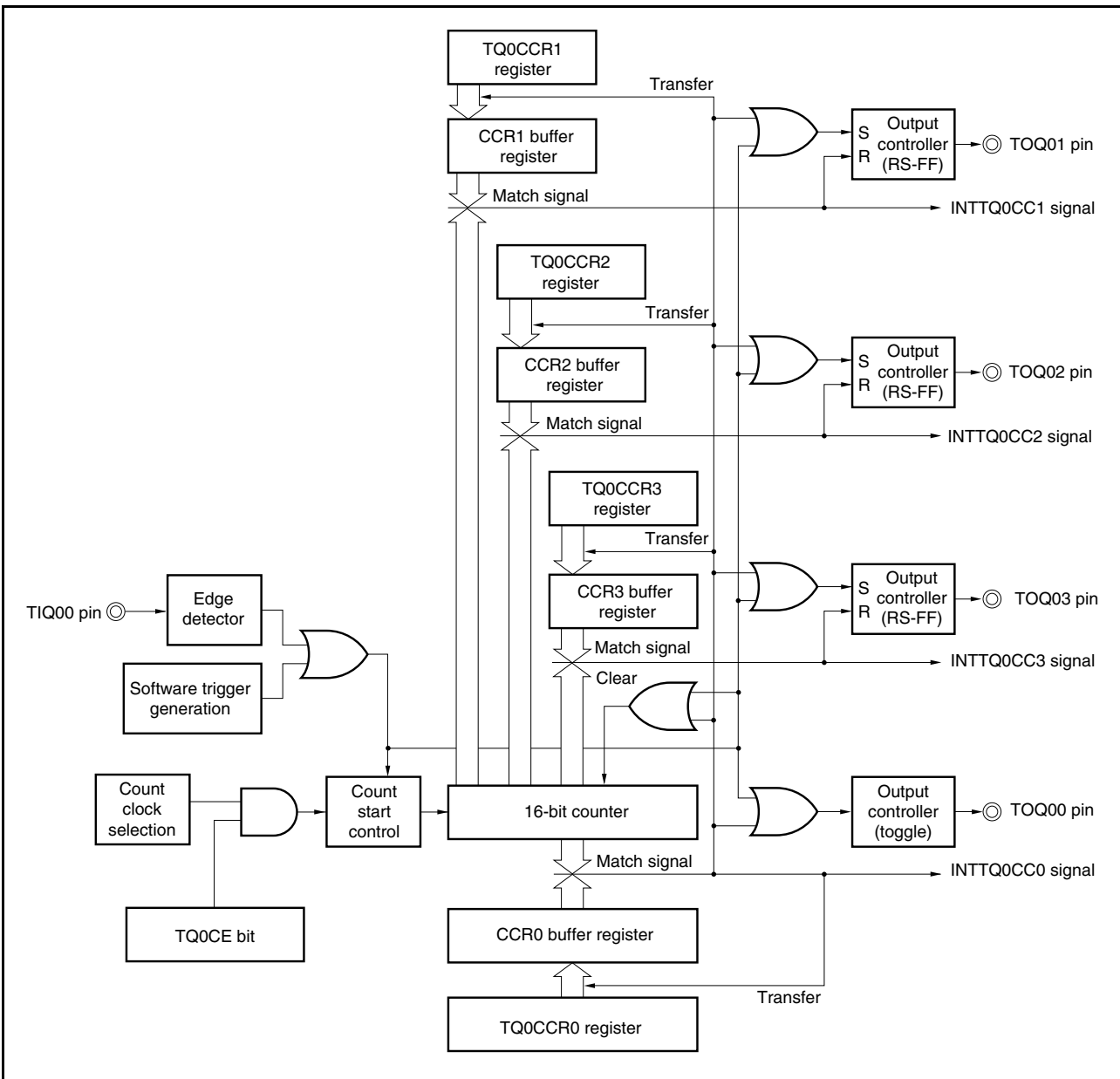
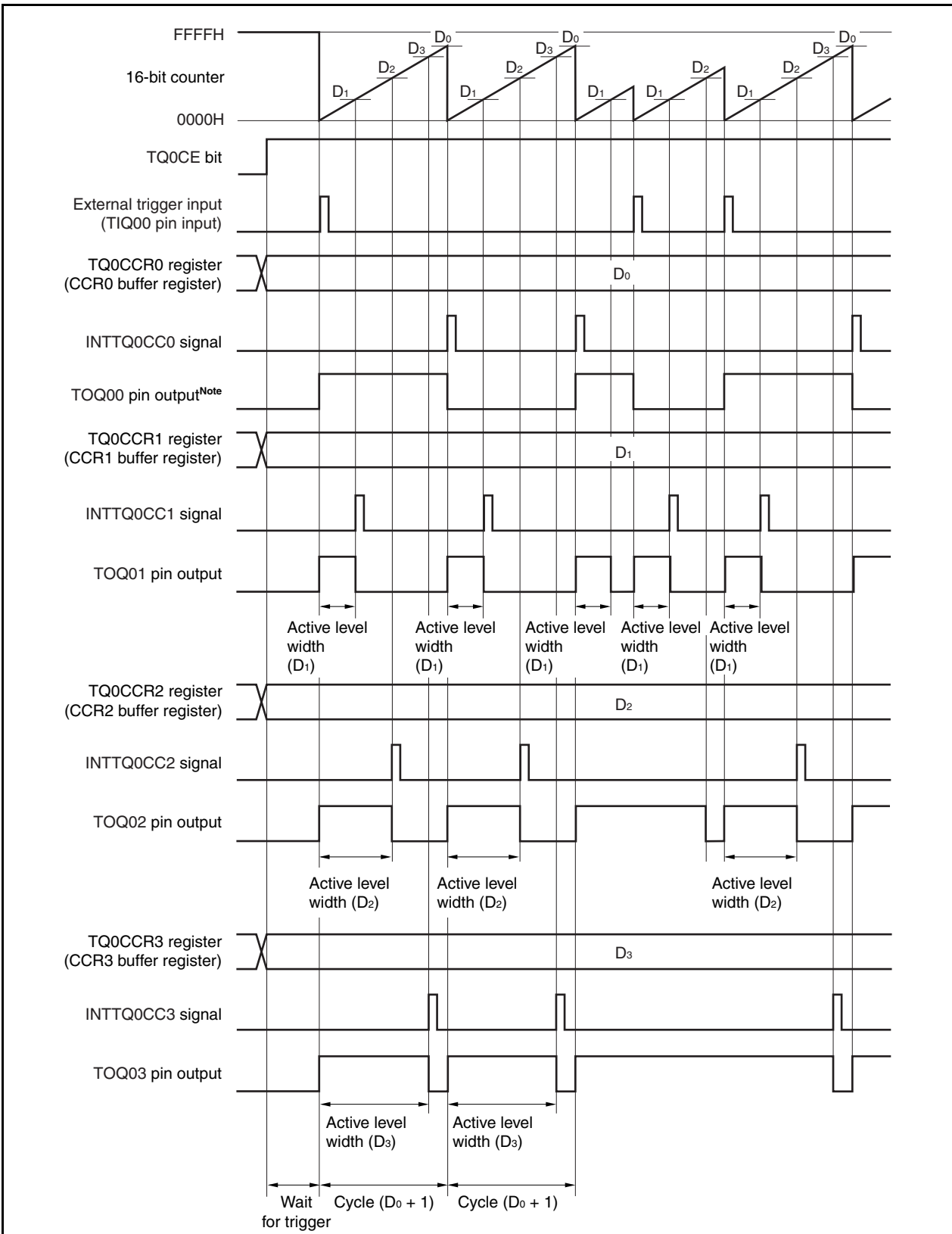


Figure 7-27. Basic Timing of Operations in External Trigger Pulse Output Mode



Note The output from the TOQ00 pin can also be used as the input to the TIQ00 pin. When using the output from the TOQ00 pin as the input to the TIQ00 pin, use a software trigger instead of an external trigger.

When the TQ0CE bit is set to 1, TMQ0 waits for a trigger. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts incrementing, and outputs a PWM waveform from the TOQ0k pin. If the trigger is generated again while the counter is incrementing, the counter is cleared to 0000H and restarts incrementing, and the output of the TOQ00 pin is inverted. (The TOQ0k pin outputs a high level signal 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 TQ0CCRk register) × Count clock cycle

Cycle = (Set value of TQ0CCR0 register + 1) × Count clock cycle

Duty factor = (Set value of TQ0CCRk register)/(Set value of TQ0CCR0 register + 1)

The INTTQ0CC0 compare match interrupt request signal is generated when the 16-bit counter increments next time after its value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The INTTQ0CCk compare match interrupt request signal is generated when the value of the 16-bit counter matches the value of the CCRk buffer register.

Either the valid edge of the external trigger input signal or setting the software trigger (TQ0CTL1.TQ0EST bit) to 1 is used as the trigger.

Remark k = 1 to 3

Figure 7-28. Register Settings in External Trigger Pulse Output Mode (1/3)

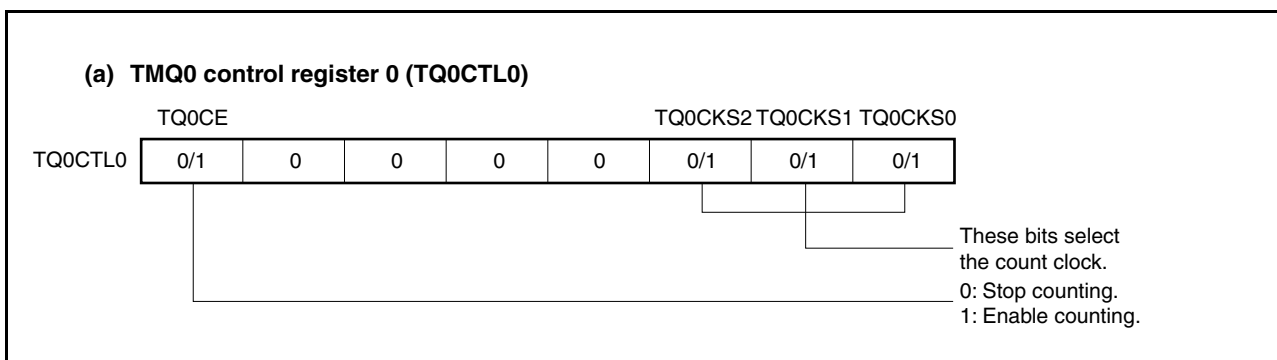


Figure 7-28. Register Settings in External Trigger Pulse Output Mode (2/3)

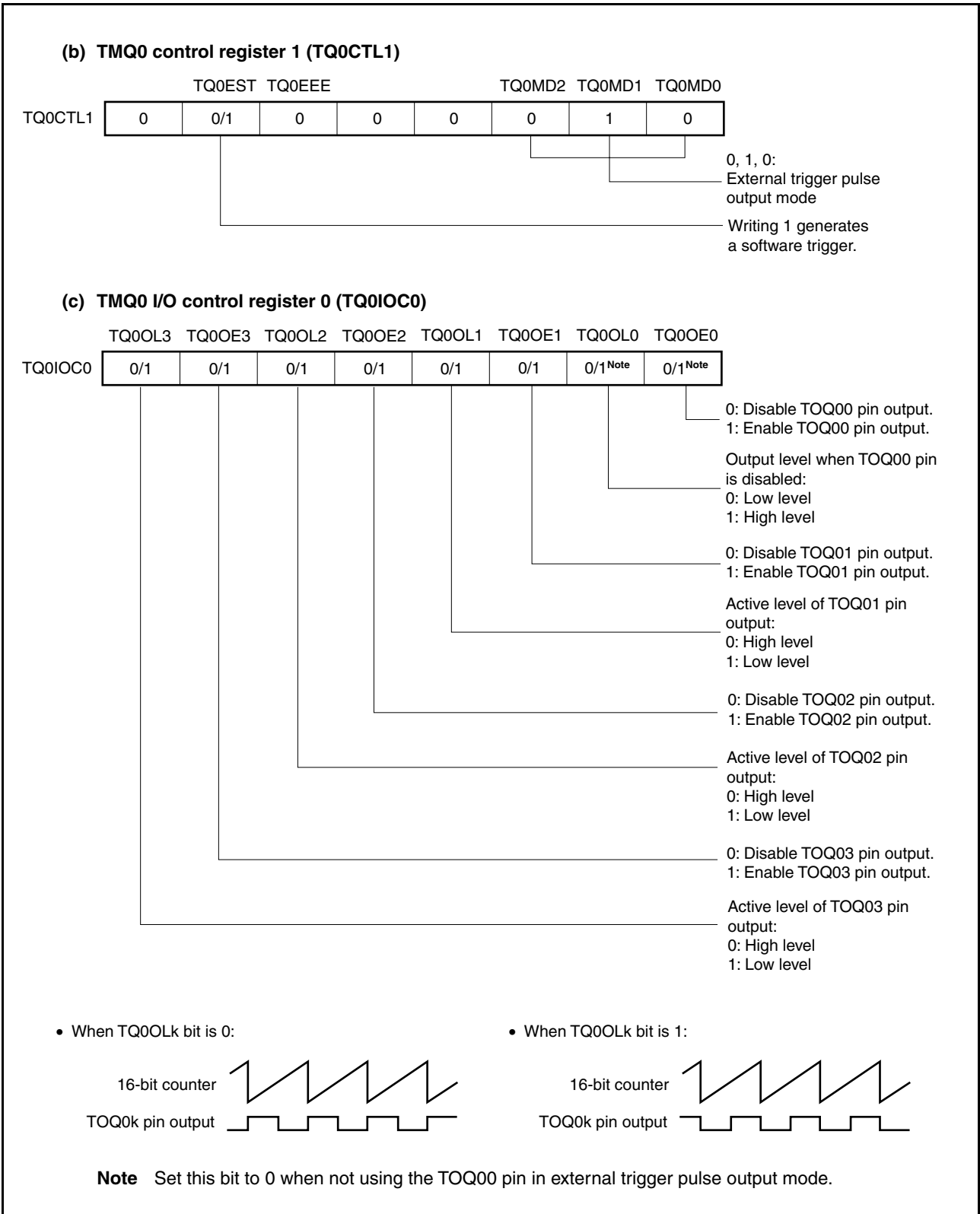
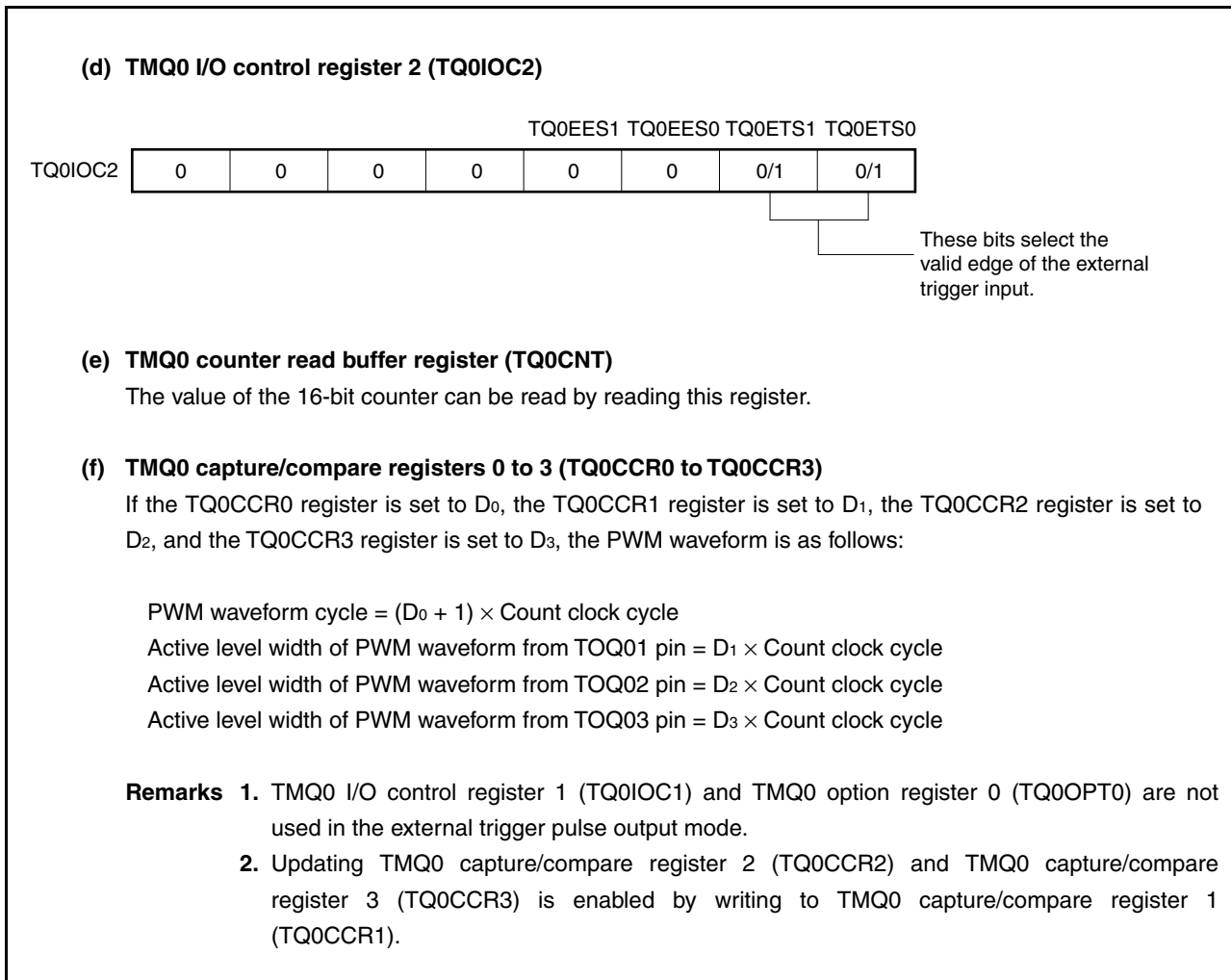


Figure 7-28. Register Settings in External Trigger Pulse Output Mode (3/3)



(1) Operations in external trigger pulse output mode

Figure 7-29. Timing and Processing of Operations in External Trigger Pulse Output Mode (1/2)

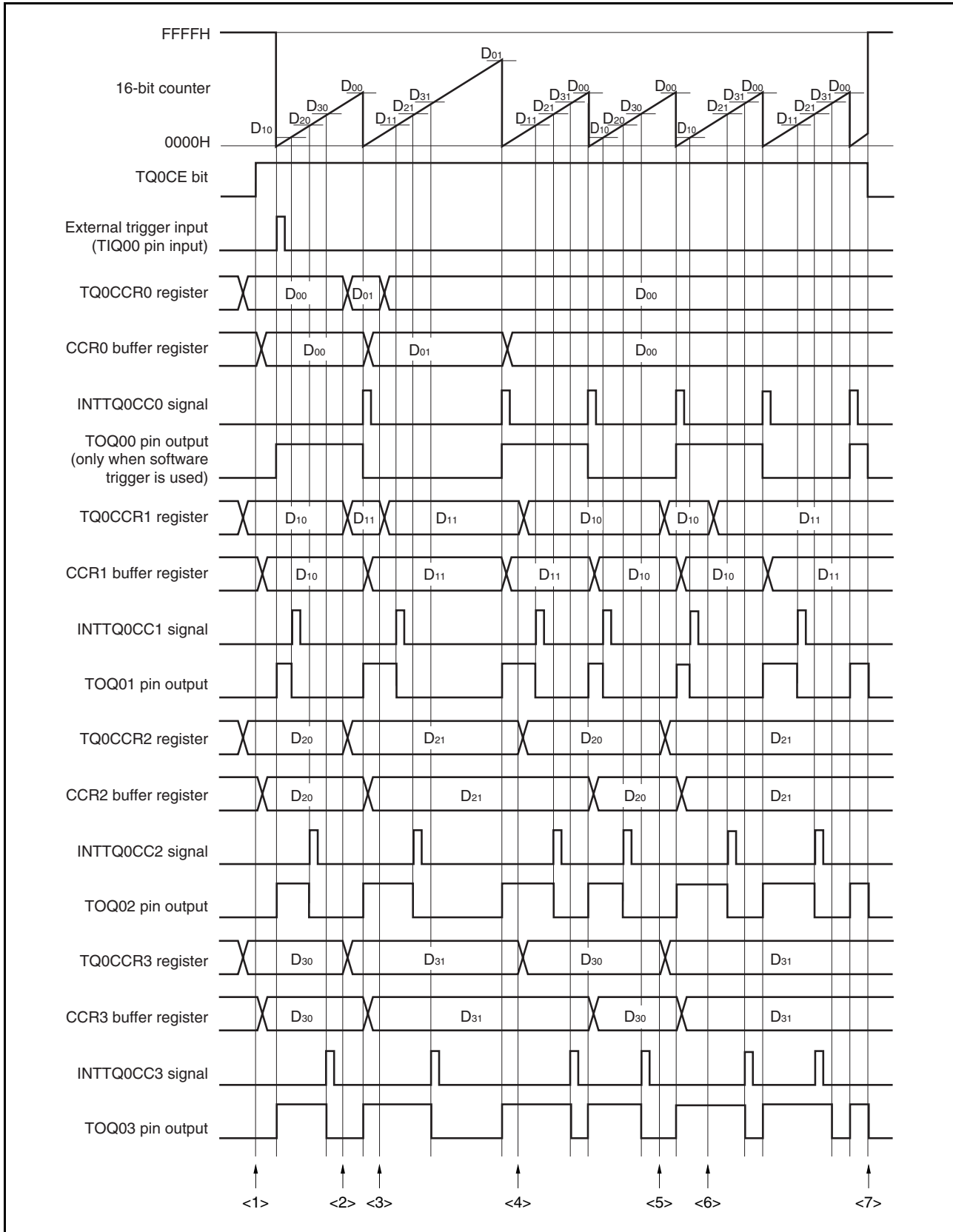
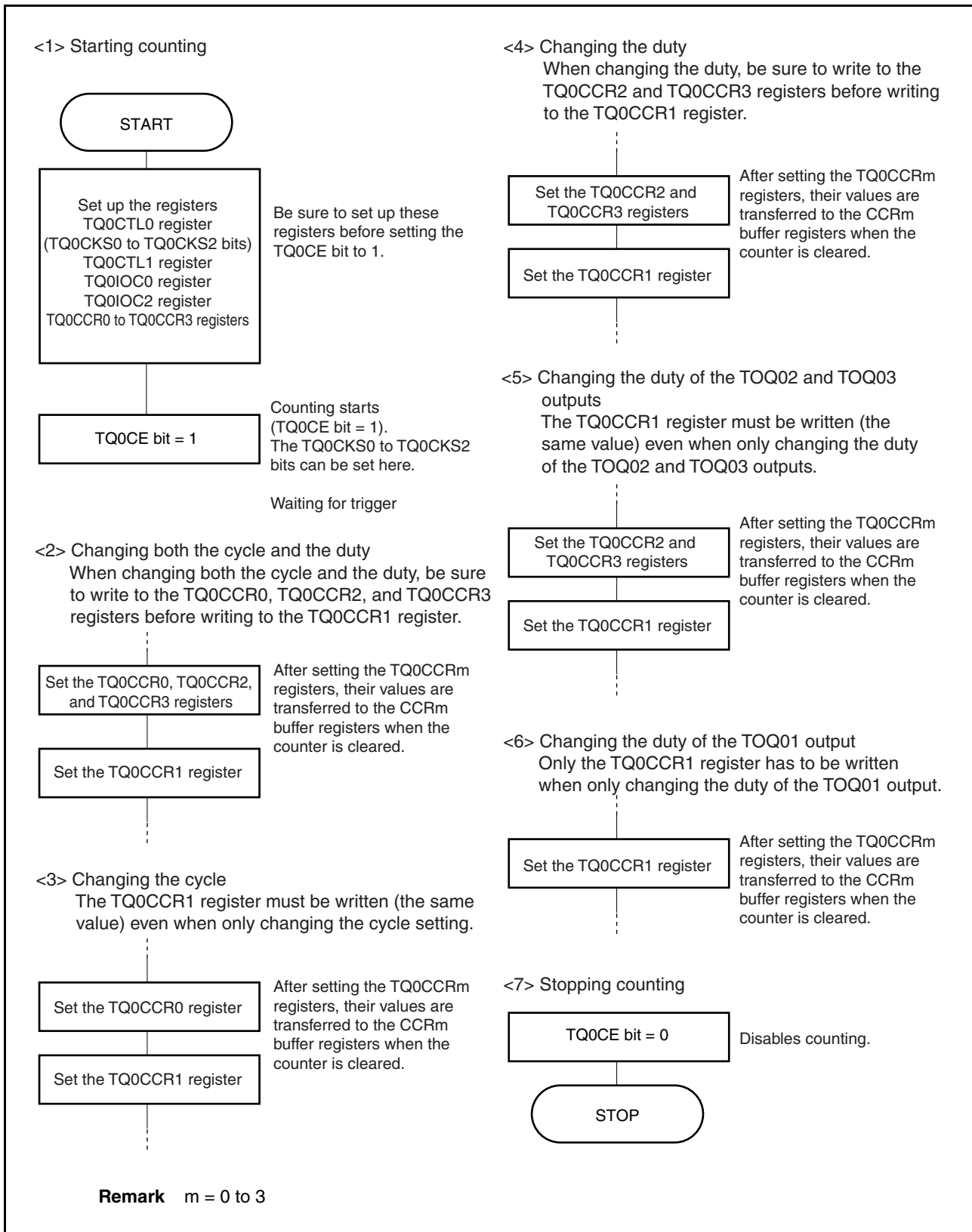


Figure 7-29. Timing and Processing of Operations in External Trigger Pulse Output Mode (2/2)



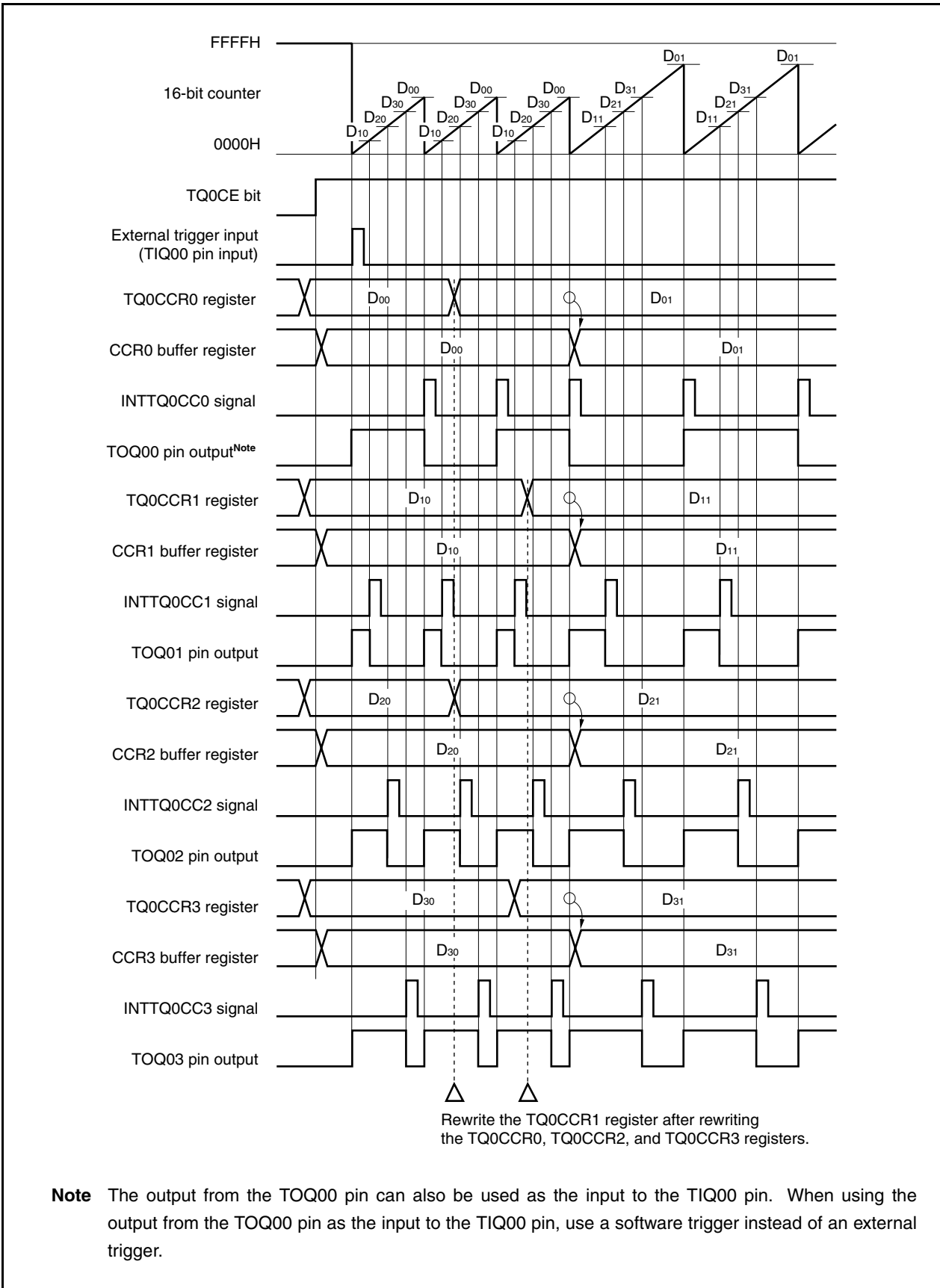
(2) Using external trigger pulse output mode

How to change the PWM waveform in the external trigger pulse output mode is described below.

(a) Changing the PWM waveform while the counter is incrementing

To change the PWM waveform while the counter is incrementing, write to the TQ0CCR1 register after changing the waveform setting. When rewriting the TQ0CCRk register after writing to the TQ0CCR1 register, do so after the INTTQ0CC0 signal has been detected.

Figure 7-30. Changing PWM Waveform While Counter Is Incrementing



In order to transfer data from the TQ0CCRm register to the CCRm buffer register, the TQ0CCR1 register must be written.

After data is written to the TQ0CCR1 register, the value written to the TQ0CCRm register is transferred to the CCRm buffer register in synchronization with clearing of the 16-bit counter, and is used as the value to be compared with the 16-bit counter value.

- <1> To change both the cycle and active level width of the PWM waveform, first set the cycle to the TQ0CCR0 register and then set the active level width to the TQ0CCR2 and TQ0CCR3 registers, before setting the active level width to the TQ0CCR1 register.
- <2> To change only the cycle of the PWM waveform, first set the cycle to the TQ0CCR0 register, and then write the same value to the TQ0CCR1 register (that is, the same value as the value already specified for the TQ0CCR1 register).
- <3> To change only the active level width (duty factor) of the PWM waveform, first set the active level width to the TQ0CCR2 and TQ0CCR3 registers, and then set the active level width to the TQ0CCR1 register.
- <4> To change only the active level width (duty factor) of the PWM waveform output from the TOQ01 pin, only the TQ0CCR1 register has to be set.
- <5> To change only the active level width (duty factor) of the PWM waveform output from the TOQ02 and TOQ03 pins, first set the active level width to the TQ0CCR2 and TQ0CCR3 registers, and then write the same value to the TQ0CCR1 register (that is, the same value as the value already specified for the TQ0CCR1 register).

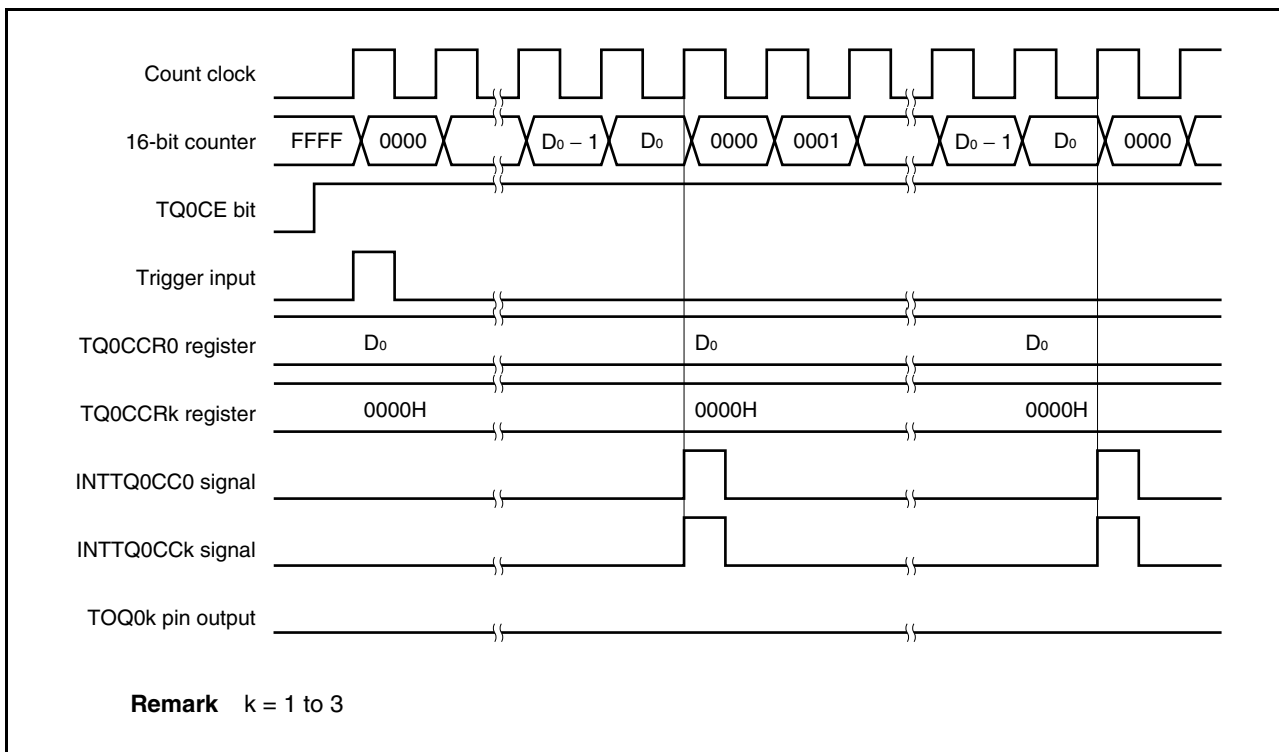
Caution To rewrite the TQ0CCR0 to TQ0CCR3 registers after writing the TQ0CCR1 register, do so after the INTTQ0CC0 signal has been generated; otherwise, the value of the CCRm buffer register may become undefined because the timing of transferring data from the TQ0CCRm register to the CCRm buffer register conflicts with writing the TQ0CCRm register.

Remark m = 0 to 3

(b) Outputting a 0% or 100% PWM waveform

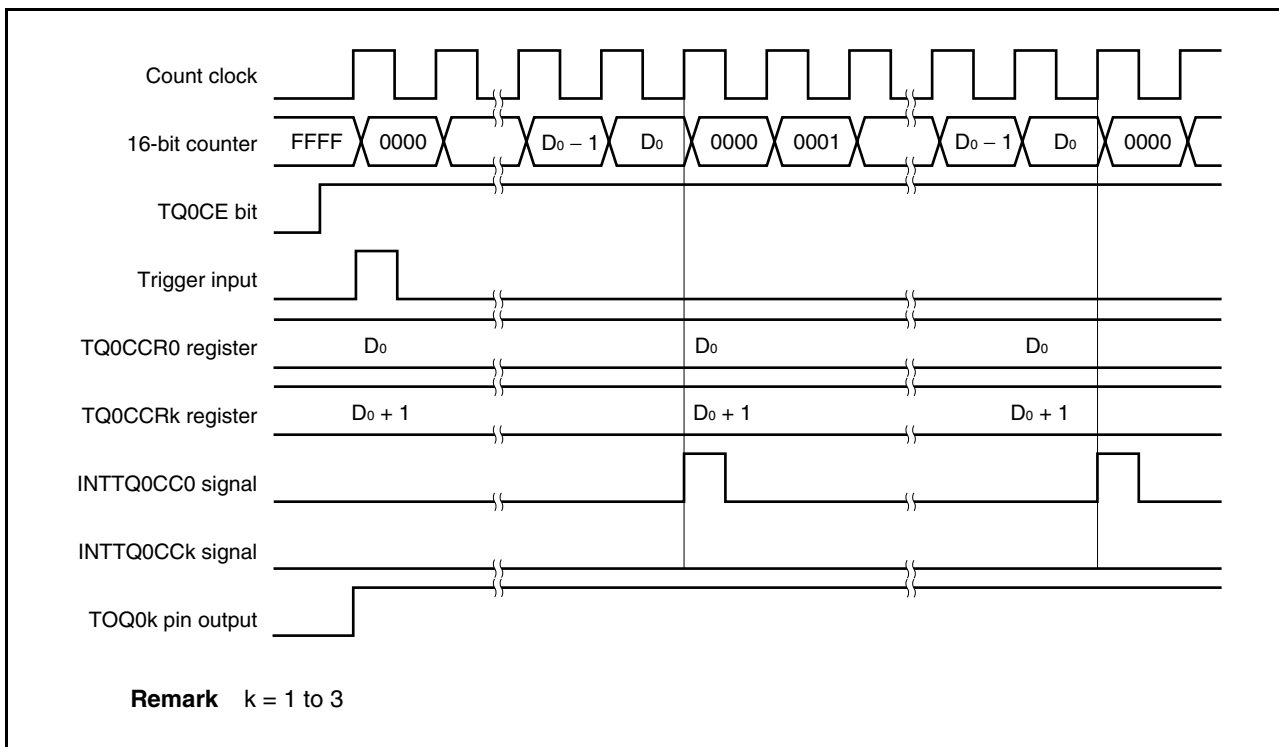
To output a 0% waveform, clear the TQ0CCRk register to 0000H. Note that if the set value of the TQ0CCR0 register is FFFFH, the INTTQ0CCK signal is generated periodically.

Figure 7-31. Outputting 0% PWM Waveform



To output a 100% waveform, set the value of TQ0CCR0 register + 1 to the TQ0CCRk register.
 If the value of the TQ0CCR0 register is FFFFH, a 100% waveform cannot be output.

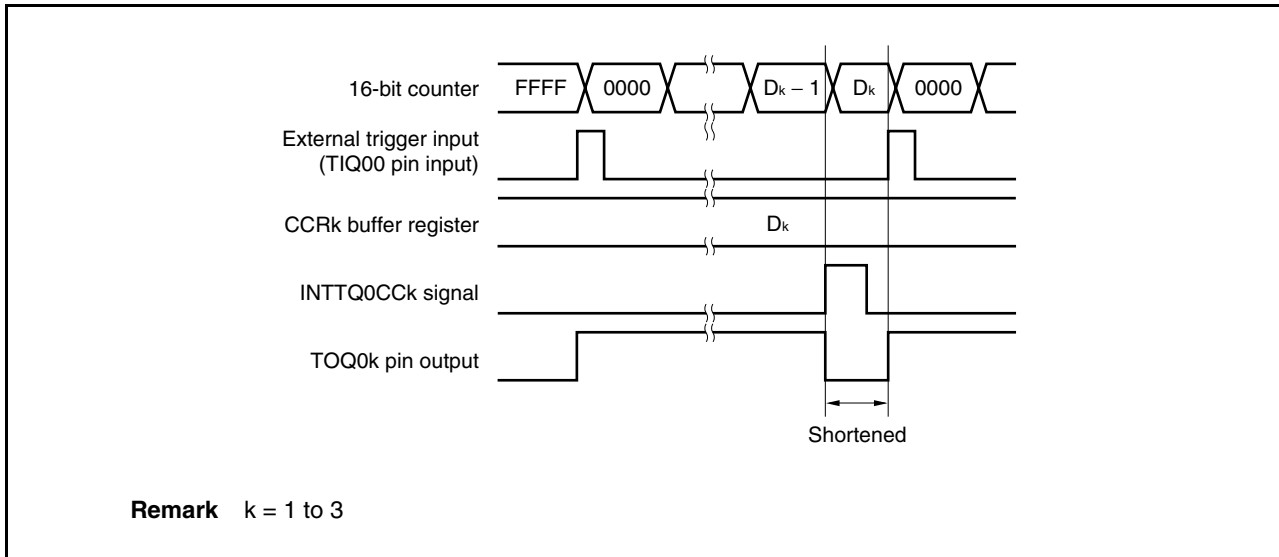
Figure 7-32. Outputting 100% PWM Waveform



(c) Detection of trigger immediately before or after INTTQ0CCk generation

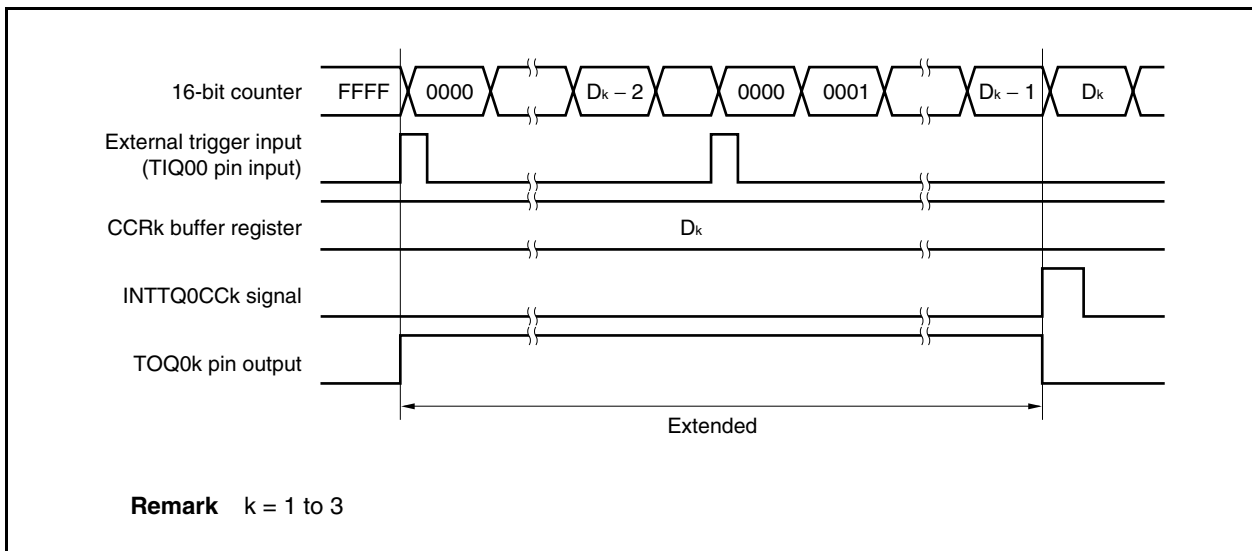
If the trigger is detected immediately after the INTTQ0CCk signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TOQ0k pin is set to the active level, and the counter continues incrementing. Consequently, the inactive period of the PWM waveform is shortened.

Figure 7-33. Detection of Trigger Immediately After INTTQ0CCk Signal Was Generated



If the trigger is detected immediately before the INTTQ0CCk signal is generated, the INTTQ0CCk signal is not generated, and the 16-bit counter is cleared to 0000H and continues incrementing. The output signal of the TOQ0k pin remains active. Consequently, the active period of the PWM waveform is extended.

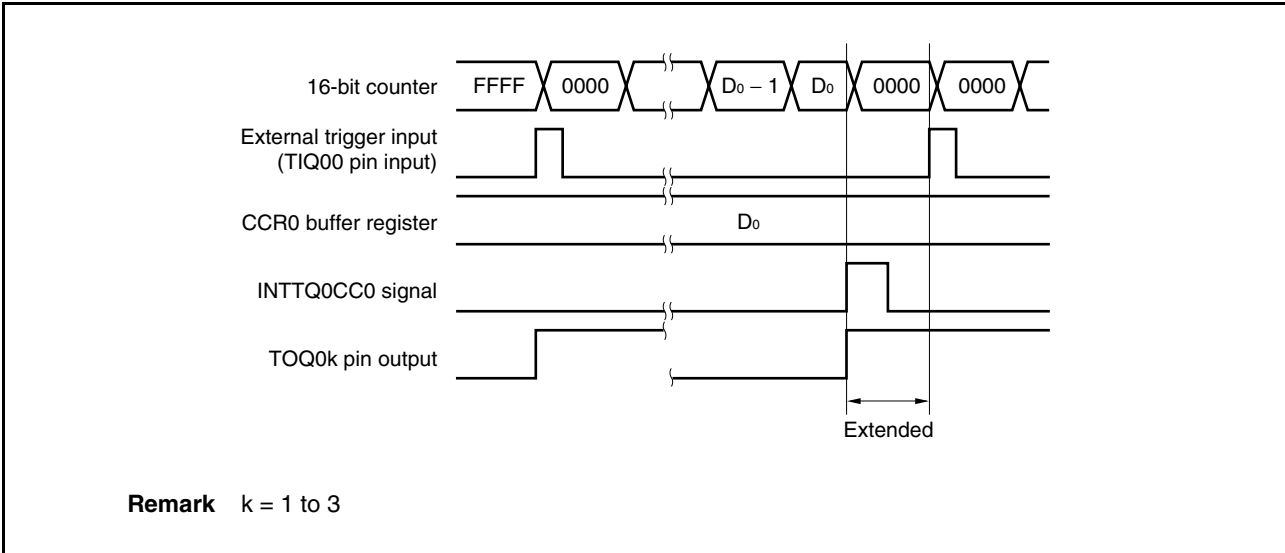
Figure 7-34. Detection of Trigger Immediately Before INTTQ0CCk Signal Is Generated



(d) Detection of trigger immediately before or after INTTQ0CC0 generation

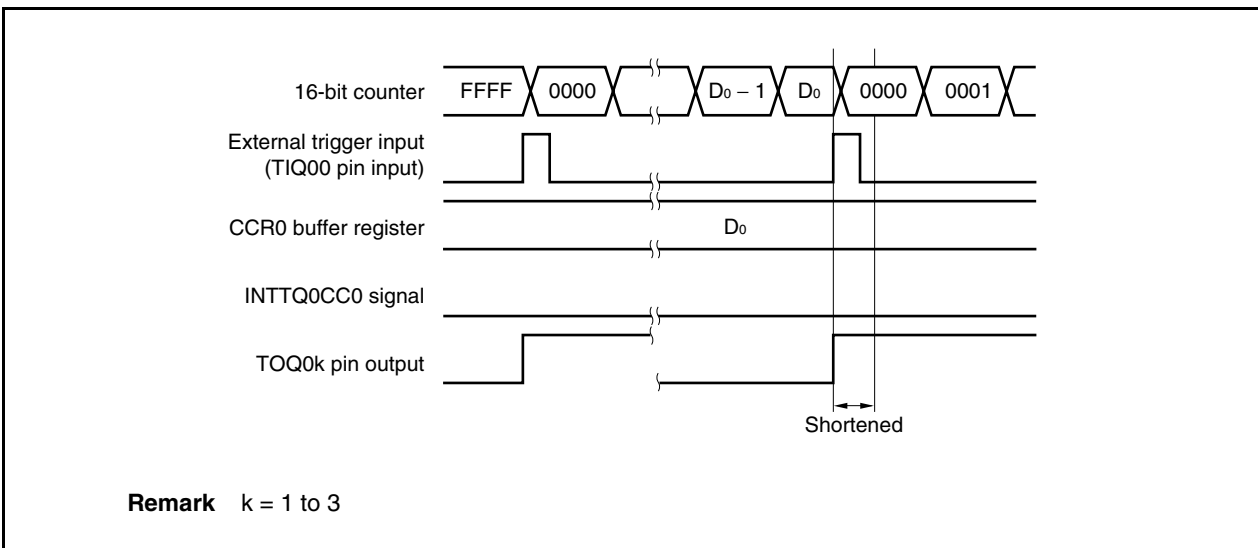
If the trigger is detected immediately after the INTTQ0CC0 signal is generated, the 16-bit counter is cleared to 0000H and continues incrementing. Therefore, the active period of the TOQ0k pin is extended by the amount of time between the generation of the INTTQ0CC0 signal and the detection of the trigger.

Figure 7-35. Detection of Trigger Immediately After INTTQ0CC0 Signal Was Generated



If the trigger is detected immediately before the INTTQ0CC0 signal is generated, the INTTQ0CC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOQ0k pin output is set to the active level, and the counter continues incrementing. Consequently, the inactive period of the PWM waveform is shortened.

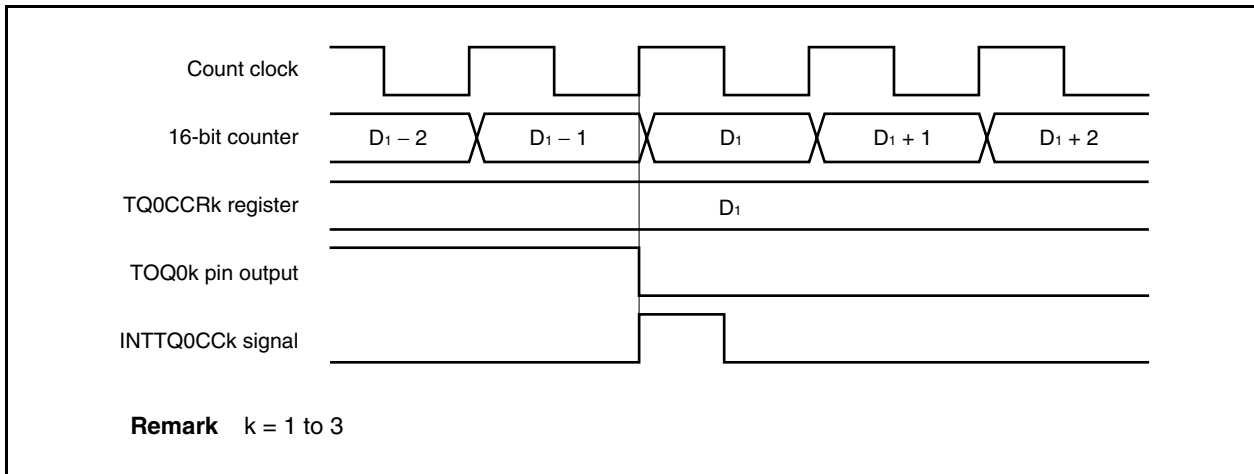
Figure 7-36. Detection of Trigger Immediately Before INTTQ0CC0 Signal Is Generated



(e) Timing of generating the compare match interrupt request signal (INTTQ0CCk)

In the external trigger pulse output mode, the INTTQ0CCk signal is generated when the value of the 16-bit counter matches the value of the TQ0CCRk register.

Figure 7-37. Timing of Generating Compare Match Interrupt Signal (INTTQ0CCk)



7.4.4 One-shot pulse output mode (TQ0MD2 to TQ0MD0 bits = 011)

In the one-shot pulse output mode, when the TQ0CTL0.TQ0CE bit is set to 1, TMQ0 waits for a trigger, which is the valid edge of the external trigger input, and starts incrementing when this trigger is detected. TMQ0 then outputs a one-shot pulse from the TOQ01 to TOQ03 pins.

Instead of the external trigger, a software trigger can also be generated to output the pulse. When the software trigger is used, the TOQ00 pin outputs the active level signal while the 16-bit counter is incrementing, and the inactive level signal when the counter is stopped (waiting for a trigger).

- Remarks 1.** For how to set the TIQ00 and TOQ00 to TOQ03 pins, see **Table 7-2 Pins Used by TMQ0** and **Table 4-17 Settings When Pins Are Used for Alternate Functions**.
- 2.** For how to enable the INTTQ0CC0 to INTTQ0CC3 interrupt signals, see **CHAPTER 19 INTERRUPT SERVICING/EXCEPTION PROCESSING FUNCTION**.

Figure 7-38. Configuration of TMQ0 in One-Shot Pulse Output Mode

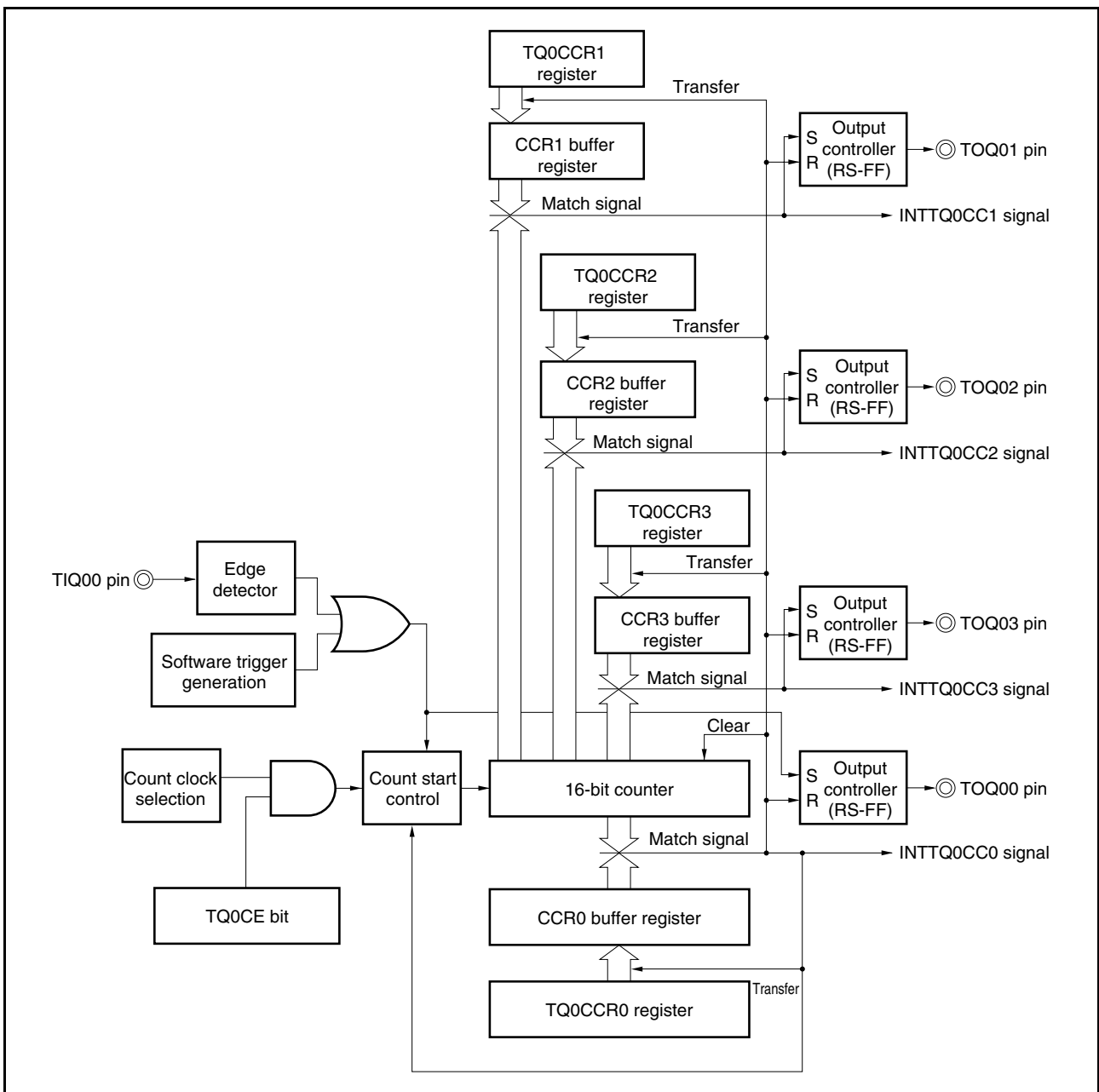
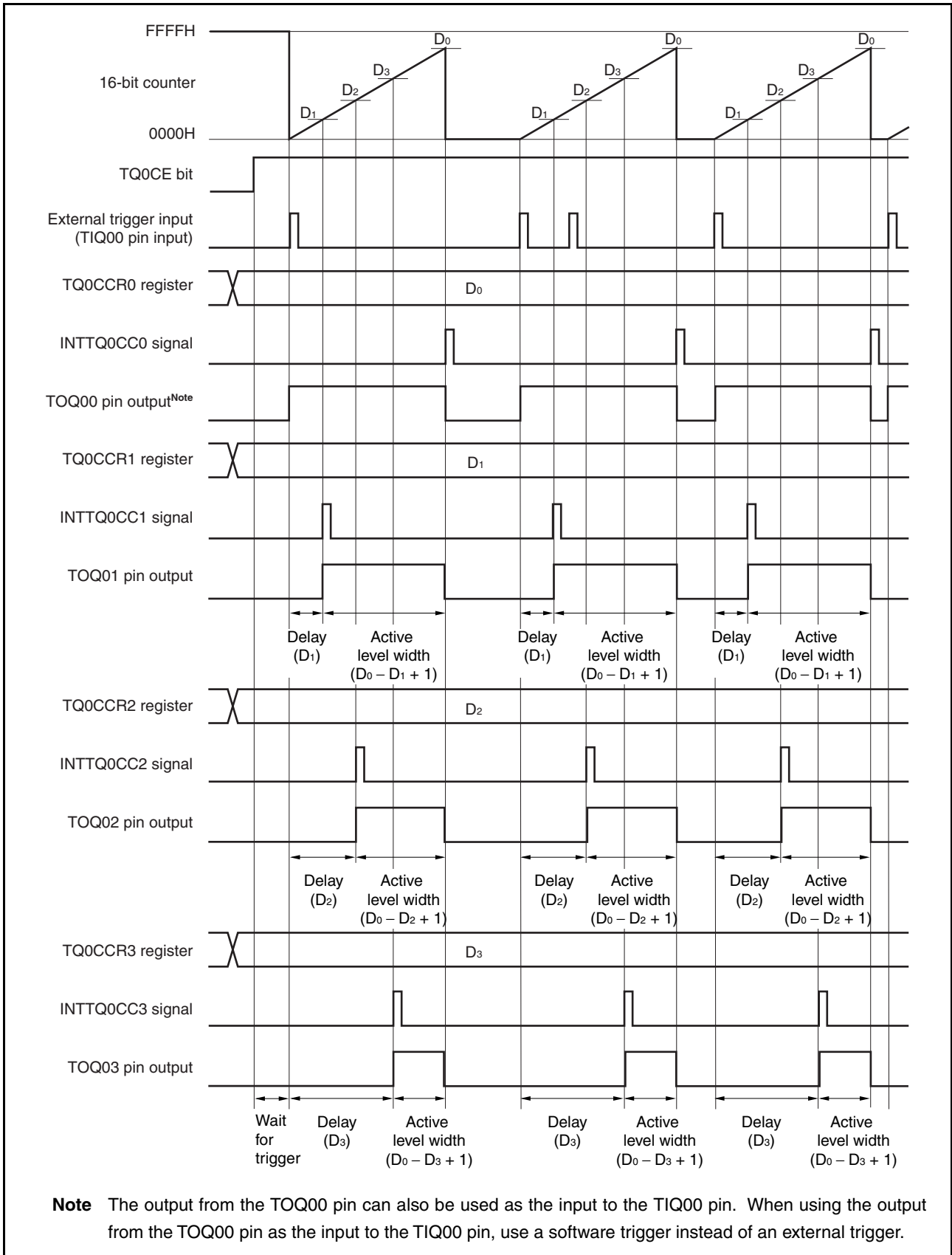


Figure 7-39. Basic Timing of Operations in One-Shot Pulse Output Mode



When the TQ0CE bit is set to 1, TMQ0 waits for a trigger. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts incrementing, and outputs a one-shot pulse from the TQ0Qk pin. After the one-shot pulse is output, the 16-bit counter is set to 0000H, stops incrementing, and waits for a trigger. 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 TQ0CCRk register) × Count clock cycle

Active level width = (Set value of TQ0CCR0 register – Set value of TQ0CCRk register + 1) × Count clock cycle

The INTTQ0CC0 compare match interrupt request signal is generated when the 16-bit counter increments next time after its value matches the value of the CCR0 buffer register. The INTTQ0CCk compare match interrupt request signal is generated when the value of the 16-bit counter matches the value of the CCRk buffer register.

Either the valid edge of the external trigger input signal or setting the software trigger (TQ0CTL1.TQ0EST bit) to 1 is used as the trigger.

Remark k = 1 to 3

Figure 7-40. Register Settings in One-Shot Pulse Output Mode (1/3)

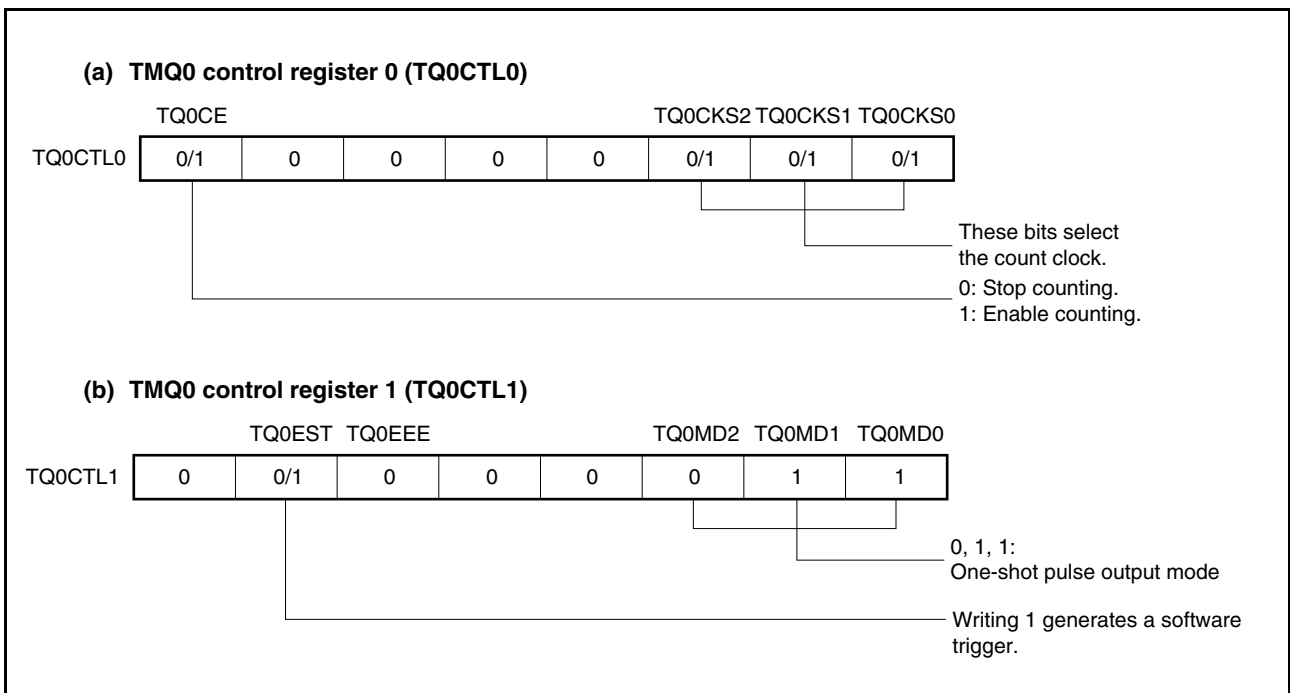


Figure 7-40. Register Settings in One-Shot Pulse Output Mode (2/3)

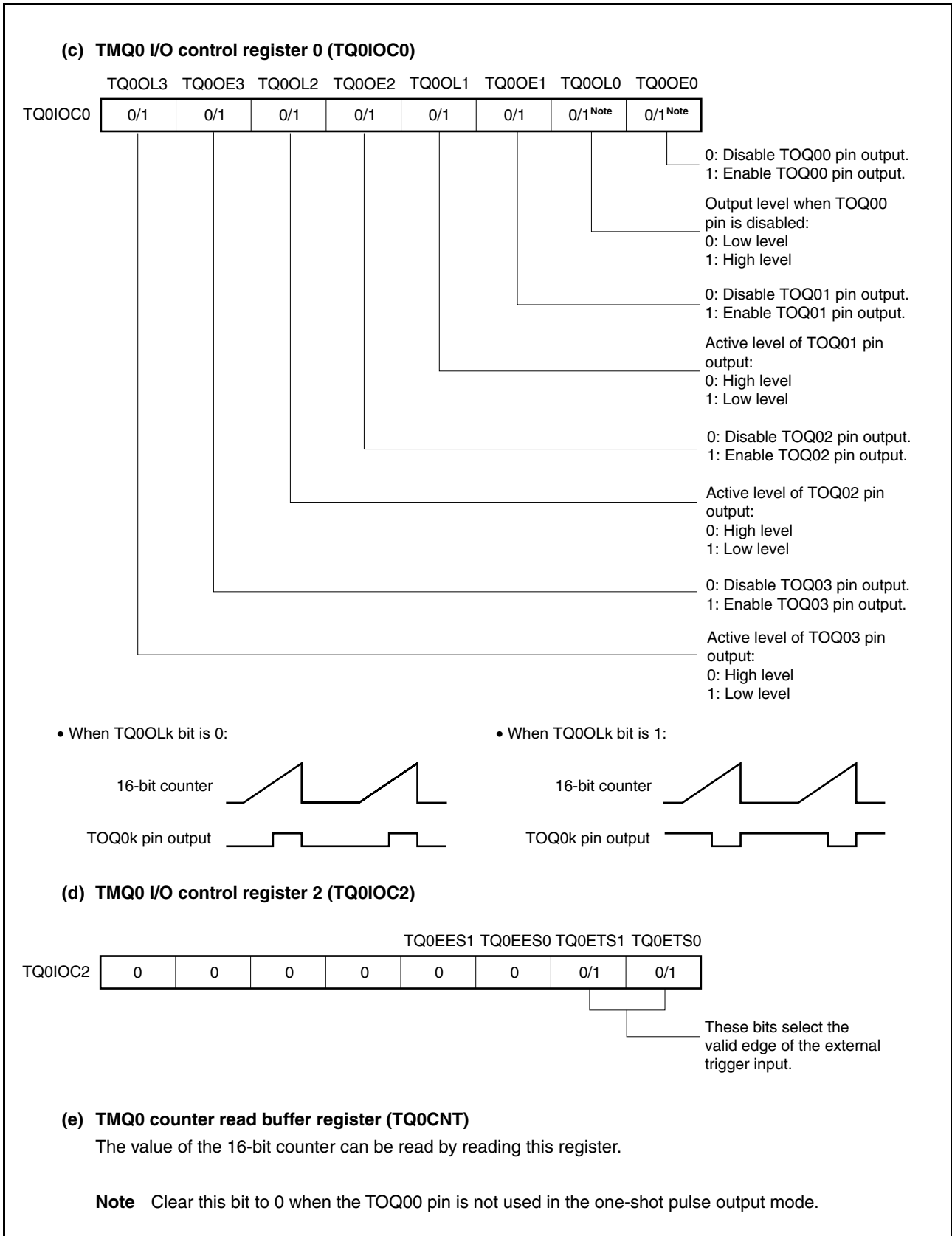


Figure 7-40. Register Settings in One-Shot Pulse Output Mode (3/3)**(f) TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3)**

If the TQ0CCR0 register is set to D_0 and the TQ0CCR k register is set to D_k , the one-shot pulse is as follows:

One-shot pulse active level width = $(D_0 - D_k + 1) \times$ Count clock cycle

One-shot pulse output delay period = $D_k \times$ Count clock cycle

Caution One-shot pulses are not output from the TOQ0 k pin in the one-shot pulse output mode if the value of the TQ0CCR k register is greater than the value of the TQ0CCR0 register.

Remarks 1. TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the one-shot pulse output mode.

2. $k = 1$ to 3

(1) Operations in one-shot pulse output mode

Figure 7-41. Timing and Processing of Operations in One-Shot Pulse Output Mode (1/2)

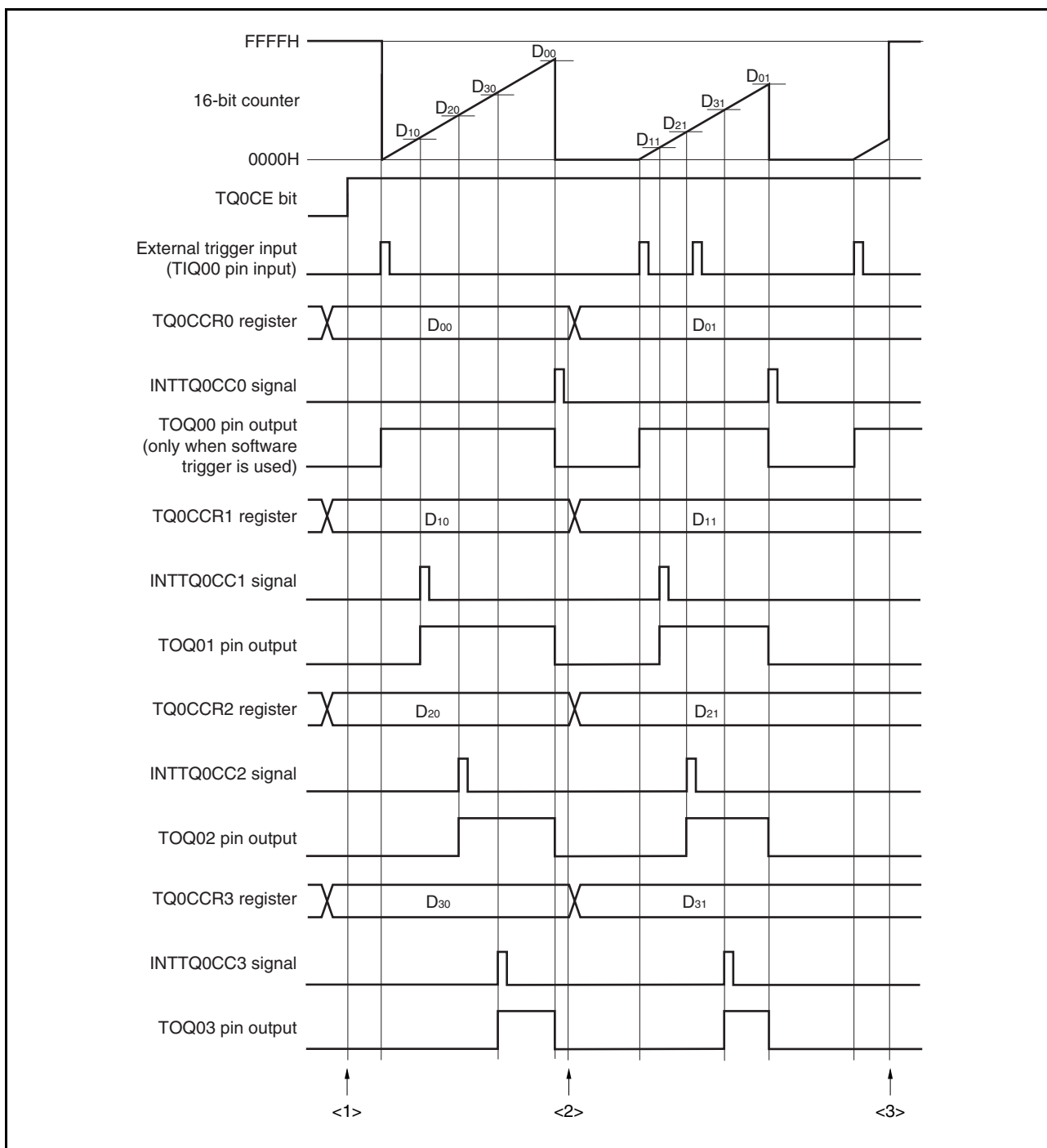
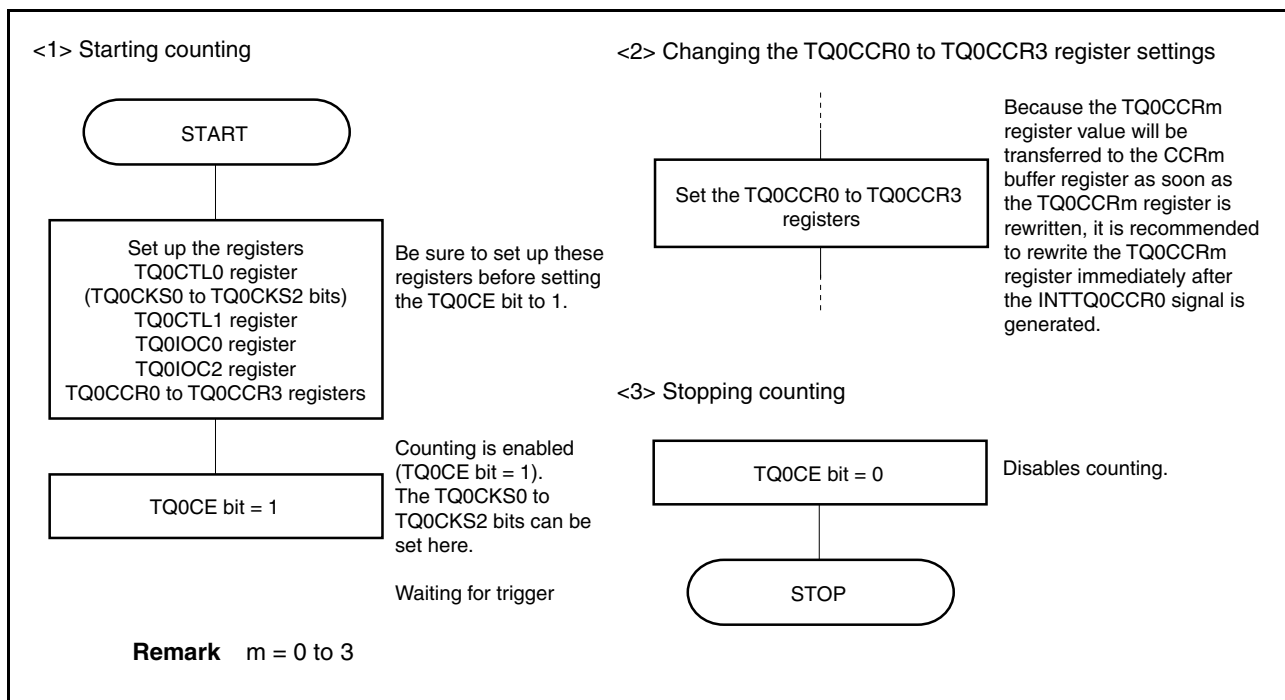


Figure 7-41. Timing and Processing of Operations in One-Shot Pulse Output Mode (2/2)



(2) Using one-shot pulse mode

(a) Rewriting the TQ0CCRm register

When rewriting the value of the TQ0CCRm register to a smaller value, stop counting first and then change the set value.

When changing the value of the TQ0CCR0 register from D_{00} to D_{01} and the value of the TQ0CCRk register from D_{k0} to D_{k1} , if the registers are rewritten under any of the following conditions, a one-shot pulse will not be output as expected.

Condition 1 When rewriting the TQ0CCR0 register, if:

$D_{00} > D_{01}$ or,

$D_{00} < 16\text{-bit counter value} < D_{01}$

In the case of condition 1, the 16-bit counter will not be cleared and will overflow in the cycle in which the new value is being written. The counter will be cleared for the first time at the newly written value (D_{01}).

Condition 2 When rewriting the TQ0CCRk register, if:

$D_{k0} > D_{k1}$ or,

$D_{k0} < 16\text{-bit counter value} < D_{k1}$

In the case of condition 2, the TOQ0k pin output cannot be inverted to the active level in the cycle in which the new value is being written.

An example of what happens when condition 1 and condition 2 are satisfied in the same cycle is shown in Figure 7-42.

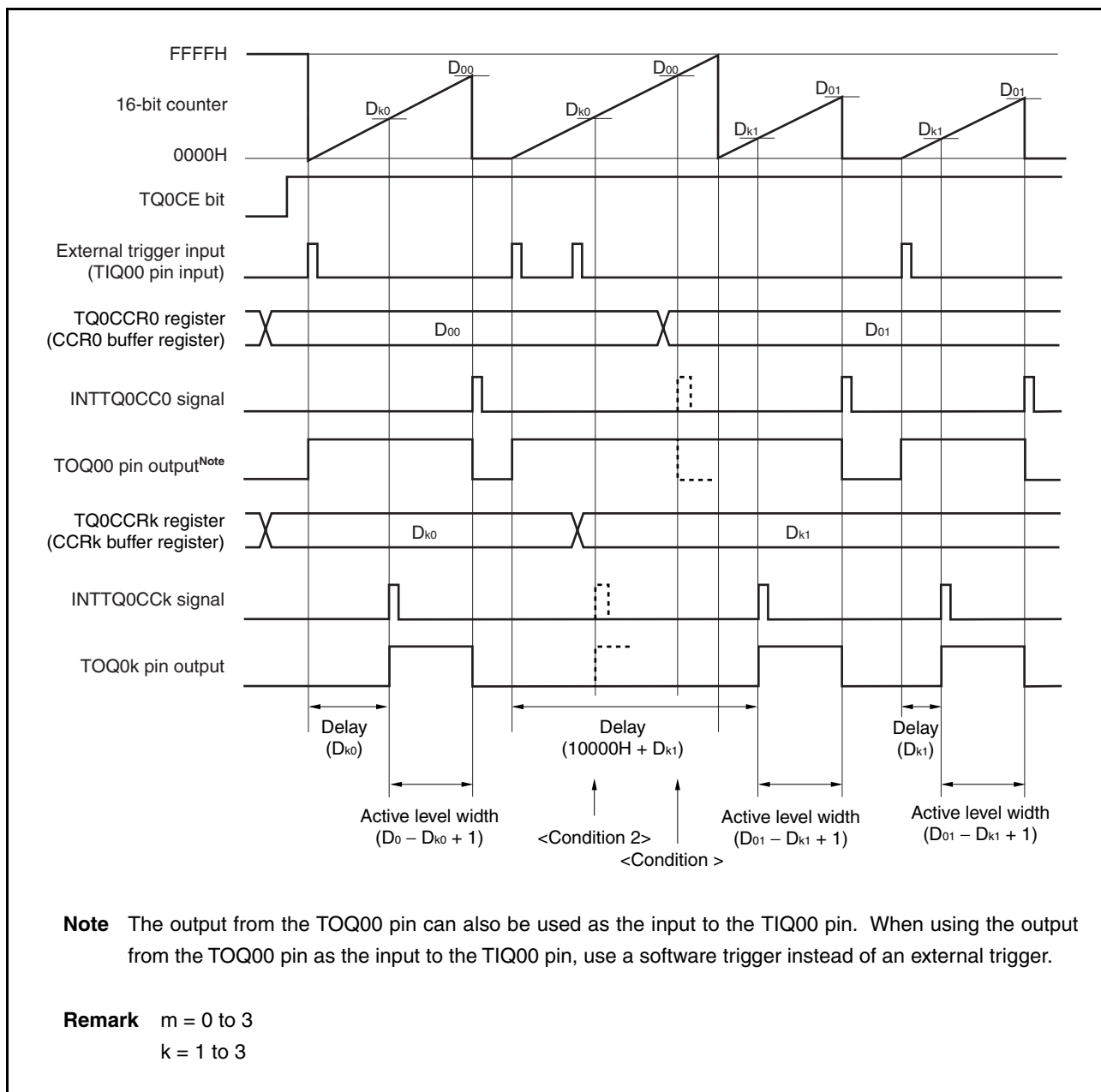
The 16-bit counter increments up to FFFFH, overflows, and starts incrementing again from 0000H.

When the 16-bit counter value matches D_{k1} , the INTTQ0CCk signal is generated and the TOQ0k pin output is set to the active level. Subsequently, when the 16-bit counter value matches D_{01} , the INTTQ0CC0 signal is generated, the TOQ0k pin output is set to the inactive level, and the counter stops incrementing.

Remark $m = 0$ to 3

$K = 1$ to 3

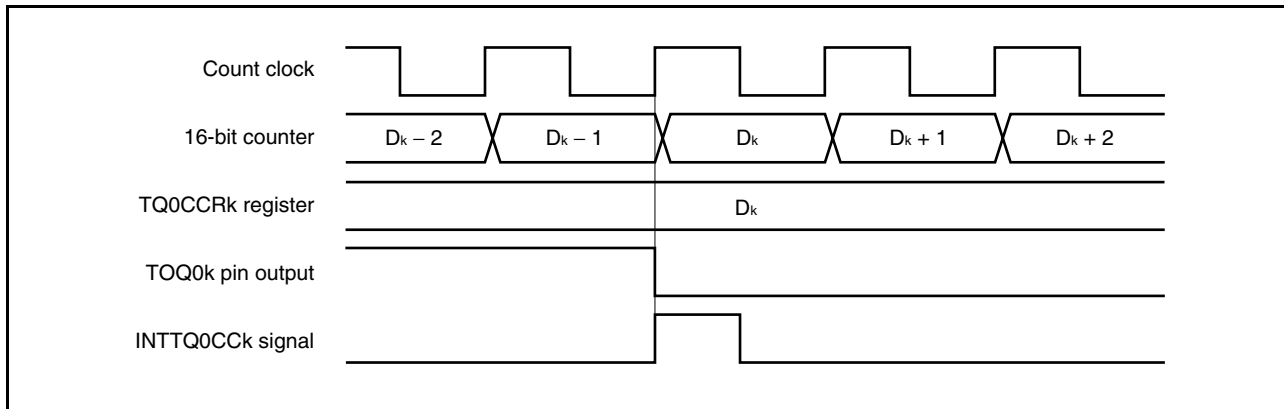
Figure 7-42. Rewriting TQ0CCRm Register



(b) Timing of generating the compare match interrupt request signal (INTTQ0CCK)

In the one-shot pulse output mode, the INTTQ0CCK signal is generated when the value of the 16-bit counter matches the value of the TQ0CCRk register.

Figure 7-43. Timing of Generating Compare Match Interrupt Signal (INTTQ0CCK)



Remark k = 1 to 3

7.4.5 PWM output mode (TQ0MD2 to TQ0MD0 bits = 100)

In the PWM output mode, when the TQ0CTL0.TQ0CE bit is set to 1, TMQ0 outputs a PWM waveform from the TOQ01 to TOQ03 pins.

A pulse that has one cycle of the PWM waveform as half its cycle can also be output from the TOQ00 pin.

- Remarks 1.** For how to set the TIQ00 and TOQ00 to TOQ03 pins, see **Table 7-2 Pins Used by TMQ0** and **Table 4-17 Settings When Pins Are Used for Alternate Functions**.
- 2.** For how to enable the INTTQ0CC0 to INTTQ0CC3 interrupt signals, see **CHAPTER 19 INTERRUPT SERVICING/EXCEPTION PROCESSING FUNCTION**.

Figure 7-44. Configuration of TMQ0 in PWM Output Mode

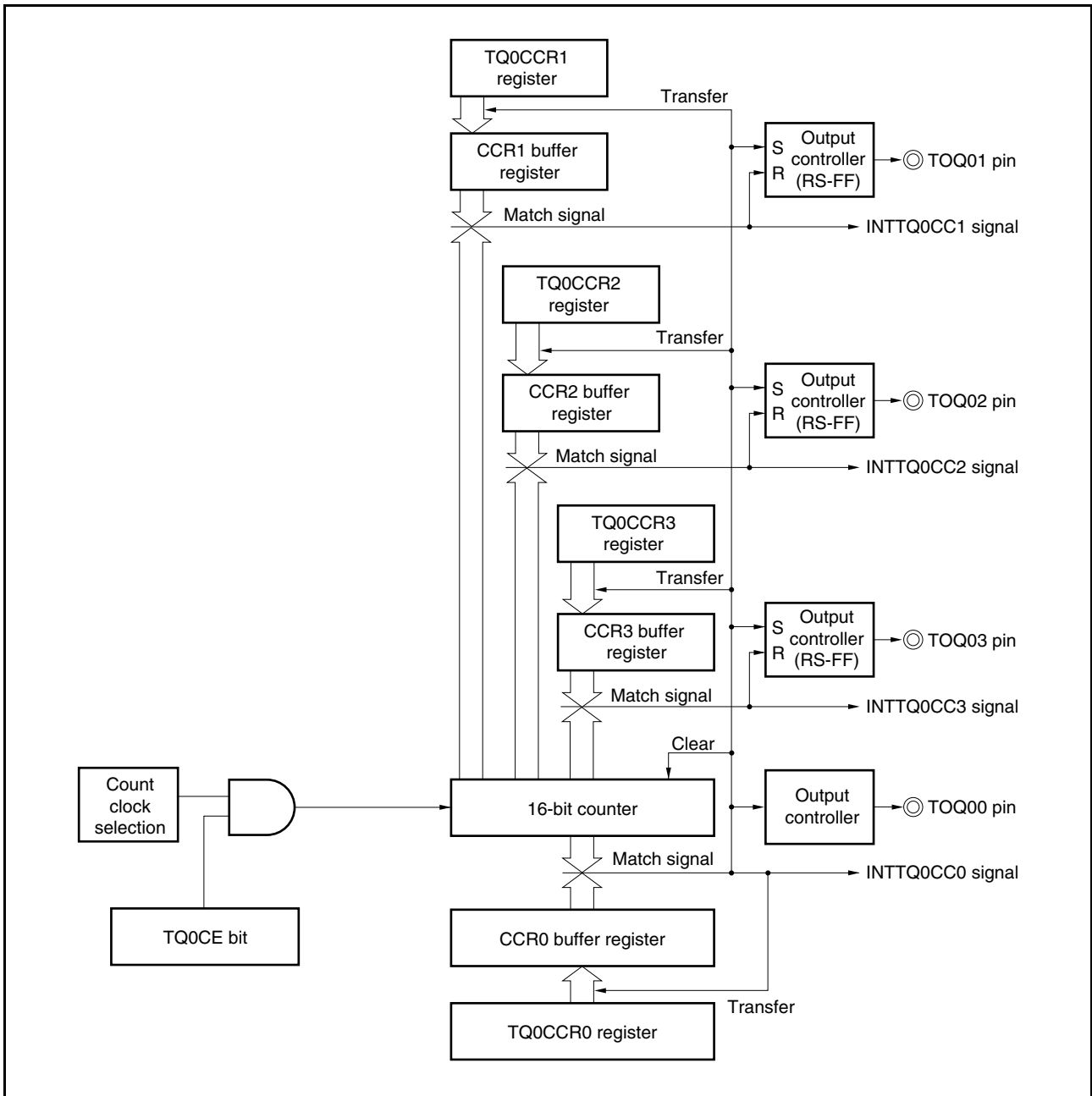
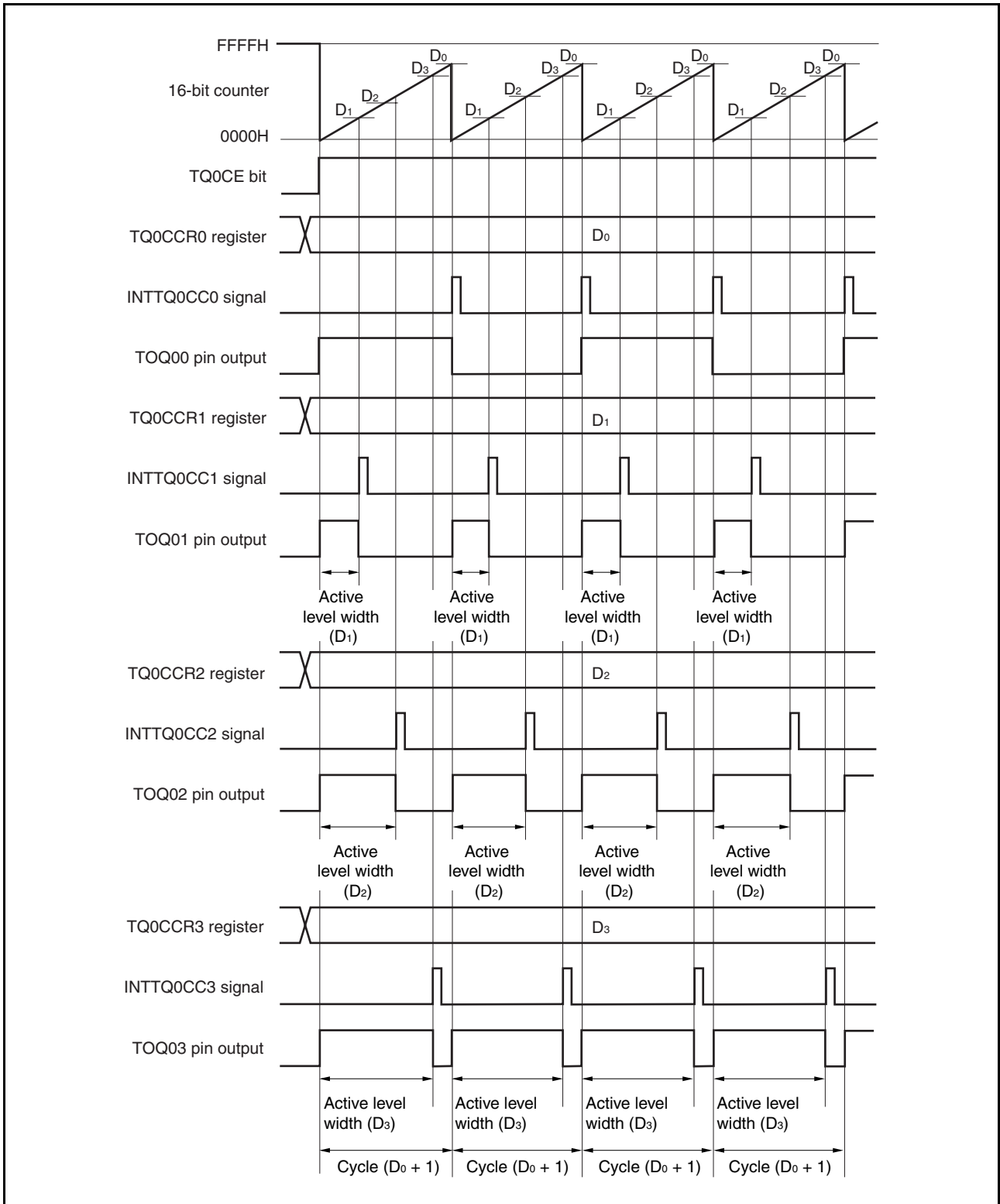


Figure 7-45. Basic Timing of Operations in PWM Output Mode



When the TQ0CE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts incrementing, and outputs a PWM waveform from the TOQ0k pin.

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

$$\text{Active level width} = (\text{Set value of TQ0CCRk register}) \times \text{Count clock cycle}$$

$$\text{Cycle} = (\text{Set value of TQ0CCR0 register} + 1) \times \text{Count clock cycle}$$

$$\text{Duty factor} = (\text{Set value of TQ0CCRk register}) / (\text{Set value of TQ0CCR0 register} + 1)$$

The PWM waveform can be changed by rewriting the TQ0CCRm register while the counter is incrementing. The newly written value is reflected when the value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

The INTTQ0CC0 compare match interrupt request signal is generated when the 16-bit counter increments next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The INTTQ0CCk compare match interrupt request signal is generated when the value of the 16-bit counter matches the value of the CCRk buffer register.

Remark k = 1 to 3
m = 0 to 3

Figure 7-46. Register Settings in PWM Output Mode (1/3)

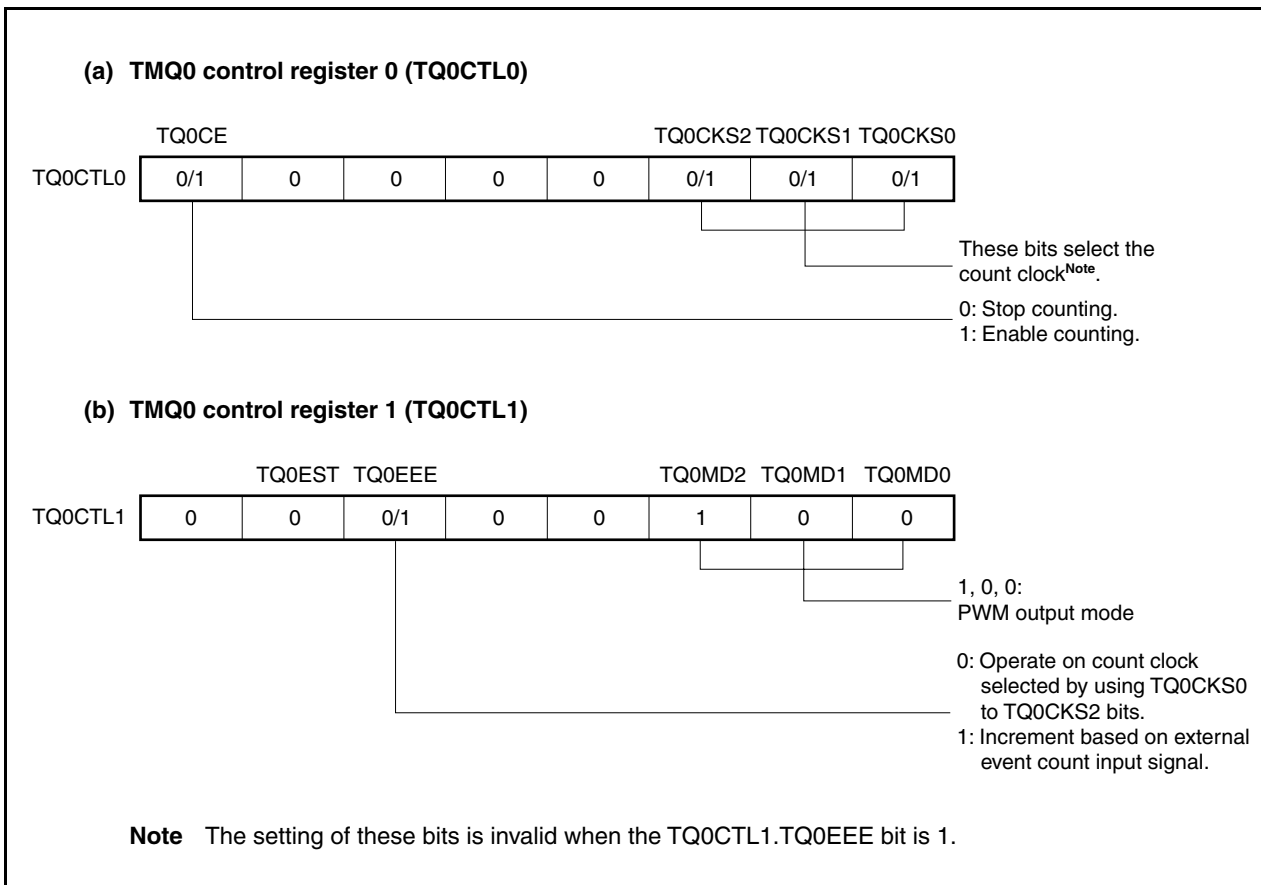


Figure 7-46. Register Settings in PWM Output Mode (2/3)

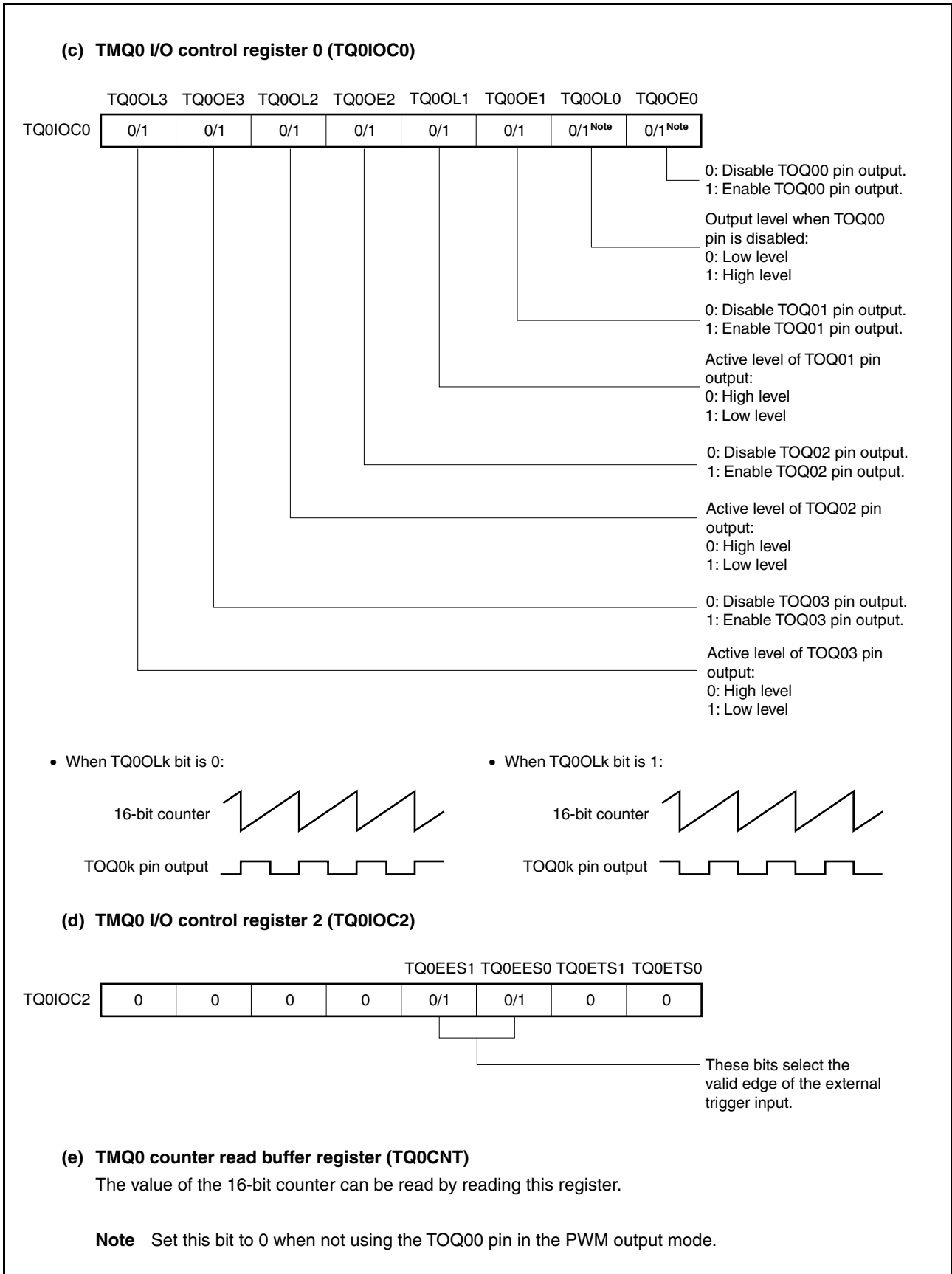


Figure 7-46. Register Settings in PWM Output Mode (3/3)**(f) TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3)**

If the TQ0CCR0 register is set to D_0 and the TQ0CCR k register is set to D_k , the PWM waveform is as follows:

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

PWM waveform active level width = $D_k \times$ Count clock cycle

Remarks 1. TMQ0 I/O control register 1 (TQ0IOC1) and TMQ0 option register 0 (TQ0OPT0) are not used in the PWM output mode.

2. Updating TMQ0 capture/compare register 2 (TQ0CCR2) and TMQ0 capture/compare register 3 (TQ0CCR3) is enabled by writing to TMQ0 capture/compare register 1 (TQ0CCR1).

(1) Operations in PWM output mode

Figure 7-47. Timing and Processing of Operations in PWM Output Mode (1/2)

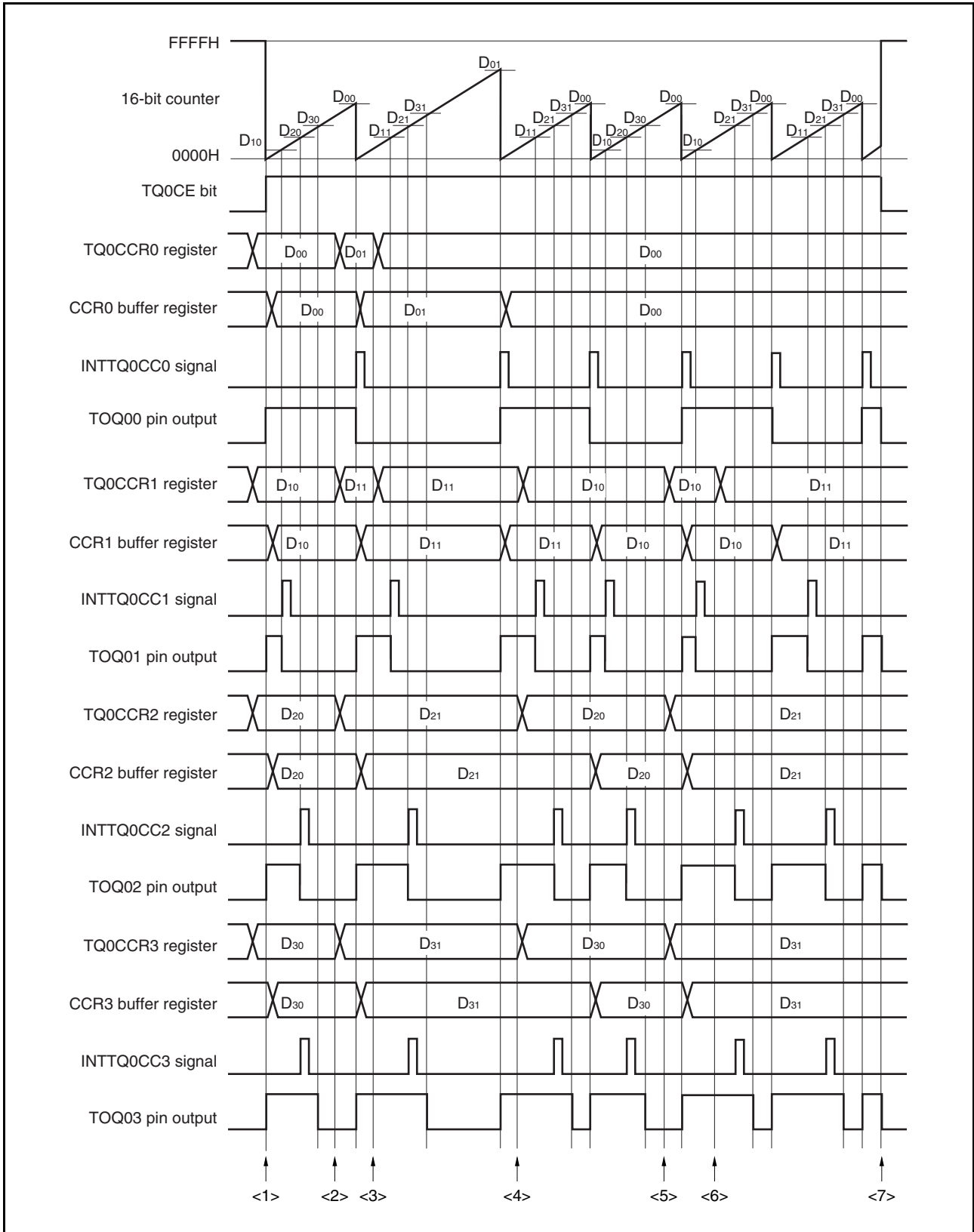
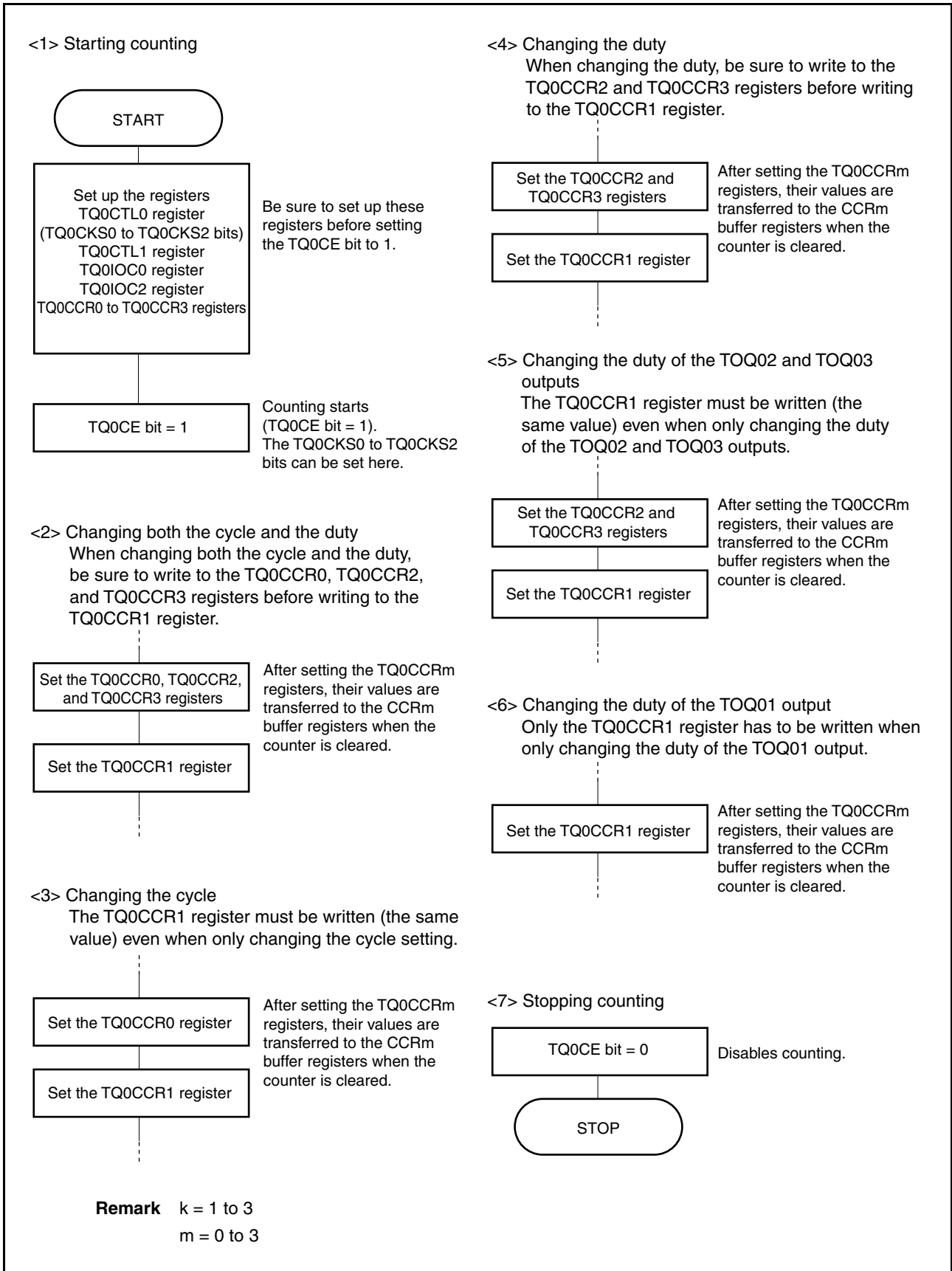


Figure 7-47. Timing and Processing of Operations in PWM Output Mode (2/2)

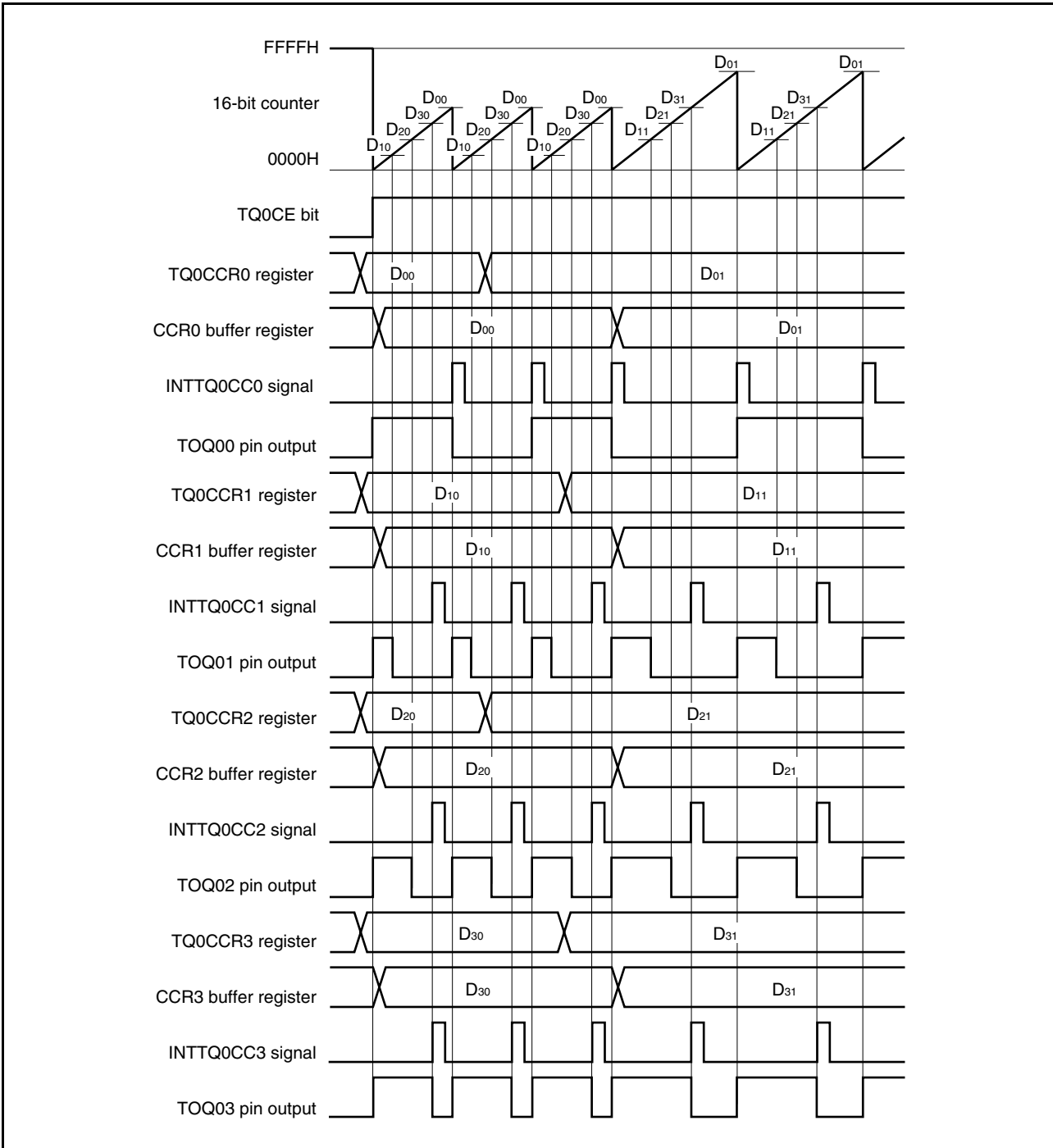


(2) Using PWM output mode

(a) Changing the PWM waveform while the counter is incrementing

To change the PWM waveform while the counter is incrementing, write to the TQ0CCR1 register after changing the waveform setting. When rewriting the TQ0CCRm register after writing to the TQ0CCR1 register, do so after the INTTQ0CC0 signal has been detected.

Figure 7-48. Changing PWM Waveform While Counter Is Incrementing



In order to transfer data from the TQ0CCRm register to the CCRm buffer register, the TQ0CCR1 register must be written.

After data is written to the TQ0CCR1 register, the value written to the TQ0CCRm register is transferred to the CCRm buffer register in synchronization with clearing of the 16-bit counter, and is used as the value to be compared with the 16-bit counter value.

- <1> To change both the cycle and active level width of the PWM waveform, first set the cycle to the TQ0CCR0 register and then set the active level width to the TQ0CCR2 and TQ0CCR3 registers, before setting the active level width to the TQ0CCR1 register.
- <2> To change only the cycle of the PWM waveform, first set the cycle to the TQ0CCR0 register, and then write the same value to the TQ0CCR1 register (that is, the same value as the value already specified for the TQ0CCR1 register).
- <3> To change only the active level width (duty factor) of the PWM waveform, first set the active level width to the TQ0CCR2 and TQ0CCR3 registers, and then set the active level width to the TQ0CCR1 register.
- <4> To change only the active level width (duty factor) of the PWM waveform output from the TOQ01 pin, only the TQ0CCR1 register has to be set.
- <5> To change only the active level width (duty factor) of the PWM waveform output from the TOQ02 and TOQ03 pins, first set the active level width to the TQ0CCR2 and TQ0CCR3 registers, and then write the same value to the TQ0CCR1 register (that is, the same value as the value already specified for the TQ0CCR1 register).

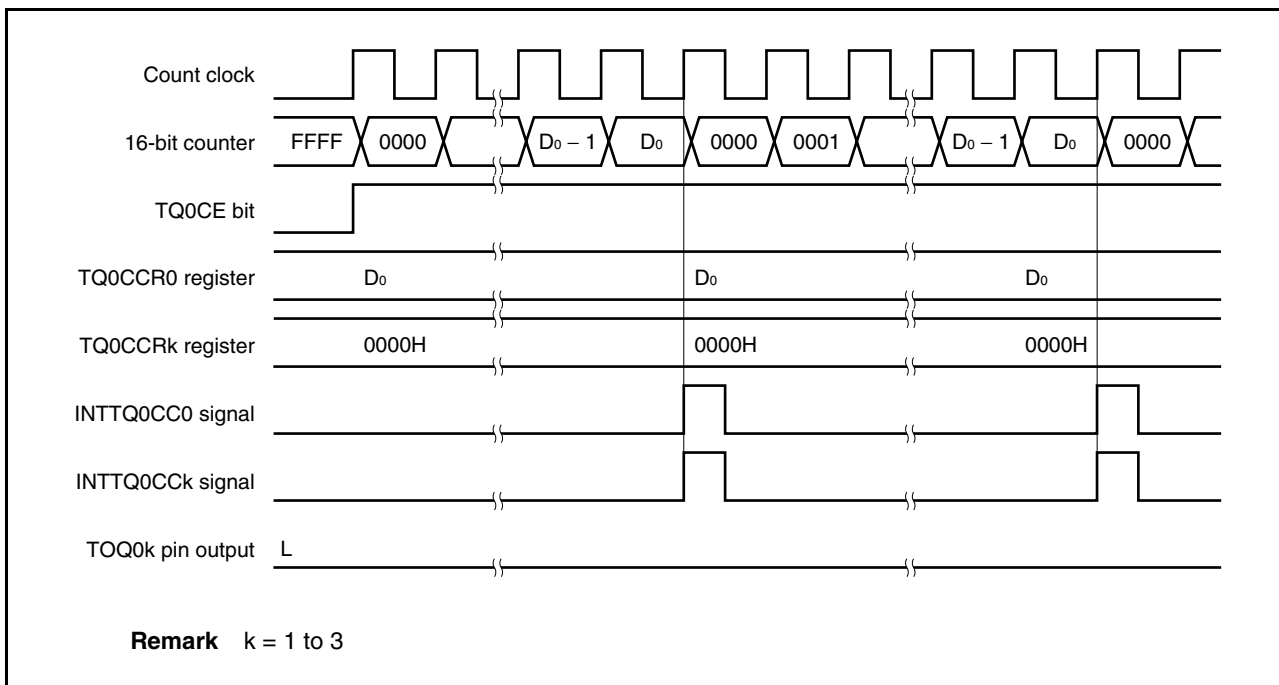
Caution To rewrite the TQ0CCR0 to TQ0CCR3 registers after writing the TQ0CCR1 register, do so after the INTTQ0CC0 signal has been generated; otherwise, the value of the CCRm buffer register may become undefined because the timing of transferring data from the TQ0CCRm register to the CCRm buffer register conflicts with writing the TQ0CCRm register.

Remark m = 0 to 3

(b) Outputting a 0% or 100% PWM waveform

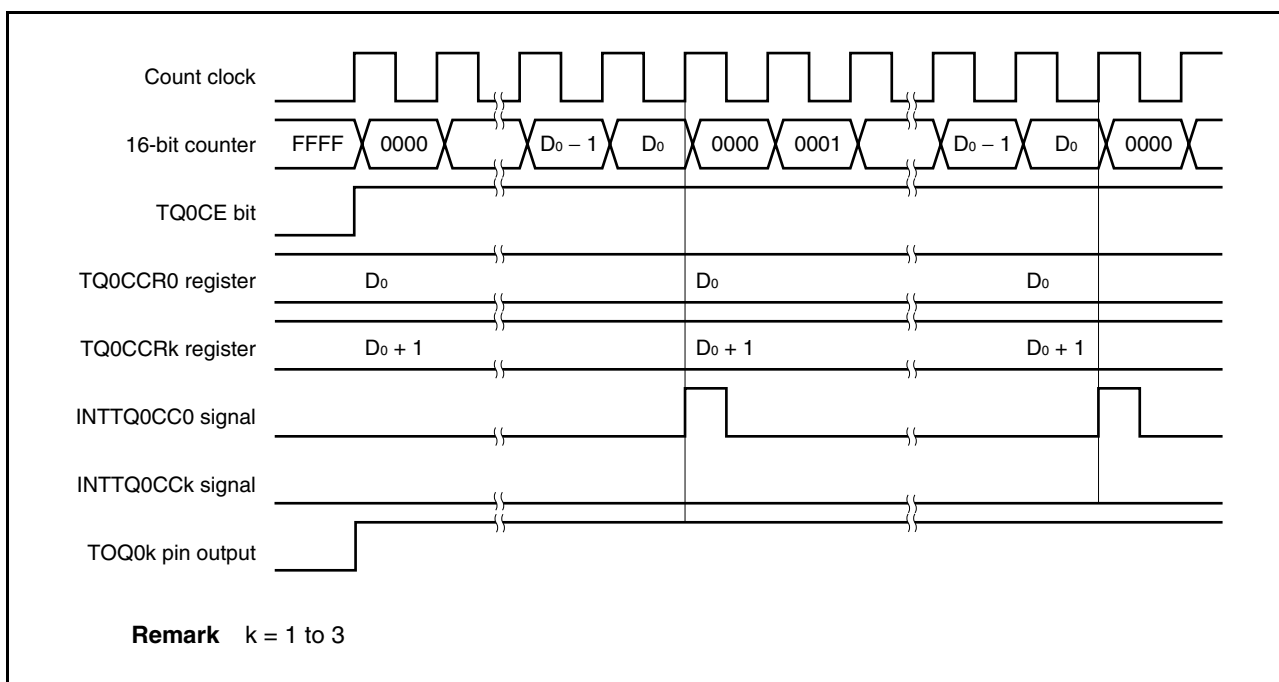
To output a 0% waveform, clear the TQ0CCRk register to 0000H.

Figure 7-49. Outputting 0% PWM Waveform



To output a 100% waveform, set the value of TQ0CCR0 register + 1 to the TQ0CCRk register. If the value of the TQ0CCR0 register is FFFFH, a 100% waveform cannot be output.

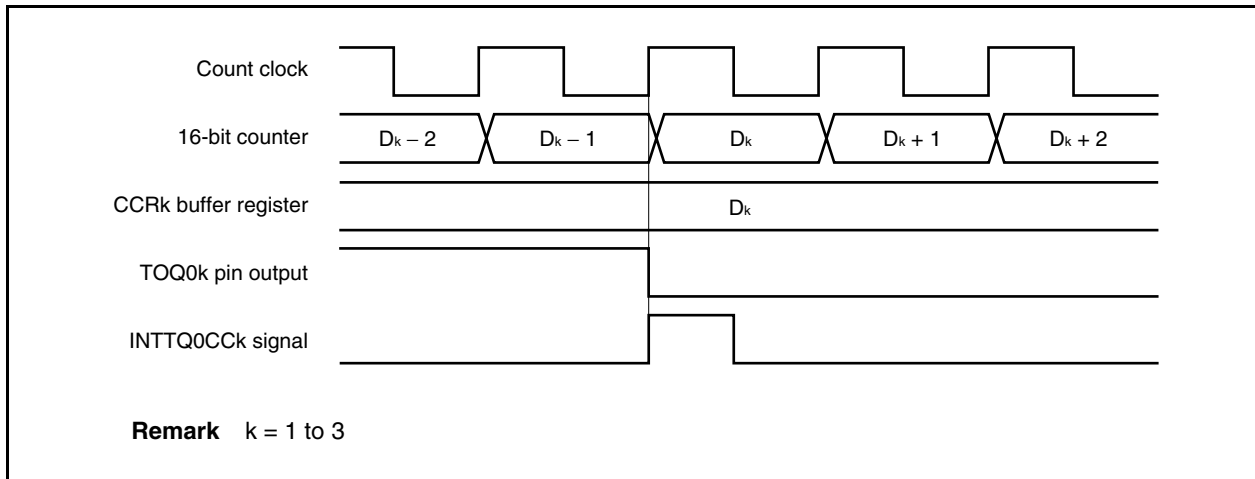
Figure 7-50. Outputting 100% PWM Waveform



(c) Timing of generating the compare match interrupt request signal (INTTQ0CCK)

In the PWM output mode, the INTTQ0CCK signal is generated when the value of the 16-bit counter matches the value of the TQ0CCRk register.

Figure 7-51. Timing of Generating Compare Match Interrupt Request Signal (INTTQ0CCK)

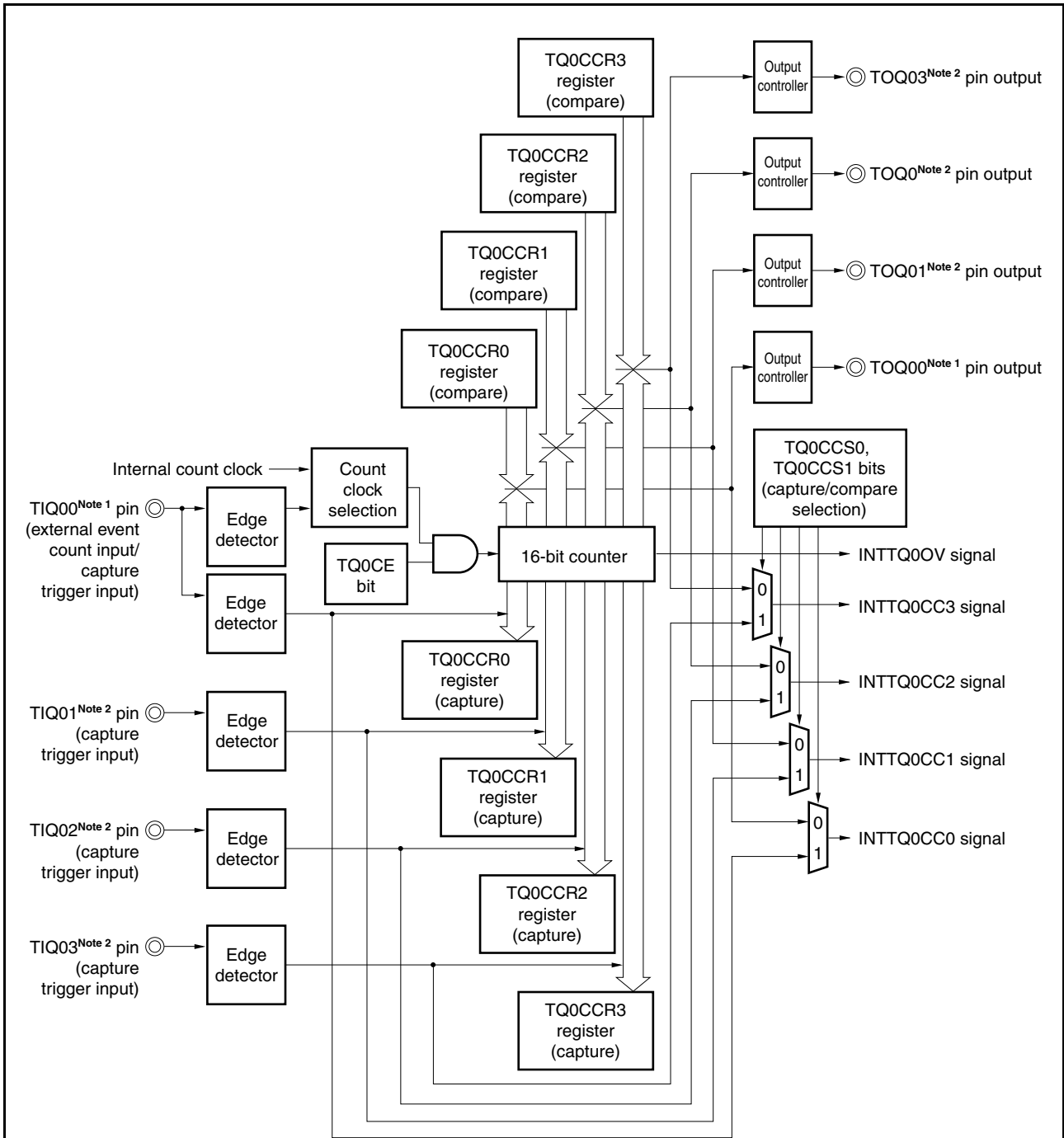


7.4.6 Free-running timer mode (TQ0MD2 to TQ0MD0 bits = 101)

In the free-running timer mode, TMQ0 starts incrementing when the TQ0CTL0.TQ0CE bit is set to 1. At this time, the TQ0CCRm register can be used as a compare register or a capture register, according to the setting of the TQ0OPT0.TQ0CCS0 and TQ0OPT0.TQ0CCS1 bits.

- Remarks**
1. For how to set the TIQ0m and TOQ0m pins, see **Table 7-2 Pins Used by TMQ0** and **Table 4-17 Settings When Pins Are Used for Alternate Functions**.
 2. For how to enable the INTTQ0CCm interrupt signal, see **CHAPTER 19 INTERRUPT SERVICING/EXCEPTION PROCESSING FUNCTION**.
 3. m = 0 to 3

Figure 7-52. Configuration of TMQ0 in Free-Running Timer Mode



- Notes**
1. The external event count input/capture trigger input pin (TIQ00) can also be used as the timer output pin (TOQ00); however, only one of these functions can be used at a time.
 2. The capture trigger input pin (TIQ0k) can also be used as the timer output pin (TOQ0k); however, only one of these functions can be used at a time. (k = 1 to 3.)

- Compare operation

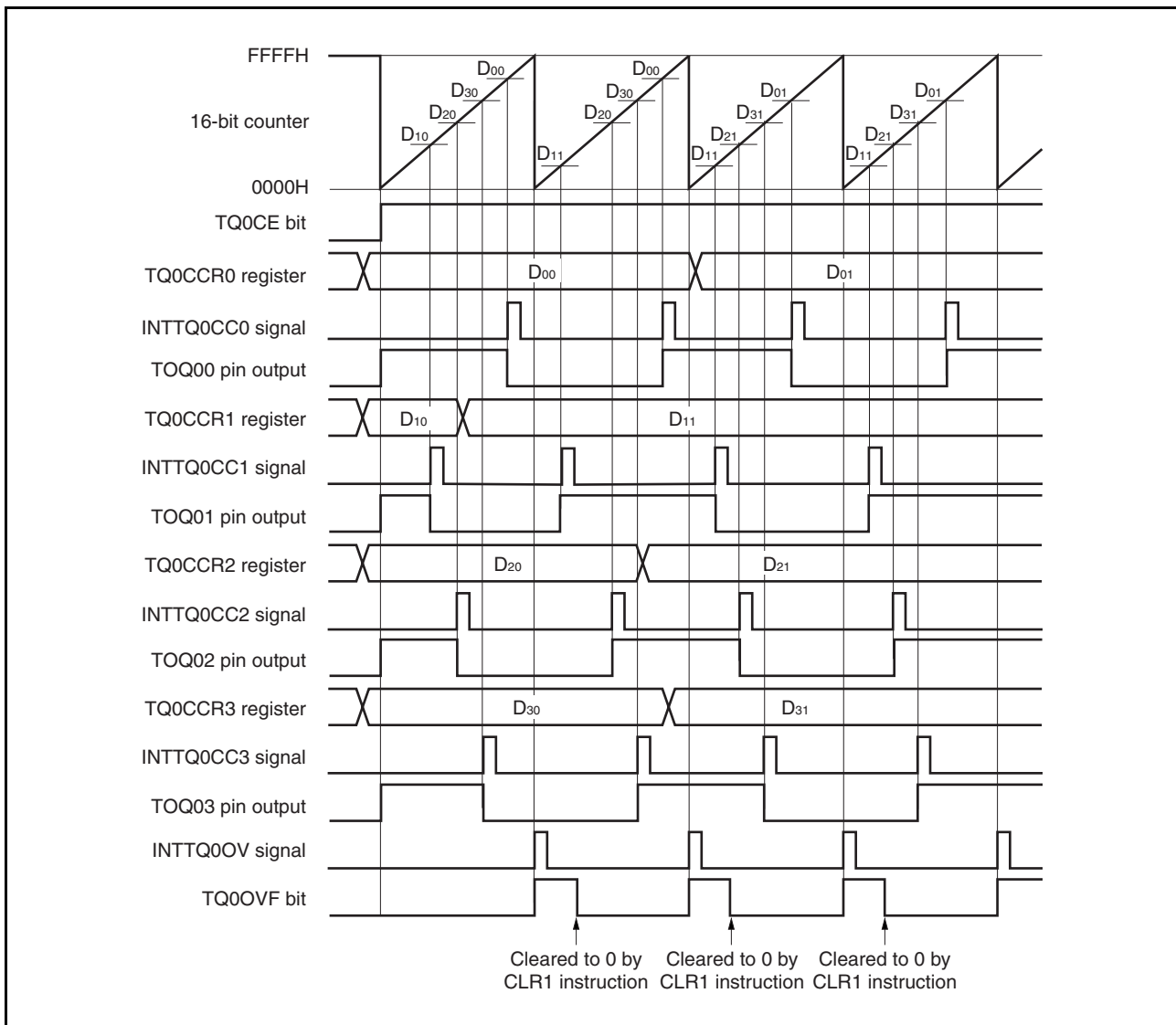
When the TQ0CE bit is set to 1, TMQ0 starts incrementing, and the output signals of the TOQ00 to TOQ03 pins are inverted. When the value of the 16-bit counter later matches the set value of the TQ0CCRm register, a compare match interrupt request signal (INTTQ0CCm) is generated, and the output signal of the TOQ0m pin is inverted.

The 16-bit counter continues incrementing in synchronization with the count clock. Once the counter reaches FFFFH, it generates an overflow interrupt request signal (INTTQ0OV) at the next clock, is cleared to 0000H, and continues incrementing. At this time, the overflow flag (the TQ0OPT0.TQ0OVF bit) is also set to 1. The overflow flag must be cleared to 0 by executing a CLR1 software instruction.

The TQ0CCRm register can be rewritten while the counter is incrementing. If it is rewritten, the new value is immediately applied, and compared with the count value.

Remark m = 0 to 3

Figure 7-53. Basic Timing of Operations in Free-Running Timer Mode (Compare Function)



• Capture operation

When the TQ0CE bit is set to 1, the 16-bit counter starts incrementing. When it is detected that a valid edge as been input to the TIQ0m pin, the value of the 16-bit counter is stored in the TQ0CCRm register, and a capture interrupt request signal (INTTQ0CCm) is generated.

The 16-bit counter continues incrementing in synchronization with the count clock. When the counter reaches FFFFH, it generates an overflow interrupt request signal (INTTQ0OV) at the next clock, is cleared to 0000H, and continues incrementing. At this time, the overflow flag (the TQ0OPT0.TQ0OVF bit) is also set to 1. The overflow flag must be cleared to 0 by executing a CLR1 software instruction.

Remark m = 0 to 3

Figure 7-54. Basic Timing of Operations in Free-Running Timer Mode (Capture Function)

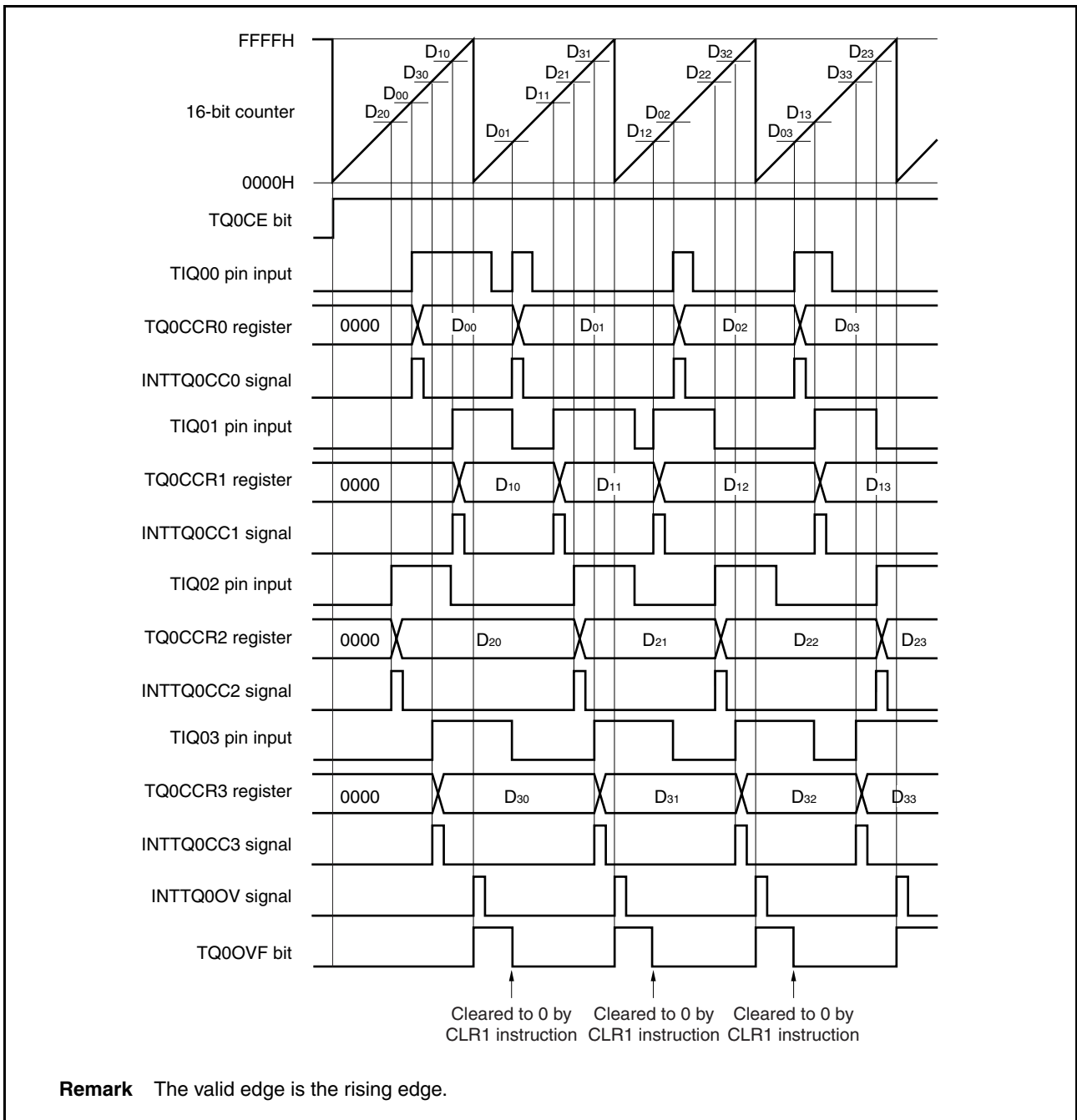


Figure 7-55. Register Settings in Free-Running Timer Mode (1/3)

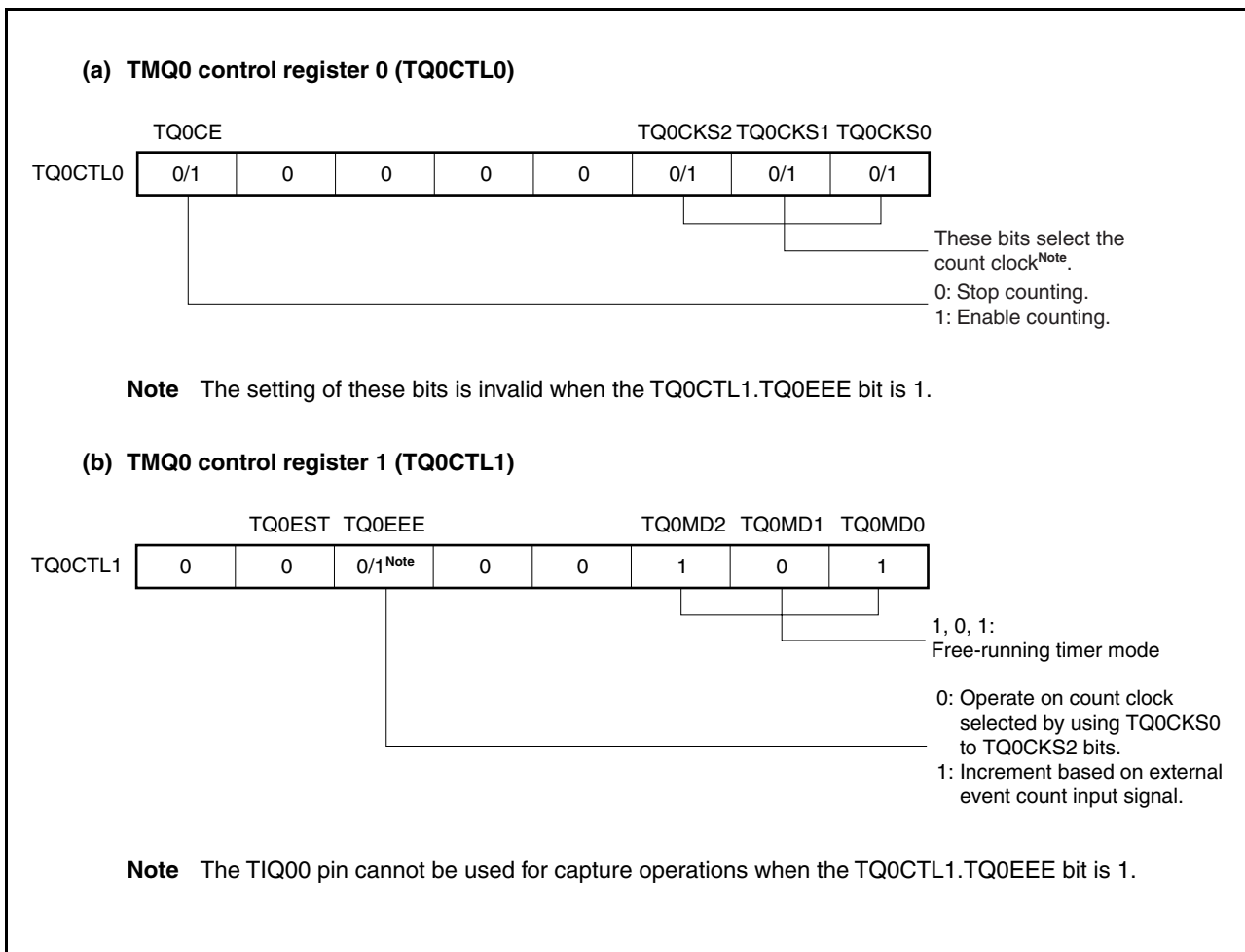


Figure 7-55. Register Settings in Free-Running Timer Mode (2/3)

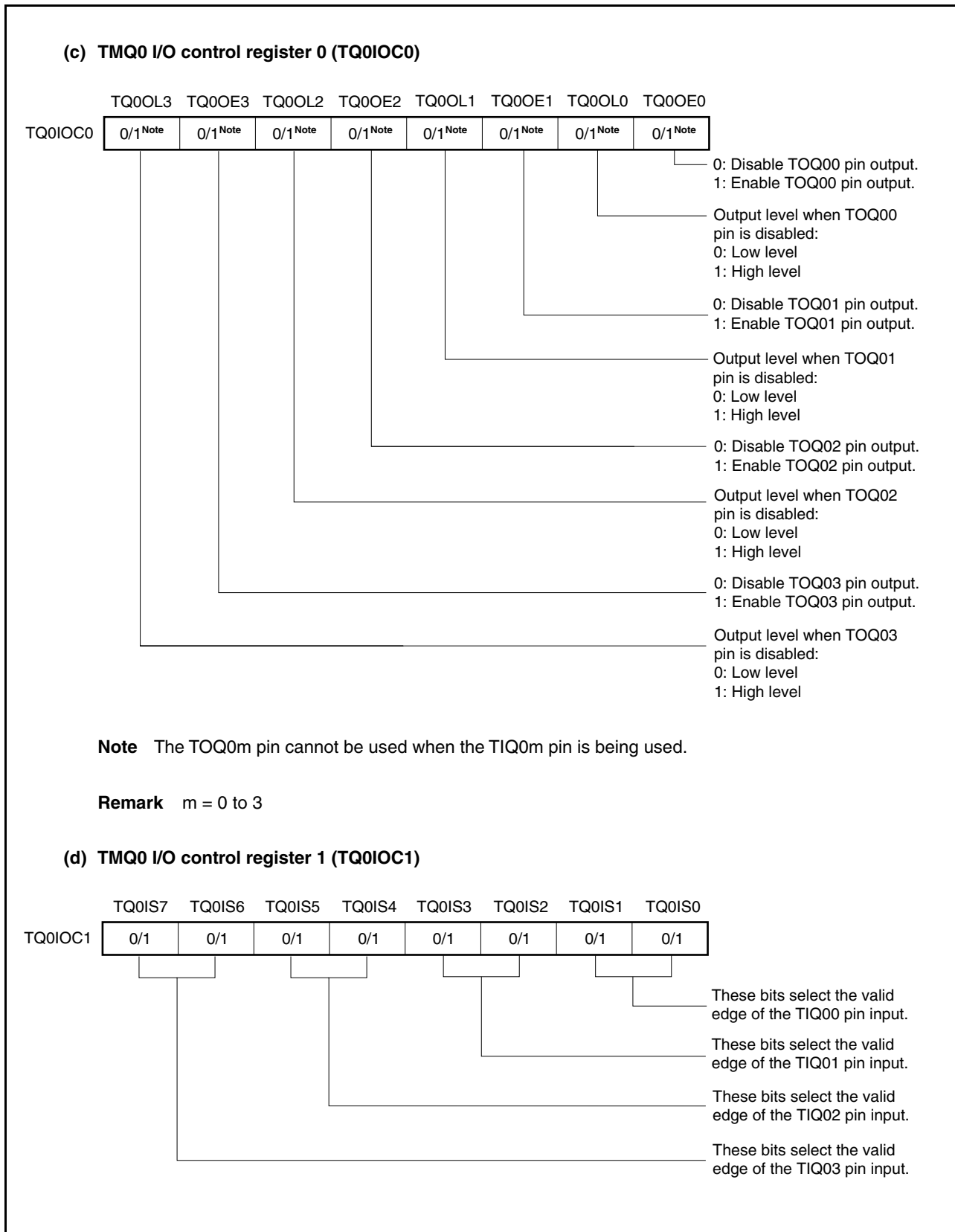
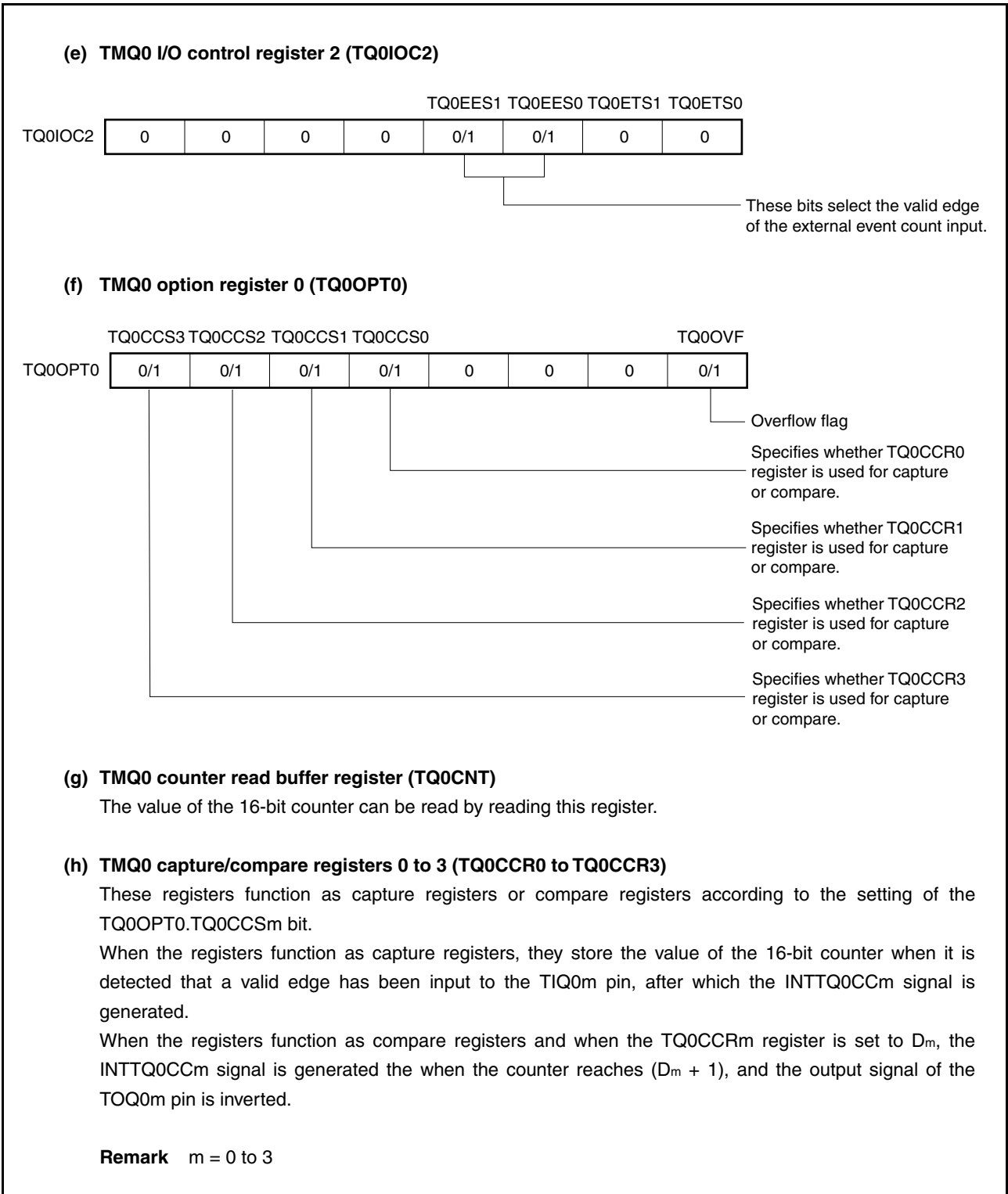


Figure 7-55. Register Settings in Free-Running Timer Mode (3/3)



(1) Operations in free-running timer mode

The following two operations occur in the free-running timer mode:

- Capture operations
- Compare operations

(a) Using a capture/compare register as a compare register

Figure 7-56. Timing and Processing of Operations in Free-Running Timer Mode (Compare Function) (1/2)

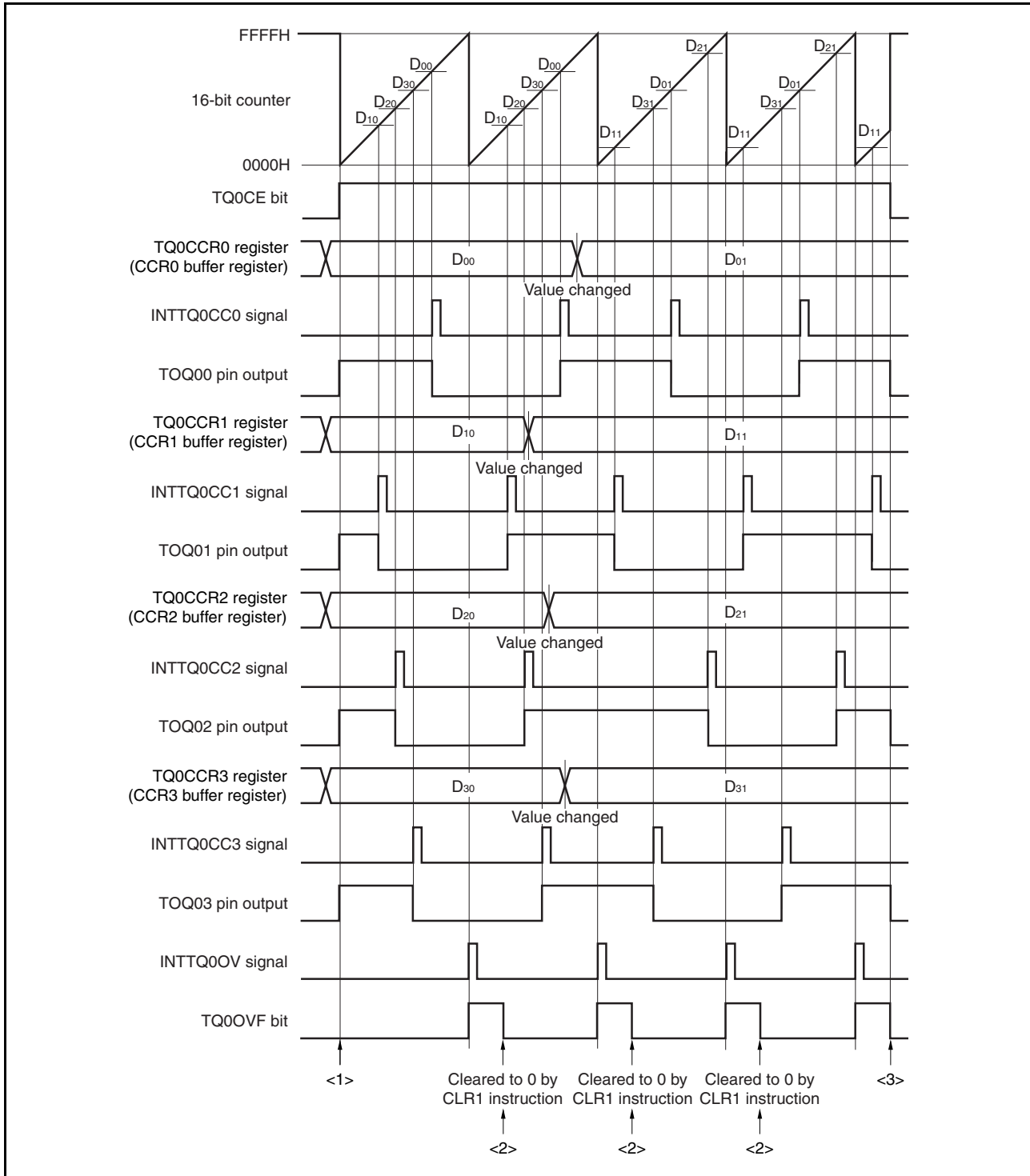
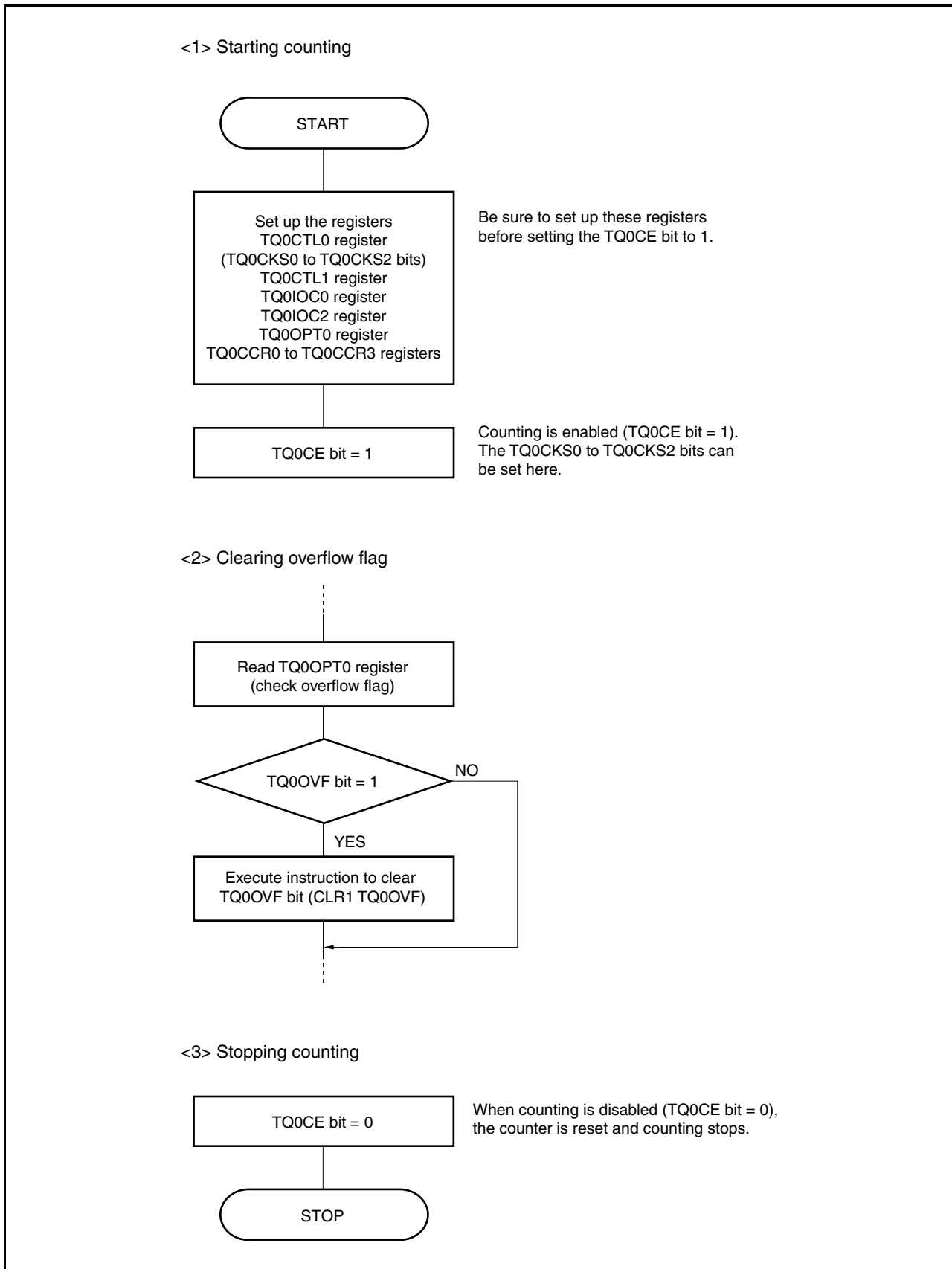


Figure 7-56. Timing and Processing of Operations in Free-Running Timer Mode (Compare Function) (2/2)



(b) Using a capture/compare register as a capture register

Figure 7-57. Timing and Processing of Operations in Free-Running Timer Mode (Capture Function) (1/2)

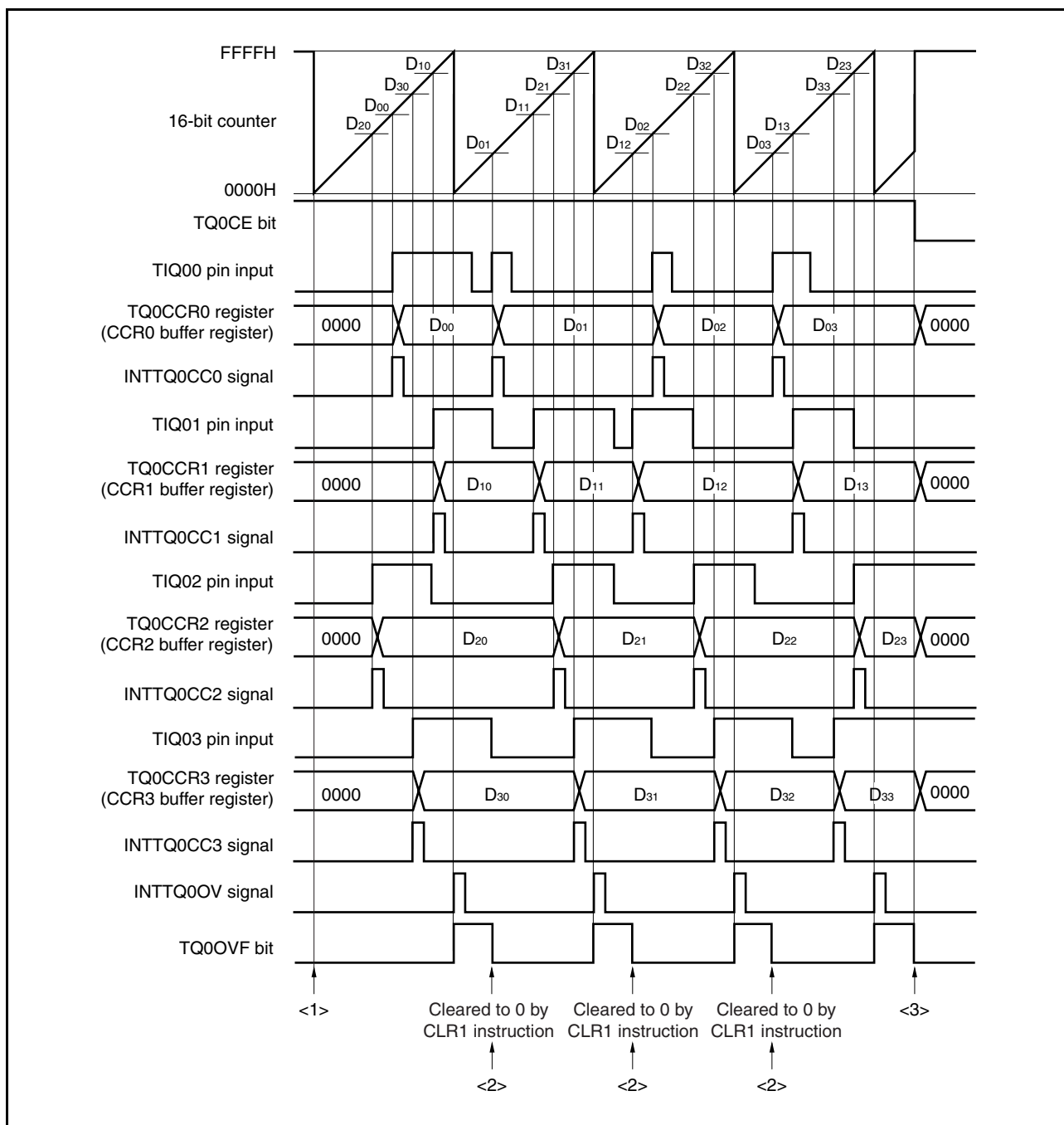
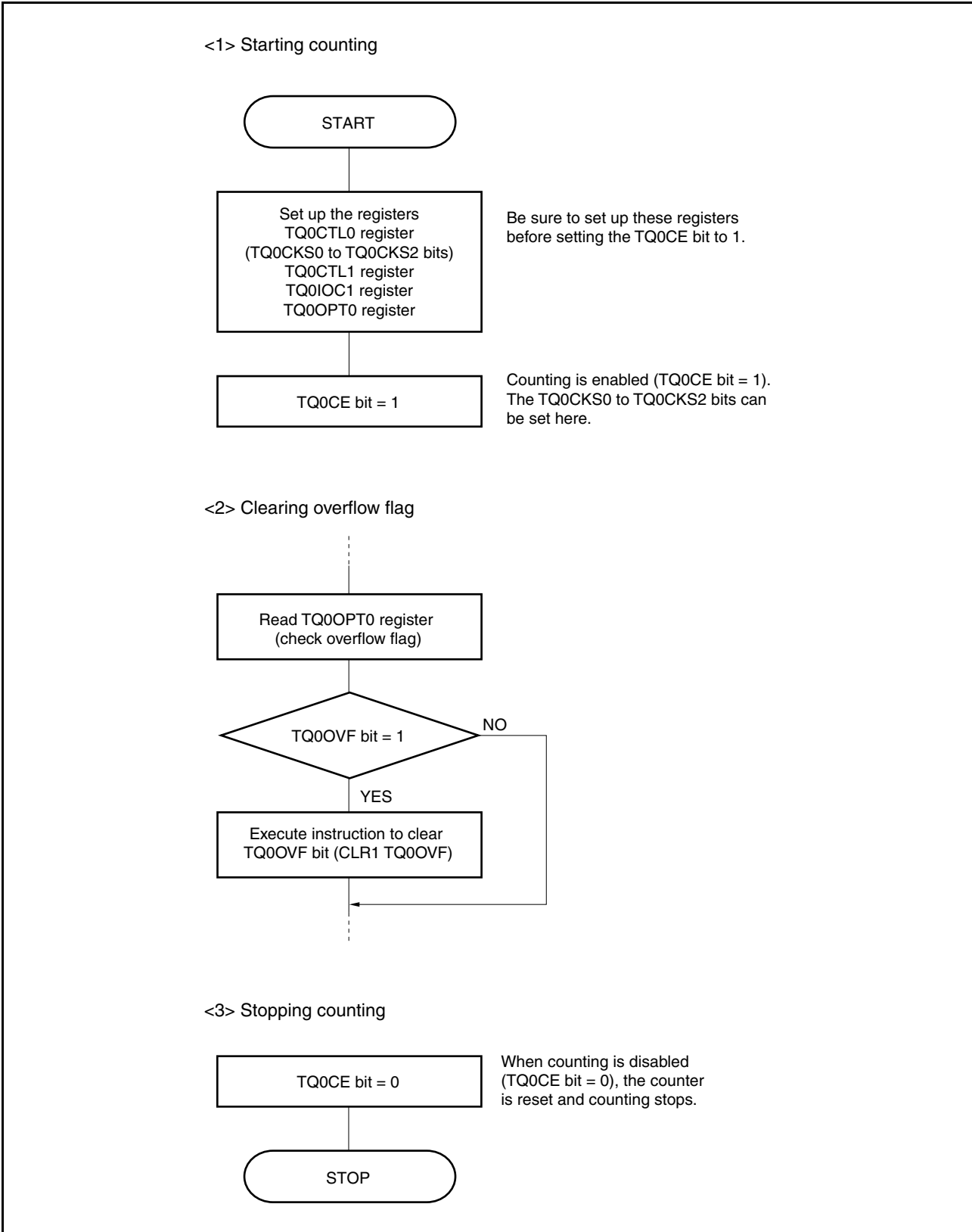


Figure 7-57. Timing and Processing of Operations in Free-Running Timer Mode (Capture Function) (2/2)

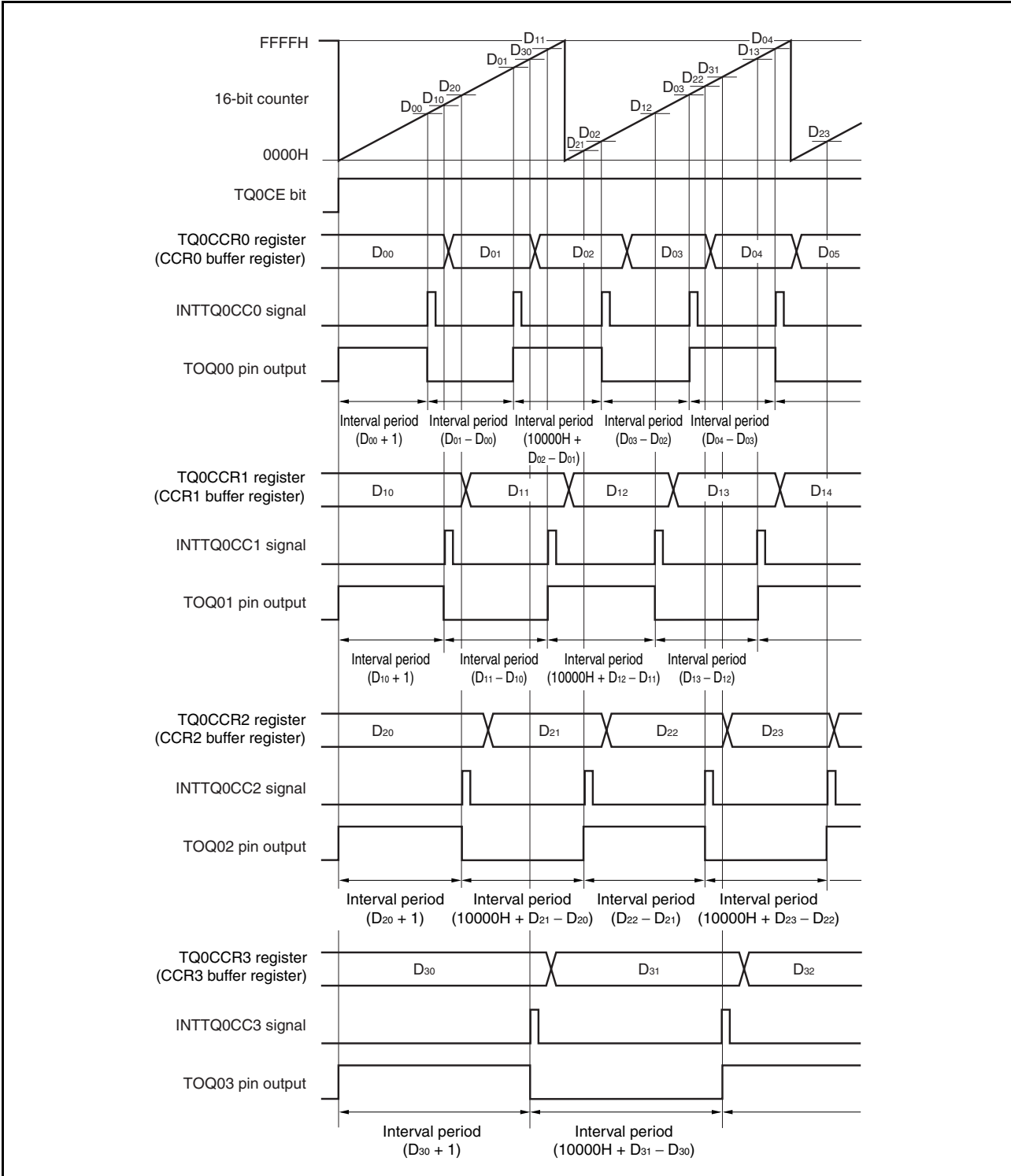


(2) Using free-running timer mode

(a) Interval operation using the TQ0CCRm register as a compare register

When TMQ0 is used as an interval timer with the TQ0CCRm register used as a compare register, the comparison value at which the next interrupt request signal is generated each time the INTTQ0CCm signal has been detected must be set by software.

Figure 7-58. Interval Operation of TMQ0 in Free-Running Timer Mode



When performing an interval operation in the free-running timer mode, four intervals can be set for one channel. To perform the interval operation, the value of the corresponding TQ0CCR_m register must be set again in the interrupt servicing that is executed when the INTTQ0CC_m signal is detected.

The value to be set in this case can be calculated by the following expression, where “D_m” is the interval period.

Compare register default value: $D_m - 1$

Value set to compare register second and subsequent time: Previous set value + D_m

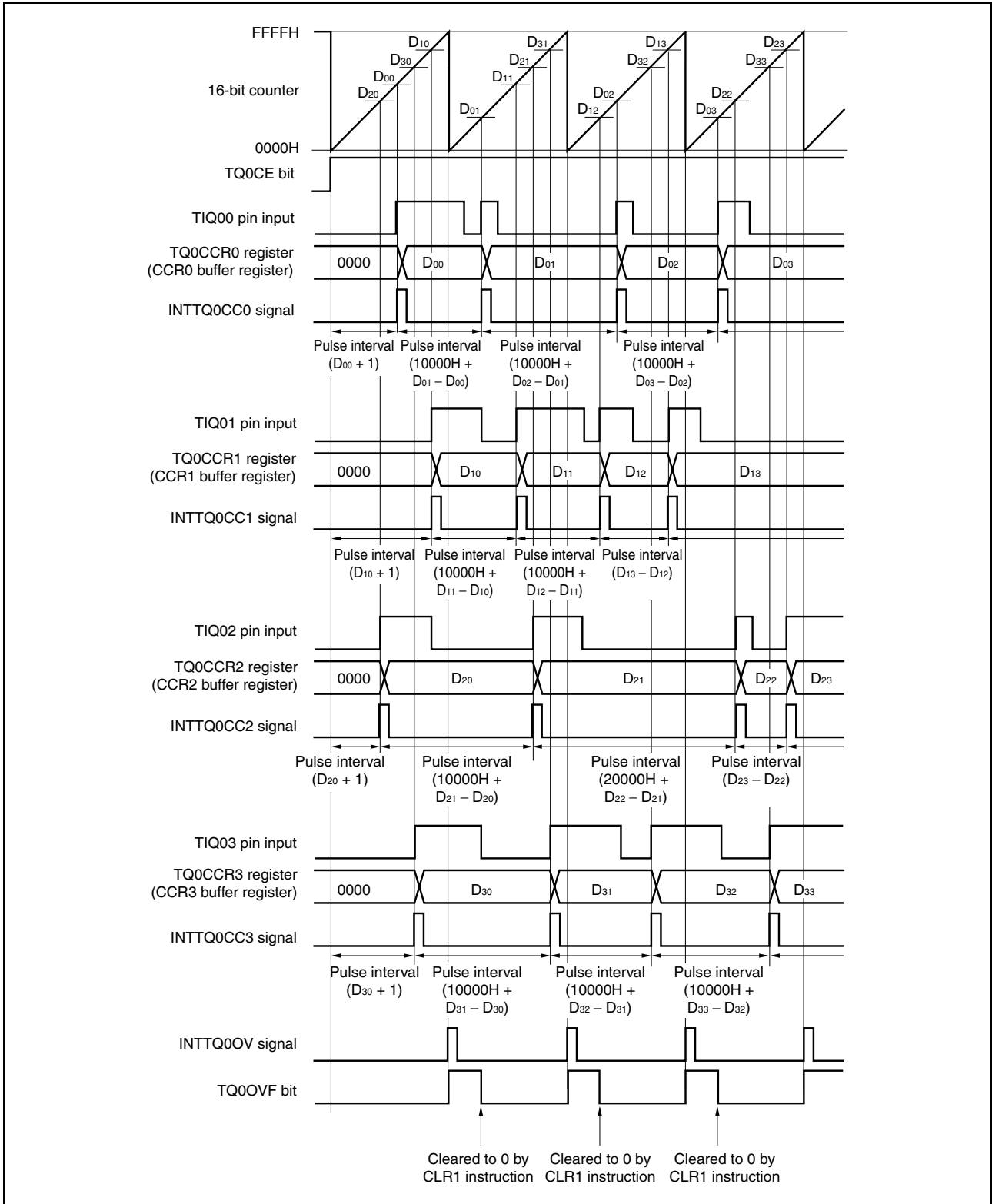
(If the calculation result is greater than FFFFH, subtract 10000H from the result and set the register to this value.)

Remark $m = 0$ to 3

(b) Pulse width measurement using the TQ0CCRm register as a capture register

When pulse width measurement is performed with the TQ0CCRm register used as a capture register, each time the INTTQ0CCm signal has been detected, the capture register must be read and the interval must be calculated by software.

Figure 7-59. Pulse Width Measurement by TMQ0 in Free-Running Timer Mode



Four pulse widths can be measured in the free-running timer mode.

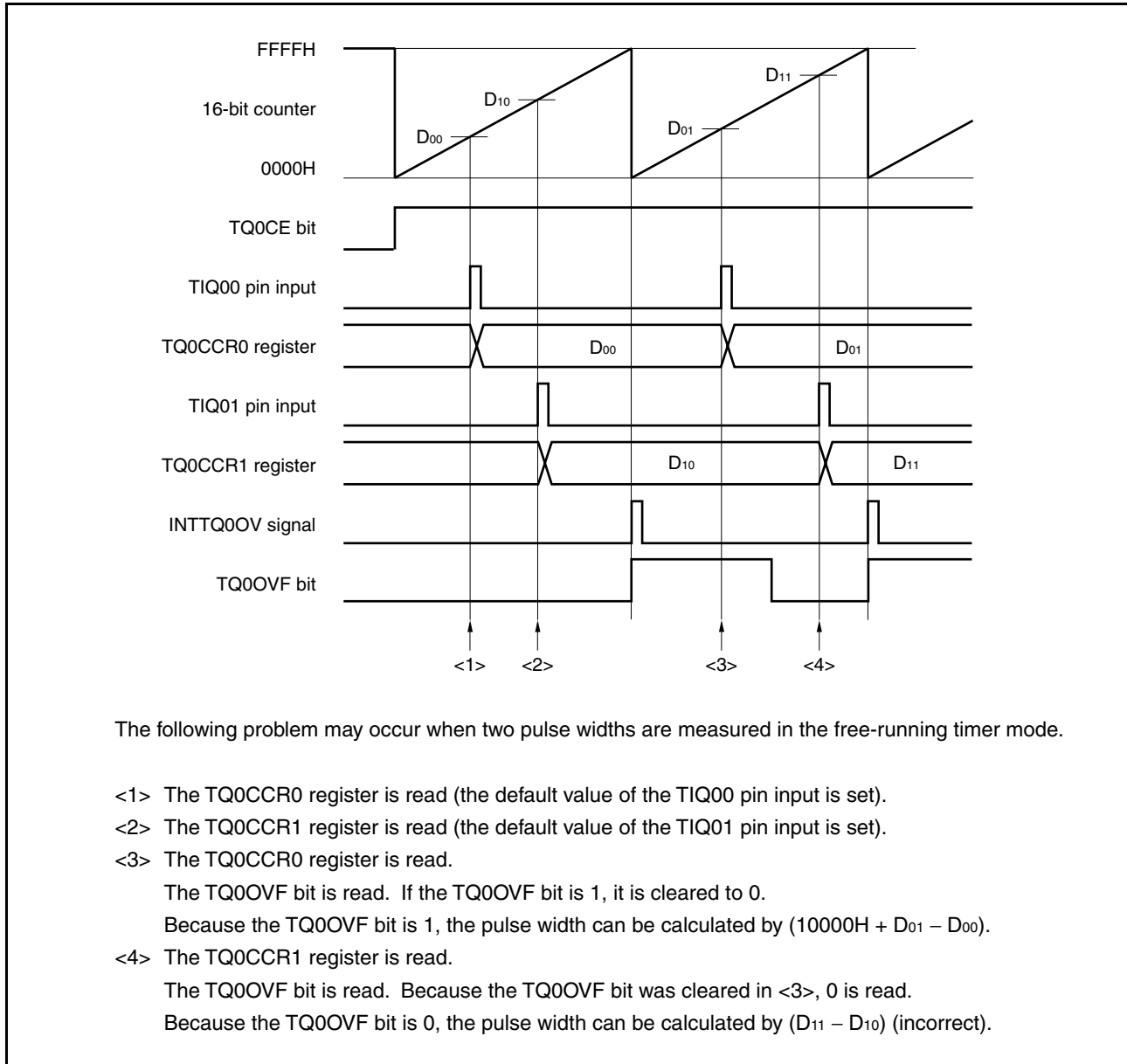
When measuring a pulse width, the pulse width can be calculated by reading the value of the TQ0CCRm register in synchronization with the INTTQ0CCm signal, and calculating the difference between that value and the previously read value.

Remark m = 0 to 3

(c) Processing an overflow when two or more capture registers are used

Care must be exercised in processing the overflow flag when two or more capture registers are used. First, an example of incorrect processing is shown below.

Figure 7-60. Example of Incorrect Processing When Two or More Capture Registers Are Used



When two or more capture registers are used, and if the TQ0OVF bit is cleared to 0 by one capture register, another capture register may not obtain the correct pulse width.

This problem can be resolved by using software, as shown in the example below.

Figure 7-61. Example of Resolving Problem When Two or More Capture Registers Are Used by Using Overflow Interrupt

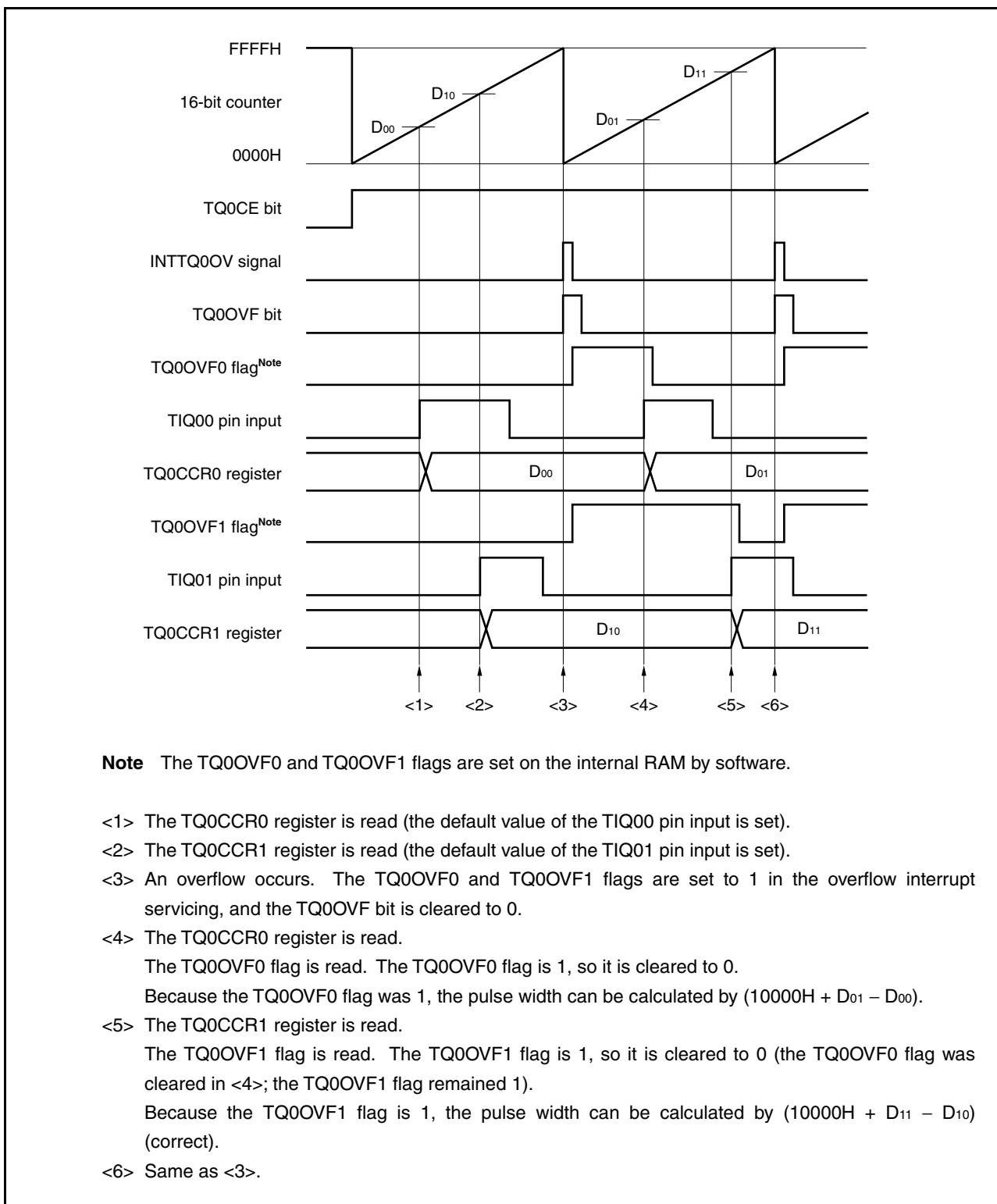
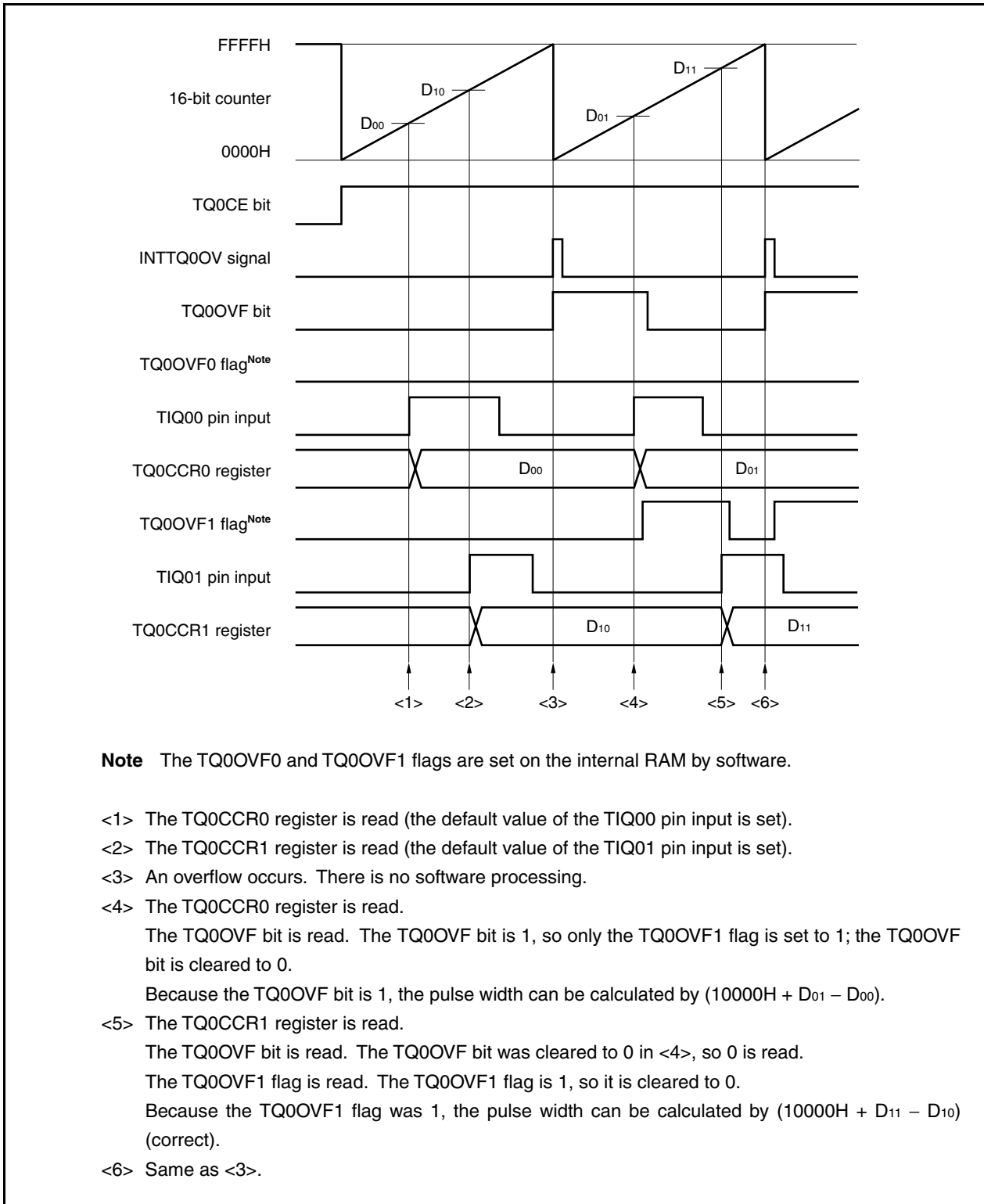


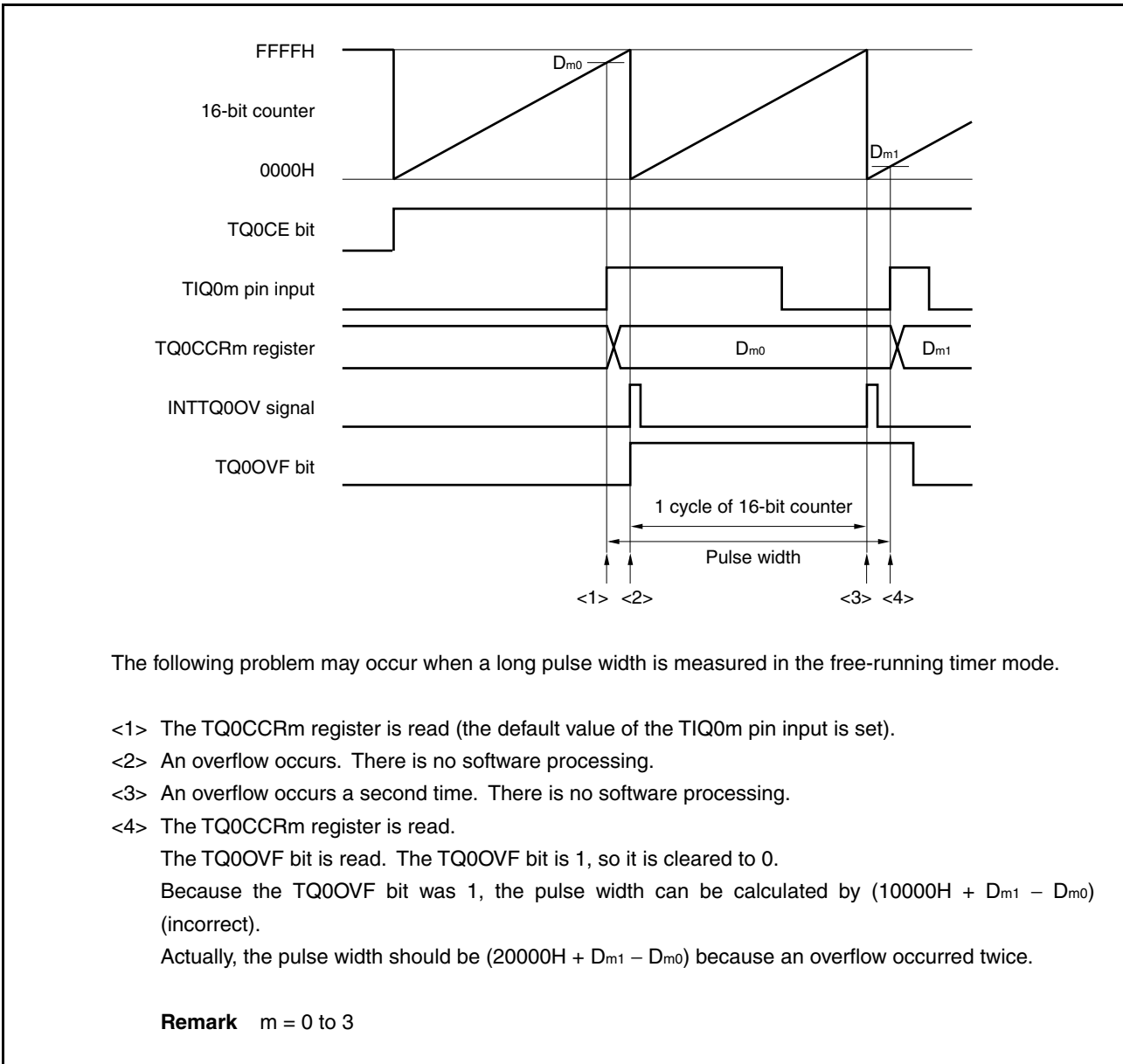
Figure 7-62. Example of Resolving Problem When Two or More Capture Registers Are Used Without Using Overflow Interrupt



(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 between the first capture trigger and the next. First, an example of incorrect processing is shown below.

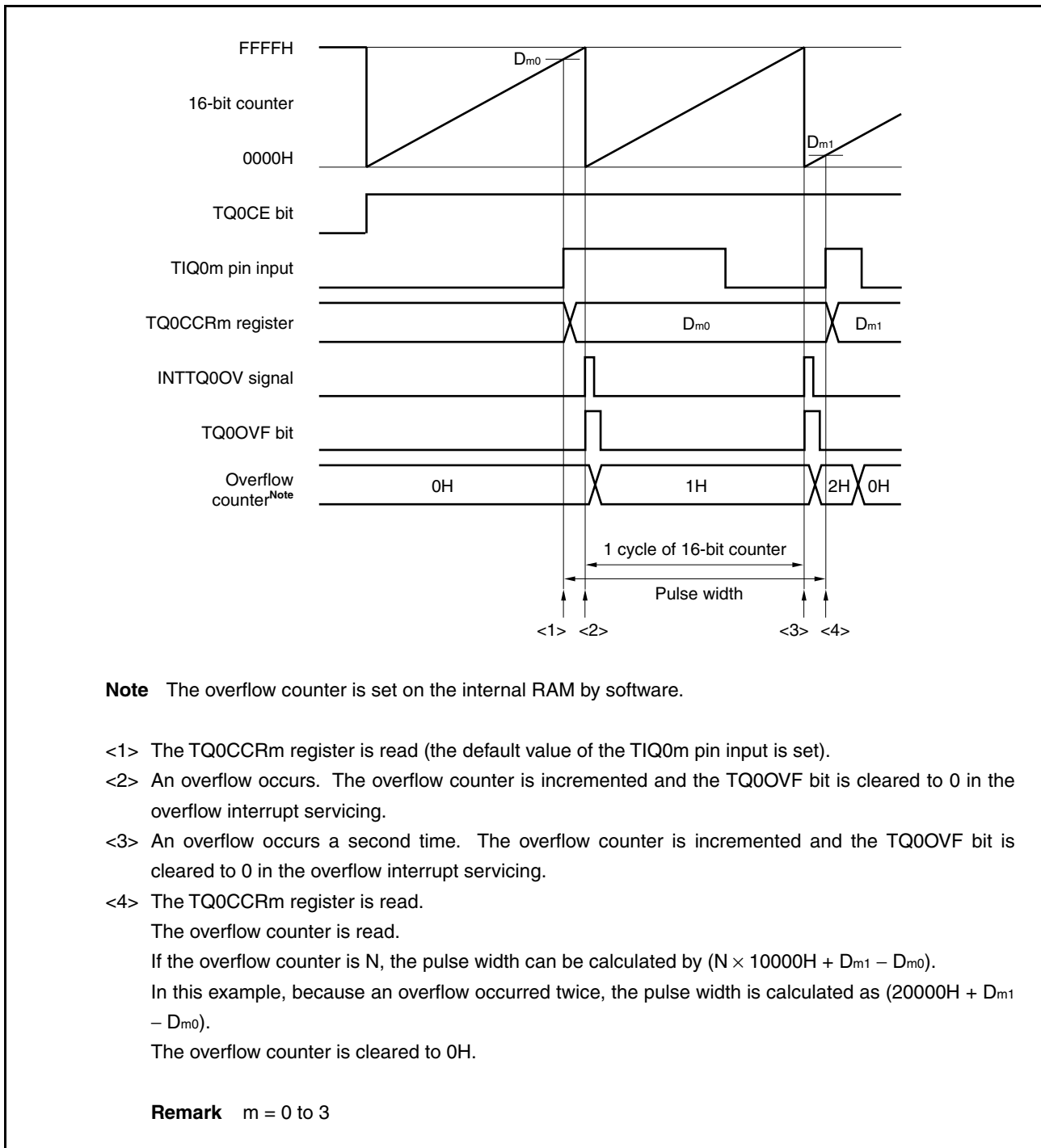
Figure 7-63. Example of Incorrect Processing When Capture Trigger Interval Is Long (When Using TIQ0m)



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 to resolve the problem. An example of how to use software to resolve the problem is shown below.

Figure 7-64. Example of Using Software Processing to Resolve Problem When Capture Trigger Interval Is Long (When Using TIQ0m)



(e) Clearing the overflow flag (TQ0OVF)

The overflow flag (TQ0OVF) can be cleared to 0 by reading the TQ0OVF bit and, if its value is 1, either clearing the bit to 0 by using the CLR1 instruction or by writing 8-bit data (with bit 0 as "0") to the TQ0OPT0 register.

7.4.7 Pulse width measurement mode (TQ0MD2 to TQ0MD0 bits = 110)

In the pulse width measurement mode, TMQ0 starts incrementing when the TQ0CTL0.TQ0CE bit is set to 1. Each time it is detected that a valid edge has been input to the TIQ0m pin, the value of the 16-bit counter is stored in the TQ0CCRm register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TQ0CCRm register after a capture interrupt request signal (INTTQ0CCm) occurs.

Select one of the TIQ00 to TIQ03 pins as the capture trigger input pin. Specify “No edge detected” by using the TQ0IOC1 register for the unused pins.

Remarks 1. For how to set the TIQ0m pin, see **Table 7-2 Pins Used by TMQ0** and **Table 4-17 Settings When Pins Are Used for Alternate Functions**.

2. For how to enable the INTTQ0CCm interrupt signal, see **CHAPTER 19 INTERRUPT SERVICING/ EXCEPTION PROCESSING FUNCTION**.

3. m = 0 to 3

k = 1 to 3

Figure 7-65. Configuration of TMQ0 in Pulse Width Measurement Mode

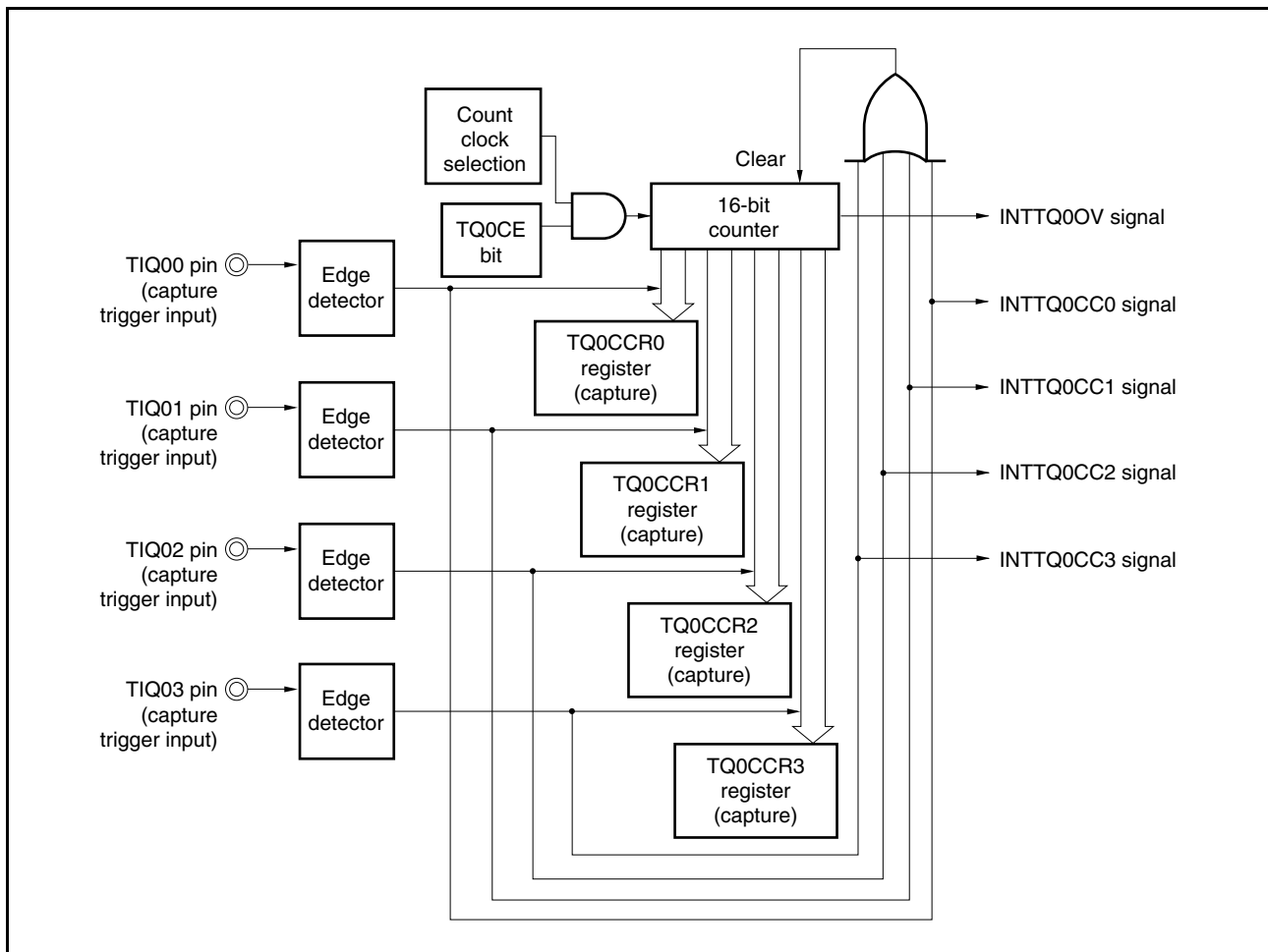
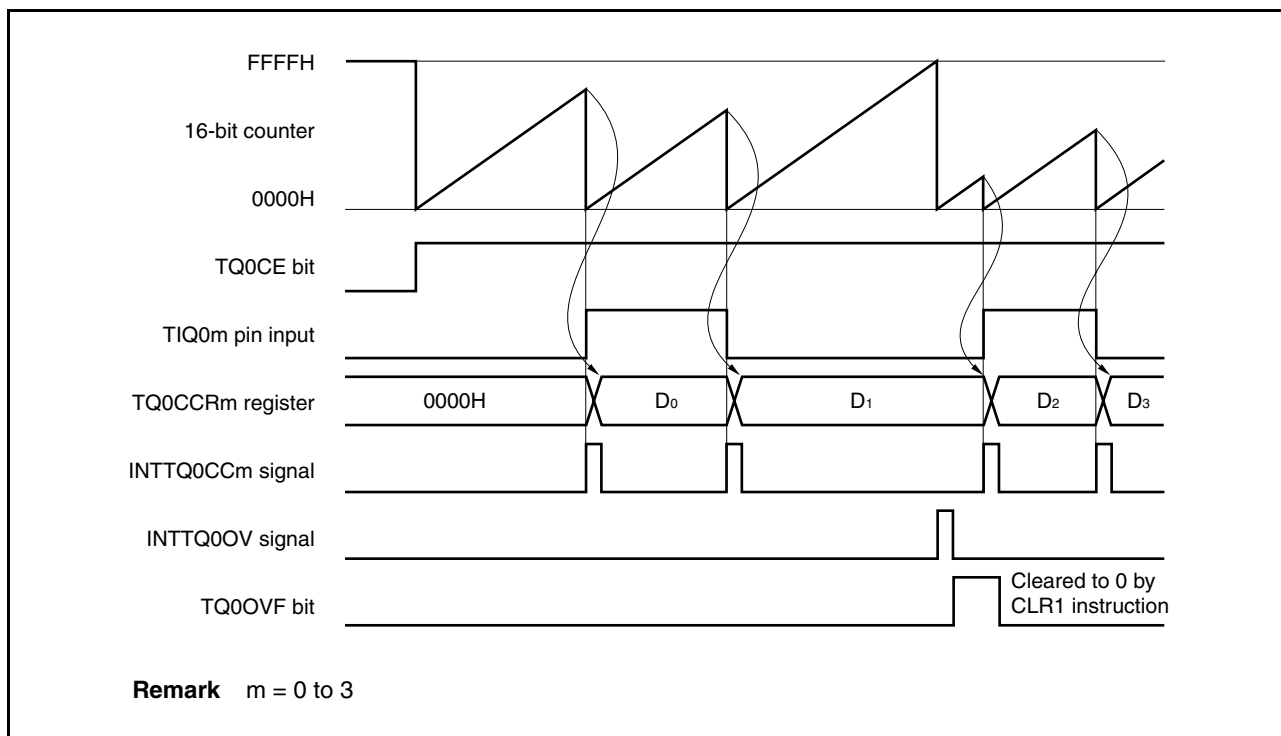


Figure 7-66. Basic Timing of Operations in Pulse Width Measurement Mode



When the TQ0CE bit is set to 1, the 16-bit counter starts incrementing. When it is subsequently detected that a valid edge has been input to the TIQ0m pin, the value of the 16-bit counter is stored in the TQ0CCRm register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTTQ0CCm) is generated.

The pulse width is calculated as follows.

$$\text{Pulse width} = \text{Captured value} \times \text{Count clock cycle}$$

If a valid edge has not been input to the TIQ0m pin by the time the 16-bit counter has incremented up to FFFFH, an overflow interrupt request signal (INTTQ0OV) is generated at the next count clock, and the counter is cleared to 0000H and continues incrementing. At this time, the overflow flag (TQ0OPT0.TQ0OVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR1 software instruction.

If the overflow flag is set to 1, the pulse width can be calculated as follows.

$$\text{Pulse width} = (10000\text{H} \times \text{Number of times the TQ0OVF bit was set (1)} + \text{Captured value}) \times \text{Count clock cycle}$$

Remark m = 0 to 3

Figure 7-67. Register Settings in Pulse Width Measurement Mode (1/2)

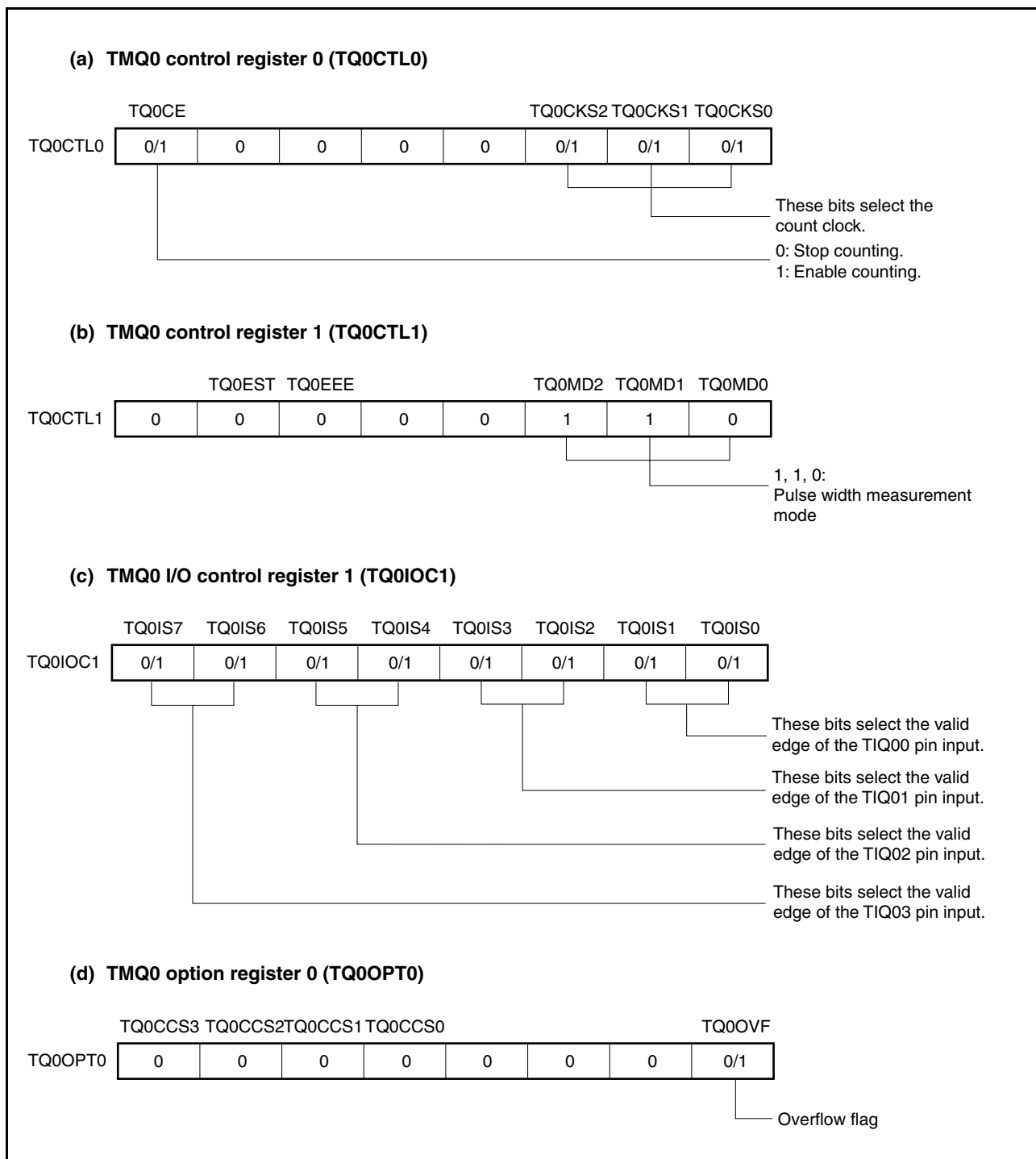


Figure 7-67. Register Settings in Pulse Width Measurement Mode (2/2)**(e) TMQ0 counter read buffer register (TQ0CNT)**

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

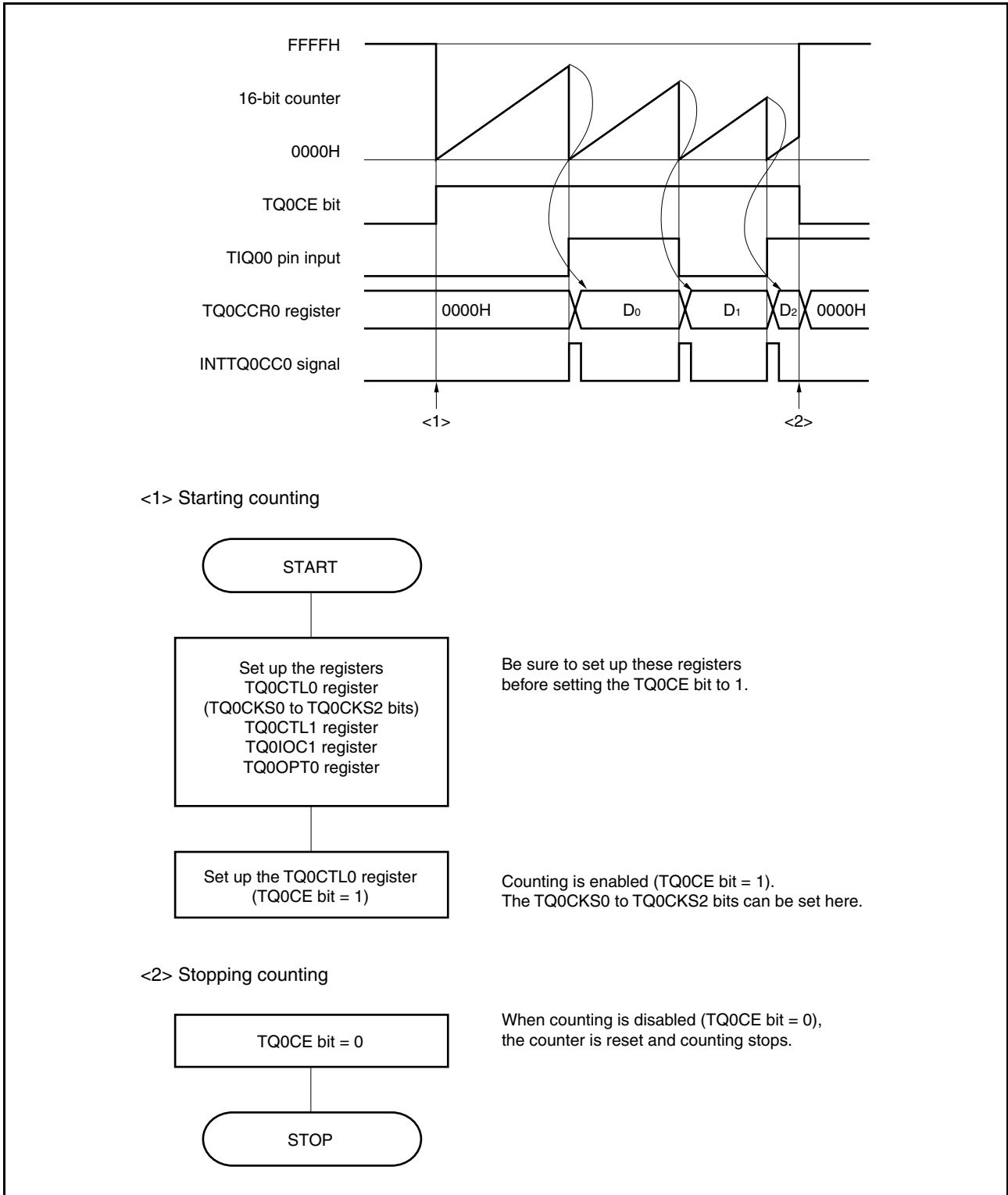
(f) TMQ0 capture/compare registers 0 to 3 (TQ0CCR0 to TQ0CCR3)

These registers store the 16-bit counter value upon detection of the input of a valid edge to the TIQ0m pin.

Remark TMQ0 I/O control register 0 (TQ0IOC0) and TMQ0 I/O control register 2 (TQ0IOC2) are not used in the pulse width measurement mode.

(1) Operations in pulse width measurement mode

Figure 7-68. Timing and Processing of Operations in Pulse Width Measurement Mode



(2) Using pulse width measurement mode**(a) Clearing the overflow flag (TQ0OVF)**

The overflow flag (TQ0OVF) can be cleared to 0 by reading the TQ0OVF bit and, if its value is 1, either clearing the bit to 0 by using the CLR1 instruction or by writing 8-bit data (with bit 0 as "0") to the TQ0OPT0 register.

7.4.8 Timer output operations

The following table shows the operations and output levels of the TOQ00 to TOQ03 pins.

Table 7-8. Timer Output Control in Each Mode

Operation Mode	TOQ00 Pin	TOQ01 Pin	TOQ02 Pin	TOQ03 Pin
Interval timer mode	Square wave output			
External event count mode	-			
External trigger pulse output mode	Square wave output	External trigger pulse output	External trigger pulse output	External trigger pulse output
One-shot pulse output mode		One-shot pulse output	One-shot pulse output	One-shot pulse output
PWM output mode		PWM output	PWM output	PWM output
Free-running timer mode	Square wave output (only when compare function is used)			
Pulse width measurement mode	-			

Table 7-9. Truth Table of TOQ00 to TOQ03 Pins Under Control of Timer Output Control Bits

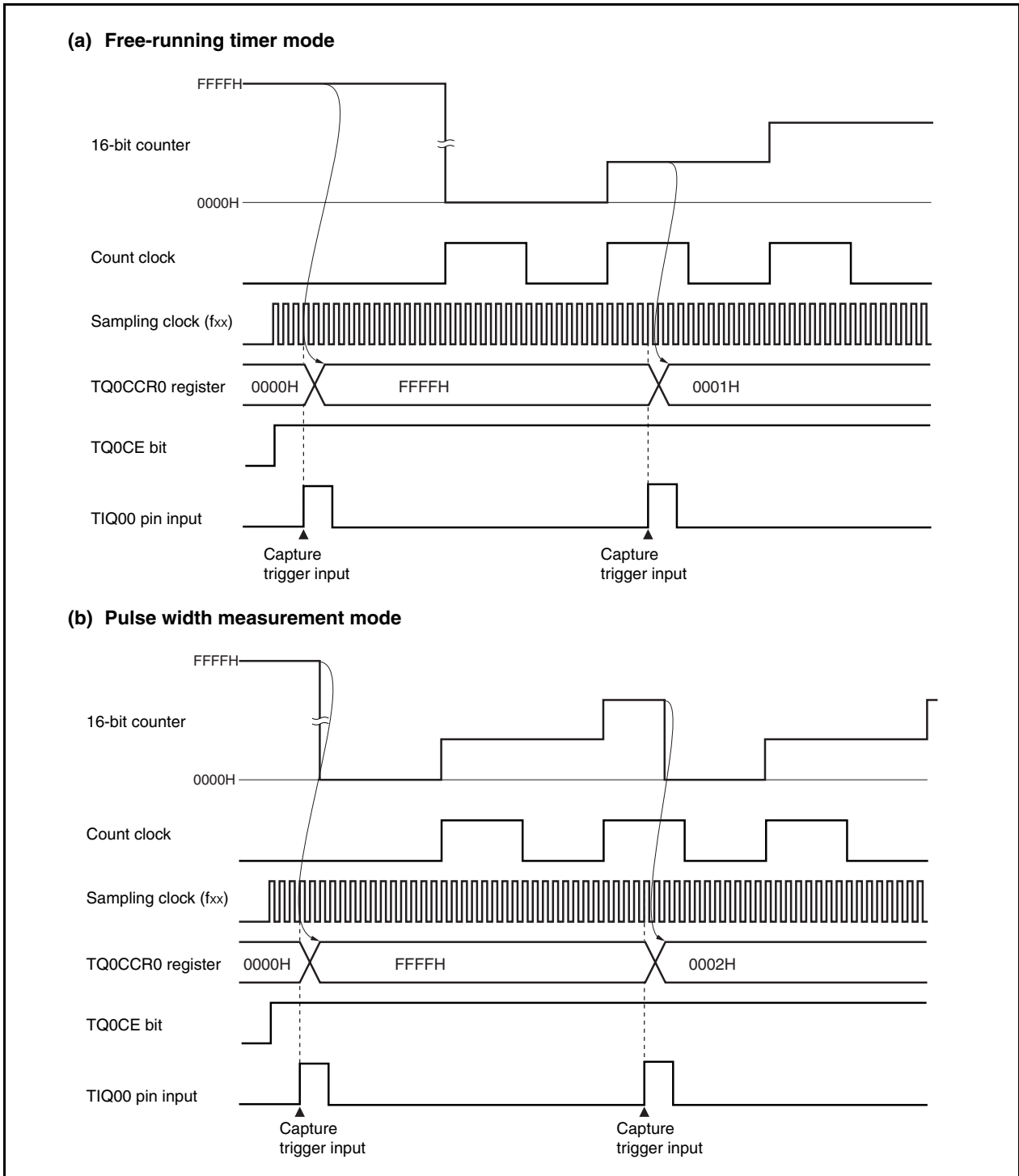
TQ0IOC0.TQ0OLm Bit	TQ0IOC0.TQ0OEm Bit	TQ0CTL0.TQ0CE Bit	Level of TOQ0m Pin
0	0	×	Low-level output
	1	0	Low-level output
		1	Low level immediately before counting, high level after counting is started
1	0	×	High-level output
	1	0	High-level output
		1	High level immediately before counting, low level after counting is started

Remark m = 0 to 3

7.5 Cautions

(1) Capture operation

When the capture operation is used and $f_{xx}/8$, $f_{xx}/16$, $f_{xx}/32$, $f_{xx}/64$, $f_{xx}/128$, or the external event counter (TQ0CLT1.TQ0EEE bit = 1) is selected as the count clock, FFFFH, not 0000H, may be captured in the TQ0CCR0, TQ0CCR1, TQ0CCR2, and TQ0CCR3 registers, or the capture operation may not be performed at all (the capture interrupt does not occur) if the capture trigger is input immediately after the TQ0CE bit is set to 1.



CHAPTER 8 16-BIT INTERVAL TIMER M (TMM)

Timer M (TMM) is a 16-bit interval timer.

The V850ES/JC3-L, V850ES/JE3-L incorporate one TMM timer, TMM0.

8.1 Features

TMM0 is a dedicated interval timer that generates interrupt requests at a specified interval based on the count clock selected from one of eight clock sources: the main clock (f_{xx}), a divided main clock ($f_{xx}/2$, $f_{xx}/4$, $f_{xx}/64$, $f_{xx}/512$), the watch timer interrupt signal (INTWT), the internal clock ($f_{\pi}/8$), and the subclock (f_{τ}).

TMM0 can only be used in the clear & start mode; it cannot be used in the free-running timer mode.

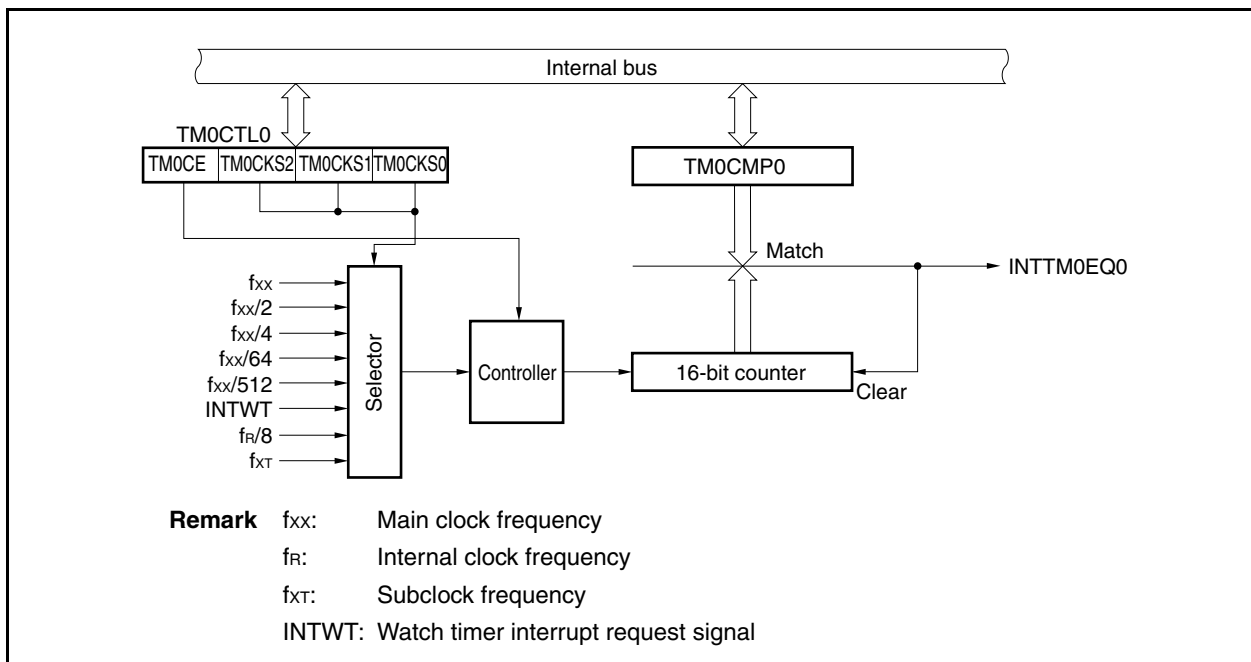
8.2 Configuration

TMM0 includes the following hardware.

Table 8-1. Configuration of TMM0

Item	Configuration
Register	16-bit counter TMM0 compare register 0 (TM0CMP0) TMM0 control register 0 (TM0CTL0)

Figure 8-1. Block Diagram of TMM0



(1) 16-bit counter

This counter counts the internal clock.
This counter cannot be read or written.

(2) TMM0 compare register 0 (TM0CMP0)

This is a 16-bit compare register.

(3) TMM0 control register 0 (TM0CTL0)

This is an 8-bit register used to control the operation of TMM0.

(4) Selector

The selector is used to select the count clock of the 16-bit counter. The count clock can be selected from eight clock sources.

8.3 Registers

(1) TMM0 control register (TM0CTL0)

The TM0CTL0 register is an 8-bit register that controls the operation of TMM0.

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

Reset sets this register to 00H.

Except for the TMOCE bit, the bits of the TM0CTL0 register cannot be rewritten to different values while TMM0 is operating (bits can be rewritten only to the same value as was previously specified).

After reset: 00H R/W Address: FFFF690H

	<7>	6	5	4	3	2	1	0
TM0CTL0	TMOCE	0	0	0	0	TM0CKS2	TM0CKS1	TM0CKS0

TMOCE	Internal clock operation enable/disable specification
0	TMM0 operation disabled (16-bit counter reset asynchronously).
1	TMM0 operation enabled.
When the TMOCE bit is cleared to 0, the internal clock of TMM0 is disabled (fixed to low level) and 16-bit counter is reset asynchronously.	

TM0CKS2	TM0CKS1	TM0CKS0	Count clock selection
0	0	0	f_{xx}
0	0	1	$f_{xx}/2$
0	1	0	$f_{xx}/4$
0	1	1	$f_{xx}/64$
1	0	0	$f_{xx}/512$
1	0	1	INTWT
1	1	0	$f_R/8$
1	1	1	f_{XT}

Cautions

1. Set the TM0CKS2 to TM0CKS0 bits while TMM0 is stopped (TMOCE bit = 0). The TM0CKS2 to TM0CKS0 bits cannot be set at the same time as changing the value of TMOCE from 0 to 1.

2. Be sure to clear bits 3 to 6 to "0".

Remark

f_{xx} : Main clock frequency
 f_R : Internal oscillator clock frequency
 f_{XT} : Subclock frequency
 INTWT: Watch timer interrupt request signal

(2) TMM0 compare register 0 (TM0CMP0)

The TM0CMP0 register is a 16-bit compare register.

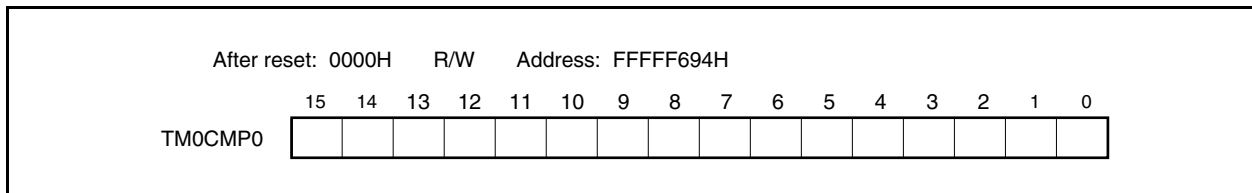
This register can be read or written in 16-bit units.

Reset sets this register to 0000H. However, if TMM0 is reset while it is stopped, this register is set to FFFFH.

The same value can always be written to the TM0CMP0 register by software.

The TM0CMP0 register cannot be rewritten while TMM0 is operating (TMOCTL0.TM0CE bit = 1).

Caution Do not set the TM0CMP0 register to FFFFH.



8.4 Operation

8.4.1 Interval timer mode

When the TMOCE 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 incrementing.

When the value of the 16-bit counter matches the value of the TMOCMP0 register, the 16-bit counter is cleared to 0000H and a compare match interrupt request signal (INTTM0EQ0) is generated at the specified interval.

Figure 8-2. Configuration of Interval Timer

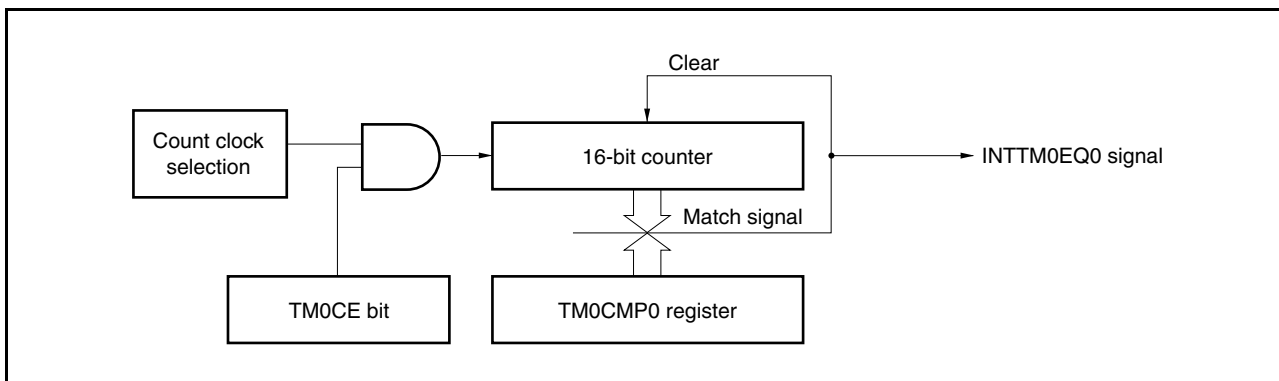
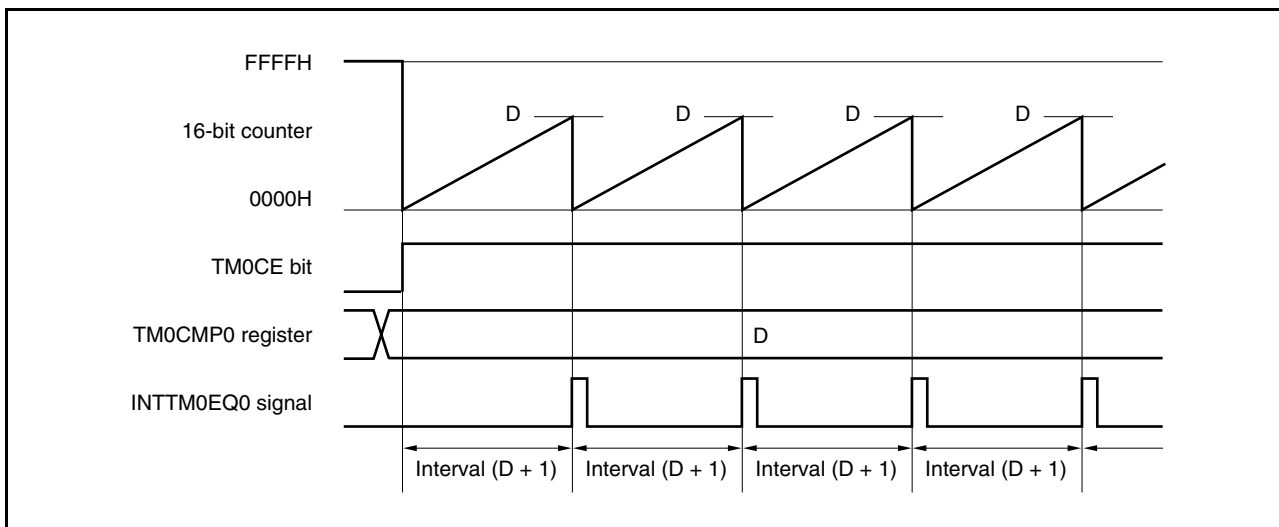
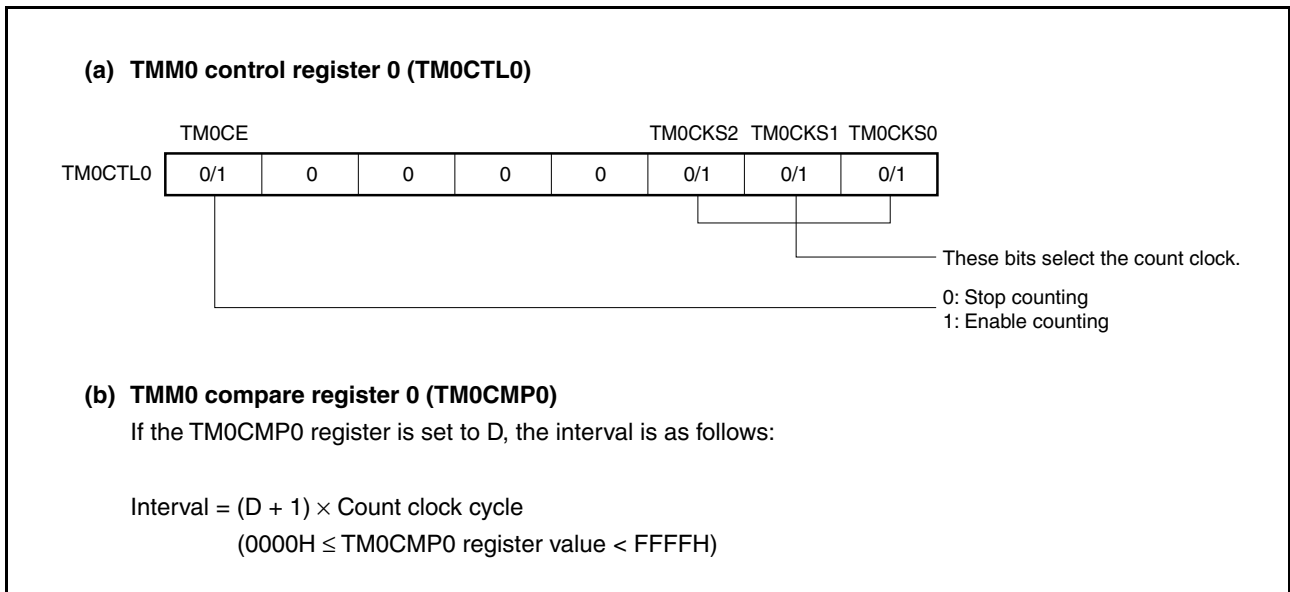


Figure 8-3. Basic Timing of Operation in Interval Timer Mode



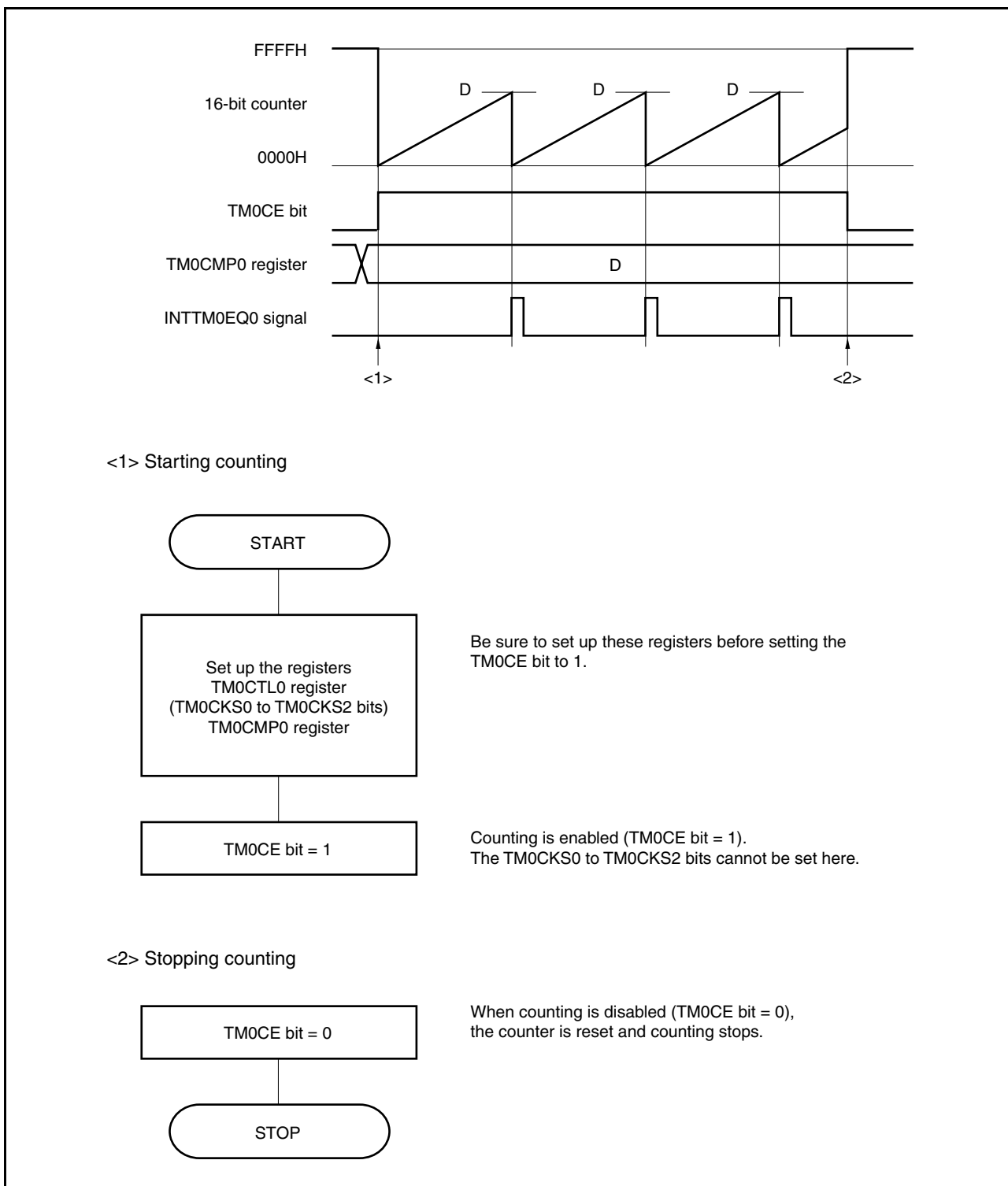
An example of the register settings when the interval timer mode is used is shown in the figure below.

Figure 8-4. Register Settings in Interval Timer Mode



(1) Operations in interval timer mode

Figure 8-5. Timing and Processing of Operations in Interval Timer Mode

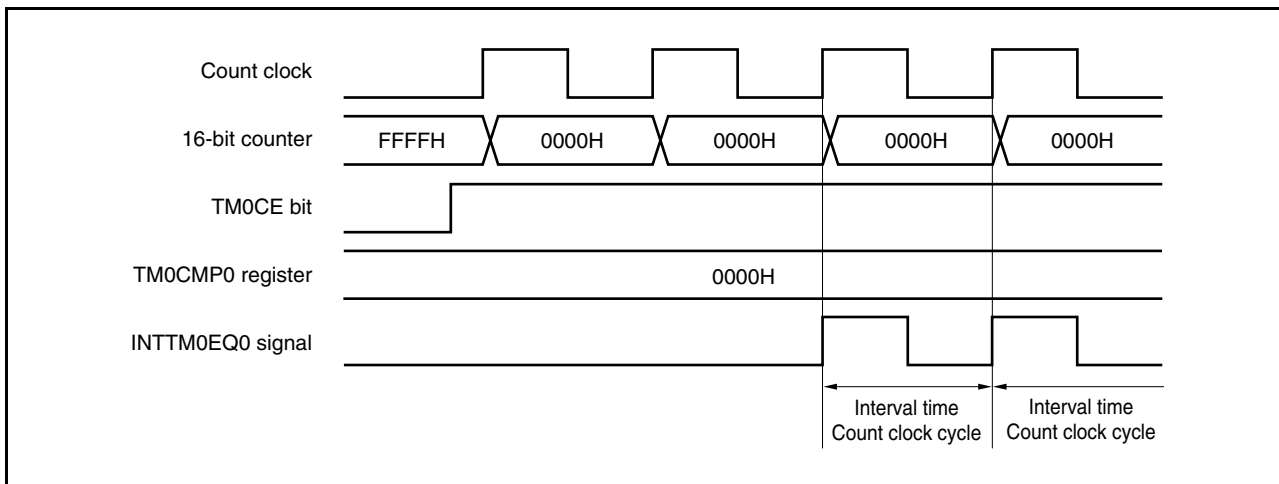


(2) Using interval timer mode

(a) Operation when TM0CMP0 register is set to 0000H

When the TM0CMP0 register is set to 0000H, the INTTM0EQ0 signal is generated for each count clock cycle. The value of the 16-bit counter is always 0000H.

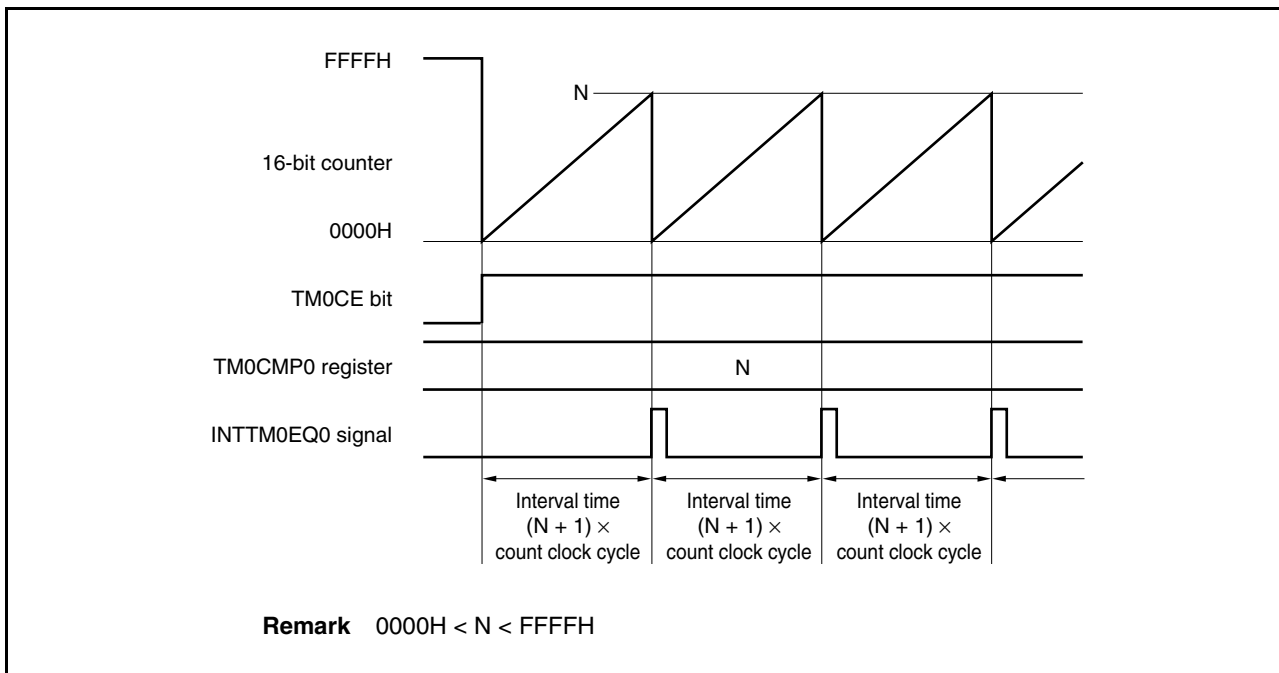
Figure 8-6. Operation of Interval Timer When TM0CMP0 Register Is Set to 0000H



(b) Operation when TM0CMP0 register is set to N

When the TM0CMP0 register is set to N, the 16-bit counter increments up to N and is reset to 0000H in synchronization with the next increment timing. The INTTM0EQ0 signal is then generated.

Figure 8-7. Operation of Interval Timer When TM0CMP0 Register is Set to Other Than 0000H, FFFFH



8.4.2 Cautions

- (1) It takes the 16-bit counter up to the following time to start counting after the TM0CTL0.TM0CE bit is set to 1, depending on the count clock selected.

Selected Count Clock	Maximum Time Before Counting Starts
f_{xx}	$2/f_{xx}$
$f_{xx}/2$	$3/f_{xx}$
$f_{xx}/4$	$6/f_{xx}$
$f_{xx}/64$	$128/f_{xx}$
$f_{xx}/512$	$1024/f_{xx}$
INTWT	Second rising edge of INTWT signal
$f_R/8$	$16/f_R$
f_{XT}	$2/f_{XT}$

- (2) Rewriting the TM0CMP0 and TM0CTL0 registers is prohibited while TMM0 is operating. If these registers are rewritten while the TMM0 is operating (TM0CE bit = 1), the operation cannot be guaranteed. If these registers are rewritten by mistake, clear the TM0CTL0.TM0CE bit to 0, and set the registers again.

CHAPTER 9 WATCH TIMER

9.1 Functions

The watch timer has the following functions.

- Watch timer: An interrupt request signal (INTWT) is generated at intervals of 0.5 or 0.25 seconds by using the main clock or subclock.
- Interval timer: An interrupt request signal (INTWTI) is generated at set intervals.

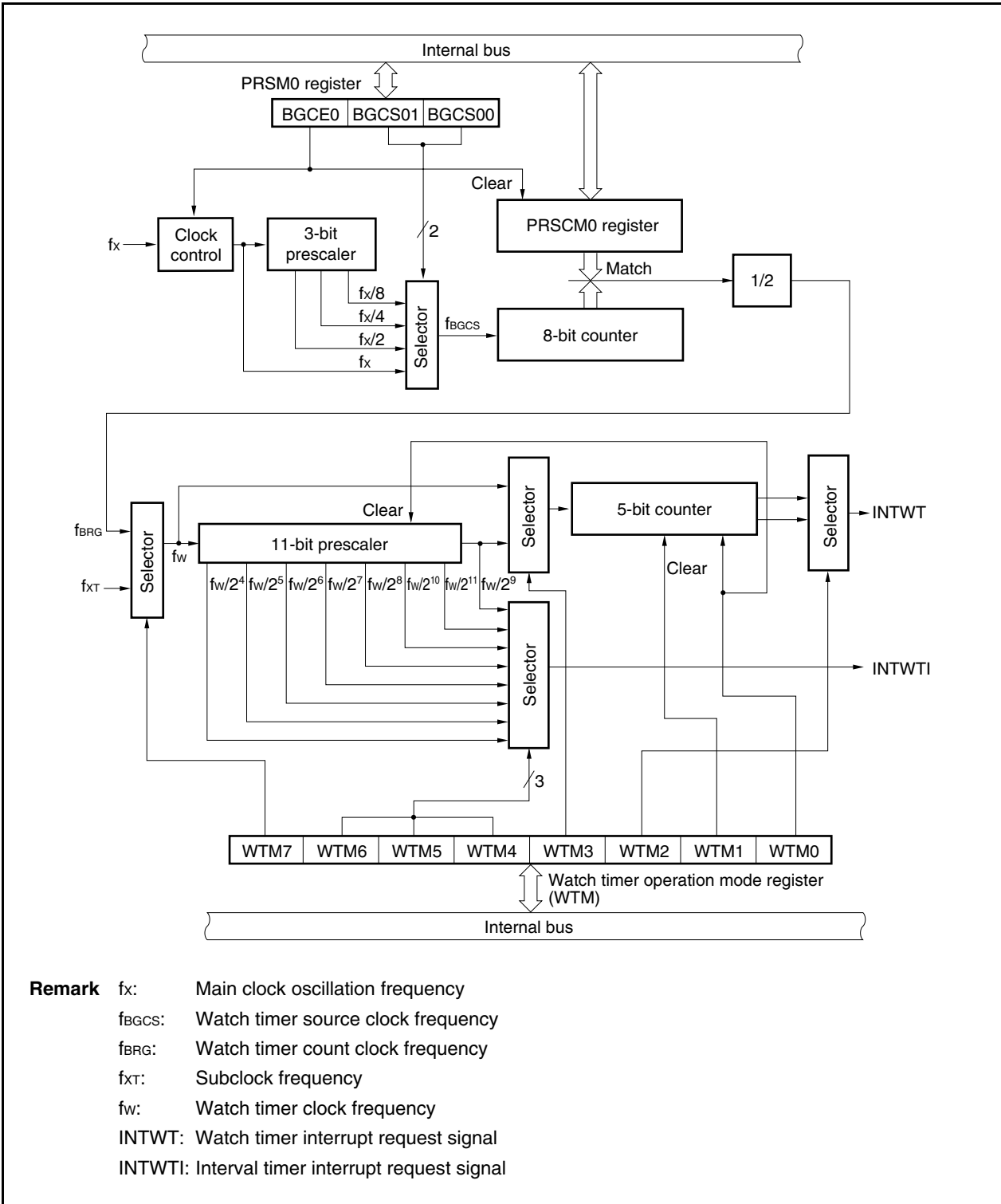
The watch timer and interval timer functions can be used at the same time.

Caution INTWTI interrupt of the watch timer and INTRTC2 interrupt of RTC, and INTWT interrupt of the watch timer and INTRTC0 interrupt of RTC are alternate interrupt signals, and therefore cannot be used simultaneously.

9.2 Configuration

The block diagram of the watch timer is shown below.

Figure 9-1. Block Diagram of Watch Timer



(1) Clock control

This block controls supplying and stopping the operating clock (f_x) when the watch timer operates on the main clock.

(2) 3-bit prescaler

This prescaler divides f_x to generate $f_x/2$, $f_x/4$, or $f_x/8$.

(3) 8-bit counter

This counter counts the source clock (f_{BGCS}).

(4) 11-bit prescaler

This prescaler divides f_w to generate a clock of $f_w/2^4$ to $f_w/2^{11}$.

(5) 5-bit counter

This counter counts f_w or $f_w/2^9$, and generates a watch timer interrupt request signal at intervals of $2^4/f_w$, $2^5/f_w$, $2^{12}/f_w$, or $2^{14}/f_w$.

(6) Selector

The watch timer has the following five selectors.

- Selector that selects one of f_x , $f_x/2$, $f_x/4$, or $f_x/8$ as the source clock of the watch timer
- Selector that selects the main clock (f_x) or subclock (f_{XT}) as the clock of the watch timer
- Selector that selects f_w or $f_w/2^9$ as the count clock frequency of the 5-bit counter
- Selector that selects $2^4/f_w$, $2^{13}/f_w$, $2^5/f_w$, or $2^{14}/f_w$ as the INTWT signal generation time interval
- Selector that selects $2^4/f_w$ to $2^{11}/f_w$ as the interval timer interrupt request signal (INTWTI) generation time interval

(7) PRSCM0 register

This is an 8-bit compare register that sets the interval time.

(8) PRSM0 register

This register controls clock supply to the watch timer.

(9) WTM register

This is an 8-bit register that controls the operation of the watch timer/interval timer, and sets the interrupt request signal generation interval.

9.3 Control Registers

The following registers are provided for the watch timer.

- Prescaler mode register 0 (PRSM0)
- Prescaler compare register 0 (PRSCM0)
- Watch timer operation mode register (WTM)

(1) Prescaler mode register 0 (PRSM0)

The PRSM0 register controls the generation of the watch timer count clock.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF8B0H

	7	6	5	<4>	3	2	1	0
PRSM0	0	0	0	BGCE0	0	0	BGCS01	BGCS00

BGCE0	Main clock operation enable
0	Disabled
1	Enabled

BGCS01	BGCS00	Selection of watch timer source clock (f_{BGCS})		
			5 MHz	4 MHz
0	0	f_x	200 ns	250 ns
0	1	$f_x/2$	400 ns	500 ns
1	0	$f_x/4$	800 ns	1 μ s
1	1	$f_x/8$	1.6 μ s	2 μ s

- Cautions**
1. Do not change the values of the BGCS00 and BGCS01 bits during watch timer operation.
 2. Set the PRSM0 register before setting the BGCE0 bit to 1.
 3. Set the PRSM0 and PRSCM0 registers according to the main clock frequency that is used so as to obtain an f_{BRG} frequency of 32.768 kHz.

(2) Prescaler compare register 0 (PRSCM0)

The PRSCM0 register is an 8-bit compare register.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF8B1H

	7	6	5	4	3	2	1	0
PRSCM0	PRSCM07	PRSCM06	PRSCM05	PRSCM04	PRSCM03	PRSCM02	PRSCM01	PRSCM00

- Cautions**
1. Do not rewrite the PRSCM0 register during watch timer operation.
 2. Set the PRSCM0 register before setting the PRSM0.BGCE0 bit to 1.
 3. Set the PRSM0 and PRSCM0 registers according to the main clock frequency that is used so as to obtain an f_{BRG} frequency of 32.768 kHz.

The calculation for f_{BRG} is shown below.

$$f_{BRG} = f_{BGCS}/2N$$

Remark f_{BGCS} : Watch timer source clock set by the PRSM0 register

N: Set value of the PRSCM0 register = 1 to 256

However, N = 256 when the PRSCM0 register is set to 00H.

(3) Watch timer operation mode register (WTM)

The WTM register enables or disables the count clock and operation of the watch timer, sets the interval time of the prescaler, controls the operation of the 5-bit counter, and sets the set time of the watch flag.

Set the PRSM0, PRSCM0 register before setting the WTM register.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF680H

	7	6	5	4	3	2	<1>	<0>
WTM	WTM7	WTM6	WTM5	WTM4	WTM3	WTM2	WTM1	WTM0

WTM7	WTM6	WTM5	WTM4	Selection of interval time of prescaler
0	0	0	0	$2^4/f_w$ (488 μ s: $f_w = f_{XT}$)
0	0	0	1	$2^5/f_w$ (977 μ s: $f_w = f_{XT}$)
0	0	1	0	$2^6/f_w$ (1.95 ms: $f_w = f_{XT}$)
0	0	1	1	$2^7/f_w$ (3.91 ms: $f_w = f_{XT}$)
0	1	0	0	$2^8/f_w$ (7.81 ms: $f_w = f_{XT}$)
0	1	0	1	$2^9/f_w$ (15.6 ms: $f_w = f_{XT}$)
0	1	1	0	$2^{10}/f_w$ (31.3 ms: $f_w = f_{XT}$)
0	1	1	1	$2^{11}/f_w$ (62.5 ms: $f_w = f_{XT}$)
1	0	0	0	$2^4/f_w$ (488 μ s: $f_w = f_{BRG}$)
1	0	0	1	$2^5/f_w$ (977 μ s: $f_w = f_{BRG}$)
1	0	1	0	$2^6/f_w$ (1.95 ms: $f_w = f_{BRG}$)
1	0	1	1	$2^7/f_w$ (3.90 ms: $f_w = f_{BRG}$)
1	1	0	0	$2^8/f_w$ (7.81 ms: $f_w = f_{BRG}$)
1	1	0	1	$2^9/f_w$ (15.6 ms: $f_w = f_{BRG}$)
1	1	1	0	$2^{10}/f_w$ (31.2 ms: $f_w = f_{BRG}$)
1	1	1	1	$2^{11}/f_w$ (62.5 ms: $f_w = f_{BRG}$)

(2/2)

WTM7	WTM3	WTM2	Selection of set time of watch flag
0	0	0	$2^{14}/f_w$ (0.5 s: $f_w = f_{XT}$)
0	0	1	$2^{13}/f_w$ (0.25 s: $f_w = f_{XT}$)
0	1	0	$2^5/f_w$ (977 μ s: $f_w = f_{XT}$)
0	1	1	$2^4/f_w$ (488 μ s: $f_w = f_{XT}$)
1	0	0	$2^{14}/f_w$ (0.5 s: $f_w = f_{BRG}$)
1	0	1	$2^{13}/f_w$ (0.25 s: $f_w = f_{BRG}$)
1	1	0	$2^5/f_w$ (977 μ s: $f_w = f_{BRG}$)
1	1	1	$2^4/f_w$ (488 μ s: $f_w = f_{BRG}$)

WTM1	Control of 5-bit counter operation
0	Clears after operation stops
1	Starts

WTM0	Watch timer operation enable
0	Stops operation (clears both prescaler and 5-bit counter)
1	Enables operation

Caution Rewrite the WTM2 to WTM7 bits while both the WTM0 and WTM1 bits are 0.

- Remarks**
1. f_w : Watch timer clock frequency
 2. Values in parentheses apply to operation with $f_w = 32.768$ kHz

9.4 Operation

9.4.1 Watch timer operations

The watch timer operates on the main clock or subclock (32.768 kHz) and generates an interrupt request signal (INTWT) at fixed, exact time intervals of 0.25 or 0.5 seconds.

Counting starts when the WTM.WTM1 and WTM.WTM0 bits are set to 11. When the WTM0 bit is cleared to 0, the 11-bit prescaler and 5-bit counter are cleared and counting stops.

The time of the watch timer can be adjusted by clearing the WTM1 bit to 0 and then clearing the 5-bit counter when the watch timer is operating at the same time as the interval timer. At this time, an error of up to 15.6 ms may occur in the watch timer, but the interval timer is not affected.

If the main clock is used as the count clock of the watch timer, set the count clock using the PRSM0.BGCS01 and BGCS00 bits and the 8-bit comparison value using the PRSCM0 register, and set the count clock frequency (f_{BRG}) of the watch timer to 32.768 kHz.

When the PRSM0.BGCE0 bit is set to 1, f_{BRG} is supplied to the watch timer.

f_{BRG} can be calculated by using the following expression.

$$f_{BRG} = fx / (2^{m+1} \times N)$$

To set f_{BRG} to 32.768 kHz, perform the following calculation and set the BGCS01 and BGCS00 bits and the PRSCM0 register.

<1> Set $N = fx / 65,536$. Set $m = 0$.

<2> When the value resulting from rounding up the first decimal place of N is even, set N before the roundup as $N/2$ and m as $m + 1$.

<3> Repeat <2> until N is odd or $m = 3$.

<4> Set the value resulting from rounding up the first decimal place of N to the PRSCM0 register and m to the BGCS01 and BGCS00 bits.

Example: When $fx = 4.00$ MHz

<1> $N = 4,000,000 / 65,536 = 61.03\dots$, $m = 0$

<2>, <3> Because N (round up the first decimal place) is odd, $N = 61$, $m = 0$.

<4> Set value of PRSCM0 register: 3DH (61), set value of BGCS01 and BGCS00 bits: 00

At this time, the actual f_{BRG} frequency is as follows.

$$\begin{aligned} f_{BRG} &= fx / (2^{m+1} \times N) = 4,000,000 / (2 \times 61) \\ &= 32.787 \text{ kHz} \end{aligned}$$

Remark m : Division value (set value of BGCS01 and BGCS00 bits) = 0 to 3

N : Set value of PRSCM0 register = 1 to 256

However, $N = 256$ when PRSCM0 register is set to 00H.

fx : Main clock oscillator frequency

9.4.2 Interval timer operations

The watch timer can also be used as an interval timer that repeatedly generates an interrupt request signal (INTWTI) at intervals determined by certain conditions.

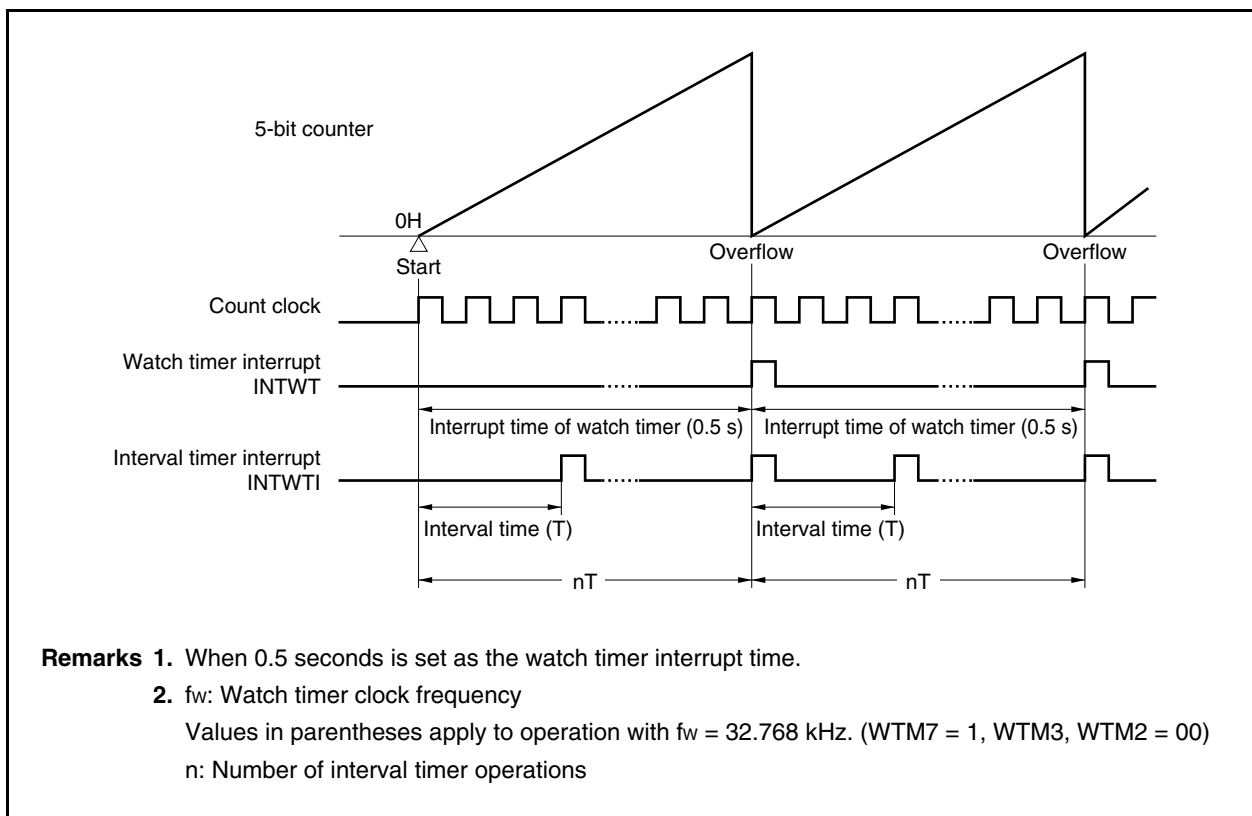
The interval time can be selected by using the WTM4 to WTM7 bits of the WTM register.

Table 9-1. Interval Time of Interval Timer

WTM7	WTM6	WTM5	WTM4	Interval Time	
0	0	0	0	$2^4 \times 1/fw$	488 μ s (operating at $f_w = f_{XT} = 32.768$ kHz)
0	0	0	1	$2^5 \times 1/fw$	977 μ s (operating at $f_w = f_{XT} = 32.768$ kHz)
0	0	1	0	$2^6 \times 1/fw$	1.95 ms (operating at $f_w = f_{XT} = 32.768$ kHz)
0	0	1	1	$2^7 \times 1/fw$	3.91 ms (operating at $f_w = f_{XT} = 32.768$ kHz)
0	1	0	0	$2^8 \times 1/fw$	7.81 ms (operating at $f_w = f_{XT} = 32.768$ kHz)
0	1	0	1	$2^9 \times 1/fw$	15.6 ms (operating at $f_w = f_{XT} = 32.768$ kHz)
0	1	1	0	$2^{10} \times 1/fw$	31.3 ms (operating at $f_w = f_{XT} = 32.768$ kHz)
0	1	1	1	$2^{11} \times 1/fw$	62.5 ms (operating at $f_w = f_{XT} = 32.768$ kHz)
1	0	0	0	$2^4 \times 1/fw$	488 μ s (operating at $f_w = f_{BRG} = 32.768$ kHz)
1	0	0	1	$2^5 \times 1/fw$	977 μ s (operating at $f_w = f_{BRG} = 32.768$ kHz)
1	0	1	0	$2^6 \times 1/fw$	1.95 ms (operating at $f_w = f_{BRG} = 32.768$ kHz)
1	0	1	1	$2^7 \times 1/fw$	3.91 ms (operating at $f_w = f_{BRG} = 32.768$ kHz)
1	1	0	0	$2^8 \times 1/fw$	7.81 ms (operating at $f_w = f_{BRG} = 32.768$ kHz)
1	1	0	1	$2^9 \times 1/fw$	15.6 ms (operating at $f_w = f_{BRG} = 32.768$ kHz)
1	1	1	0	$2^{10} \times 1/fw$	31.3 ms (operating at $f_w = f_{BRG} = 32.768$ kHz)
1	1	1	1	$2^{11} \times 1/fw$	62.5 ms (operating at $f_w = f_{BRG} = 32.768$ kHz)

Remark fw: Watch timer clock frequency

Figure 9-2. Timing of Watch Timer and Interval Timer Operations

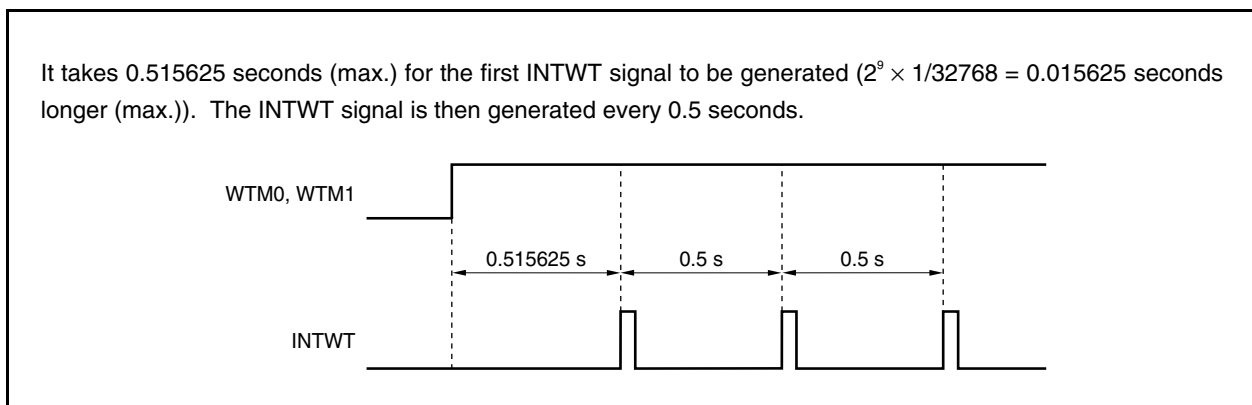


9.5 Cautions

(1) Operation as watch timer

The first watch timer interrupt request signal (INTWT) is not generated at the exact time specified using the WTM2 and WTM3 bits after operation is enabled (WTM.WTM1 and WTM.WTM0 bits = 11). The second and subsequent INTWT signals are generated at the specified time.

**Figure 9-3. Example of Generation of Watch Timer Interrupt Request Signal (INTWT)
(When Interrupt Cycle = 0.5 s)**



(2) When watch timer and interval timer WT operate simultaneously

The interval time of interval timer WT can be set to a value between 488 μ s and 62.5 ms. It cannot be changed later.

Do not stop interval timer WT (by clearing the WTM.WTM0 bit to 0) while the watch timer is operating. If the WTM0 bit is set to 1 again after it had been cleared to 0, the watch timer will have a discrepancy of up to 0.5 or 0.25 seconds.

CHAPTER 10 REAL-TIME COUNTER

10.1 Functions

The real-time counter (RTC) has the following features.

- Counting up to 99 years using year, month, day-of-week, day, hour, minute, and second sub-counters provided
- Year, month, day-of-week, day, hour, minute, and second counter display using BCD codes^{Note 1}
- Alarm interrupt function
- Constant-period interrupt function (period: 1 month to 0.5 second)
- Interval interrupt function (period: 1.95 ms to 125 ms)
- Pin output function of 1 Hz
- Pin output function of 32.768 kHz (V850ES/JE3-L only)
- Pin output function of 512 Hz or 16.384 kHz (V850ES/JE3-L only)
- Watch error correction function
- Subclock operation or main clock operation^{Note 2} selectable

Notes 1. A BCD (binary coded decimal) code expresses each digit of a decimal number in 4-bit binary format.

- 2.** Use the baud rate generator dedicated to the real-time counter to divide the main clock frequency to 32.768 kHz for use.

Caution The watch timer and RTC alternate interrupt signal and therefore cannot be used simultaneously.

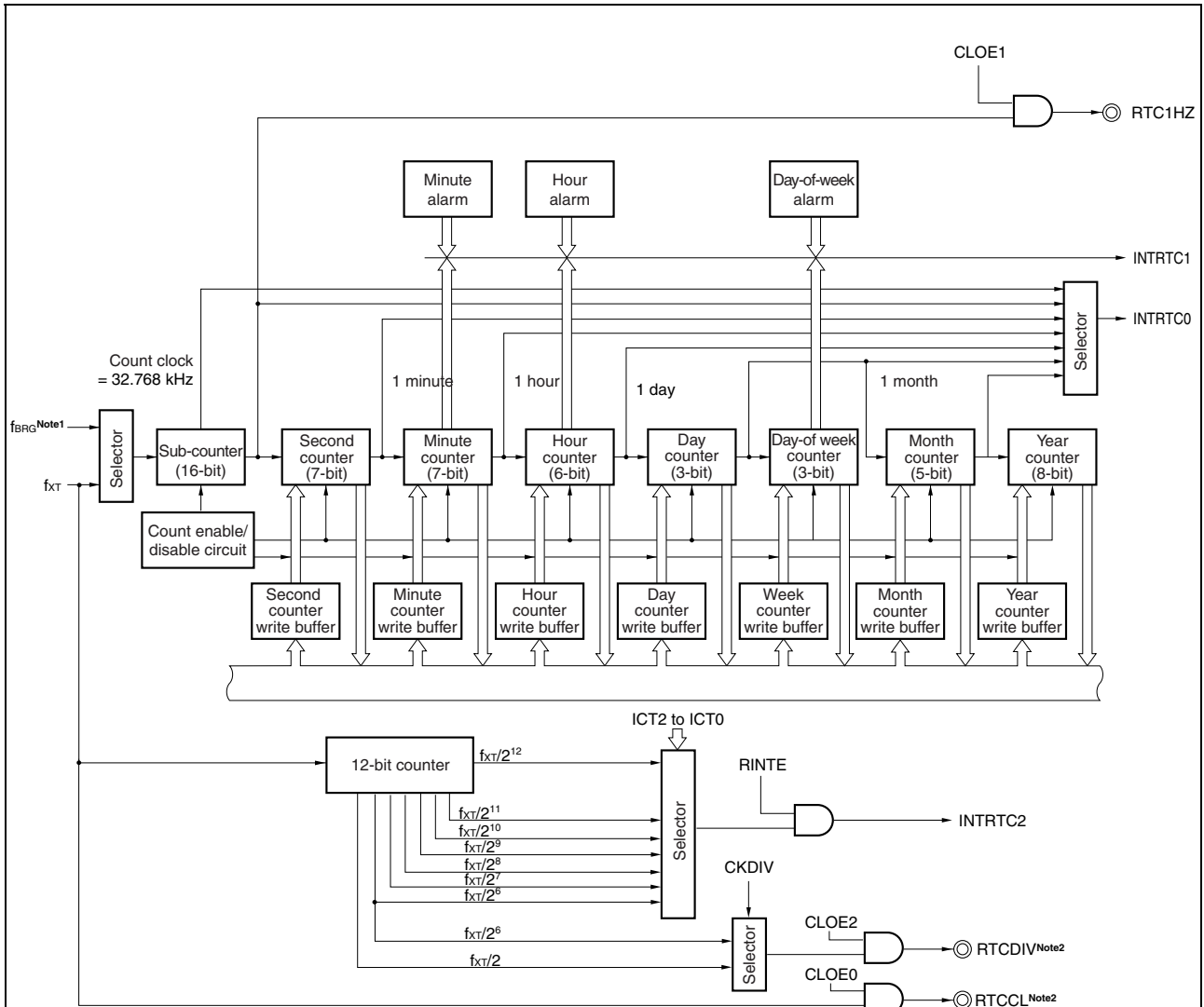
10.2 Configuration

The real-time counter includes the following hardware.

Table 10-1. Configuration of Real-Time Counter

Item	Configuration
Control registers	Real-time counter control register 0 (RC1CC0) Real-time counter control register 1 (RC1CC1) Real-time counter control register 2 (RC1CC2) Real-time counter control register 3 (RC1CC3) Sub-count register (RC1SUBC) Second count register (RC1SEC) Minute count register (RC1MIN) Hour count register (RC1HOUR) Day count register (RC1DAY) Day-of-week count register (RC1WEEK) Month count register (RC1MONTH) Year count register (RC1YEAR) Watch error correction register (RC1SUBU) Alarm minute register (RC1ALM) Alarm hour register (RC1ALH) Alarm week register (RC1ALW) Prescaler mode register 0 (PRSM0) Prescaler compare register 0 (PRSCM0)

Figure 10-1. Block Diagram of Real-Time Counter



Notes 1. For detail of f_{BRG} , refer to 10.3 (17) Prescaler mode register 0 (PRSM0) and 10.3 (18) Prescaler compare register 0 (PRSCM0).

2. V850ES/JE3-L only.

- Remark**
- f_{BRG} : Real-time counter count clock frequency
 - f_{XT} : Subclock frequency
 - INTRTC0: Real-time counter fixed-cycle interrupt signal
 - INTRTC1: Real-time counter alarm match interrupt signal
 - INTRTC2: Real-time counter interval interrupt signal

10.2.1 Pin configuration

The RTC outputs included in the real-time counter are alternatively used as shown in Table 10-2. The port function must be set when using each pin (see Table 4-17 **Settings When Pins Are Used for Alternate Functions**).

Table 10-2. Pin Configuration

Pin Number			Port	RTC Output	Other Alternate Function
JC3L(40)	JC3L(48)	JE3L			
12	14	18	P03	RTC1HZ	INTP0/ADTRG
–	–	19	P04	RTCDIV	INTP1/RTCCL
–	–	19	P04	RTCCL	INTP1/RTCDIV

Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

10.2.2 Interrupt functions

The RTC includes the following three types of interrupt signals.

(1) INTRTC0

A fixed-cycle interrupt signal is generated every 0.5 second, second, minute, hour, day, or month.

(2) INTRTC1

Alarm interrupt signal

(3) INTRTC2

An interval interrupt signal of a cycle of $f_{XT}/2^6$, $f_{XT}/2^7$, $f_{XT}/2^8$, $f_{XT}/2^9$, $f_{XT}/2^{10}$, $f_{XT}/2^{11}$, or $f_{XT}/2^{12}$ is generated.

10.3 Registers

The real-time counter is controlled by the following 18 registers.

(1) Real-time counter control register 0 (RC1CC0)

The RC1CC0 register selects the real-time counter input clock.

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

Reset sets this register to 00H.

After reset: 00H		R/W	Address: FFFFFADDH					
	7	6	5	4	3	2	1	0
RC1CC0	RC1PWR	RC1CKS	0	0	0	0	0	0
	RC1PWR	Real-time counter operation control						
	0	Stops real-time counter operation.						
	1	Enables real-time counter operation.						
	RC1CKS	Operation clock selection						
	0	Selects f_{XT} as operation clock.						
	1	Selects f_{BRG} as operation clock.						

Cautions

1. Follow the description in 10.4.8 Initializing real-time counter when stopping (RC1PWR = 1 → 0) the real-time counter while it is operating.
2. The RC1CKS bit can be rewritten only when the real-time counter is stopped (RC1PWR bit = 0). Furthermore, rewriting the RC1CKS bit at the same time as setting the RC1PWR bit from 0 to 1 is prohibited.

(2) Real-time counter control register 1 (RC1CC1)

The RC1CC1 register is an 8-bit register that starts or stops the real-time counter, controls the RTCCL and RTC1HZ pins, selects the 12-hour or 24-hour system, and sets the fixed-cycle interrupt function.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFADEH

	7	6	5	4	3	2	1	0
RC1CC1	RTCE	0	CLOE1	CLOE0 ^{Note}	AMPM	CT2	CT1	CT0

RTCE	Control of operation of each counter
0	Stops counter operation.
1	Enables counter operation.

CLOE1	RTC1HZ pin output control
0	Disables RTC1HZ pin output (1 Hz)
1	Enables RTC1HZ pin output (1 Hz)

CLOE0 ^{Note}	RTCCL pin output control
0	Disables RTCCL pin output (32.768 kHz)
1	Enables RTCCL pin output (32.768 kHz)

AMPM	12-hour system/24-hour system selection
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system

CT2	CT1	CT0	Fixed-cycle interrupt (INTRTC0) selection
0	0	0	Does not use fixed-cycle interrupts
0	0	1	Once in 0.5 second (synchronous with second count-up)
0	1	0	Once in 1 second (simultaneous with second count-up)
0	1	1	Once in 1 minute (every minute at 00 seconds)
1	0	0	Once in 1 hour (every hour at 00 minutes 00 seconds)
1	0	1	Once in 1 day (every day at 00 hours 00 minutes 00 seconds)
1	1	×	Once in 1 month (one day every month at 00 hours 00 minutes 00 seconds a.m.)

Note V850ES/JE3-L only

- Cautions**
- Writing 0 to the RTCE bit while the RTCE bit is 1 is prohibited. Clear the RTCE bit by clearing the RC1PWR bit according to 10.4.8 Initializing real-time counter.
 - The RTC1HZ output operates as follows when the CLOE1 bit setting is changed.
 - When changed from 0 to 1: The RTC1HZ output outputs a 1 Hz pulse after two clocks or less (2×32.768 kHz).
 - When changed from 1 to 0: The RTC1HZ output is stopped (fixed to low level) after two clocks or less (2×32.768 kHz).
 - See 10.4.1 Initial settings and 10.4.2 Rewriting each counter during real-time counter operation for setting or changing the AMPM bit. Furthermore, re-set the RC1HOUR register when the AMPM bit is rewritten.
 - See 10.4.4 Changing INTRTC0 interrupt setting during real-time counter operation when rewriting the CT2 to CT0 bits while the real-time counter operates (RC1PWR bit = 1).

(3) Real-time counter control register 2 (RC1CC2)

The RC1CC2 register is an 8-bit register that controls the alarm interrupt function and waiting of counters. This register can be read or written in 8-bit or 1-bit units. Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFADFH

RC1CC2	7	6	5	4	3	2	1	0
	WALE	0	0	0	0	0	RWST	RWAIT

WALE	Alarm interrupt (INTRTC1) operation control
0	Does not generate interrupt upon alarm match.
1	Generates interrupt upon alarm match.

RWST	Real-time counter wait state
0	Counter operating
1	Counting up of second to year counters stopped (Reading and writing of counter values enabled)

This is a status flag indicating whether the RWAIT bit setting is valid.
Read or write counter values after confirming that the RWST bit is 1.

RWAIT	Real-time counter wait control
0	Sets counter operation.
1	Stops count operation of second to year counters. (Counter value read/write mode)

This bit controls the operation of the counters.
Be sure to write 1 to this bit when reading or writing counter values.
If the RC1SUBC register overflows while the RWAIT bit is 1, the overflow information is retained internally and the RC1SEC register is counted up after two clocks or less (2×32.768 kHz) after 0 is written to the RWAIT bit.
However, if the second counter value is rewritten while the RWAIT bit is 1, the retained overflow information is discarded.

Cautions

1. See 10.4.5 Changing INTRTC1 interrupt setting during real-time counter operation when rewriting the WALE bit while the real-time counter operates (RC1PWR bit = 1).
2. Confirm that the RWST bit is set to 1 when reading or writing each counter value.
3. The RWST bit does not become 0 while each counter is being written, even if the RWAIT bit is set to 0. It becomes 0 when writing to each counter is completed.

(4) Real-time counter control register 3 (RC1CC3)

The RC1CC3 register is an 8-bit register that controls the interval interrupt function and RTCDIV pin.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFFAE0H

	7	6	5	4	3	2	1	0
RC1CC3	RINTE	CLOE2 ^{Note}	CKDIV ^{Note}	0	0	ICT2	ICT1	ICT0

RINTE	Interval interrupt (INTRTC2) control
0	Does not generate interval interrupt.
1	Generates interval interrupt.

CLOE2 ^{Note}	RTCDIV pin output control
0	Disables RTCDIV pin output.
1	Enables RTCDIV pin output.

CKDIV ^{Note}	RTCDIV pin output frequency selection
0	Outputs 512 Hz (1.95 ms) from RTCDIV pin.
1	Outputs 16.384 kHz (0.061 ms) from RTCDIV pin.

ICT2	ICT1	ICT0	Interval interrupt (INTRTC2) selection
0	0	0	$2^6/f_{XT}$ (1.953125 ms)
0	0	1	$2^7/f_{XT}$ (3.90625 ms)
0	1	0	$2^8/f_{XT}$ (7.8125 ms)
0	1	1	$2^9/f_{XT}$ (15.625 ms)
1	0	0	$2^{10}/f_{XT}$ (31.25 ms)
1	0	1	$2^{11}/f_{XT}$ (62.5 ms)
1	1	×	$2^{12}/f_{XT}$ (125 ms)

Note V850ES/JE3-L only.

- Cautions**
- See 10.4.7 Changing INTRTC2 interrupt setting during real-time counter operation when rewriting the RINTE bit during real-time counter operation (RC1PWR bit = 1).
 - The RTCDIV output operates as follows when the CLOE2 bit setting is changed.
 - When changed from 0 to 1: A pulse set by the CKDIV bit is output after two clocks or less ($2 \times 32.768\text{kHz}$).
 - When changed from 1 to 0: Output of the RTCDIV output is stopped after two clocks or less (fixed to low level, $2 \times 32.768\text{kHz}$).
 - See 10.4.7 Changing INTRTC2 interrupt setting during real-time counter operation when rewriting the ICT2 to ICT0 bits while the real-time counter operates (RC1PWR bit = 1).

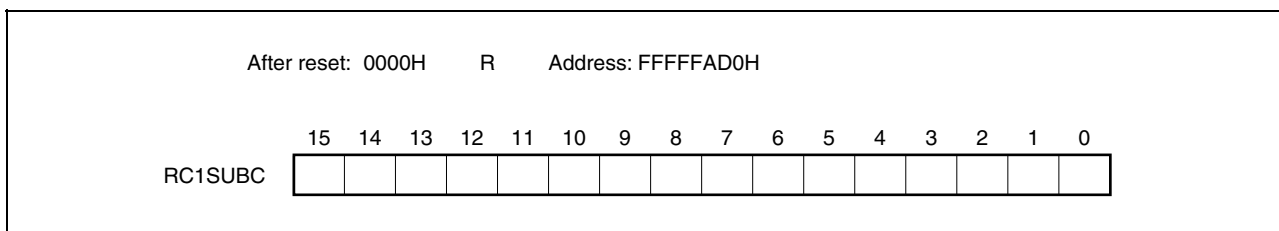
(5) Sub-count register (RC1SUBC)

The RC1SUBC register is a 16-bit register that counts the reference time of 1 second of the real-time counter. It takes a value of 0000H to 7FFFH and counts one second with a clock of 32.768 kHz.

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

Reset sets this register to 0000H.

- Cautions**
1. When a correction is made by using the RC1SUBU register, the value may become 8000H or more.
 2. This register is also cleared by writing to the second count register.
 3. The value read from this register is not guaranteed if it is read during operation, because a changing value is read.

**(6) Second count register (RC1SEC)**

The RC1SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds.

It counts up when the sub-counter overflows.

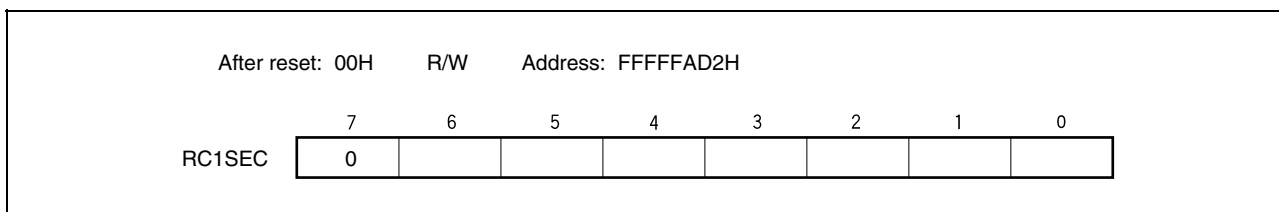
When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (2×32.768 kHz) later. Set a decimal value of 00 to 59 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after one period.

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

Reset sets this register to 00H.

Caution Setting the RC1SEC register to values other than 00 to 59 is prohibited.

Remark See 10.4.1 Initial settings, 10.4.2 Rewriting each counter during real-time counter operation, and 10.4.3 Reading each counter during real-time counter operation when reading or writing the RC1SEC register.



(7) Minute count register (RC1MIN)

The RC1MIN register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes.

It counts up when the second counter overflows.

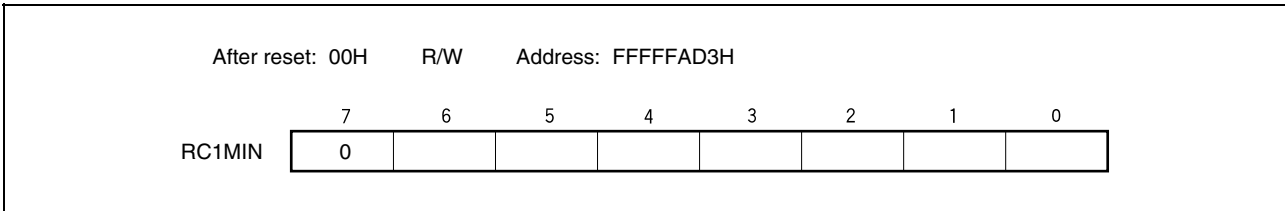
When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (2 × 32.768 kHz) later. Set a decimal value of 00 to 59 to this register in BCD code.

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

Reset sets this register to 00H.

Caution Setting a value other than 00 to 59 to the RC1MIN register is prohibited.

Remark See 10.4.1 Initial settings, 10.4.2 Rewriting each counter during real-time counter operation, and 10.4.3 Reading each counter during real-time counter operation when reading or writing the RC1MIN register.



(8) Hour count register (RC1HOUR)

The RC1HOUR register is an 8-bit register that takes a value of 0 to 23 or 1 to 12 (decimal) and indicates the count value of hours.

It counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (2 × 32.768 kHz) later. Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after 1 period.

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

Reset sets this register to 12H.

However, the value of this register is 00H if the AMPM bit is set to 1 after reset.

- Cautions**
1. Bit 5 of the RC1HOUR register indicates a.m. (0) or p.m. (1) if AMPM = 0 (if the 12-hour system is selected).
 2. Setting a value other than 01 to 12, 21 to 32 (AMPM bit= 0), or 00 to 23 (AMPM bit = 1) to the RC1HOUR register is prohibited.

Remark See 10.4.1 Initial settings, 10.4.2 Rewriting each counter during real-time counter operation, and 10.4.3 Reading each counter during real-time counter operation when reading or writing the RC1HOUR register.

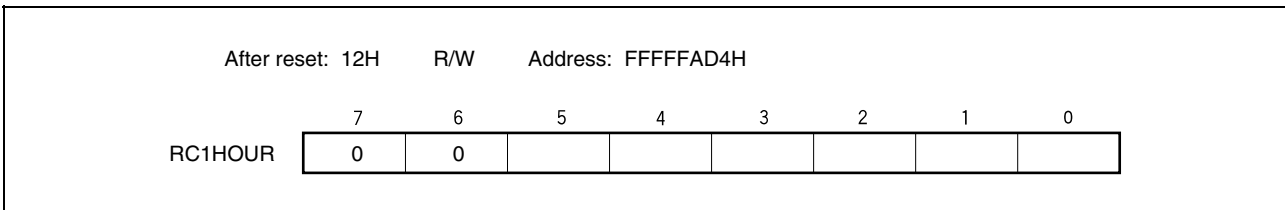


Table 10-3 shows the relationship among the AMPM bit setting value, RC1HOUR register value, and time.

Table 10-3. Time Digit Display

12-Hour Display (AMPM Bit = 0)		24-Hour Display (AMPM Bit = 1)	
Time	RC1HOUR Register Value	Time	RC1HOUR Register Value
0:00 a.m.	12 H	0:00	00H
1:00 a.m.	01 H	1:00	01 H
2:00 a.m.	02 H	2:00	02 H
3:00 a.m.	03 H	3:00	03 H
4:00 a.m.	04 H	4:00	04 H
5:00 a.m.	05 H	5:00	05 H
6:00 a.m.	06 H	6:00	06 H
7:00 a.m.	07 H	7:00	07 H
8:00 a.m.	08 H	8:00	08 H
9:00 a.m.	09 H	9:00	09 H
10:00 a.m.	10 H	10:00	10 H
11:00 a.m.	11 H	11:00	11 H
0:00 p.m.	32 H	12:00	12 H
1:00 p.m.	21 H	13:00	13 H
2:00 p.m.	22 H	14:00	14 H
3 :00 p.m.	23 H	15:00	15 H
4:00 p.m.	24 H	16:00	16 H
5:00 p.m.	25 H	17:00	17 H
6:00 p.m.	26 H	18:00	18 H
7:00 p.m.	27 H	19:00	19 H
8:00 p.m.	28 H	20:00	20 H
9:00 p.m.	29 H	21:00	21 H
10:00 p.m.	30 H	22:00	22 H
11:00 p.m.	31 H	23:00	23 H

The RC1HOUR register value is displayed in 12 hour-format if the AMPM bit is 0 and in 24-hour format when the AMPM bit is 1.

In 12-hour display, a.m. or p.m. is indicated by the fifth bit of RCHOUR: 0 indicating before noon (a.m.) and 1 indicating noon or afternoon (p.m.).

(9) Day count register (RC1DAY)

The RC1DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days.

It counts up when the hour counter overflows.

This counter counts as follows.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February in leap year)
- 01 to 28 (February in normal year)

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later.

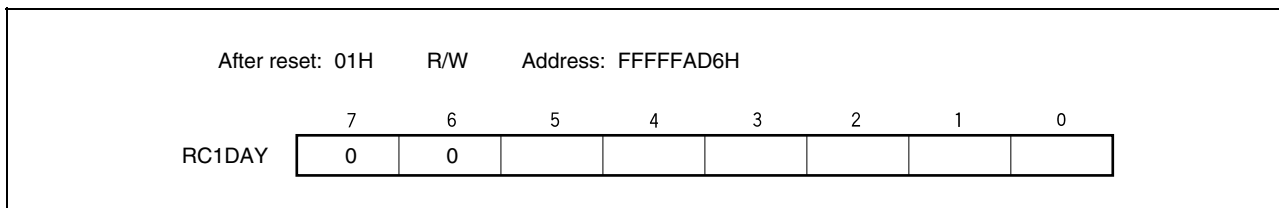
Set a decimal value of 00 to 31 to this register in BCD code.

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

Reset sets this register to 01H.

Caution Setting a value other than 01 to 31 to the RC1DAY register is prohibited. Setting a value outside the above-mentioned count range, such as “February 30” is also prohibited.

Remark See 10.4.1 Initial settings, 10.4.2 Rewriting each counter during real-time counter operation, and 10.4.3 Reading each counter during real-time counter operation when reading or writing the RC1DAY register.



(10) Day-of-week count register (RC1WEEK)

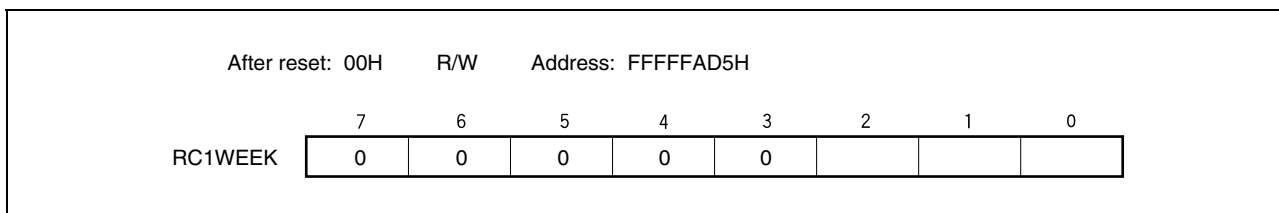
The RC1WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the day-of-week count value.

It counts up in synchronization with the day counter.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (2×32.768 kHz) later. Set a decimal value of 00 to 06 to this register in BCD code. If a value outside this range is set, the register value returns to the normal value after 1 period.

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

Reset sets this register to 00H.



- Cautions**
1. Setting a value other than 00 to 06 to the RC1WEEK register is prohibited.
 2. Values corresponding to the month count register and day count register are not automatically stored to the day-of-week register.
Be sure to set as follows after reset.

Day of Week	RC1WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

Remark See 10.4.1 Initial settings, 10.4.2 Rewriting each counter during real-time counter operation, and 10.4.3 Reading each counter during real-time counter operation when reading or writing the RC1WEEK register.

(11) Month count register (RC1MONTH)

The RC1MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.

It counts up when the day counter overflows.

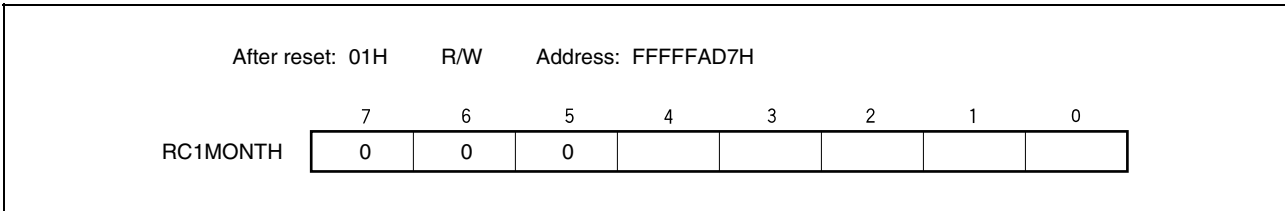
When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (2×32.768 kHz) later. Set a decimal value of 01 to 12 to this register in BCD code.

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

Reset sets this register to 01H.

Caution Setting a value other than 01 to 12 to the RC1MONTH register is prohibited.

Remark See 10.4.1 Initial settings, 10.4.2 Rewriting each counter during real-time counter operation, and 10.4.3 Reading each counter during real-time counter operation when reading or writing the RC1MONTH register.



(12) Year count register (RC1YEAR)

The RC1YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.

It counts up when the month counter overflows.

Values 00, 04, 08, ..., 92, and 96 indicate a leap year.

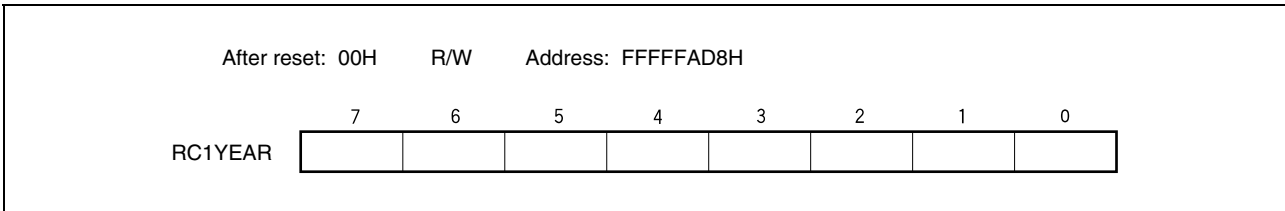
When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (2×32.768 kHz) later. Set a decimal value of 00 to 99 to this register in BCD code.

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

Reset sets this register to 00H.

Caution Setting a value other than 00 to 99 to the RC1YEAR register is prohibited.

Remark See 10.4.1 Initial settings, 10.4.2 Rewriting each counter during real-time counter operation, and 10.4.3 Reading each counter during real-time counter operation when reading or writing the RC1YEAR register.



(13) Watch error correction register (RC1SUBU)

The RC1SUBU register is an 8-bit register that can be used to correct the watch with high accuracy when the watch is early or late, by changing the value (reference value: 7FFFH) overflowing from the sub-count register (RSUBC) to the second counter register.

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

Reset sets this register to 00H.

- Remarks 1.** The RC1SUBU register can be rewritten only when the real-time counter is set to its initial values. Be sure to see **10.4.1 Initial settings**.
- 2.** See **10.4.9 Watch error correction example of real-time counter** for details of watch error correction.

After reset: 00H R/W Address: FFFFFAD9H

	7	6	5	4	3	2	1	0
RC1SUBU	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of watch error correction timing
0	Corrects watch errors when RC1SEC (second counter) is at 00, 20, or 40 seconds (every 20 seconds).
1	Corrects watch errors when RC1SEC (second counter) is at 00 seconds (every 60 seconds).

F6	Setting of watch error correction value
0	Increments the RC1SUBC count value by the value set using the F5 to F0 bits (positive correction). Expression for calculating increment value: $(\text{Setting value of F5 to F0 bits} - 1) \times 2$
1	Decrements the RC1SUBC count value by the value set using the F5 to F0 bits (negative correction). Expression for calculating decrement value: $(\text{Inverted value of setting value of F5 to F0 bits} + 1) \times 2$
If the F6 to F0 bit values are {1/0, 0, 0, 0, 0, 0, 1/0}, watch error correction is not performed.	

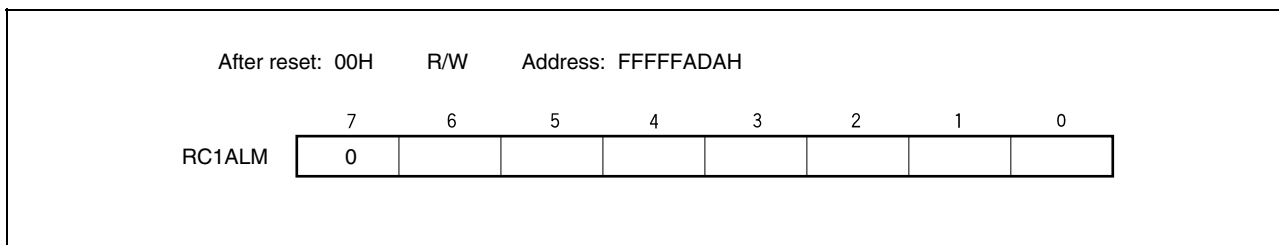
(14) Alarm minute setting register (RC1ALM)

The RC1ALM register is an 8-bit register that is used to set minutes of alarm.

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

Reset sets this register to 00H.

Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.



(15) Alarm hour setting register (RC1ALH)

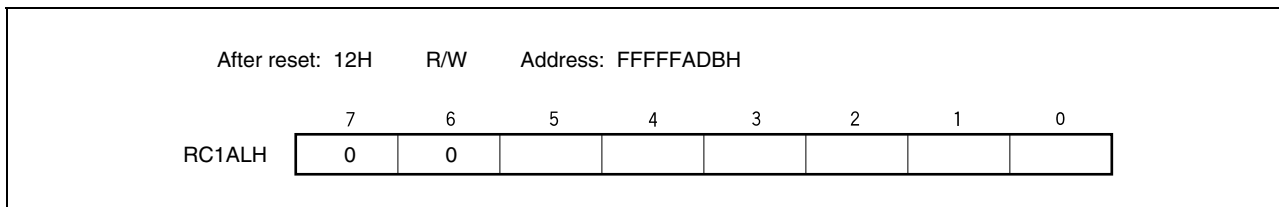
The RC1ALH register is an 8-bit register that is used to set hours of alarm.

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

Reset sets this register to 12H.

Cautions 1. Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

2. Bit 5 of the RC1ALH register indicates a.m. (0) or p.m. (1) if the AMPM bit = 0 (12-hour system) is selected.



(16) Alarm day-of-week setting register (RC1ALW)

The RC1ALW register is an 8-bit register that is used to set the day-of-week of the alarm.

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

Reset sets this register to 00H.

Caution See 10.4.5 Changing INTRTC1 interrupt setting during clock operation when rewriting the RC1ALW register while the real-time counter operates (RC1PWR bit = 1).

After reset: 00H	R/W	Address: FFFFFADCH						
7	6	5	4	3	2	1	0	
RC1ALW	0	RC1ALW6	RC1ALW5	RC1ALW4	RC1ALW3	RC1ALW2	RC1ALW1	RC1ALW0
RC1ALWn	Alarm interrupt day-of-week bit (n = 0 to 6)							
0	Does not generate alarm interrupt if RC1WEEK = nH.							
1	Generates an alarm interrupt if the time specified by using the RC1ALM and RC1ALH registers is reached while RC1WEEK is set to nH.							

Remark The relationship between the day-of-week and the RC1WEEK register is described below.

Day of Week	RC1WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

(a) Alarm interrupt setting examples (RC1ALM, RC1ALH, and RC1ALW setting examples)

Tables 10-4 and 10-5 show setting examples if Sunday is RC1WEEK = 00, Monday is RC1WEEK = 01, Tuesday is RC1WEEK = 02, ..., and Saturday is RC1WEEK = 06.

Table 10-4. Alarm Setting Example if AMPM = 0 (RC1HOUR Register 12-Hour Display)

	Register	RC1ALW	RC1ALH	RC1ALM
Alarm Setting Time				
Sunday, 7:00 a.m.		01H	07H	00H
Sunday/Monday, 00:15 p.m.		03H	32H	15H
Monday/Tuesday/Friday, 5:30 p.m.		26H	25H	30H
Everyday, 10:45 p.m.		7FH	30H	45H

Table 10-5. Alarm Setting Example if AMPM = 1 (RC1HOUR Register 24-Hour Display)

Alarm Setting Time	Register	RC1ALW	RC1ALH	RC1ALM
Sunday, 7:00		01H	07H	00H
Sunday/Monday, 12:15		03H	12H	15H
Monday/Tuesday/Friday, 17:30		26H	17H	30H
Everyday, 22:45		7FH	22H	45H

(17) Prescaler mode register 0 (PRSM0)

The PRSM0 register is an 8-bit register that controls the generation of the real time counter count clock (f_{BRG}).

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

Reset sets this register to 00H.

After reset: 00H R/W Address : FFFF8B0H

	7	6	5	<4>	3	2	1	0
PRSM0	0	0	0	BGCE0	0	0	BGCS01	BGCS00

BGCE0	Main clock operation enable
0	Disabled
1	Enabled

BGCS01	BGCS00	Selection of real time counter source clock(f_{BGS})		
			5 MHz	4 MHz
0	0	f_x	200 ns	250 ns
0	1	$f_x/2$	400 ns	500 ns
1	0	$f_x/4$	800 ns	1 μ s
1	1	$f_x/8$	1.6 μ s	2 μ s

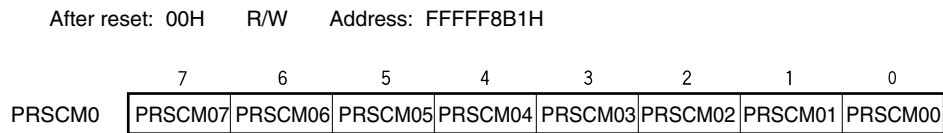
- Cautions**
1. Do not change the values of the BGCS00 and BGCS01 bits during real time counteroperation.
 2. Set the PRSM0 register before setting the BGCE0 bit to 1.
 3. Set the PRSM0 and PRSCM0 registers according to the main clock frequency that is used so as to obtain an f_{BRG} frequency of 32.768 kHz.

(18) Prescaler compare register 0 (PRSCM0)

The PRSCM0 register is an 8-bit compare register.

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

Reset sets this register to 00H.



- Cautions**
1. Do not rewrite the PRSCM0 register during real time counter operation.
 2. Set the PRSCM0 register before setting the PRSM0.BGCE0 bit to 1.
 3. Set the PRSM0 and PRSCM0 registers according to the main clock frequency that is used so as to obtain an f_{BRG} frequency of 32.768 kHz.

The calculation for f_{BRG} is shown below.

$$f_{BRG} = f_{BGCS}/2N$$

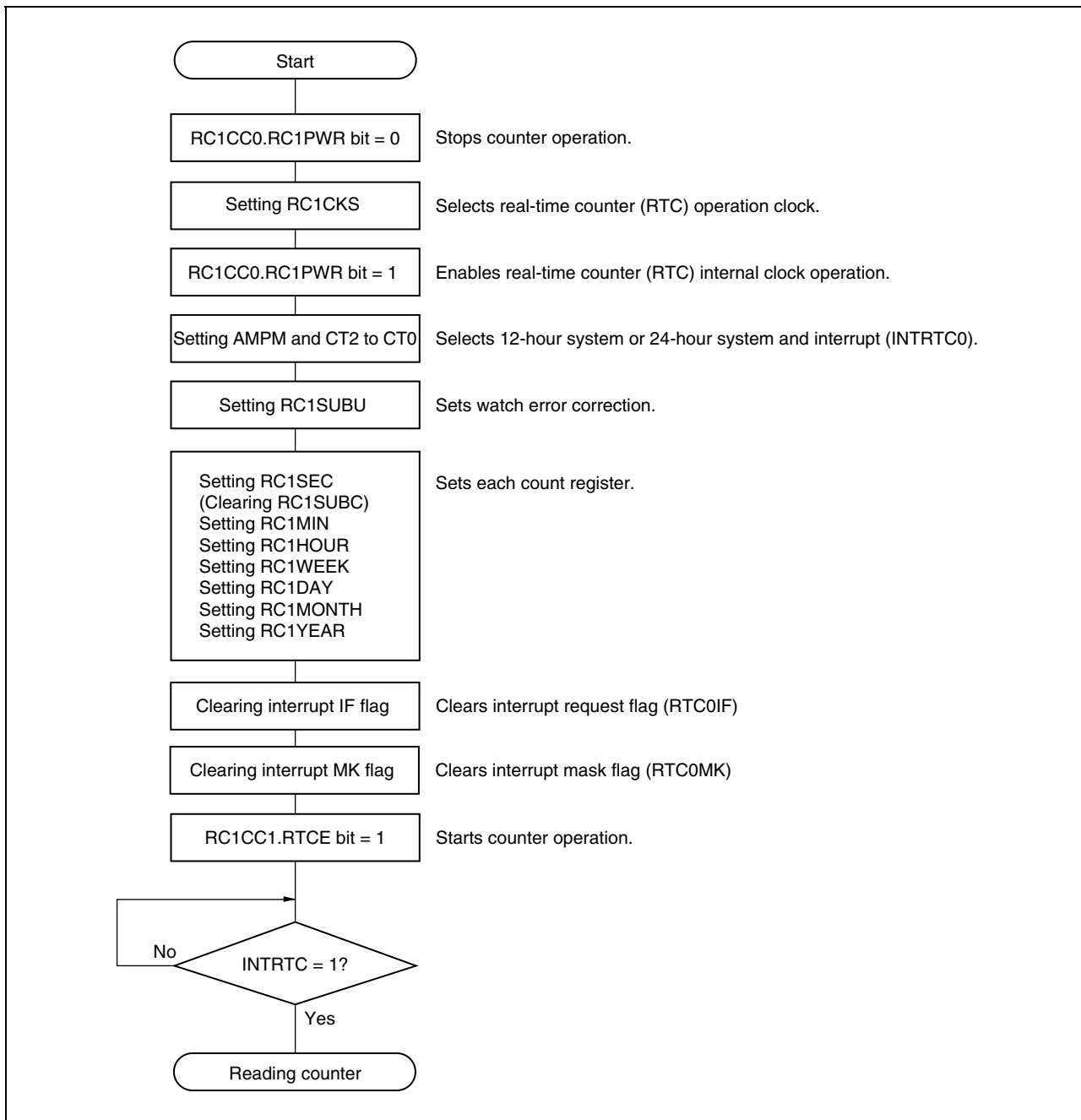
- Remark**
- f_{BGCS} : Watch timer source clock set by the PRSM0 register
 - N: Set value of the PRSCM0 register = 1 to 256
However, N = 256 when the PRSCM0 register is set to 00H.

10.4 Operation

10.4.1 Initial settings

The initial settings are set when operating the watch function and performing a fixed-cycle interrupt operation.

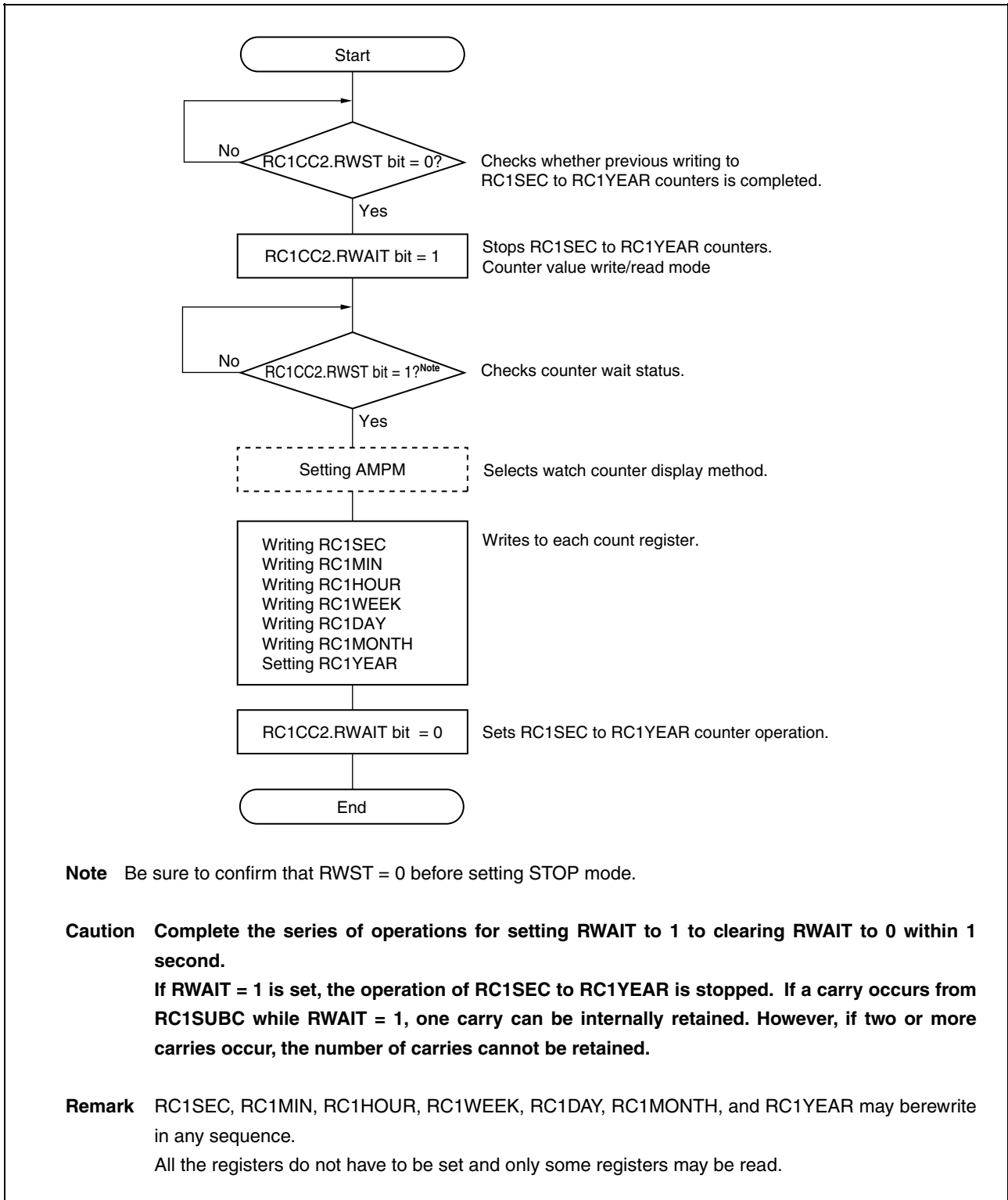
Figure 10-2. Initial Setting Procedure



10.4.2 Rewriting each counter during real-time counter operation

Set as follows when rewriting each counter (RC1SEC, RC1MIN, RC1HOUR, RC1WEEK, RC1DAY, RC1MONTH, RC1YEAR) during real-time counter operation (RC1PWR = 1).

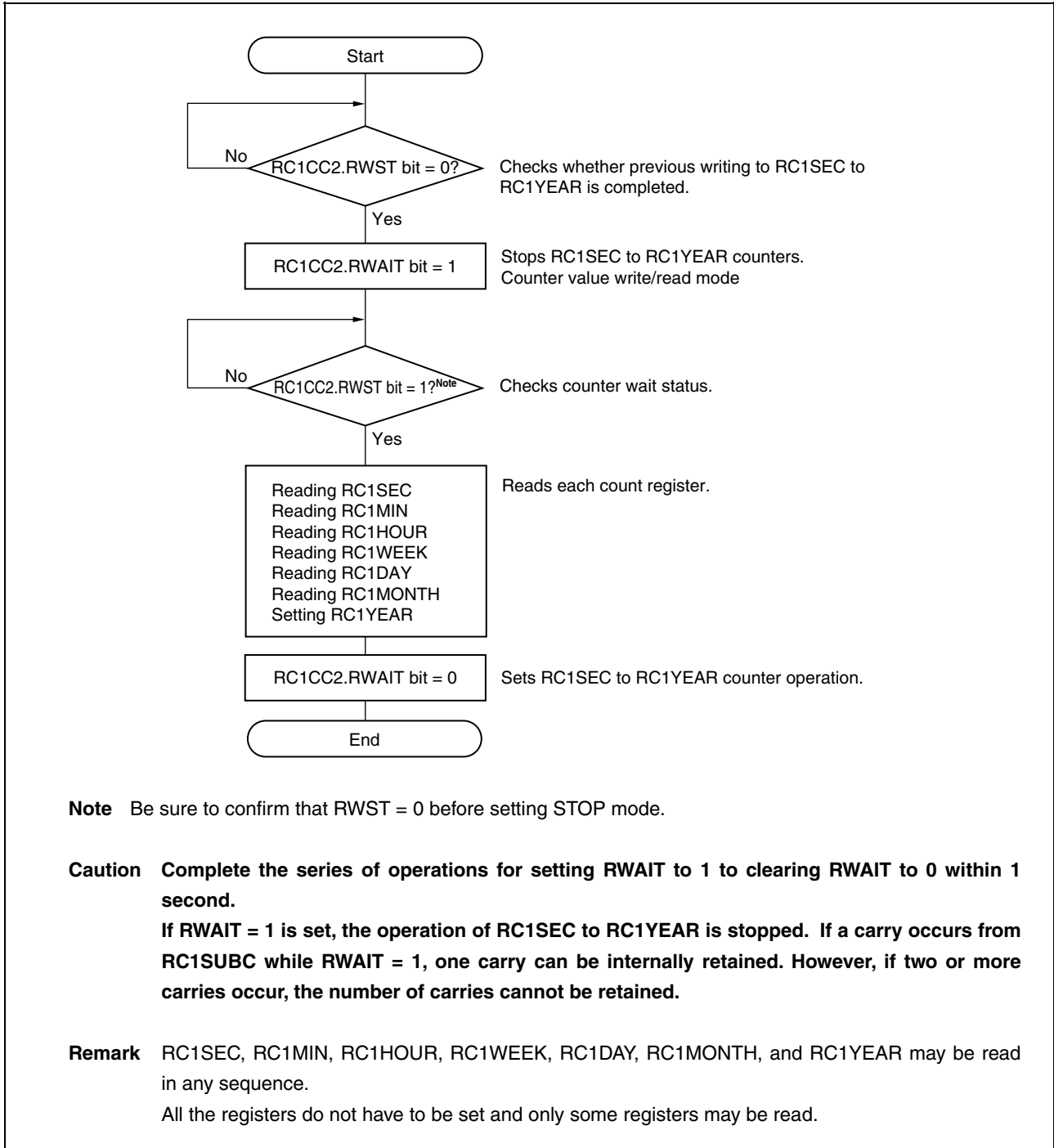
Figure 10-3. Rewriting Each Counter During Clock Operation



10.4.3 Reading each counter during real-time counter operation

Set as follows when reading each counter (RC1SEC, RC1MIN, RC1HOUR, RC1WEEK, RC1DAY, RC1MONTH, RC1YEAR) during real-time counter operation (RC1PWR = 1, RTCE = 1).

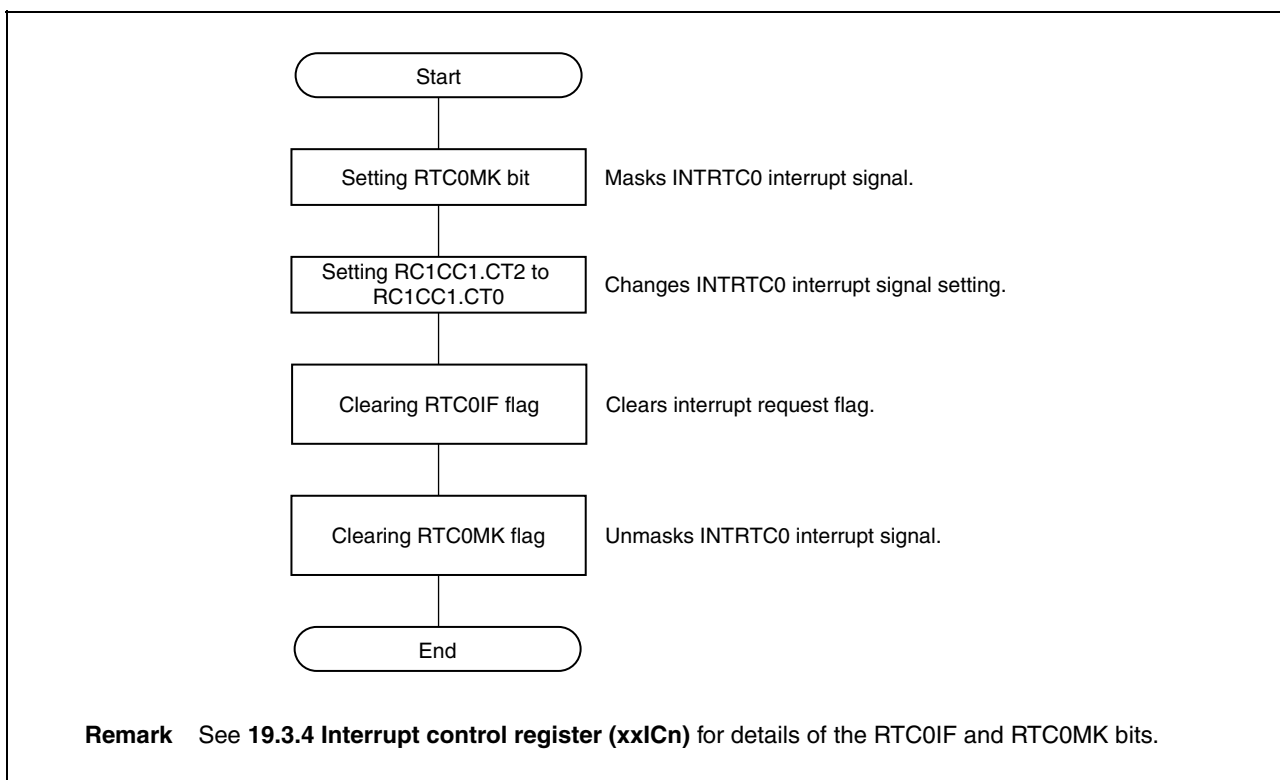
Figure 10-4. Reading Each Counter During Clock Operation



10.4.4 Changing INTRTC0 interrupt setting during real-time counter operation

If the setting of the INTRTC0 interrupt (fixed-cycle interrupt) signal is changed while the real-time counter clock operates (PC1PWR = 1, RTCE =1), the INTRTC0 interrupt waveform may include whiskers and unintended signals may be output. Set as follows when changing the setting of the INTRTC0 interrupt signal during real-time counter operation (RC1PWR = 1), in order to mask the whiskers.

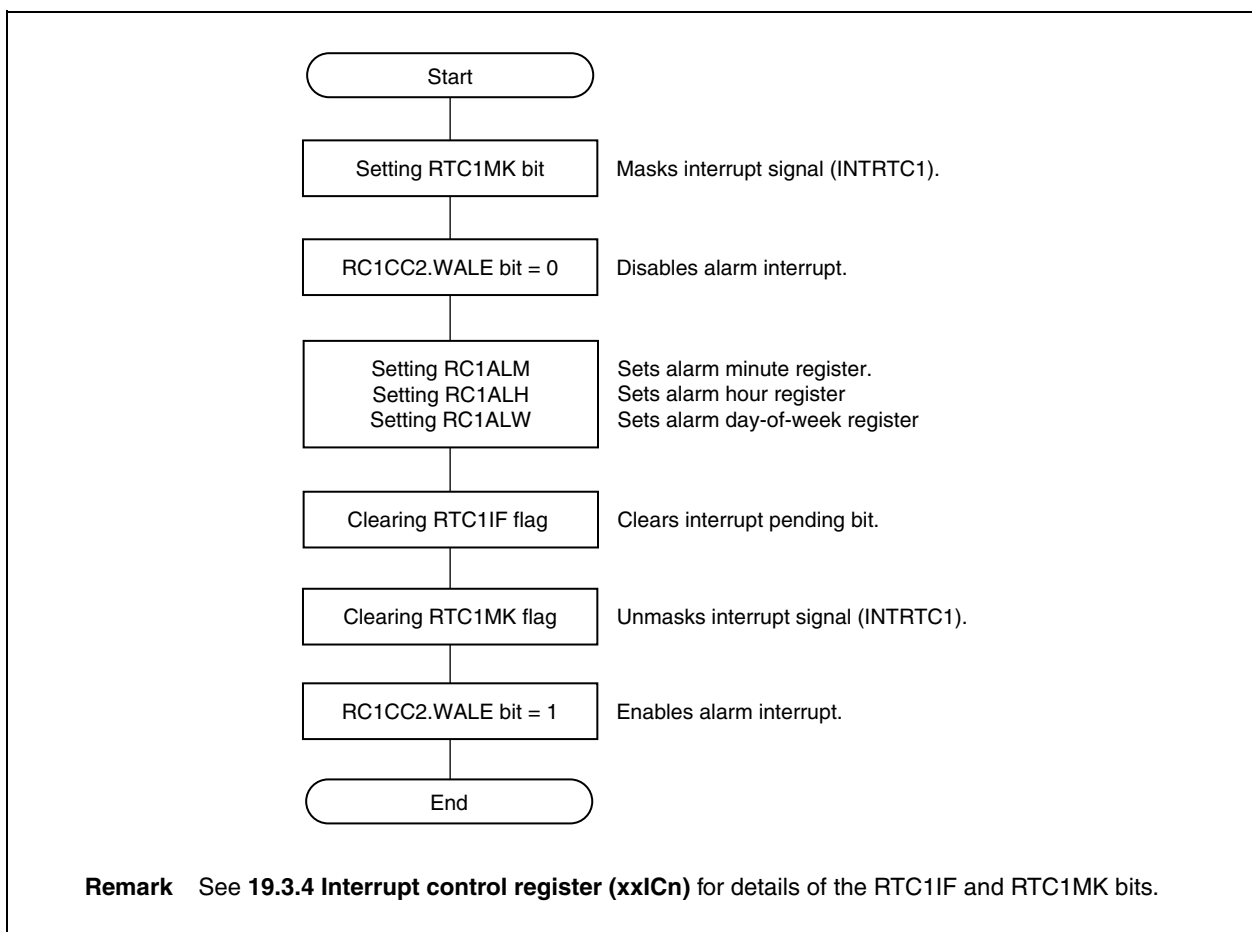
Figure 10-5. Changing INTRTC0 Interrupt Setting During Clock Operation



10.4.5 Changing INTRTC1 interrupt setting during real-time counter operation

If the setting of the INTRTC1 interrupt (alarm interrupt) signal is changed while the real-time counter operates (RC1PWR = 1, RTCE = 1), the INTRTC1 interrupt waveform may include whiskers and unintended signals may be output. Set as follows when changing the setting of the INTRTC1 interrupt signal during real-time counter operation (PC1PWR = 1, RTCE = 1), in order to mask the whiskers.

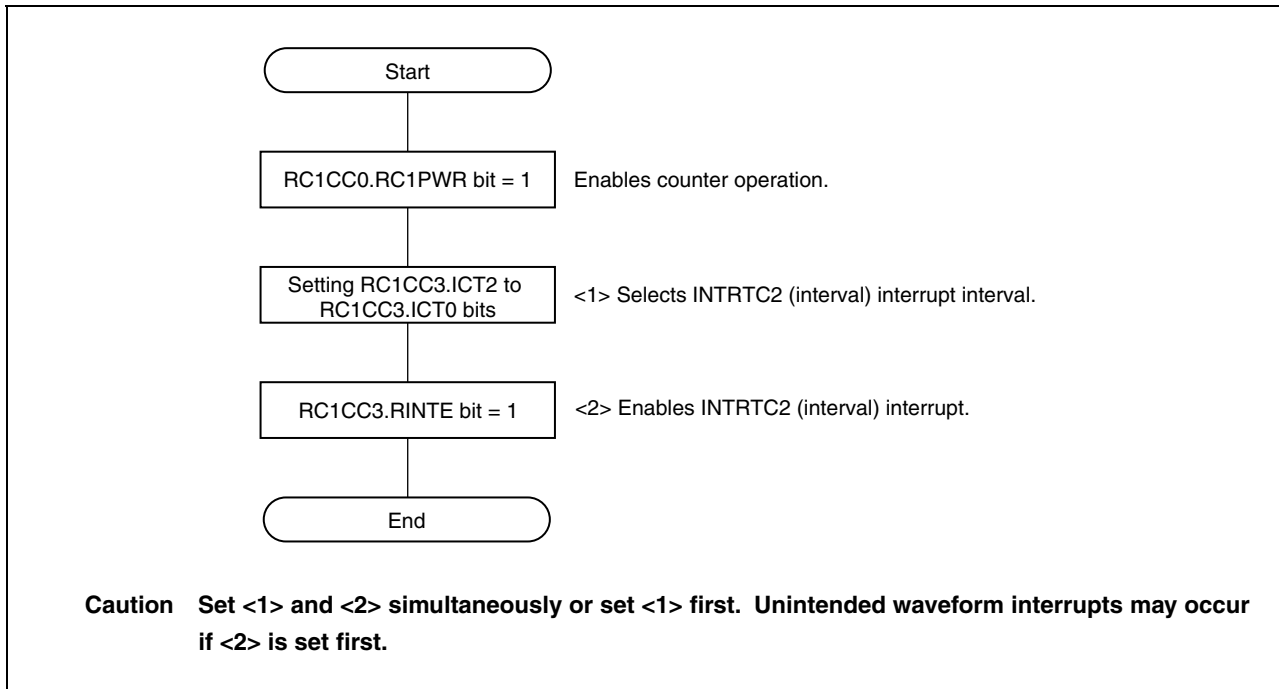
Figure 10-6. Changing INTRTC1 Interrupt Setting During Clock Operation



10.4.6 Initial INTRTC2 interrupt settings

Set as follows to set the INTRTC1 interrupt (interval interrupt).

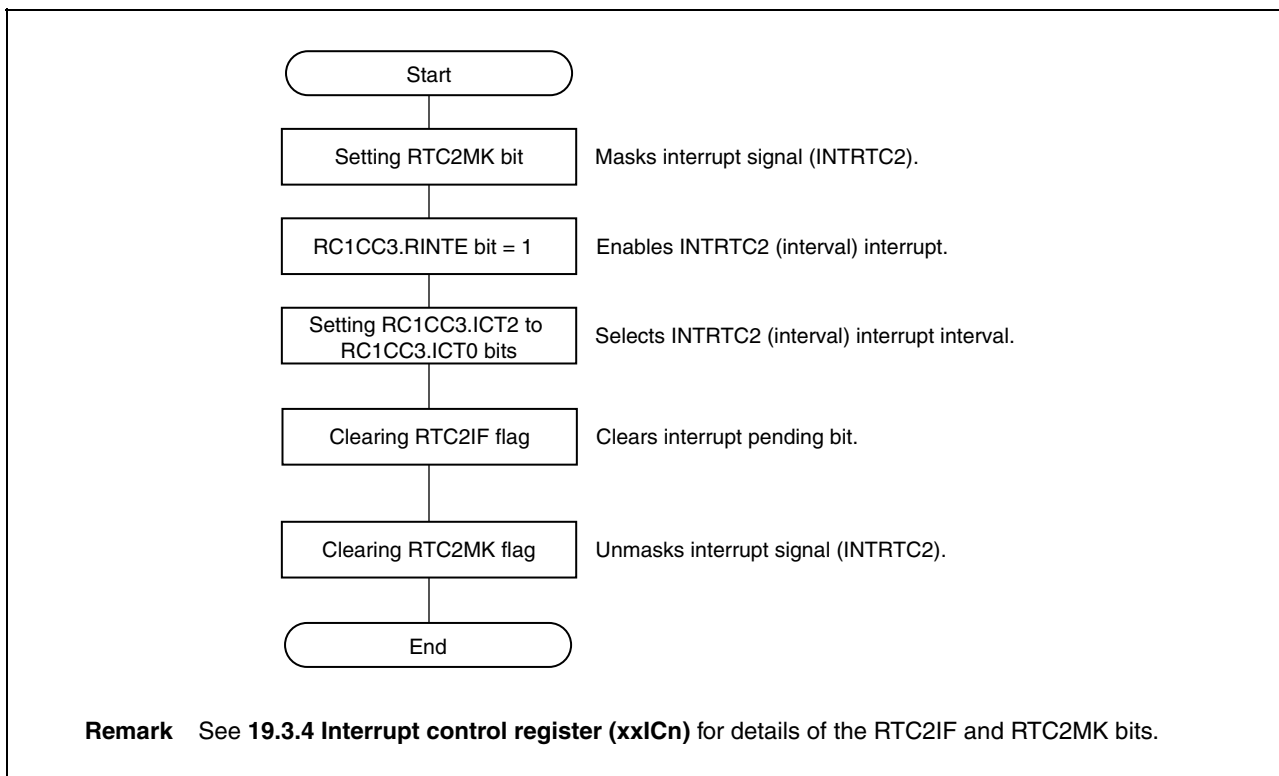
Figure 10-7. INTRTC2 Interrupt Setting



10.4.7 Changing INTRTC2 interrupt setting during real-time counter operation

If the setting of the INTRTC2 interrupt (interval interrupt) is changed while the real-time counter clock operates (PC1PWR = 1, RTCE = 1), the INTRTC2 interrupt waveform may include whiskers and unintended signals may be output. Set as follows when changing the setting of the INTRTC2 interrupt signal during real-time counter operation (PC1PWR = 1, RTCE = 1), in order to mask the whiskers.

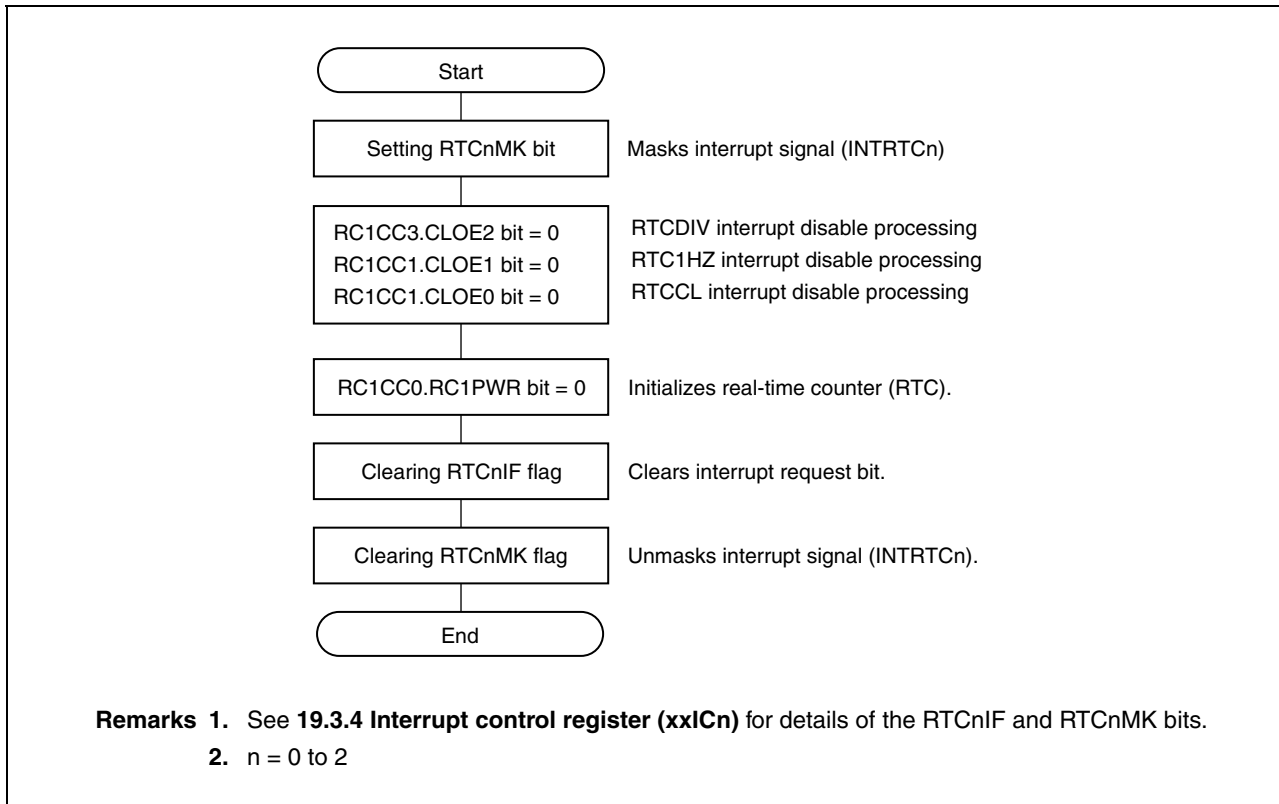
Figure 10-8. Changing INTRTC2 Interrupt Setting During Clock Operation



10.4.8 Initializing real-time counter

The procedure for initializing the real-time counter is shown below.

Figure 10-9. Initializing Real-Time Counter



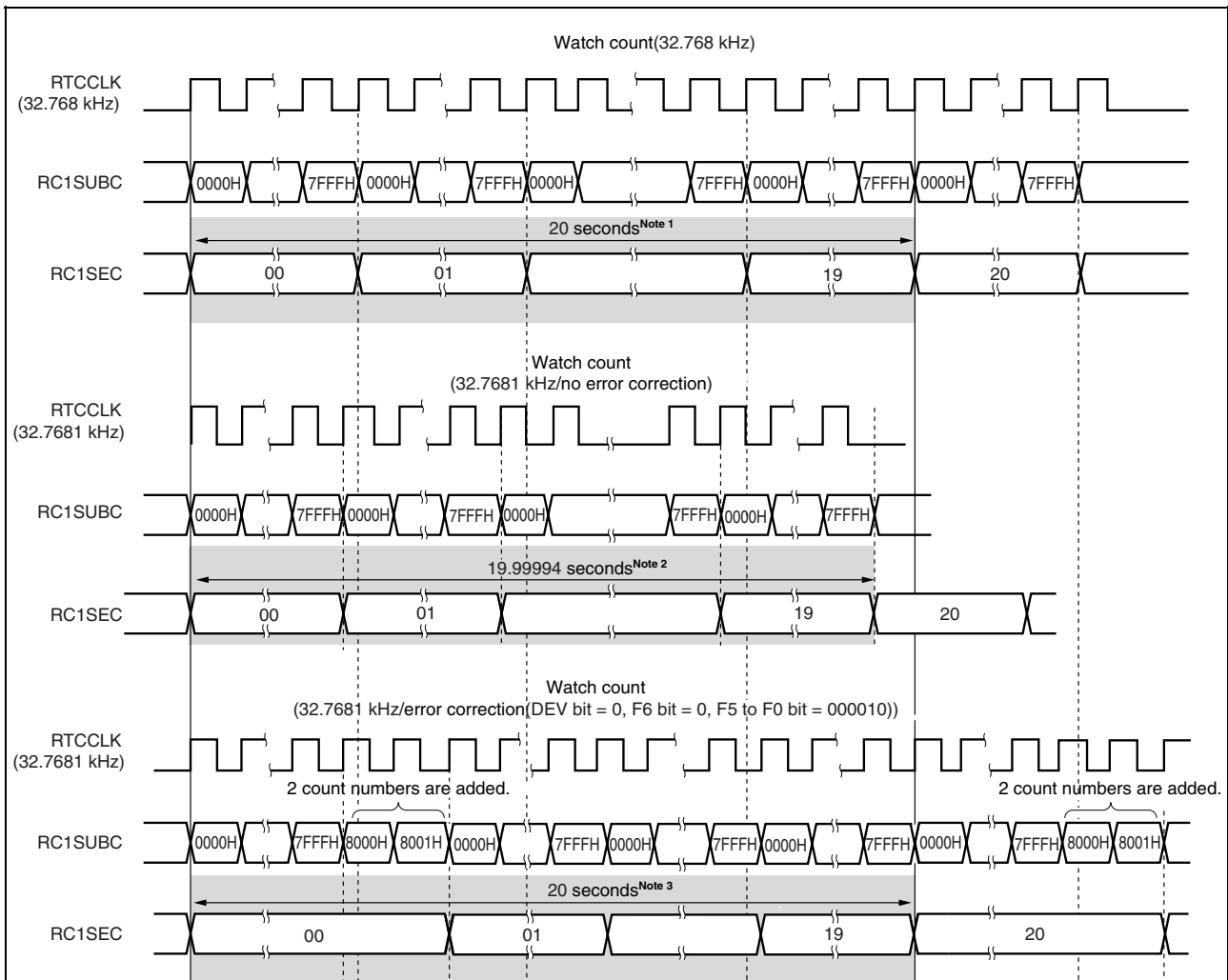
10.4.9 Watch error correction example of real-time counter

The watch error correction function corrects deviation in the oscillation frequency of a resonator connected to the V850ES/JC3-L and V850ES/JE3-L.

Deviation, here, refers to steady-state deviation, which is deviation in the frequency when the resonator is designed.

Next, the timing chart when an error has occurred in the input clock intended to be 32.768 kHz but a 32.7681 kHz resonator has been connected when designing the system, and the RC1SUBC and RC1SEC count operations to correct the error are shown below.

Figure 10-10. Watch Error Correction Example



Notes 1. The RC1SEC counter counts 20 seconds every 32,768 cycles (0000H to 7FFFH) of the 32.768 kHz clock.

2. When 32,768 cycles (0000H to 7FFFH) of the 32.7681 kHz clock are input, the time counted by the RC1SEC counter is calculated as follows: $32,768/3,268.1 \cong 0.999997$ seconds
If this counting continues 20 times, the time is calculated as follows: $(32,768/32,768.1) \times 20 \cong 19.99994$ seconds, which causes an error of 0.00006 seconds.

3. To precisely count 20 seconds by using a 32.7681 kHz clock, clear the DEV and F6 bits to 0 and set the F5 to F0 bits to 2H (000010B) in the RC1SUBU register. As a result, two additional cycles are counted every 20 seconds (when the RC1SEC counter count is 00, 20, and 40 seconds), so that the number of cycles counted at these points is not 32,768, but 32,770 (0000H to 8001H), which is exactly 20 seconds.

As shown in Figure 10-10, the watch can be accurately counted by incrementing the RC1SUBC count value, if a positive error faster than 32.768 kHz occurs at the resonator. Similarly, if a negative error slower than 32.768 kHz occurs at the resonator, the watch can be accurately counted by decrementing the RC1SUBC count value.

The RC1SUBC correction value is determined by using the RC1SUBU.F6 to RC1SUBU.F0 bits.

The F6 bit is used to determine whether to increment or decrement RC1SUBC and the F5 to F0 bits to determine the RC1SUBC value.

(1) Incrementing the RC1SUBC count value

The RC1SUBC count value is incremented by the value set using the F5 to F0 bits, by setting the F6 bit to 0.

Expression for calculating the increment value: $(F5 \text{ to } F0 \text{ bit value} - 1) \times 2$

[Example of incrementing the RC1SUBC count value: F6 bit = 0]

If 15H (010101B) is set to the F5 to F0 bits

$(15H - 1) \times 2 = 40$ (increments the RC1SUBC count value by 40)

RC1SUBC count value = $32,768 + 40 = 32,808$

(2) Decrementing the RC1SUBC count value

The RC1SUBC count value is decremented by an inverted value of the value set using the F5 to F0 bits, by setting the F6 bit to 1.

Expression for calculating the decrement value: $(\text{Inverted value of } F5 \text{ to } F0 \text{ bit value} + 1) \times 2$

[Example of decrementing the RC1SUBC count value: F6 bit = 1]

If 15H (010101B) is set to the F5 to F0 bits

Inverted data of 15H (010101B) = 2AH (101010B)

$(2AH + 1) \times 2 = 86$ (decrements the RC1SUBC count value by 86)

RC1SUBC count value = $32,768 - 86 = 32,682$

(3) DEV bit

The DEV bit determines when the setting by the F6 to F0 bits is enabled.

The value set by the F6 to F0 bits is reflected upon the next timing, but not to the RC1SUBC count value every time.

Table 10-6. DEV Bit Setting

DEV Bit Value	Timing of Reflecting Value to RC1SUBC
0	When RC1SEC is 00, 20, or 40 seconds.
1	When RC1SEC is 00 seconds.

[Example when 0010101B is set to F6 to F0 bits]

- If the DEV bit is 0
 - The RC1SUBC count value is 32,808 at 00, 20, or 40 seconds.
 - Otherwise, it is 32,768.
- IF DEV bit is 1
 - The RC1SUBC count value is 32,808 at 00 seconds.
 - Otherwise, it is 32,768.

As described above, the RC1SUBC count value is corrected every 20 seconds or 60 seconds, instead of every second, in order to match the RC1SUBC count value with the deviation width of the resonator.

The range in which the resonator frequency can be actually corrected is shown below.

- If the DEV bit is 0: 32.76180000 kHz to 32.77420000 kHz
- If the DEV bit is 1: 32.76593333 kHz to 32.77006667 kHz

The range in which the frequency can be corrected when the DEV bit is 0 is three times wider than when the DEV bit is 1.

However, the accuracy of setting the frequency when the DEV bit is 1 is three times that when the DEV bit is 0.

Tables 10-7 and 10-8 show the setting values of the DEV, and F6 to F0 bits, and the corresponding frequencies that can be corrected.

Table 10-7. Range of Frequencies That Can Be Corrected When DEV Bit = 0

F6	F5 to F0	RC1SUBC Correction Value	Frequency of Connected Clock (Including Steady-State Deviation)
0	000000	No correction	–
0	000001	No correction	–
0	000010	Increments RC1SUBC count value by 2 once every 20 seconds	32.76810000 kHz
0	000011	Increments RC1SUBC count value by 4 once every 20 seconds	32.76820000 kHz
0	000100	Increments RC1SUBC count value by 6 once every 20 seconds	32.76830000 kHz
		⋮	
0	111011	Increments RC1SUBC count value by 120 once every 20 seconds	32.77400000 kHz
0	111110	Increments RC1SUBC count value by 122 once every 20 seconds	32.77410000 kHz
0	111111	Increments RC1SUBC count value by 124 once every 20 seconds	32.77420000 kHz (upper limit)
1	000000	No correction	–
1	000001	No correction	–
1	000010	Decrements RC1SUBC count value by 124 once every 20 seconds	32.76180000 kHz (lower limit)
1	000011	Decrements RC1SUBC count value by 122 once every 20 seconds	32.76190000 kHz
1	000100	Decrements RC1SUBC count value by 120 once every 20 seconds	32.76200000 kHz
		⋮	
1	11011	Decrements RC1SUBC count value by 6 once every 20 seconds	32.76770000 kHz
1	11110	Decrements RC1SUBC count value by 4 once every 20 seconds	32.76780000 kHz
1	11111	Decrements RC1SUBC count value by 2 once every 20 seconds	32.76790000 kHz

Table 10-8. Range of Frequencies That Can Be Corrected When DEV Bit = 1

F6	F5 to F0	RC1SUBC Correction Value	Frequency of Connected Clock (Including Steady-State Deviation)
0	000000	No correction	–
0	000001	No correction	–
0	000010	Increments RC1SUBC count value by 2 once every 60 seconds	32.76803333 kHz
0	000011	Increments RC1SUBC count value by 4 once every 60 seconds	32.76806667 kHz
0	000100	Increments RC1SUBC count value by 6 once every 60 seconds	32.76810000 kHz
		⋮	
0	111011	Increments RC1SUBC count value by 120 once every 60 seconds	32.77000000 kHz
0	111110	Increments RC1SUBC count value by 122 once every 60 seconds	32.77003333 kHz
0	111111	Increments RC1SUBC count value by 124 once every 60 seconds	32.77006667 kHz (upper limit)
1	000000	No correction	–
1	000001	No correction	–
1	000010	Decrements RC1SUBC count value by 124 once every 60 seconds	32.76593333 kHz (lower limit)
1	000011	Decrements RC1SUBC count value by 122 once every 60 seconds	32.76596667 kHz
1	000100	Decrements RC1SUBC count value by 120 once every 60 seconds	32.76600000 kHz
		⋮	
1	11011	Decrements RC1SUBC count value by 6 once every 60 seconds	32.76790000 kHz
1	11110	Decrements RC1SUBC count value by 4 once every 60 seconds	32.76793333 kHz
1	11111	Decrements RC1SUBC count value by 2 once every 60 seconds	32.76796667 kHz

CHAPTER 11 WATCHDOG TIMER 2

11.1 Functions

Watchdog timer 2 is the default-start watchdog timer and starts up automatically immediately after a reset ends. Watchdog timer 2 starts up in reset mode and with the overflow time set to internal oscillator clock = $2^{19}/f_R$. When watchdog timer 2 overflows, it generates the WDT2RES signal to trigger a reset.

Watchdog timer 2 has the following features:

- It is the default-start watchdog timer^{Note 1}.
- It triggers the following operations when it overflows:
 - Reset mode: Watchdog timer 2 triggers a reset when it overflows (by generating the WDT2RES signal).
 - Non-maskable interrupt request mode: Watchdog timer 2 triggers NMI servicing when it overflows (by generating the INTWDT2 signal)^{Note 2}.
- Either the main clock, internal oscillator clock, or subclock can be input as the source clock.

Notes 1. Watchdog timer 2 automatically starts in the reset mode following reset release.

When not using watchdog timer 2, either stop it operating before it triggers a reset, or clear it once and stop it before the next overflow.

Also, write to the WDTM2 register for verification purposes only once, even if the default settings (reset mode, loop detection time interval: $2^{19}/f_R$) do not need to be changed.

- 2.** For details of the non-maskable interrupt servicing that occurs due to the generation of the non-maskable interrupt request signal (INTWDT2), see **19.2.2 (2) From INTWDT2 signal**.

11.2 Configuration

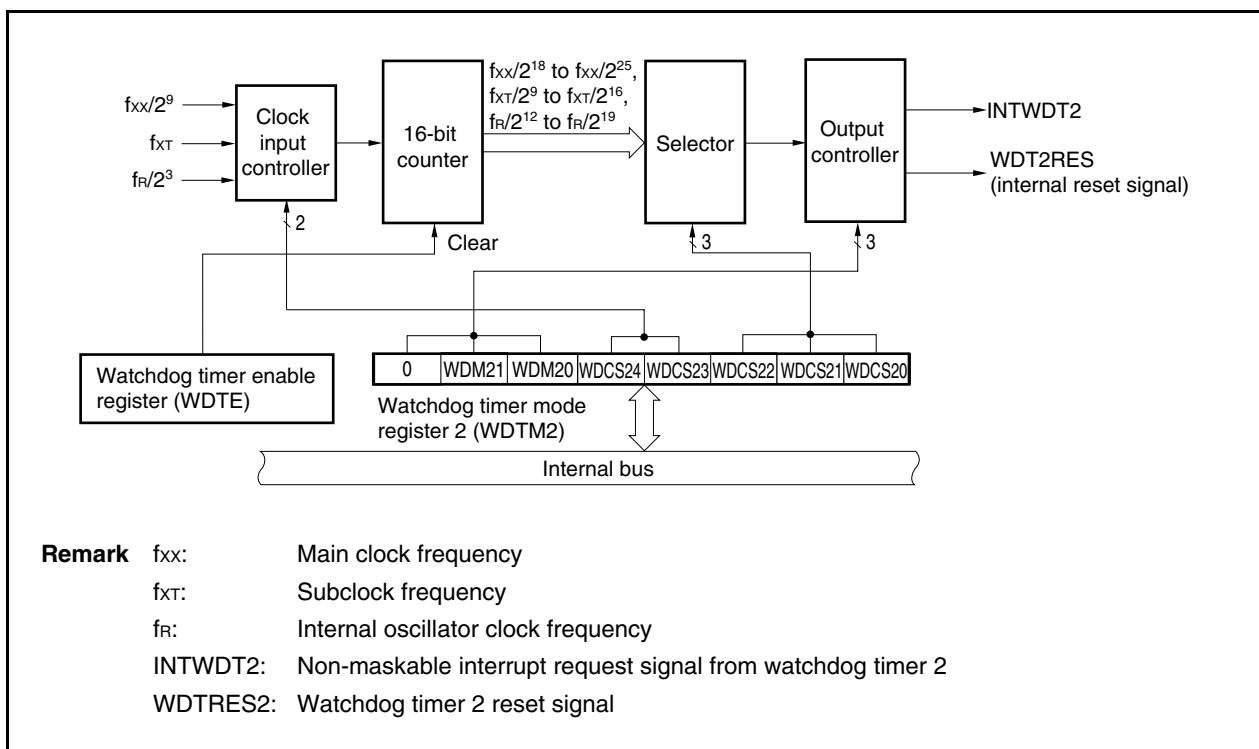
Watchdog timer 2 includes the following hardware.

Table 11-1. Configuration of Watchdog Timer 2

Item	Configuration
Control registers	Watchdog timer mode register 2 (WDTM2) Watchdog timer enable register (WDTE)

The following shows the block diagram of watchdog timer 2.

Figure 11-1. Block Diagram of Watchdog Timer 2



11.3 Registers

(1) Watchdog timer mode register 2 (WDTM2)

The WDTM2 register sets the overflow time and operation clock of watchdog timer 2.

This register can be read or written in 8-bit units. This register can be read any number of times, but it can be written only once following reset release.

Reset sets this register to 67H.

Caution Accessing the WDTM2 register is prohibited in the following statuses. Moreover, if the system is in the wait status, the only way to cancel the wait status is to execute a reset. For details, see 3.4.9 (1) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates on the subclock and main clock oscillation is stopped
- When the CPU operates on the internal oscillator clock

After reset: 67H R/W Address: FFFFF6D0H

	7	6	5	4	3	2	1	0
WDTM2	0	WDM21	WDM20	WDCS24	WDCS23	WDCS22	WDCS21	WDCS20

WDM21	WDM20	Selection of operation mode of watchdog timer 2
0	0	Stops operation
0	1	Non-maskable interrupt request mode (generation of INTWDT2 signal)
1	-	Reset mode (generation of WDT2RES signal)

Cautions

1. For details of the WDCS20 to WDCS24 bits, see Table 11-2 Loop Detection Time Interval of Watchdog Timer 2.
2. If the WDTM2 register is rewritten twice after a reset, an overflow signal is forcibly generated and the counter is reset.
3. To intentionally generate an overflow signal, write data to the WDTM2 register twice, or write a value other than “ACH” to the WDTE register once.
However, when watchdog timer 2 is set to “stop operation”, an overflow signal is not generated even if data is written to the WDTM2 register twice, or a value other than “ACH” is written to the WDTE register once.
4. To stop the operation of watchdog timer 2, set the RCM.RSTOP bit to 1 (to stop the internal oscillator) and write 00H in the WDTM2 register. If the RCM.RSTOP bit cannot be set to 1, set the WDCS23 bit to 1 (2ⁿ/f_{xx} is selected and the clock can be stopped in the IDLE1, IDLE2, sub-IDLE, and subclock operation modes).

Table 11-2. Loop Detection Time Interval of Watchdog Timer 2

	WDCS24	WDCS23	WDCS22	WDCS21	WDCS20	Selected Clock	100 kHz (MIN.)	220 kHz (TYP.)	400 kHz (MAX.)
Internal oscillator clock	0	0	0	0	0	$2^{12}/f_R$	41.0 ms	18.6 ms	10.2 ms
	0	0	0	0	1	$2^{13}/f_R$	81.9 ms	37.2 ms	20.5 ms
	0	0	0	1	0	$2^{14}/f_R$	163.8 ms	74.5 ms	41.0 ms
	0	0	0	1	1	$2^{15}/f_R$	327.7 ms	148.9 ms	81.9 ms
	0	0	1	0	0	$2^{16}/f_R$	655.4 ms	297.9 ms	163.8 ms
	0	0	1	0	1	$2^{17}/f_R$	1310.7 ms	595.8 ms	327.7 ms
	0	0	1	1	0	$2^{18}/f_R$	2621.4 ms	1191.6 ms	655.4 ms
	0	0	1	1	1	$2^{19}/f_R$	5242.9 ms	2383.1 ms	1310.7 ms
Main clock							$f_{XX} = 20 \text{ MHz}$	$f_{XX} = 16 \text{ MHz}$	$f_{XX} = 10 \text{ MHz}$
	0	1	0	0	0	$2^{18}/f_{XX}$	13.1 ms	16.4 ms	26.2 ms
	0	1	0	0	1	$2^{19}/f_{XX}$	26.2 ms	32.8 ms	52.4 ms
	0	1	0	1	0	$2^{20}/f_{XX}$	52.4 ms	65.5 ms	104.9 ms
	0	1	0	1	1	$2^{21}/f_{XX}$	104.9 ms	131.1 ms	209.7 ms
	0	1	1	0	0	$2^{22}/f_{XX}$	209.7 ms	262.1 ms	419.4 ms
	0	1	1	0	1	$2^{23}/f_{XX}$	419.4 ms	524.3 ms	838.9 ms
	0	1	1	1	0	$2^{24}/f_{XX}$	838.9 ms	1048.6 ms	1677.7 ms
Subclock							$f_{XT} = 32.768 \text{ kHz}$		
	1	×	0	0	0	$2^9/f_{XT}$	15.625 ms		
	1	×	0	0	1	$2^{10}/f_{XT}$	31.25 ms		
	1	×	0	1	0	$2^{11}/f_{XT}$	62.5 ms		
	1	×	0	1	1	$2^{12}/f_{XT}$	125 ms		
	1	×	1	0	0	$2^{13}/f_{XT}$	250 ms		
	1	×	1	0	1	$2^{14}/f_{XT}$	500 ms		
	1	×	1	1	0	$2^{15}/f_{XT}$	1000 ms		
1	×	1	1	1	$2^{16}/f_{XT}$	2000 ms			

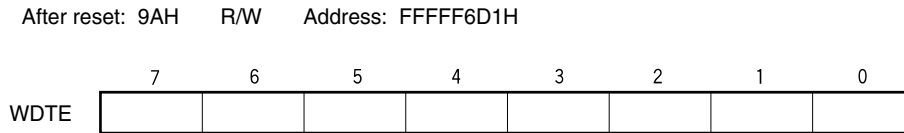
Remark × = Either 0 or 1

(2) Watchdog timer enable register (WDTE)

The counter of watchdog timer 2 is cleared and counting is restarted by writing “ACH” to the WDTE register.

The WDTE register can be read or written in 8-bit units. (When a 1-bit memory manipulation instruction is executed on the WDTE register, an overflow signal is forcibly generated.)

Reset sets this register to 9AH.



- Cautions**
1. When a value other than “ACH” is written to the WDTE register, an overflow signal is forcibly output.
 2. To intentionally generate an overflow signal, write a value other than “ACH” to the WDTE register once, or write data to the WDTM2 register twice.
However, when watchdog timer 2 is set to “stop operation”, an overflow signal is not generated even if a value other than “ACH” is written to the WDTE register once, or data is written to the WDTM2 register twice.
 3. The read value of the WDTE register is “9AH” (which differs from the written value “ACH”).

11.4 Operation

Watchdog timer 2 automatically starts in the reset mode immediately after a reset.

The WDTM2 register can be written to only once immediately after a reset using byte access. To use watchdog timer 2, write the operation mode setting and the loop detection time interval to the WDTM2 register using an 8-bit memory manipulation instruction. After this, the operation of watchdog timer 2 cannot be stopped.

The WDCS24 to WDCS20 bits of the WDTM2 register are used to select the watchdog timer 2 loop detection time interval.

Writing ACH to the WDTE register clears the counter of watchdog timer 2 and starts the count operation again. After the count operation has started, write ACH to WDTE within the loop detection time interval.

If the loop detection time interval expires without ACH being written to the WDTE register, a reset signal (WDT2RES) or a non-maskable interrupt request signal (INTWDT2) is generated, depending on the values of the WDTM2.WDM21 and WDTM2.WDM20 bits.

When the WDTM2.WDM21 bit is set to 1 (reset mode), if watchdog timer 2 overflows during oscillation stabilization immediately after a reset ends or after a standby is released, no internal reset will occur and the CPU clock will switch to the internal oscillator clock.

To not use watchdog timer 2, write 00H to the WDTM2 register.

For details of the non-maskable interrupt servicing that occurs when the non-maskable interrupt request mode is set, see **19.2.2 (2) From INTWDT2 signal**.

CHAPTER 12 REAL-TIME OUTPUT FUNCTION (RTO)

12.1 Function

The real-time output function transfers preset data to the real-time output buffer registers (RTBL0 and RTBH0), and then transfers this data by hardware to an external device via the output latches, upon occurrence of a timer interrupt. The pins through which the data is output to an external device constitute a port called the real-time output function (RTO).

Because RTO can output signals without jitter, it is suitable for controlling a stepper motor.

In the V850ES/JC3-L and V850ES/JE3-L, one 6-bit real-time output port channel is provided.

The real-time output port can be set to the port mode or real-time output port mode in 1-bit units.

12.2 Configuration

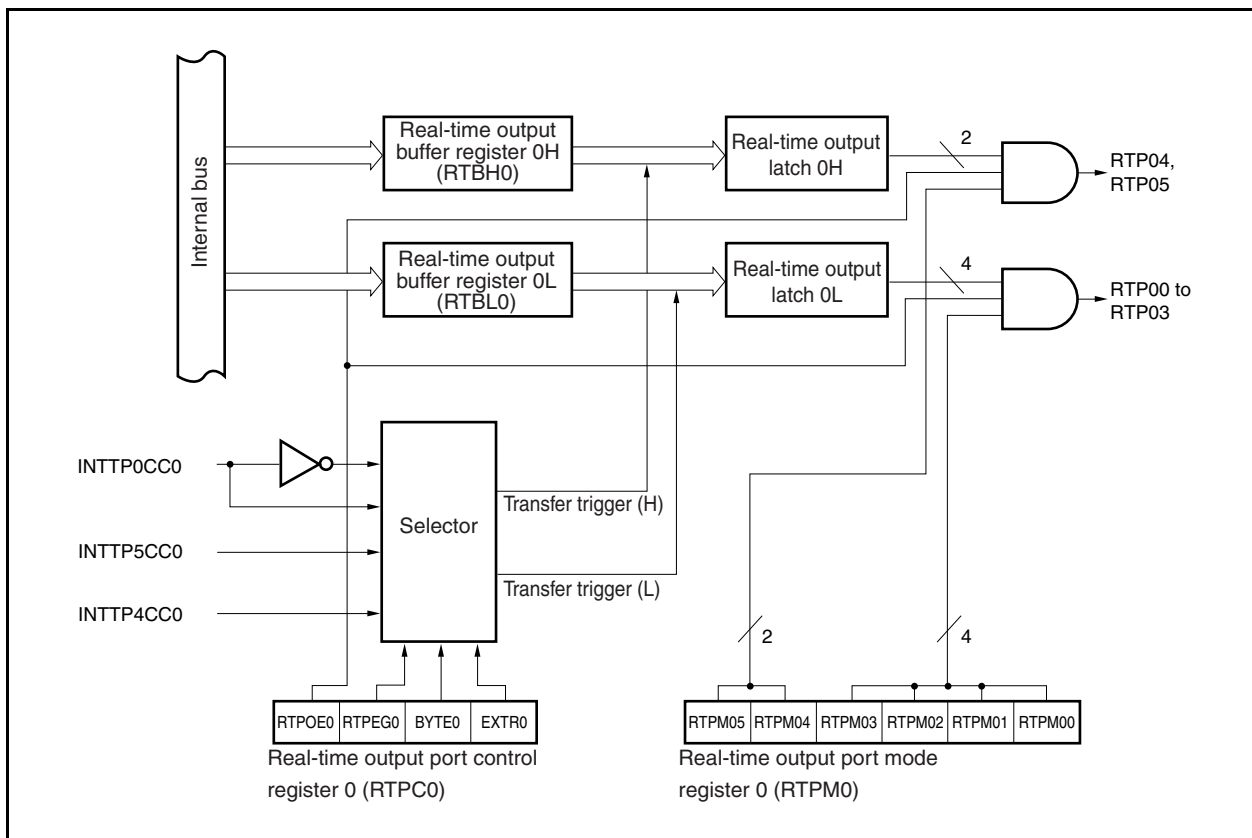
RTO includes the following hardware.

Table 12-1. Configuration of RTO

Item	Configuration
Registers	Real-time output buffer registers 0L, 0H (RTBL0, RTBH0) Real-time output latches 0H, 0L Real-time output port mode register 0 (RTPM0) Real-time output port control register 0 (RTPC0)

The block diagram of RTO is shown below.

Figure 12-1. Block Diagram of RTO



(1) Real-time output buffer registers 0L, 0H (RTBL0, RTBH0)

The RTBL0 and RTBH0 registers are 4-bit registers that hold preset output data.

These registers are mapped to different addresses in the peripheral I/O register area.

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

Reset sets these registers to 00H.

If an operation mode of 4 bits × 1 channel or 2 bits × 1 channel is specified (RTPC0.BYTE0 bit = 0), data can be individually set to the RTBL0 and RTBH0 registers. The data of both these registers can be read at once by specifying the address of either of these registers.

If an operation mode of 6 bits × 1 channel is specified (BYTE0 bit = 1), 8-bit data can be set to both the RTBL0 and RTBH0 registers by writing the data to either of these registers. Moreover, the data of both these registers can be read at once by specifying the address of either of these registers.

Table 12-2 shows the operation when the RTBL0 and RTBH0 registers are manipulated.

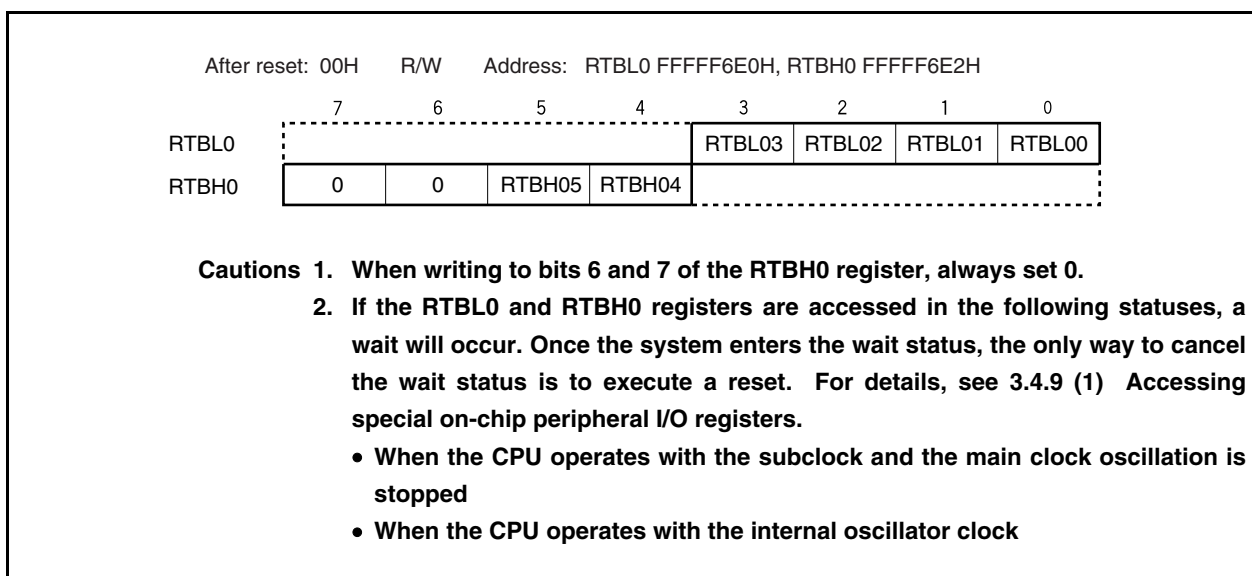


Table 12-2. Operation During Manipulation of RTBL0 and RTBH0 Registers

Operation Mode	Register to Be Manipulated	Read		Write ^{Note}	
		Higher 4 Bits	Lower 4 Bits	Higher 4 Bits	Lower 4 Bits
4 bits × 1 channel, 2 bits × 1 channel	RTBL0	RTBH0	RTBL0	Invalid	RTBL0
	RTBH0	RTBH0	RTBL0	RTBH0	Invalid
6 bits × 1 channel	RTBL0	RTBH0	RTBL0	RTBH0	RTBL0
	RTBH0	RTBH0	RTBL0	RTBH0	RTBL0

Note After setting the real-time output port, output data must be set to the RTBL0 and RTBH0 registers before a real-time output trigger is generated.

12.3 Registers

RTO is controlled using the following two registers.

- Real-time output port mode register 0 (RTPM0)
- Real-time output port control register 0 (RTPC0)

Caution In order to use the real-time output pins (RTP00 to RTP05), set these pins as real-time output port pins using the PMC and PFC registers.

(1) Real-time output port mode register 0 (RTPM0)

The RTPM0 register enables the selection of real-time output port mode or port mode in 1-bit units.

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

Reset sets this register to 00H.

After reset: 00H		R/W	Address: FFFFF6E4H					
	7	6	5	4	3	2	1	0
RTPM0	0	0	RTPM05	RTPM04	RTPM03	RTPM02	RTPM01	RTPM00
	RTPM0m	Control of real-time output port (m = 0 to 5)						
	0	Real-time output disabled						
	1	Real-time output enabled						

Cautions

1. By enabling the real-time output operation (RTPC0.RTPOE0 bit = 1), the bits enabled to real-time output among the RTP00 to RTP05 signals perform real-time output, and the bits set to port mode output 0.
2. If real-time output is disabled (RTPOE0 bit = 0), the real-time output pins (RTP00 to RTP05) all output 0, regardless of the RTPM0 register setting.
3. When writing to bits 6 and 7 of the RTPM0 register, always set 0.

(2) Real-time output port control register 0 (RTPC0)

The RTPC0 register is a register that sets the operation mode and output trigger of the real-time output port.

The relationship between the operation mode and output trigger of the real-time output port is as shown in Table 12-3.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF6E5H								
	<7>	6	5	4	3	2	1	0
RTPC0	RTPOE0	RTPEG0	BYTE0	EXTR0	0	0	0	0
	RTPOE0	Control of real-time output operation						
	0	Disables operation ^{Note 1}						
	1	Enables operation						
	RTPEG0	Valid edge of INTTP0CC0 signal						
	0	Falling edge ^{Note 2}						
	1	Rising edge						
	BYTE0	Specification of channel configuration for real-time output						
	0	4 bits × 1 channel, 2 bits × 1 channel						
	1	6 bits × 1 channel						

Notes

1. When real-time output operation is disabled (RTPOE0 bit = 0), all the bits of the real-time output pins (RTP00 to RTP05) output "0".
2. With this setting, the transfer of data between the buffer and the latch will be delayed by one clock cycle.

Caution Set the RTPEG0, BYTE0, and EXTR0 bits only when RTPOE0 bit = 0.

Table 12-3. Operation Modes and Output Triggers of Real-Time Output Port

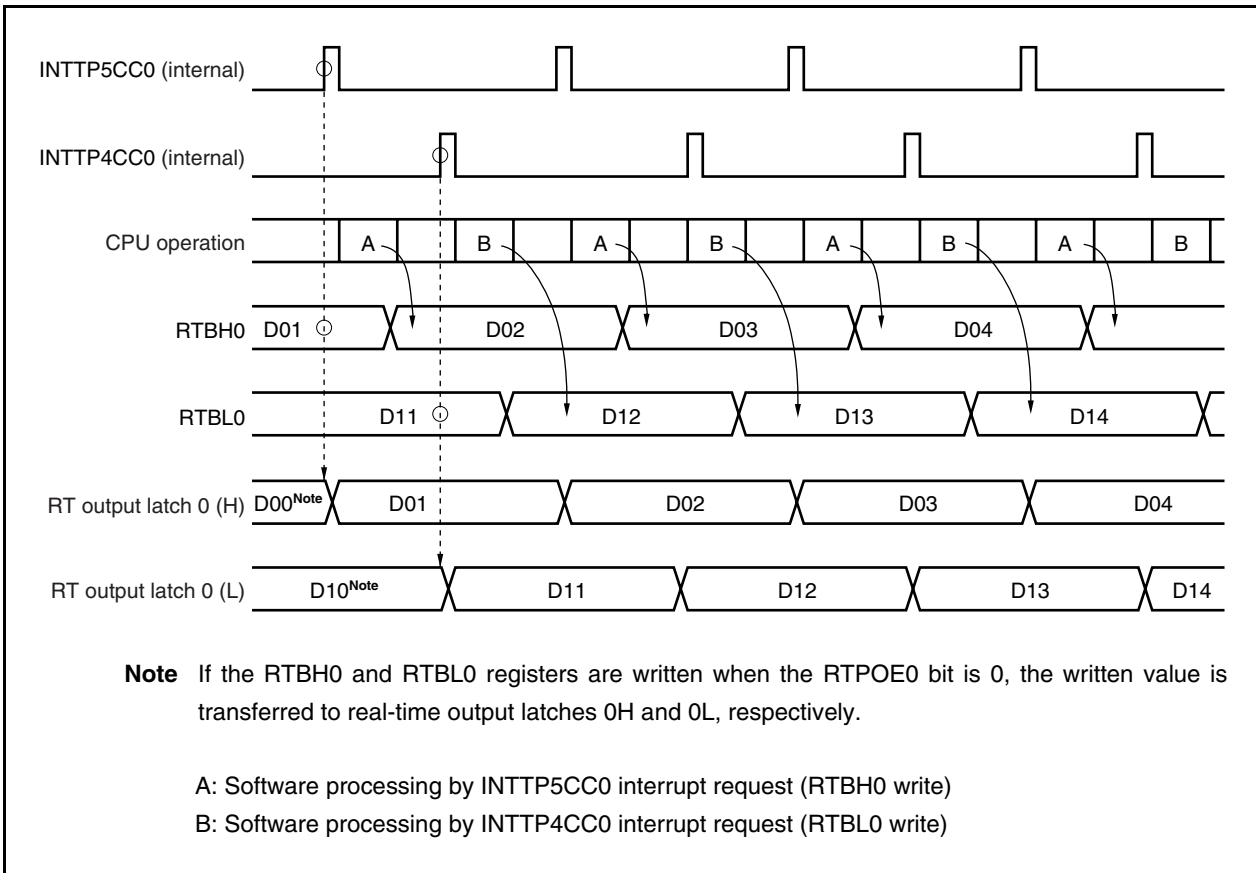
BYTE0	EXTR0	Operation Mode	RTBH0 (RTP04, RTP05)	RTBL0 (RTP00 to RTP03)
0	0	4 bits × 1 channel,	INTTP5CC0	INTTP4CC0
	1	2 bits × 1 channel	INTTP4CC0	INTTP0CC0
1	0	6 bits × 1 channel	INTTP4CC0	
	1		INTTP0CC0	

12.4 Operation

If the real-time output operation is enabled by setting the RTPC0.RTPOE0 bit to 1, the data of the RTBH0 and RTBL0 registers is transferred to the real-time output latch in synchronization with the generation of the selected transfer trigger (set by the RTPC0.EXTR0 and RTPC0.BYTE0 bits). Of the transferred data, only the data of the bits for which real-time output is enabled by the RTPM0 register is output from the RTP00 to RTP05 bits. The bits for which real-time output is disabled by the RTPM0 register output 0.

If the real-time output operation is disabled by clearing the RTPOE0 bit to 0, the RTP00 to RTP05 pins output 0 regardless of the setting of the RTPM0 register.

Figure 12-2. Example of Operation Timing of RTO0 (When EXTR0 Bit = 0, BYTE0 Bit = 0)



Remark For the operation during standby, see **CHAPTER 23 STANDBY FUNCTION**.

12.5 Usage

- (1) Disable real-time output.
Clear the RTPC0.RTPOE0 bit to 0.
- (2) Perform initialization as follows.
 - Set the alternate-function pins of port 5
Set the PFC5.PFC5m bit and PFCE5.PFCE5m bit to 1, and then set the PMC5.PMC5m bit to 1 (m = 0 to 5).
 - Specify the real-time output port mode or port mode in 1-bit units.
Set the RTPM0 register.
 - Channel configuration: Select the trigger and valid edge.
Set the RTPC0.EXTR0, RTPC0.BYTE0, and RTPC0.RTPEG0 bits.
 - Set the initial values to the RTBH0 and RTBL0 registers^{Note 1}.
- (3) Enable real-time output.
Set the RTPOE0 bit = 1.
- (4) Set the next output value to the RTBH0 and RTBL0 registers by the time the selected transfer trigger is generated^{Note 2}.
- (5) Set the next real-time output value to the RTBH0 and RTBL0 registers via interrupt servicing corresponding to the selected trigger.

- Notes**
1. If the RTBH0 and RTBL0 registers are written when the RTPOE0 bit is 0, the written value is transferred to real-time output latches 0H and 0L, respectively.
 2. Even if the RTBH0 and RTBL0 registers are written when the RTPOE0 bit = 1, data is not transferred to real-time output latches 0H and 0L.

Caution To apply the above settings to the real-time output pins (RTP00 to RTP05), set the real-time output pins by using the PMC5 and PFC5 registers.

12.6 Cautions

- (1) Prevent the following conflicts by using software, such as by writing to the RTBL0, RTBH0, and RTPC0 registers inside the interrupt servicing routine of the selected real-time output trigger.
 - Conflict between real-time output disable/enable switching (RTPOE0 bit) and selected real-time output trigger.
 - Conflict between writing to the RTBH0 and RTBL0 registers in the real-time output enabled status and the selected real-time output trigger.
- (2) Before performing initialization, disable real-time output (RTPOE0 bit = 0).
- (3) Once real-time output has been disabled (RTPOE0 bit = 0), be sure to initialize the RTBH0 and RTBL0 registers before enabling real-time output again (RTPOE0 bit = 0 → 1).

CHAPTER 13 A/D CONVERTER

13.1 Overview

The A/D converter of the V850ES/JC3-L and V850ES/JE3-L have a resolution of 10 bits and converts an input analog signal into a digital value.

The number of A/D converter in the V850ES/JC3-L and V850ES/JE3-L are shown below.

Product Name	V850ES/JC3-L (40-pin)	V850ES/JC3-L (48-pin)	V850ES/JE3-L
Number of ANI channels (m)	5 channels (m = 5)	6 channels (m = 6)	10 channels (m = 10)
Number of mounted (n)	ANI0 to ANI4 (n = 0 to 4)	ANI0 to ANI5 (n = 0 to 5)	ANI0 to ANI9 (n = 0 to 9)

In this chapter, the number of ANI channels is expressed as m and the number of ANI pins (analog input function pins) is expressed as n.

The A/D converter has the following features.

- 10-bit resolution
- Successive approximation method
- Operating voltage: $AV_{REF0} = 2.7$ to 3.6 V
- Analog input voltage: 0 V to AV_{REF0}
- The following functions are provided as operation modes.
 - Continuous select mode
 - Continuous scan mode
 - One-shot select mode
 - One-shot scan mode
- The following functions are provided as trigger modes.
 - Software trigger mode
 - External trigger mode (external, 1)
 - Timer trigger mode
- Conversion time
 - 2.6 to 24 μ s @ 3.0 V $\leq AV_{REF0} \leq 3.6$ V
 - 3.9 to 24 μ s @ 2.7 V $\leq AV_{REF0} < 3.0$ V
- Power-fail monitor function (conversion result compare function)

13.2 Functions

(1) 10-bit resolution A/D conversion

A/D conversion is repeated at a resolution of 10 bits for an analog signal that is input to a channel selected from ANIn. Each time A/D conversion is completed, an interrupt request signal (INTAD) is generated.

(2) Power-fail detection

This function is used to detect a drop in the battery voltage. The result of A/D conversion (the value of the ADA0CRnH register) is compared with the value of the ADA0PFT register, and the INTAD signal is generated only when the comparison condition specified by the ADA0PFM register is satisfied.

13.3 Configuration

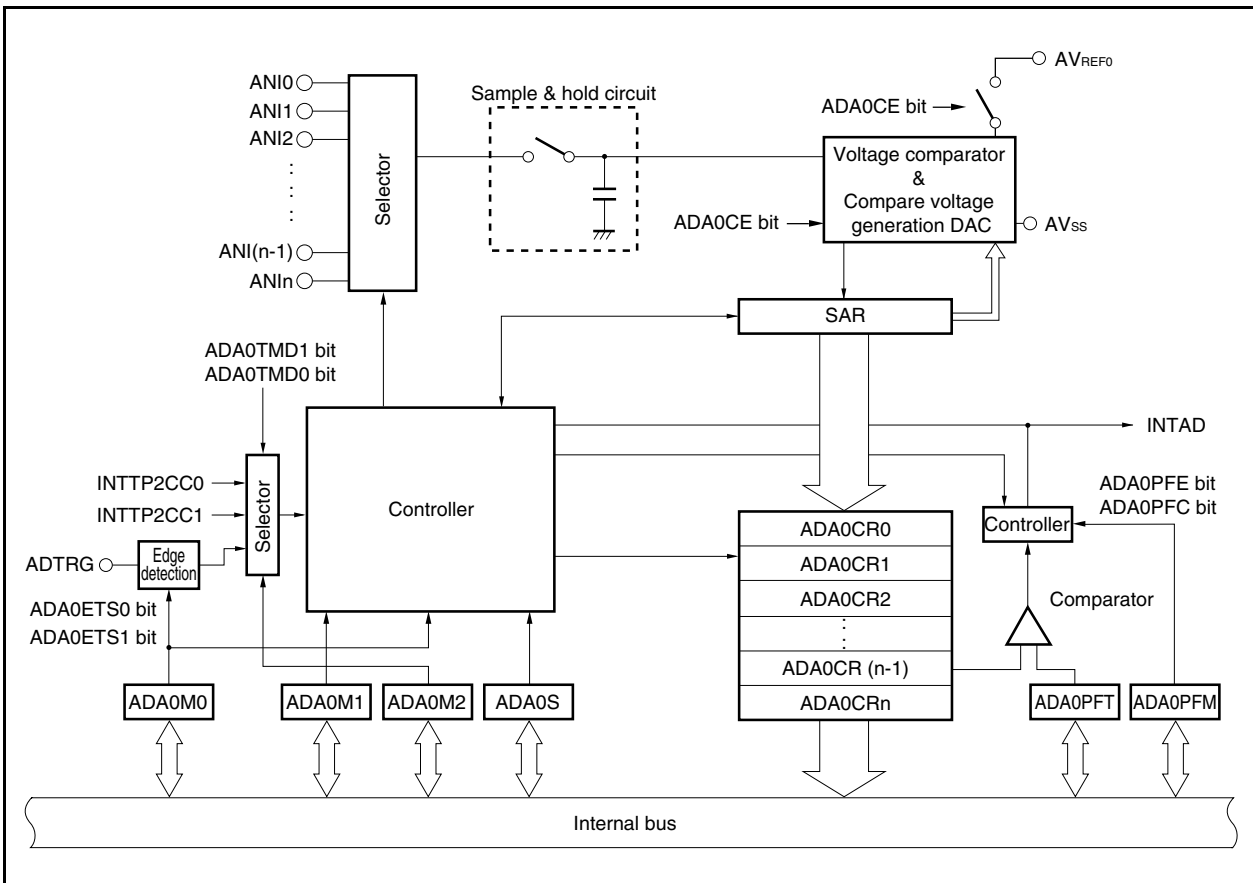
The A/D converter includes the following hardware.

Table 13-1. Configuration of A/D Converter

Item	Configuration
Analog inputs	m channels (ANIn pins)
Registers	Successive approximation register (SAR) A/D conversion result registers n (ADA0CRn) A/D conversion result registers nH (ADCRnH): Only higher 8 bits can be read A/D converter mode registers 0 to 2 (ADA0M0 to ADA0M2) A/D converter channel specification register 0 (ADA0S) Power fail compare mode register (ADA0PFM) Power fail compare threshold value register (ADA0PFT)

The block diagram of the A/D converter is shown below.

Figure 13-1. Block Diagram of A/D Converter



(1) ANIn pins

These are analog input pins for the m A/D converter channels and are used to input analog signals to be converted into digital signals. Pins other than the ones selected as analog input pins by the ADA0S register can be used as I/O port pins.

Caution Make sure that the voltages input to the ANIn pins do not exceed the rated values. In particular if a voltage of AV_{REF0} or higher is input to a channel, the conversion value of that channel becomes undefined, and the conversion values of the other channels may also be affected.

(2) Sample & hold circuit

The sample & hold circuit samples each of the analog input signals selected by the input circuit and sends the sampled data to the voltage comparator. This circuit also holds the sampled analog input signal voltage during A/D conversion.

(3) Compare voltage generation DAC

The compare voltage generation DAC is connected between AV_{REF0} and AV_{SS} and generates the voltage to be compared with the value that was sampled and held by the sample & hold circuit.

(4) Voltage comparator

The voltage comparator compares the voltage value that was sampled and held with the output voltage of the compare voltage generation DAC.

(5) Successive approximation register (SAR)

This register compares the voltage of the sampled analog input signal with the output voltage of the compare voltage generation DAC (compare voltage), and sequentially retains the comparison result bit by bit starting from the most significant bit (MSB).

When the comparison result has been held down to the least significant bit (LSB) (that is, when A/D conversion is complete), the contents of the SAR register are transferred to the ADA0CRn register.

(6) 10-bit AD conversion result register n (ADA0CRn)

The ADA0CRn register is a 16-bit register that stores the A/D conversion result. ADA0ARn consist of m registers and the A/D conversion result is stored in the 10 higher bits of the ADA0CRn register corresponding to analog input. (The lower 6 bits are fixed to 0.)

(7) A/D conversion result register nH (ADA0CRnH)

This is a 8-bit register that stores the A/D conversion result. ADA0CRnH consists of m registers and the A/D conversion result is stored in the higher 8 bits of the ADA0CRnH register corresponding to the analog input signal.

(8) A/D converter mode register 0 (ADA0M0)

This register specifies the operation mode and controls conversion by the A/D converter.

(9) A/D converter mode register 1 (ADA0M1)

This register specifies the time required to convert an analog input signal to a digital signal.

(10) A/D converter mode register 2 (ADA0M2)

This register specifies the hardware trigger mode.

(11) A/D converter channel specification register (ADA0S)

This register specifies the pin to which the analog voltage to be converted is input.

(12) Power-fail compare mode register (ADA0PFM)

This register controls power-fail monitoring.

(13) Power-fail compare threshold value register (ADA0PFT)

The ADA0PFT register sets the threshold value that is compared with the value of A/D conversion result register nH (ADA0CRnH). The 8-bit data set to the ADA0PFT register is compared with the higher 8 bits of the A/D conversion result register (ADA0CRnH).

(14) Controller

The controller compares the result of A/D conversion (the value of the ADA0CRnH register) with the value of the ADA0PFT register when A/D conversion is completed or when the power-fail detection function is used, and generates the INTAD signal only when the specified comparison condition is satisfied.

(15) AV_{REF0} pin

This is the pin used to input the reference voltage of the A/D converter. Always make the potential of this pin the same as that of the V_{DD} pin even when the A/D converter is not used. The signals input to the ANIn pins are converted to digital signals based on the voltage applied between the AV_{REF0} and AV_{SS} pins.

(16) AV_{SS} pin

This is the ground pin of the A/D converter. Always make the potential of this pin the same as that of the V_{SS} pin even when the A/D converter is not used.

13.4 Registers

The A/D converter is controlled by the following registers.

- A/D converter mode registers 0, 1, 2 (ADA0M0, ADA0M1, ADA0M2)
- A/D converter channel specification register 0 (ADA0S)
- Power-fail compare mode register (ADA0PFM)

The following registers are also used.

- A/D conversion result register n (ADA0CRn)
- A/D conversion result register nH (ADA0CRnH)
- Power-fail compare threshold value register (ADA0PFT)

(1) A/D converter mode register 0 (ADA0M0)

The ADA0M0 register is an 8-bit register that specifies the operation mode and controls conversion.

This register can be read or written in 8-bit or 1-bit units. However, the ADA0EF bit is read-only.

Reset sets this register to 00H.

Caution Accessing the ADA0M0 register is prohibited in the following statuses. If a wait cycle is generated, it can be cleared only by a reset. For details, see 4.4.9 (1) Accessing special on-chip peripheral I/O registers.

- When the CPU operates on the subclock and main clock oscillation is stopped
- When the CPU operates on the internal oscillator clock

(1/2)

After reset: 00H		R/W	Address: FFFFF200H							
			<7>	6	5	4	3	2	1	<0>
ADA0M0	ADA0CE	0	ADA0MD1	ADA0MD0	ADA0ETS1	ADA0ETS0	ADA0TMD	ADA0EF		
ADA0CE		A/D conversion control								
0		Stops A/D conversion								
1		Enables A/D conversion								
ADA0MD1	ADA0MD0	Specification of A/D converter operation mode								
0	0	Continuous select mode								
0	1	Continuous scan mode								
1	0	One-shot select mode								
1	1	One-shot scan mode								

(2/2)

ADA0ETS1	ADA0ETS0	Specification of external trigger (ADTRG pin) input valid edge
0	0	No edge detection
0	1	Falling edge detection
1	0	Rising edge detection
1	1	Detection of both rising and falling edges

ADA0TMD	Trigger mode selection
0	Software trigger mode
1	External trigger mode/timer trigger mode

ADA0EF	A/D converter status display
0	A/D conversion stopped
1	A/D conversion in progress

- Cautions**
1. A write operation to bit 0 is ignored.
 2. Changing the ADA0M1.ADA0FR2 to ADA0M1.ADA0FR0 bits is prohibited while A/D conversion is enabled (ADA0CE bit = 1).
 3. In the following modes, write data to the ADA0M0, ADA0M2, ADA0S, ADA0PFM, or ADA0PFT registers while A/D conversion is stopped (ADA0CE bit = 0), and then enable A/D conversion (ADA0CE bit = 1).
 - Normal conversion mode
 - One-shot select mode/one-shot scan mode in high-speed conversion mode

If the ADA0M0, ADA0M2, ADA0S, ADA0PFM, and ADA0PFT registers are written in other modes during A/D conversion (ADA0EF bit = 1), the following will be performed, according to the mode.

 - In software trigger mode
A/D conversion is stopped and started again from the beginning.
 - In hardware trigger mode
A/D conversion is stopped, and the trigger standby status is set.
 4. To select the external trigger mode/timer trigger mode (ADA0TMD bit = 1), set the high-speed conversion mode (ADA0M1.ADA0HS1 bit = 1). Do not input a trigger during the stabilization time that is inserted once after A/D conversion is enabled (ADA0CE bit = 1).
 5. When not using the A/D converter, stop A/D conversion by setting the ADA0CE bit to 0 to reduce power consumption.

(2) A/D converter mode register 1 (ADA0M1)

The ADA0M1 register is an 8-bit register that specifies the conversion time.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF201H

	7	6	5	4	3	2	1	0
ADA0M1	ADA0HS1	0	0	0	0	ADA0FR2	ADA0FR1	ADA0FR0

ADA0HS1	Normal conversion mode/high-speed mode (A/D conversion time) selection
0	Normal conversion mode
1	High-speed conversion mode

- Cautions**
1. Changing the ADA0M1 register is prohibited while A/D conversion is enabled (ADA0M0.ADA0CE bit = 1).
 2. When selecting the external trigger mode/timer trigger mode (ADA0M0.ADA0TMD bit = 1), set the high-speed conversion mode (ADA0HS1 bit = 1). Do not input a trigger during the stabilization time that is inserted once after A/D conversion is enabled (ADA0CE bit = 1).
 3. Be sure to clear bits 6 to 3 to "0".

Remark For A/D conversion time setting examples, see **Tables 13-2** and **13-3**.

Examples of the conversion time for each clock are shown below.

Table 13-2. Conversion Time Selection in Normal Conversion Mode (ADA0HS1 Bit = 0)

ADA0 FR2	ADA0 FR1	ADA0 FR0	A/D Conversion Time						Trigger Response Time
			Stabilization Time + Conversion Time + Wait Time	$f_{xx} = 20$ MHz	$f_{xx} = 16$ MHz	$f_{xx} = 12$ MHz	$f_{xx} = 10$ MHz	$f_{xx} = 4$ MHz	
0	0	0	$66/f_{xx}$ ($13/f_{xx} + 26/f_{xx} + 27/f_{xx}$)	Setting prohibited	Setting prohibited	Setting prohibited	$6.6 \mu\text{s}^{\text{Note}}$	$16.5 \mu\text{s}$	$3/f_{xx}$
0	0	1	$131/f_{xx}$ ($26/f_{xx} + 52/f_{xx} + 53/f_{xx}$)	$6.55 \mu\text{s}^{\text{Note}}$	$8.19 \mu\text{s}^{\text{Note}}$	$10.92 \mu\text{s}$	$13.1 \mu\text{s}$	Setting prohibited	$3/f_{xx}$
0	1	0	$196/f_{xx}$ ($39/f_{xx} + 78/f_{xx} + 79/f_{xx}$)	$9.8 \mu\text{s}$	$12.25 \mu\text{s}$	$16.33 \mu\text{s}$	$19.6 \mu\text{s}$	Setting prohibited	$3/f_{xx}$
0	1	1	$259/f_{xx}$ ($50/f_{xx} + 104/f_{xx} + 105/f_{xx}$)	$12.95 \mu\text{s}$	$16.19 \mu\text{s}$	$21.58 \mu\text{s}$	Setting prohibited	Setting prohibited	$3/f_{xx}$
1	0	0	$311/f_{xx}$ ($50/f_{xx} + 130/f_{xx} + 131/f_{xx}$)	$15.55 \mu\text{s}$	$19.44 \mu\text{s}$	Setting prohibited	Setting prohibited	Setting prohibited	$3/f_{xx}$
1	0	1	$363/f_{xx}$ ($50/f_{xx} + 156/f_{xx} + 157/f_{xx}$)	$18.15 \mu\text{s}$	$22.69 \mu\text{s}$	Setting prohibited	Setting prohibited	Setting prohibited	$3/f_{xx}$
1	1	0	$415/f_{xx}$ ($50/f_{xx} + 182/f_{xx} + 183/f_{xx}$)	$20.75 \mu\text{s}$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	$3/f_{xx}$
1	1	1	$467/f_{xx}$ ($50/f_{xx} + 208/f_{xx} + 209/f_{xx}$)	$23.35 \mu\text{s}$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	$3/f_{xx}$
Other than above			Setting prohibited						

Note Setting prohibited when $2.7 \text{ V} \leq AV_{\text{REF0}} < 3.0 \text{ V}$

Remarks 1. Stabilization time: A/D converter setup time ($1 \mu\text{s}$ or longer)
 Conversion time: Actual A/D conversion time (2.6 to $10.4 \mu\text{s}$)
 Wait time: Wait time inserted before the next conversion
 Trigger response time: If a software trigger is generated after the stabilization time, it is inserted before the conversion time.

2. For details about the operation timing, see **13.5.2 Conversion operation timing**.

In the normal conversion mode, conversion is started after the stabilization time has elapsed after the ADA0M0.ADA0CE bit is set to 1, and A/D conversion is performed only during the specified conversion time (2.6 to $10.4 \mu\text{s}$). Conversion stops after the conversion ends and the A/D conversion end interrupt request signal (INTAD) is generated after the wait time has elapsed.

Because conversion is stopped during the wait time, the operating current can be reduced.

Cautions 1. Set as $2.6 \mu\text{s} \leq \text{conversion time} \leq 10.4 \mu\text{s}$ when $3.0 \text{ V} \leq AV_{\text{REF0}} \leq 3.6 \text{ V}$.

Set as $3.9 \mu\text{s} \leq \text{conversion time} \leq 10.4 \mu\text{s}$ when $2.7 \text{ V} \leq AV_{\text{REF0}} < 3.0 \text{ V}$.

2. During A/D conversion, if the ADA0M0, ADA0M2, ADA0S, ADA0PFM, and ADA0PFT registers are written or a trigger is input, reconversion is carried out. However, if the stabilization time end timing conflicts with writing to these registers, or if the stabilization time end timing conflicts with the trigger input, the stabilization time of 64 clocks is reinserted.

If a conflict occurs again with the reinserted stabilization time end timing, the stabilization time is reinserted. Therefore do not set the trigger input interval and control register write interval to 64 clocks or lower.

Table 13-3. Conversion Time Selection in High-Speed Conversion Mode (ADA0HS1 Bit = 1)

ADA0 FR2	ADA0 FR1	ADA0 FR0	A/D Conversion Time					Trigger Response Time	
			Conversion Time (+ Stabilization Time)	$f_{xx} = 20$ MHz	$f_{xx} = 16$ MHz	$f_{xx} = 12$ MHz	$f_{xx} = 10$ MHz		$f_{xx} = 4$ MHz
0	0	0	$26/f_{xx}$ (+ $13/f_{xx}$)	Setting prohibited	Setting prohibited	Setting prohibited	$2.6 \mu\text{s}$ ^{Note} (+ $1.3 \mu\text{s}$)	$6.5 \mu\text{s}$ (+ $3.25 \mu\text{s}$)	$3/f_{xx}$
0	0	1	$52/f_{xx}$ (+ $26/f_{xx}$)	$2.6 \mu\text{s}$ ^{Note} (+ $1.3 \mu\text{s}$)	$3.25 \mu\text{s}$ ^{Note} (+ $1.625 \mu\text{s}$)	$4.333 \mu\text{s}$ (+ $2.167 \mu\text{s}$)	$5.2 \mu\text{s}$ (+ $2.6 \mu\text{s}$)	Setting prohibited	$3/f_{xx}$
0	1	0	$78/f_{xx}$ (+ $39/f_{xx}$)	$3.9 \mu\text{s}$ (+ $1.95 \mu\text{s}$)	$4.875 \mu\text{s}$ (+ $2.438 \mu\text{s}$)	$6.5 \mu\text{s}$ (+ $3.25 \mu\text{s}$)	$7.8 \mu\text{s}$ (+ $3.9 \mu\text{s}$)	Setting prohibited	$3/f_{xx}$
0	1	1	$104/f_{xx}$ (+ $50/f_{xx}$)	$5.2 \mu\text{s}$ (+ $2.5 \mu\text{s}$)	$6.5 \mu\text{s}$ (+ $3.125 \mu\text{s}$)	$8.667 \mu\text{s}$ (+ $4.167 \mu\text{s}$)	$10.4 \mu\text{s}$ (+ $5 \mu\text{s}$)	Setting prohibited	$3/f_{xx}$
1	0	0	$130/f_{xx}$ (+ $50/f_{xx}$)	$6.5 \mu\text{s}$ (+ $2.5 \mu\text{s}$)	$8.125 \mu\text{s}$ (+ $3.125 \mu\text{s}$)	Setting prohibited	Setting prohibited	Setting prohibited	$3/f_{xx}$
1	0	1	$156/f_{xx}$ (+ $50/f_{xx}$)	$7.8 \mu\text{s}$ (+ $2.5 \mu\text{s}$)	$9.75 \mu\text{s}$ (+ $3.125 \mu\text{s}$)	Setting prohibited	Setting prohibited	Setting prohibited	$3/f_{xx}$
1	1	0	$182/f_{xx}$ (+ $50/f_{xx}$)	$9.1 \mu\text{s}$ (+ $2.5 \mu\text{s}$)	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	$3/f_{xx}$
1	1	1	$208/f_{xx}$ (+ $50/f_{xx}$)	$10.4 \mu\text{s}$ (+ $2.5 \mu\text{s}$)	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	$3/f_{xx}$
Other than above			Setting prohibited						

Note Setting prohibited when $2.7 \text{ V} \leq AV_{REF0} < 3.0 \text{ V}$

Remarks 1. Conversion time: Actual A/D conversion time (2.6 to 10.4 μs)
 Stabilization time: A/D converter setup time (1 μs or longer)
 Trigger response time: If a software trigger, external trigger, or timer trigger is generated after the stabilization time, it is inserted before the conversion time.

2. For details about the operation timing, see **13.5.2 Conversion operation timing**.

In the high-speed conversion mode, conversion is started after the stabilization time has elapsed after the ADA0M0.ADA0CE bit is set to 1, and A/D conversion is performed only during the specified conversion time (2.6 to 10.4 μs). The A/D conversion end interrupt request signal (INTAD) is generated immediately after conversion ends.

In continuous conversion mode, the stabilization time is inserted only before the first conversion, and is not inserted after the second conversion (the A/D converter continues running).

Cautions 1. Set as $2.6 \mu\text{s} \leq$ conversion time $\leq 10.4 \mu\text{s}$ when $3.0 \text{ V} \leq AV_{REF0} \leq 3.6 \text{ V}$.

Set as $3.9 \mu\text{s} \leq$ conversion time $\leq 10.4 \mu\text{s}$ when $2.7 \text{ V} \leq AV_{REF0} < 3.0 \text{ V}$.

2. In the high-speed conversion mode, rewriting the ADA0M0, ADA0M2, ADA0S, ADA0PFM, and ADA0PFT registers and inputting a trigger are prohibited during the stabilization time.

(3) A/D converter mode register 2 (ADA0M2)

The ADA0M2 register specifies the hardware trigger mode.

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

Reset sets this register to 00H.

After reset: 00H	R/W	Address: FFFFF203H							
		7	6	5	4	3	2	1	0
ADA0M2		0	0	0	0	0	0	ADA0TMD1	ADA0TMD0

ADA0TMD1	ADA0TMD0	Specification of hardware trigger mode
0	0	External trigger mode (when ADTRG pin valid edge is detected)
0	1	Timer trigger mode 0 (when INTTP2CC0 interrupt request is generated)
1	0	Timer trigger mode 1 (when INTTP2CC1 interrupt request is generated)
1	1	Setting prohibited

Cautions 1. In the following modes, write data to the ADA0M2 register while A/D conversion is stopped (ADA0M0.ADA0CE bit = 0), and then enable A/D conversion (ADA0CE bit = 1).

- Normal conversion mode
 - One-shot select mode/one-shot scan mode in high-speed conversion mode
- 2.** Be sure to clear bits 7 to 2 to "0".

(4) Analog input channel specification register 0 (ADA0S)

The ADA0S register specifies the pin that inputs the analog voltage to be converted into a digital signal.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF202H

	7	6	5	4	3	2	1	0
ADA0S	0	0	0	0	ADA0S3	ADA0S2	ADA0S1	ADA0S0

ADA0S3	ADA0S2	ADA0S1	ADA0S0	Select mode	Scan mode
0	0	0	0	ANI0	ANI0
0	0	0	1	ANI1	ANI0, ANI1
0	0	1	0	ANI2	ANI0 to ANI2
0	0	1	1	ANI3	ANI0 to ANI3
0	1	0	0	ANI4	ANI0 to ANI4
0	1	0	1	ANI5 ^{Note1}	ANI0 to ANI5 ^{Note1}
0	1	1	0	ANI6 ^{Note2}	ANI0 to ANI6 ^{Note2}
0	1	1	1	ANI7 ^{Note2}	ANI0 to ANI7 ^{Note2}
1	0	0	0	ANI8 ^{Note2}	ANI0 to ANI8 ^{Note2}
1	0	0	1	ANI9 ^{Note2}	ANI0 to ANI9 ^{Note2}
1	0	1	0	Setting prohibited	Setting prohibited
1	0	1	1	Setting prohibited	Setting prohibited
1	1	0	0	Setting prohibited	Setting prohibited
1	1	0	1	Setting prohibited	Setting prohibited
1	1	1	0	Setting prohibited	Setting prohibited
1	1	1	1	Setting prohibited	Setting prohibited

Notes 1. V850ES/JC3-L (48-pin), V850ES/JE3-L only.

2. V850ES/JE3-L only.

Cautions 1. In the following modes, write data to the ADA0S register while A/D conversion is stopped (ADA0M0.ADA0CE bit = 0), and then enable A/D conversion (ADA0CE bit = 1).

- Normal conversion mode
- One-shot select mode/one-shot scan mode in high-speed conversion mode

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

(5) A/D conversion result registers n, nH (ADA0CRn, ADA0CRnH)

The ADA0CRn and ADA0CRnH registers store the A/D conversion results.

These registers are read-only, in 16-bit or 8-bit units. However, specify the ADA0CRn register for 16-bit access and the ADA0CRnH register for 8-bit access. For ADA0CRn, the 10 bits of the conversion result are read from the higher 10 bits, and 0 is read from the lower 6 bits. For ADA0CRnH, the higher 8 bits of the conversion result are read.

Caution Accessing the ADA0CRn and ADA0CRnH registers is prohibited in the following statuses. If a wait cycle is generated, it can be cleared only by a reset. For details, see 3.4.9 (1) Accessing special on-chip peripheral I/O registers.

- When the CPU operates on the subclock and main clock oscillation is stopped
- When the CPU operates on the internal oscillator clock

After reset: Undefined R Address: ADA0CR0 FFFFF210H, ADA0CR1 FFFFF212H,
 ADA0CR2 FFFFF214H, ADA0CR3 FFFFF216H,
 ADA0CR4 FFFFF218H, ADA0CR5 FFFFF21AH,
 ADA0CR6 FFFFF21CH, ADA0CR7 FFFFF21EH,
 ADA0CR8 FFFFF220H, ADA0CR9 FFFFF222H,

ADA0CRn	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0

After reset: Undefined R Address: ADA0CR0H FFFFF211H, ADA0CR1H FFFFF213H,
 ADA0CR2H FFFFF215H, ADA0CR3H FFFFF217H,
 ADA0CR4H FFFFF219H, ADA0CR5H FFFFF21BH,
 ADA0CR6H FFFFF21DH, ADA0CR7H FFFFF21FH,
 ADA0CR8H FFFFF221H, ADA0CR9H FFFFF223H,

ADA0CRnH	7	6	5	4	3	2	1	0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

Caution A write operation to the ADA0M0 and ADA0S registers may cause the contents of the ADA0CRn register to become undefined. After conversion, read the conversion result before writing to the ADA0M0 and ADA0S registers. Correct conversion results may not be read if a sequence other than the above is used.

The relationship between the analog voltage input to the analog input pins (ANIn) and the A/D conversion result (ADA0CRn register) is as follows.

$$SAR = INT \left(\frac{V_{IN}}{AV_{REF0}} \times 1,024 + 0.5 \right)$$

$$ADA0CR^{Note} = SAR \times 64$$

Or,

$$(SAR - 0.5) \times \frac{AV_{REF0}}{1,024} \leq V_{IN} < (SAR + 0.5) \times \frac{AV_{REF0}}{1,024}$$

INT(): Function that returns the integer of the value in ()

V_{IN}: Analog input voltage

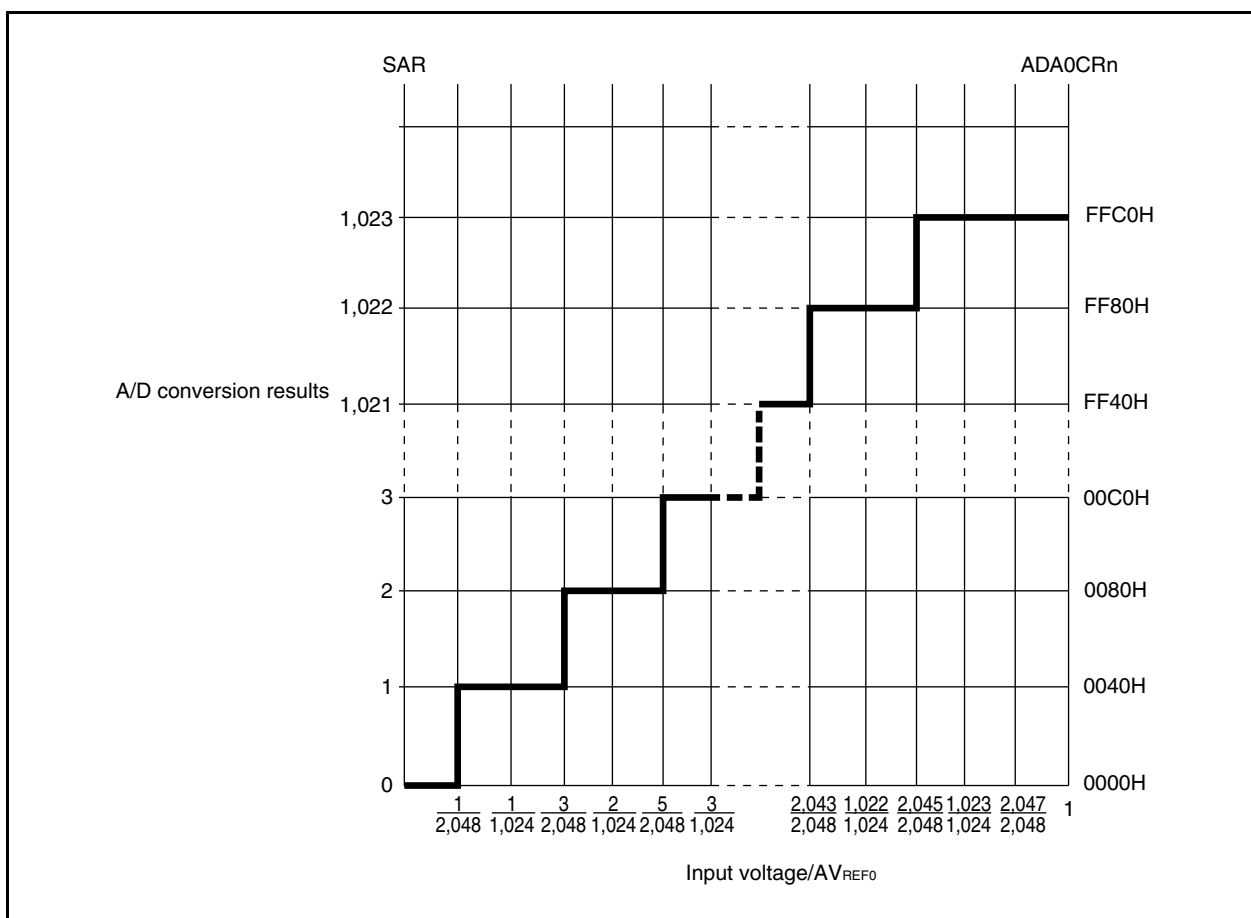
AV_{REF0}: AV_{REF0} pin voltage

ADA0CR: Value of ADA0CRn register

Note The lower 6 bits of the ADA0CRn register are fixed to 0.

The following shows the relationship between the analog input voltage and the A/D conversion results.

Figure 13-2. Relationship Between Analog Input Voltage and A/D Conversion Results



(6) Power-fail compare mode register (ADA0PFM)

The ADA0PFM register is an 8-bit register that sets the power-fail compare mode.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF204H

	<7>	6	5	4	3	2	1	0
ADA0PFM	ADA0PFE	ADA0PFC	0	0	0	0	0	0

ADA0PFE	Selection of power-fail compare enable/disable
0	Power-fail compare disabled
1	Power-fail compare enabled

ADA0PFC	Selection of power-fail compare mode
0	Generate an interrupt request signal (INTAD) when $ADA0CRnH \geq ADA0PFT$
1	Generate an interrupt request signal (INTAD) when $ADA0CRnH < ADA0PFT$

- Cautions 1.** In the select mode, the 8-bit data set to the ADA0PFT register is compared with the conversion result of the channel specified by the ADA0S register. If the result matches the condition specified by the ADA0PFC bit, the conversion result is stored in the ADA0CRn register and the INTAD signal is generated. If it does not match, the conversion result is stored in the ADA0CR0 register and the INTAD signal is not generated.
- 2.** In the scan mode, the 8-bit data set to the ADA0PFT register is compared with the conversion result of channel 0. If the result matches the condition specified by the ADA0PFC bit, the conversion result is stored in the ADA0CR0 register and the INTAD signal is generated. If it does not match, the conversion result is stored in the ADA0CR0 register but the INTAD signal is not generated. Also, regardless of the comparison result, scanning continues after comparison and the conversion result continue to be stored in the ADA0CRn register until scanning ends. However, the INTAD signal is not generated after the scanning has finished.
- 3.** In the following modes, write data to the ADA0PFM register while A/D conversion is stopped (ADA0M0.ADA0CE bit = 0), and then enable A/D conversion (ADA0CE bit = 1).
- Normal conversion mode
 - One-shot select mode/one-shot scan mode in high-speed conversion mode

(7) Power-fail compare threshold value register (ADA0PFT)

The ADA0PFT register sets the compare value in the power-fail compare mode.

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

Reset sets this register to 00H.

After reset: 00H	R/W	Address: FFFFF205H						
ADA0PFT	7	6	5	4	3	2	1	0

Caution In the following modes, write data to the ADA0PFT register while A/D conversion is stopped (ADA0M0.ADA0CE bit = 0), and then enable A/D conversion (ADA0CE bit = 1).

- Normal conversion mode
- One-shot select mode/one-shot scan mode in high-speed conversion mode

13.5 Operation

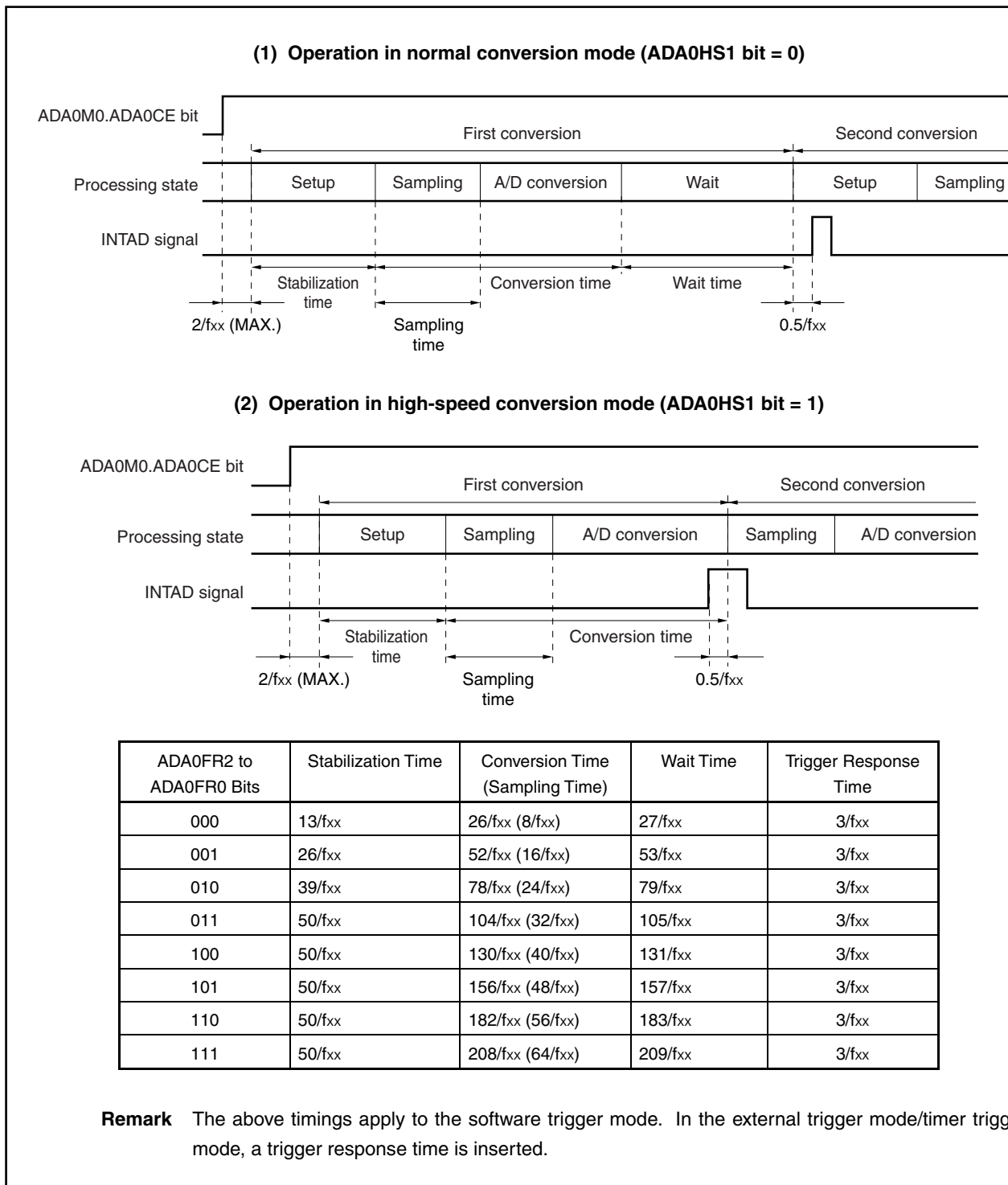
13.5.1 Basic operation

- <1> Set the operation mode, trigger mode, and conversion time for executing A/D conversion by using the ADA0M0, ADA0M1, ADA0M2, and ADA0S registers. When the ADA0CE bit of the ADA0M0 register is set, conversion is started in the software trigger mode and the A/D converter waits for a trigger in the external or timer trigger mode.
- <2> When A/D conversion is started, the voltage input to the selected analog input channel is sampled by the sample & hold circuit.
- <3> Once the sample & hold circuit has sampled the input channel for a specific time, it enters the hold status, and holds the input analog voltage until A/D conversion is complete.
- <4> Set bit 9 of the successive approximation register (SAR). The tap selector selects $(1/2) AV_{REF0}$ as the compare voltage generation DAC.
- <5> The voltage difference between the voltage of the compare voltage generation DAC and the analog input voltage is compared by the voltage comparator. If the analog input voltage is higher than $(1/2) AV_{REF0}$, the MSB of the SAR register remains set (1). If it is lower than $(1/2) AV_{REF0}$, the MSB is reset.
- <6> Next, bit 8 of the SAR register is automatically set and the next comparison is started. Depending on the value of bit 9, to which a result has been already set, the voltage tap of the compare voltage generation DAC is selected as follows.
- Bit 9 = 1: $(3/4) AV_{REF0}$
 - Bit 9 = 0: $(1/4) AV_{REF0}$
- This compare voltage and the analog input voltage are compared and, depending on the result, bit 8 is manipulated as follows.
- Analog input voltage \geq Compare voltage: Bit 8 = 1
Analog input voltage \leq Compare voltage: Bit 8 = 0
- <7> This comparison is continued to bit 0 of the SAR register.
- <8> When comparison of the 10 bits is complete, the valid digital result is stored in the SAR register, and is then transferred to and stored in the ADA0CRn register. After that, an A/D conversion end interrupt request signal (INTAD) is generated at the following timing.
- Continuous/one-shot select mode: After the first A/D conversion is complete
 - Continuous/one-shot scan mode: After A/D conversions are performed sequentially for analog input pins up to the one specified by the ADA0S register
- <9> In one-shot select mode, conversion stops here^{Note}. In one-shot scan mode, conversion stops after scanning once^{Note}. In continuous select mode, repeat steps <2> to <8> until the ADA0M0.ADA0CE bit is cleared to 0. In continuous scan mode, repeat steps <2> to <8> for each channel.
- Note** In the external trigger mode, timer trigger mode 0, or timer trigger mode 1, the trigger standby status is entered.

Remark The trigger standby status means the status after the stabilization time has passed.

13.5.2 Conversion timing

Figure 13-3. Conversion Timing (Continuous Conversion)



13.5.3 Trigger modes

The timing of starting conversion is specified by setting a trigger mode. The trigger modes include a software trigger mode and hardware trigger modes. The hardware trigger modes include timer trigger modes 0 and 1, and external trigger mode. The ADA0M0.ADA0TMD bit is used to set the trigger mode. The hardware trigger modes are set by the ADA0M2.ADA0TMD1 and ADA0M2.ADA0TMD0 bits.

Table 13-4. Trigger Modes

ADA0M0 Register	ADA0M2 Register		Trigger Mode
ADA0TMD Bit	ADA0TMD1 Bit	ADA0TMD0 Bit	
0	–	–	Software trigger mode
1	0	0	External trigger mode (based on ADTRG pin valid edge detection)
	0	1	Timer trigger mode 0 (based on INTTP2CC0 interrupt request occurrence)
	1	0	Timer trigger mode 1 (based on INTTP2CC1 interrupt request occurrence)
	1	1	Setting prohibited

(1) Software trigger mode

When the ADA0M0.ADA0CE bit is set to 1, the signal of the analog input pin (ANIn pin) specified by the ADA0S register is converted. When conversion is complete, the result is stored in the ADA0CRn register. At the same time, the A/D conversion end interrupt request signal (INTAD) is generated.

If the operation mode specified by the ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits is the continuous select/scan mode, the next conversion is started, unless the ADA0CE bit is cleared to 0 after completion of the first conversion. Conversion is performed once and ends if the operation mode is the one-shot select/scan mode.

When conversion is started, the ADA0M0.ADA0EF bit is set to 1 (indicating that conversion is in progress).

If the ADA0M0, ADA0M2, ADA0S, ADA0PFM, or ADA0PFT register is written during conversion, the conversion is aborted and started again from the beginning. However, writing to these registers is prohibited in the normal conversion mode and one-shot select mode/one-shot scan mode in the high-speed conversion mode.

(2) External trigger mode

In this mode, converting the signal of the analog input pin (ANIn pin) specified by the ADA0S register is started when an external trigger is input (to the ADTRG pin). Which edge of the external trigger is to be detected (that is, the rising edge, falling edge, or both rising and falling edges) can be specified by using the ADA0M0.ADA0ETS1 and ADA0M0.ADA0ETS0 bits. When the ADA0CE bit is set to 1, the A/D converter waits for the trigger, and starts conversion after the external trigger has been input.

When conversion is completed, the result of conversion is stored in the ADA0CRn register, regardless of whether the continuous select, continuous scan, one-shot select, or one-shot scan mode is set as the operation mode by the ADA0MD1 and ADA0MD0 bits. At the same time, the INTAD signal is generated, and the A/D converter waits for the trigger again.

When conversion is started, the ADA0EF bit is set to 1 (indicating that conversion is in progress). While the A/D converter is waiting for the trigger, however, the ADA0EF bit is cleared to 0 (indicating that conversion is stopped).

If a valid trigger is input during conversion, the conversion is aborted and started again from the beginning.

If the ADA0M0, ADA0M2, ADA0S, ADA0PFM, or ADA0PFT register is written during conversion, the conversion is aborted, and the A/D converter waits for the trigger again. However, writing to these registers is prohibited in the one-shot select mode/one-shot scan mode.

Caution When selecting the external trigger mode, set the high-speed conversion mode. Do not input a trigger during the stabilization time that is inserted once after A/D conversion is enabled (ADA0M0.ADA0CE bit = 1).

Remark The trigger standby status means the status after the stabilization time has passed.

(3) Timer trigger mode

In this mode, converting the signal of the analog input pin (ANIn) specified by the ADA0S register is started by the compare match interrupt request signal (INTTP2CC0 or INTTP2CC1) of the capture/compare register connected to the timer. The INTTP2CC0 or INTTP2CC1 signal is selected by the ADA0TMD1 and ADA0TMD0 bits, and conversion is started at the rising edge of the specified compare match interrupt request signal. When the ADA0CE bit is set to 1, the A/D converter waits for a trigger, and starts conversion when the compare match interrupt request signal of the timer is input.

When conversion is completed, regardless of whether the continuous select, continuous scan, one-shot select, or one-shot scan mode is set as the operation mode by the ADA0MD1 and ADA0MD0 bits, the result of the conversion is stored in the ADA0CRn register. At the same time, the INTAD signal is generated, and the A/D converter waits for the trigger again.

When conversion is started, the ADA0EF bit is set to 1 (indicating that conversion is in progress). While the A/D converter is waiting for the trigger, however, the ADA0EF bit is cleared to 0 (indicating that conversion is stopped). If a valid trigger is input during conversion, the conversion is aborted and started again from the beginning.

If the ADA0M0, ADA0M2, ADA0S, ADA0PFM, or ADA0PFT register is written during conversion, the conversion is stopped and the A/D converter waits for the trigger again. However, writing to these registers is prohibited in the one-shot select mode/one-shot scan mode.

Caution When selecting the timer trigger mode, set the high-speed conversion mode. Do not input a trigger during the stabilization time that is inserted once after A/D conversion is enabled (ADA0M0.ADA0CE bit = 1).

Remark The trigger standby status means the status after the stabilization time has passed.

13.5.4 Operation mode

Four operation modes are available: continuous select mode, continuous scan mode, one-shot select mode, and one-shot scan mode.

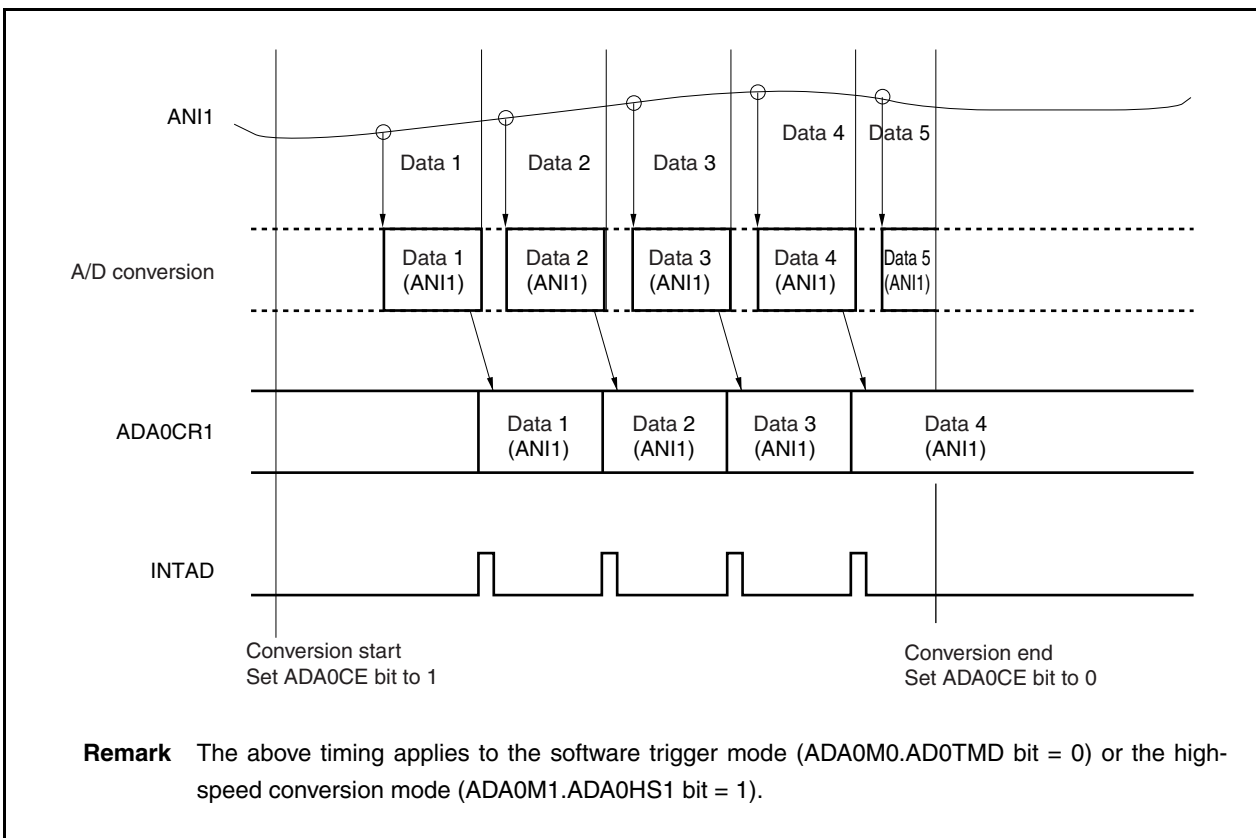
The operation mode is selected by the ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits.

(1) Continuous select mode

In this mode, the voltage of one analog input pin selected by the ADA0S register is continuously converted into a digital value.

The conversion result is stored in the ADA0CRn register corresponding to the analog input pin. Each time A/D conversion is completed, the A/D conversion end interrupt request signal (INTAD) is generated. After completion of conversion, the next conversion is started, unless the ADA0M0.ADA0CE bit is cleared to 0.

Figure 13-4. Example of Timing in Continuous Select Mode (ADA0S Register = 01H)



(2) Continuous scan mode

In this mode, analog input pins are sequentially selected, from the ANI0 pin to the pin specified by the ADA0S register, and their values are converted into digital values.

The result of each conversion is stored in the ADA0CRn register corresponding to the analog input pin. When conversion of the analog input pin specified by the ADA0S register is complete, the INTAD signal is generated, and A/D conversion is started again from the ANI0 pin, unless the ADA0CE bit is cleared to 0.

Figure 13-5. Example of Timing in Continuous Scan Mode (ADA0S Register = 03H) (1/2)

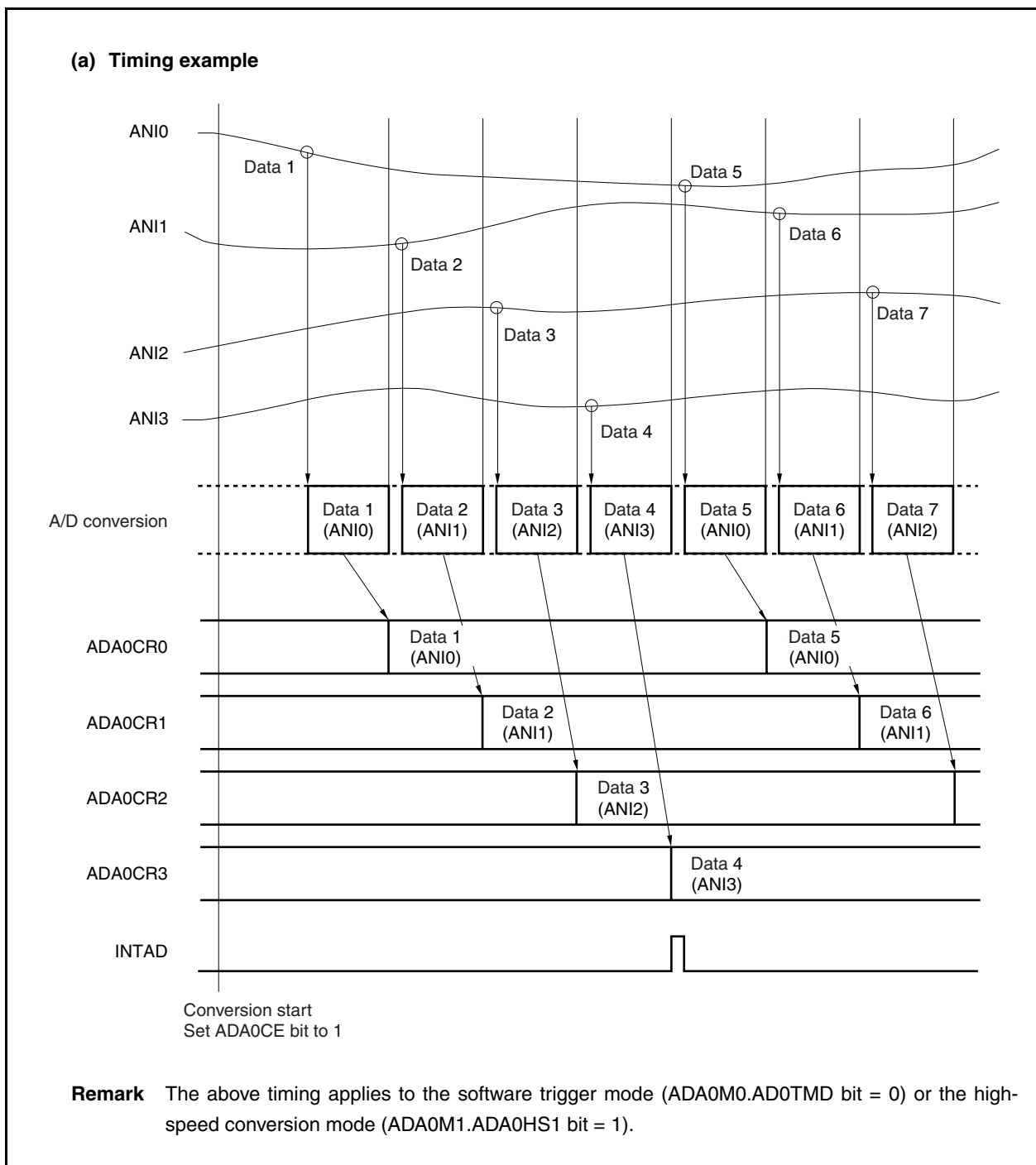
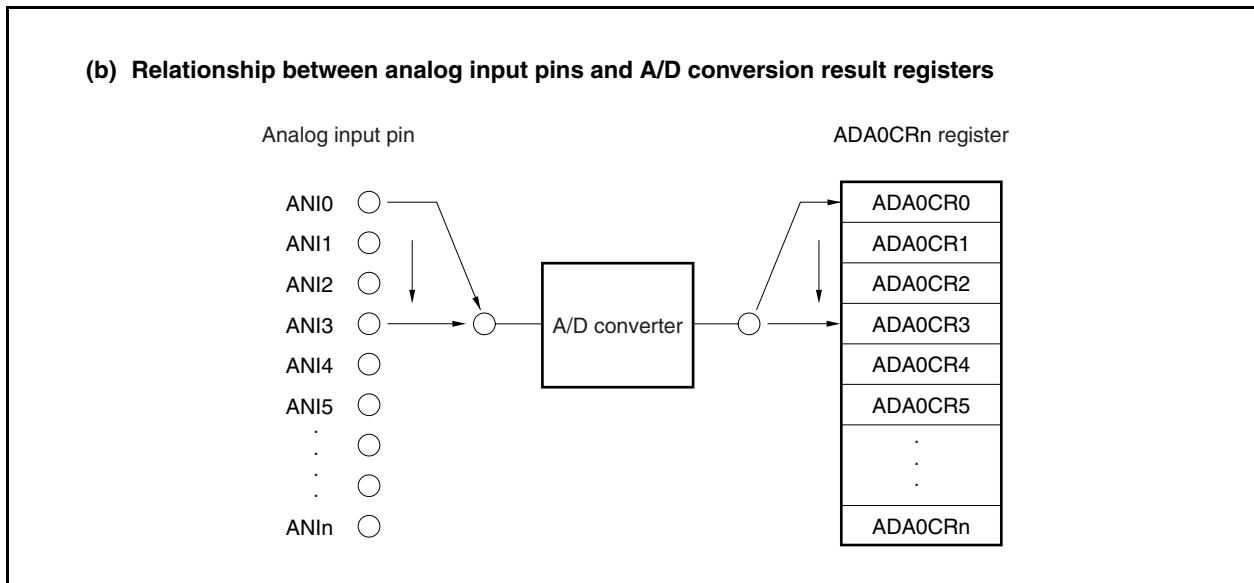


Figure 13-5. Example of Timing in Continuous Scan Mode (ADA0S Register = 03H) (2/2)

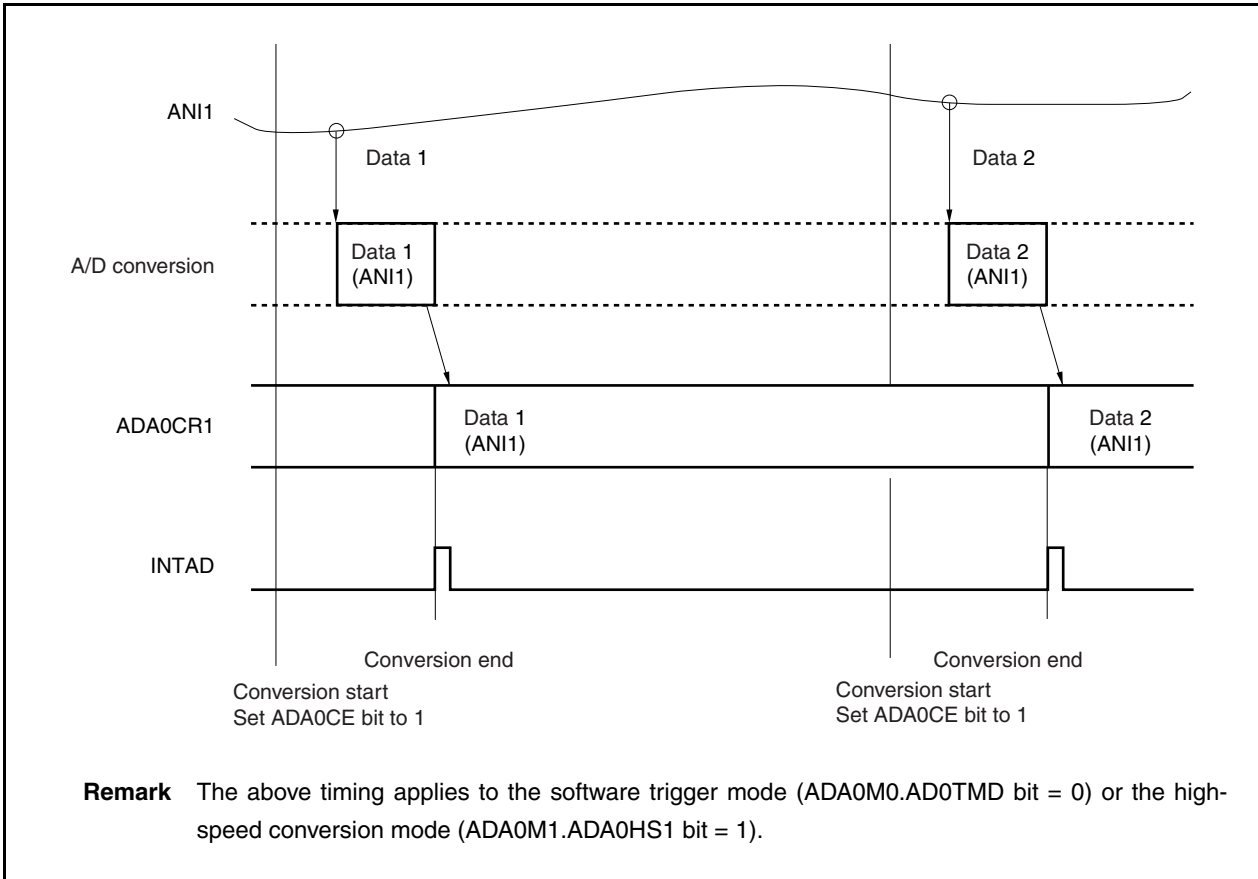


(3) One-shot select mode

In this mode, the voltage of the analog input pin specified by the ADA0S register is converted into a digital value only once.

The conversion result is stored in the ADA0CRn register corresponding to the analog input pin. In this mode, an analog input pin and an ADA0CRn register correspond on a one-to-one basis. When A/D conversion has been completed once, the INTAD signal is generated. A/D conversion is stopped after it has been completed.

Figure 13-6. Example of Timing in One-Shot Select Mode (ADA0S Register = 01H)



(4) One-shot scan mode

In this mode, analog input pins are sequentially selected, from the ANI0 pin to the pin specified by the ADA0S register, and their values are converted into digital values .

Each conversion result is stored in the ADA0CRn register corresponding to the analog input pin. When conversion of the analog input pin specified by the ADA0S register is complete, the INTAD signal is generated. A/D conversion is stopped after it has been completed.

Figure 13-7. Example of Timing in One-Shot Scan Mode (ADA0S Register = 03H) (1/2)

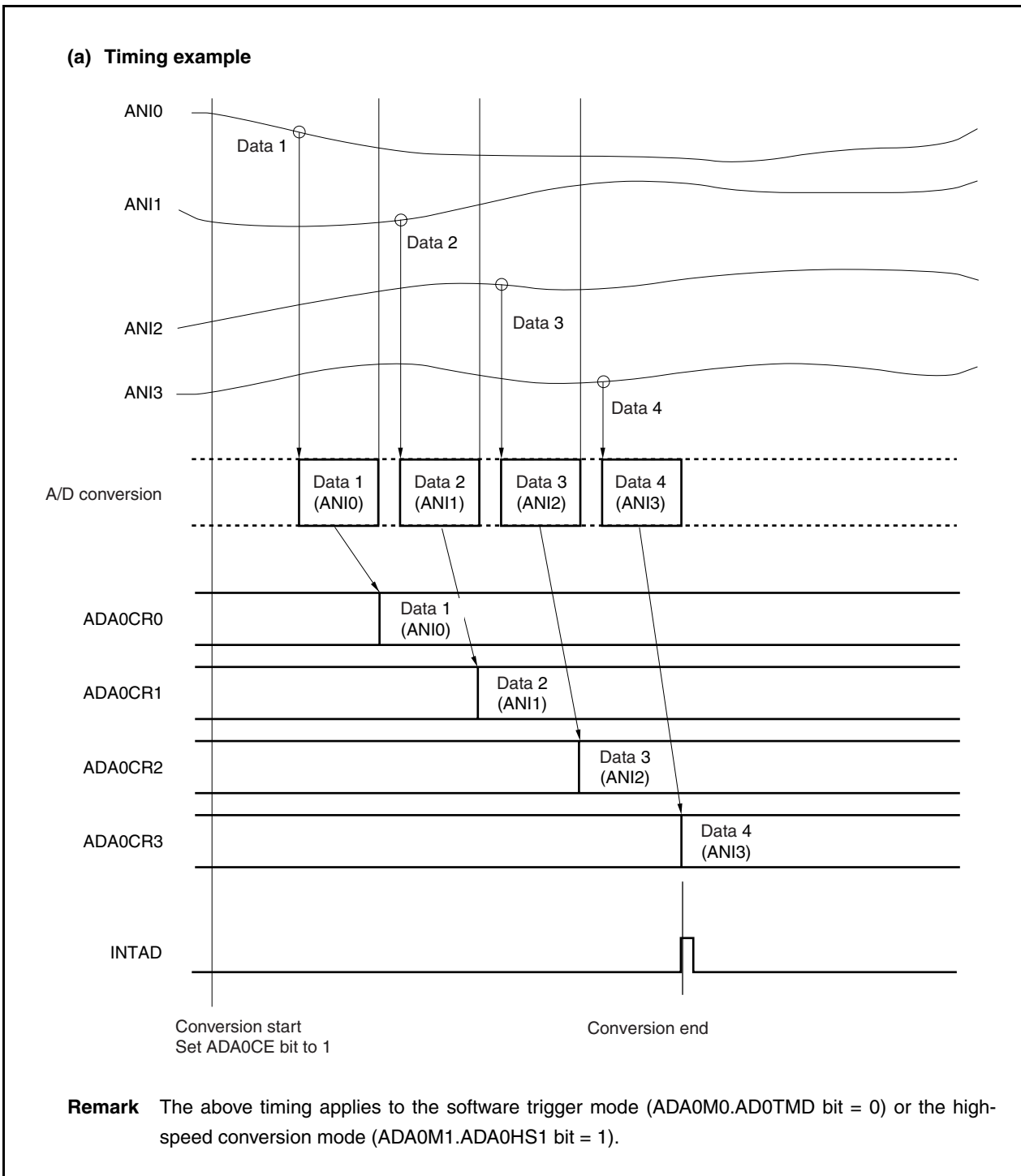
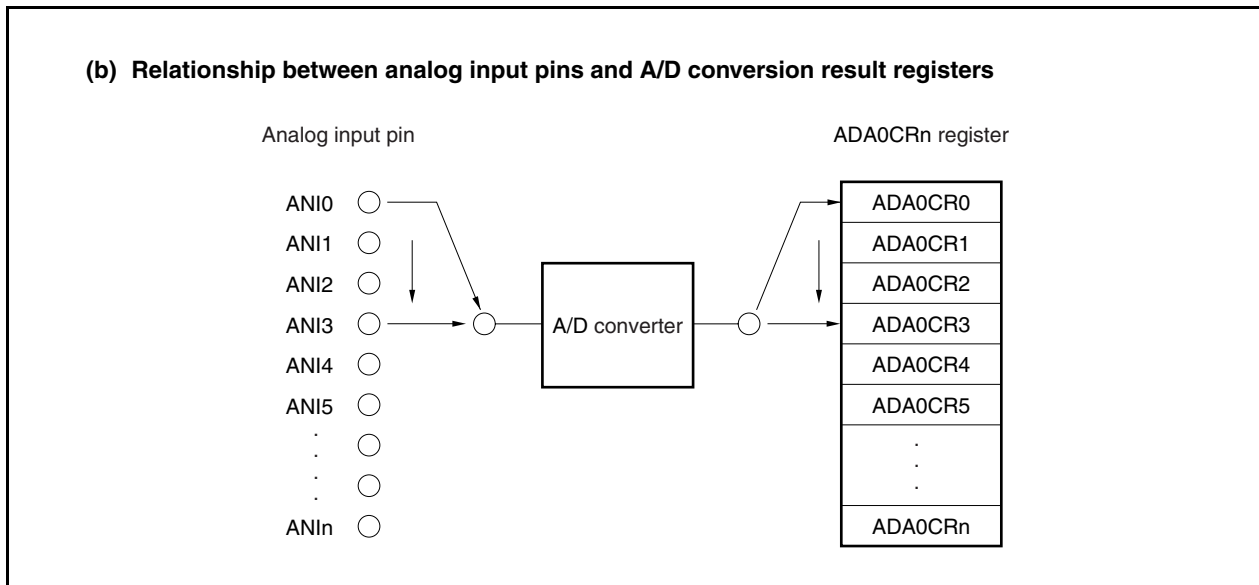


Figure 13-7. Example of Timing in One-Shot Scan Mode (ADA0S Register = 03H) (2/2)



13.5.5 Power-fail compare mode

In this mode, whether the input analog signal voltage is the specified voltage or higher or whether it is lower than the specified voltage is judged, and if the condition specified by the ADA0PFC bit is satisfied, the A/D conversion end interrupt request signal (INTAD) is generated.

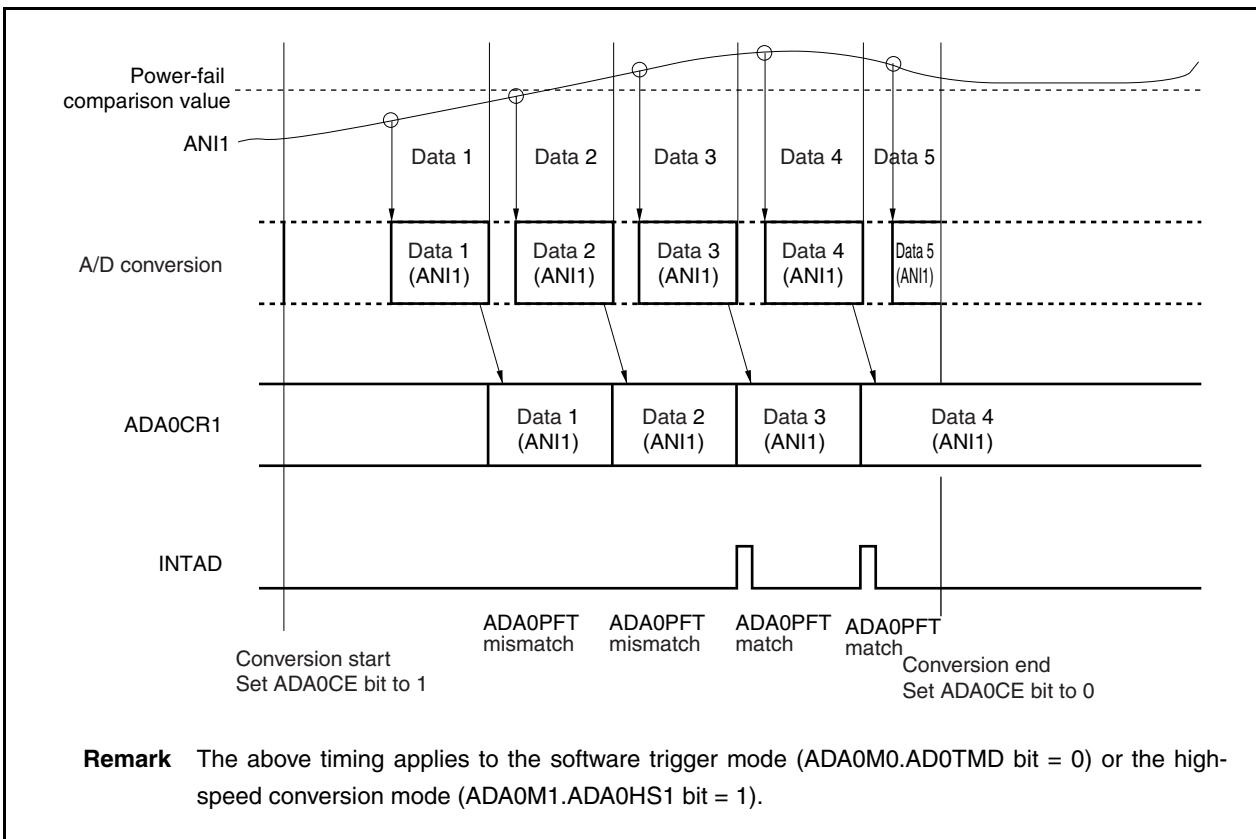
- When the ADA0PFM.ADA0PFE bit is 0, the INTAD signal is generated each time A/D conversion is completed at the following timing (normal use of the A/D converter) .
 - Continuous/one-shot select mode: After the first A/D conversion is complete
 - Continuous/one-shot scan mode: After A/D conversions are performed sequentially for the analog input pins up to the one specified by the ADA0S register
- When the ADA0PFE bit is 1 and when the ADA0PFM.ADA0PFC bit is 0, the value of the ADA0CRnH register is compared with the value of the ADA0PFT register when conversion is completed, and the INTAD signal is generated only if $ADA0CRnH \geq ADA0PFT$.
- When the ADA0PFE bit is 1 and when the ADA0PFC bit is 1, the value of the ADA0CRnH register is compared with the value of the ADA0PFT register when conversion is completed, and the INTAD signal is generated only if $ADA0CRnH < ADA0PFT$.

In the power-fail compare mode, four modes are available: continuous select mode, continuous scan mode, one-shot select mode, and one-shot scan mode.

(1) Continuous select mode

In this mode, the result of converting the voltage of the analog input pin specified by the ADA0S register is compared with the set value of the ADA0PFT register. If the result of power-fail comparison matches the condition set by the ADA0PFC bit, the conversion result is stored in the ADA0CRn register, and the INTAD signal is generated. If it does not match, the conversion result is stored in the ADA0CRn register, and the INTAD signal is not generated. After completion of the first conversion, the next conversion is started, unless the ADA0M0.ADA0CE bit is cleared to 0.

Figure 13-8. Example of Timing in Continuous Select Mode
 (When Power-Fail Comparison Is Made: ADA0PFM.ADA0PFC bit = 0, ADA0S Register = 01H)



(2) Continuous scan mode

In this mode, the results of converting the voltages of the analog input pins, from the ANI0 pin to the pin specified by the ADA0S register, are stored sequentially.

First, the conversion result of channel 0 is compared. If the result of power-fail comparison matches the condition set by the ADA0PFC bit, the conversion result is stored in the ADA0CR0 register, and the INTAD signal is generated. If it does not match, the conversion result is stored in the ADA0CR0 register, and the INTAD signal is not generated.

After the result of the first conversion has been stored in the ADA0CR0 register, the results of sequentially converting the voltages of the analog input pins up to the pin specified by the ADA0S register are continuously stored. After completion of conversion, the next conversion is started from the ANI0 pin again, unless the ADA0CE bit is cleared to 0.

Figure 13-9. Example of Timing in Continuous Scan Mode
 (When Power-Fail Comparison Is Made: ADA0PFM.ADA0PFC bit = 0, ADA0S Register = 03H) (1/2)

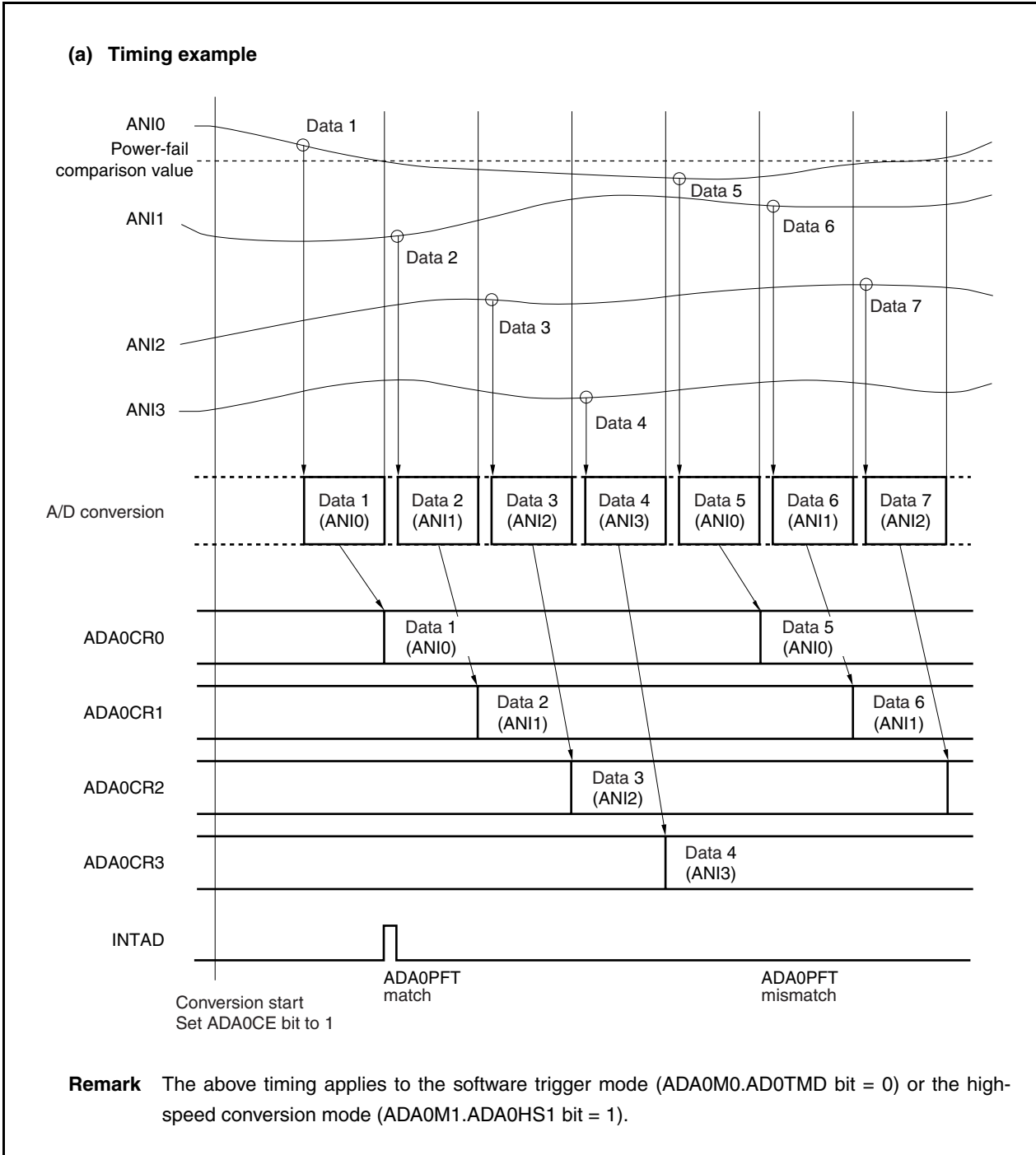
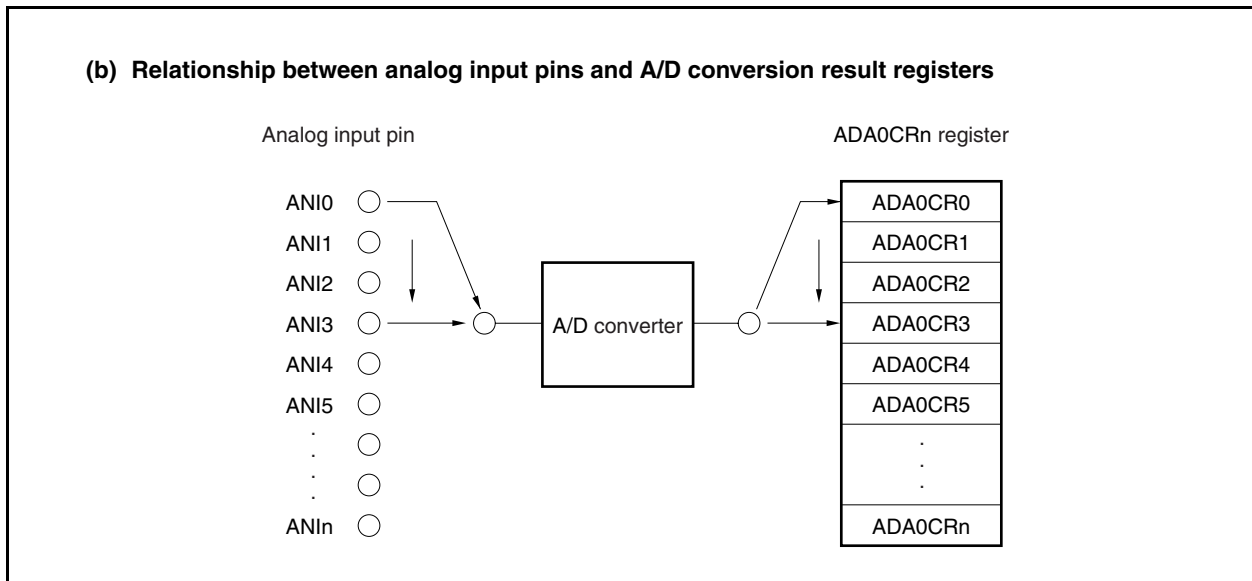


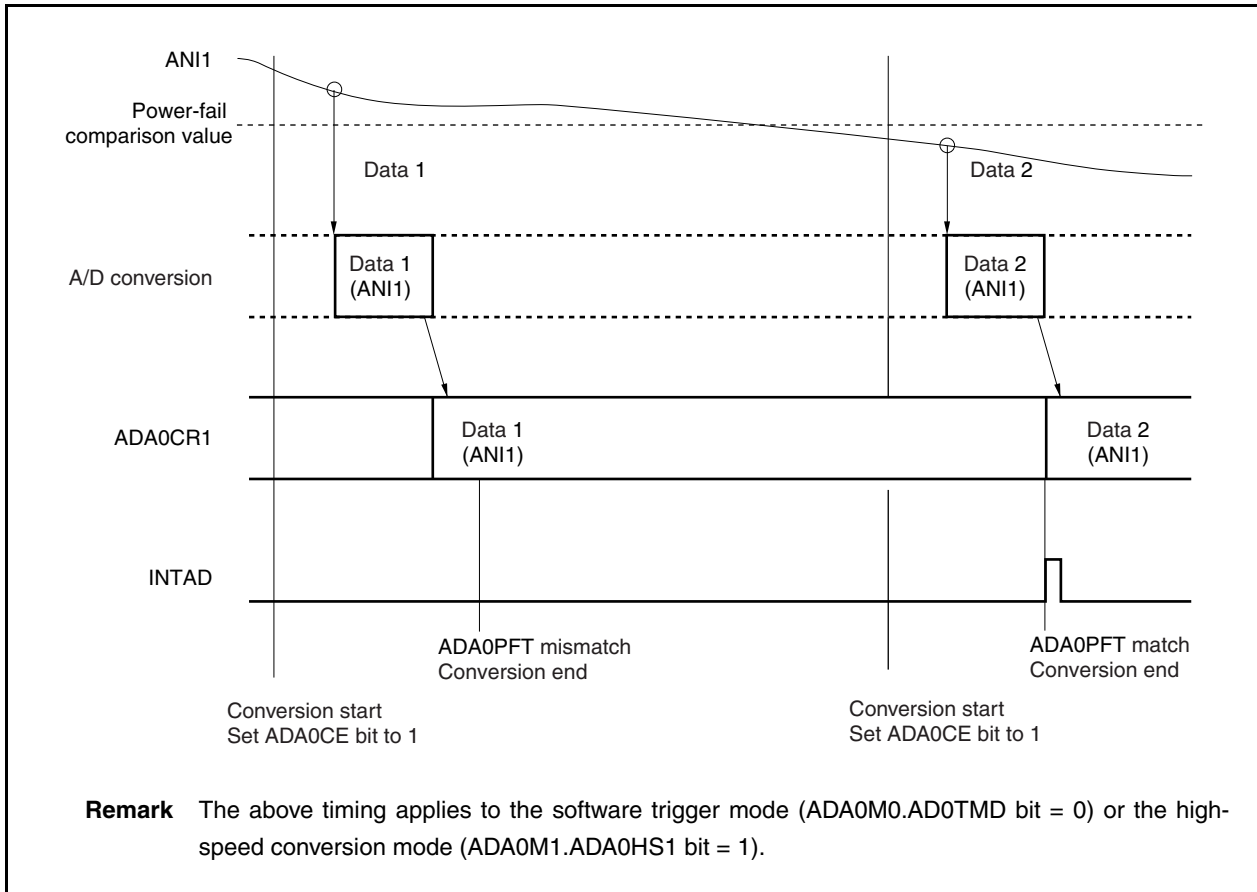
Figure 13-9. Example of Timing in Continuous Scan Mode
(When Power-Fail Comparison Is Made: ADA0PFM.ADA0PFC bit = 0, ADA0S Register = 03H) (2/2)



(3) One-shot select mode

In this mode, the result of converting the voltage of the analog input pin specified by the ADA0S register is compared with the set value of the ADA0PFT register. If the result of power-fail comparison matches the condition set by the ADA0PFC bit, the conversion result is stored in the ADA0CRn register, and the INTAD signal is generated. If it does not match, the conversion result is stored in the ADA0CRn register, and the INTAD signal is not generated. Conversion is stopped after it has been completed.

Figure 13-10. Example of Timing in One-Shot Select Mode
(When Power-Fail Comparison Is Made: ADA0PFM.ADA0PFC bit = 1, ADA0S Register = 01H)

**(4) One-shot scan mode**

In this mode, the results of converting the voltages of the analog input pins sequentially selected from the ANI0 pin to the pin specified by the ADA0S register are stored, and the set value of the ADA0CR0H register of channel 0 is compared with the set value of the ADA0PFT register. If the result of power-fail comparison matches the condition set by the ADA0PFC bit, the conversion result is stored in the ADA0CR0 register and the INTAD signal is generated. If it does not match, the conversion result is stored in the ADA0CR0 register, and the INTAD0 signal is not generated. After the result of the first conversion has been stored in the ADA0CR0 register, the results of converting the signals on the analog input pins specified by the ADA0S register are sequentially stored. Conversion is stopped after it has been completed.

Figure 13-11. Example of Timing in One-Shot Scan Mode
 (When Power-Fail Comparison Is Made: ADA0PFM.ADA0PFC bit = 0, ADA0S Register = 03H) (1/2)

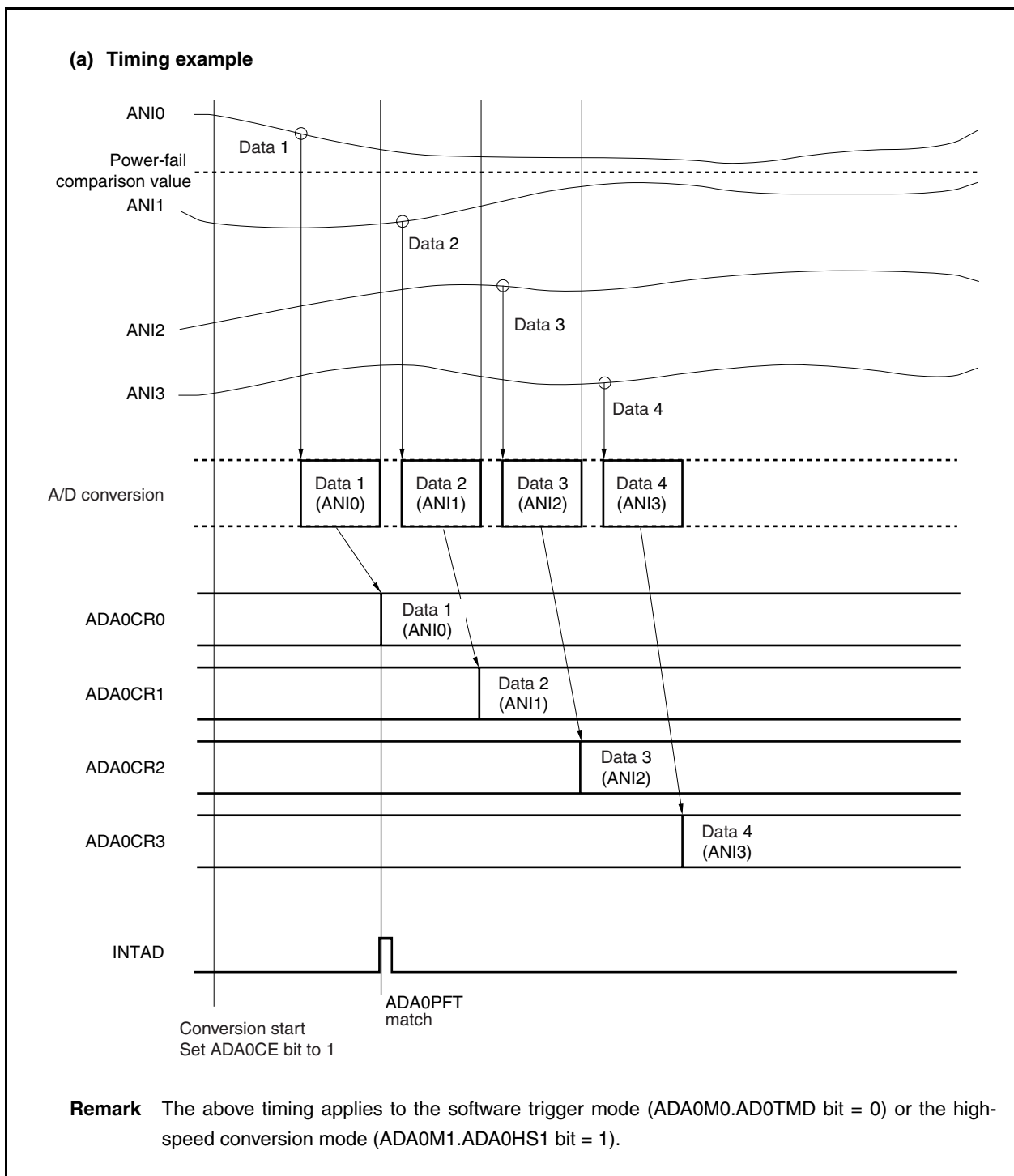
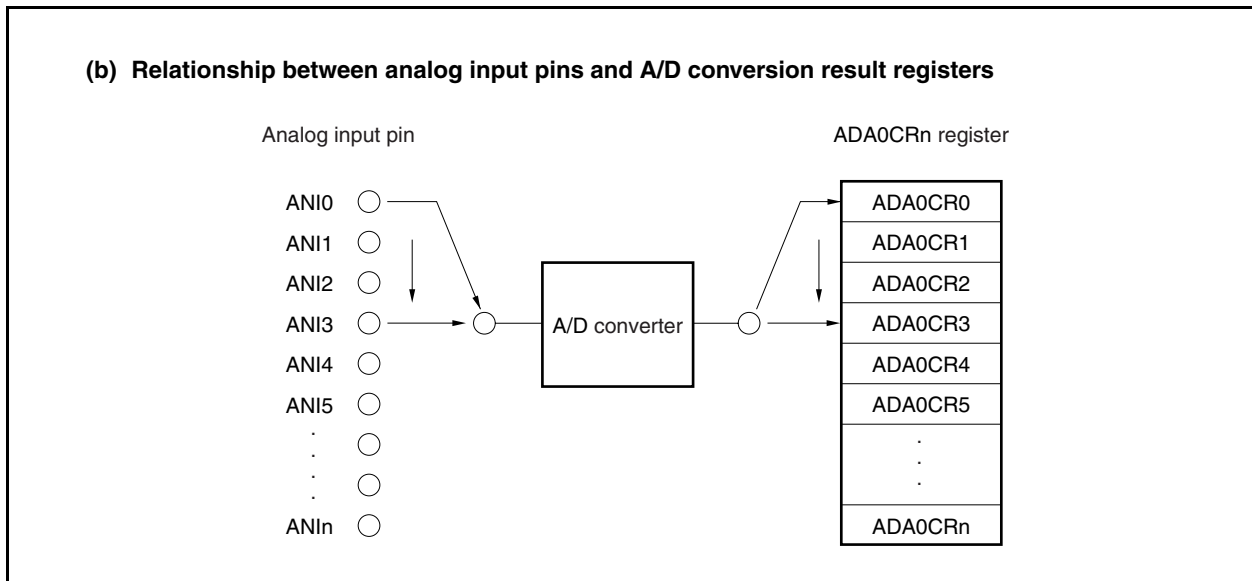


Figure 13-11. Example of Timing in One-Shot Scan Mode
(When Power-Fail Comparison Is Made: ADA0PFM.ADA0PFC bit = 0, ADA0S Register = 03H) (2/2)



13.6 Cautions

(1) When A/D converter is not used

When the A/D converter is not used, the power consumption can be reduced by clearing the ADA0M0.ADA0CE bit to 0.

(2) Input range of ANIn pins

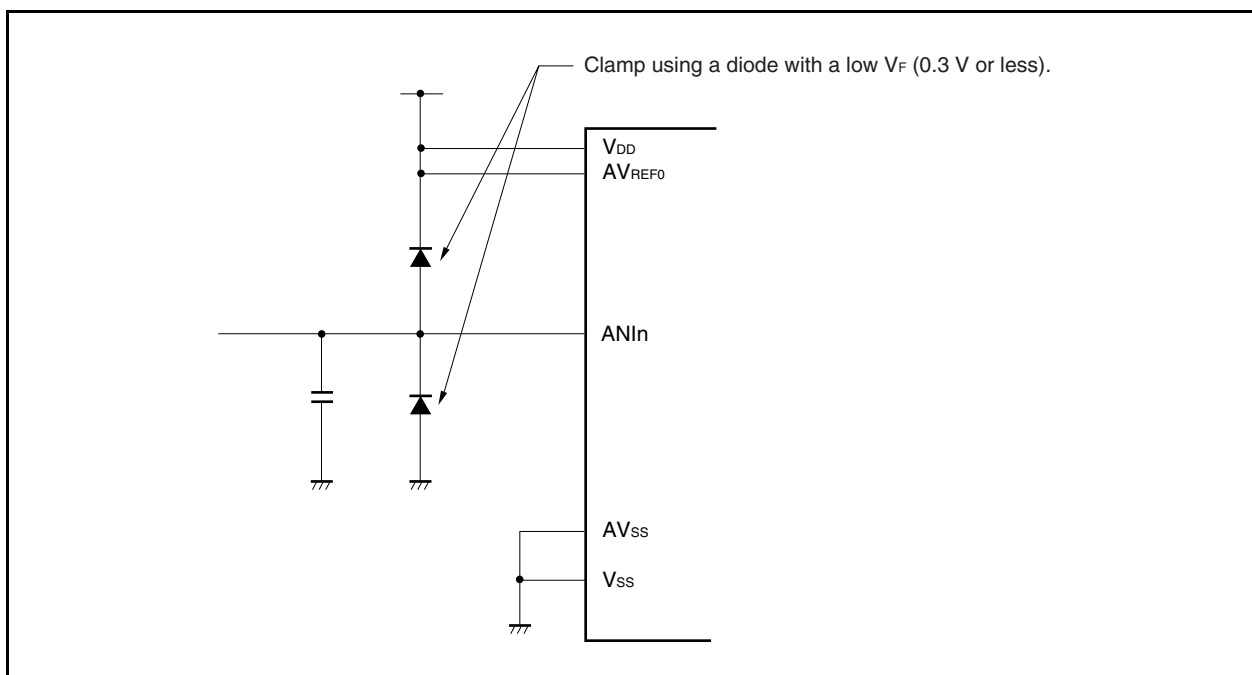
Input a voltage within the ratings to the ANIn pins. If a voltage equal to or higher than AV_{REF0} or equal to or lower than AV_{SS} (even within the range of the absolute maximum ratings) is input to any of these pins, the conversion value of that channel is undefined, and the conversion value of the other channels may also be affected.

(3) Countermeasures against noise

To maintain the 10-bit resolution, the ANIn pins must be effectively protected from noise. The influence of noise increases as the output impedance of the analog input source becomes higher. To lower the noise, connecting an external capacitor as shown in Figure 13-12 is recommended.

The capacitor must have a capacitance appropriate for the input signal change speed.

Figure 13-12. Handling of Analog Input Pin



(4) Alternate I/O

The analog input pins (ANIn) function alternately as port pins. When selecting one of the ANIn pins to execute A/D conversion, do not execute an instruction to read an input port or write to an output port during conversion as the conversion resolution may drop.

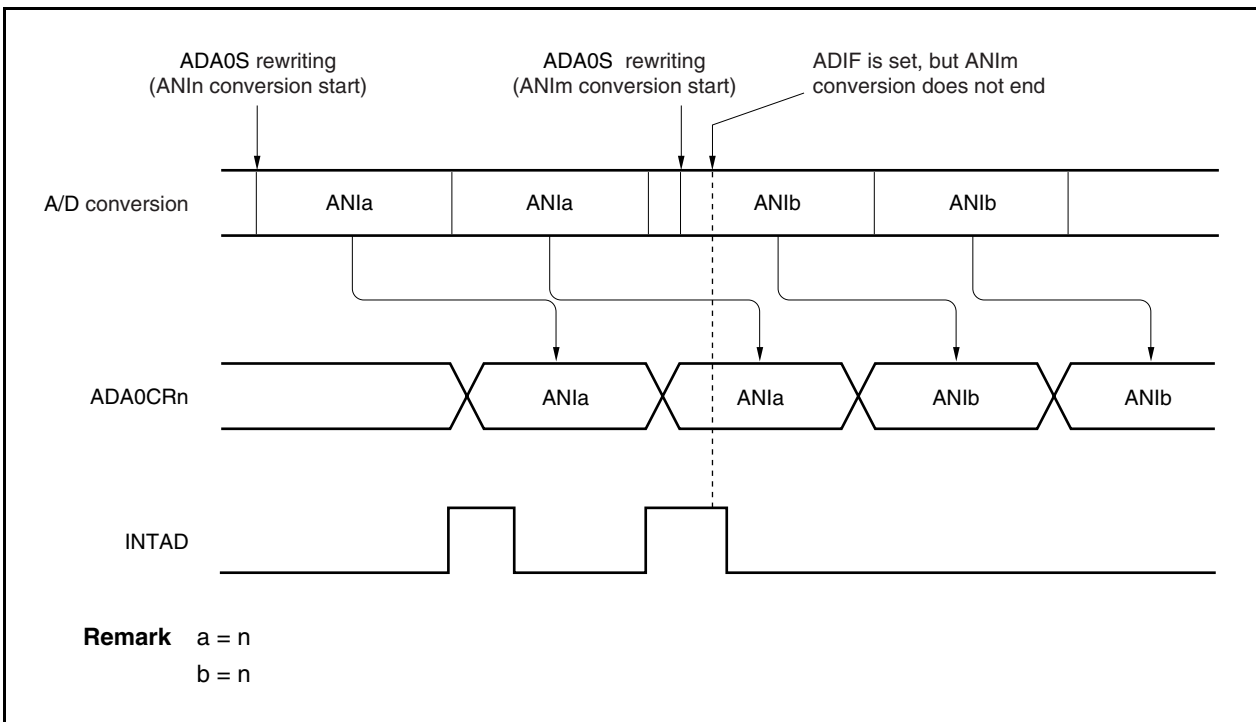
Also the conversion resolution may drop at the pins set as output port pins during A/D conversion if the output current fluctuates due to the effect of the external circuit connected to the port pins.

If a digital pulse is applied to a pin adjacent to the pin whose input signal is being converted, the A/D conversion value may not be as expected due to the influence of coupling noise. Therefore, do not apply a pulse to a pin adjacent to the pin undergoing A/D conversion.

(5) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared (0) even if the contents of the ADA0S register are changed. If the analog input pin is changed during A/D conversion, therefore, the result of converting the previously selected analog input signal may be stored and the conversion end interrupt request flag may be set (1) immediately before the ADA0S register is rewritten. If the ADIF flag is read immediately after the ADA0S register is rewritten, the ADIF flag may be set (1) even though the A/D conversion of the newly selected analog input pin has not been completed. When A/D conversion is stopped, clear (0) the ADIF flag before resuming conversion.

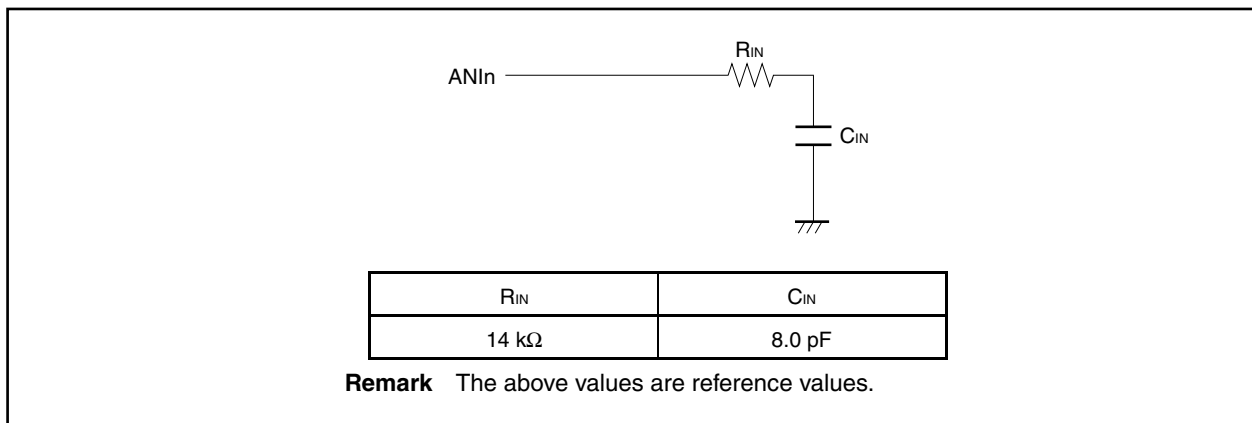
Figure 13-13. Timing of Generating A/D Conversion End Interrupt Request



(6) Internal equivalent circuit

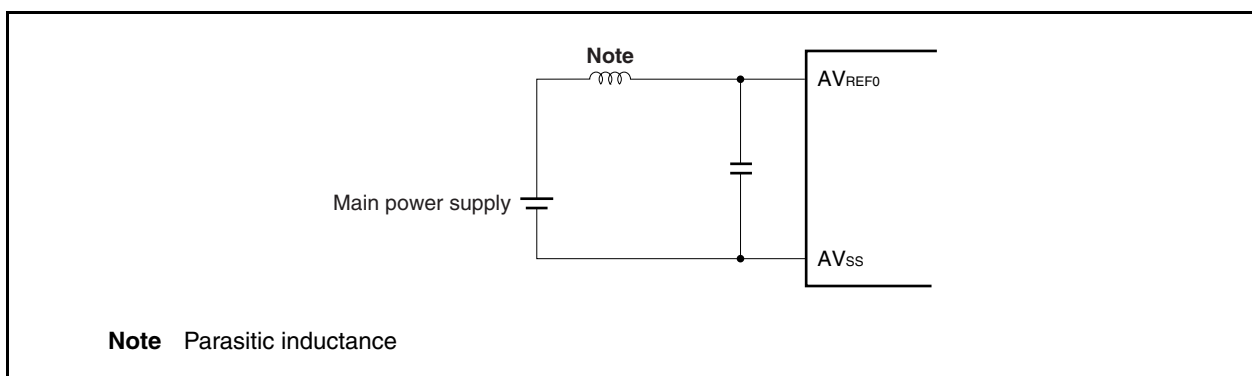
The following shows the equivalent circuit of the analog input block.

Figure 13-14. Internal Equivalent Circuit of ANIn Pin

**(7) AVREF0 pin**

- (a) The AVREF0 pin is used as the power supply pin of the A/D converter and also supplies power to the alternate-function ports. In an application where a backup power supply is used, be sure to supply the same voltage as V_{DD} to the AVREF0 pin as shown in Figure 13-15.
- (b) The AVREF0 pin is also used as the reference voltage pin of the A/D converter. If the source supplying power to the AVREF0 pin has a high impedance or if the power supply has a low current supply capability, the reference voltage may fluctuate due to the current that flows during conversion (especially, immediately after the conversion enable bit ADA0CE has been set to 1). As a result, the conversion accuracy may drop. To avoid this, it is recommended to connect a capacitor across the AVREF0 and AVSS pins to suppress the reference voltage fluctuation as shown in Figure 13-15.
- (c) If the source supplying power to the AVREF0 pin has a high DC resistance, the voltage when conversion is enabled may be lower than the voltage when conversion is stopped, because of a voltage drop caused by the A/D conversion current.

Figure 13-15. Example of Handling AVREF0 Pin



(8) Reading ADA0CRn register

When the ADA0M0 to ADA0M2, ADA0S, ADA0PFM, or ADA0PFT register is written, the contents of the ADA0CRn register may be undefined. Read the conversion result after completion of conversion and before writing to the ADA0M0 to ADA0M2, ADA0S, ADA0PFM, or ADA0PFT register. Also, when an external/timer trigger is acknowledged, the contents of the ADA0CRn register may be undefined. Read the conversion result after completion of conversion and before the next external/timer trigger is acknowledged. The correct conversion result may not be read at a timing different from the above.

(9) Standby mode

Because the A/D converter stops operating in the STOP mode, power consumption can be reduced, but the conversion results will be invalid. Operations are resumed after the STOP mode is released, but the A/D conversion results after the STOP mode is released are invalid. When using the A/D converter after the STOP mode is released, before setting the STOP mode or releasing the STOP mode, clear the ADA0M0.ADA0CE bit to 0 then set the ADA0CE bit to 1 after releasing the STOP mode.

In the IDLE1, IDLE2, or subclock operation mode, operation continues. In the IDLE1 and IDLE2 modes, since the analog input voltage value cannot be retained, the A/D conversion results after the IDLE1 and IDLE2 modes are released are invalid. The results of conversions before the IDLE1 and IDLE2 modes were set are valid. To lower the power consumption, therefore, clear the ADA0M0.ADA0CE bit to 0.

(10) High-speed conversion mode

In the high-speed conversion mode, rewriting the ADA0M0, ADA0M2, ADA0S, ADA0PFM, and ADA0PFT registers and inputting a trigger during the stabilization time are prohibited.

(11) A/D conversion time

A/D conversion time is the total time of stabilization time, conversion time, wait time, and trigger response time (for details of these times, refer to **Table 13-2 Conversion Time Selection in Normal Conversion Mode (ADA0HS1 Bit = 0)** and **Table 13-3 Conversion Time Selection in High-Speed Conversion Mode (ADA0HS1 Bit = 1)**).

During A/D conversion in the normal conversion mode, if the ADA0M0, ADA0M2, ADA0S, ADA0PFM, and ADA0PFT registers are written or a trigger is input, reconversion is carried out. However, if the stabilization time end timing conflicts with writing to these registers, or if the stabilization time end timing conflicts with the trigger input, the stabilization time of 64 clocks is reinserted.

If a conflict occurs again with the reinserted stabilization time end timing, the stabilization time is reinserted. Therefore do not set the trigger input interval and control register write interval to 64 clocks or lower.

(12) 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 countermeasures in the program such as averaging the A/D conversion results.

(13) A/D conversion result hysteresis characteristics

The successive comparison type A/D converter holds the analog input voltage in the internal sample & hold capacitor and then performs A/D conversion. After A/D conversion has finished, the analog input voltage remains in the internal sample & hold capacitor. As a result, the following phenomena may occur.

- When the same channel is used for A/D conversions, if the voltage is higher or lower than the previous A/D conversion, then hysteresis characteristics may appear where the conversion result is affected by the previous value. Thus, even if the conversion is performed at the same potential, the result may vary.
- When switching the analog input channel, hysteresis characteristics may appear where the conversion result is affected by the previous channel value. This is because one A/D converter is used for the A/D conversions. Thus, even if the conversion is performed at the same potential, the result may vary.

Therefore, to obtain a more accurate conversion result, perform A/D conversion twice successively for the same channel, and discard the first conversion result.

13.7 How to Read A/D Converter Characteristics Table

This section describes the terms related to the A/D converter.

(1) Resolution

The minimum analog input voltage that can be recognized, that is, the ratio of an analog input voltage to 1 bit of digital output is called 1 LSB (least significant bit). The ratio of 1 LSB to the full scale is expressed as %FSR (full-scale range). %FSR is the ratio of a range of convertible analog input voltages expressed as a percentage, and can be expressed as follows, independently of the resolution.

$$\begin{aligned} 1\%FSR &= (\text{Maximum value of convertible analog input voltage} - \text{Minimum value of convertible analog input voltage})/100 \\ &= (AV_{REF0} - 0)/100 \\ &= AV_{REF0}/100 \end{aligned}$$

When the resolution is 10 bits, 1 LSB is as follows:

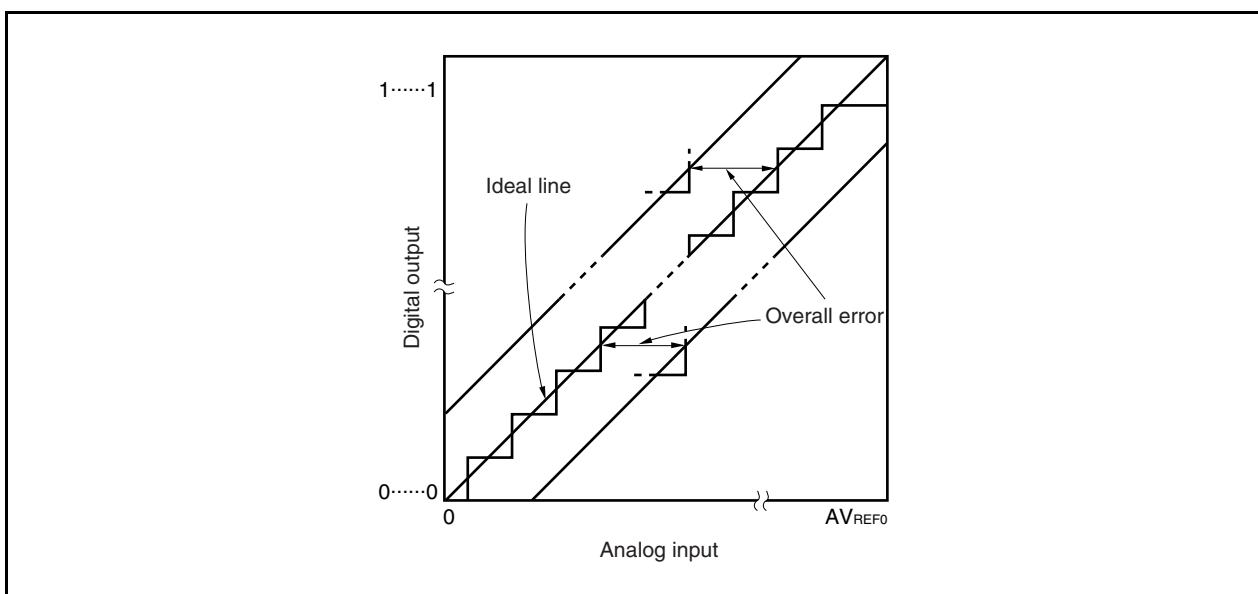
$$\begin{aligned} 1 \text{ LSB} &= 1/2^{10} = 1/1,024 \\ &= 0.098\%FSR \end{aligned}$$

The accuracy is determined by the overall error, independently of the resolution.

(2) Overall error

This is the maximum value of the difference between an actually measured value and a theoretical value. It is a total of zero-scale error, full-scale error, linearity error, and a combination of these errors. The overall error in the characteristics table does not include the quantization error.

Figure 13-16. Overall Error

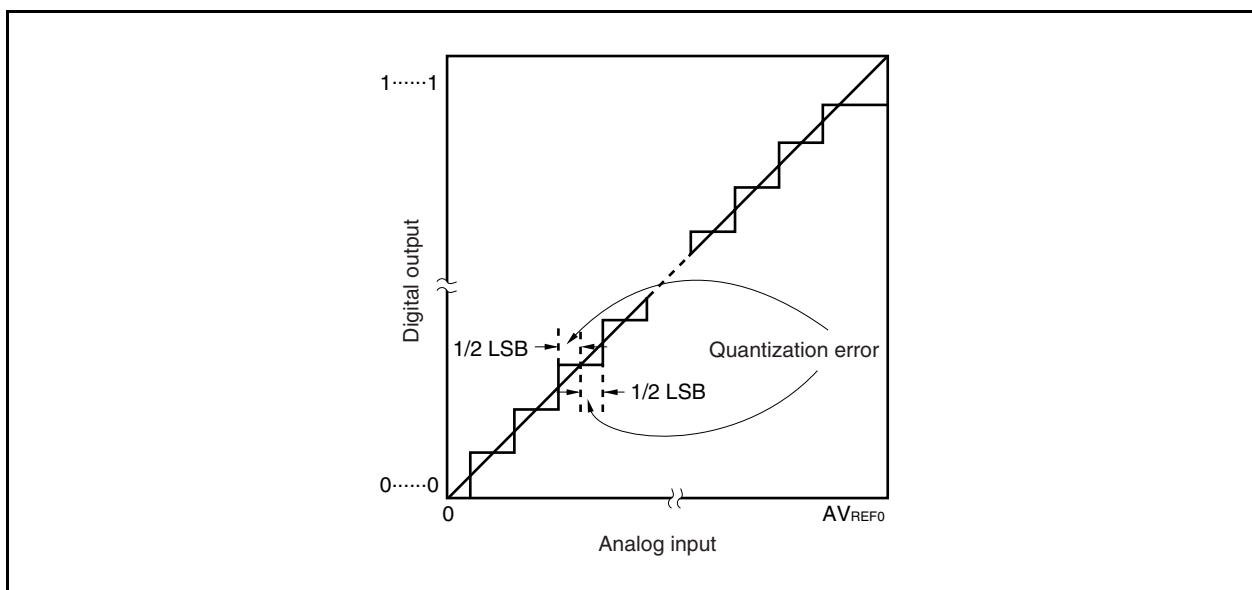


(3) Quantization error

This is an error of $\pm 1/2$ LSB that inevitably occurs when an analog value is converted into a digital value. Because the A/D converter converts analog input voltages in a range of $\pm 1/2$ LSB into the same digital codes, a quantization error is unavoidable.

This error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, or differential linearity error in the characteristics table.

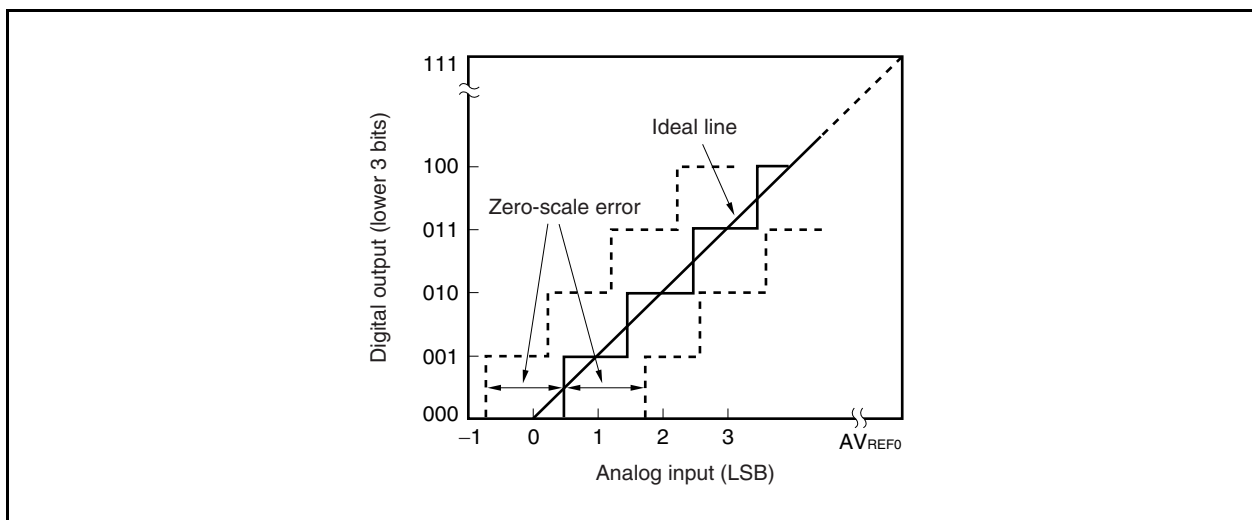
Figure 13-17. Quantization Error



(4) Zero-scale error

This is the difference between the actually measured analog input voltage and its theoretical value when the digital output changes from 0...000 to 0...001 (1/2 LSB).

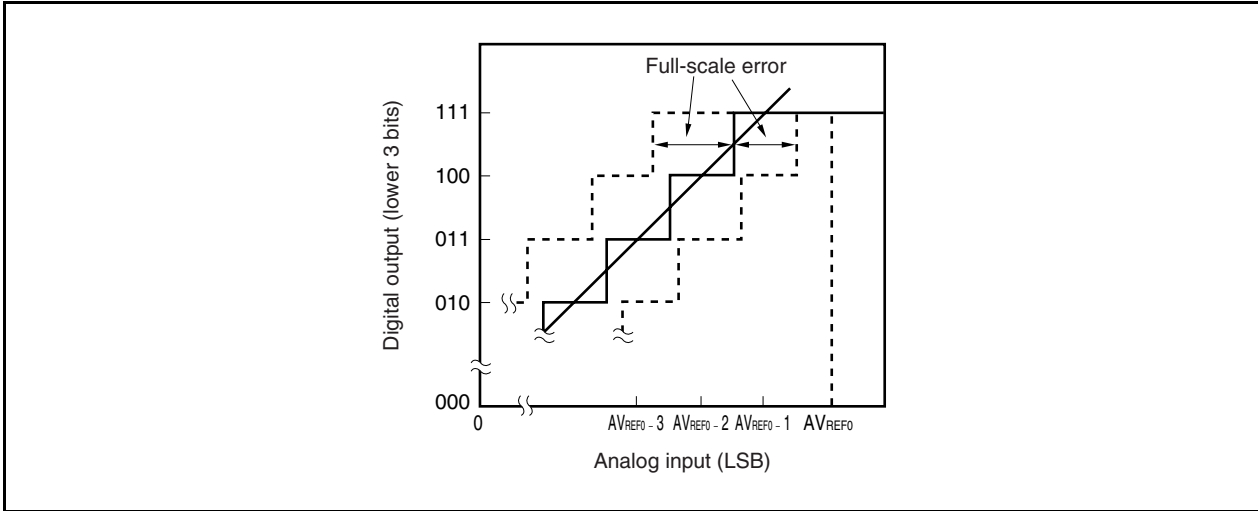
Figure 13-18. Zero-Scale Error



(5) Full-scale error

This is the difference between the actually measured analog input voltage and its theoretical value when the digital output changes from 1...110 to 1...111 (full scale - 3/2 LSB).

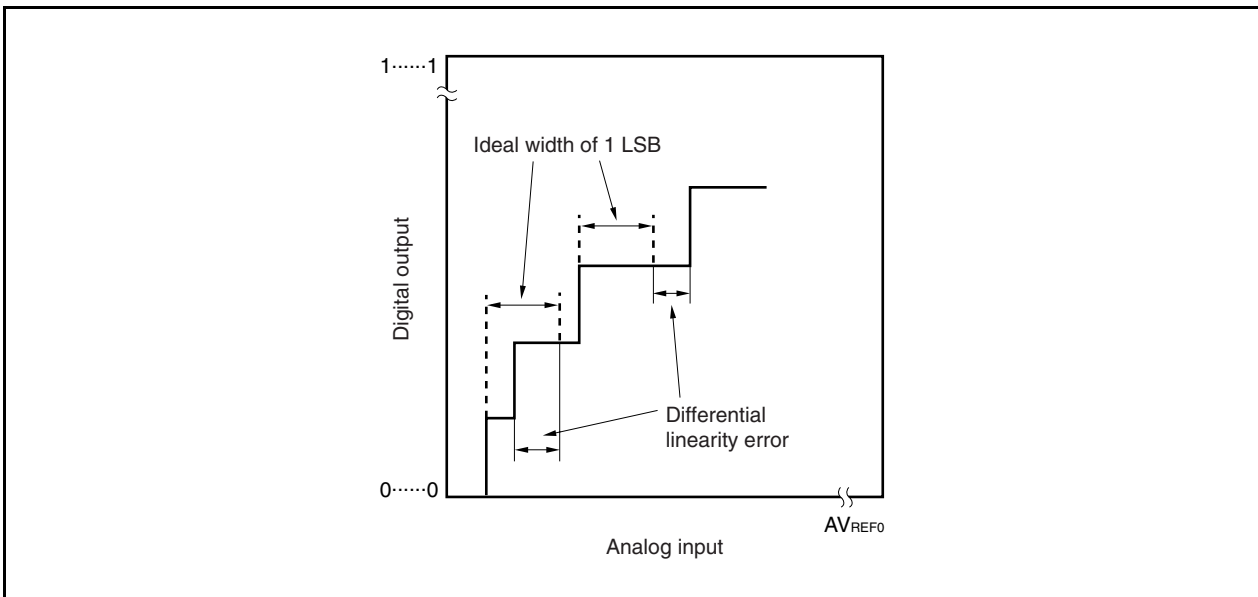
Figure 13-19. Full-Scale Error



(6) Differential linearity error

Ideally, the width to output a specific code is 1 LSB. This error indicates the difference between the actually measured value and its theoretical value when a specific code is output. 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_{SS} to AV_{REF0} . When the input voltage is increased or decreased, or when two or more channels are used, see 13.7 (2) Overall error.

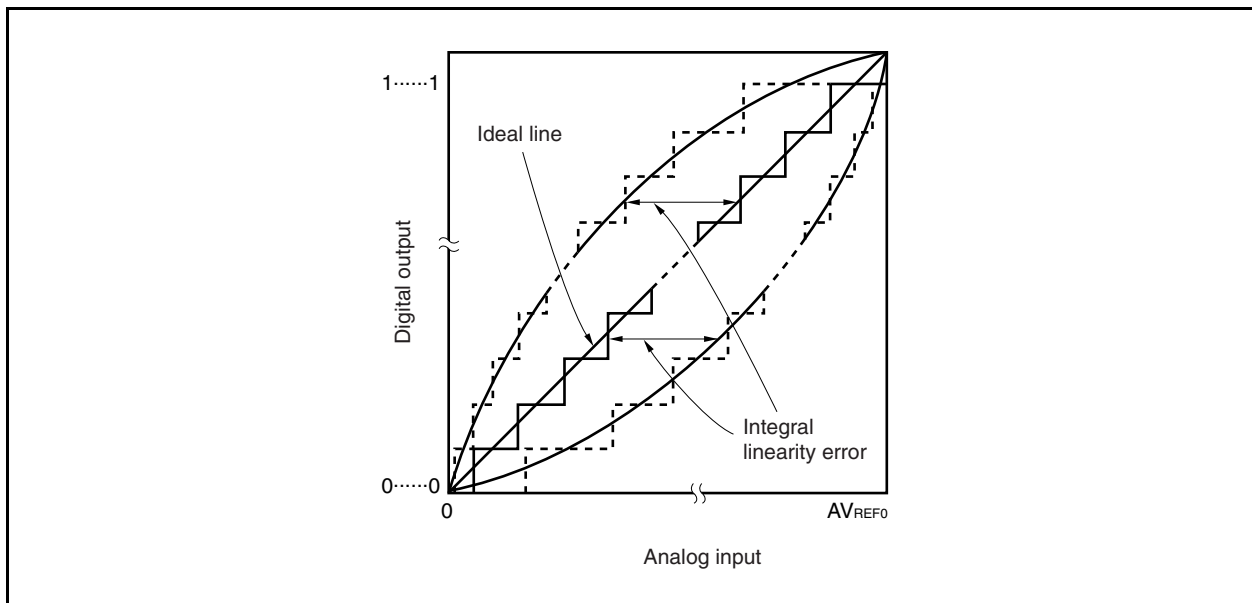
Figure 13-20. Differential Linearity Error



(7) Integral linearity error

This error indicates the extent to which the conversion characteristics differ from the ideal linear relationship. It indicates the maximum value of the difference between the actually measured value and its theoretical value where the zero-scale error and full-scale error are 0.

Figure 13-21. Integral Linearity Error



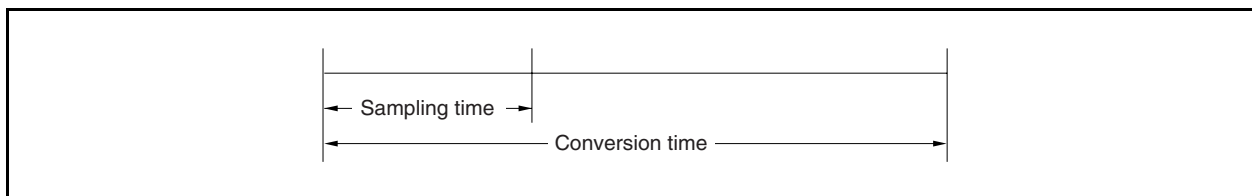
(8) Conversion time

This is the time required to obtain a digital output after sampling has started. The conversion time in the characteristics table includes the sampling time.

(9) Sampling time

This is the time for which the analog switch is ON to load an analog voltage to the sample & hold circuit.

Figure 13-22. Relationship Between Conversion Time and Sampling Time



CHAPTER 14 D/A CONVERTER (V850ES/JC3-L (48-pin), V850ES/JE3-L)

14.1 Functions

In the V850ES/JC3-L (48-pin), V850ES/JE3-L, one R-2R ladder type D/A converter channel is provided (DA0CS0). The D/A converter has the following functions.

- 8-bit resolution × 1 channel
- R-2R ladder method
- Conversion time: 3 μ s (MAX.) (when AV_{REF1} = 2.7 to 3.6 V, external load = 20 pF)
- Analog output voltage: $AV_{REF1} \times m/256$ (m = 0 to 255; value set to DA0CS0 register)
- Operation modes: Normal mode, real-time output mode

14.2 Configuration

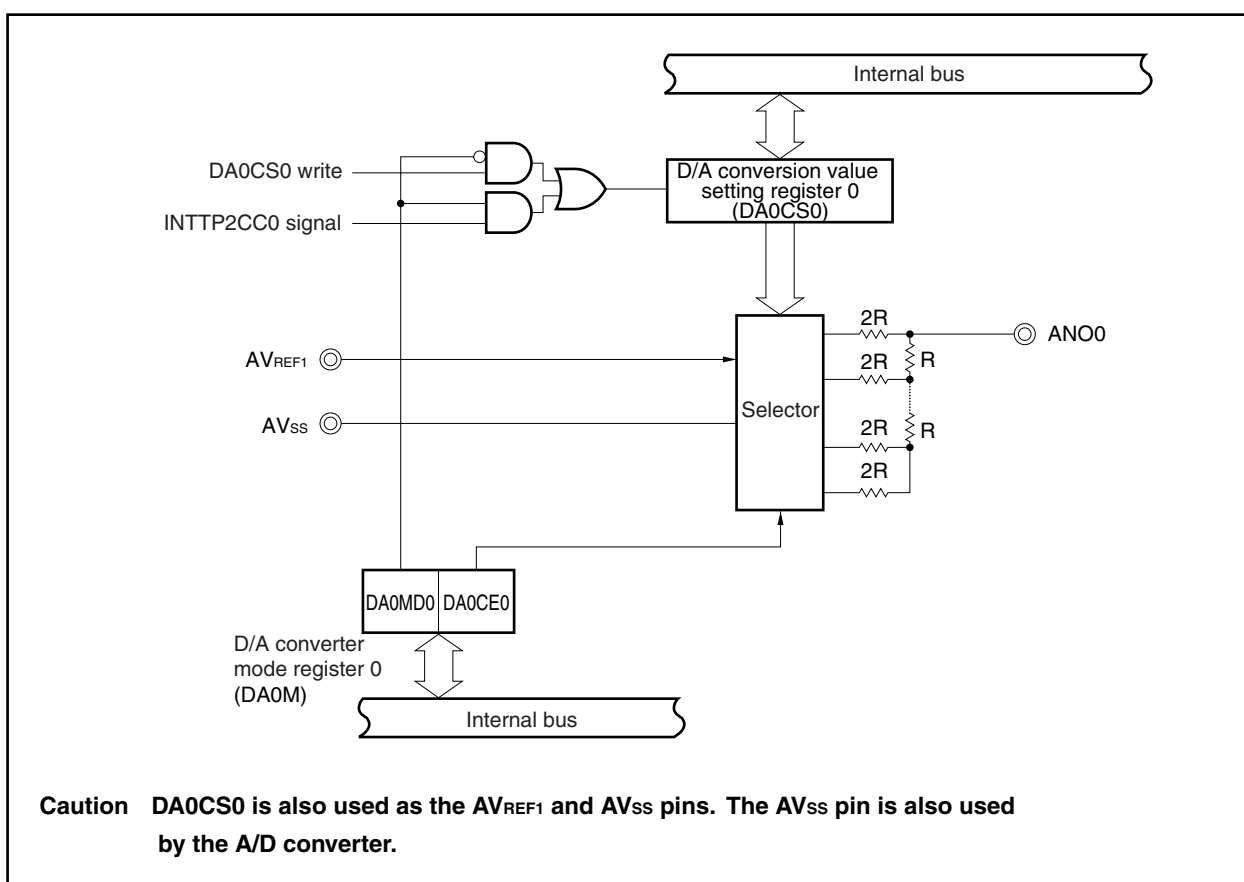
The D/A converter includes the following hardware.

Table 14-1. D/A Converter Registers Used by Software

Item	Configuration
Control registers	D/A converter mode register (DA0M) D/A conversion value setting registers 0 (DA0CS0)

The block diagram of the D/A converter is shown below.

Figure 14-1. Block Diagram of D/A Converter



14.3 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 (DA0CS0)

(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 sets this register to 00H.

After reset: 00H R/W Address: FFFFF282H

	7	6	5	<4>	3	2	1	0
DA0M	0	0	0	DA0CE0	0	0	0	DA0MD0

DA0CE0	D/A converter operation enable/disable
0	Disables operation
1	Enables operation

DA0MD0	Selection of D/A converter operation mode
0	Normal mode
1	Real-time output mode ^{Note}

Note The output trigger in the real-time output mode (DA0MD0 bit = 1) is as follow.

- INTTP2CC0 signal (see **CHAPTER 6 16-BIT TIMER/EVENT COUNTER P (TMP)**)

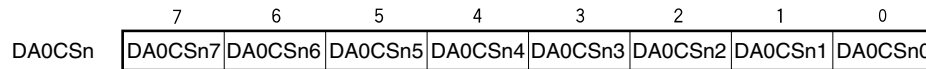
(2) D/A conversion value setting registers 0 (DA0CS0)

The DA0CS0 register sets the analog voltage value output to the ANO0 pin.

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

Reset sets these registers to 00H.

After reset: 00H R/W Address: DA0CS0 FFFFF280H, DA0CS1 FFFFF281H



Caution In the real-time output mode (DA0M.DA0MD0 bit = 1), set the DA0CS0 register before the INTTP2CC0 signal is generated. D/A conversion starts when the INTTP2CC0 signal is generated.

14.4 Operation

14.4.1 Operation in normal mode

D/A conversion is performed using a write operation to the DA0CS0 register as the trigger.

The setting method is described below.

- <1> Set the PM10 bit to 1 (input mode).
- <2> Clear the DA0M.DA0MD0 bit to 0 (normal mode).
- <3> Set the analog voltage value to be output to the ANO0 pin to the DA0CS0 register.
Steps <2> and <3> above constitute the initial settings.
- <4> Set the DA0M.DA0CE0 bit to 1 (D/A conversion enable).
The D/A converted analog voltage value is output from the ANO0 pin when this setting is performed.
- <5> To change the analog voltage value, write to the DA0CS0 register.
The analog voltage value set immediately before is held until the next write operation is performed.

Remark For the alternate-function pin settings, refer to **Table 4-17 Settings When Pins Are Used for Alternate Functions**.

14.4.2 Operation in real-time output mode

D/A conversion is performed using the interrupt request signal of TMP2 (INTTP2CC0) as the trigger.

The setting method is described below.

- <1> Set the PM10 bit to 1 (input mode).
- <2> Set the DA0M.DA0MD0 bit to 1 (real-time output mode).
- <3> Set the analog voltage value to be output to the ANO0 pin to the DA0CS0 register.
- <4> Set the DA0M.DA0CE0 bit to 1 (D/A conversion enable).
Steps <2> to <4> above constitute the initial settings.
- <5> Operate TMP2.
- <6> The D/A converted analog voltage value is output from the ANOn pin when the INTTP2CC0 signal is generated.
Set the analog voltage value to be output to the DA0CS0 register next, before the next INTTP2CC0 signal is generated.
- <7> After that, the value set to the DA0CS0 register is output from the ANO0 pin every time the INTTP2CC0 signal is generated.

Remarks 1. The output value of the ANO0 pin up to <6> above is undefined.

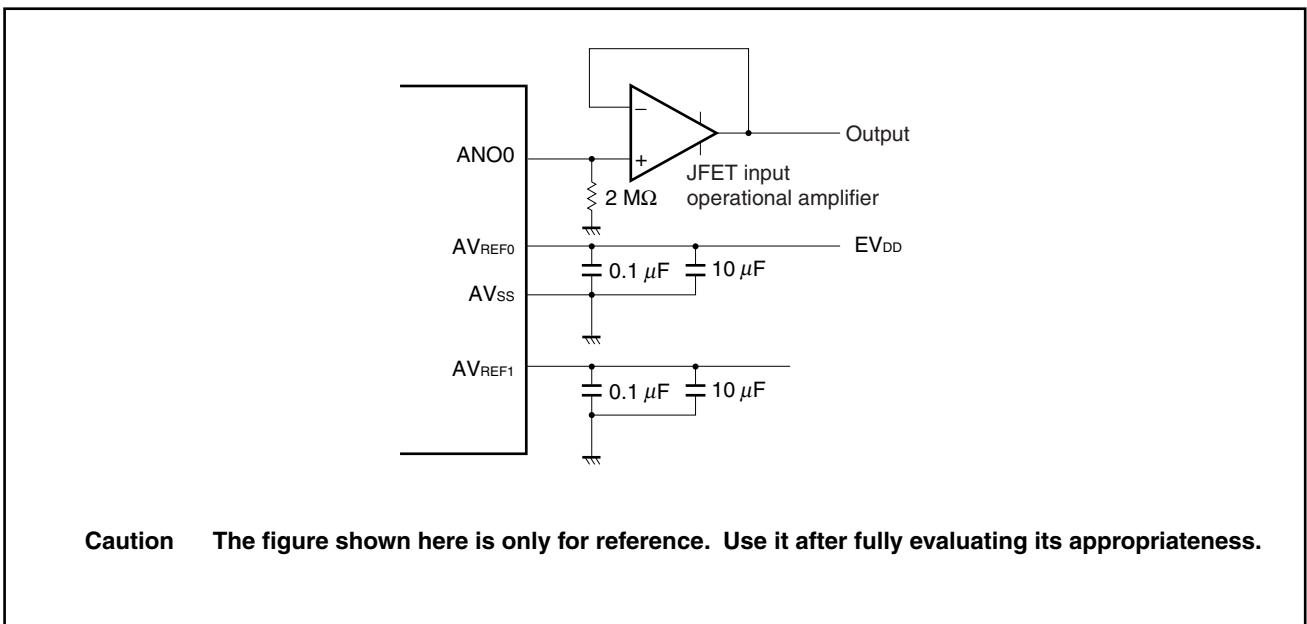
- 2.** For the output value of the ANO0 pin in the IDLE1, IDLE2, HALT, and STOP modes, refer to **CHAPTER 21 STANDBY FUNCTION**.

14.4.3 Cautions

Observe the following cautions when using the D/A converter.

- (1) Set the port pins to the input mode (PM10 bit = 1).
- (2) Do not read or write the P1 register during D/A conversion.
- (3) In the real-time output mode, avoid a conflict between writing the DA0CS0 register by software and trigger signal output, which can occur by writing the DA0CS0 register while the interrupt requested by the selected trigger signal is being serviced.
- (4) Make sure that $AV_{REF1} \leq V_{DD}$ and $AV_{REF1} = 2.7$ to 3.6 V. The operation is not guaranteed if ranges other than the above are used.
- (5) Turn on or off AV_{REF1} at the same time as turning on or off AV_{REF0} .
- (6) Because the output impedance of the D/A converter is high, a current cannot be supplied from the ANO0 pin. When connecting a resistor of $2\text{ M}\Omega$ or lower, take appropriate measures such as inserting a JFET input type operational amplifier between the resistor and the ANO0 pin.

Figure 14-2. Example of External Pin Connection



- (7) Because the D/A converter stops operating in the STOP mode, the ANO0 pin goes into a high-impedance state, and the power consumption can be reduced.

In the IDLE1, IDLE2, or subclock operation mode, however, the D/A converter continues operating. To lower the power consumption, therefore, clear the DA0M.DA0CE0 bit to 0.

CHAPTER 15 ASYNCHRONOUS SERIAL INTERFACE A (UARTA)

The number of UARTA channels in the V850ES/JC3-L, V850ES/JE3-L are shown below.

Product Name	V850ES/JC3-L (40-pin)	V850ES/JC3-L (48-pin)	V850ES/JE3-L
Number of channels	2 channels UARTA0, UARTA1	3 channels UARTA0 to UARTA2	3 channels UARTA0 to UARTA2

In this chapter, the number of channels is expressed as n.

15.1 Features

- On-chip dedicated baud rate generator
- Transfer rate: 300 bps to 625 kbps (using dedicated baud rate generator)
- Full-duplex communication
- Double buffer configuration Internal UARTAn receive data register (UAnRX)
Internal UARTAn transmit data register (UAnTX)
- Reception error detection function
 - Parity error
 - Framing error
 - Overrun error
- Interrupt sources: 2
 - Reception complete interrupt (INTUAnR): This interrupt occurs upon transfer of receive data from the receive shift register to the receive data register after serial transfer completion, in the reception enabled status.
 - Transmission enable interrupt (INTUAnT): This interrupt occurs upon transfer of transmit data from the transmit data register to the transmit shift register in the transmission enabled status. (Continuous transmission is possible.)
- Character length: 7, 8 bits
- Parity function: Odd, even, 0, none
- Transmission stop bit: 1, 2 bits
- MSB-/LSB-first transfer selectable
- Internal digital noise filter
- Inverted input/output of transmit/receive data possible
- SBF (Sync Break Field) transmission/reception in the LIN (Local Interconnect Network) communication format
 - 13 to 20 bits selectable for SBF transmission
 - Recognition of 11 bits or more possible for SBF reception
 - SBF reception flag provided

15.2 Configuration

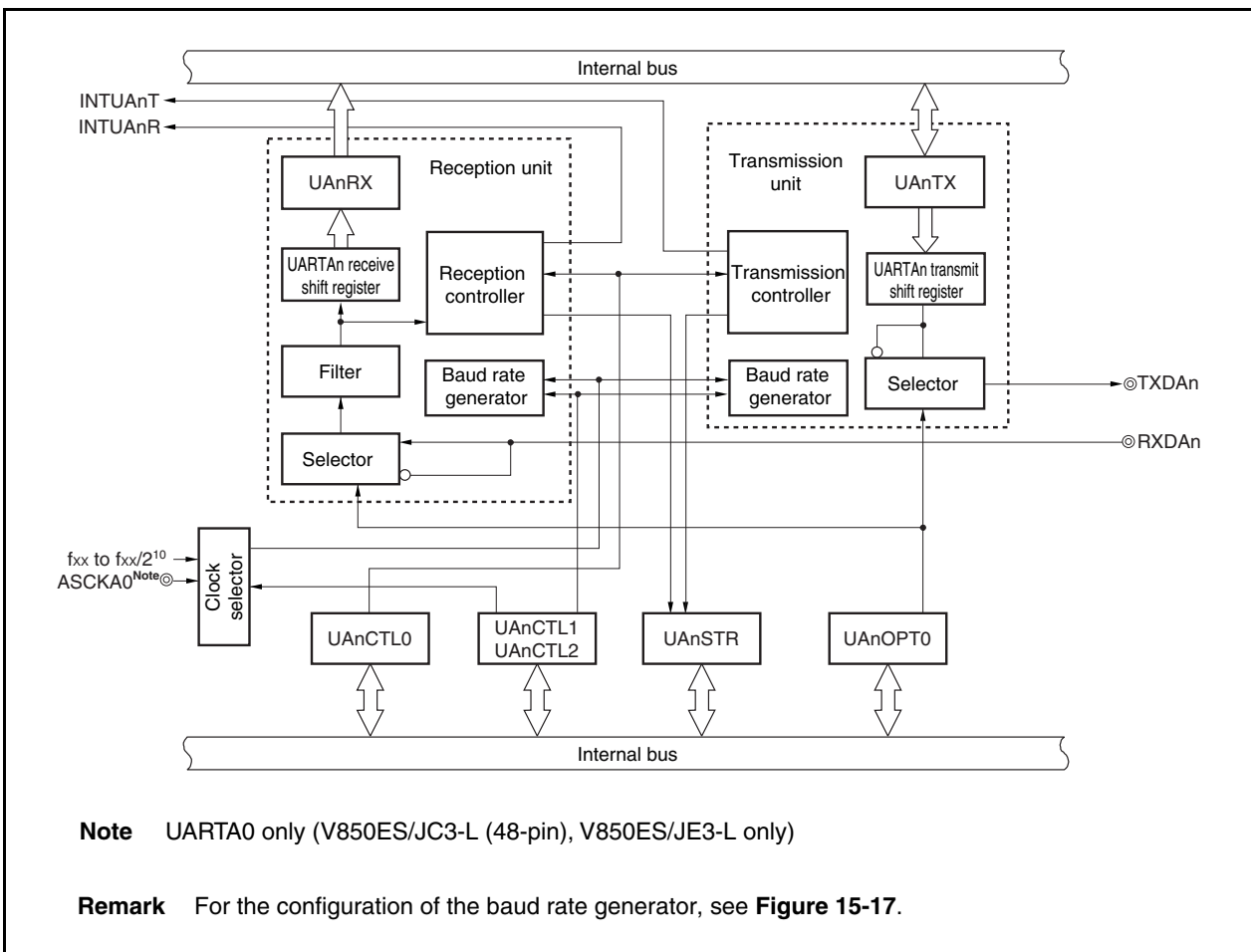
UARTAn includes the following hardware.

Table 15-1. 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)

The block diagram of the UARTAn is shown below.

Figure 15-1. Block Diagram of UARTAn



(1) UARTAn control register 0 (UAnCTL0)

The UAnCTL0 register is an 8-bit register used to specify the operation of UARTAn.

(2) UARTAn control register 1 (UAnCTL1)

The UAnCTL1 register is an 8-bit register used to select the base clock (f_{CLK}) for UARTAn.

(3) UARTAn control register 2 (UAnCTL2)

The UAnCTL2 register is an 8-bit register used with the UAnCTL1 register to generate the baud rate for UARTAn.

(4) UARTAn option control register 0 (UAnOPT0)

The UAnOPT0 register is an 8-bit register used to control SBF transmission/reception in the LIN communication format and the level of the transmission/reception signals for the UARTAn.

(5) UARTAn status register (UAnSTR)

The UAnSTR register is an 8-bit register that indicates the contents of a reception error. Each one of the reception error flags is set (to 1) upon the occurrence of a reception error.

(6) UARTAn receive shift register

This is a shift register used to convert the serial data input to the RXDAn pin into parallel data. Upon reception of 1-character 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 buffer register that holds receive data.

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 character.

Transfer to the UAnRX register also causes the reception complete interrupt request signal (INTUAnR) to be output.

(8) UARTAn transmit shift register

The transmit shift register is a shift register used to convert the parallel data transferred from the UAnTX register into serial data.

When 1-character data is transferred from the UAnTX register, the shift register data is output from the TXDAn pin.

This register cannot be manipulated directly.

(9) UARTAn transmit data register (UAnTX)

The UAnTX register is an 8-bit transmit data buffer. Transmission starts when transmit data is written to the UAnTX register. When data can be written to the UAnTX register (when 1-character data is transferred from the UAnTX register to the UARTAn transmit shift register), the transmission enable interrupt request signal (INTUAnT) is generated.

15.2.1 Pin functions of each channel

The RXDAn, TXDAn, and ASCKA0 pins used by UARTA in the V850ES/JC3-L, V850ES/JE3-L are used for other functions as shown in Table 15-2. To use these pins for UARTA, set the related registers as described in **Table 4-17 Settings When Pins Are Used for Alternate Functions**.

Table 15-2. Pins Used by UARTA

Channel	Pin No.			Port	UARTA Reception Input	UARTA Transmission Output	UARTA Clock I/O ^{Note}	Other Functions
	JC3L (40)	JC3L (48)	JE3L					
UARTA0	–	34	46	P31	RXDA0	–	–	INTP7/SIB4
	30	–	–					INTP7
	–	33	45	P30	–	TXDA0	–	SOB4
	29	–	–					–
–	35	47	P32	–	–	ASCKA0 ^{Note}	$\overline{\text{SCKB4}}$ /TIP00/TOP00	
UARTA1	20	24	32	P91	RXDA1	–	–	KR7/SCL02
	19	23	31	P90	–	TXDA1	–	KR6/SDA02
UARTA2	–	19	27	P39	RXDA2	–	–	SCL00
	–	18	26	P38	–	TXDA2	–	SDA00

Note The ASCKA0 function is provided only for UARTA0 (V850ES/JC3-L (48-pin), V850ES/JE3-L only).

Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

15.3 Mode Switching of UARTA and Other Serial Interfaces

15.3.1 UARTA0 and CSIB4 mode switching

In the V850ES/JC3-L (48-pin), V850ES/JE3-L), UARTA0 and CSIB4 share pins and therefore cannot be used simultaneously. To use the UARTA0 function, specify the UARTA0 mode in advance by using the PMC3, PFC3, and PFCE3 registers.

Switching the operation mode between UARTA0 and CSIB4 is described below.

Caution Transmission and reception by UARTA0 and CSIB4 are not guaranteed if these operation modes are switched during transmission or reception. Be sure to stop the serial interface that is not being used.

Figure 15-2. Switching UARTA0 and CSIB4 Operation Modes

After reset: 0000H R/W Address: FFFFF446H, FFFFF447H																							
PMC3	15	14	13	12	11	10	9	8															
	0	0	0	0	0	0	PMC39	PMC38															
	7	6	5	4	3	2	1	0															
	0	0	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30															
After reset: 0000H R/W Address: FFFFF466H, FFFFF467H																							
PFC3	15	14	13	12	11	10	9	8															
	0	0	0	0	0	0	PFC39	PFC38															
	7	6	5	4	3	2	1	0															
	0	0	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30															
After reset: 00H R/W Address: FFFFF706H																							
PFCE3L	7	6	5	4	3	2	1	0															
	0	0	0	0	0	PFCE32	0	0															
<table border="1"> <thead> <tr> <th>PMC32</th><th>PFCE32</th><th>PFC32</th><th>Operation mode</th></tr> </thead> <tbody> <tr> <td>0</td><td>×</td><td>×</td><td>Port I/O mode</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>ASCKA0 mode</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>SCKB4 mode</td></tr> </tbody> </table>				PMC32	PFCE32	PFC32	Operation mode	0	×	×	Port I/O mode	1	0	0	ASCKA0 mode	1	0	1	SCKB4 mode				
PMC32	PFCE32	PFC32	Operation mode																				
0	×	×	Port I/O mode																				
1	0	0	ASCKA0 mode																				
1	0	1	SCKB4 mode																				
<table border="1"> <thead> <tr> <th>PMC3m</th><th>PFC3m</th><th>Operation mode</th></tr> </thead> <tbody> <tr> <td>0</td><td>×</td><td>Port I/O mode</td></tr> <tr> <td>1</td><td>0</td><td>UARTA0 mode</td></tr> <tr> <td>1</td><td>1</td><td>CSIB4 mode</td></tr> </tbody> </table>				PMC3m	PFC3m	Operation mode	0	×	Port I/O mode	1	0	UARTA0 mode	1	1	CSIB4 mode								
PMC3m	PFC3m	Operation mode																					
0	×	Port I/O mode																					
1	0	UARTA0 mode																					
1	1	CSIB4 mode																					
<p>Remarks 1. m = 0, 1 2. × = don't care</p>																							

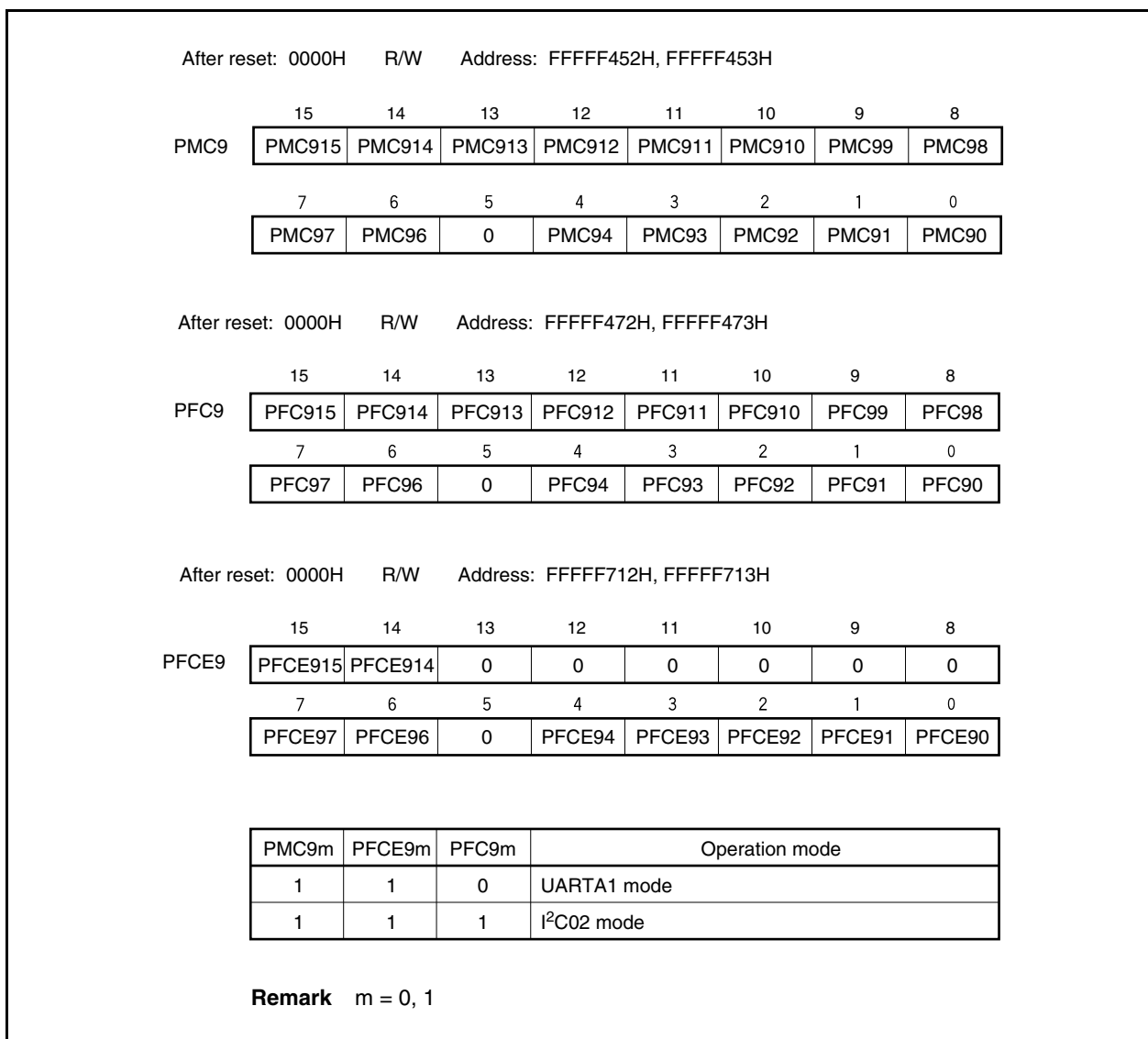
15.3.2 UARTA1 and I²C02 mode switching

In the V850ES/JC3-L (48-pin), V850ES/JE3-L, UARTA1 and I²C02 share pins and therefore cannot be used simultaneously. To use the UARTA1 function, specify the UARTA1 mode in advance by using the PMC9, PFC9, and PFCE9 registers.

Switching the operation mode between UARTA1 and I²C02 are described below.

Caution Transmission and reception by UARTA1 and I²C02 are not guaranteed if these operation modes are switched during transmission or reception. Be sure to stop the serial interface that is not being used.

Figure 15-3. Switching UARTA1 and I²C02 Operation Modes



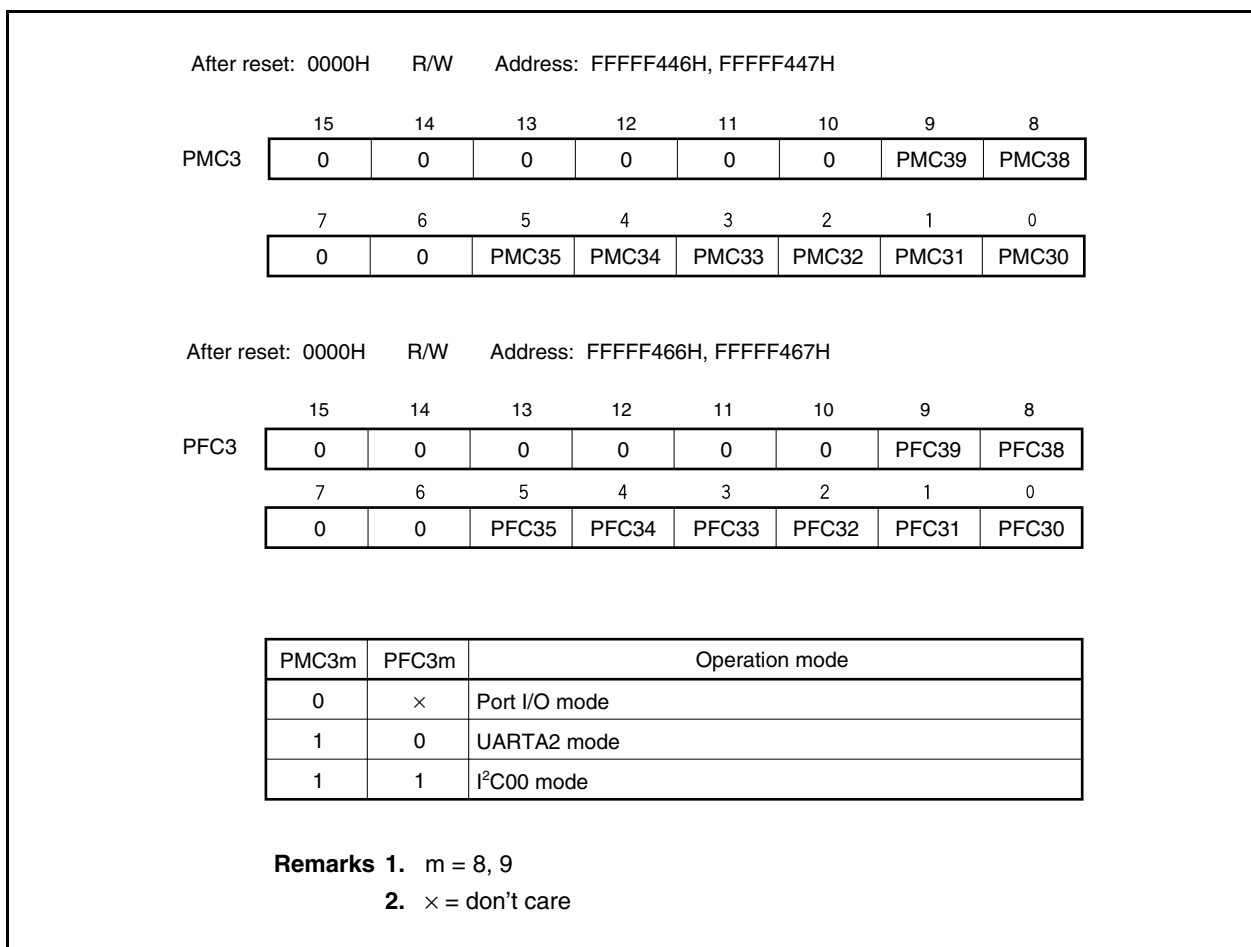
15.3.3 UARTA2 and I²C00 mode switching

In the V850ES/JC3-L (48-pin), V850ES/JE3-L, UARTA2 and I²C00 share pins and therefore cannot be used simultaneously. To use the UARTA2 function, specify the UARTA2 mode in advance by using the PMC3 and PFC3 registers.

Switching the operation mode between UARTA2 and I²C00 are described below.

Caution Transmission and reception by UARTA2 and I²C00 are not guaranteed if these operation modes are switched during transmission or reception. Be sure to stop the serial interface that is not being used.

Figure 15-4. Switching UARTA2 and I²C00 Operation Modes



15.4 Registers

(1) UARTAn control register 0 (UAnCTL0)

The UAnCTL0 register is an 8-bit register that controls the UARTAn serial transfer operation.

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

Reset sets this register to 10H.

(1/2)

After reset: 10H R/W Address: UA0CTL0 FFFFA00H, UA1CTL0 FFFFA10H, UA2CTL0 FFFFA20H								
UAnCTL0	<7>	<6>	<5>	<4>	3	2	1	0
	UAnPWR	UAnTXE	UAnRXE	UAnDIR	UAnPS1	UAnPS0	UAnCL	UAnSL
	UAnPWR	UARTAn operation control						
	0	Disable UARTAn operation (UARTAn reset asynchronously)						
	1	Enable UARTAn operation						
	The UARTAn operation is controlled by the UAnPWR bit. The TXDAn pin output is fixed to high level by clearing the UAnPWR bit to 0 (fixed to low level if UAnOPT0.UAnTDL bit = 1).							
	UAnTXE	Transmission operation enable						
	0	Disable transmission operation						
	1	Enable transmission operation						
	<ul style="list-style-type: none"> • To start transmission, set the UAnPWR bit to 1 and then set the UAnTXE bit to 1. To stop transmission, clear the UAnTXE bit to 0 and then UAnPWR bit to 0. • To initialize the transmission unit, clear the UAnTXE bit to 0, wait for two cycles of the base clock, and then set the UAnTXE bit to 1 again. Otherwise, initialization may not be executed (for the base clock, see 15.7 (1) (a) Base clock). • When UARTAn operation is enabled (UAnPWR bit = 1) and the UAnTXE bit is set to 1, transmission is enabled after at least two cycles of the base clock (f_{uclk}) have elapsed. 							
	UAnRXE	Reception operation enable						
	0	Disable reception operation						
	1	Enable reception operation						
	<ul style="list-style-type: none"> • To start reception, set the UAnPWR bit to 1 and then set the UAnRXE bit to 1. To stop reception, clear the UAnRXE bit to 0 and then UAnPWR bit to 0. • 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 15.7 (1) (a) Base clock). • When UARTAn operation is enabled (UAnPWR bit = 1) and the UAnRXE bit is set to 1, reception is enabled after at least two cycles of the base clock (f_{uclk}) have elapsed. If a start bit is received before reception is enabled, the start bit is ignored. 							

(2/2)

UAnDIR	Data transfer order
0	MSB first
1	LSB first

- This register can be rewritten only when the UAnPWR bit is 0 or the UAnTXE bit and the UAnRXE bit are 0.
- When transmission and reception are performed in the LIN format, set the UAnDIR bit to 1.

UAnPS1	UAnPS0	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

- This register is rewritten only when the UAnPWR bit is 0 or the UAnTXE bit and the UAnRXE bit are 0.
- If “Reception with 0 parity” is selected during reception, a parity check is not performed. Therefore, the UAnSTR.UAnPE bit is not set.
- When transmission and reception are performed in the LIN format, clear the UAnPS1 and UAnPS0 bits to 00.

UAnCL	Specification of data character length of 1 frame of transmit/receive data
0	7 bits
1	8 bits

- This register can be rewritten only when the UAnPWR bit is 0 or the UAnTXE bit and the UAnRXE bit are 0.
- When transmission and reception are performed in the LIN format, set the UAnCL bit to 1.

UAnSL	Specification of length of stop bit for transmit data
0	1 bit
1	2 bits

This register can be rewritten only when the UAnPWR bit is 0 or the UAnTXE bit and the UAnRXE bit are 0.

Remark For details of parity, see **15.6.6 Parity types and operations**.

(2) UARTAn control register 1 (UAnCTL1)

For details, see **15.7 (2) UARTAn control register 1 (UAnCTL1)**.

(3) UARTAn control register 2 (UAnCTL2)

For details, see **15.7 (3) UARTAn control register 2 (UAnCTL2)**.

(4) UARTAn option control register 0 (UAnOPT0)

The UAnOPT0 register is an 8-bit register used to control SBF transmission/reception in the LIN communication format and the level of the transmission/reception signals for the UARTAn.

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

Reset sets this register to 14H.

(1/2)

After reset: 14H R/W Address: UA0OPT0 FFFFFFFA03H, UA1OPT0 FFFFFFFA13H,
UA2OPT0 FFFFFFFA23H

	<7>	6	5	4	3	2	1	0
UAnOPT0	UAnSRF	UAnSRT	UAnSTT	UAnSLS2	UAnSLS1	UAnSLS0	UAnTDL	UAnRDL

UAnSRF	SBF reception flag
0	The UAnCTL0.UAnPWR bit or the UAnCTL0.UAnRXE bit is set to 0, or SBF reception ends normally.
1	During SBF reception
<ul style="list-style-type: none"> • This bit indicates whether SBF (Sync Brake Field) is received in LIN communication. • When an SBF reception error occurs, the UAnSRF bit remains 1 and SBF reception is started again. • The UAnSRF bit is a read-only bit. 	

UAnSRT	SBF reception trigger
0	-
1	SBF reception trigger
<ul style="list-style-type: none"> • This is the SBF reception trigger bit during LIN communication, and is always 0 when read. • For SBF reception, set the UAnSRT bit (to 1) to enable SBF reception. • Set the UAnSRT bit after setting the UAnPWR bit and UAnRXE bit to 1. 	

UAnSTT	SBF transmission trigger
0	-
1	SBF transmission trigger
<ul style="list-style-type: none"> • This is the SBF transmission trigger bit during LIN communication, and is always 0 when read. • Setting this bit to 1 triggers SBF transmission. • Set the UAnSTT bit after setting the UAnPWR bit and UAnTXE bit to 1. 	

Caution Do not set the UAnSRT and UAnSTT bits (to 1) during SBF reception (UAnSRF bit = 1).

(2/2)

UAnSLS2	UAnSLS1	UAnSLS0	SBF transmit length selection
1	0	1	13-bit output (initial value)
1	1	0	14-bit output
1	1	1	15-bit output
0	0	0	16-bit output
0	0	1	17-bit output
0	1	0	18-bit output
0	1	1	19-bit output
1	0	0	20-bit output

This register can be set when the UAnPWR bit or the UAnTXE bit is 0.

UAnTDL	Transmit data level bit
0	Normal output of transfer data
1	Inverted output of transfer data

- The output level of the TXDAn pin can be inverted by using the UAnTDL bit.
- This register can be set when the UAnPWR bit or the UAnTXE bit is 0.

UAnRDL	Receive data level bit
0	Normal input of transfer data
1	Inverted input of transfer data

- The input level of the RXDAn pin can be inverted by using the UAnRDL bit.
- This register can be set when the UAnPWR bit or the UAnRXE bit is 0.

(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 be both 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 previous value is retained).

The conditions for clearing the UAnSTR register are shown below.

Table 15-3. Conditions for Clearing STR Register

Register/Bit	Conditions for Clearing
UAnSTR register	<ul style="list-style-type: none"> • Reset • UAnCTL0.UAnPWR = 0
UAnTSF bit	<ul style="list-style-type: none"> • UAnCTL0.UAnTXE = 0
UAnPE, UAnFE, UAnOVE bits	<ul style="list-style-type: none"> • 0 write • UAnCTL0.UAnRXE = 0

After reset: 00H R/W Address: UA0STR FFFFA04H, UA1STR FFFFA14H,
UA2STR FFFFA24H

	<7>	6	5	4	3	<2>	<1>	<0>
UAnSTR	UAnTSF	0	0	0	0	UAnPE	UAnFE	UAnOVE

UAnTSF	Transfer status flag
0	The transmit shift register does not have data. <ul style="list-style-type: none"> • When the UAnPWR bit or the UAnTXE bit has been set to 0. • When, following transfer completion, there was no next data transfer from UAnTX register
1	The transmit shift register has data. (Write to UAnTX register)
<p>The UAnTSF bit is always 1 when performing continuous transmission. When initializing the transmission unit, check that the UAnTSF bit is 0 before performing initialization. The transmit data is not guaranteed when initialization is performed while the UAnTSF bit is 1.</p>	

UAnPE	Parity error flag
0	<ul style="list-style-type: none"> • When the UAnPWR bit or the UAnRXE bit has been set to 0. • When 0 has been written
1	The received parity bit does not match the specified parity.
<ul style="list-style-type: none"> • The operation of the UAnPE bit is controlled by the settings of the UAnCTL0.UAnPS1 and UAnCTL0.UAnPS0 bits. • Once the UAnPE bit is set (1), the value is retained until the bit is cleared (0). • The UAnPE bit can be read and written, but it can only be cleared by writing 0 to it; it cannot be set by writing 1 to it. When 1 is written to this bit, the previous value is retained. 	

UAnFE	Framing error flag
0	<ul style="list-style-type: none"> • When the UAnPWR bit or the UAnRXE bit has been set to 0 • When 0 has been written
1	When no stop bit is detected during reception
<ul style="list-style-type: none"> • Only the first bit of the receive data stop bits is checked, regardless of the value of the UAnCTL0.UAnSL bit. • Once the UAnFE bit is set (1), the value is retained until the bit is cleared (0). • The UAnFE bit can be both read and written, but it can only be cleared by writing 0 to it; it cannot be set by writing 1 to it. When 1 is written to this bit, the previous value is retained. 	

UAnOVE	Overrun error flag
0	<ul style="list-style-type: none"> • When the UAnPWR bit or the UAnRXE bit has been set to 0. • When 0 has been written
1	When receive data has been set to the UAnRX register and the next receive operation is completed before that receive data has been read
<ul style="list-style-type: none"> • When an overrun error occurs, the data is discarded without the next receive data being written to the receive buffer. • Once the UAnOVE bit is set (1), the value is retained until the bit is cleared (0). • The UAnOVE bit can be both read and written, but it can only be cleared by writing 0 to it; it cannot be set by writing 1 to it. When 1 is written to this bit, the previous 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 receive shift register.

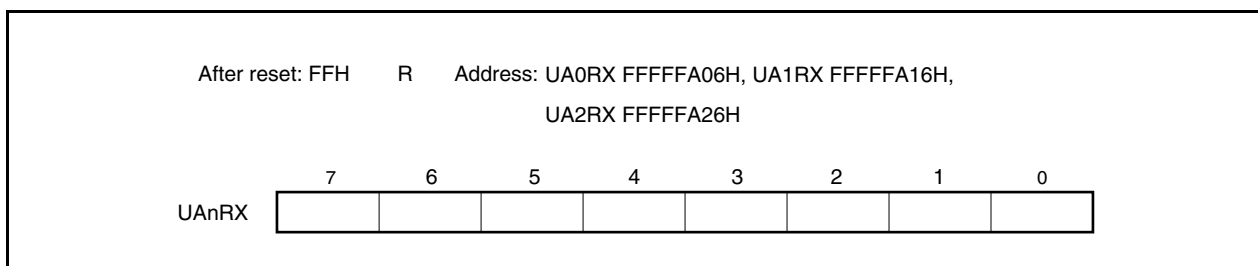
The data stored in the receive shift register is transferred to the UAnRX register upon completion of reception of 1-character data.

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 (UAnOVE) occurs, 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.



(7) UARTAn transmit data register (UAnTX)

The UAnTX register is an 8-bit register used to set transmit data.

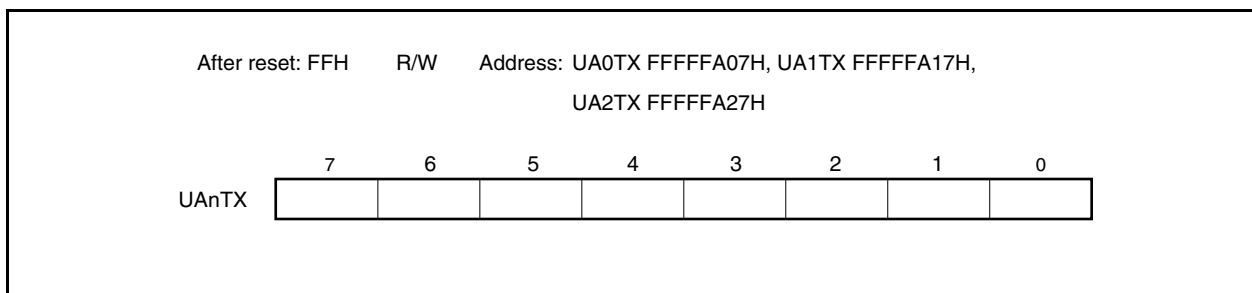
Writing transmit data to the UAnTX register with transmission enabled (UAnCTL0.UAnTXE bit = 1) triggers transmission. When transfer of the UAnTX register data to the UARTAn transmit shift register is complete, the transmission enable interrupt request signal (INTUAnT) is generated.

When 7-bit data is transmitted LSB-first, the data is transferred to bits 6 to 0 of the UAnTX register. When the data is transmitted MSB-first, it is transferred to bits 7 to 1 of the UAnTX register.

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

Reset sets this register to FFH.

Caution Writing the UAnTX register with transmission enabled (UAnPWR bit = 1 and UAnTXE bit = 1) triggers transmission. If the same value as the one immediately before is written, therefore, the same data is transmitted twice. To write new transmit data during processing of the preceding one, wait until the transmission enable interrupt request signal (INTUAnT) has been generated. Even if transmission is enabled after data is written to the UAnTX register with transmission disabled (UAnPWR bit = 0 or UAnTXE bit = 0), transmission does not start.



15.5 Interrupt Request Signals

The following two interrupt request signals are generated from UARTAn.

- Reception complete interrupt request signal (INTUAnR)
- Transmission enable interrupt request signal (INTUAnT)

The default priority for these two interrupt request signals is reception complete interrupt request signal then transmission enable interrupt request signal.

Table 15-4. Interrupts and Their Default Priorities

Interrupt Request Signal	Priority
Reception complete	High
Transmission enable	Low

(1) Reception complete interrupt request signal (INTUAnR)

When the data stored in the receive shift register is transferred to the UAnRX register with reception enabled, the reception complete interrupt request signal is generated.

A reception complete interrupt request signal is also output when a reception error occurs. Therefore, when a reception complete interrupt request signal is acknowledged and the data is read, read the UAnSTR register and check that the reception result is not an error.

No reception complete interrupt request signal is generated in the reception disabled status.

(2) Transmission enable interrupt request signal (INTUAnT)

If transmit data is transferred from the UAnTX register to the UARTAn transmit shift register with transmission enabled, the transmission enable interrupt request signal is generated.

15.6 Operation

15.6.1 Data format

As shown in Figure 15-5, one frame of transmit/receive data consists of a start bit, character bits, parity bit, and stop bit(s).

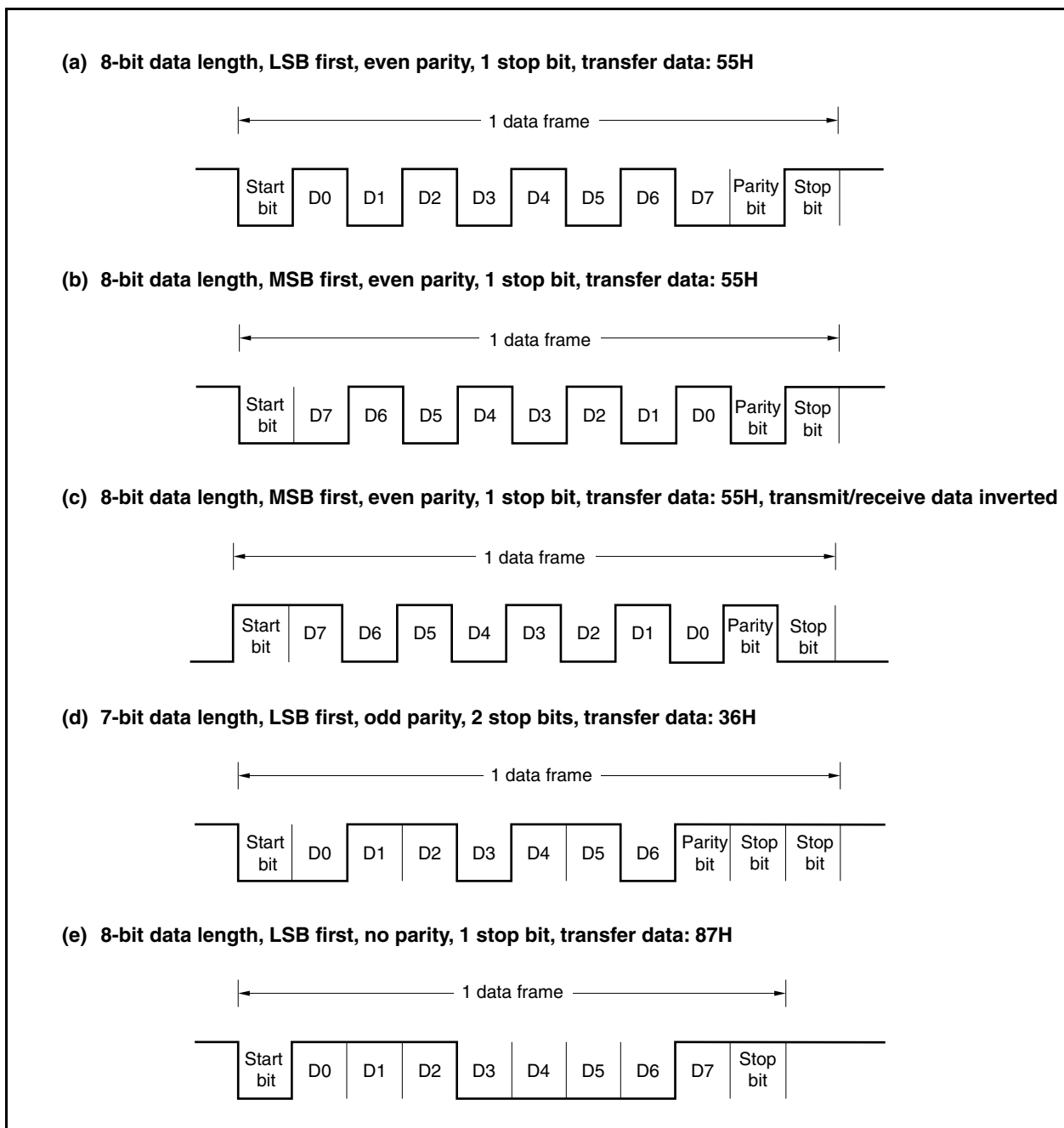
Specification of the character bit length within 1 data frame, parity selection, specification of the stop bit length, and specification of MSB-first/LSB-first transfer are performed using the UAnCTL0 register.

The UAnOPT0.UAnTDL bit is used to specify normal output/inverted output for the data to be transferred via the TXDAn pin.

The UAnOPT0.UAnRDL bit is used to specify normal input/inverted input for the data to be received via the RXDAn pin.

- 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
- Input logic Normal input/inverted input
- Output logic Normal output/inverted output
- Communication direction MSB/LSB

Figure 15-5. UARTA Transmit/Receive Data Format



15.6.2 UART transmission

Transmission is enabled by setting the UAnCTL0.UAnPWR and UAnCTL0.UAnTXE bits 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.

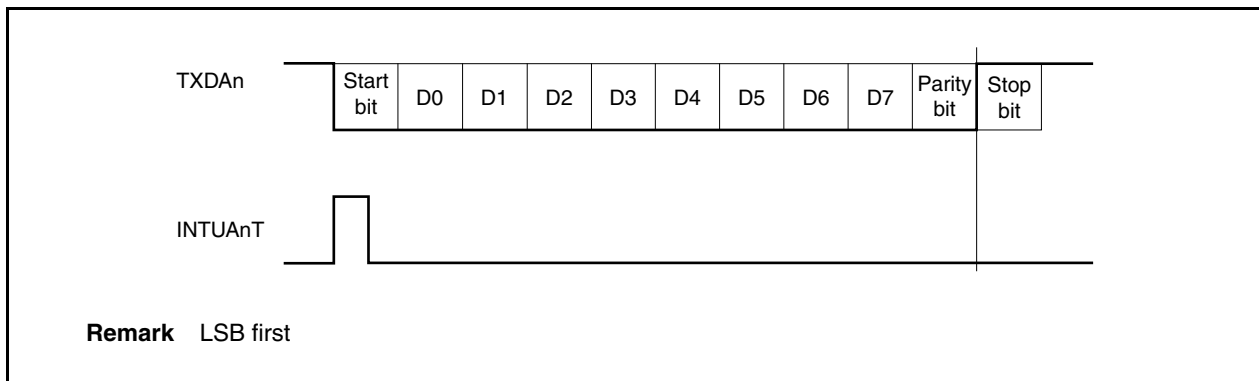
Since the CTS (transmit enable signal) input pin is not provided in UARTAn, use a port to check that reception is enabled at the transmit destination.

The data in the UAnTX register is transferred to the UARTAn transmit shift register upon the start of transmission.

A transmission enable interrupt request signal (INTUAnT) is generated upon completion of transmission of the data of the UAnTX register to the UARTAn transmit shift register, and the contents of the UARTAn transmit shift register are output to the TXDAn pin.

Writing the next transmit data to the UAnTX register is enabled after the INTUAnT signal is generated.

Figure 15-6. UART Transmission



15.6.3 Continuous transmission procedure

Writing transmit data to the UAnTX register with transmission enabled triggers transmission. The data in the UAnTX register is transferred to the UARTAn transmit shift register, the transmission enable interrupt request signal (INTUAnT) is generated, and then shifting is started. After the transmission enable interrupt request signal (INTUAnT) is generated, the next transmit data can be written to the UAnTX register. The timing of UARTAn transmit shift register transmission can be judged from the transmission enable interrupt request signal (INTUAnT).

An efficient communication rate is realized by writing the data to be transmitted next to the UAnTX register during transfer.

Caution When initializing transmission during the execution of continuous transmission, make sure that the UAnSTR.UAnTSF bit is 0, then perform initialization. Transmit data that is initialized when the UAnTSF bit is 1 cannot be guaranteed.

Figure 15-7. Continuous Transmission Processing

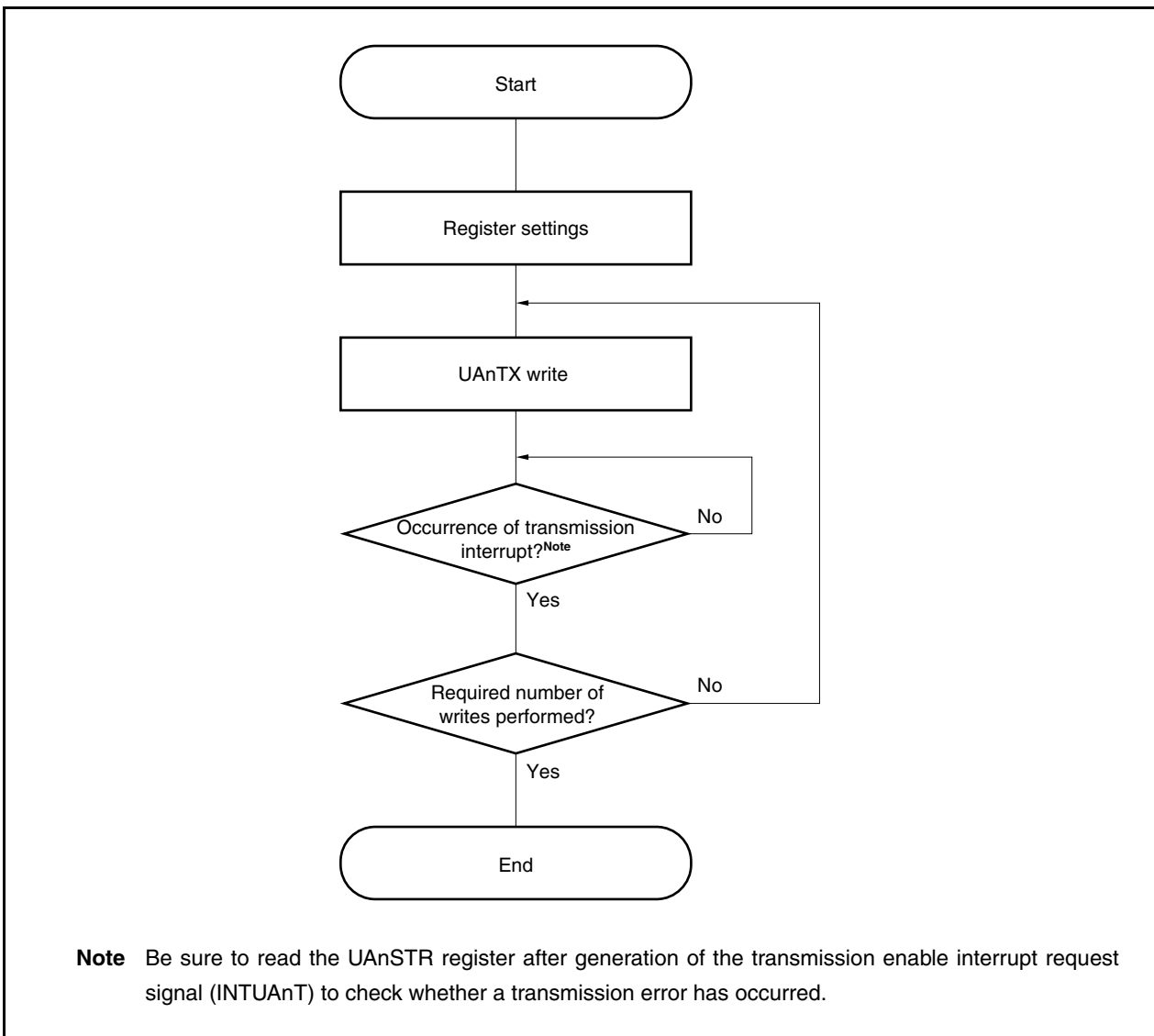
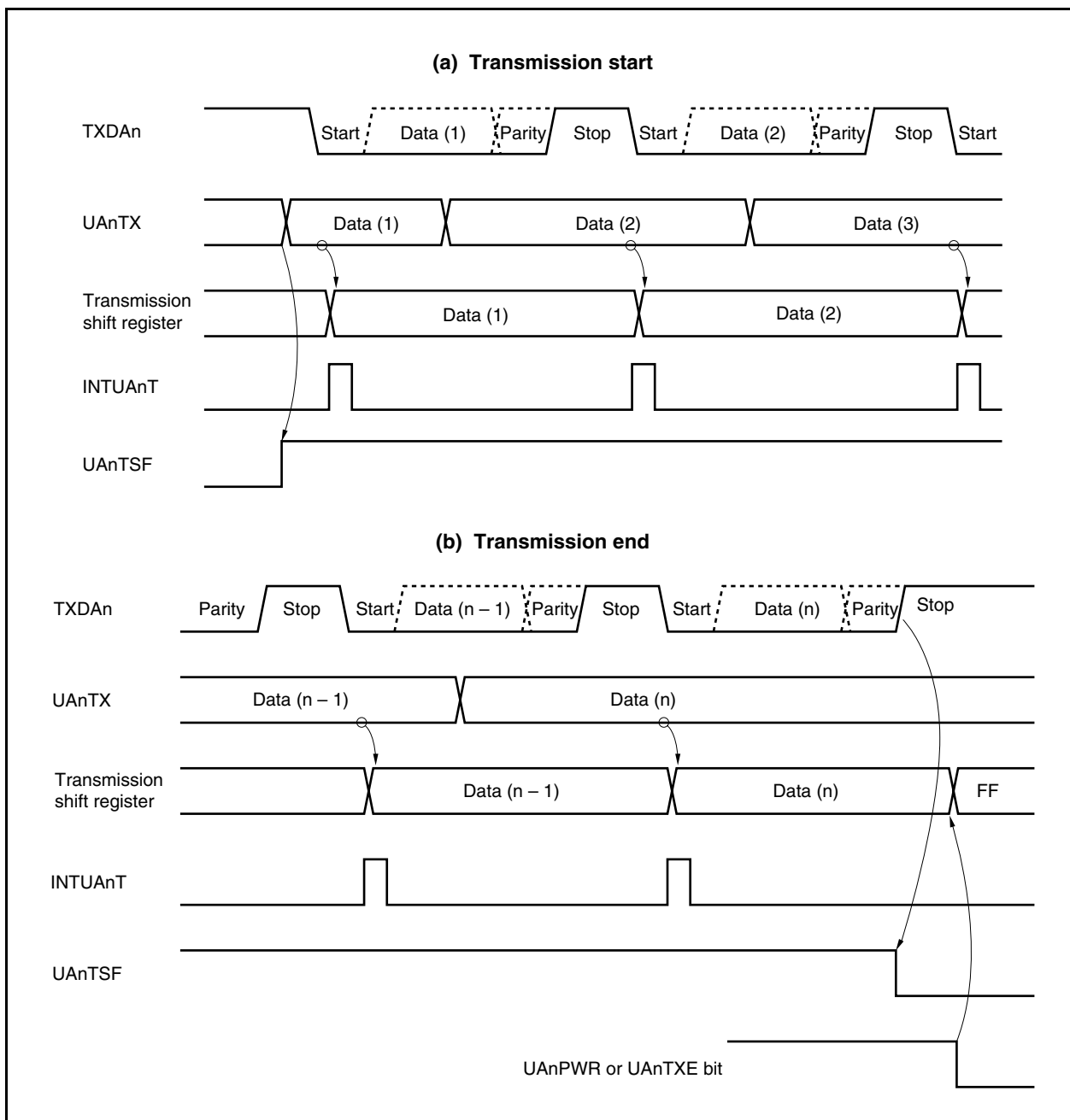


Figure 15-8. Continuous Transmission Operation Timing



15.6.4 UART reception

First, enable reception by executing the following operations and monitor the RXDAn input to detect the start bit.

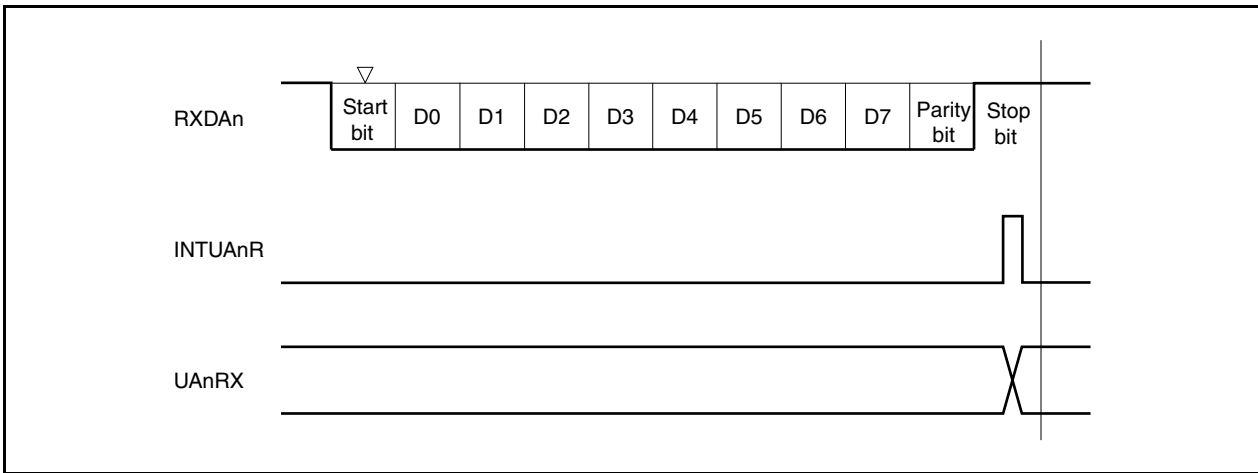
- Specify the operating clock by using UARTA control register 1 (UAnCTL1).
- Specify the baud rate by using UARTA control register 2 (UAnCTL2).
- Specify the output logic level by using UARTA option control register 0 (UAnOPT0).
- Specify the communication direction, parity, data character length, and stop bit length by using UARTA control register 0 (UAnCTL0).
- Set the power bit and reception enable bit (UAnPWR = 1 and UAnRXE = 1).

To change the communication direction, parity, data character length, and/or stop bit length, clear the power bit (UAnPWR = 0) or clear both the transmission enable bit and reception enable bit (UAnTXE = 0 and UAnRXE = 0) beforehand.

The level input to the RXDAn pin is sampled by using the operating clock. If the falling edge is detected, sampling of data input to RXDAn is started. If the data is low level half a bit after detection of the falling edge (indicated by ▽ in Figure 15-9), it is recognized as a start bit. When the start bit has been recognized, reception is started, and serial data is sequentially stored in the receive shift register at the specified baud rate. When the stop bit has been received, the reception complete interrupt request signal (INTUAnR) is generated and, at the same time, the data stored in the receive shift register is transferred to the receive data register (UAnRX).

If an overrun error occurs (UAnOVE = 1), however, the receive data is not transferred to UAnRX, but is discarded. On the other hand, even if a parity error (UAnPE = 1) or framing error (UAnFE = 1) occurs, reception continues and the receive data is transferred to the UAnRX register. No matter which reception error has occurred, the INTUAnR interrupt is generated after reception is complete.

Figure 15-9. UART Reception



- 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. 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 complete interrupt request signal (INTUAnR) has been generated, and clear the UAnRXE bit to 0. If the UAnRXE bit is cleared to 0 before the INTUAnR signal is generated, the read value of the UAnRX register cannot be guaranteed.**
 - 4. If the receive completion processing (INTUAnR signal generation) of UARTAn conflicts with setting the UAnPWR bit or UAnRXE bit to 0, the INTUAnR signal may be generated in spite of there being no data stored in the UAnRX register.**

To complete reception without waiting for INTUAnR signal generation, be sure to clear (0) the interrupt request flag (UAnRIF) of the UAnRIC register, after setting (1) the interrupt mask flag (UAnRMK) of the interrupt control register (UAnRIC) and then set (1) the UAnPWR bit or UAnRXE bit to 0.

15.6.5 Reception errors

Three types of errors can occur during reception: parity errors, framing errors, and overrun errors. The data reception result error flag is set in the UAnSTR register and a reception complete interrupt request signal (INTUAnR) 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.

Figure 15-10. Reading Receive Data

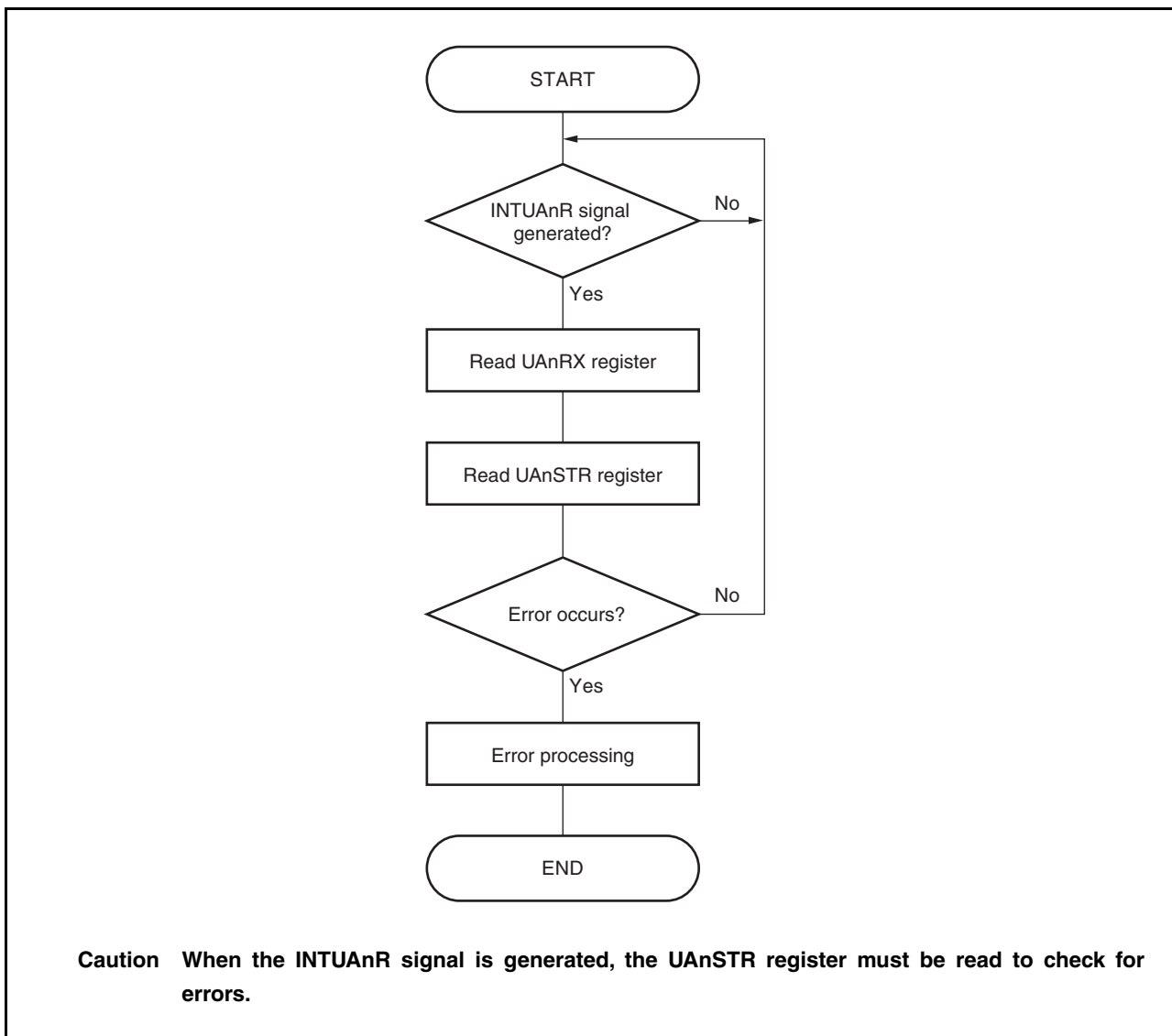


Table 15-5. Reception Error Causes

Error Flag	Reception Error	Cause
UAnPE	Parity error	The received parity bit does not match the setting.
UAnFE	Framing error	The stop bit was not detected.
UAnOVE	Overrun error	Reception of the next data was completed before data was read from the receive buffer.

When a reception error occurs, perform the following procedure according to the kind of error.

- Parity error
If false data is received due to problems such as noise on the reception line, discard the received data and retransmit.
- Framing error
A baud rate error may have occurred between the reception side and transmission side or a start bit may have been erroneously detected. Since this is a fatal error for the communication format, check that operation on the transmission side has stopped, initialize both sides, and then start the communication again.
- Overrun error
1 frame of data is discarded because the next reception is completed before data was read from the receive buffer. If this data was needed, retransmit the data.

Caution In reception, be sure to read the UAnSTR register before completion of the next reception to check whether an error has occurred. If an error has occurred, perform error processing.

15.6.6 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, a 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.

Caution When using the LIN function, fix the UAnPS1 and UAnPS0 bits of the UAnCTL0 register to 0, 0.

15.6.7 LIN transmission/reception format

The V850ES/JC3-L, V850ES/JE3-L have an SBF (Sync Break Field) transmission/reception control function to enable use of the LIN function.

Remark LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to reduce costs of automotive networks.

LIN communication is single-master communication, and up to 15 slaves can be connected to the master. The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.

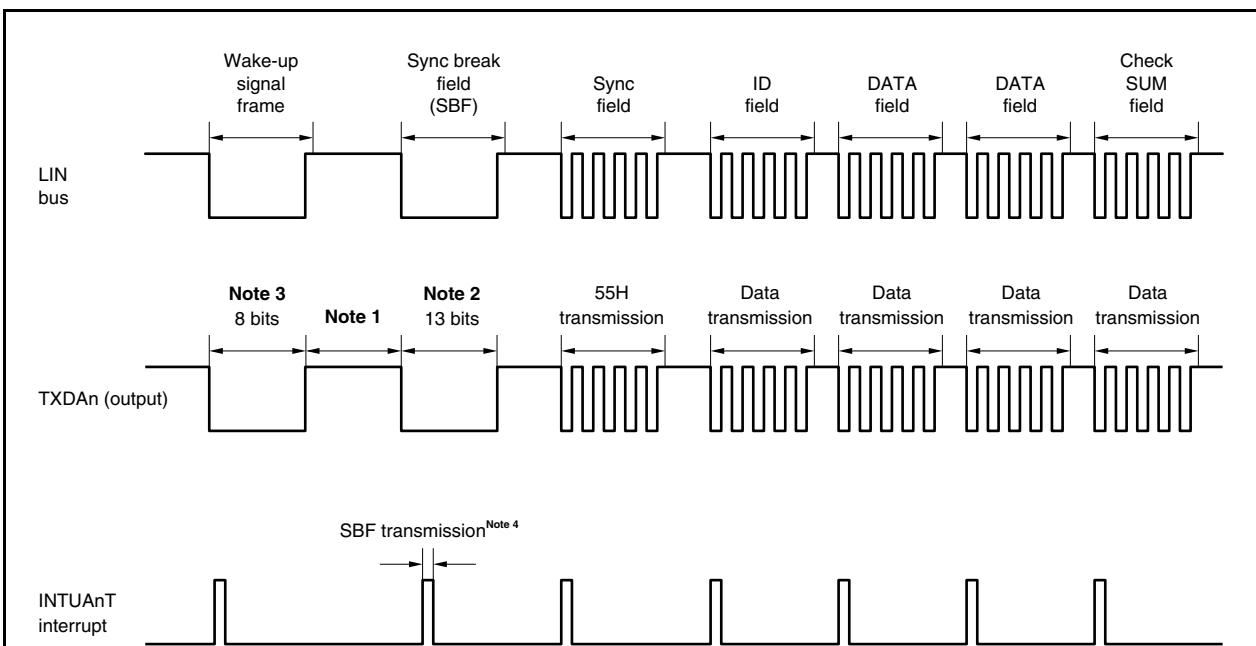
Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.

In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is $\pm 15\%$ or less.

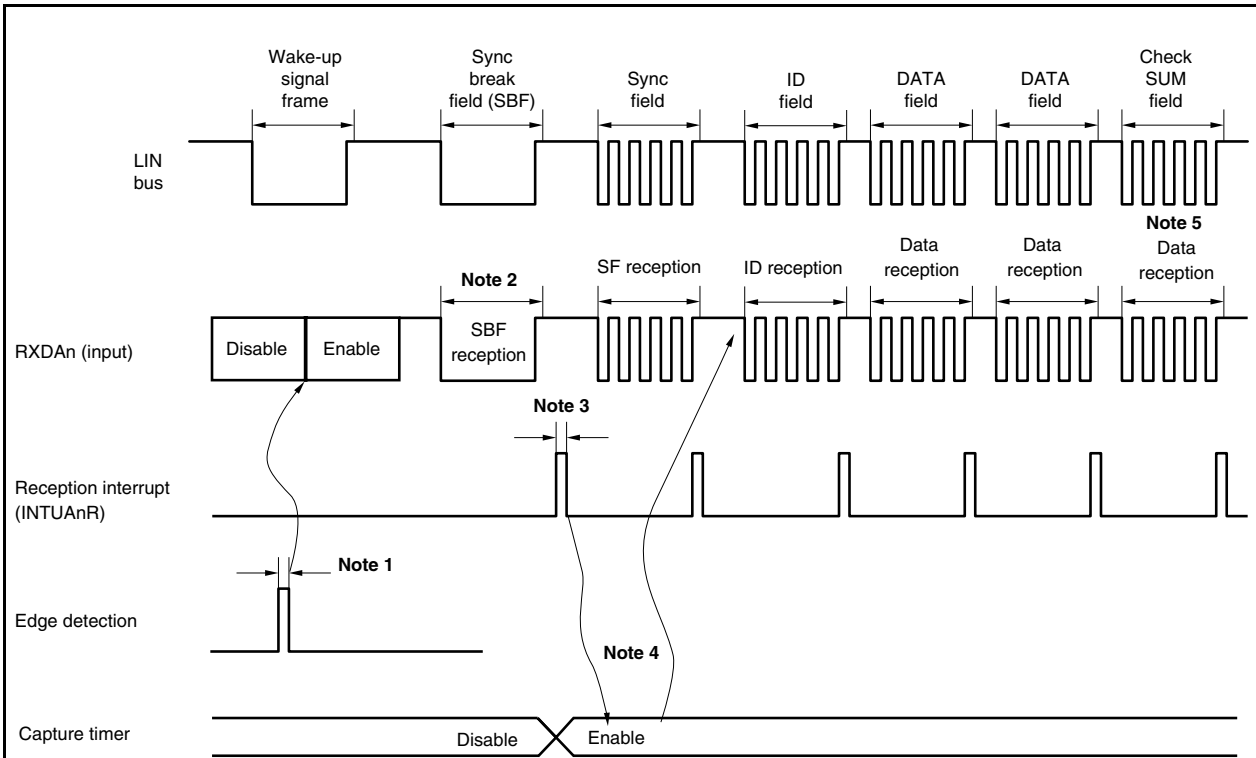
Figures 15-11 and 15-12 outline the transmission and reception manipulations of LIN.

Figure 15-11. LIN Transmission Format



- Notes**
1. The interval between each field is controlled by software.
 2. SBF output is performed by hardware. The output width is the bit length set by the UAnOPT0.UAnSBL2 to UAnOPT0.UAnSBL0 bits. If even finer output width adjustments are required, such adjustments can be performed using the UAnCTLn.UAnBRS7 to UAnCTLn.UAnBRS0 bits.
 3. 80H transfer in the 8-bit mode is substituted for the wakeup signal frame.
 4. A transmission enable interrupt request signal (INTUAnT) is output at the start of each transmission. The INTUAnT signal is also output at the start of each SBF transmission.

Figure 15-12. LIN Reception Format



- Notes 1.** The wakeup signal is detected by the pin edge detector, UARTAn is enabled, and the SBF reception mode is set.
- 2.** Reception is performed until detection of the stop bit. Upon detection of SBF reception of 11 or more bits, it is judged as normal SBF reception end, and an interrupt signal is output. Upon detection of SBF reception of less than 11 bits, it is judged as an SBF reception error, no interrupt signal is output, and the mode returns to the SBF reception mode.
- 3.** If SBF reception ends normally, an interrupt request signal is output. The timer is enabled by an SBF reception complete interrupt. Moreover, error detection for the UAnSTR.UAnOVE, UAnSTR.UAnPE, and UAnSTR.UAnFE bits is suppressed and UART communication error detection processing and data transfer of the UARTAn receive shift register and UAnRX register is not performed. The UARTAn receive shift register holds the initial value, FFH.
- 4.** The RXDAn pin is connected to T1 (capture input) of the timer and the transfer rate is calculated. The value of the UAnCTL2 register obtained by correcting the baud rate error after UARTA enable goes low is set again, causing the status to become the reception status.
- 5.** A check-sum field is identified by software. UARTAn is initialized following reception of the check-sum field, and the processing for re-specifying the SBF reception mode is performed, also by software.

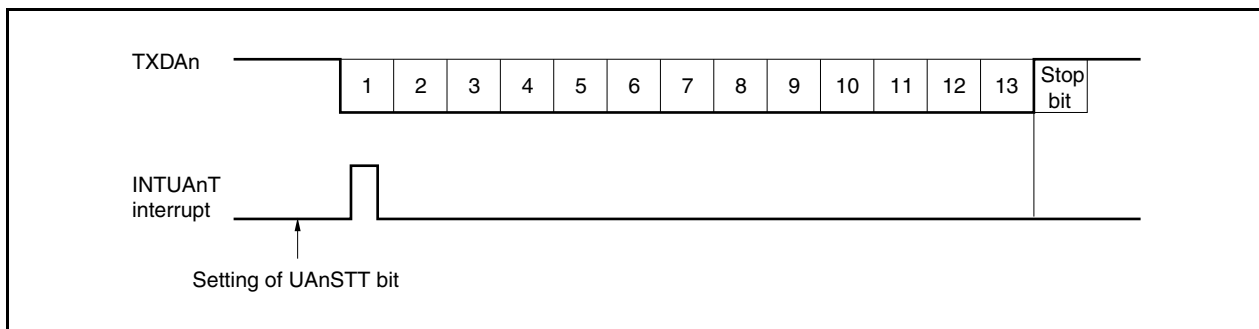
15.6.8 SBF transmission

When the UAnCTL0.UAnPWR bit and UAnCTL0.UAnTXE bit are 1, the transmission enabled status is entered, and SBF transmission is started by setting the SBF transmission trigger (UAnOPT0.UAnSTT bit) to 1.

Thereafter, a low level signal having a length of 13 to 20 bits, as specified by the UAnOPT0.UAnSLS2 to AnOPT0.UAnSLS0 bits, is output. A transmission enable interrupt request signal (INTUAnT) is generated upon the start of SBF transmission. Following the end of SBF transmission, the UAnSTT bit is automatically cleared.

Transmission is suspended until the data to be transmitted next is written to the UAnTX register, or until the SBF transmission trigger (UAnSTT bit) is set.

Figure 15-13. Example of SBF Transmission



15.6.9 SBF reception

The reception enabled status is entered by setting the UAnCTL0.UAnPWR bit to 1 and then setting the UAnCTL0.UAnRXE bit to 1.

The SBF reception wait status is set by setting the SBF reception trigger (UAnOPT0.UAnSTR bit) to 1.

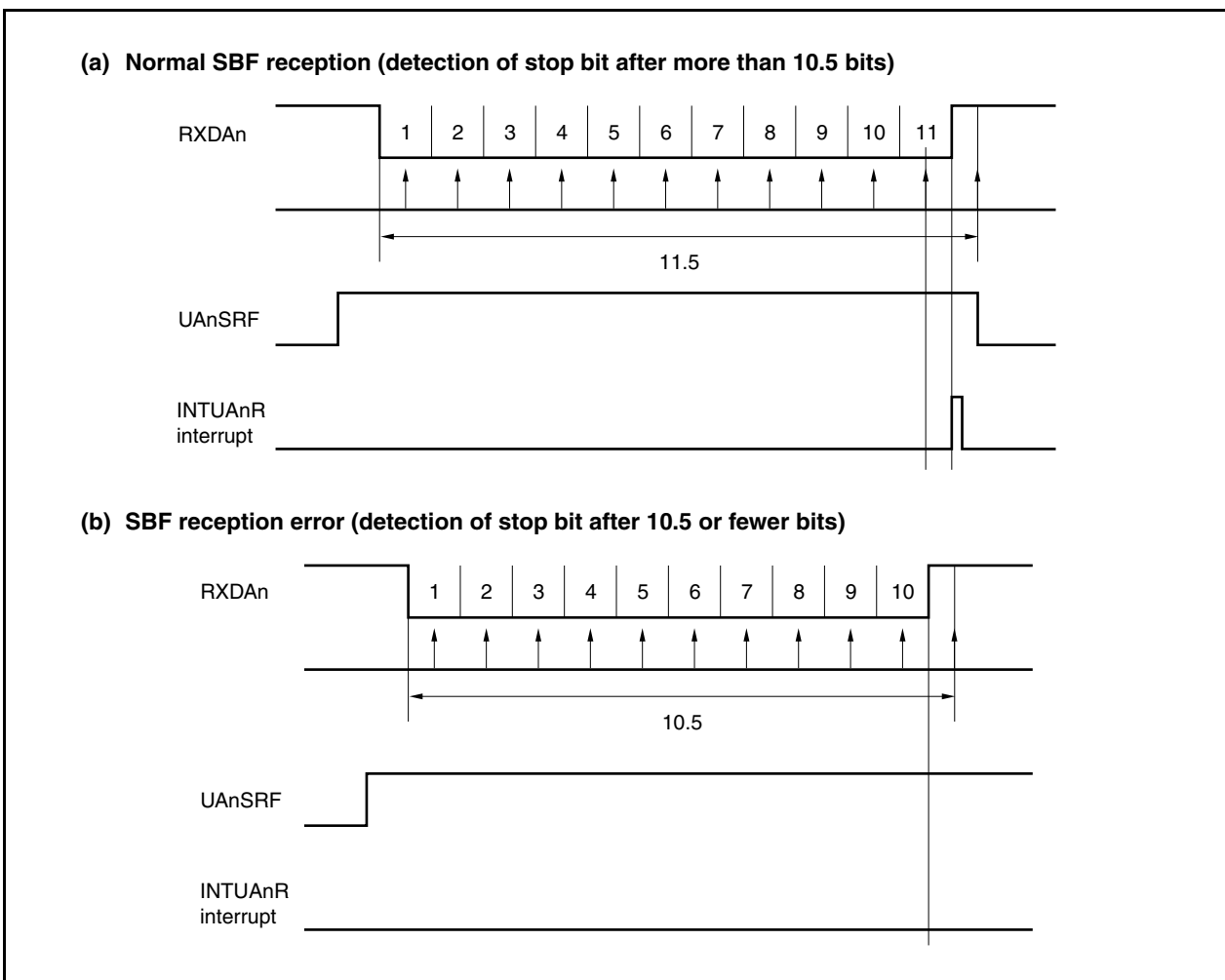
In the SBF reception wait status, similarly to the UART reception wait status, the RXDAn pin is monitored and start bit detection is performed.

Following detection of the start bit, reception is started and the internal counter increments according to the set baud rate.

When a stop bit is received, if the SBF width is 11 or more bits, it is judged as normal processing and a reception complete interrupt request signal (INTUAnR) is output. The UAnOPT0.UAnSRF bit is automatically cleared and SBF reception ends. Error detection for the UAnSTR.UAnOVE, UAnSTR.UAnPE, and UAnSTR.UAnFE bits is suppressed and UART communication error detection processing is not performed. Moreover, data transfer of the UARTAn reception shift register and UAnRX register is not performed and FFH, the initial value, is held. If the SBF width is 10 or fewer bits, reception is terminated as an error, an interrupt is not generated, and the SBF reception mode is restored. The UAnSRF bit is not cleared at this time.

- Cautions 1. If SBF is transmitted during data reception, a framing error occurs.**
- 2. Do not set the SBF reception trigger bit (UAnSRT) and SBF transmission trigger bit (UAnSTT) to 1 during SBF reception (UAnSRF = 1).**

Figure 15-14. SBF Reception



15.6.10 Receive data noise filter

This filter samples signals received via the RXDAn pin using the base clock supplied by the dedicated baud rate generator.

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 1 clock cycle width is judged to be noise and is not delivered to the internal circuit (see Figure 15-16). See 15.7 (1) (a) Base clock for details of the base clock.

Moreover, since the circuit is as shown in Figure 15-15, the processing that goes on within the receive operation is delayed by 3 clocks in relation to the external signal status.

Figure 15-15. Noise Filter Circuit

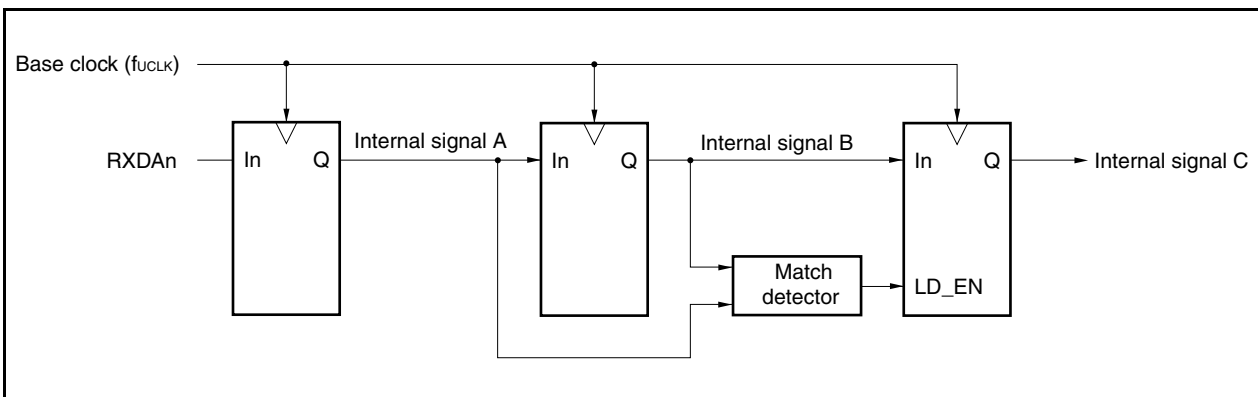
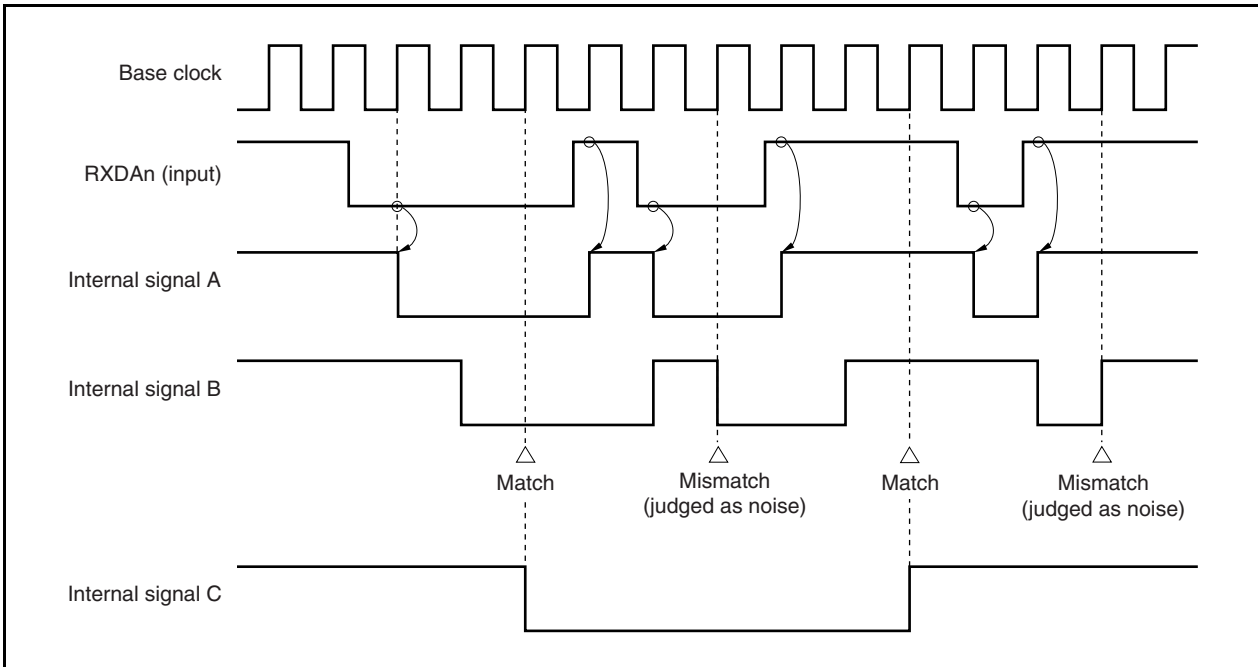


Figure 15-16. Timing of RXDAn Signal Judged as Noise



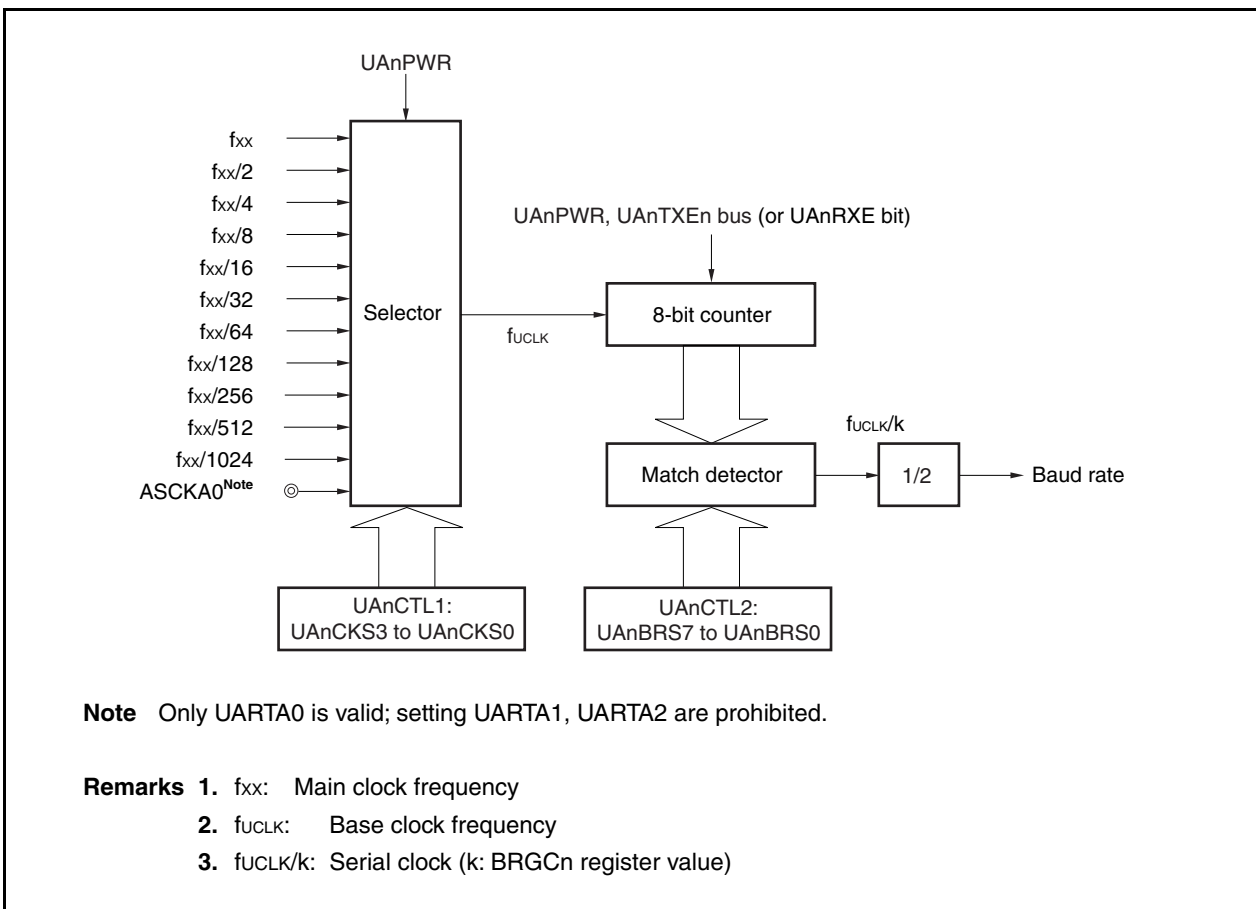
15.7 Dedicated Baud Rate Generator

The dedicated baud rate generator consists of a source clock selector block and an 8-bit programmable counter block, and generates a serial clock during transmission and reception using 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 15-17. Configuration of Baud Rate Generator



(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 (fUCLK).

(b) Serial clock generation

A serial clock can be generated by setting the UAnCTL1 register and the UAnCTL2 register (n = 0 to 2).

The base clock 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.

The baud rate clock is generated by dividing the serial clock by two.

(2) UARTAn control register 1 (UAnCTL1)

The UAnCTL1 register is an 8-bit register that selects the UARTAn base clock.

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

Reset sets this register to 00H.

Caution Clear the UAnCTL0.UAnPWR bit to 0 before rewriting the UAnCTL1 register.

After reset: 00H R/W Address: UA0CTL1 FFFFFFFA01H, UA1CTL1 FFFFFFFA11H,
UA2CTL1 FFFFFFFA21H

	7	6	5	4	3	2	1	0
UAnCTL1	0	0	0	0	UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0

UAnCKS3	UAnCKS2	UAnCKS1	UAnCKS0	Base clock (f _{CLK}) selection
0	0	0	0	fxx
0	0	0	1	fxx/2
0	0	1	0	fxx/4
0	0	1	1	fxx/8
0	1	0	0	fxx/16
0	1	0	1	fxx/32
0	1	1	0	fxx/64
0	1	1	1	fxx/128
1	0	0	0	fxx/256
1	0	0	1	fxx/512
1	0	1	0	fxx/1,024
1	0	1	1	External clock ^{Note} (ASCKA0 pin)
Other than above				Setting prohibited

Note Only UARTA0 is valid; setting UARTA1, UART2, V850ES/JC3-L (40-pin) are prohibited.

Remark fxx: Main 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. The baud rate clock is generated by dividing the serial clock specified by this register by two. This register can be read or written in 8-bit units. Reset sets this register to FFH.

Caution Either clear the UAnCTL0.UAnPWR bit to 0, or clear the UAnTXE and UAnRXE bits to 0, 0, before rewriting the UAnCTL2 register.

After reset FFH R/W Address: UA0CTL2 FFFFFFFA02H, UA1CTL2 FFFFFFFA12H,
UA2CTL2 FFFFFFFA22H

UAnCTL2	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}: Clock frequency selected by the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits

(4) Baud rate

The baud rate is obtained by the following equation.

$$\text{Baud rate} = \frac{f_{\text{UCLK}}}{2 \times k} \text{ [bps]}$$

When using the internal clock, the equation will be as follows (when using the ASCKA0 pin input as the clock for UARTA0, calculate using the above equation).

$$\text{Baud rate} = \frac{f_{\text{XX}}}{2^{m+1} \times k} \text{ [bps]}$$

Remark f_{UCLK} = Frequency of base clock selected by the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits
 f_{XX} : Main clock frequency
 m = Value set using the UAnCTL1.UAnCKS3 to UAnCTL1.UAnCKS0 bits ($m = 0$ to 10)
 k = Value set using the UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits ($k = 4$ to 255)

The baud rate error is obtained by the following equation.

$$\begin{aligned} \text{Error (\%)} &= \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Target baud rate (correct baud rate)}} - 1 \right) \times 100 \text{ [\%]} \\ &= \left(\frac{f_{\text{UCLK}}}{2 \times k \times \text{Target baud rate}} - 1 \right) \times 100 \text{ [\%]} \end{aligned}$$

When using the internal clock, the equation will be as follows (when using the ASCKA0 pin input as the clock for UARTA0, calculate the baud rate error using the above equation).

$$\text{Error (\%)} = \left(\frac{f_{\text{XX}}}{2^{m+1} \times k \times \text{Target baud rate}} - 1 \right) \times 100 \text{ [\%]}$$

- 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 (5) Allowable baud rate range during reception.

To set the baud rate, perform the following calculation for setting the UAnCTL1 and UAnCTL2 registers (when using the internal clock).

<1> Set k to $f_{xx}/(2 \times \text{target baud rate})$ and m to 0.

<2> If k is 256 or greater ($k \geq 256$), reduce k to half ($k/2$) and increment m by 1 ($m + 1$).

<3> Repeat Step <2> until k becomes less than 256 ($k < 256$).

<4> Round off the first decimal point of k to the nearest whole number.

If k is 256 after round-off, reduce k to half ($k/2$) and increment m by 1 ($m + 1$) to obtain $k = 128$.

<5> Set the value of m to the UAnCTL1 register and the value of k to the UAnCTL2 register.

Example: When $f_{xx} = 20$ MHz and target baud rate = 153,600 bps

<1> $k = 20,000,000/(2 \times 153,600) = 65.10\dots$, $m = 0$

<2>, <3> $k = 65.10\dots < 256$, $m = 0$

<4> Set value of UAnCTL2 register: $k = 65 = 41H$, set value of UAnCTL1 register: $m = 0$

Actual baud rate = $20,000,000/(2 \times 65)$
= 153,846 [bps]

Baud rate error = $\{20,000,000/(2 \times 65 \times 153,600) - 1\} \times 100$
= 0.160 [%]

Representative examples of baud rate settings are shown below.

Table 15-6. Baud Rate Generator Setting Data

Baud Rate (bps)	$f_{xx} = 20$ MHz			$f_{xx} = 16$ MHz			$f_{xx} = 10$ MHz		
	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)	UAnCTL1	UAnCTL2	ERR (%)
300	08H	82H	0.16	07H	D0H	0.16	07H	82H	0.16
600	07H	82H	0.16	06H	D0H	0.16	06H	82H	0.16
1200	06H	82H	0.16	05H	D0H	0.16	05H	82H	0.16
2400	05H	82H	0.16	04H	D0H	0.16	04H	82H	0.16
4800	04H	82H	0.16	03H	D0H	0.16	03H	82H	0.16
9600	03H	82H	0.16	02H	D0H	0.16	02H	82H	0.16
19200	02H	82H	0.16	01H	D0H	0.16	01H	82H	0.16
31250	01H	A0H	0	01H	80H	0	00H	A0H	0
38400	01H	82H	0.16	00H	D0H	0.16	00H	82H	0.16
76800	00H	82H	0.16	00H	68H	0.16	00H	41H	0.16
153600	00H	41H	0.16	00H	34H	0.16	00H	21H	-1.36
312500	00H	20H	0	00H	1AH	-1.54	00H	10H	0
625000	00H	10H	0	00H	0DH	-1.54	00H	08H	0

Remark f_{xx} : Main clock frequency

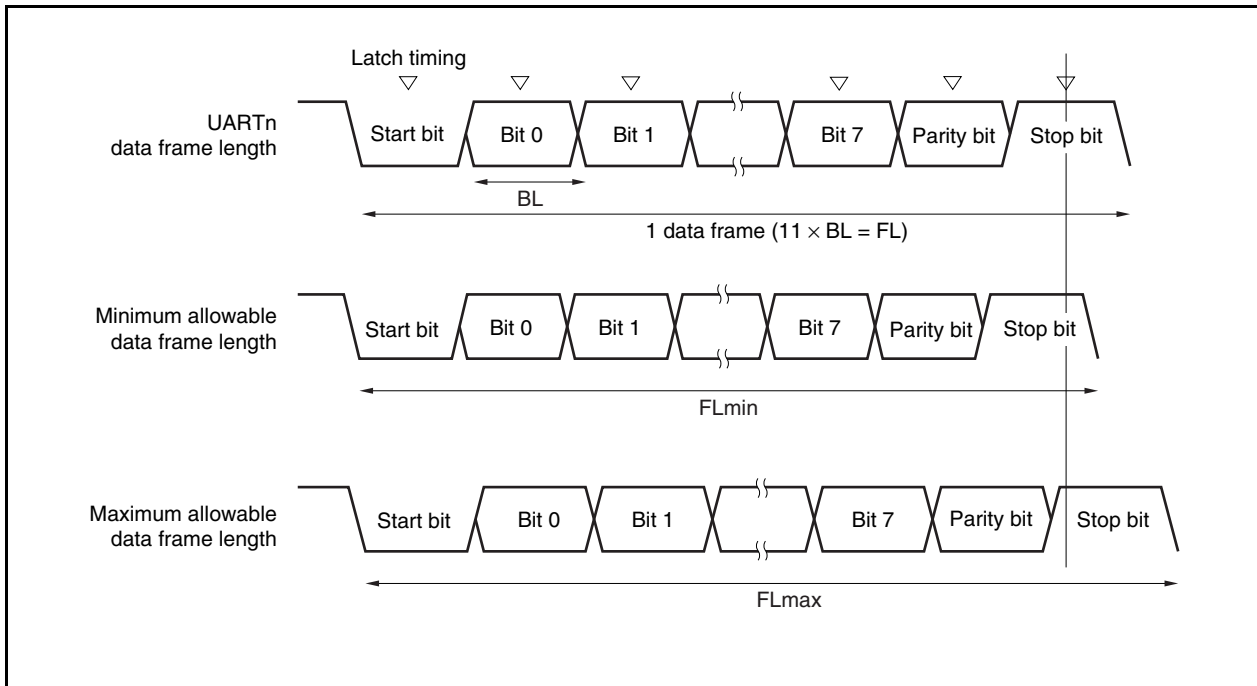
ERR: Baud rate error (%)

(5) 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 indicated below is a theoretical value. In practice, the signal might be distorted, or communication might not be performed normally even if the error is within the allowable range. Therefore, the error must be minimized.

Figure 15-18. Allowable Baud Rate Range During Reception



As shown in Figure 15-18, the receive data latch timing is determined by the counter set using the UAnCTL2 register following start bit detection. The transmit data can be received normally 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.

$$BL = (\text{Brate})^{-1}$$

Brate: UARTAn baud rate ($n = 0$ to 2)

k: Setting value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits ($n = 0$ to 2)

BL: 1-bit data length

FL: Length of 1 data frame

Latch timing margin: 2 clock cycles

$$\text{Minimum allowable data frame length: } FL_{\min} = 11 \times BL - \frac{k-2}{2k} \times BL = \frac{21k+2}{2k} BL$$

Therefore, the maximum baud rate that can be received by the destination is as follows.

$$BR_{\max} = (FL_{\min}/11)^{-1} = \frac{22k}{21k + 2} \text{ Brate}$$

Similarly, obtaining the maximum allowable data frame length yields the following.

$$\frac{10}{11} \times FL_{\max} = 11 \times BL - \frac{k + 2}{2 \times k} \times BL = \frac{21k - 2}{2 \times k} BL$$

$$FL_{\max} = \frac{21k - 2}{20k} BL \times 11$$

Therefore, the minimum baud rate that can be received by the destination is as follows.

$$BR_{\min} = (FL_{\max}/11)^{-1} = \frac{20k}{21k - 2} \text{ Brate}$$

Obtaining the allowable baud rate error for UARTA and the destination from the above-described equations yields the following.

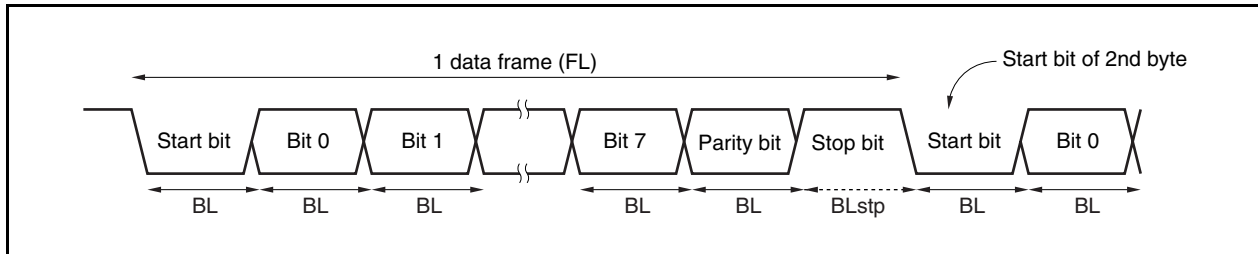
Table 15-7. Maximum/Minimum Allowable Baud Rate Error (11-Bit Length)

Division Ratio (k)	Maximum Allowable Baud Rate Error	Minimum Allowable Baud Rate Error
4	+2.32%	-2.43%
8	+3.53%	-3.61%
20	+4.26%	-4.31%
50	+4.56%	-4.58%
100	+4.66%	-4.67%
255	+4.72%	-4.73%

- Remarks 1.** The reception accuracy depends on the bit count in 1 frame, the base clock frequency (f_{CLK}), and the division ratio (k). The higher the base clock frequency (f_{CLK}) and the larger the division ratio (k), the higher the accuracy.
- 2.** k: Setting value of UAnCTL2.UAnBRS7 to UAnCTL2.UAnBRS0 bits

(6) Data frame length during continuous transmission

In continuous transmission, the data frame length from the stop bit to the next start bit is 2 base clock cycles longer than usual. However, timing initialization is performed via start bit detection by the receiving side, so this has no influence on the transfer result.

Figure 15-19. Data Frame Length During Continuous Transmission

Assuming a 1 bit data length of BL ; a stop bit length of BL_{stp} ; and a base clock frequency of f_{CLK} , we obtain the following equation.

$$BL_{stp} = BL + 2/f_{CLK}$$

Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Data frame length} = 11 \times BL + (2/f_{CLK})$$

15.8 Cautions

- (1) When the clock supply to UARTAn is stopped (for example, in IDLE1, IDLE2, or STOP mode), the operation stops with each register retaining the value it had immediately before the clock supply was stopped. The TXDAn pin output also holds and outputs the value it had immediately before the clock supply was stopped. However, the operation is not guaranteed after the clock supply is resumed. Therefore, after the clock supply is resumed, the circuits should be initialized by setting the UAnCTL0.UAnPWR, UAnCTL0.UAnRXEn, and UAnCTL0.UAnTXEn bits to 0, 0, 0.
- (2) The RXDA1 and KR7 pins must not be used at the same time. When using the RXDA1 pin, do not use the KR7 pin. When using the KR7 pin, do not use the RXDA1 pin. (It is recommended to set the PFC91 bit to 1 and clear PFCE91 bit to 0.)
- (3) In UARTAn, the interrupt caused by a communication error does not occur. When transferring transmit data and receive data using DMA transfer, error processing cannot be performed even if errors (parity, overrun, framing) occur during transfer. Either read the UAnSTR register after DMA transfer has been completed to make sure that there are no errors, or read the UAnSTR register during communication to check for errors.
- (4) RXDA0 and INTP7 use the same pin. To use the pin for the RXDA0 function, disable edge detection for INTP7 (INTF3.INTF31 bit = 0, INTR3.INTR31 bit = 0).
- (5) Start up UARTAn in the following sequence.
 - <1> Set the UAnCTL0.UAnPWR bit to 1.
 - <2> Set the ports.
 - <3> Set the UAnCTL0.UAnTXE bit to 1 and the UAnCTL0.UAnRXE bit to 1.
- (6) Stop UARTAn in the following sequence.
 - <1> Set the UAnCTL0.UAnTXE bit to 0 and the UAnCTL0.UAnRXE bit to 0.
 - <2> Set the ports and set the UAnCTL0.UAnPWR bit to 0 (it is not a problem if the port settings are not changed).
- (7) In transmit mode (UAnCTL0.UAnPWR bit = 1 and UAnCTL0.UAnTXE bit = 1), do not overwrite the same value to the UAnTX register by software because transmission starts by writing to this register. To transmit the same value continuously, overwrite the same value.
- (8) In continuous transmission, the period from the stop bit to the next start bit is 2 base clock cycles longer than usual. However, the reception side initializes the timing by detecting the start bit, so the reception result is not affected.
- (9) UARTA cannot identify the start bit if low level signals are continuously input to the RXDAn pin.

CHAPTER 16 CLOCKED SERIAL INTERFACE B (CSIB)

The number of CSIB channels in the V850ES/JC3-L, V850ES/JE3-L are shown below.

Product Name	V850ES/JC3-L (40-pin)	V850ES/JC3-L (48-pin)	V850ES/JE3-L
Number of channels	2 channels CSIB0, CSIB2	4 channels CSIB0 to CSIB2, CSIB4	5 channels CSIB0 to CSIB4

In this chapter, the number of channels is expressed as n.

16.1 Features

○ 3-wire serial interface

SOBn: Serial data output

SIBn: Serial data input

SCKBn: Serial clock I/O

Transmission mode, reception mode, and transmission/reception mode can be specified.

○ Transfer rate: 8 Mbps max

○ Master mode and slave mode can be selected.

○ Two interrupt request signals:

- Reception complete interrupt (INTCBnR): This interrupt occurs when receive data is transferred to the CBnRX register with reception enabled, or when an overrun error occurs. In the single transfer mode, this interrupt occurs upon completion of transmission, even when only transmission is executed.
- Transmission enable interrupt (INTCBnT): In continuous transmission or continuous transmission/reception mode, this interrupt occurs when transmit data is transferred from the CBnTX register and it becomes possible to write data to CBnTX.

○ Timing of data reception/transmission via SCKBn can be specified

○ Transfer data length can be selected in 1-bit units from between 8 and 16 bits

○ Transfer data can be switched between MSB-first and LSB-first

○ Double buffers for both transmission and reception

○ Overrun error detection

16.2 Configuration

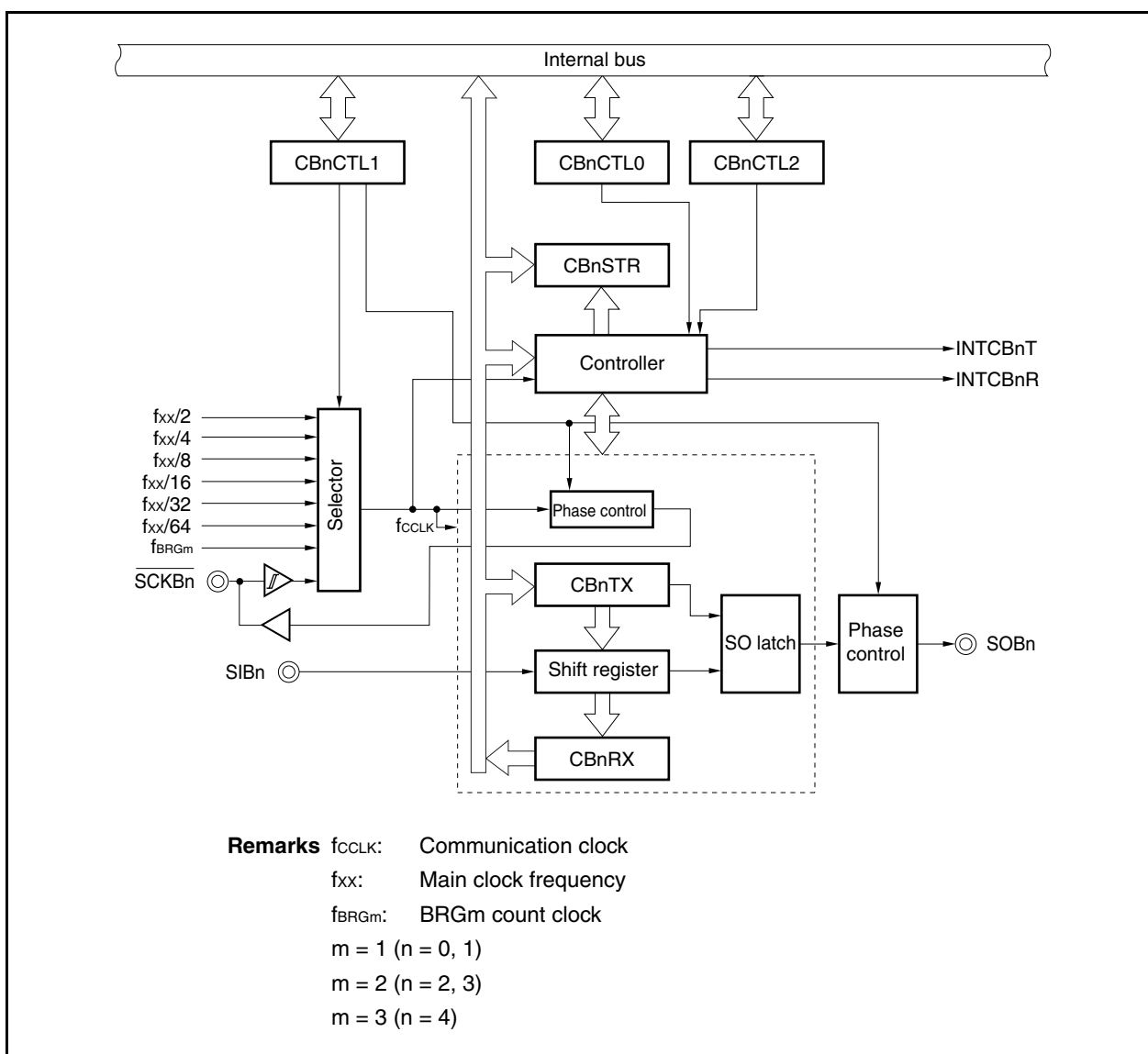
CSIBn includes the following hardware.

Table 16-1. Configuration of CSIBn

Item	Configuration
Registers	CSIBn receive data register (CBnRX) CSIBn transmit data register (CBnTX) CSIBn control register 0 (CBnCTL0) CSIBn control register 1 (CBnCTL1) CSIBn control register 2 (CBnCTL2) CSIBn status register (CBnSTR)

The following shows the block diagram of CSIBn.

Figure 16-1. Block Diagram of CSIBn



16.2.1 Pin functions of each channel

The SIBn, SOBn, and SCKBn pins used by CSIB in the V850ES/JC3-L, V850ES/JE3-L are used for other functions as shown in Table 16-2. To use these pins for CSIB, set the related registers as described in **Table 4-17 Settings When Pins Are Used for Alternate Functions**.

Table 16-2. Pins Used by CSIB

Channel	Pin No.			Port	CSIB Reception Input	CSIB Transmission Output	CSIB Clock I/O	Other Functions
	JC3L (40)	JC3L (48)	JE3L					
CSIB0	16	20	28	P40	SIB0	–	–	SDA01
	17	21	29	P41	–	SOB0	–	SCL01
	18	22	30	P42	–	–	$\overline{\text{SCKB0}}$	–
CSIB1	–	38	50	P97	SIB1	–	–	TIP20/TOP20
	–	39	51	P98	–	SOB1	–	–
	–	36	48	P99	–	–	$\overline{\text{SCKB1}}$	–
CSIB2	24	28	38	P53	SIB2	–	–	KR3/TIQ00/TOQ00/RTP03/DDO
	25	29	39	P54	–	SOB2	–	KR4/RTP04/DCK
	26	30	40	P55	–	–	$\overline{\text{SCKB2}}$	KR5/RTP05/DMS
CSIB3	–	–	16	P910	SIB3	–	–	–
	–	–	15	P911	–	SOB3	–	–
	–	–	14	P912	–	–	$\overline{\text{SCKB3}}$	–
CSIB4	–	34	46	P31	SIB4	–	–	RXDA0/INTP7
	–	33	45	P30	–	SOB4	–	TXDA0
	–	35	47	P32	–	–	$\overline{\text{SCKB4}}$	ASCKA0/TIP00/TOP00

Remark JC3L (40): V850ES/JC3-L (40-pin products)
 JC3L (48): V850ES/JC3-L (48-pin products)
 JE3L: V850ES/JE3-L

16.3 Mode Switching of CSIB and Other Serial Interfaces

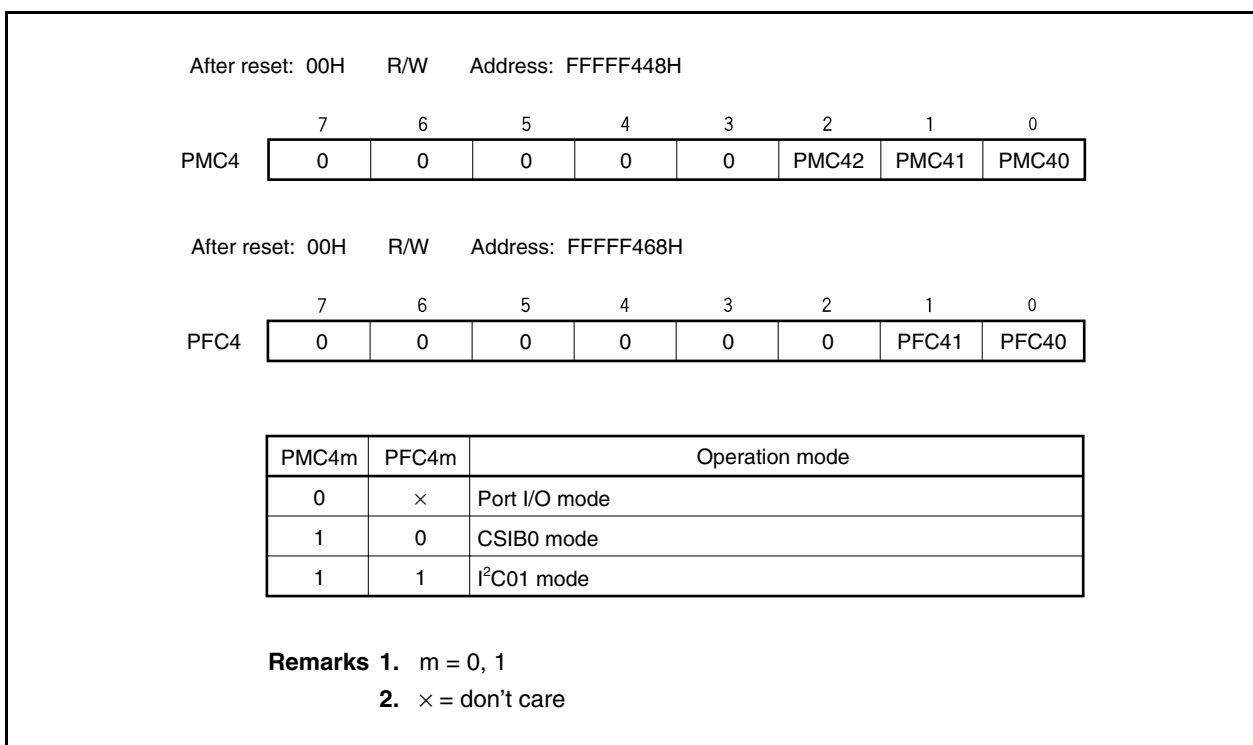
16.3.1 CSIB0 and I²C01 mode switching

In the V850ES/JC3-L and V850ES/JE3-L, CSIB0 and I²C01 share pins and therefore cannot be used simultaneously. To use the CSIB0 function, specify the CSIB0 mode in advance by using the PMC4 and PFC4 registers.

Switching the operation mode between CSIB0 and I²C01 is described below.

Caution Transmission and reception by CSIB0 and I²C01 are not guaranteed if these operation modes are switched during transmission or reception. Be sure to disable the serial interface that is not being used.

Figure 16-2. Switching CSIB0 and I²C01 Operation Modes



16.3.2 CSIB4 and UARTA0 mode switching

In the V850ES/JE3-L, CSIB4 and UARTA0 share pins and therefore cannot be used simultaneously. To use the CSIB4 function, specify the CSIB4 mode in advance by using the PMC3, PFC3, and PFCE3L registers.

Switching the operation mode between CSIB4 and UARTA0 is described below.

Caution Transmission and reception by CSIB4 and UARTA0 are not guaranteed if these operation modes are switched during transmission or reception. Be sure to disable the serial interface that is not being used.

Figure 16-3. Switching CSIB4 and UARTA0 Operation Modes

After reset: 0000H R/W Address: FFFFF446H, FFFFF447H

	15	14	13	12	11	10	9	8
PMC3	0	0	0	0	0	0	PMC39	PMC38
	7	6	5	4	3	2	1	0
	0	0	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

After reset: 0000H R/W Address: FFFFF466H, FFFFF467H

	15	14	13	12	11	10	9	8
PFC3	0	0	0	0	0	0	PFC39	PFC38
	7	6	5	4	3	2	1	0
	0	0	PFC35	PFC34	PFC33	PFC32	PFC31	PFC30

After reset: 00H R/W Address: FFFFF706H

	7	6	5	4	3	2	1	0
PFCE3L	0	0	0	0	0	PFCE32	0	0

PMC32	PFCE32	PFC32	Operation mode
0	×	×	Port I/O mode
1	0	0	ASCKA0 mode
1	0	1	SCKB4 mode

PMC3m	PFC3m	Operation mode
0	×	Port I/O mode
1	0	UARTA0 mode
1	1	CSIB4 mode

Remarks

- m = 0, 1
- × = don't care

16.4 Registers

The following registers are used to control CSIBn.

- CSIBn receive data register (CBnRX)
- CSIBn transmit data register (CBnTX)
- 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.

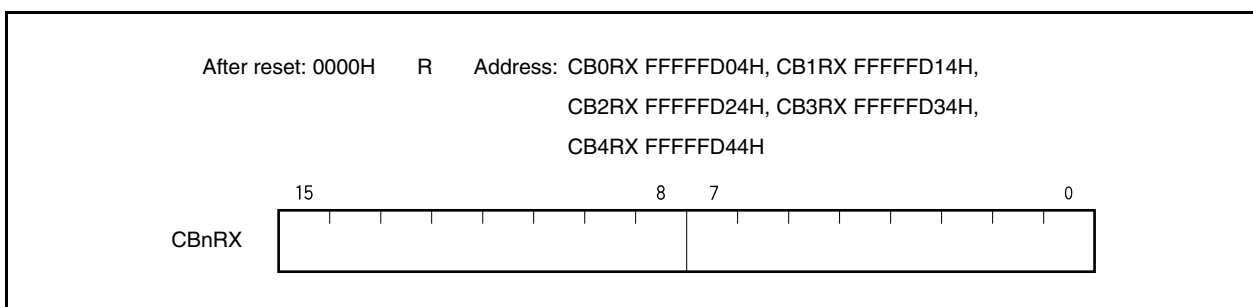
This register is read-only, in 16-bit units.

The receive operation is started by reading the CBnRX register in reception mode.

If the transfer data length is 8 bits, the lower 8 bits of this register are read-only in 8-bit units as the CBnRXL register.

Reset sets this register to 0000H.

In addition to reset input, the CBnRX register can be initialized by clearing (to 0) the CBnPWR bit of the CBnCTL0 register.



(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 when transmission is enabled.

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 sets this register to 0000H.



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

(3) CSIBn control register 0 (CBnCTL0)

CBnCTL0 is an 8-bit register that controls CSIBn serial transfer.

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

Reset sets this register to 01H.

(1/3)

After reset: 01H R/W Address: CB0CTL0 FFFFFFFD00H, CB1CTL0 FFFFFFFD10H,
CB2CTL0 FFFFFFFD20H, CB3CTL0 FFFFFFFD30H,
CB4CTL0 FFFFFFFD40H

	<7>	<6>	<5>	<4>	3	2	1	<0>
CBnCTL0	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
<ul style="list-style-type: none"> 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
<ul style="list-style-type: none"> The SOBn output is low level when the CBnTXE bit is 0. 	

CBnRXE ^{Note}	Specification of receive operation disable/enable
0	Disable receive operation
1	Enable receive operation
<ul style="list-style-type: none"> When the CBnRXE bit is 0, no reception complete interrupt is output even when the specified data is transferred, and the receive data (in the CBnRX register) is not updated. 	

Note These bits can only be rewritten when the CBnPWR bit is 0. However, the values of these bits can be changed to 0 or 1 at the same time the CBnPWR bit is set.

Caution To forcibly suspend transmission/reception, clear the CBnPWR bit to 0 instead of the CBnRXE and CBnTXE bits. At this time, the clock output is stopped.

(2/3)

CBnDIR ^{Note}	Specification of transfer direction mode (MSB/LSB)
0	MSB-first transfer
1	LSB-first transfer

CBnTMS ^{Note}	Transfer mode specification
0	Single transfer mode
1	Continuous transfer mode

[In single transfer mode]

- In this mode, the reception complete interrupt (INTCBnR) occurs upon completion of communication. The transmission enable interrupt (INTCBnT) does not occur even if transmission is enabled (CBnTXE bit = 1).
- After the reception complete interrupt (INTCBnR) occurs, writing/reading the next transmit/receive data triggers the next communication.
- The next communication does not start even if the next transmit/receive data is written/read during the preceding communication (CBnSTR.CBnTSF bit = 1).

[In continuous transfer mode]

- In this mode and with transmission enabled (CBnTXE bit = 1), the transmission enable interrupt (INTCBnT) occurs when writing the next transmit data becomes possible. With reception enabled (CBnRXE bit = 1), the reception complete interrupt (INTCBnR) occurs upon completion of transfer.
- Writing the next transmit data becomes possible after INTCBnT occurs. If new data is written at this time, continuous transfer can be performed.
- If reception-only is specified (CBnTXE bit = 0, CBnRXE bit = 1), the next transmission starts immediately after INTCBnR has occurred, regardless of the progress of reading the CBnRX register. Be sure to read receive data immediately after INTCBnR has occurred. If receive data is not read before the next INTCBnR occurs, an overrun error will occur (CBnSTR.CBnOVE bit = 1).

Note These bits can only be rewritten when the CBnPWR bit is 0. However, the values of these bits can be changed to 0 or 1 at the same time the CBnPWR bit is set.

CBnSCE	Specification of start transfer disable/enable
0	Communication start trigger invalid
1	Communication start trigger valid

This bit enables or disables the communication start trigger in reception mode.

- In master mode
 - (a) In single transmission or transmission/reception mode, or continuous transmission or continuous transmission/reception mode:
The setting of the CBnSCE bit has no effect on communication.
 - (b) In single reception mode:
Clear the CBnSCE bit to 0 before reading the last receive data to disable the start of reception because reception is started by reading the receive data (CBnRX register)^{Note 1}.
 - (c) In continuous reception mode
Clear the CBnSCE bit to 0 one communication clock cycle before reception of the last data is completed to disable the start of reception after the last data is received^{Note 2}.
- In slave mode
Set the CBnSCE bit to 1.

[Usage of CBnSCE bit]

- In single reception mode
 - <1> When reception of the last data is completed by INTCBnR interrupt servicing, clear the CBnSCE bit to 0 before reading the CBnRX register.
 - <2> After confirming that the CBnSTR.CBnTSF bit is 0, clear the CBnPWR and CBnRXE bits to 0 to disable reception.
To receive data again, set the CBnSCE bit to 1 to start the next reception by dummy-reading the CBnRX register.
- In continuous reception mode
 - <1> Clear the CBnSCE bit to 0 in the INTCBnR interrupt servicing for the receive data immediately before the last one.
 - <2> Read the CBnRX register.
 - <3> Read the last reception data by reading the CBnRX register after acknowledging the CBnTIR interrupt.
 - <4> After confirming that the CBnSTR.CBnTSF bit is 0, clear the CBnPWR and CBnRXE bits to 0 to disable reception.
To receive data again, set the CBnSCE bit to 1 to wait for the next reception by dummy-reading the CBnRX register.

Notes 1. If the CBnRX register is read with the CBnSCE bit set to 1, the next communication is started.

2. If the CBnSCE bit is not cleared to 0, one communication clock cycle before reception of the last data is completed, the next communication is automatically started.

Caution Be sure to clear bits 3 and 2 to "0".

(4) CSIBn control register 1 (CBnCTL1)

CBnCTL1 is an 8-bit register that is used to specify the CSIBn serial transfer operation mode. This register can be read or written in 8-bit or 1-bit units. Reset sets this register to 00H.

Caution The CBnCTL1 register can be rewritten only when the CBnCTL0.CBnPWR bit is 0.

After reset 00H R/W Address: CB0CTL1 FFFFFFFD01H, CB1CTL1 FFFFFFFD11H,
CB2CTL1 FFFFFFFD21H, CB3CTL1 FFFFFFFD31H,
CB4CTL1 FFFFFFFD41H

	7	6	5	4	3	2	1	0
CBnCTL1	0	0	0	CBnCKP	CBnDAP	CBnCKS2	CBnCKS1	CBnCKS0

	CBnCKP	CBnDAP	Specification of data transmission/reception timing in relation to SCKBn
Communication type 1	0	0	
Communication type 2	0	1	
Communication type 3	1	0	
Communication type 4	1	1	

CBnCKS2	CBnCKS1	CBnCKS0	Communication clock (f _{CCLK}) ^{Note}	Mode
0	0	0	f _{xx} /2	Master mode
0	0	1	f _{xx} /4	Master mode
0	1	0	f _{xx} /8	Master mode
0	1	1	f _{xx} /16	Master mode
1	0	0	f _{xx} /32	Master mode
1	0	1	f _{xx} /64	Master mode
1	1	0	f _{BRGm}	Master mode
1	1	1	External clock (SCKBn)	Slave mode

Note Set the communication clock (f_{CCLK}) to 8 MHz or lower.

Remark When n = 0, 1, m = 1
 When n = 2, 3, m = 2
 When n = 4, m = 3
 For details of f_{BRGm}, see **16.8 Baud Rate Generator**.

(5) CSIBn control register 2 (CBnCTL2)

CBnCTL2 is an 8-bit register that is used to specify the CSIBn serial transfer data length.

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

Reset sets this register to 00H.

Caution The CBnCTL2 register can be rewritten only when the CBnCTL0.CBnPWR bit is 0 or when both the CBnTXE and CBnRXE bits are 0.

After reset: 00H R/W Address: CB0CTL2 FFFFFFFD02H, CB1CTL2 FFFFFFFD12H,
CB2CTL2 FFFFFFFD22H, CB3CTL2 FFFFFFFD32H,
CB4CTL2 FFFFFFFD42H

	7	6	5	4	3	2	1	0
CBnCTL2	0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0

CBnCL3	CBnCL2	CBnCL1	CBnCL0	Transfer data 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

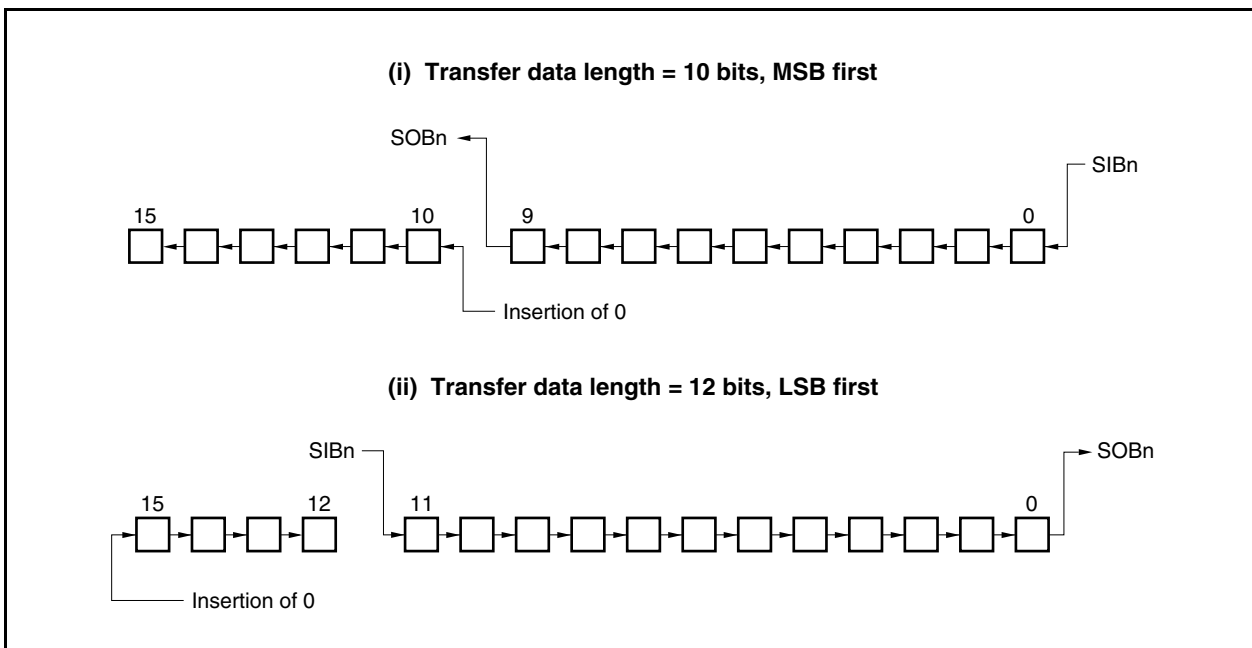
Remarks 1. If the transfer data length is other than 8 or 16 bits, set the data to the CBnTX or CBnRX register starting from the LSB.
2. ×: don't care

(a) Changing the transfer data length

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 data length is set to a value other than 16 bits, the data must be set 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 received data becomes 0 following serial transfer.

Figure 16-4. Example of Operation with Transfer Data Length Set to Other Than 16 Bits



(6) CSIBn status register (CBnSTR)

CBnSTR is an 8-bit register that displays the CSIBn status.

This register can be read or written in 8-bit or 1-bit units, but the CBnTSF flag is read-only.

Reset sets this register to 00H.

In addition to reset input, the CBnSTR register can be initialized by clearing (0) the CBnCTL0.CBnPWR bit.

After reset 00H		R/W	Address: CB0STR FFFFFFFD03H, CB1STR FFFFFFFD13H, CB2STR FFFFFFFD23H, CB3STR FFFFFFFD33H, CB4STR FFFFFFFD43H					
CBnSTR	<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							
<ul style="list-style-type: none"> 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 cycle. 								
CBnOVE	Overrun error flag							
0	No overrun							
1	Overrun							
<ul style="list-style-type: none"> An overrun error occurs when the next reception is completed without the CPU reading the value of the receive buffer, during reception or upon completion of a receive operation. The CBnOVE flag displays the overrun error occurrence status in this case. The CBnOVE bit is valid also in the single transfer mode. Therefore, when only using transmission, note the following. <ul style="list-style-type: none"> Do not check the CBnOVE flag. Read this bit even if reading the receive data is not required. The CBnOVE flag is cleared by writing 0 to it. It cannot be set even by writing 1 to it. 								

16.5 Interrupt Request Signals

CSIBn can generate the following two interrupt request signals.

- Reception complete interrupt request signal (INTCBnR)
- Transmission enable interrupt request signal (INTCBnT)

Of these two interrupt request signals, the reception complete interrupt request signal has the higher priority by default, and the priority of the transmission enable interrupt request signal is lower.

Table 16-3. Interrupts and Their Default Priority

Interrupt Request Signal	Priority
Reception complete	High
Transmission enable	Low

(1) Reception complete interrupt request signal (INTCBnR)

When receive data is transferred to the CBnRX register while reception is enabled, the reception complete interrupt request signal is generated.

This interrupt request signal can also be generated if an overrun error occurs.

When the reception complete interrupt request signal is acknowledged and the data is read, read the CBnSTR register to check that the result of reception is not an error.

In the single transfer mode, the INTCBnR interrupt request signal is generated upon completion of transmission, even when only transmission is executed.

(2) Transmission enable interrupt request signal (INTCBnT)

In the continuous transmission or continuous transmission/reception mode, transmit data is transferred from the CBnTX register and, as soon as writing to CBnTX has been enabled, the transmission enable interrupt request signal is generated.

In the single transmission and single transmission/reception modes, the INTCBnT interrupt is not generated.

16.6 Operation

16.6.1 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 ($f_{CLK} = f_{xx}/2$ (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

Figure 16-5. Single Transfer Mode Operation (Master Mode, Transmission Mode)

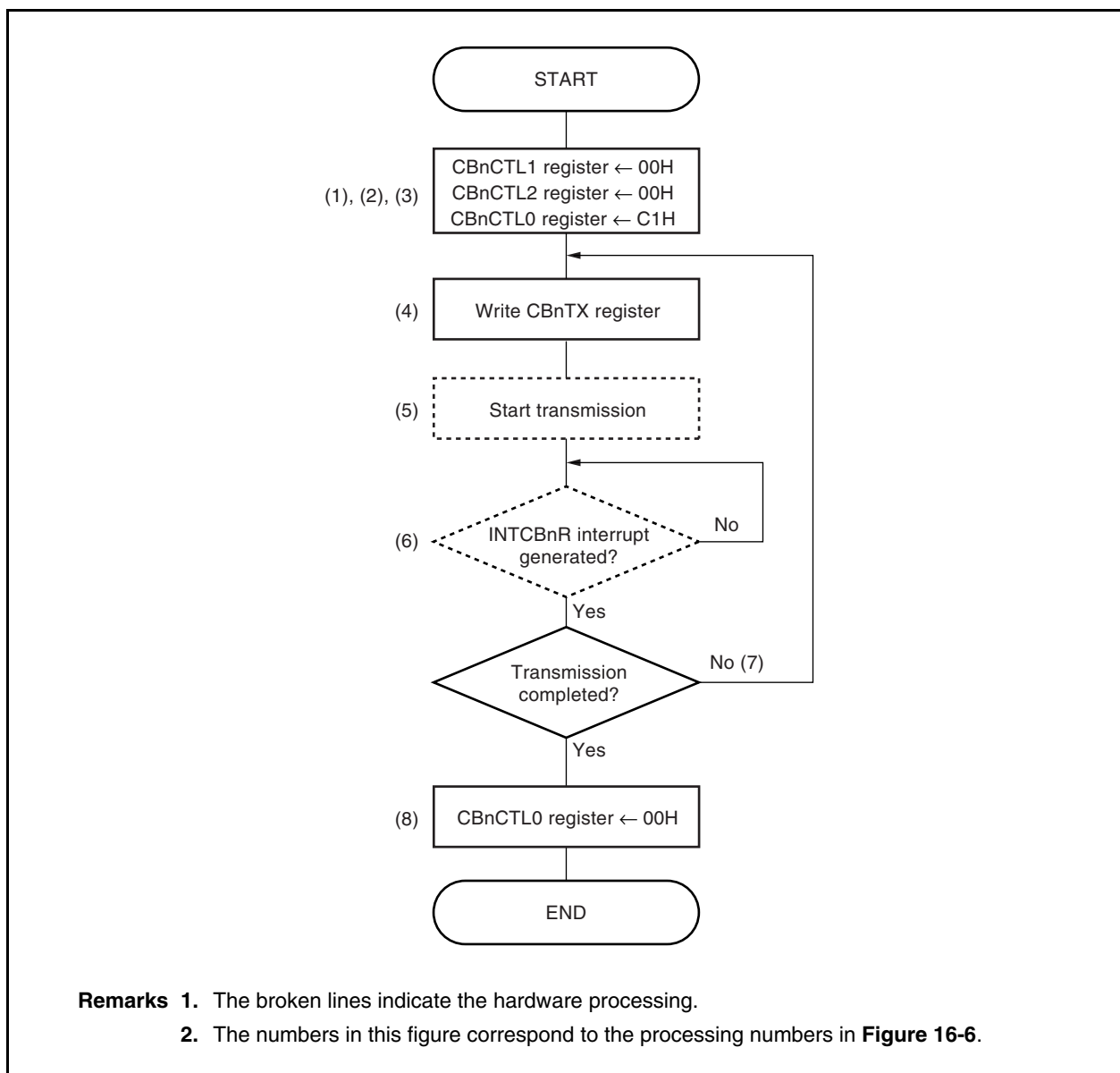
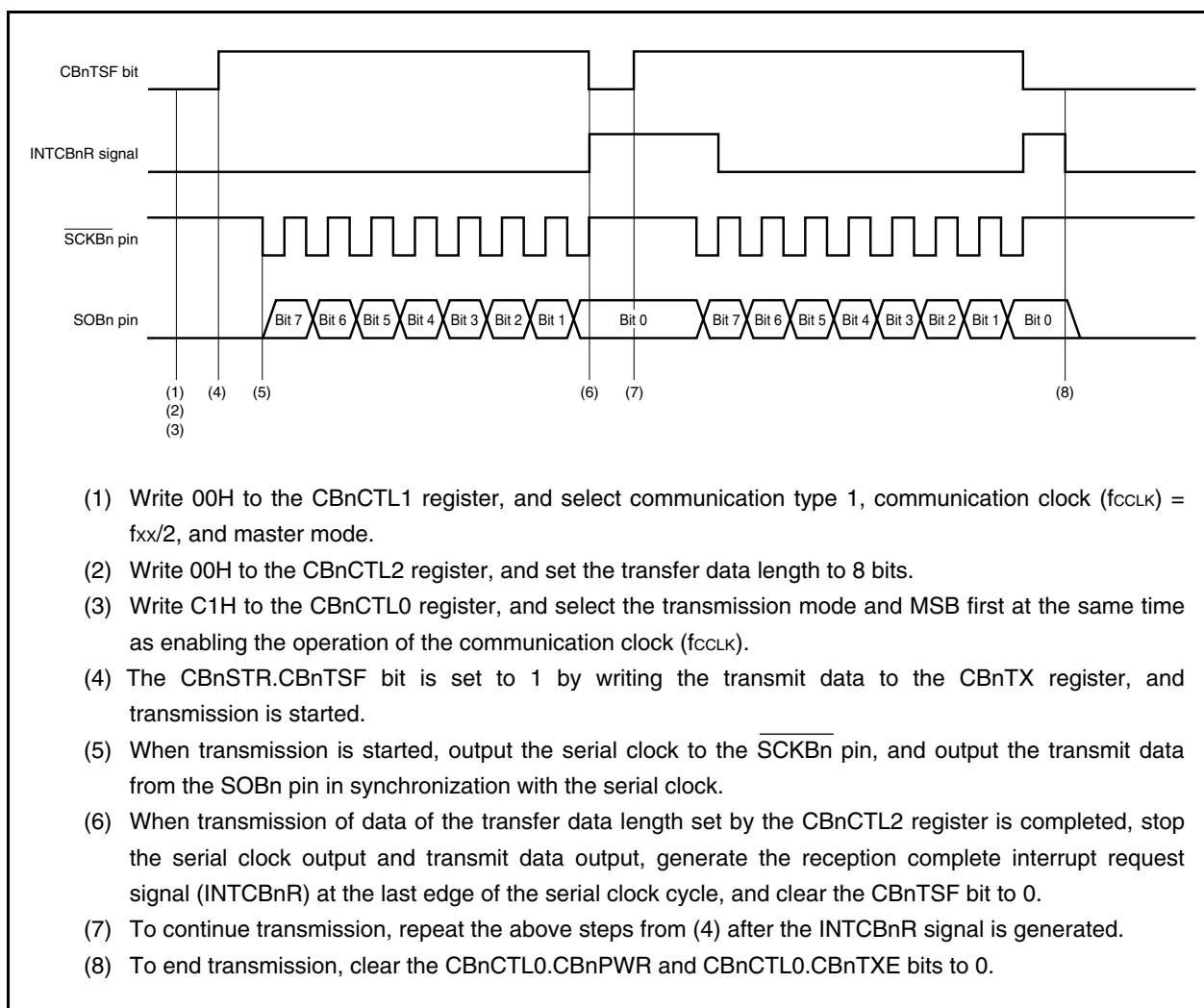


Figure 16-6. Single Transfer Mode Operation Timing (Master Mode, Transmission Mode)

16.6.2 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_{CLK}) = f_{xx}/2 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

Figure 16-7. Single Transfer Mode Operation (Master Mode, Reception Mode)

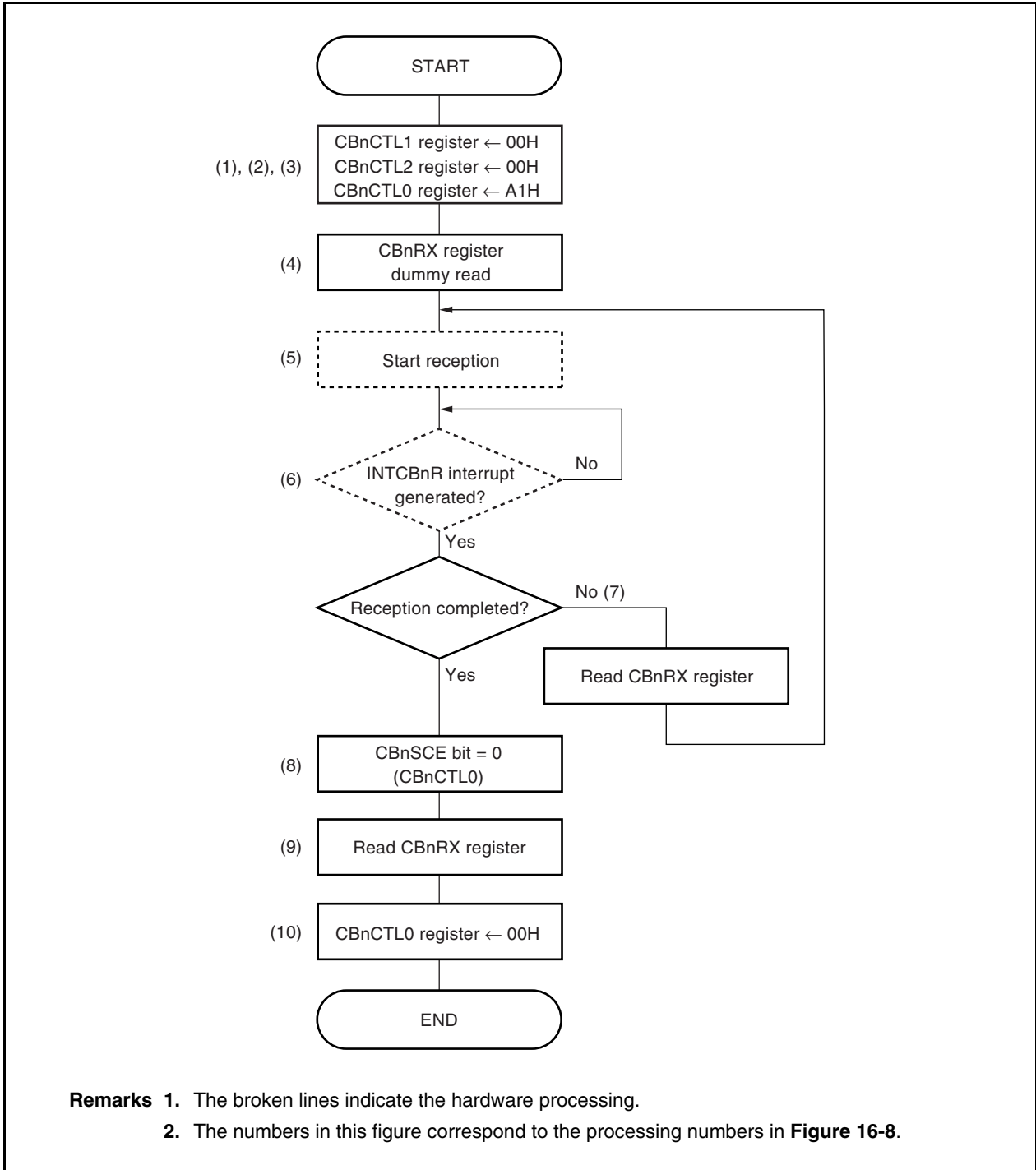
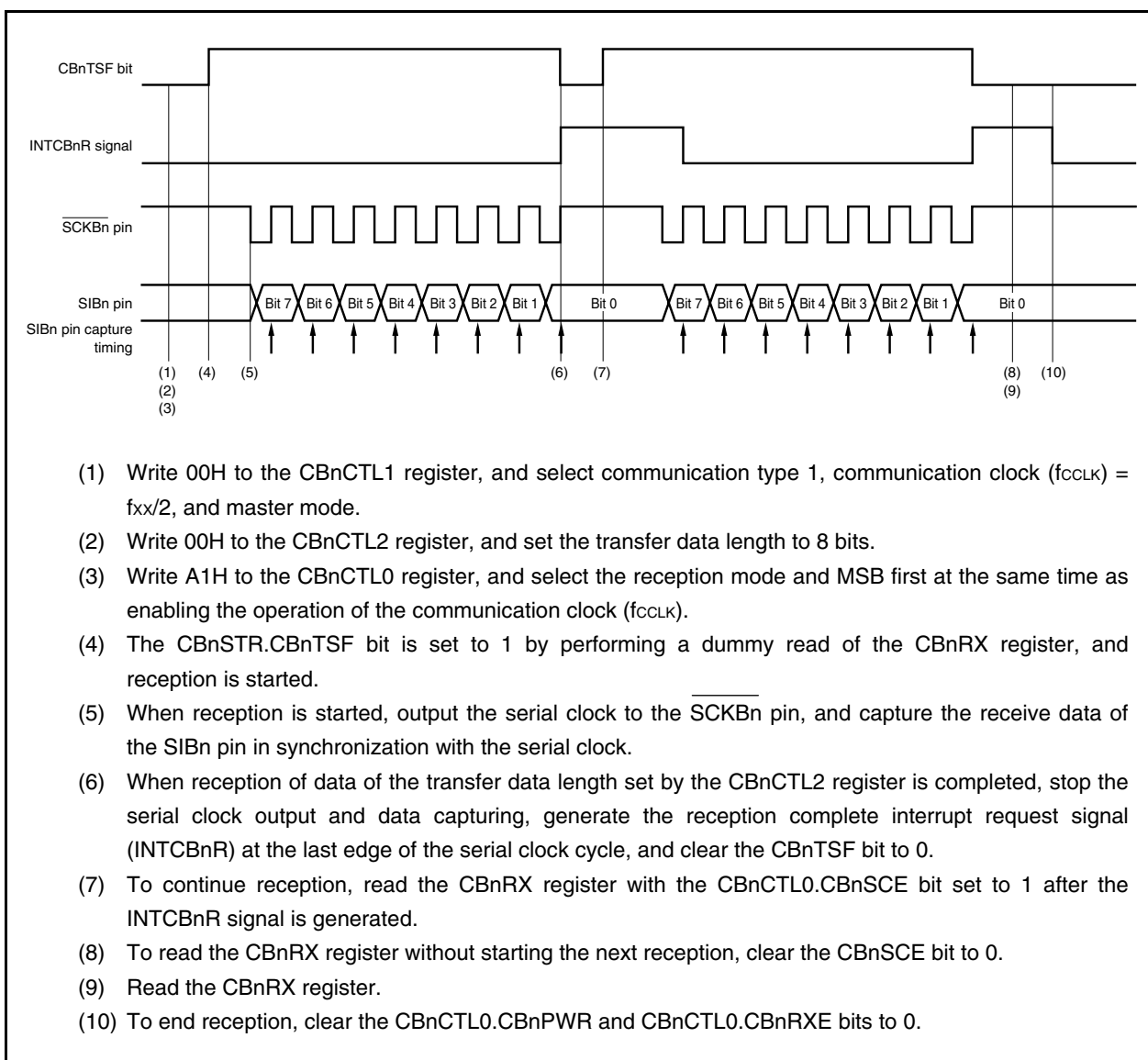
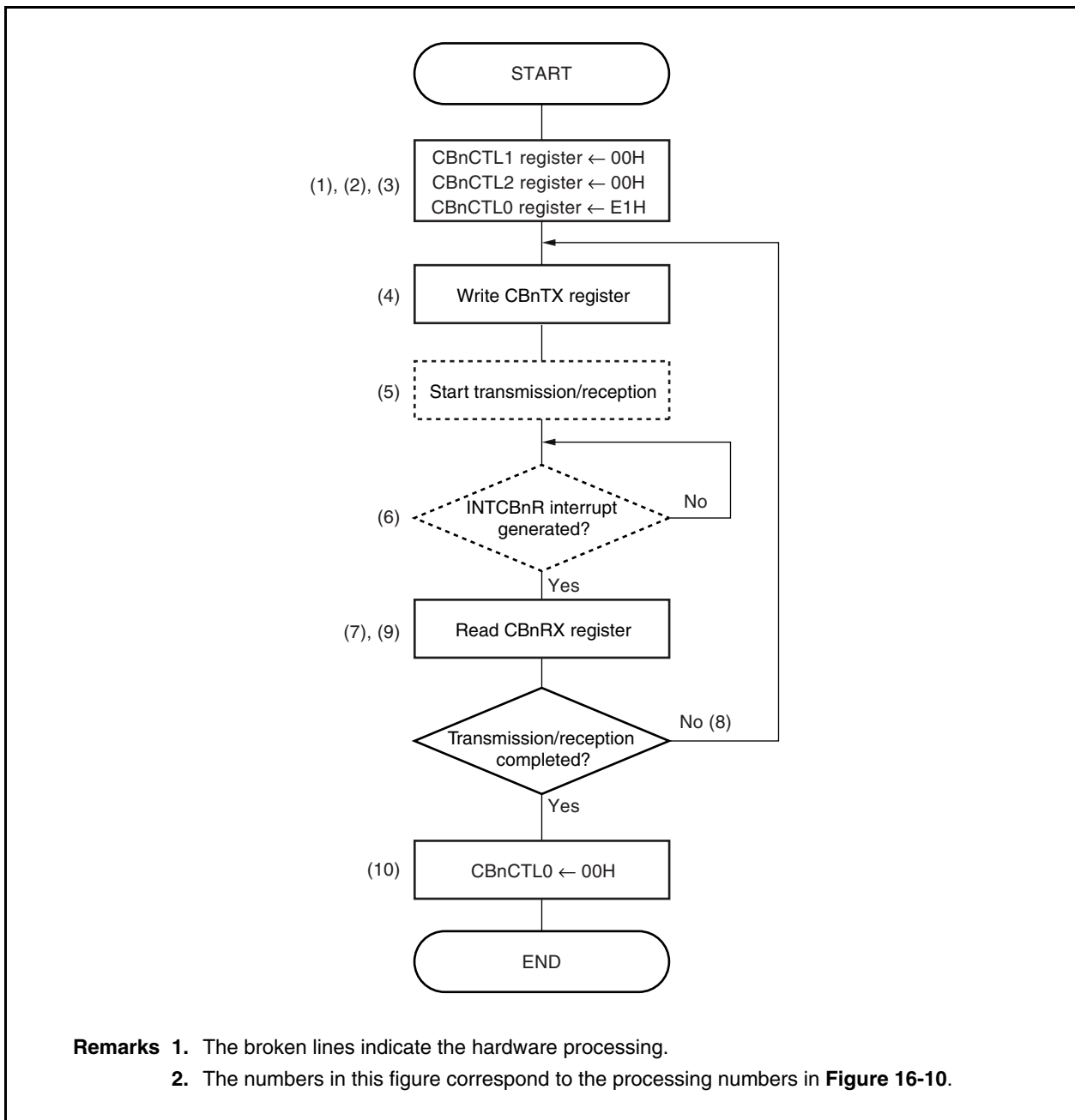


Figure 16-8. Single Transfer Mode Operation Timing (Master Mode, Reception Mode)

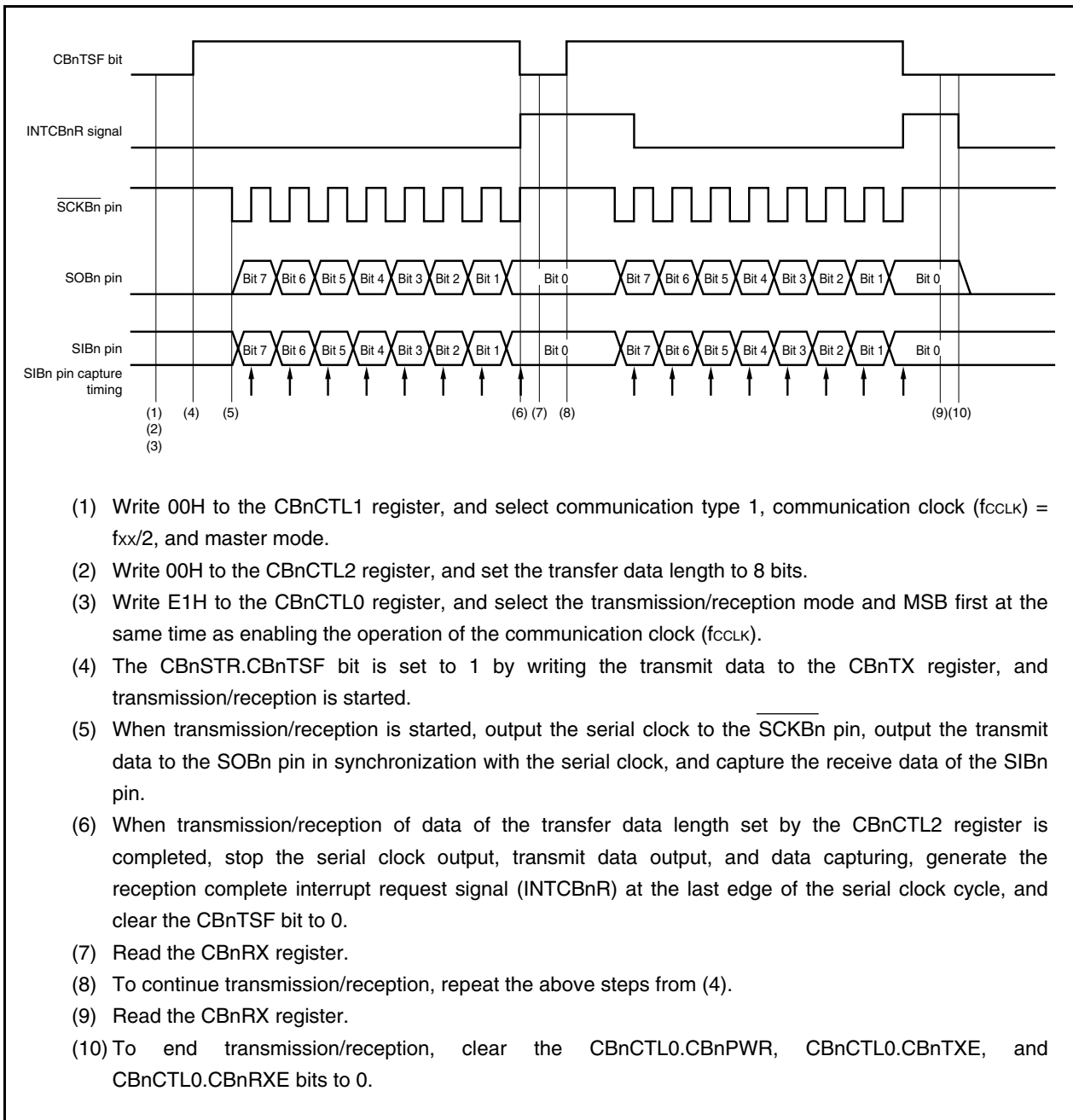
16.6.3 Single transfer mode (master mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CCLK}) = f_{xx}/2 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

Figure 16-9. Single Transfer Mode Operation (Master Mode, Transmission/Reception Mode)



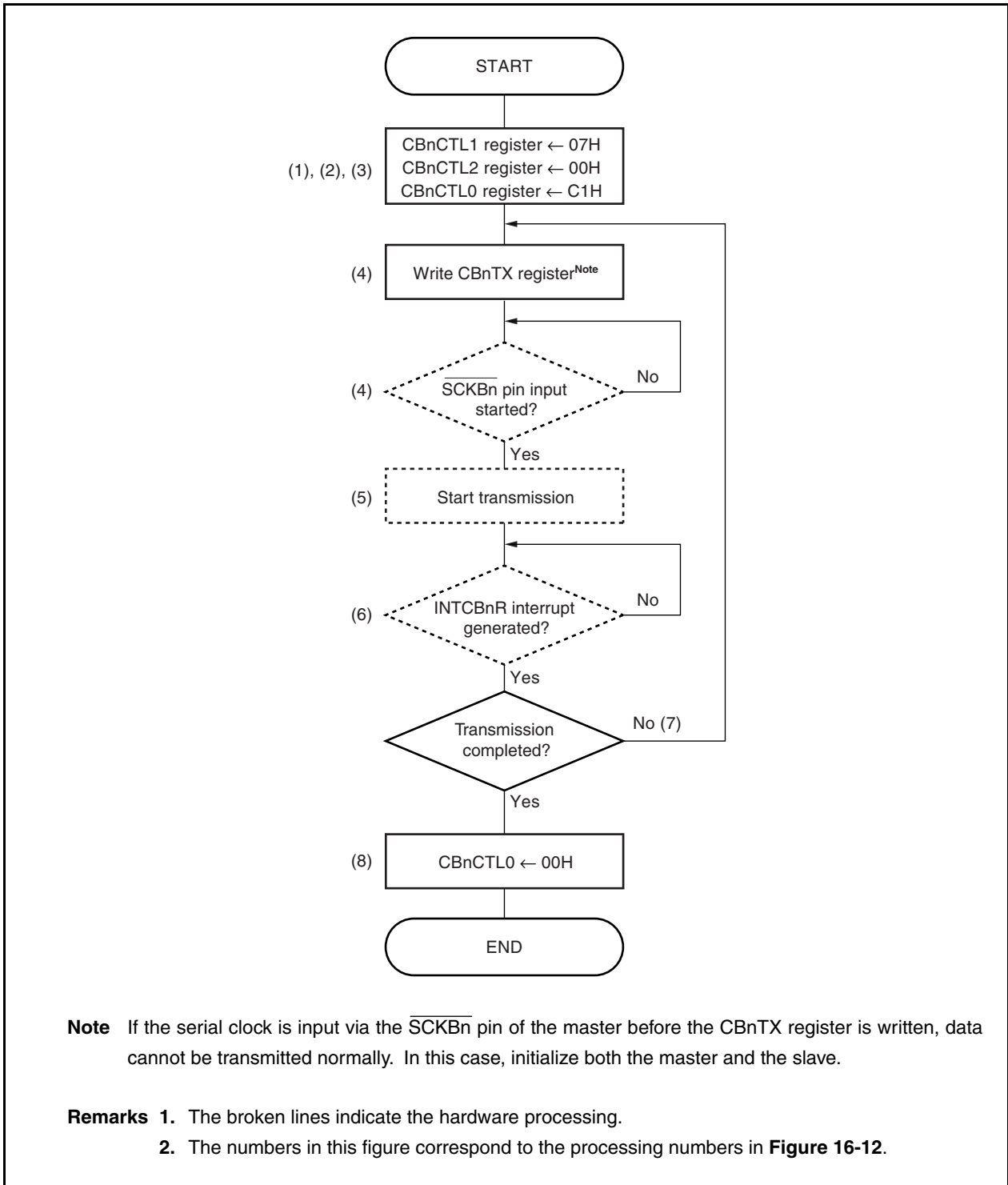
Remarks 1. The broken lines indicate the hardware processing.
 2. The numbers in this figure correspond to the processing numbers in **Figure 16-10**.

Figure 16-10. Single Transfer Mode Operation Timing (Master Mode, Transmission/Reception Mode)

16.6.4 Single transfer mode (slave mode, transmission mode)

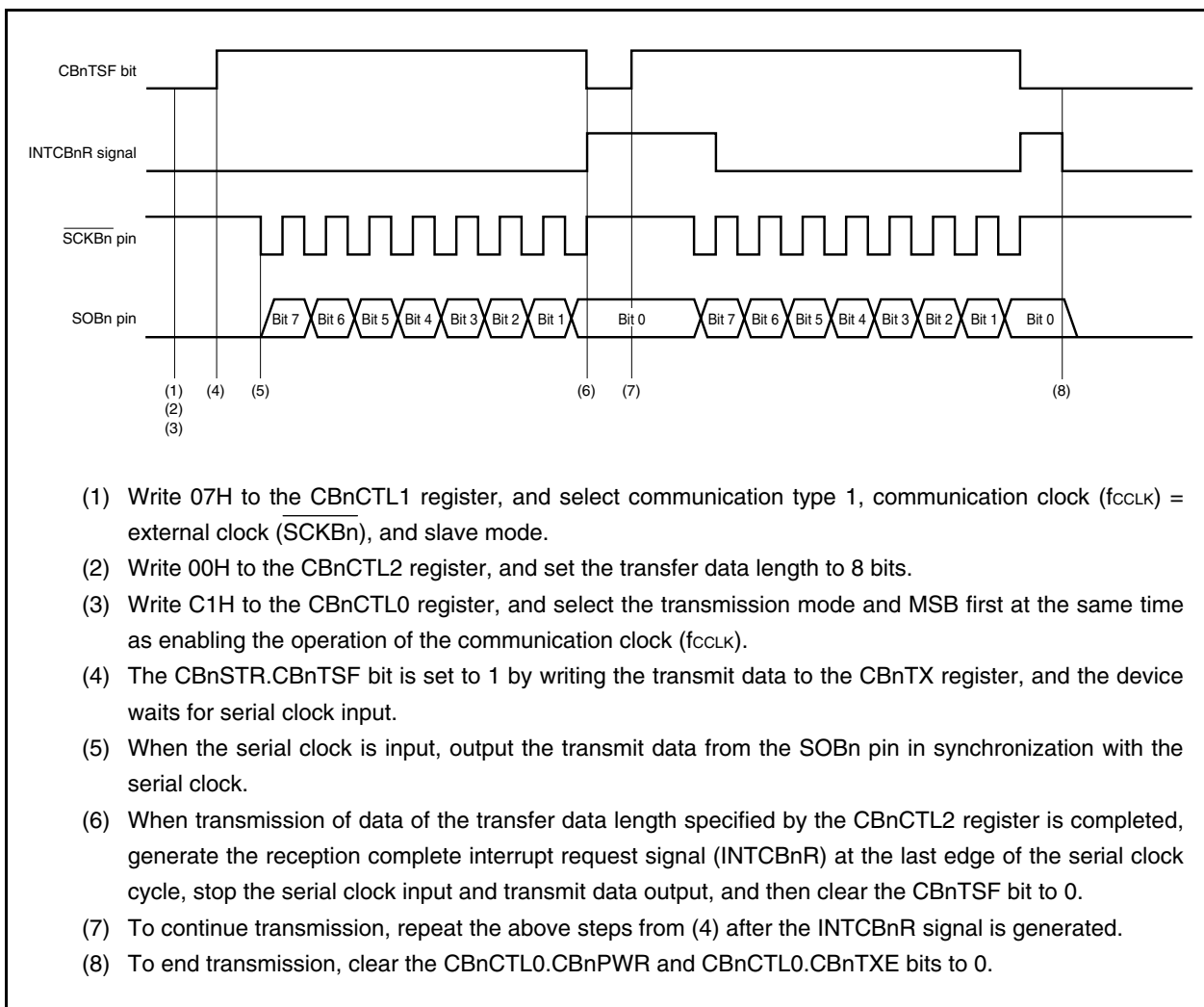
MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CLK}) = external clock (\overline{SCKBn}) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

Figure 16-11. Single Transfer Mode Operation (Slave Mode, Transmission Mode)



Note If the serial clock is input via the \overline{SCKBn} pin of the master before the CBnTX register is written, data cannot be transmitted normally. In this case, initialize both the master and the slave.

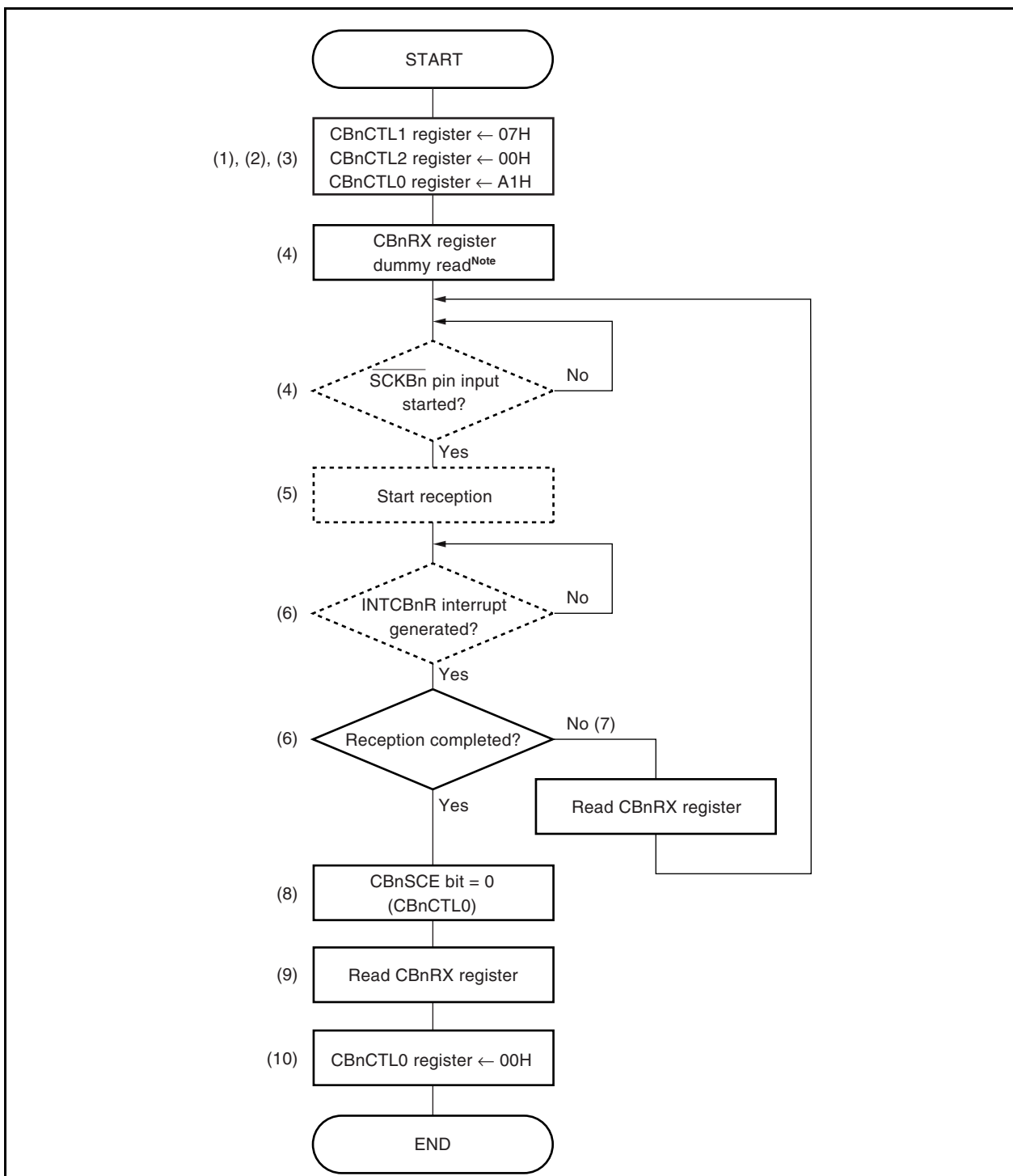
Remarks 1. The broken lines indicate the hardware processing.
 2. The numbers in this figure correspond to the processing numbers in **Figure 16-12**.

Figure 16-12. Single Transfer Mode Operation Timing (Slave Mode, Transmission Mode)

16.6.5 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_{CLK}) = external clock ($\overline{\text{SCKBn}}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

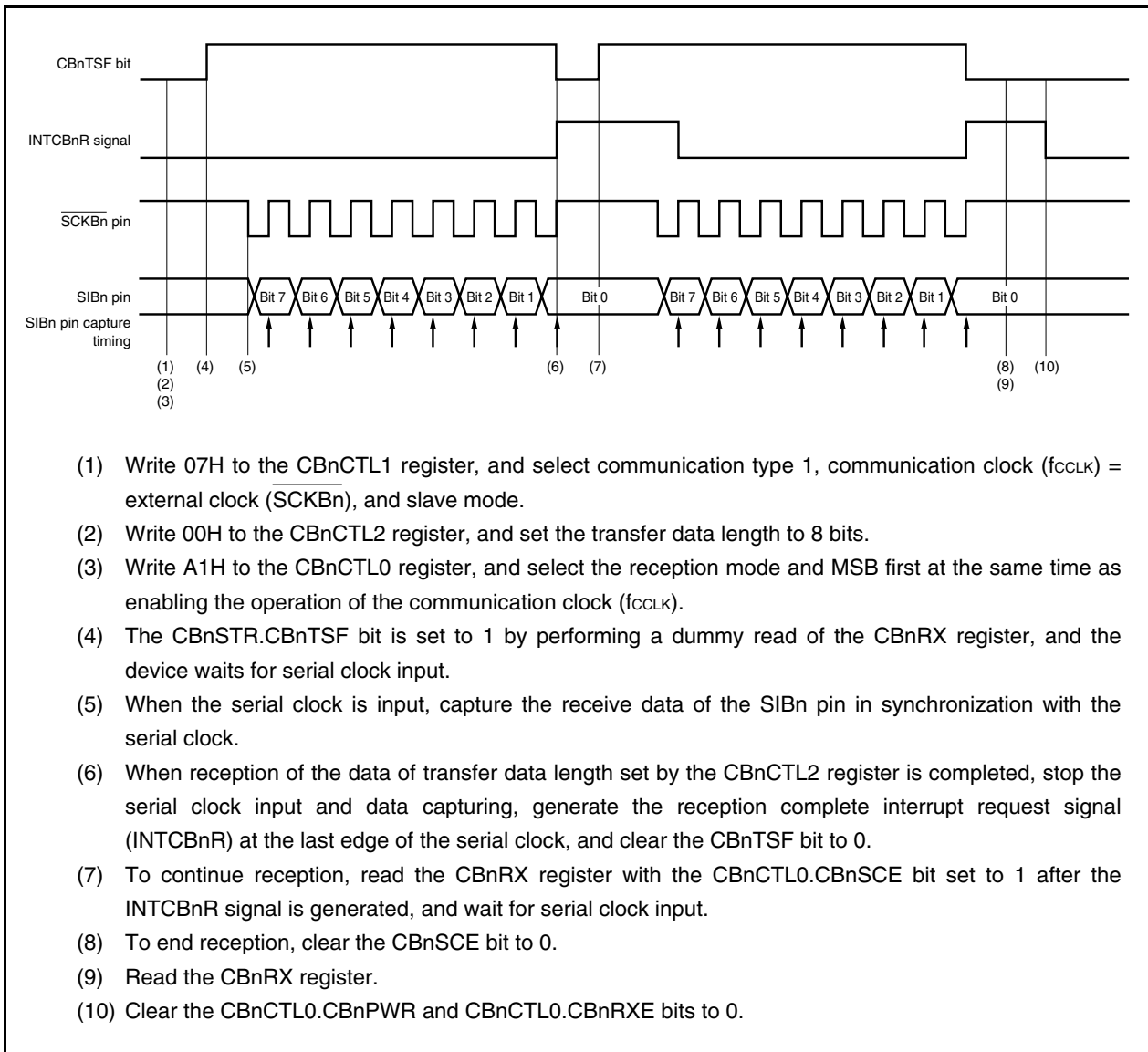
Figure 16-13. Single Transfer Mode Operation (Slave Mode, Reception Mode)



Note If the serial clock is input via the $\overline{\text{SCKBn}}$ pin of the master before a dummy-read of the CBnRX register is executed, data cannot be received normally. In this case, initialize both the master and the slave.

Remarks 1. The broken lines indicate the hardware processing.
 2. The numbers in this figure correspond to the processing numbers in **Figure 16-14**.

Figure 16-14. Single Transfer Mode Operation Timing (Slave Mode, Reception Mode)



16.6.6 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_{CLK}) = external clock (\overline{SCKBn}) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

Figure 16-15. Single Transfer Mode Operation (Slave Mode, Transmission/Reception Mode)

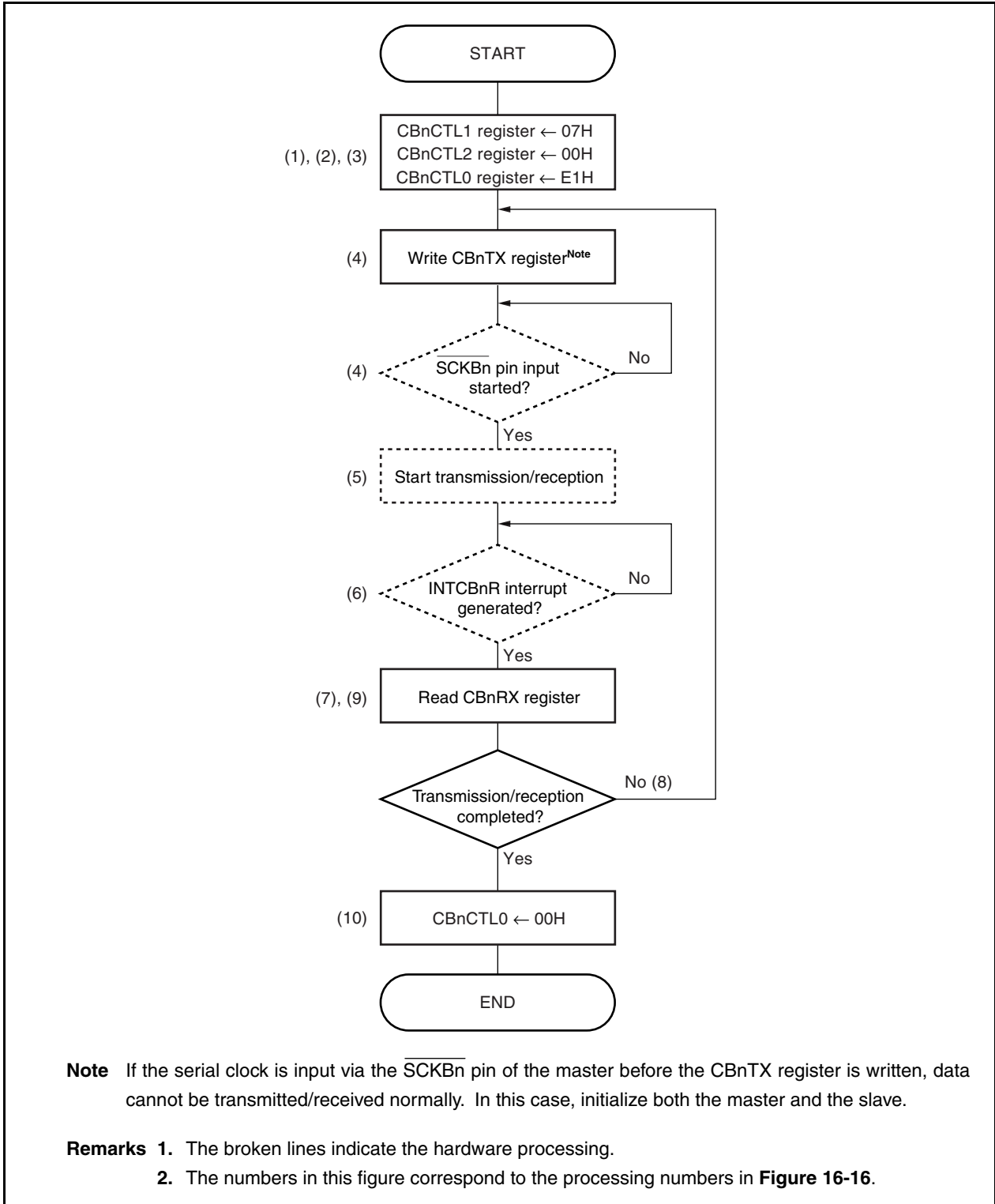
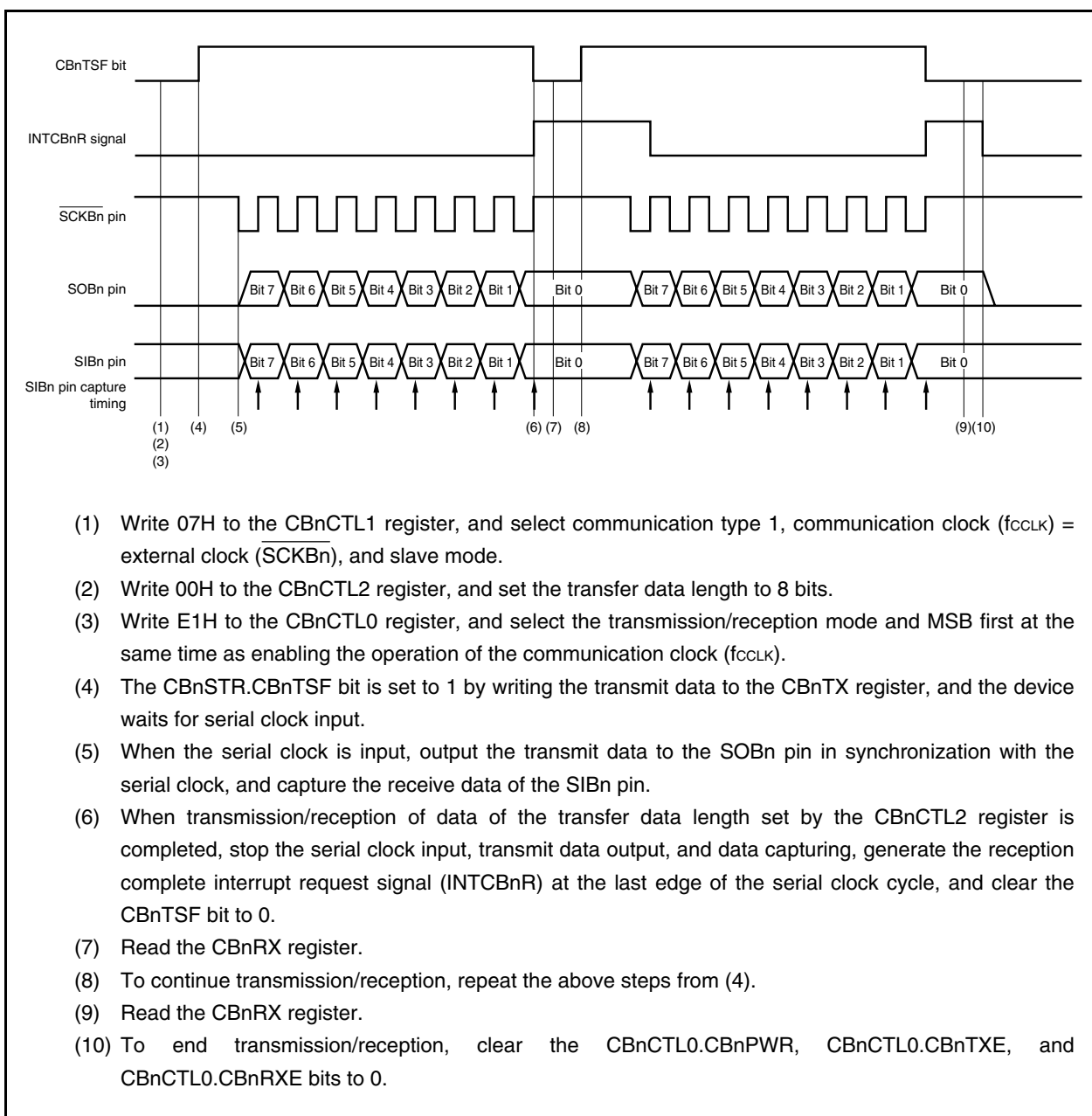


Figure 16-16. Single Transfer Mode Operation Timing (Slave Mode, Transmission/Reception Mode)

16.6.7 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}/2 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

Figure 16-17. Continuous Transfer Mode Operation (Master Mode, Transmission Mode)

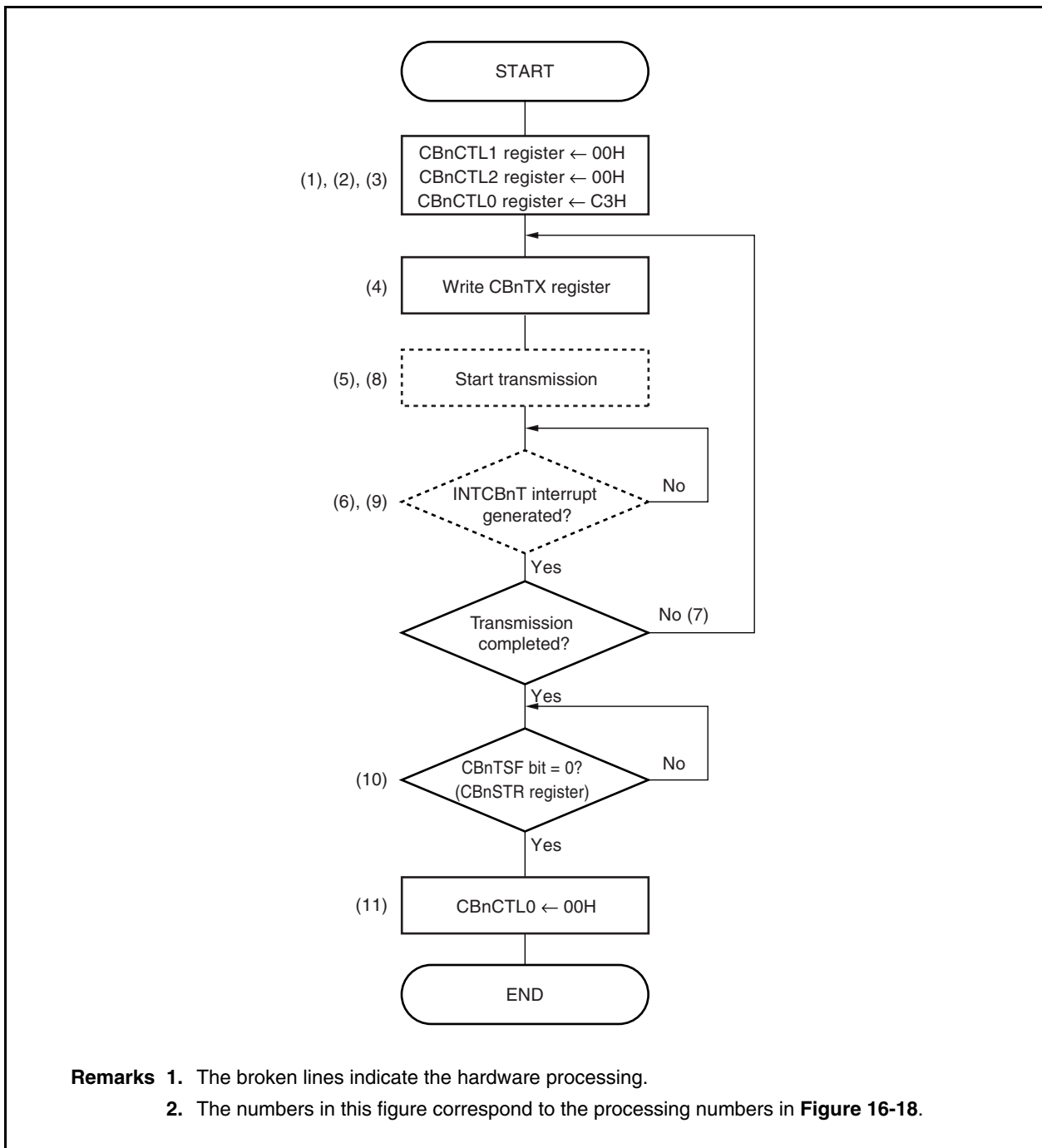
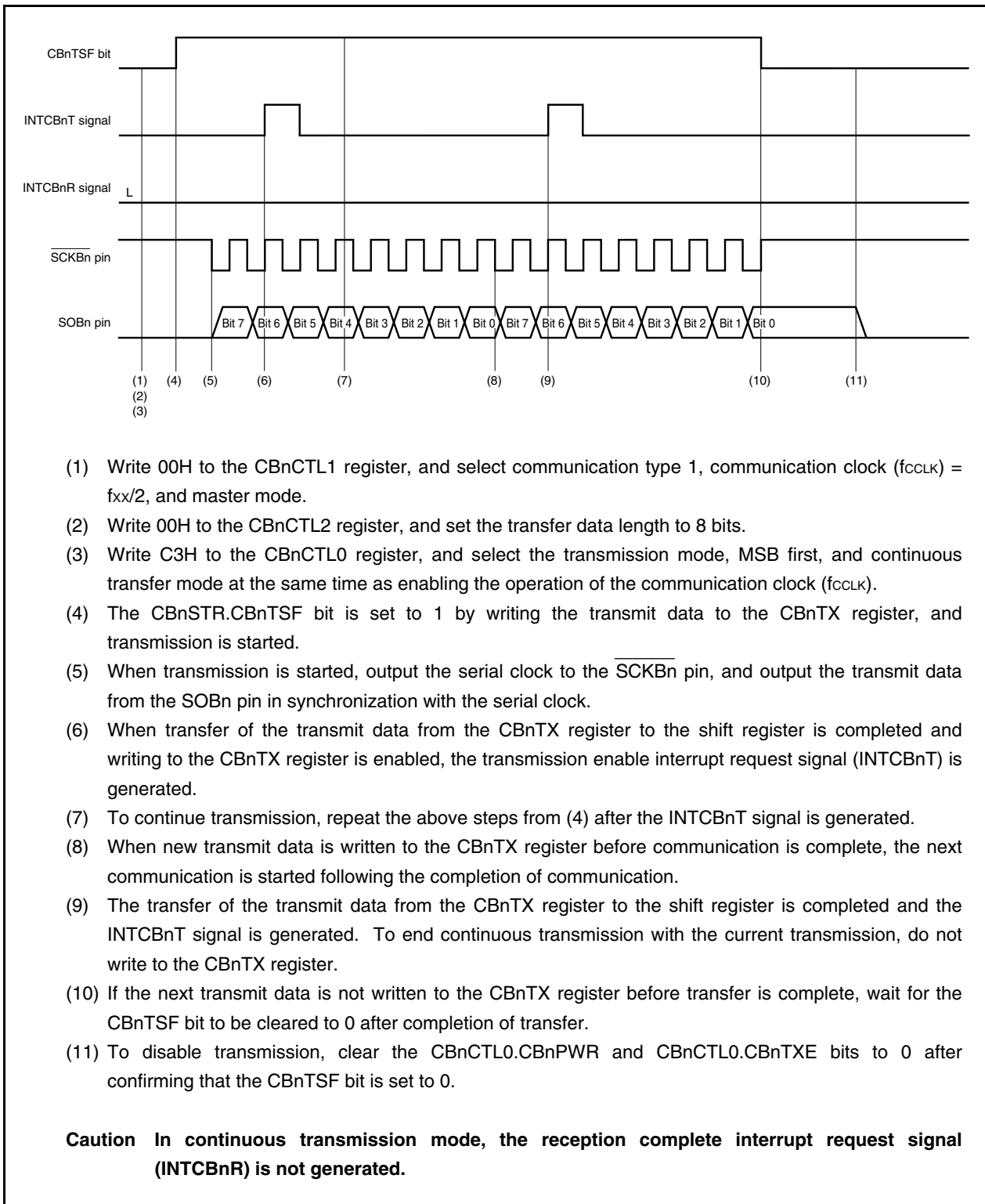


Figure 16-18. Continuous Transfer Mode Operation Timing (Master Mode, Transmission Mode)

16.6.8 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}) = $f_{\text{xx}}/2$ (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

The flowchart in Figure 16-19 shows the operation where a specified number of data items are received in the master mode. Operations are repeated until all the specified data items are received. If an overrun error occurs, however, transfer ends. Perform error processing as necessary. For details about the overrun error, see **16.6.13 Reception errors**.

The operation timing in Figure 16-20 shows a case where no error occurred.

Figure 16-19. Continuous Transfer Mode Operation (Master Mode, Reception Mode)

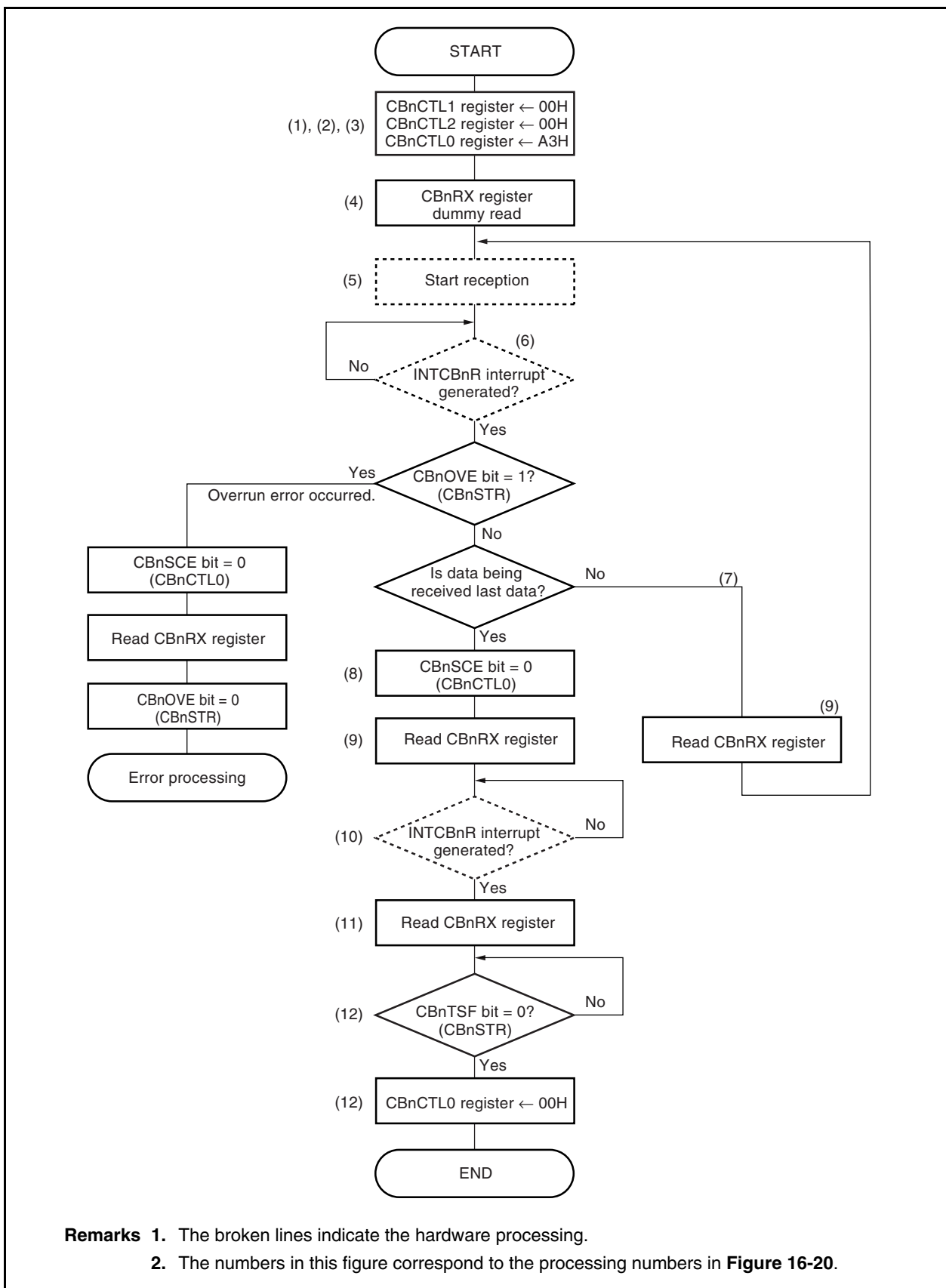
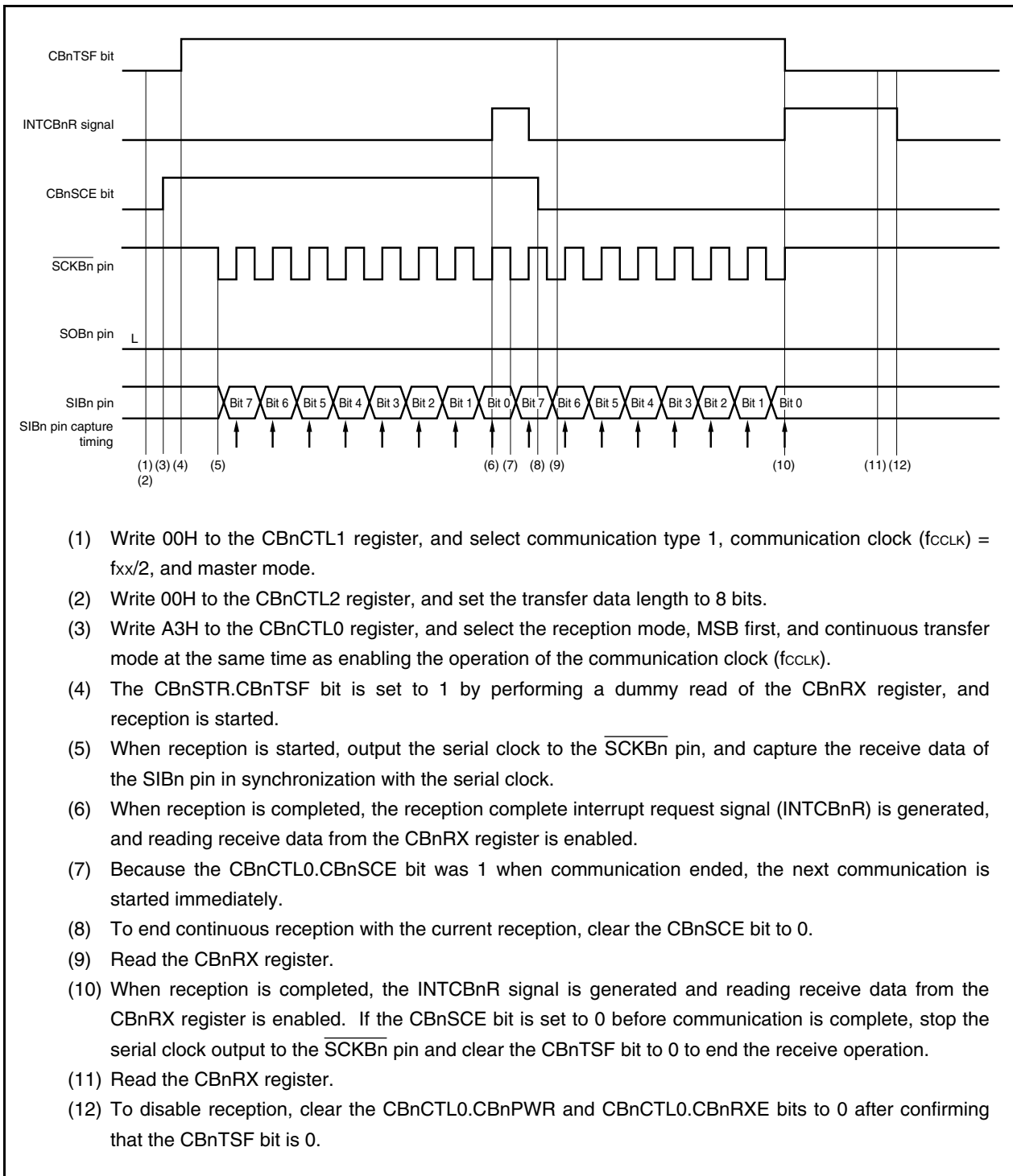


Figure 16-20. Continuous Transfer Mode Operation Timing (Master Mode, Reception Mode)



16.6.9 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 ($f_{\text{CCLK}} = f_{\text{xx}}/2$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

The flowchart in Figure 16-21 shows the operation where the specified number of transmit/receive data items are transmitted/received in the master mode. Operations are repeated until all the specified data items are transmitted/received. If an overrun error occurs, however, transfer ends. Perform error processing as necessary. For details about the overrun error, see **16.6.13 Reception errors**.

The operation timing in Figure 16-22 shows a case where no error occurred.

Figure 16-21. Continuous Transfer Mode Operation (Master Mode, Transmission/Reception Mode)

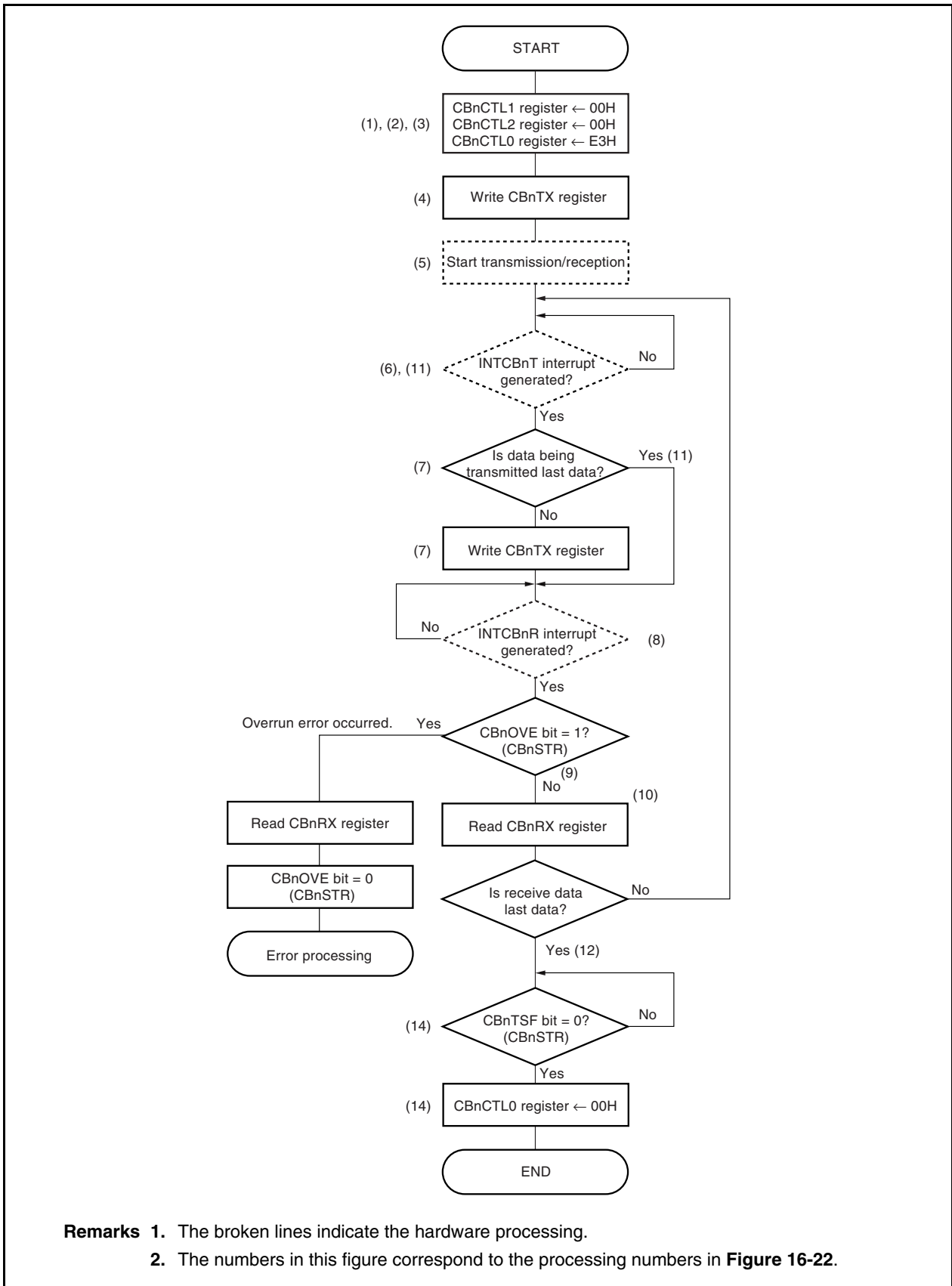


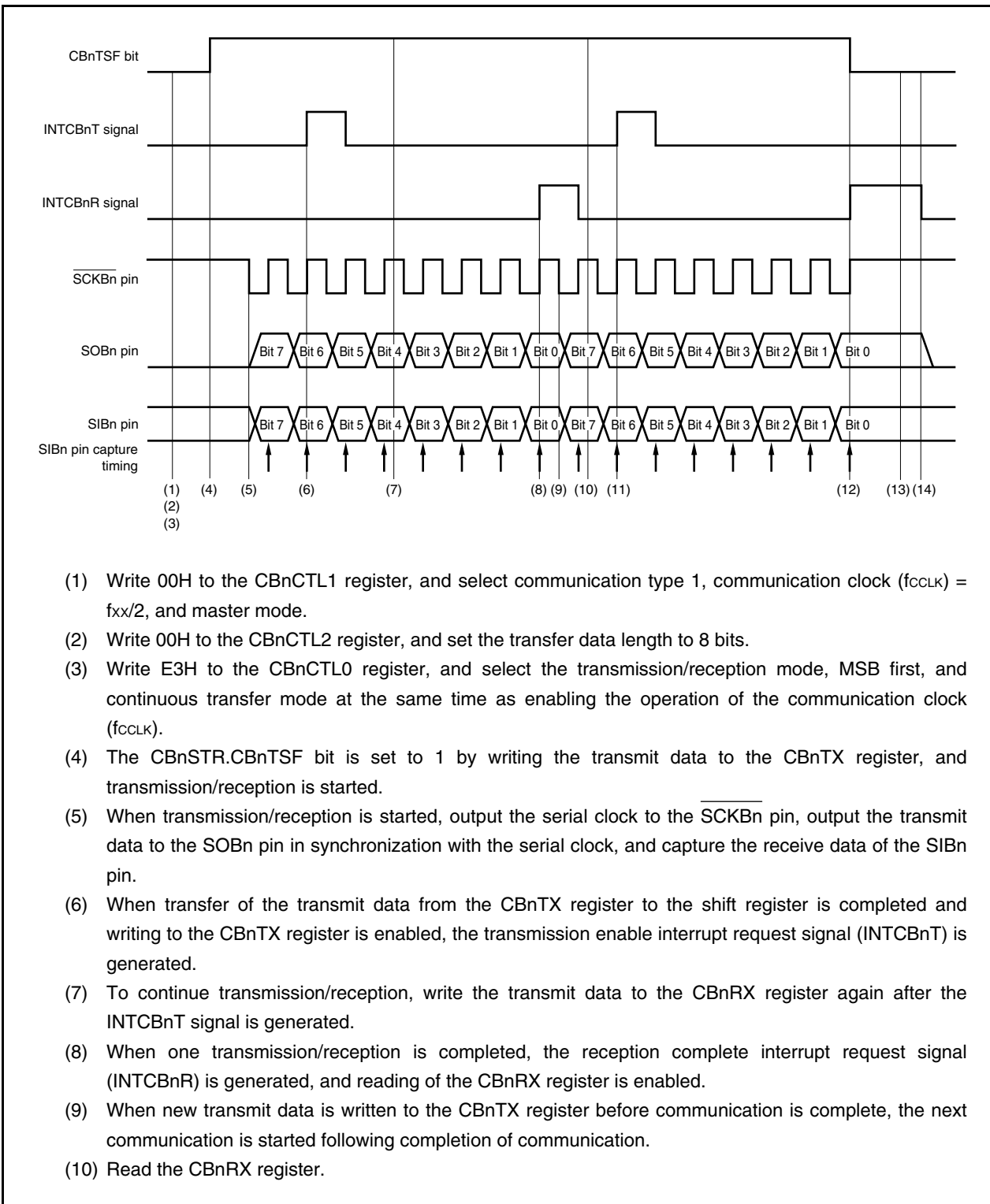
Figure 16-22. Continuous Transfer Mode Operation Timing (Master Mode, Transmission/Reception Mode) (1/2)

Figure 16-22. Continuous Transfer Mode Operation Timing (Master Mode, Transmission/Reception Mode) (2/2)

- (11) The transfer of the transmit data from the CBnTX register to the shift register is completed and the INTCBnT signal is generated. To end continuous transmission/reception with the current transmission/reception, do not write to the CBnTX register.
- (12) If the next transmit data is not written to the CBnTX register before transfer is complete, stop outputting the serial clock to the $\overline{\text{SCKBn}}$ pin and wait for the CBnTSF bit to be cleared to 0 after completion of transfer.
- (13) When the reception complete interrupt request signal (INTCBnR) is generated, read the CBnRX register.
- (14) To disable transmission/reception, clear the CBnCTL0.CBnPWR, CBnCTL0.CBnTXE, and CBnCTL0.CBnRXE bits to 0 after confirming that the CBnTSF bit is set to 0.

16.6.10 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_{CLK}) = external clock (\overline{SCKBn}) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

Figure 16-23. Continuous Transfer Mode Operation (Slave Mode, Transmission Mode)

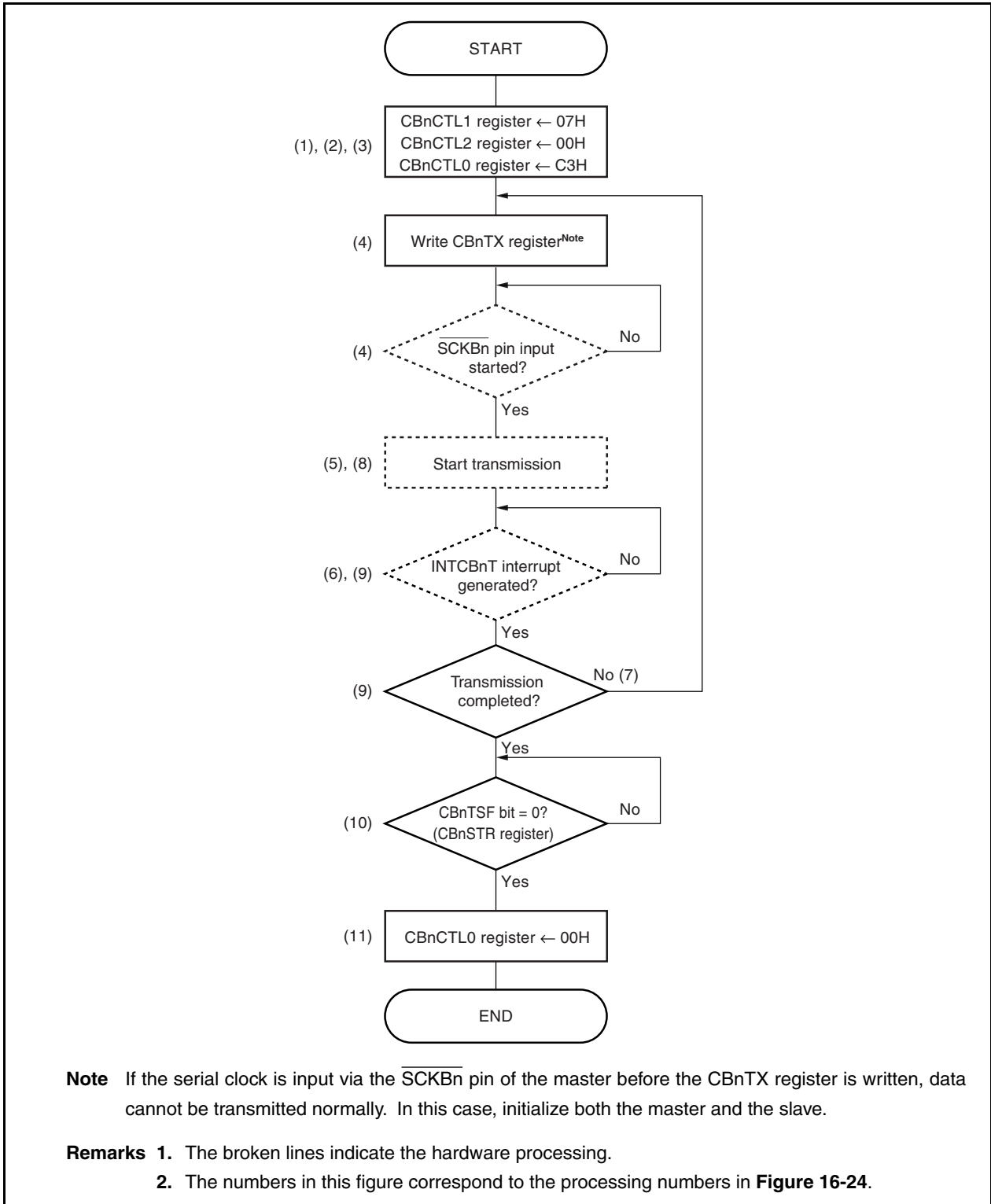
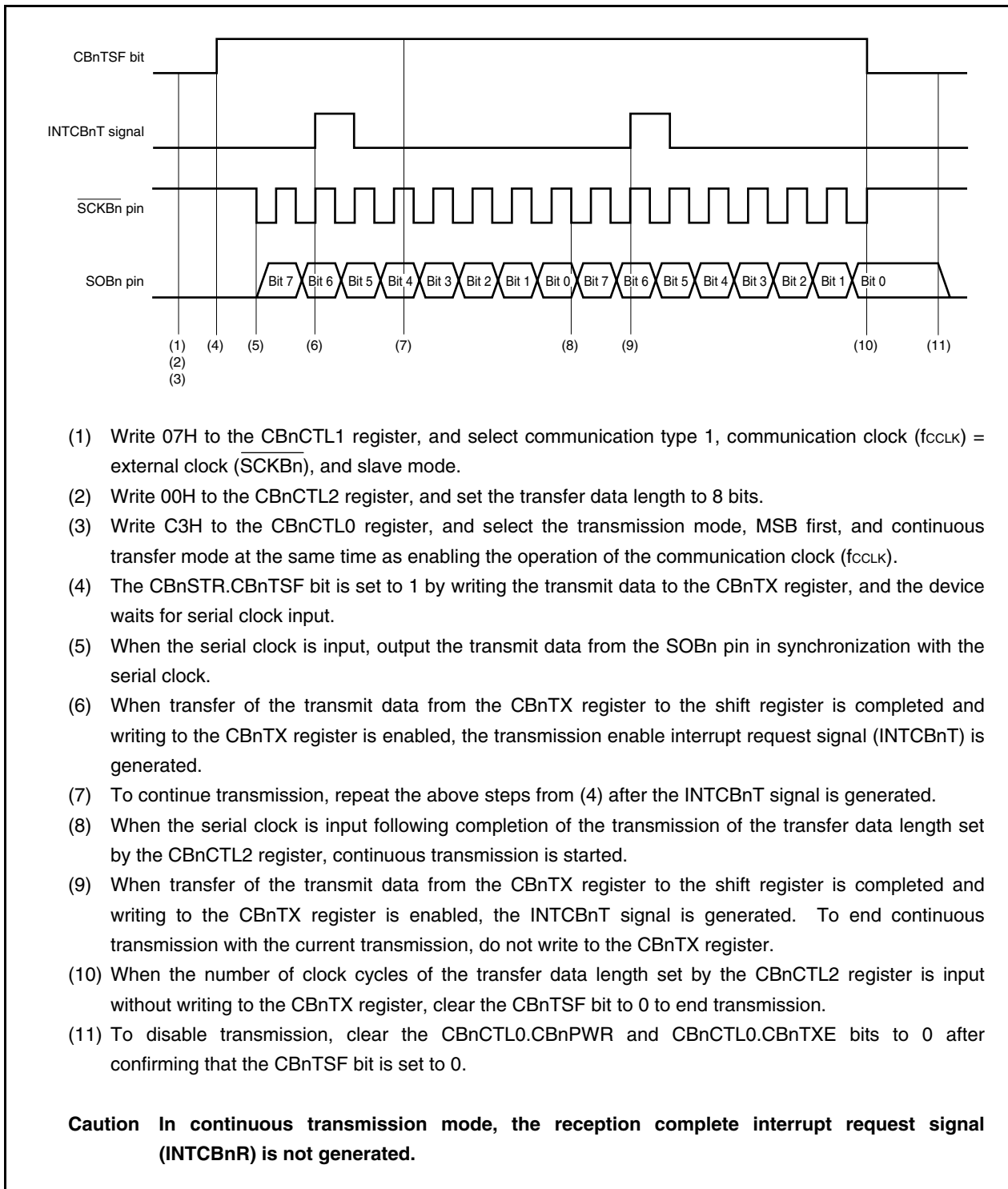


Figure 16-24. Continuous Transfer Mode Operation Timing (Slave Mode, Transmission Mode)

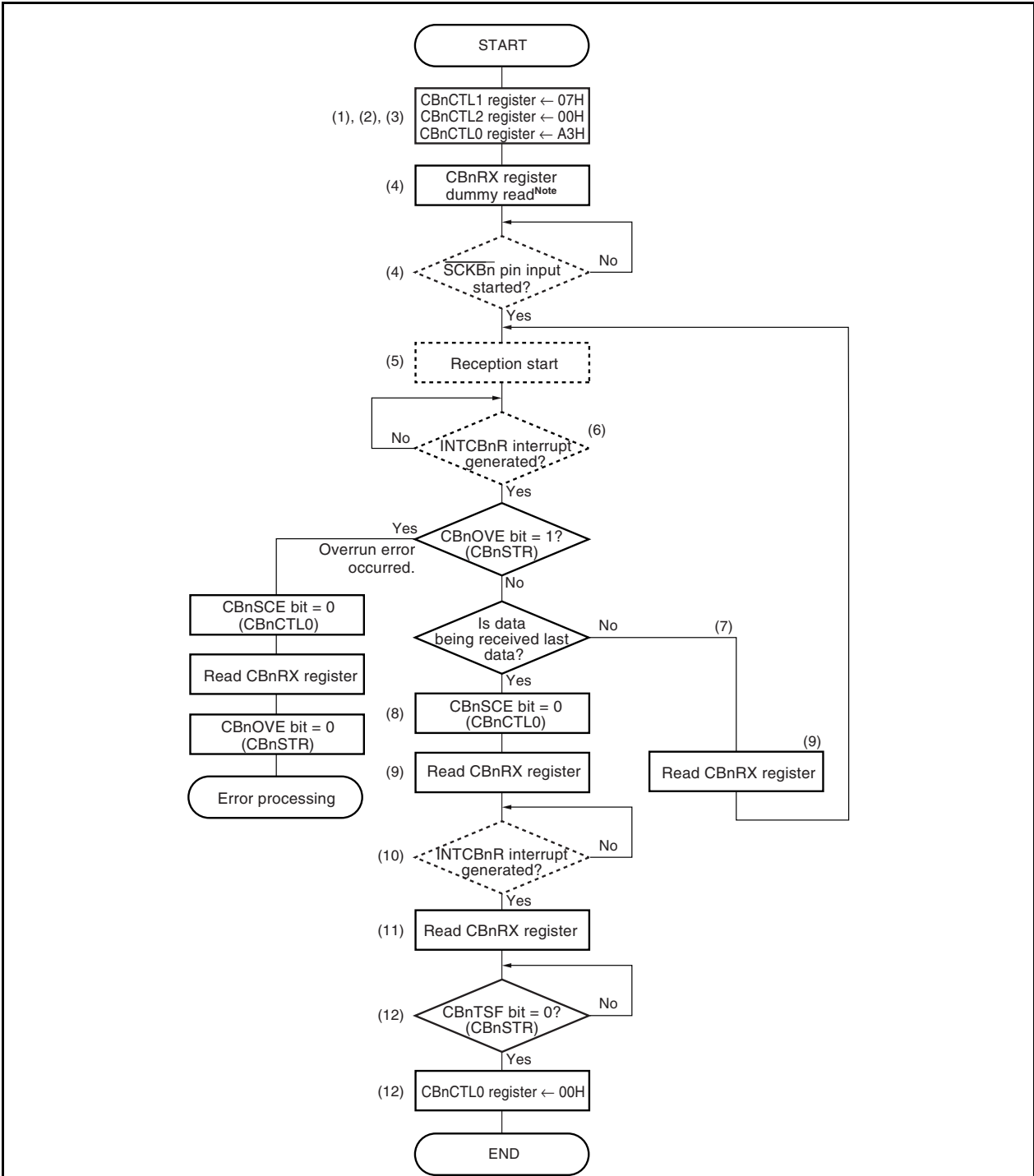
16.6.11 Continuous transfer mode (slave mode, reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CLK}) = external clock ($\overline{\text{SCKBn}}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

The flowchart in Figure 16-25 shows the operation where the specified number of data items are received in the slave mode. Operations are repeated until all the specified data items are received. If an overrun error occurs, however, transfer ends. Perform error processing as necessary. For details about the overrun error, see **16.6.13 Reception errors**.

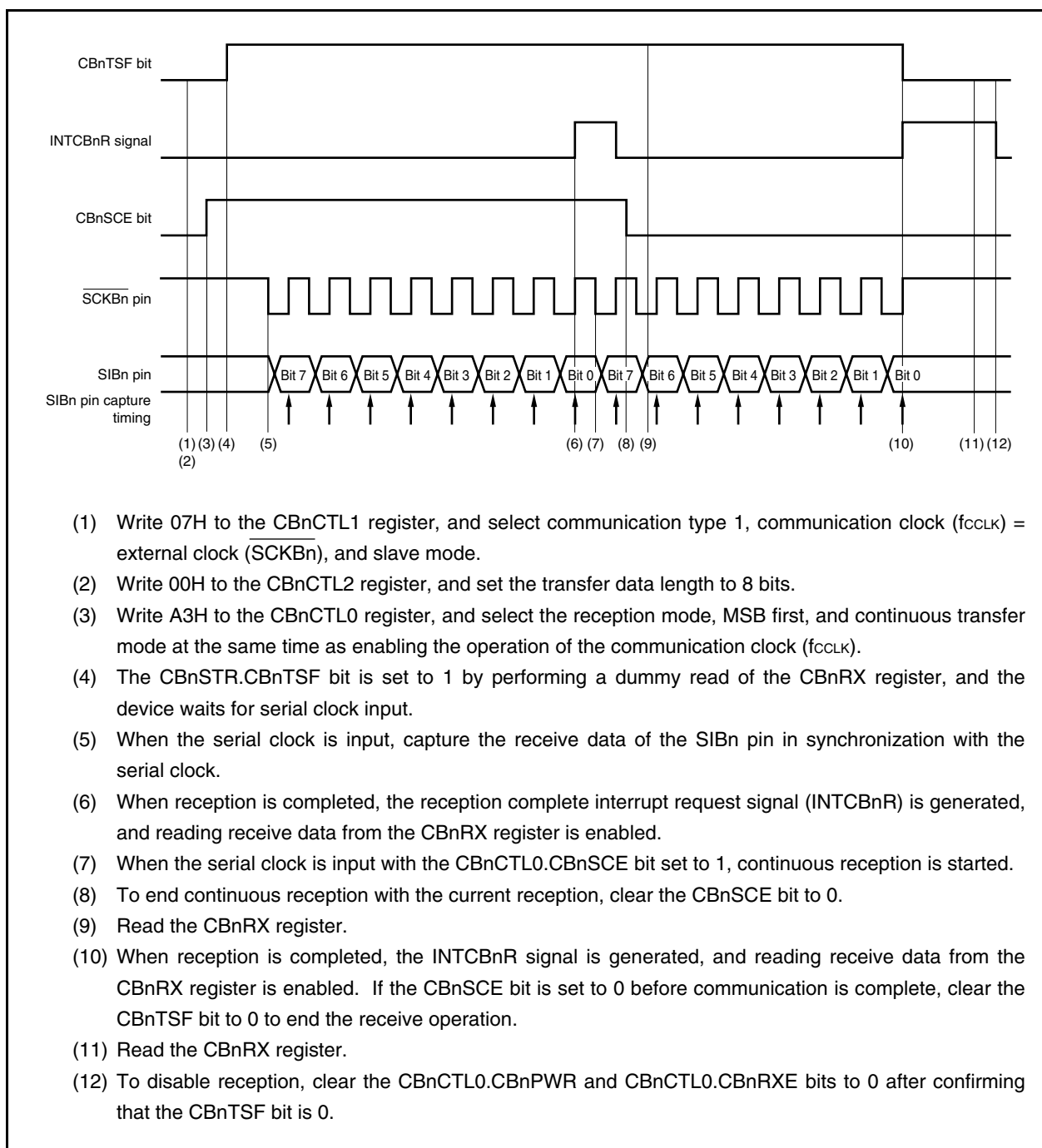
The operation timing in Figure 16-26 shows a case where no error occurred.

Figure 16-25. Continuous Transfer Mode Operation (Slave Mode, Reception Mode)



Note If the serial clock is input via the SCKBn pin of the master before a dummy-read of the CBnRX register is executed, data cannot be received normally. In this case, initialize both the master and the slave.

Remarks 1. The broken lines indicate the hardware processing.
 2. The numbers in this figure correspond to the processing numbers in Figure 16-26.

Figure 16-26. Continuous Transfer Mode Operation Timing (Slave Mode, Reception Mode)

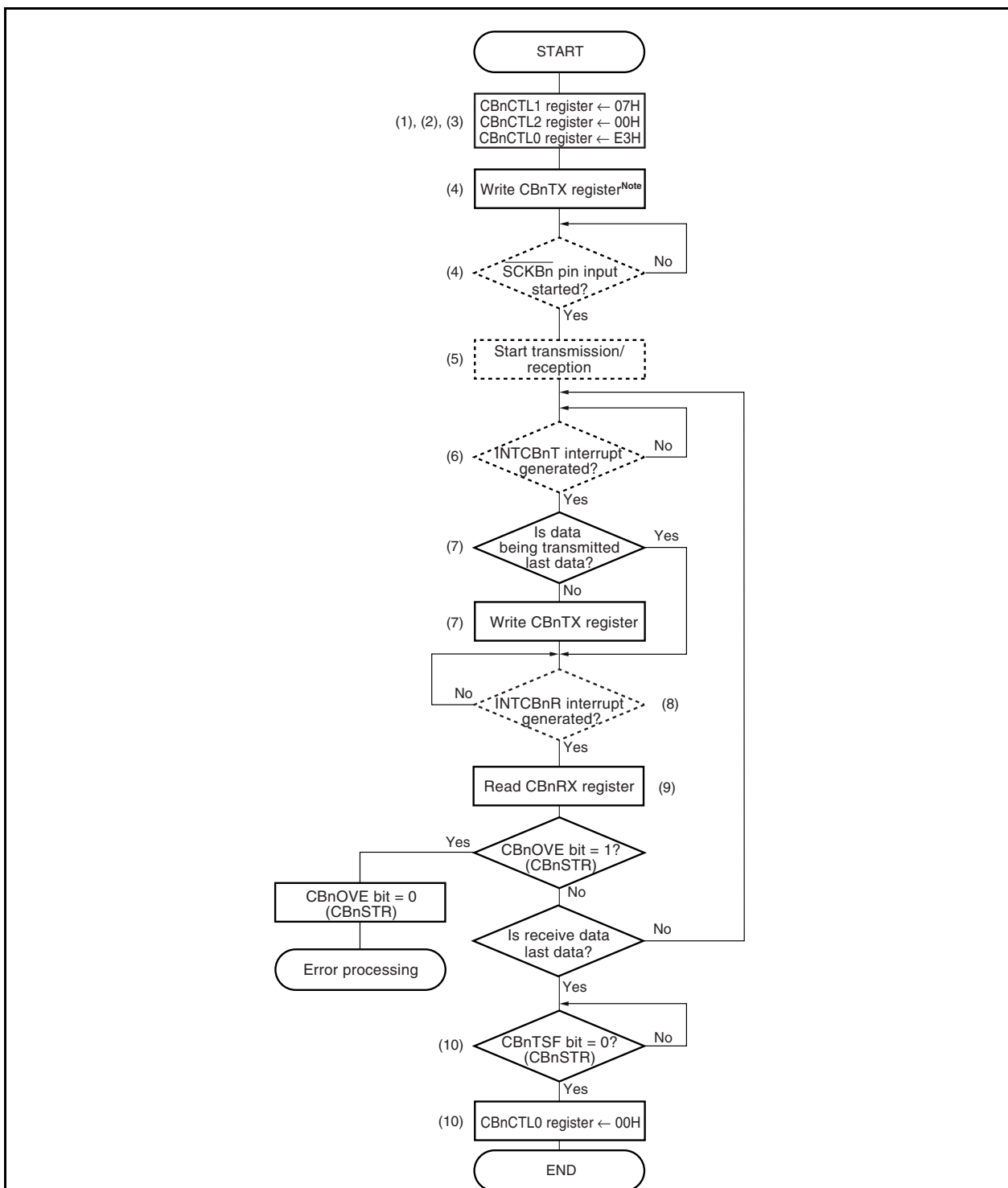
16.6.12 Continuous transfer mode (slave mode, transmission/reception mode)

MSB first (CBnCTL0.CBnDIR bit = 0), communication type 1 (CBnCTL1.CBnCKP and CBnCTL1.CBnDAP bits = 00), communication clock (f_{CLK}) = external clock ($\overline{\text{SCKBn}}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

The flowchart in Figure 16-27 shows the operation where the specified number of transmit/receive data items are transmitted/received in the slave mode. Operations are repeated until all the specified data items are transmitted/received. If an overrun error occurs, however, transfer ends. Perform error processing as necessary. For details about the overrun error, see **16.6.13 Reception errors**.

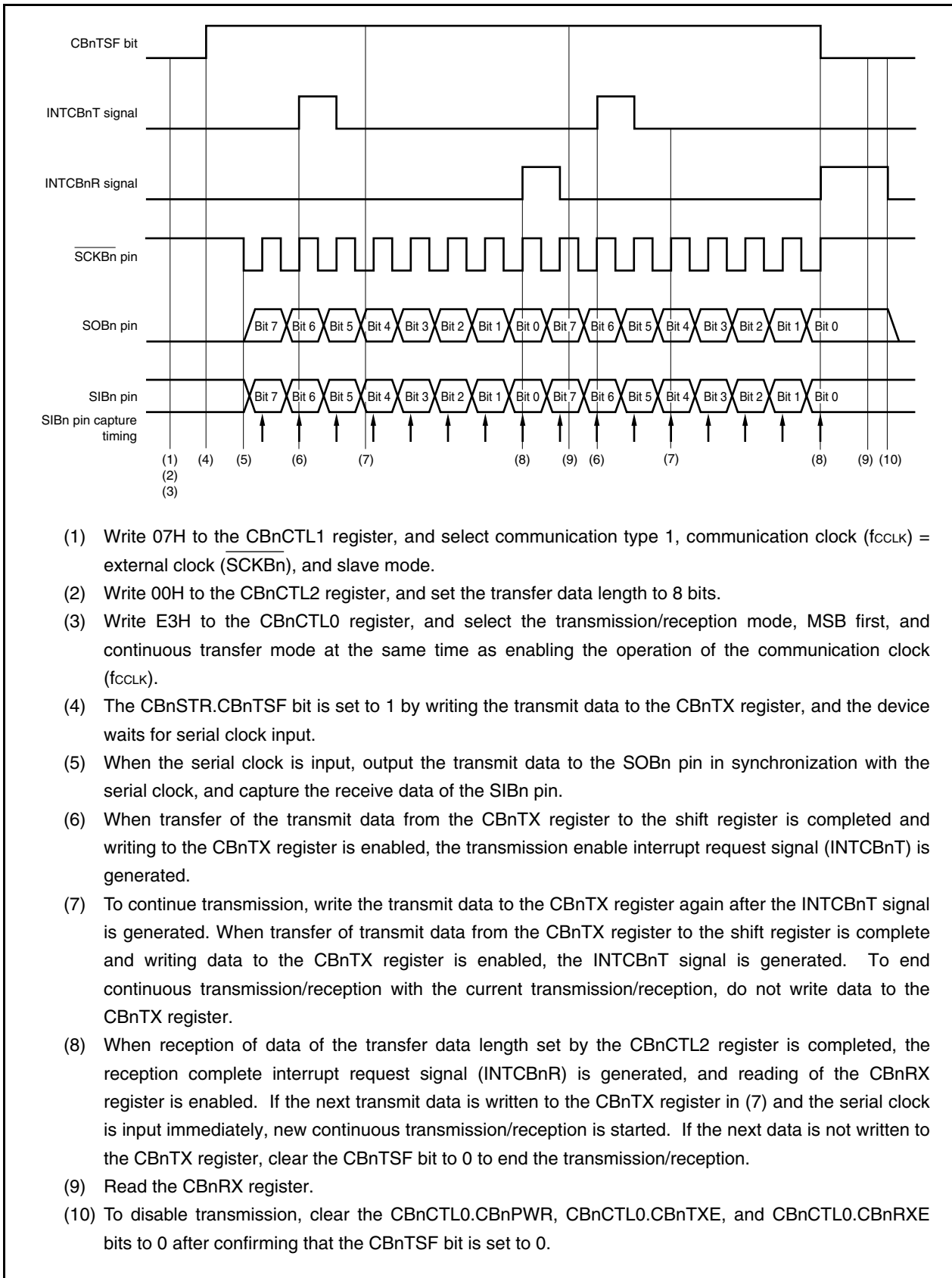
The operation timing in Figure 16-28 shows a case where no error occurred.

Figure 16-27. Continuous Transfer Mode Operation (Slave Mode, Transmission/Reception Mode)



Note If the serial clock is input via the \overline{SCKBn} pin of the master before the CBnTX register is written, data cannot be transmitted/received normally. In this case, initialize both the master and the slave.

- Remarks**
1. The broken lines indicate the hardware processing.
 2. The numbers in this figure correspond to the processing numbers in **Figure 16-28**.

Figure 16-28. Continuous Transfer Mode Operation Timing (Slave Mode, Transmission/Reception Mode)

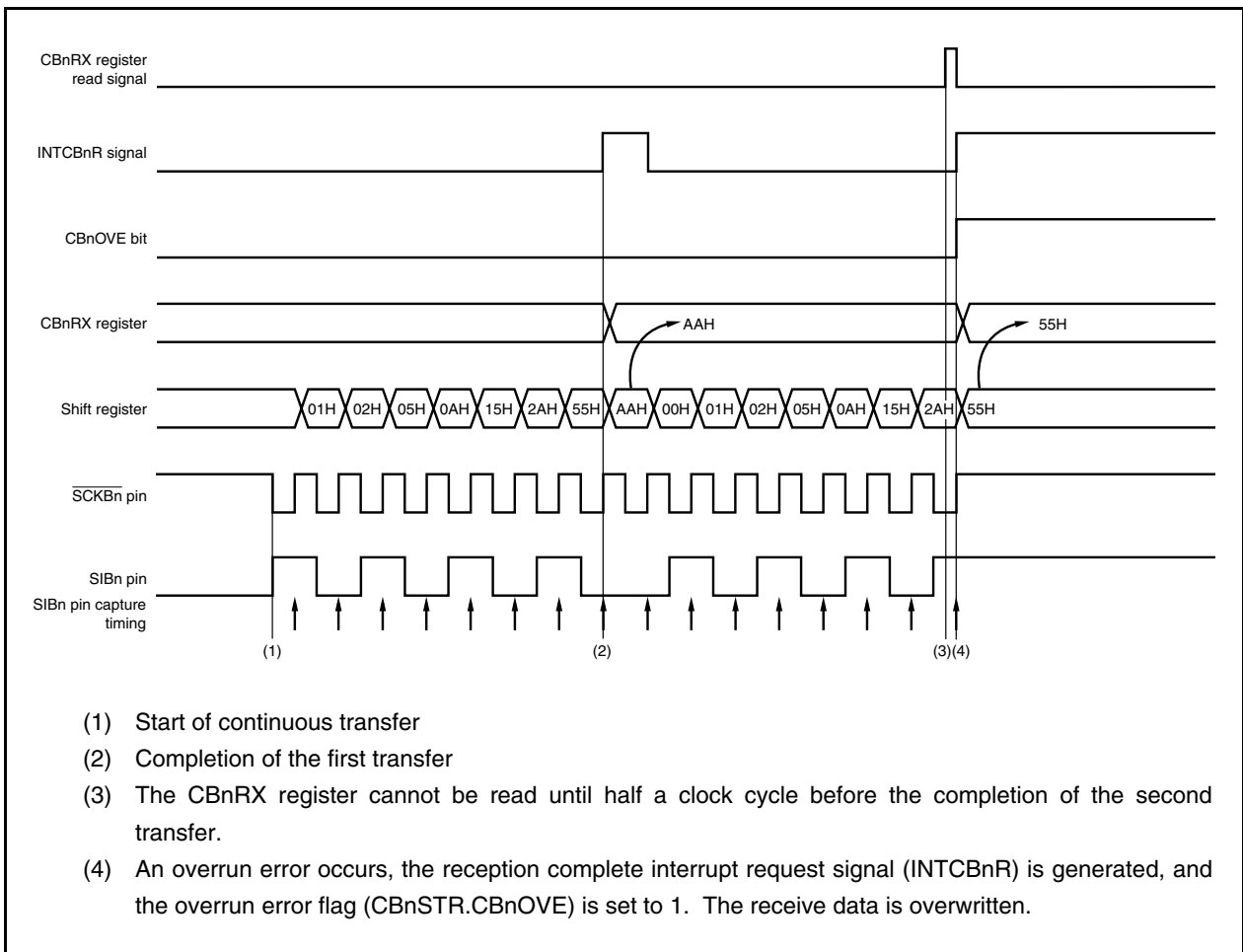
16.6.13 Reception errors

When transfer is performed with reception enabled (CBnCTL0.CBnRXE bit = 1) in the continuous transfer mode, the reception complete interrupt request signal (INTCBnR) is generated again if the next receive operation is completed before the CBnRX register is read after the INTCBnR signal is generated, and the overrun error flag (CBnSTR.CBnOVE) is set to 1.

If an overrun error occurs, the previous receive data is lost because the CBnRX register is updated. Even if a reception error occurs, the INTCBnR signal is generated again upon completion of the next reception if the CBnRX register is not read.

An overrun error occurs if reading the CBnRX register has not been completed half a clock cycle before the last bit of the next receive data is sampled after the INTCBnR signal is generated.

Figure 16-29. Overrun Error Timing



16.6.14 Clock timing

Figure 16-30. Clock Timing (1/2)

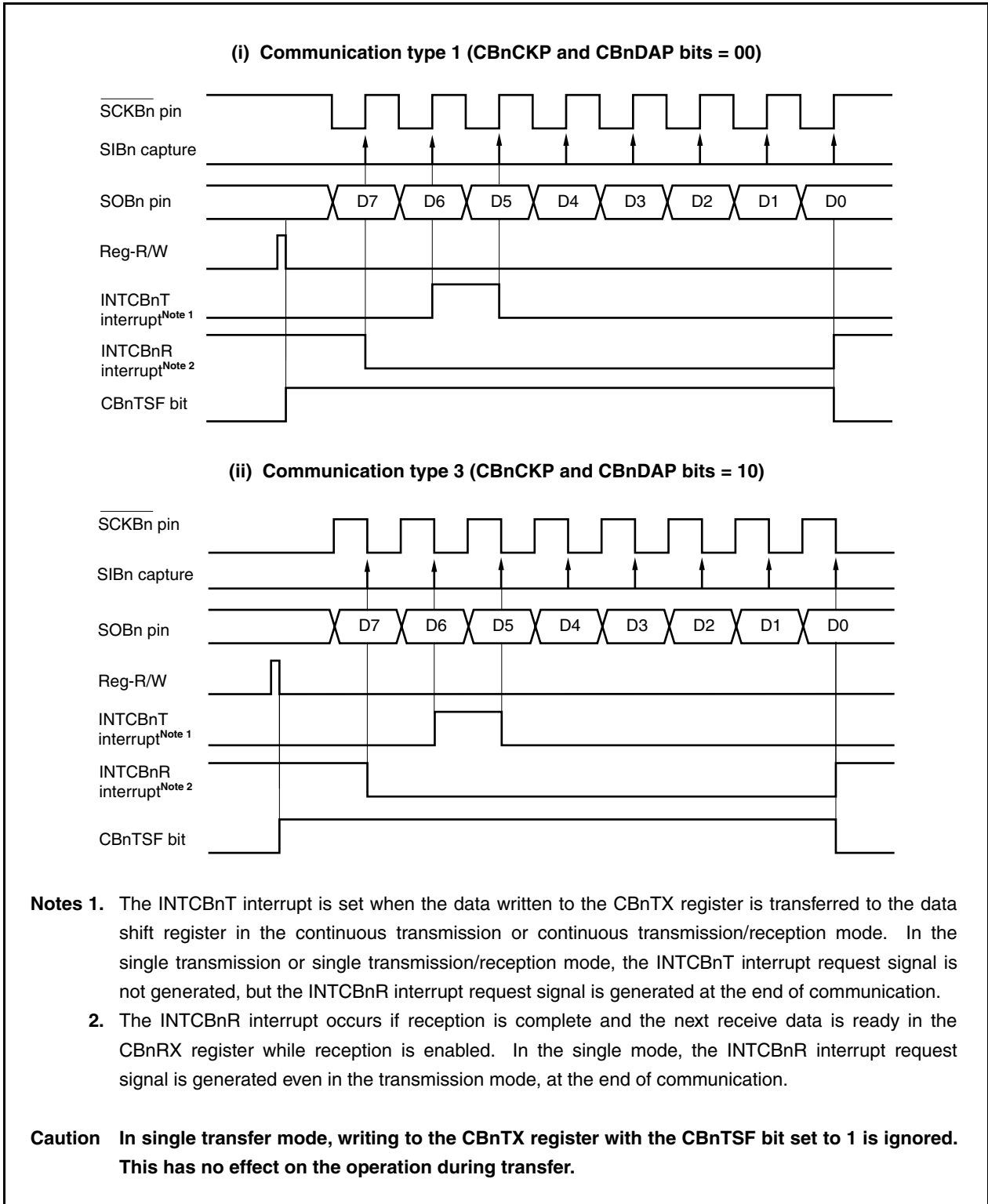
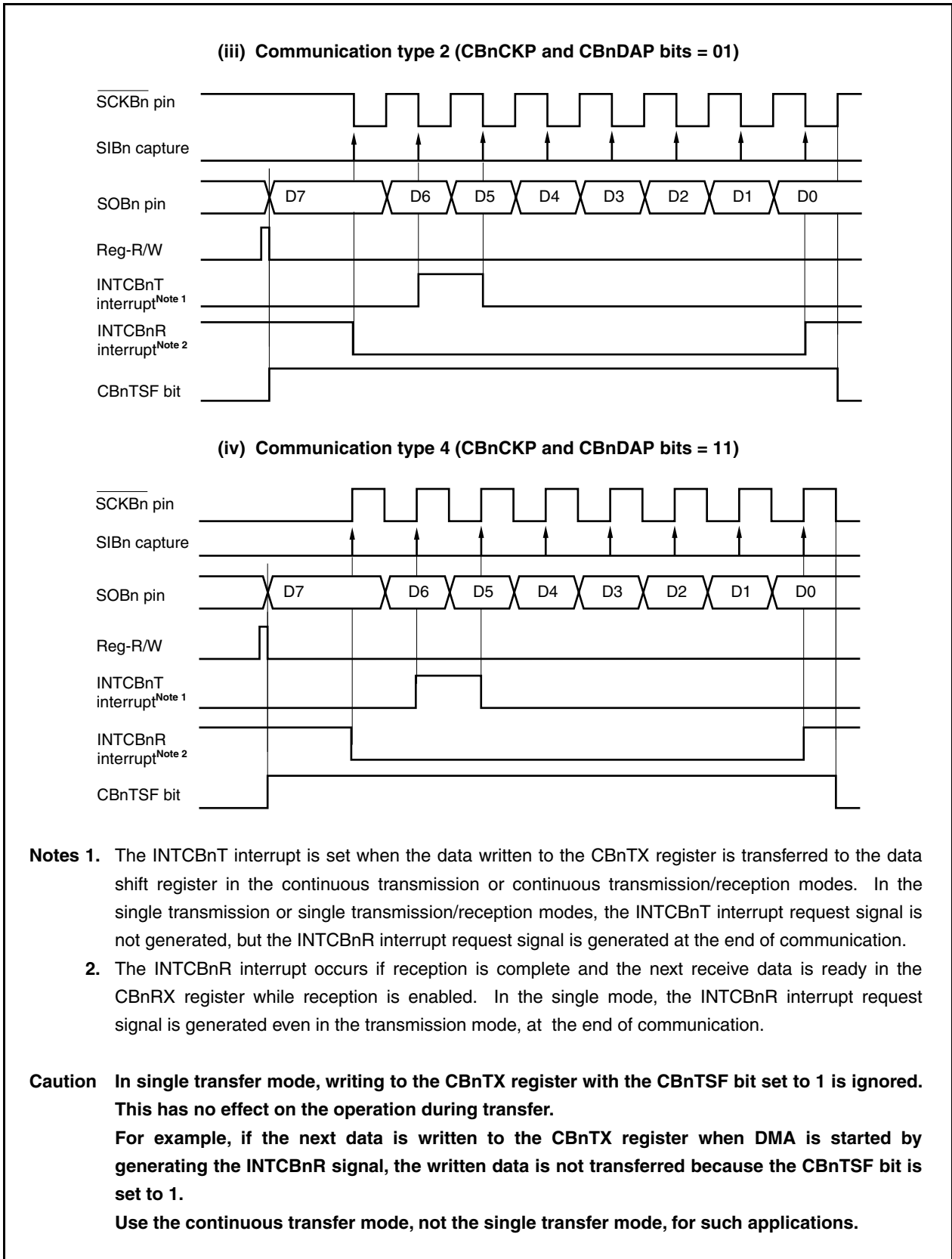


Figure 16-30. Clock Timing (2/2)



16.7 Output Pins

(1) $\overline{\text{SCKBn}}$ pin

When CSIBn is disabled (CBnCTL0.CBnPWR bit = 0), the $\overline{\text{SCKBn}}$ pin output status is as follows.

Table 16-4. $\overline{\text{SCKBn}}$ Pin Output Status with CSIBn Disabled

CBnCKP	CBnCKS2	CBnCKS1	CBnCKS0	$\overline{\text{SCKBn}}$ Pin Output
0	1	1	1	High impedance
	Other than above			High level
1	1	1	1	High impedance
	Other than above			Low level

Remark The output level of the $\overline{\text{SCKBn}}$ pin changes if any of the CBnCTL1.CBnCKP and CBnCKS2 to CBnCKS0 bits is rewritten.

(2) SOBn pin

When CSIBn is disabled (CBnPWR bit = 0), the SOBn pin output status is as follows.

Table 16-5. SOBn Pin Output Status with CSIBn Disabled

CBnTXE	CBnDAP	CBnDIR	SOBn Pin Output
0	×	×	Low level
1	0	×	Low level
	1	0	CBnTX0 value (MSB)
		1	CBnTX0 value (LSB)

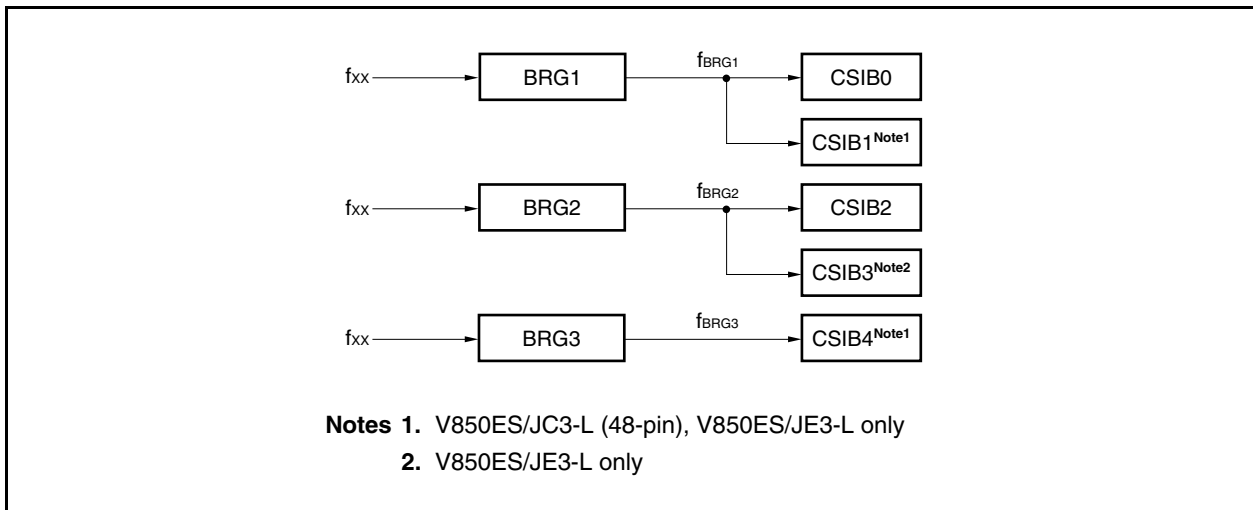
Remarks 1. The SOBn pin output changes when any one of the CBnCTL0.CBnTXE, CBnCTL0.CBnDIR, and CBnCTL1.CBnDAP bits is rewritten.

2. ×: Don't care

16.8 Baud Rate Generator

The BRG1 to BRG3 and CSIB0 to CSIB4 baud rate generators are connected as shown in the following block diagram.

Figure 16-31. Baud Rate Generator Connection



(1) Prescaler mode registers 1 to 3 (PRSM1 to PRSM3)

The PRSM1 to PRSM3 registers control generation of the baud rate signal for CSIBn.

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

Reset sets these registers to 00H.

After reset: 00H R/W Address: PRSM1 FFFF320H, PRSM2 FFFF324H, PRSM3 FFFF328H

	7	6	5	<4>	3	2	1	0
PRSMm (m = 1 to 3)	0	0	0	BGCEm	0	0	BGCSm1	BGCSm0

BGCEm	Baud rate output
0	Disabled
1	Enabled

BGCSm1	BGCSm0	Input clock selection (f_{BGCSm})	Setting value (k)
0	0	f_{xx}	0
0	1	$f_{xx}/2$	1
1	0	$f_{xx}/4$	2
1	1	$f_{xx}/8$	3

Cautions

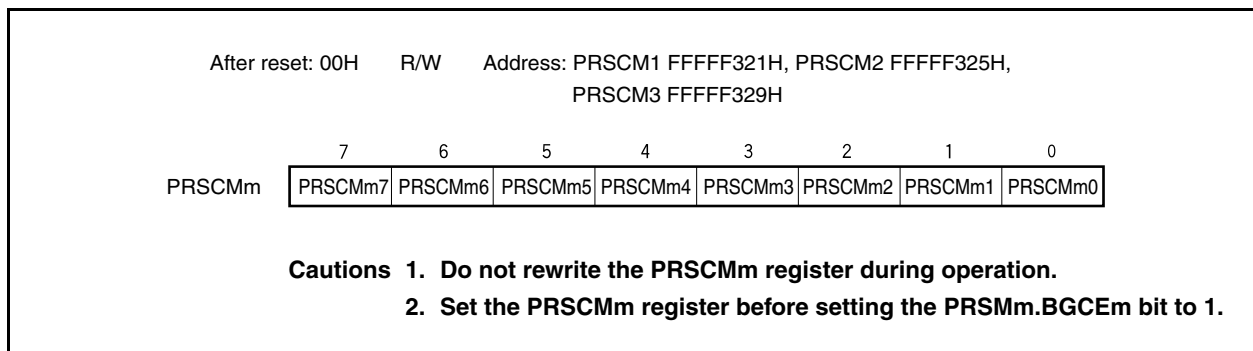
1. Do not rewrite the PRSMm register during operation.
2. Before setting the BGCEm bit to 1, set the BGCSm1 and BGCSm0 bits and prescaler compare registers 1 to 3 (PRSCM1 to PRSCM3).
3. Be sure to clear bits 7 to 5, 3 and 2 to “0”.

(2) Prescaler compare registers 1 to 3 (PRSCM1 to PRSCM3)

The PRSCM1 to PRSCM3 registers are 8-bit compare registers.

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

Reset sets these registers to 00H.

**16.8.1 Baud rate generation**

The transmission/reception clock is generated by dividing the main clock. The baud rate generated from the main clock is obtained by the following equation.

$$f_{BRGm} = \frac{f_{xx}}{2^{k+1} \times N}$$

Caution Set f_{BRGm} to 8 MHz or lower.

Remark f_{BRGm} : BRGm count clock

f_{xx} : Main clock oscillation frequency

k: PRSMm register setting value = 0 to 3

N: PRSCMm register setting value = 1 to 256

However, N = 256 only when PRSCMm register is set to 00H.

m = 1 to 3

16.9 Cautions

- (1) When transferring transmit data and receive data using DMA transfer, error processing cannot be performed even if an overrun error occurs during serial transfer. Check that the no overrun error has occurred by reading the CBnSTR.CBnOVE bit after DMA transfer is complete.
- (2) In regards to registers that are forbidden must not be rewritten during operations (when the CBnCTL0.CBnPWR bit is 1), if rewriting has been carried out by mistake, set the CBnCTL0.CBnPWR bit to 0 once, then initialize CSIBn.

The registers that must no be rewritten during operation are shown below.

- CBnCTL0 register: CBnTXE, CBnRXE, CBnDIR, CBnTMS bits
- CBnCTL1 register: CBnCKP, CBnDAP, CBnCKS2 to CBnCKS0 bits
- CBnCTL2 register: CBnCL3 to CBnCL0 bits

- (3) In communication type 2 or 4 (CBnCTL1.CBnDAP bit = 1), the CBnSTR.CBnTSF bit is cleared half a \overline{SCKBn} clock cycle after occurrence of a reception complete interrupt (INTCBnR).

In the single transfer mode, writing the next transmit data is ignored during communication (CBnTSF bit = 1), and the next communication is not started. Also if reception-only communication (CBnCTL0.CBnTXE bit = 0, CBnCTL0.CBnRXE bit = 1) is set, the next communication is not started even if the receive data is read during communication (CBnTSF bit = 1).

Therefore, when using the single transfer mode with communication type 2 or 4 (CBnDAP bit = 1), pay particular attention to the following.

- To start the next transmission, confirm that the CBnTSF bit is 0 and then write the transmit data to the CBnTX register.
- To perform the next reception continuously when reception-only communication (CBnTXE bit = 0, CBnRXE bit = 1) is set, confirm that the CBnTSF bit is 0 and then read the CBnRX register.

Or, use the continuous transfer mode instead of the single transfer mode. Use of the continuous transfer mode is recommended especially when using DMA.

CHAPTER 17 I²C BUS

To use the I²C bus function, set the P38/SDA00, P39/SCL00, P40/SDA01, P41/SCL01, P90/SDA02, and P91/SCL02 pins as the serial transmit/receive data I/O pins (SDA00 to SDA02) and serial clock I/O pins (SCL00 to SCL02), and set them to N-ch open-drain output.

The number of I²C channels in the V850ES/JC3-L, V850ES/JE3-L are shown below.

Product Name	V850ES/JC3-L (40-pin)	V850ES/JC3-L (48-pin)	V850ES/JE3-L
Number of channels	2 channels I ² C01, I ² C02	3 channels I ² C00 to I ² C02	3 channels I ² C00 to I ² C02

In this chapter, the number of channels is expressed as n.

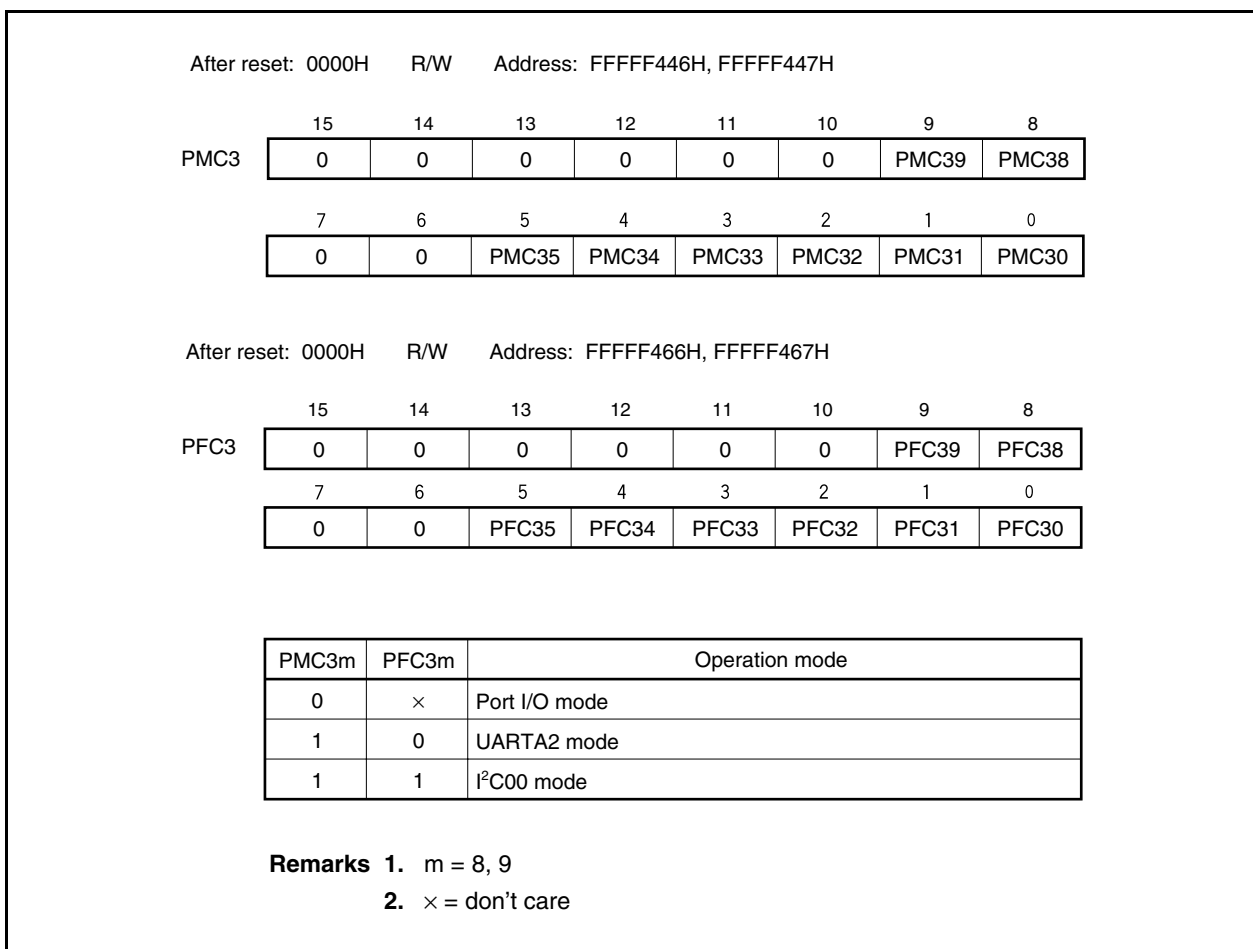
17.1 Mode Switching of I²C Bus and Other Serial Interfaces

17.1.1 UARTA2 and I²C00 mode switching

In the V850ES/JC3-L and V850ES/JE3-L, UARTA2 and I²C00 share pins and therefore cannot be used simultaneously. Set the operation mode to I²C00 in advance, using the PMC3 and PFC3 registers.

Caution The transmit/receive operation of UARTA2 and I²C00 is not guaranteed if these functions are switched during transmission or reception. Be sure to disable the one that is not used.

Figure 17-1. Switching UARTA2 and I²C00 Mode



17.1.2 CSIB0 and I²C01 mode switching

In the V850ES/JC3-L and V850ES/JE3-L, CSIB0 and I²C01 share pins and therefore cannot be used simultaneously. Set the operation mode to I²C01 in advance, using the PMC4 and PFC4 registers.

Caution The transmit/receive operation of CSIB0 and I²C01 is not guaranteed if these functions are switched during transmission or reception. Be sure to disable the one that is not used.

Figure 17-2. Switching CSIB0 and I²C01 Mode

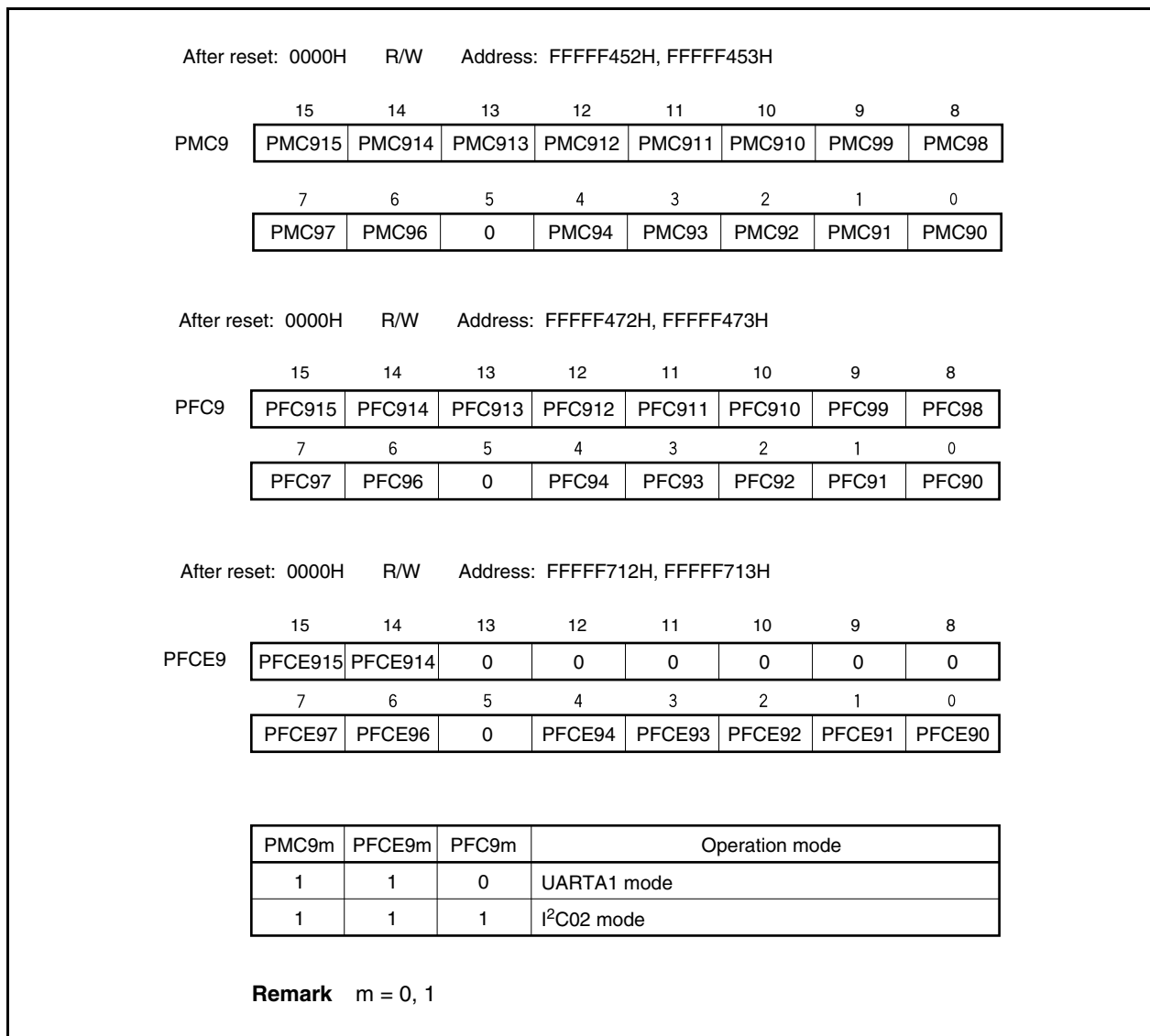
After reset: 00H	R/W	Address: FFFFF448H												
PMC4	7 6 5 4 3 2 1 0	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">0</td> <td style="width: 25%;">0</td> <td style="width: 25%;">0</td> <td style="width: 25%;">0</td> <td style="width: 25%;">0</td> <td style="width: 25%;">PMC42</td> <td style="width: 25%;">PMC41</td> <td style="width: 25%;">PMC40</td> </tr> </table>	0	0	0	0	0	PMC42	PMC41	PMC40				
0	0	0	0	0	PMC42	PMC41	PMC40							
After reset: 00H	R/W	Address: FFFFF468H												
PFC4	7 6 5 4 3 2 1 0	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;">0</td> <td style="width: 25%;">0</td> <td style="width: 25%;">0</td> <td style="width: 25%;">0</td> <td style="width: 25%;">0</td> <td style="width: 25%;">0</td> <td style="width: 25%;">PFC41</td> <td style="width: 25%;">PFC40</td> </tr> </table>	0	0	0	0	0	0	PFC41	PFC40				
0	0	0	0	0	0	PFC41	PFC40							
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">PMC4m</th> <th style="width: 15%;">PFC4m</th> <th style="width: 70%;">Operation mode</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td> <td style="text-align: center;">×</td> <td>Port I/O mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> <td>CSIB0 mode</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">1</td> <td>I²C01 mode</td> </tr> </tbody> </table>			PMC4m	PFC4m	Operation mode	0	×	Port I/O mode	1	0	CSIB0 mode	1	1	I ² C01 mode
PMC4m	PFC4m	Operation mode												
0	×	Port I/O mode												
1	0	CSIB0 mode												
1	1	I ² C01 mode												
<p>Remarks</p> <ol style="list-style-type: none"> 1. m = 0, 1 2. × = don't care 														

17.1.3 UARTA1 and I²C02 mode switching

In the V850ES/JC3-L and V850ES/JE3-L, UARTA1 and I²C02 share pins and therefore cannot be used simultaneously. Set the operation mode to I²C02 in advance, using the PMC9, PFC9, and PFCE9 registers.

Caution The transmit/receive operation of UARTA1 and I²C02 is not guaranteed if these functions are switched during transmission or reception. Be sure to disable the one that is not used.

Figure 17-3. Switching UARTA1 and I²C02 Mode



17.2 Features

I²C0n have the following two modes.

- Operation stopped mode
- I²C (Inter IC) bus mode (multimasters supported)

(1) Operation stopped mode

In this mode, serial transfers are not performed, thus enabling a reduction in power consumption.

(2) I²C bus mode (multimaster support)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock pin (SCL0n) and a serial data bus pin (SDA0n).

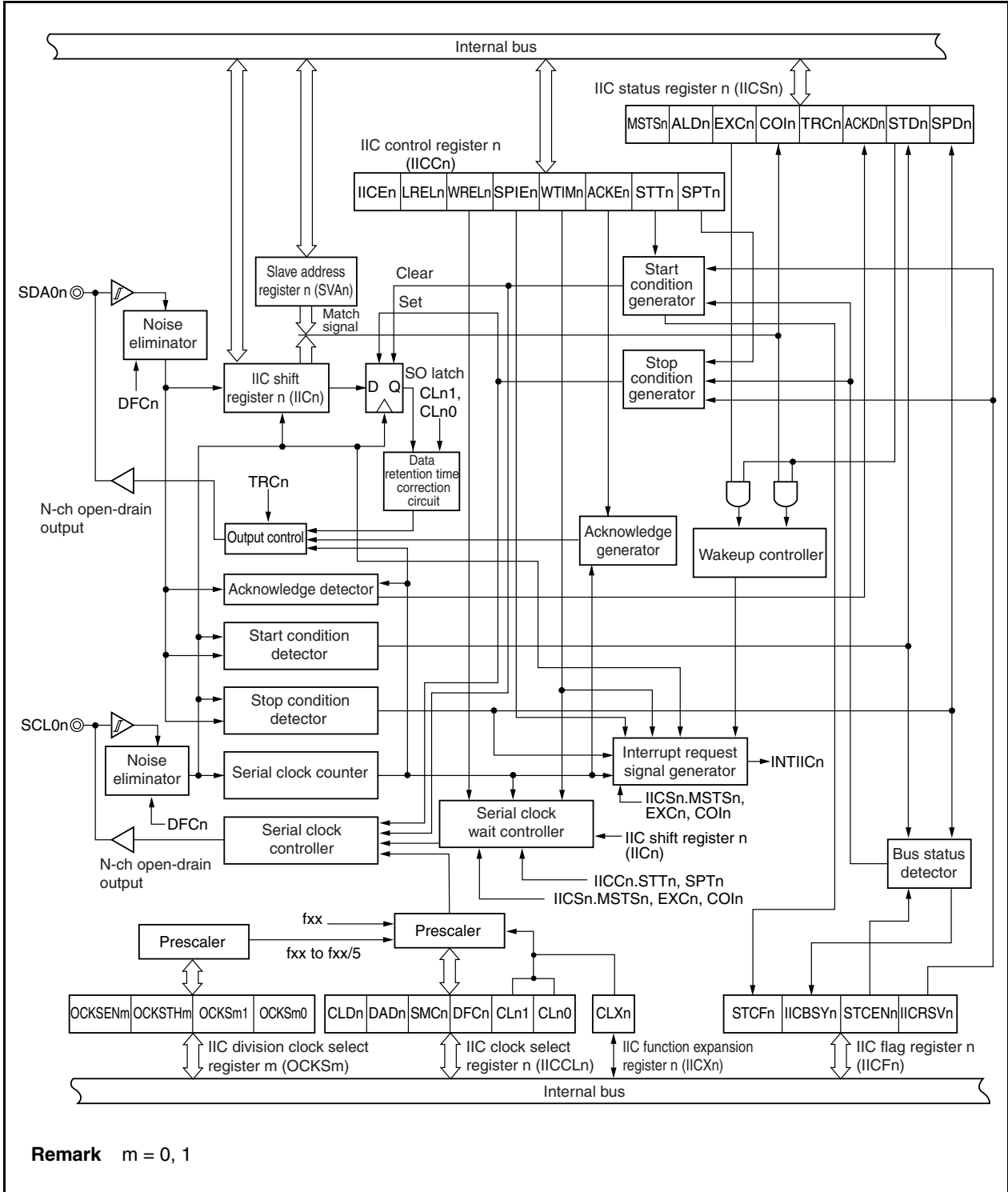
This mode complies with the I²C bus format and the master device can generate “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device via the serial data bus. The slave device automatically detects the received statuses and data by hardware. This function can simplify the part of an application program that controls the I²C bus.

Since SCL0n and SDA0n pins are used for N-ch open-drain outputs, I²C0n requires pull-up resistors for the serial clock line and the serial data bus line.

17.3 Configuration

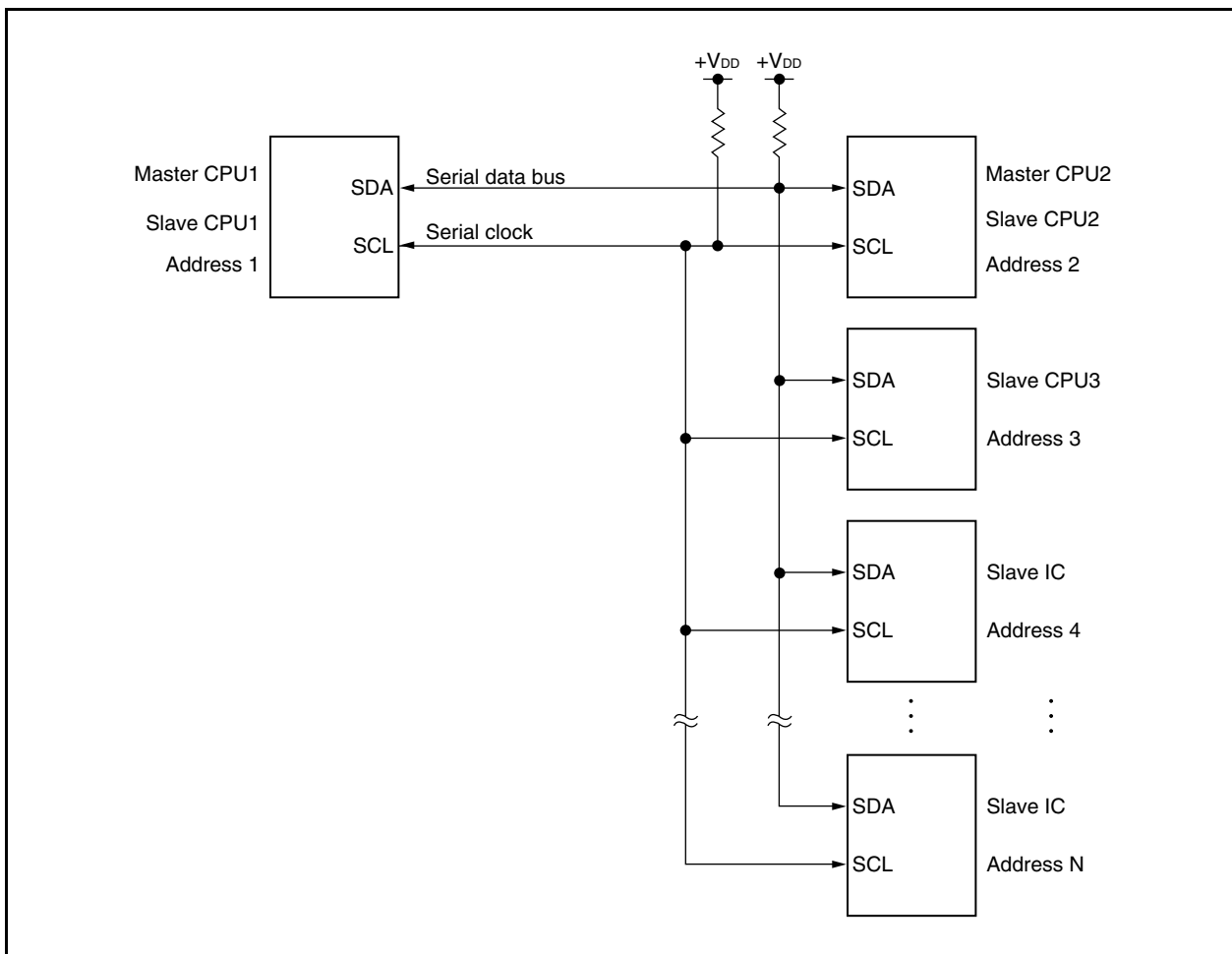
The block diagram of the I²C0n is shown below.

Figure 17-4. Block Diagram of I²C0n



A serial bus configuration example is shown below.

Figure 17-5. Serial Bus Configuration Example Using I²C Bus



I²C0n includes the following hardware.

Table 17-1. Configuration of I²C0n

Item	Configuration
Registers	IIC shift register n (IICn) Slave address register n (SVAn)
Control registers	IIC control register n (IICCN) IIC status register n (IICSn) IIC flag register n (IICF0n) IIC clock select register n (IICCLn) IIC function expansion register n (IICXn) IIC division clock select registers 0, 1 (OCKS0, OCKS1)

(1) IIC shift register n (IICn)

The IICn register converts 8-bit serial data into 8-bit parallel data and vice versa, and can be used for both transmission and reception.

Write and read operations to the IICn register are used to control the actual transmit and receive operations.

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

Reset sets this register to 00H.

(2) Slave address register n (SVAn)

The SVAn register sets local addresses when in slave mode.

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

Reset sets this register to 00H.

(3) SO latch

The SO latch is used to retain the output level of the SDA0n pin.

(4) Wakeup controller

This circuit generates an interrupt request signal (INTIICn) when the address received by this register matches the address value set to the SVAn 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 transmitted or received.

(7) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIICn).

An I²C interrupt is generated following either of two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by IICn.WTIMn bit)
- Interrupt occurrence due to stop condition detection (set by IICn.SPIEn bit)

(8) Serial clock controller

In master mode, this circuit generates the clock output via the SCL0n pin from the sampling clock.

(9) Serial clock wait controller

This circuit controls the wait timing.

(10) $\overline{\text{ACK}}$ generator, stop condition detector, start condition detector, and $\overline{\text{ACK}}$ detector

These circuits are used to generate and detect various statuses.

(11) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the SCL0n pin.

(12) Start condition generator

A start condition is generated when the IICn.STTn bit is set.

However, in the communication reservation disabled status (IICFn.IICRSVn bit = 1), this request is ignored and the IICFn.STCFn bit is set to 1 if the bus is not released (IICFn.IICBSYn bit = 1).

(13) Stop condition generator

A stop condition is generated when the IICn.SPTn bit is set.

(14) Bus status detector

Whether the bus is released or not is ascertained by detecting a start condition and stop condition.

However, the bus status cannot be detected immediately after operation, so set the bus status detector to the initial status by using the IICFn.STCENn bit.

17.4 Registers

I²C0n is controlled by the following registers.

- IIC control registers n (IICCN)
- IIC status registers n (IICSn)
- IIC flag registers n (IICFn)
- IIC clock select registers n (IICCLn)
- IIC function expansion registers n (IICXn)
- IIC division clock select registers 0, 1 (OCKS0, OCKS1)

The following registers are also used.

- IIC shift registers n (IICn)
- Slave address registers n (SVAn)

Remark For the alternate-function pin settings, see **Table 4-17 Settings When Pins Are Used for Alternate Functions**.

(1) IIC control registers n (IICCN)

The IICCN register enables/stops I²C0n operations, sets the wait timing, and sets other I²C operations.

These registers can be read or written in 8-bit or 1-bit units. However, set the SPIEn, WTIMn, and ACKEn bits when the IICEn bit is 0 or during the wait period. When setting the IICEn bit from “0” to “1”, these bits can also be set at the same time.

Reset sets these registers to 00H.

After reset: 00H R/W Address: IICC0 FFFF82H, IICC1 FFFF92H, IICC2 FFFFDA2H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICn	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

IICEn	Specification of I ² Cn operation enable/disable
0	Operation stopped. IICSn register reset ^{Note 1} . Internal operation stopped.
1	Operation enabled.
Be sure to set this bit to 1 when the SCL0n and SDA0n lines are high level.	
Condition for clearing (IICEn bit = 0)	
<ul style="list-style-type: none"> • Cleared by instruction • After reset 	Condition for setting (IICEn bit = 1)
<ul style="list-style-type: none"> • Set by instruction 	

LRELn ^{Note 2}	Exit from communications
0	Normal operation
1	This exits from the current communication operation and sets standby mode. This setting is automatically cleared after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCL0n and SDA0n lines are set to high impedance. The STTn and SPTn bits and the MSTSn, EXCn, COIn, TRCn, ACKDn, and STDn bits of the IICSn register are cleared.
The standby mode following exit from communications remains in effect until the following communication entry conditions are met.	
<ul style="list-style-type: none"> • After a stop condition is detected, restart is in master mode. • An address match occurs or an extension code is received after the start condition. 	
Condition for clearing (LRELn bit = 0)	
<ul style="list-style-type: none"> • Automatically cleared after execution • After reset 	Condition for setting (LRELn bit = 1)
<ul style="list-style-type: none"> • Set by instruction 	

WRELn ^{Note 2}	Wait state cancellation control
0	Wait state not canceled
1	Wait state canceled. This setting is automatically cleared after wait state is canceled.
Condition for clearing (WRELn bit = 0)	
<ul style="list-style-type: none"> • Automatically cleared after execution • After reset 	Condition for setting (WRELn bit = 1)
<ul style="list-style-type: none"> • Set by instruction 	

- Notes**
1. The IICSn register, IICFn.STCFn and IICFn.IICBSYn bits, and IICCLn.CLDn and IICCLn.DADn bits are reset.
 2. This flag's signal is invalid when the IICEn bit = 0.

Caution If the I²Cn operation is enabled (IICEn bit = 1) when the SCL0n line is high level and the SDA0n line is low level, the start condition is detected immediately. To avoid this, after enabling the I²Cn operation, immediately set the LRELn bit to 1 with a bit manipulation instruction.

Remark The LRELn and WRELn bits are 0 when read after the data has been set.

(2/4)

SPIEn ^{Note}	Enable/disable generation of interrupt request when stop condition is detected	
0	Disabled	
1	Enabled	
Condition for clearing (SPIEn bit = 0)		Condition for setting (SPIEn bit = 1)
<ul style="list-style-type: none"> • Cleared by instruction • After reset 		<ul style="list-style-type: none"> • Set by instruction

WTIMn ^{Note}	Control of wait state 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 the wait state is set. Slave mode: After input of eight clocks, the clock is set to low level and the wait state is set for the 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 the wait state is set. Slave mode: After input of nine clocks, the clock is set to low level and the wait state is set for the master device.	
During address transfer, an interrupt occurs at the falling edge of the ninth clock regardless of this bit setting. This bit setting becomes valid when the address transfer is completed. In master mode, a wait state is inserted at the falling edge of the ninth clock during address transfer. For a slave device that has received a local address, a wait state is inserted at the falling edge of the ninth clock after ACK is generated. When the slave device has received an extension code, however, a wait state is inserted at the falling edge of the eighth clock.		
Condition for clearing (WTIMn bit = 0)		Condition for setting (WTIMn bit = 1)
<ul style="list-style-type: none"> • Cleared by instruction • After reset 		<ul style="list-style-type: none"> • Set by instruction

ACKEn ^{Note}	Acknowledgment control	
0	Acknowledgment disabled.	
1	Acknowledgment enabled. During the ninth clock period, the SDA0n line is set to low level.	
The ACKEn bit setting is invalid for address reception by the slave device. In this case, $\overline{\text{ACK}}$ is generated when the addresses match. However, the ACKEn bit setting is valid for reception of the extension code. Set the ACKEn bit in the system that receives the extension code.		
Condition for clearing (ACKEn bit = 0)		Condition for setting (ACKEn bit = 1)
<ul style="list-style-type: none"> • Cleared by instruction • After reset 		<ul style="list-style-type: none"> • Set by instruction

Note This flag's signal is invalid when the IICEn bit = 0.

(3/4)

STTn	Start condition trigger				
0	Start condition is not generated.				
1	<p>When bus is released (in STOP mode): A start condition is generated (for starting as master). The SDA0n line is changed from high level to low level while the SCLn line is high level and then the start condition is generated. Next, after the rated amount of time has elapsed, the SCL0n line is changed to low level.</p> <p>During communication with a third party: If the communication reservation function is enabled (IICFn.IICRSVn bit = 0)</p> <ul style="list-style-type: none"> • This trigger functions as a start condition reserve flag. When set to 1, it releases the bus and then automatically generates a start condition. <p>If the communication reservation function is disabled (IICRSVn = 1)</p> <ul style="list-style-type: none"> • The IICFn.STCFn bit is set to 1 and information set (1) to the STTn bit is cleared. This trigger does not generate a start condition. <p>In the wait state (when master device): A restart condition is generated after the wait state is released.</p>				
<p>Cautions concerning set timing</p> <p>For master reception: Cannot be set to 1 during transfer. Can be set to 1 only when the ACKEn bit has been set to 0 and the slave has been notified of final reception.</p> <p>For master transmission: A start condition cannot be generated normally during the $\overline{\text{ACK}}$ period. Set to 1 during the wait period that follows output of the ninth clock.</p> <p>For slave: Even when the communication reservation function is disabled (IICRSVn bit = 1), the communication reservation status is entered.</p> <ul style="list-style-type: none"> • Setting to 1 at the same time as the SPTn bit is prohibited. • When the STTn bit is set to 1, setting the STTn bit to 1 again is disabled until the setting is cleared to 0. 					
<table border="1"> <thead> <tr> <th>Condition for clearing (STTn bit = 0)</th> <th>Condition for setting (STTn bit = 1)</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> • When the STTn bit is set to 1 in the communication reservation disabled status • Cleared by loss in arbitration • Cleared after start condition is generated by master device • When the LRELn bit = 1 (communication save) • When the IICEn bit = 0 (operation stop) • After reset </td> <td> <ul style="list-style-type: none"> • Set by instruction </td> </tr> </tbody> </table>		Condition for clearing (STTn bit = 0)	Condition for setting (STTn bit = 1)	<ul style="list-style-type: none"> • When the STTn bit is set to 1 in the communication reservation disabled status • Cleared by loss in arbitration • Cleared after start condition is generated by master device • When the LRELn bit = 1 (communication save) • When the IICEn bit = 0 (operation stop) • After reset 	<ul style="list-style-type: none"> • Set by instruction
Condition for clearing (STTn bit = 0)	Condition for setting (STTn bit = 1)				
<ul style="list-style-type: none"> • When the STTn bit is set to 1 in the communication reservation disabled status • Cleared by loss in arbitration • Cleared after start condition is generated by master device • When the LRELn bit = 1 (communication save) • When the IICEn bit = 0 (operation stop) • After reset 	<ul style="list-style-type: none"> • Set by instruction 				

Remark The STTn bit is 0 if it is read immediately after data setting.

(4/4)

SPTn	Stop condition trigger				
0	Stop condition is not generated.				
1	Stop condition is generated (termination of master device's transfer). After the SDA0n line goes to low level, either set the SCL0n line to high level or wait until the SCL0n pin goes to high level. Next, after the rated amount of time has elapsed, the SDA0n line is changed from low level to high level and a stop condition is generated.				
<p>Cautions concerning set timing</p> <p>For master reception: Cannot be set to 1 during transfer. Can be set to 1 only when the ACKEn bit has been set to 0 and during the wait period after the slave has been notified of final reception.</p> <p>For master transmission: A stop condition cannot be generated normally during the $\overline{\text{ACK}}$ reception period. Set to 1 during the wait period that follows output of the ninth clock.</p> <ul style="list-style-type: none"> • Cannot be set to 1 at the same time as the STTn bit. • The SPTn bit can be set to 1 only when in master mode^{Note}. • When the WTIMn bit has been set to 0, if the SPTn 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 WTIMn bit should be changed from 0 to 1 during the wait period following output of eight clocks, and the SPTn bit should be set to 1 during the wait period that follows output of the ninth clock. • When the SPTn bit is set to 1, setting the SPTn bit to 1 again is disabled until the setting is cleared to 0. 					
<table border="1"> <thead> <tr> <th>Condition for clearing (SPTn bit = 0)</th> <th>Condition for setting (SPTn bit = 1)</th> </tr> </thead> <tbody> <tr> <td> <ul style="list-style-type: none"> • Cleared by loss in arbitration • Automatically cleared after stop condition is detected • When the LRELn bit = 1 (communication save) • When the IICEn bit = 0 (operation stop) • After reset </td> <td> <ul style="list-style-type: none"> • Set by instruction </td> </tr> </tbody> </table>		Condition for clearing (SPTn bit = 0)	Condition for setting (SPTn bit = 1)	<ul style="list-style-type: none"> • Cleared by loss in arbitration • Automatically cleared after stop condition is detected • When the LRELn bit = 1 (communication save) • When the IICEn bit = 0 (operation stop) • After reset 	<ul style="list-style-type: none"> • Set by instruction
Condition for clearing (SPTn bit = 0)	Condition for setting (SPTn bit = 1)				
<ul style="list-style-type: none"> • Cleared by loss in arbitration • Automatically cleared after stop condition is detected • When the LRELn bit = 1 (communication save) • When the IICEn bit = 0 (operation stop) • After reset 	<ul style="list-style-type: none"> • Set by instruction 				

Note Set the SPTn bit to 1 only in master mode. However, when the IICRSVn bit is 0, the SPTn bit must be set to 1 and a stop condition generated before the first stop condition is detected following the switch to the operation enabled status. For details, see **17.15 Cautions**.

Caution When the TRCn bit = 1, the WRELn bit is set to 1 during the ninth clock and the wait state is canceled, after which the TRCn bit is cleared to 0 and the SDA0n line is set to high impedance.

Remarks The SPTn bit is 0 if it is read immediately after data setting.

(2) IIC status registers n (IICSn)

The IICSn register indicates the status of I²C0n.

These registers are read-only, in 8-bit or 1-bit units. However, the IICSn register can only be read when the IICn.STTn bit is 1 or during the wait period.

Reset sets these registers to 00H.

Caution Accessing the IICSn register is prohibited in the following statuses. For details, see 3.4.9 (1) Accessing specific on-chip peripheral I/O registers.

- When the CPU operates on the subclock and main clock oscillation is stopped
- When the CPU operates on the internal oscillator clock

(1/3)

After reset: 00H		R		Address: IICS0 FFFF86H, IICS1 FFFF96H, IICS2 FFFFDA6H				
	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICSn	MSTS _n	ALD _n	EXC _n	COL _n	TRC _n	ACKD _n	STD _n	SPD _n
MSTS _n	Master device status							
0	Slave device status or communication standby status							
1	Master device communication status							
Condition for clearing (MSTS _n bit = 0)				Condition for setting (MSTS _n bit = 1)				
<ul style="list-style-type: none"> • When a stop condition is detected • When the ALD_n bit = 1 (arbitration loss) • Cleared by LREL_n bit = 1 (communication save) • When the IICEn bit changes from 1 to 0 (operation stop) • After reset 				<ul style="list-style-type: none"> • When a start condition is generated 				
ALD _n	Arbitration loss detection							
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 _n bit is cleared to 0.							
Condition for clearing (ALD _n bit = 0)				Condition for setting (ALD _n bit = 1)				
<ul style="list-style-type: none"> • Automatically cleared after the IICS_n register is read^{Note} • When the IICEn bit changes from 1 to 0 (operation stop) • After reset 				<ul style="list-style-type: none"> • When the arbitration result is a “loss”. 				
EXC _n	Detection of extension code reception							
0	Extension code was not received.							
1	Extension code was received.							
Condition for clearing (EXC _n bit = 0)				Condition for setting (EXC _n bit = 1)				
<ul style="list-style-type: none"> • When a start condition is detected • When a stop condition is detected • Cleared by LREL_n bit = 1 (communication save) • When the IICEn bit changes from 1 to 0 (operation stop) • After reset 				<ul style="list-style-type: none"> • When the higher four bits of the received address data are either “0000” or “1111” (set at the rising edge of the eighth clock). 				
<p>Note This bit is also cleared when a bit manipulation instruction is executed for another bit in the IICS_n register.</p>								

(2/3)

COIn	Matching address detection	
0	Addresses do not match.	
1	Addresses match.	
Condition for clearing (COIn bit = 0)		Condition for setting (COIn bit = 1)
<ul style="list-style-type: none"> When a start condition is detected When a stop condition is detected Cleared by LRELn bit = 1 (communication save) When the IICEn bit changes from 1 to 0 (operation stop) After reset 		<ul style="list-style-type: none"> When the received address matches the local address (SVAn register) (set at the rising edge of the eighth clock).

TRCn	Transmit/receive status detection	
0	Receive status (other than transmit status). The SDA0n line is set to high impedance.	
1	Transmit status. The value in the SO latch is enabled for output to the SDA0n line (valid starting at the falling edge of the first byte's ninth clock).	
Condition for clearing (TRCn bit = 0)		Condition for setting (TRCn bit = 1)
<ul style="list-style-type: none"> When a stop condition is detected Cleared by LRELn bit = 1 (communication save) When the IICEn bit changes from 1 to 0 (operation stop) Cleared by IICn.WRELn bit = 1^{Note} When the ALDn bit changes from 0 to 1 (arbitration loss) After reset 		<p>Master</p> <ul style="list-style-type: none"> When a start condition is generated When "0" is output to the first byte's LSB (transfer direction specification bit) <p>Slave</p> <ul style="list-style-type: none"> When "1" is input by the first byte's LSB (transfer direction specification bit)
<p>Master</p> <ul style="list-style-type: none"> When "1" is output to the first byte's LSB (transfer direction specification bit) <p>Slave</p> <ul style="list-style-type: none"> When a start condition is detected <p>When not used for communication</p>		

ACKDn	$\overline{\text{ACK}}$ detection	
0	$\overline{\text{ACK}}$ was not detected.	
1	$\overline{\text{ACK}}$ was detected.	
Condition for clearing (ACKDn bit = 0)		Condition for setting (ACKD bit = 1)
<ul style="list-style-type: none"> When a stop condition is detected At the rising edge of the next byte's first clock Cleared by LRELn bit = 1 (communication save) When the IICEn bit changes from 1 to 0 (operation stop) After reset 		<ul style="list-style-type: none"> After the SDA0n bit is set to low level at the rising edge of the SCL0n pin's ninth clock

Note The TRCn bit is cleared to 0 and SDA0n line becomes high impedance when the WRELn bit is set to 1 and the wait state is canceled to 0 at the ninth clock by TRCn bit = 1.

(3/3)

STDn	Start condition detection	
0	Start condition was not detected.	
1	Start condition was detected. This indicates that the address transfer period is in effect	
Condition for clearing (STDn bit = 0)		Condition for setting (STDn bit = 1)
<ul style="list-style-type: none"> • When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LRELn bit = 1 (communication save) • When the IICEn bit changes from 1 to 0 (operation stop) • After reset 		<ul style="list-style-type: none"> • When a start condition is detected

SPDn	Stop condition detection	
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 (SPDn bit = 0)		Condition for setting (SPDn bit = 1)
<ul style="list-style-type: none"> • 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 IICEn bit changes from 1 to 0 (operation stop) • After reset 		<ul style="list-style-type: none"> • When a stop condition is detected

(3) IIC flag registers n (IICFn)

The IICFn register sets the I²C0n operation mode and indicates the I²C bus status.

These registers can be read or written in 8-bit or 1-bit units. However, the STCFn and IICBSYn bits are read-only.

IICRSVn enables/disables the communication reservation function (see **17.14 Communication Reservation**).

The initial value of the IICBSYn bit is set by using the STCENn bit (see **17.15 Cautions**).

The IICRSVn and STCENn bits can be written only when operation of I²C0n is disabled (IICn.IICEn bit = 0). After operation is enabled, IICFn can be read.

Reset sets these registers to 00H.

After reset: 00H R/W^{Note} Address: IICF0 FFFFFFFD8AH, IICF1 FFFFFFFD9AH, IICF2 FFFFFFFDAAH

	<7>	<6>	5	4	3	2	<1>	<0>
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

STCFn	STTn bit clear
0	Start condition issued
1	Start condition cannot be issued, STTn bit cleared
Condition for clearing (STCFn bit = 0)	
<ul style="list-style-type: none"> • Cleared by IICn.STTn bit = 1 • When the IICn.IICEn bit = 0 • After reset 	
Condition for setting (STCFn bit = 1)	
<ul style="list-style-type: none"> • When start condition is not issued and STTn flag is cleared to 0 during communication reservation is disabled (IICRSVn bit = 1). 	

IICBSYn	I ² C0n bus status
0	Bus released status (default communication status when STCENn bit = 1)
1	Bus communication status (default communication status when STCENn bit = 0)
Condition for clearing (IICBSYn bit = 0)	
<ul style="list-style-type: none"> • When stop condition is detected • When the IICEn bit = 0 • After reset 	
Condition for setting (IICBSYn bit = 1)	
<ul style="list-style-type: none"> • When start condition is detected • By setting the IICEn bit when the STCENn bit = 0 	

STCENn	Initial start enable trigger
0	Start conditions cannot be generated until a stop condition is detected following operation enable (IICEn bit = 1).
1	Start conditions can be generated even if a stop condition is not detected following operation enable (IICEn bit = 1).
Condition for clearing (STCENn bit = 0)	
<ul style="list-style-type: none"> • When start condition is detected • After reset 	
Condition for setting (STCENn bit = 1)	
<ul style="list-style-type: none"> • Setting by instruction 	

IICRSVn	Communication reservation function disable bit
0	Communication reservation enabled
1	Communication reservation disabled
Condition for clearing (IICRSVn bit = 0)	
<ul style="list-style-type: none"> • Clearing by instruction • After reset 	
Condition for setting (IICRSVn bit = 1)	
<ul style="list-style-type: none"> • Setting by instruction 	

Note Bits 6 and 7 are read-only bits.

- Cautions**
1. Write the STCENn bit only when operation is stopped (IICEn bit = 0).
 2. When the STCENn bit = 1, the bus released status (IICBSYn bit = 0) is recognized regardless of the actual bus status immediately after the I²Cn bus operation is enabled. Therefore, to issue the first start condition (STTn bit = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.
 3. Write the IICRSVn bit only when operation is stopped (IICEn bit = 0).

(4) IIC clock select registers n (IICCLn)

The IICCLn register sets the transfer clock for I²C0n.

These registers can be read or written in 8-bit or 1-bit units. However, the CLDn and DADn bits are read-only.

Set the IICCLn register when the IICCN.IICEN bit = 0.

The SMCn, CLn1, and CLn0 bits are set by the combination of the IICXn.CLXn bit and the OCKSTHm, OCKSm1, and OCKSm0 bits of the OCKSm register (see 17.4 (6) I²C0n transfer clock setting method) (m = 0, 1).

Reset sets these registers to 00H.

After reset: 00H	R/W ^{Note}	Address: IICCL0 FFFF84H, IICCL1 FFFF94H, IICCL2 FFFFDA4H							
		7	6	<5>	<4>	3	2	1	0
IICCLn		0	0	CLDn	DADn	SMCn	DFCn	CLn1	CLn0
CLDn		Detection of SCL0n pin level (valid only when IICCN.IICEN bit = 1)							
0		The SCL0n pin was detected at low level.							
1		The SCL0n pin was detected at high level.							
Condition for clearing (CLDn bit = 0)				Condition for setting (CLDn bit = 1)					
<ul style="list-style-type: none"> When the SCL0n pin is at low level When the IICEN bit = 0 (operation stop) After reset 				<ul style="list-style-type: none"> When the SCL0n pin is at high level 					
DADn		Detection of SDA0n pin level (valid only when IICCN.IICEN bit = 1)							
0		The SDA0n pin was detected at low level.							
1		The SDA0n pin was detected at high level.							
Condition for clearing (DADn bit = 0)				Condition for setting (DADn bit = 1)					
<ul style="list-style-type: none"> When the SDA0n pin is at low level When the IICEN bit = 0 (operation stop) After reset 				<ul style="list-style-type: none"> When the SDA0n pin is at high level 					
SMCn		Operation mode switching							
0		Operation in standard mode							
1		Operation in high-speed mode							
DFCn		Digital filter operation control							
0		Digital filter off							
1		Digital filter on							
<p>The digital filter can be used only in high-speed mode. In high-speed mode, the transfer clock does not vary regardless of the DFCn bit setting (on/off). The digital filter is used to eliminate noise in high-speed mode.</p>									
Note Bits 4 and 5 are read-only bits.									
Caution Be sure to clear bits 7 and 6 to "0".									
Remark When the IICCN.IICEN bit = 0, 0 is read when reading the CLDn and DADn bits.									

(5) IIC function expansion registers n (IICXn)

The IICXn register sets I²C0n function expansion (valid only in the high-speed mode).

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

Setting of the CLXn bit is performed in combination with the SMCn, CLn1, and CLn0 bits of the IICCLn register and the OCKSTHm, OCKSm1, and OCKSm0 bits of the OCKSm register (see **17.4 (6) I²C0n transfer clock setting method**).

Set the IICXn register when the IICCN.IICEn bit = 0.

Reset sets these registers to 00H.

After reset: 00H		R/W	Address: IICX0 FFFFFD85H, IICX1 FFFFFD95H, IICX2 FFFFFDA5H					
	7	6	5	4	3	2	1	<0>
IICXn	0	0	0	0	0	0	0	CLXn

(6) I²C0n transfer clock setting method

The I²C0n transfer clock frequency (f_{SCL}) is calculated using the following expression.

$$f_{SCL} = 1/(m \times T + t_R + t_F)$$

m = 12, 18, 24, 36, 44, 48, 54, 60, 66, 72, 86, 88, 96, 132, 172, 176, 198, 220, 258, 344 (see **Table 17-2 Clock Settings**).

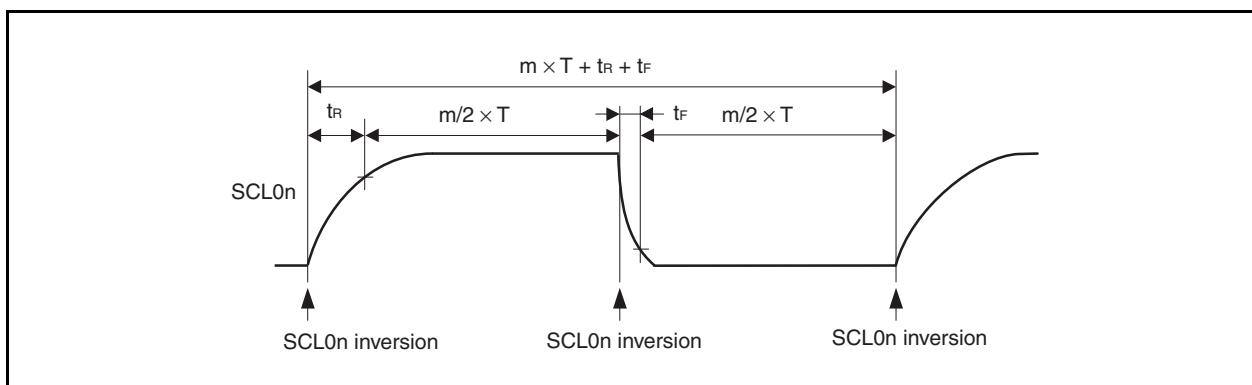
T: 1/f_{xx}

t_R: SCL0n pin rise time

t_F: SCL0n pin fall time

For example, the I²C0n transfer clock frequency (f_{SCL}) when f_{xx} = 19.2 MHz, m = 198, t_R = 200 ns, and t_F = 50 ns is calculated using following expression.

$$f_{SCL} = 1/(198 \times 52 \text{ ns} + 200 \text{ ns} + 50 \text{ ns}) \cong 94.7 \text{ kHz}$$



The clock to be selected can be set by the combination of the SMCn, CLn1, and CLn0 bits of the IICCLn register, the CLXn bit of the IICXn register, and the OCKSTHm, OCKSm1, and OCKSm0 bits of the OCKSm register (m = 0, 1).

Table 17-2. Clock Settings (1/2)

IICX0	IICCL0			Selection Clock	Transfer Clock	Settable Main Clock Frequency (f _{xx}) Range	Operating Mode
	Bit 0	Bit 3	Bit 1				
CLX0	SMC0	CL01	CL00				
0	0	0	0	f _{xx} (when OCKS0 = 18H set)	f _{xx} /44	2.50 MHz ≤ f _{xx} ≤ 4.19 MHz	Standard mode (SMC0 bit = 0)
				f _{xx} /2 (when OCKS0 = 10H set)	f _{xx} /88	4.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
				f _{xx} /3 (when OCKS0 = 11H set)	f _{xx} /132	6.00 MHz ≤ f _{xx} ≤ 12.57 MHz	
				f _{xx} /4 (when OCKS0 = 12H set)	f _{xx} /176	8.00 MHz ≤ f _{xx} ≤ 16.76 MHz	
				f _{xx} /5 (when OCKS0 = 13H set)	f _{xx} /220	10.00 MHz ≤ f _{xx} ≤ 20.00 MHz	
0	0	0	1	f _{xx} (when OCKS0 = 18H set)	f _{xx} /86	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	
				f _{xx} /2 (when OCKS0 = 10H set)	f _{xx} /172	8.38 MHz ≤ f _{xx} ≤ 16.76 MHz	
				f _{xx} /3 (when OCKS0 = 11H set)	f _{xx} /258	12.57 MHz ≤ f _{xx} ≤ 20.00 MHz	
				f _{xx} /4 (when OCKS0 = 12H set)	f _{xx} /344	16.76 MHz ≤ f _{xx} ≤ 20.00 MHz	
0	0	1	0	f _{xx} ^{Note}	f _{xx} /86	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	
0	0	1	1	f _{xx} (when OCKS0 = 18H set)	f _{xx} /66	f _{xx} = 6.40 MHz	
				f _{xx} /2 (when OCKS0 = 10H set)	f _{xx} /132	f _{xx} = 12.80 MHz	
				f _{xx} /3 (when OCKS0 = 11H set)	f _{xx} /198	f _{xx} = 19.20 MHz	
0	1	0	×	f _{xx} (when OCKS0 = 18H set)	f _{xx} /24	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	High-speed mode (SMC0 bit = 1)
				f _{xx} /2 (when OCKS0 = 10H set)	f _{xx} /48	8.00 MHz ≤ f _{xx} ≤ 16.76 MHz	
				f _{xx} /3 (when OCKS0 = 11H set)	f _{xx} /72	12.00 MHz ≤ f _{xx} ≤ 20.00 MHz	
				f _{xx} /4 (when OCKS0 = 12H set)	f _{xx} /96	16.00 MHz ≤ f _{xx} ≤ 20.00 MHz	
0	1	1	0	f _{xx} ^{Note}	f _{xx} /24	4.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
0	1	1	1	f _{xx} (when OCKS0 = 18H set)	f _{xx} /18	f _{xx} = 6.40 MHz	
				f _{xx} /2 (when OCKS0 = 10H set)	f _{xx} /36	f _{xx} = 12.80 MHz	
				f _{xx} /3 (when OCKS0 = 11H set)	f _{xx} /54	f _{xx} = 19.20 MHz	
1	1	0	×	f _{xx} (when OCKS0 = 18H set)	f _{xx} /12	4.00 MHz ≤ f _{xx} ≤ 4.19 MHz	
				f _{xx} /2 (when OCKS0 = 10H set)	f _{xx} /24	8.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
				f _{xx} /3 (when OCKS0 = 11H set)	f _{xx} /36	12.00 MHz ≤ f _{xx} ≤ 12.57 MHz	
				f _{xx} /4 (when OCKS0 = 12H set)	f _{xx} /48	16.00 MHz ≤ f _{xx} ≤ 16.67 MHz	
				f _{xx} /5 (when OCKS0 = 13H set)	f _{xx} /60	f _{xx} = 20.00 MHz	
1	1	1	0	f _{xx} ^{Note}	f _{xx} /12	4.00 MHz ≤ f _{xx} ≤ 4.19 MHz	
Other than above				Setting prohibited	–	–	–

Note Since the selection clock is f_{xx} regardless of the value set to the OCKS0 register, clear the OCKS0 register to 00H (I²C division clock stopped status).

Remark ×: don't care

Table 17-2. Clock Settings (2/2)

IICXm	IICCLm			Selection Clock	Transfer Clock	Settable Main Clock Frequency (f _{xx}) Range	Operating Mode
	Bit 0	Bit 3	Bit 1				
CLXm	SMCm	CLm1	CLm0				
0	0	0	0	f _{xx} (when OCKS1 = 18H set)	f _{xx} /44	2.50 MHz ≤ f _{xx} ≤ 4.19 MHz	Standard mode (SMCm bit = 0)
				f _{xx} /2 (when OCKS1 = 10H set)	f _{xx} /88	4.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
				f _{xx} /3 (when OCKS1 = 11H set)	f _{xx} /132	6.00 MHz ≤ f _{xx} ≤ 12.57 MHz	
				f _{xx} /4 (when OCKS1 = 12H set)	f _{xx} /176	8.00 MHz ≤ f _{xx} ≤ 16.76 MHz	
				f _{xx} /5 (when OCKS1 = 13H set)	f _{xx} /220	10.00 MHz ≤ f _{xx} ≤ 20.00 MHz	
0	0	0	1	f _{xx} (when OCKS1 = 18H set)	f _{xx} /86	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	
				f _{xx} /2 (when OCKS1 = 10H set)	f _{xx} /172	8.38 MHz ≤ f _{xx} ≤ 16.76 MHz	
				f _{xx} /3 (when OCKS1 = 11H set)	f _{xx} /258	12.57 MHz ≤ f _{xx} ≤ 20.00 MHz	
				f _{xx} /4 (when OCKS1 = 12H set)	f _{xx} /344	16.76 MHz ≤ f _{xx} ≤ 20.00 MHz	
0	0	1	0	f _{xx} ^{Note}	f _{xx} /86	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	
0	0	1	1	f _{xx} (when OCKS1 = 18H set)	f _{xx} /66	f _{xx} = 6.40 MHz	
				f _{xx} /2 (when OCKS1 = 10H set)	f _{xx} /132	f _{xx} = 12.80 MHz	
				f _{xx} /3 (when OCKS1 = 11H set)	f _{xx} /198	f _{xx} = 19.20 MHz	
0	1	0	×	f _{xx} (when OCKS1 = 18H set)	f _{xx} /24	4.19 MHz ≤ f _{xx} ≤ 8.38 MHz	High-speed mode (SMCm bit = 1)
				f _{xx} /2 (when OCKS1 = 10H set)	f _{xx} /48	8.00 MHz ≤ f _{xx} ≤ 16.76 MHz	
				f _{xx} /3 (when OCKS1 = 11H set)	f _{xx} /72	12.00 MHz ≤ f _{xx} ≤ 20.00 MHz	
				f _{xx} /4 (when OCKS1 = 12H set)	f _{xx} /96	16.00 MHz ≤ f _{xx} ≤ 20.00 MHz	
0	1	1	0	f _{xx} ^{Note}	f _{xx} /24	4.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
0	1	1	1	f _{xx} (when OCKS1 = 18H set)	f _{xx} /18	f _{xx} = 6.40 MHz	
				f _{xx} /2 (when OCKS1 = 10H set)	f _{xx} /36	f _{xx} = 12.80 MHz	
				f _{xx} /3 (when OCKS1 = 11H set)	f _{xx} /54	f _{xx} = 19.20 MHz	
1	1	0	×	f _{xx} (when OCKS1 = 18H set)	f _{xx} /12	4.00 MHz ≤ f _{xx} ≤ 4.19 MHz	
				f _{xx} /2 (when OCKS1 = 10H set)	f _{xx} /24	8.00 MHz ≤ f _{xx} ≤ 8.38 MHz	
				f _{xx} /3 (when OCKS1 = 11H set)	f _{xx} /36	12.00 MHz ≤ f _{xx} ≤ 12.57 MHz	
				f _{xx} /4 (when OCKS1 = 12H set)	f _{xx} /48	16.00 MHz ≤ f _{xx} ≤ 16.67 MHz	
				f _{xx} /5 (when OCKS1 = 13H set)	f _{xx} /60	f _{xx} = 20.00 MHz	
1	1	1	0	f _{xx} ^{Note}	f _{xx} /12	4.00 MHz ≤ f _{xx} ≤ 4.19 MHz	
Other than above				Setting prohibited	–	–	–

Note Since the selection clock is f_{xx} regardless of the value set to the OCKS1 register, clear the OCKS1 register to 00H (I²C division clock stopped status).

- Remarks** 1. m = 1, 2
2. ×: don't care

(7) IIC division clock select registers 0, 1 (OCKS0, OCKS1)

The OCKSm register controls the I²C0n division clock (m = 0, 1).

These registers control the I²C00 division clock via the OCKS0 register and the I²C01 and I²C02 division clocks via the OCKS1 register.

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

Reset sets these registers to 00H.

After reset: 00H								R/W		Address: OCKS0 FFFFF340H, OCKS1 FFFFF344H											
		7		6		5		4		3		2		1		0					
OCKSm		0		0		0		OCKSEnM		OCKSTHm		0		OCKSm1		OCKSm0					
OCKSEnM		Operation setting of I ² C division clock																			
0		Disable I ² C division clock operation																			
1		Enable I ² C division clock operation																			
OCKSTHm		OCKSm1		OCKSm0		Selection of I ² C division clock															
0		0		0		fxx/2															
0		0		1		fxx/3															
0		1		0		fxx/4															
0		1		1		fxx/5															
1		0		0		fxx															
Other than above		Setting prohibited																			

(8) IIC shift registers n (IICn)

The IICn register is used for serial transmission/reception (shift operations) synchronized with the serial clock. These registers can be read or written in 8-bit units, but data should not be written to the IICn register during a data transfer.

Access (read/write) the IICn 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 IICn register can be written once only after the transmission trigger bit (IICn.STTn bit) has been set to 1.

A wait state is released by writing the IICn register during the wait period, and data transfer is started.

Reset sets these registers to 00H.

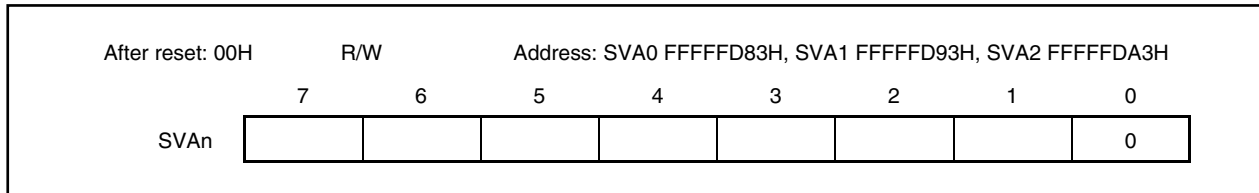
After reset: 00H								R/W		Address: IIC0 FFFFFD80H, IIC1 FFFFFD90H, IIC2 FFFFFDA0H							
		7		6		5		4		3		2		1		0	
IICn																	

(9) Slave address registers n (SVAn)

The SVAn register holds the I²C bus's slave address.

These registers can be read or written in 8-bit units, but bit 0 should be fixed to 0. However, rewriting this register is prohibited when the IICSn.STDn bit = 1 (start condition detection).

Reset sets these registers to 00H.



17.5.1 Pin configuration

The serial clock pin (SCL0n) and serial data bus pin (SDA0n) are configured as follows.

SCL0nThis pin is used for serial clock input and output.

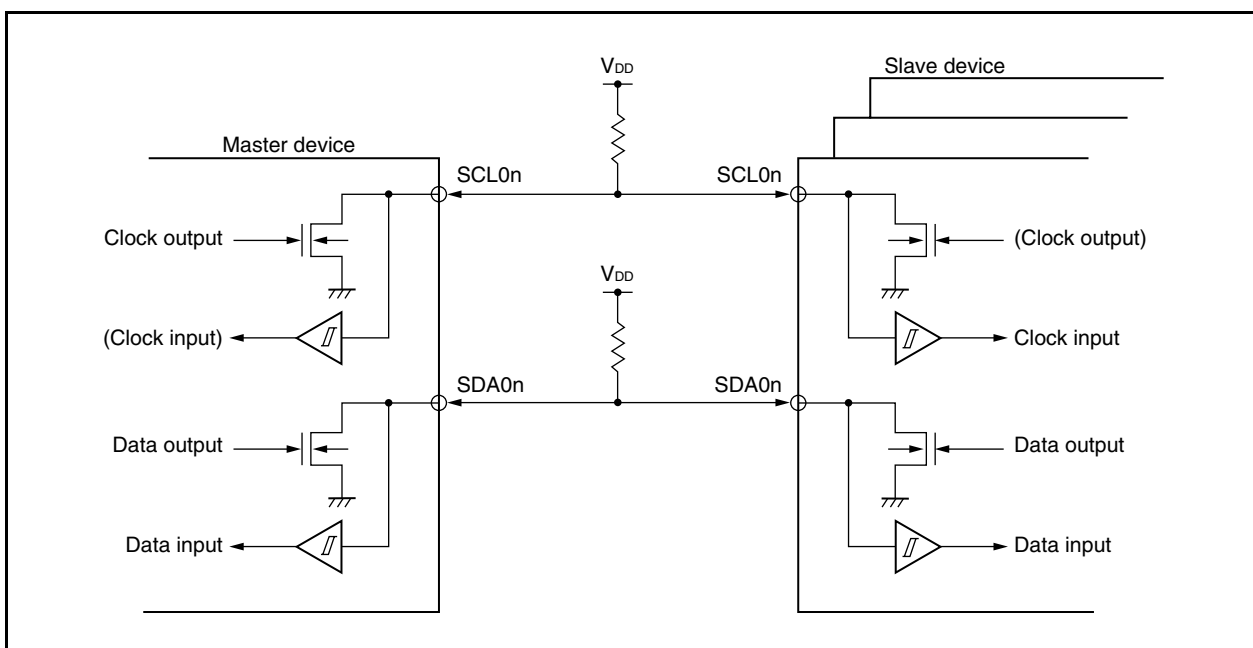
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

SDA0nThis pin is used for serial data input and output.

This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

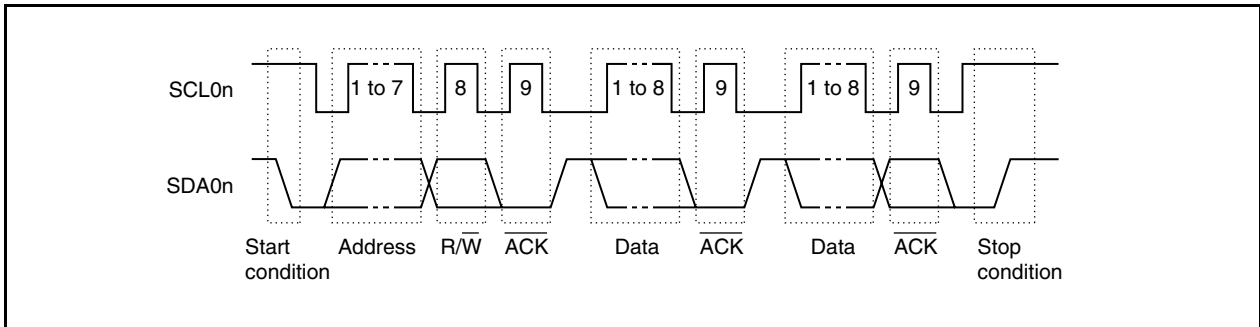
Figure 17-6. Pin Configuration Diagram



17.6 I²C Bus Definitions and Control Methods

The following section describes the I²C bus's serial data communication format and the signals used by the I²C bus. The transfer timing for the "start condition", "address", "transfer direction specification", "data", and "stop condition" generated on the I²C bus's serial data bus is shown below.

Figure 17-7. I²C Bus Serial Data Transfer Timing



The master device generates the start condition, slave address, and stop condition.

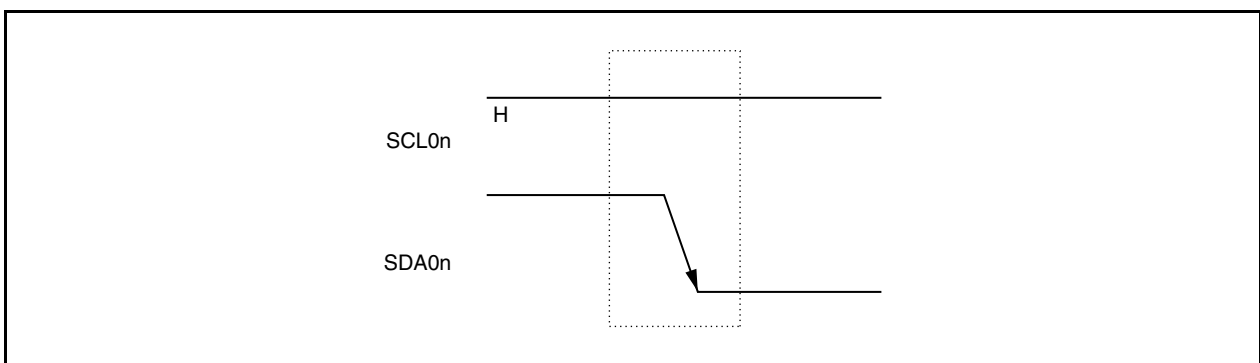
ACK can be generated by either the master or slave device (normally, it is generated by the device that receives 8-bit data).

The serial clock (SCL0n) is continuously output by the master device. However, in the slave device, the SCL0n pin's low-level period can be extended and a wait state can be inserted.

17.6.1 Start condition

A start condition is met when the SCL0n pin is high level and the SDA0n pin changes from high level to low level. The start condition for the SCL0n and SDA0n pins is a signal that the master device outputs to the slave device when starting a serial transfer. The slave device can detect the start condition.

Figure 17-8. Start Condition



A start condition is output when the IICn.STTn bit is set (1) after a stop condition has been detected (IICSn.SPDn bit = 1). When a start condition is detected, the IICSn.STDn bit is set (1).

Caution When the IICn.IICEn bit of the V850ES/JC3-L and V850ES/JE3-L are 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 IICn.IICEn bit to 1 when the SCL0n and SDA0n lines are high level.

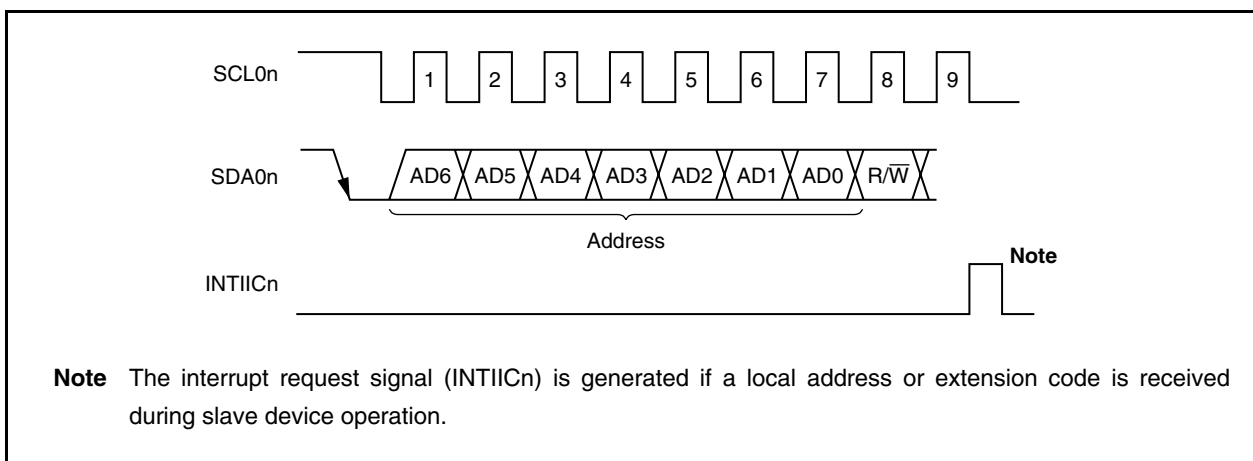
17.6.2 Addresses

The 7 bits of data that follow the start condition are defined as an address.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the SVAn register. If the address data matches the values of the SVAn register, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 17-9. Address



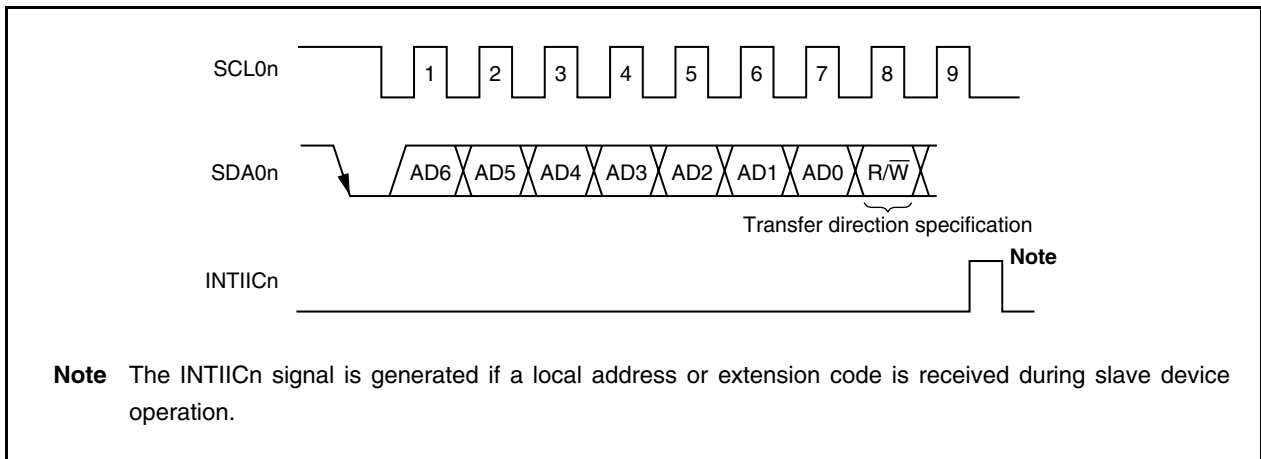
The slave address and the eighth bit, which specifies the transfer direction as described in **17.6.3 Transfer direction specification** below, are written together to IIC shift register n (IICn) and then output. Received addresses are written to the IICn register.

The slave address is assigned to the higher 7 bits of the IICn register.

17.6.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction. When this transfer direction specification bit has a value of 0, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of 1, it indicates that the master device is receiving data from a slave device.

Figure 17-10. Transfer Direction Specification



17.6.4 $\overline{\text{ACK}}$

$\overline{\text{ACK}}$ is used to confirm the serial data status of the transmitting and receiving devices.

The receiving device returns $\overline{\text{ACK}}$ for every 8 bits of data it receives.

The transmitting device normally receives $\overline{\text{ACK}}$ after transmitting 8 bits of data. When $\overline{\text{ACK}}$ is returned from the receiving device, the reception is judged as normal and processing continues. The detection of $\overline{\text{ACK}}$ is confirmed with the IICSn.ACKDn bit.

When the master device is the receiving device, after receiving the final data, it does not return $\overline{\text{ACK}}$ and generates the stop condition. When the slave device is the receiving device and does not return $\overline{\text{ACK}}$, the master device generates either a stop condition or a restart condition, and then stops the current transmission. Failure to return $\overline{\text{ACK}}$ may be caused by the following factors.

- (a) Reception was not performed normally.
- (b) The final data was received.
- (c) The receiving device (slave) does not exist for the specified address.

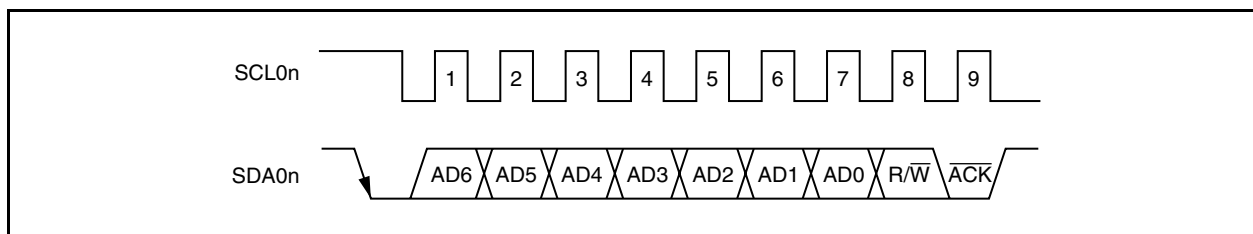
When the receiving device sets the SDA0n line to low level during the ninth clock, $\overline{\text{ACK}}$ is generated (normal reception).

When the IICCN.ACKEn 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 IICSn.TRCn bit to be set. Normally, set the ACKEn bit to 1 for reception (TRCn bit = 0).

When the slave device is receiving (when TRCn bit = 0), if the slave device cannot receive data or does not need to receive any more data, clear the ACKEn bit to 0 to indicate to the master that no more data can be received.

Similarly, when the master device is receiving (when TRCn bit = 0) and the subsequent data is not needed, clear the ACKEn bit to 0 to prevent $\overline{\text{ACK}}$ from being generated. This notifies the slave device (transmitting device) of the end of the data transmission (transmission stopped).

Figure 17-11. $\overline{\text{ACK}}$



When the local address is received, $\overline{\text{ACK}}$ is automatically generated regardless of the value of the ACKEn bit. No $\overline{\text{ACK}}$ is generated if the received address is not a local address (NACK).

When receiving the extension code, set the ACKEn bit to 1 in advance to generate $\overline{\text{ACK}}$.

The $\overline{\text{ACK}}$ generation method during data reception is based on the wait timing setting, as described by the following.

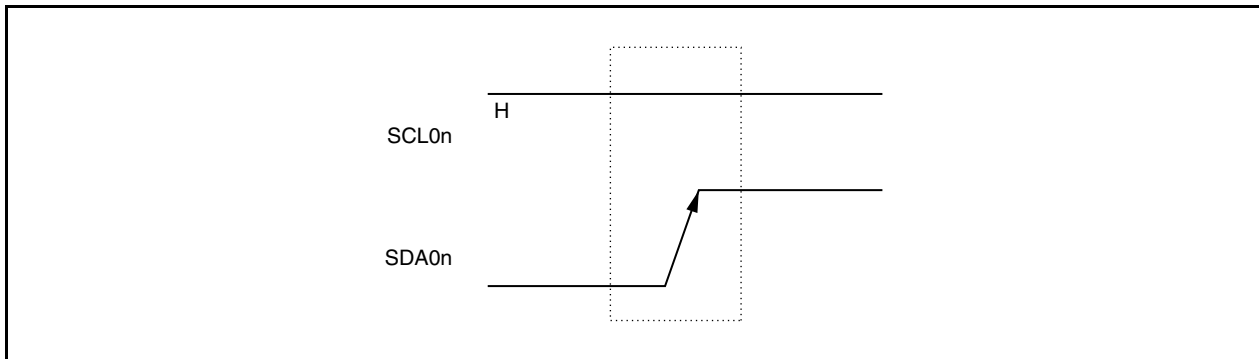
- When 8-clock wait is selected (IICCN.WTIMn bit = 0):
 $\overline{\text{ACK}}$ is generated at the falling edge of the SCL0n pin's eighth clock if the ACKEn bit is set to 1 before the wait state cancellation.
- When 9-clock wait is selected (IICCN.WTIMn bit = 1):
 $\overline{\text{ACK}}$ is generated if the ACKEn bit is set to 1 in advance.

17.6.5 Stop condition

When the SCL0n pin is high level, changing the SDA0n pin from low level to high level generates a stop condition.

A stop condition is generated when the master device outputs to the slave device when serial transfer has been completed. When used as the slave device, the start condition can be detected.

Figure 17-12. Stop Condition



A stop condition is generated when the IICn.SPTn bit is set to 1. When the stop condition is detected, the IICn.SPDn bit is set to 1 and the interrupt request signal (INTIICn) is generated when the IICn.SPIEn bit is set to 1.

17.6.6 Wait state

A wait state is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCL0n pin to low level notifies the communication partner of the wait state. When the wait state has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 17-13. Wait State (1/2)

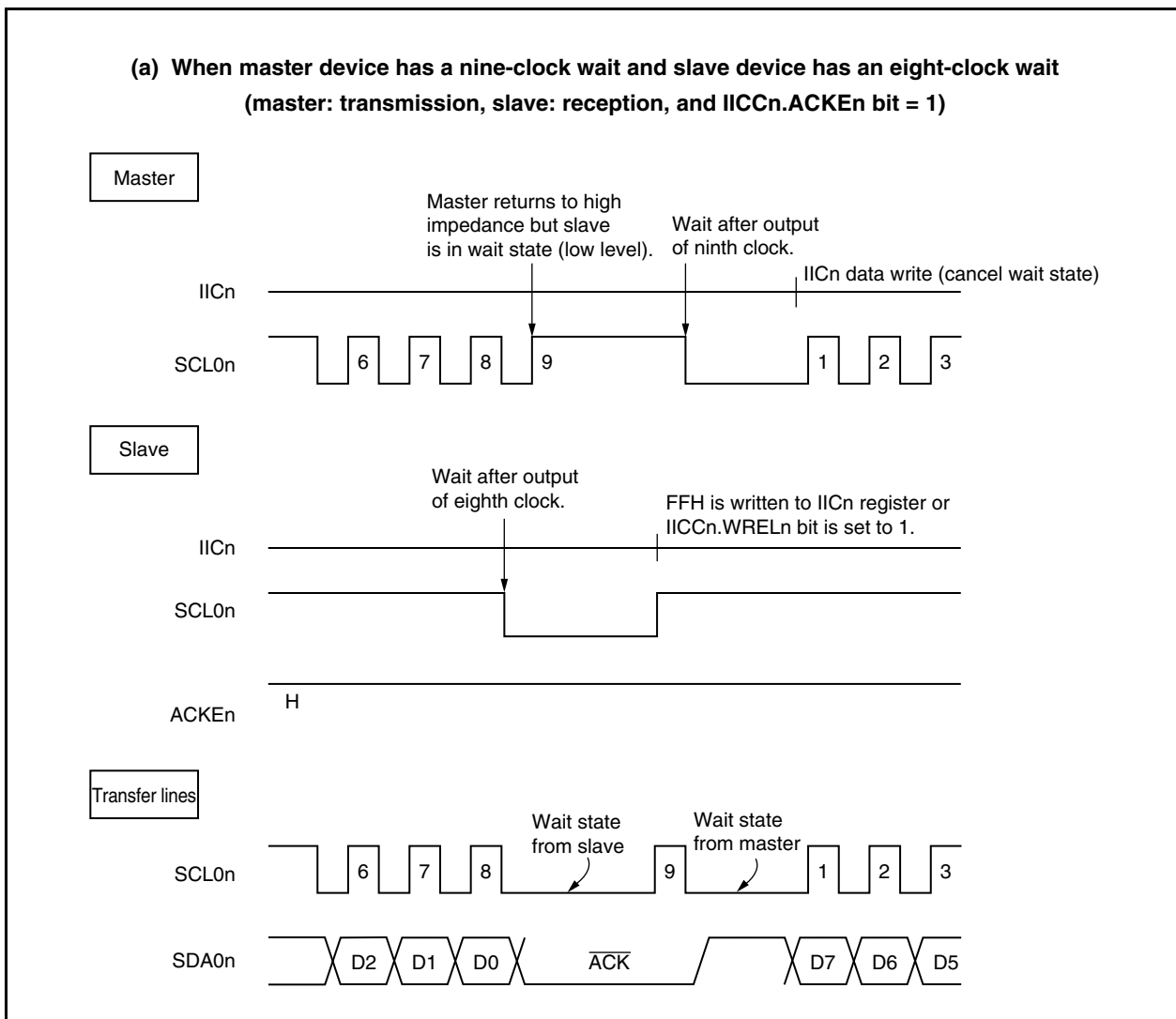
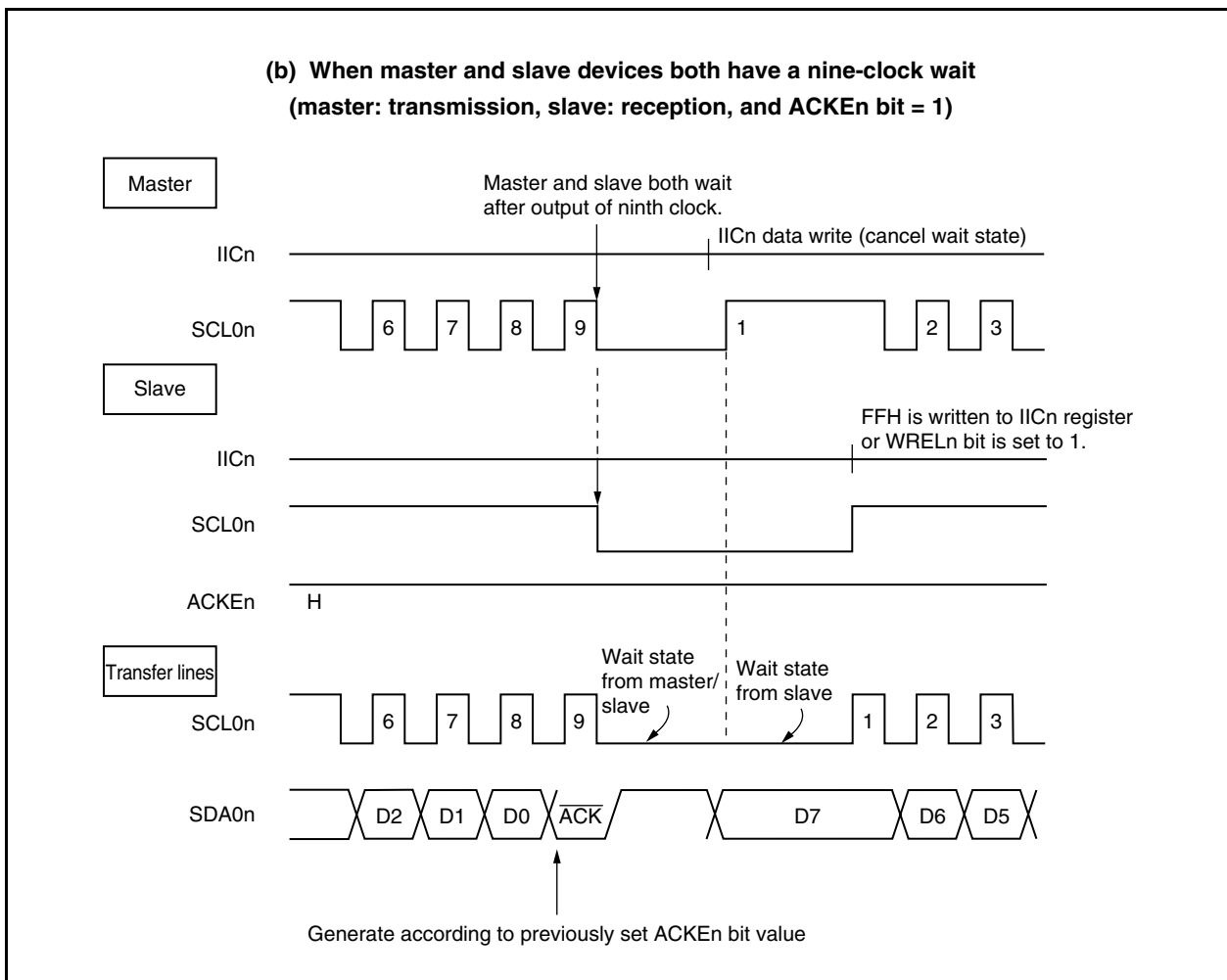


Figure 17-13. Wait State (2/2)



A wait state may be automatically generated depending on the setting of the IICn.WTIMn bit.

Normally, when the IICn.WRELn bit is set to 1 or when FFH is written to the IICn register on the receiving side, the wait state is canceled and the transmitting side writes data to the IICn register to cancel the wait state.

The master device can also cancel the wait state via either of the following methods.

- By setting the IICn.STTn bit to 1
- By setting the IICn.SPTn bit to 1

17.6.7 Wait state cancellation method

In the case of I²C0n, a wait state can be canceled normally in the following ways.

- By writing data to the IICn register
- By setting the IICn.WRELn bit to 1 (wait state cancellation)
- By setting the IICn.STTn bit to 1 (start condition generation)
- By setting the IICn.SPTn bit to 1 (stop condition generation)

If any of these wait state cancellation actions is performed, I²C0n will cancel the wait state and restart communication.

When canceling the wait state and sending data (including addresses), write data to the IICn register.

To receive data after canceling the wait state, or to complete data transmission, set the WRELn bit to 1.

To generate a restart condition after canceling the wait state, set the STTn bit to 1.

To generate a stop condition after canceling the wait state, set the SPTn bit to 1.

Cancel each wait state only once.

For example, if data is written to the IICn register following wait state cancellation by setting the WRELn bit to 1, a conflict between the SDA0n line change timing and the IICn register write timing may result in the data output to the SDA0n line being incorrect.

Even in other operations, if communication is stopped halfway, clearing the IICn.IICEn bit to 0 will stop communication, enabling the wait state to be cancelled.

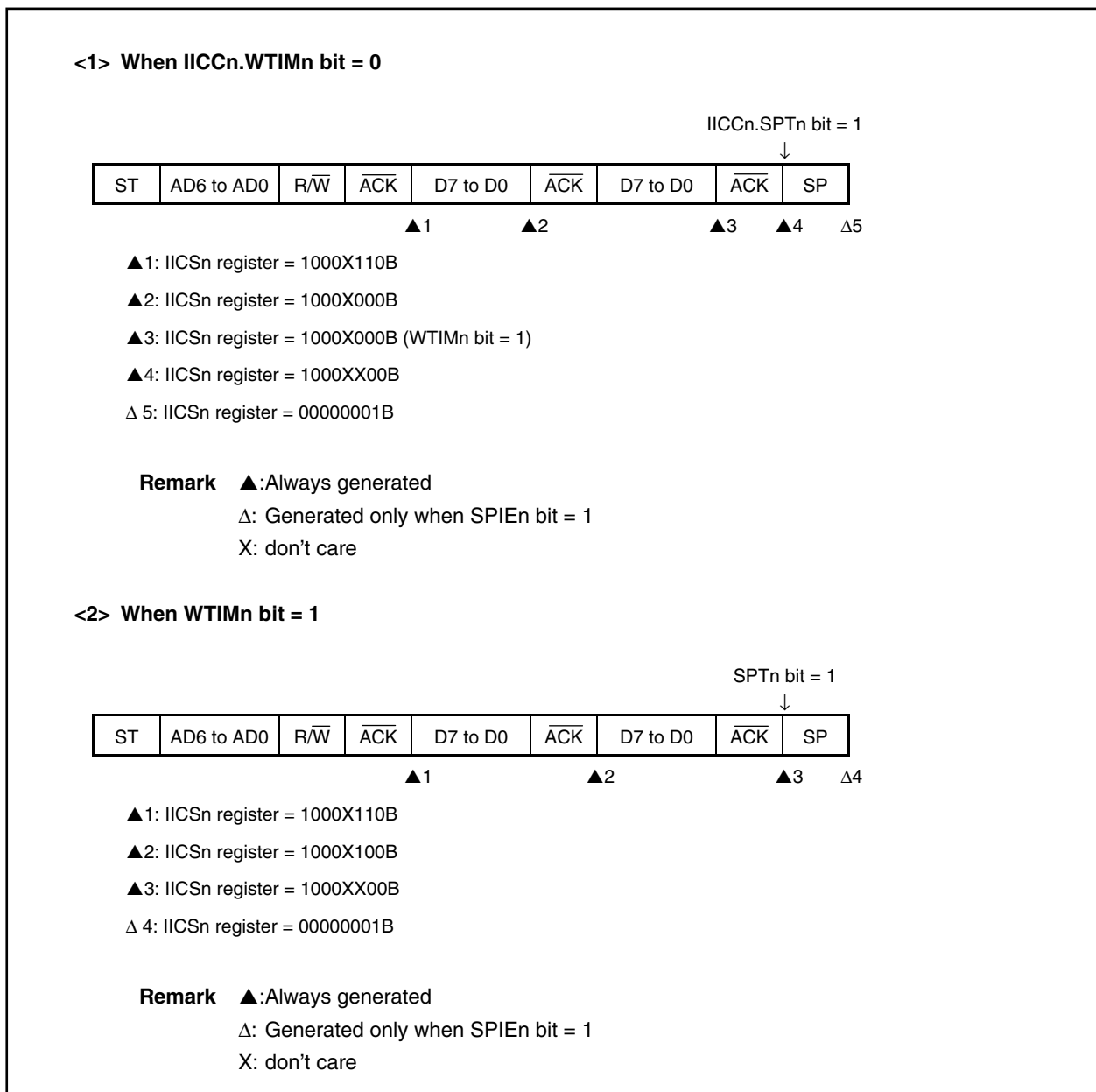
If the I²C bus deadlocks due to noise, etc., setting the IICn.LRELn bit to 1 causes the communication to stop, enabling the wait state to be cancelled.

17.7 I²C Interrupt Request Signals (INTIICn)

The following shows the value of the IICSn register at the INTIICn interrupt request signal generation timing and at the INTIICn signal timing.

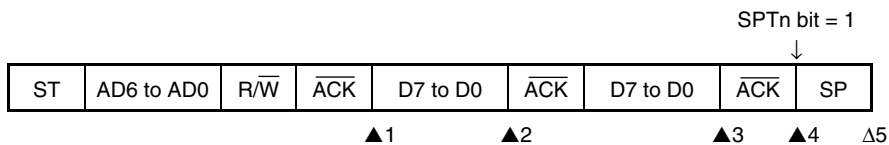
17.7.1 Master device operation

(1) Start ~ Address ~ Data ~ Data ~ Stop (normal transmission/reception)



(3) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

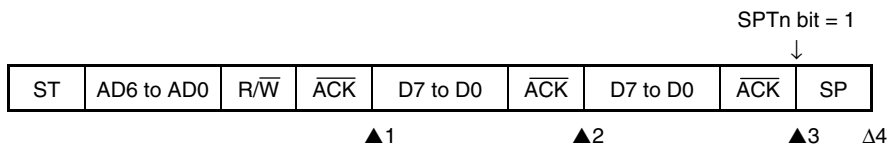
<1> When WTIMn bit = 0



- ▲1: IICSn register = 1010X110B
- ▲2: IICSn register = 1010X000B
- ▲3: IICSn register = 1010X000B (WTIMn bit = 1)
- ▲4: IICSn register = 1010XX00B
- Δ5: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

<2> When WTIMn bit = 1



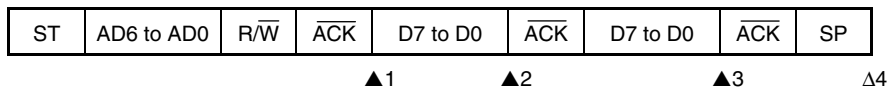
- ▲1: IICSn register = 1010X110B
- ▲2: IICSn register = 1010X100B
- ▲3: IICSn register = 1010XX00B
- Δ4: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

17.7.2 Slave device operation (when receiving slave address data (address match))

(1) Start ~ Address ~ Data ~ Data ~ Stop

<1> When IICn.WTIMn bit = 0



▲1: IICSn register = 0001X110B

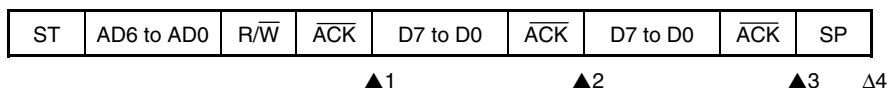
▲2: IICSn register = 0001X000B

▲3: IICSn register = 0001X000B

Δ 4: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when IICn.SPIEn bit = 1
 X: don't care

<2> When WTIMn bit = 1



▲1: IICSn register = 0001X110B

▲2: IICSn register = 0001X100B

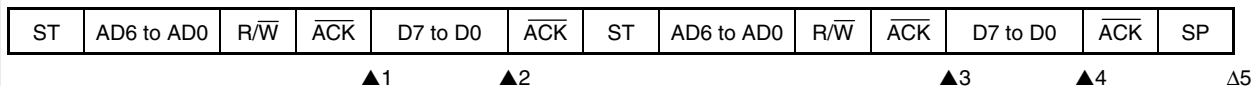
▲3: IICSn register = 0001XX00B

Δ 4: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

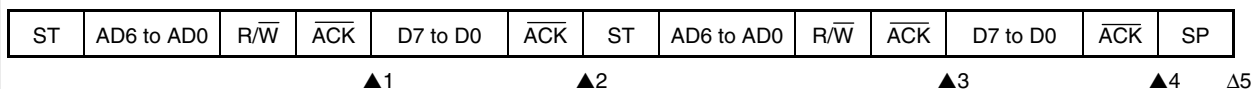
<1> When WTIMn bit = 0 (after restart, address match)



- ▲1: IICSn register = 0001X110B
- ▲2: IICSn register = 0001X000B
- ▲3: IICSn register = 0001X110B
- ▲4: IICSn register = 0001X000B
- Δ 5: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

<2> When WTIMn bit = 1 (after restart, address match)

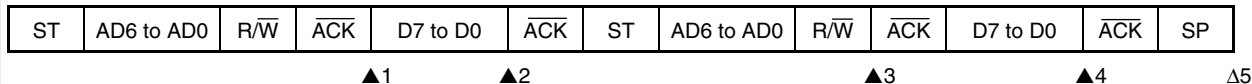


- ▲1: IICSn register = 0001X110B
- ▲2: IICSn register = 0001XX00B
- ▲3: IICSn register = 0001X110B
- ▲4: IICSn register = 0001XX00B
- Δ 5: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

(3) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

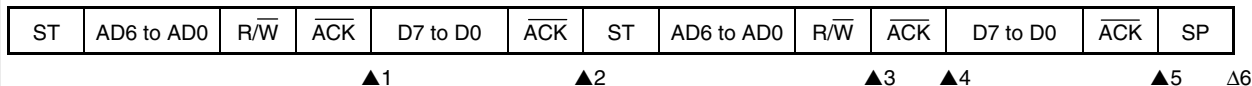
<1> When WTIMn bit = 0 (after restart, extension code reception)



- ▲1: IICSn register = 0001X110B
- ▲2: IICSn register = 0001X000B
- ▲3: IICSn register = 0010X010B
- ▲4: IICSn register = 0010X000B
- Δ 5: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

<2> When WTIMn bit = 1 (after restart, extension code reception)



- ▲1: IICSn register = 0001X110B
- ▲2: IICSn register = 0001XX00B
- ▲3: IICSn register = 0010X010B
- ▲4: IICSn register = 0010X110B
- ▲5: IICSn register = 0010XX00B
- Δ 6: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

(4) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIMn bit = 0 (after restart, address mismatch (= not extension code))

ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	SP
			▲1		▲2				▲3			Δ4

▲1: IICSn register = 0001X110B

▲2: IICSn register = 0001X000B

▲3: IICSn register = 00000X10B

Δ 4: IICSn register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIEn bit = 1

X: don't care

<2> When WTIMn bit = 1 (after restart, address mismatch (= not extension code))

ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/W	ACK	D7 to D0	ACK	SP
			▲1		▲2				▲3			Δ4

▲1: IICSn register = 0001X110B

▲2: IICSn register = 0001XX00B

▲3: IICSn register = 00000X10B

Δ 4: IICSn register = 00000001B

Remark ▲: Always generated

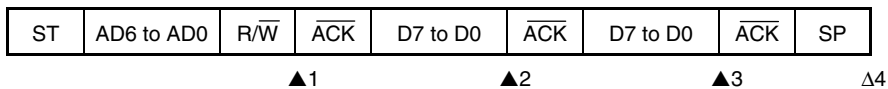
Δ: Generated only when SPIEn bit = 1

X: don't care

17.7.3 Slave device operation (when receiving extension code)

(1) Start ~ Code ~ Data ~ Data ~ Stop

<1> When IICn.WTIMn bit = 0



▲1: IICSn register = 0010X010B

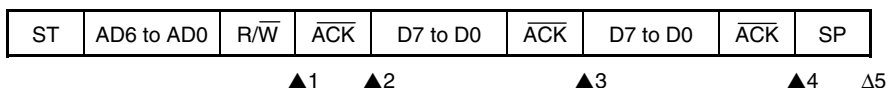
▲2: IICSn register = 0010X000B

▲3: IICSn register = 0010X000B

Δ 4: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when IICn.SPIEn bit = 1
 X: don't care

<2> When WTIMn bit = 1



▲1: IICSn register = 0010X010B

▲2: IICSn register = 0010X110B

▲3: IICSn register = 0010X100B

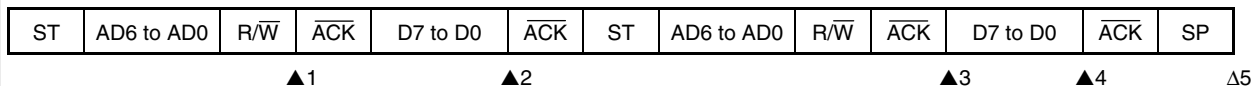
▲4: IICSn register = 0010XX00B

Δ 5: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

(2) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

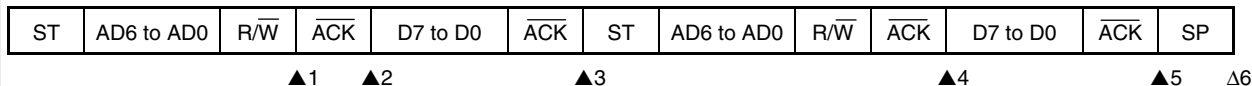
<1> When WTIMn bit = 0 (after restart, address match)



- ▲1: IICSn register = 0010X010B
- ▲2: IICSn register = 0010X000B
- ▲3: IICSn register = 0001X110B
- ▲4: IICSn register = 0001X000B
- Δ 5: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

<2> When WTIMn bit = 1 (after restart, address match)

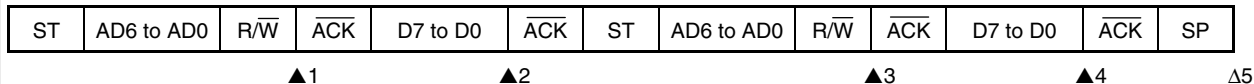


- ▲1: IICSn register = 0010X010B
- ▲2: IICSn register = 0010X110B
- ▲3: IICSn register = 0010XX00B
- ▲4: IICSn register = 0001X110B
- ▲5: IICSn register = 0001XX00B
- Δ 6: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

(3) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

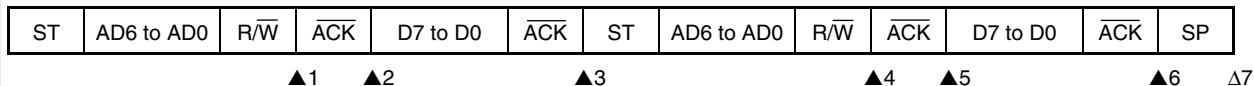
<1> When WTIMn bit = 0 (after restart, extension code reception)



- ▲1: IICSn register = 0010X010B
- ▲2: IICSn register = 0010X000B
- ▲3: IICSn register = 0010X010B
- ▲4: IICSn register = 0010X000B
- Δ 5: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

<2> When WTIMn bit = 1 (after restart, extension code reception)

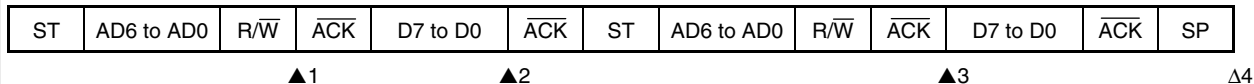


- ▲1: IICSn register = 0010X010B
- ▲2: IICSn register = 0010X110B
- ▲3: IICSn register = 0010XX00B
- ▲4: IICSn register = 0010X010B
- ▲5: IICSn register = 0010X110B
- ▲6: IICSn register = 0010XX00B
- Δ 7: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

(4) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

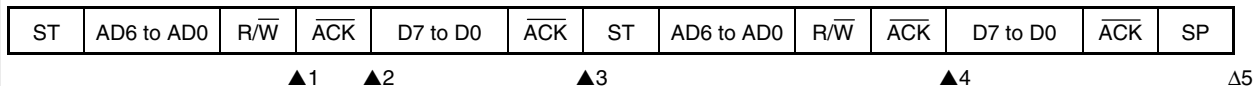
<1> When WTIMn bit = 0 (after restart, address mismatch (= not extension code))



- ▲1: IICSn register = 0010X010B
- ▲2: IICSn register = 0010X000B
- ▲3: IICSn register = 00000X10B
- Δ 4: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

<2> When WTIMn bit = 1 (after restart, address mismatch (= not extension code))

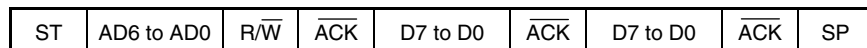


- ▲1: IICSn register = 0010X010B
- ▲2: IICSn register = 0010X110B
- ▲3: IICSn register = 0010XX00B
- ▲4: IICSn register = 00000X10B
- Δ 5: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

17.7.4 Operation without communication

(1) Start ~ Code ~ Data ~ Data ~ Stop



Δ1

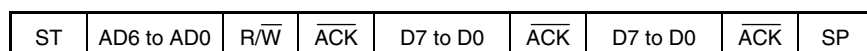
Δ 1: IICSn register = 00000001B

Remark Δ: Generated only when SPIEn bit = 1

17.7.5 Operation when arbitration loss occurs (operation as slave after arbitration loss)

(1) When arbitration loss occurs during transmission of slave address data

<1> When IICn.WTIMn bit = 0



▲1

▲2

▲3

Δ4

▲1: IICSn register = 0101X110B (Example: When IICSn.ALDn bit is read during interrupt servicing)

▲2: IICSn register = 0001X000B

▲3: IICSn register = 0001X000B

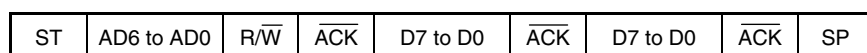
Δ 4: IICSn register = 00000001B

Remark ▲: Always generated

Δ: Generated only when IICn.SPIEn bit = 1

X: don't care

<2> When WTIMn bit = 1



▲1

▲2

▲3

Δ4

▲1: IICSn register = 0101X110B (Example: When ALDn bit is read during interrupt servicing)

▲2: IICSn register = 0001X100B

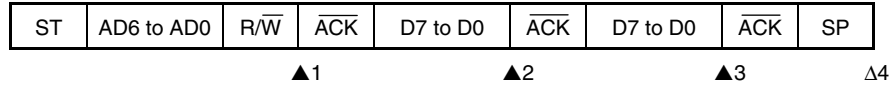
▲3: IICSn register = 0001XX00B

Δ 4: IICSn register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIEn bit = 1

X: don't care

(2) When arbitration loss occurs during transmission of extension code**<1> When WTIMn bit = 0**

▲1: IICSn register = 0110X010B (Example: When ALDn bit is read during interrupt servicing)

▲2: IICSn register = 0010X000B

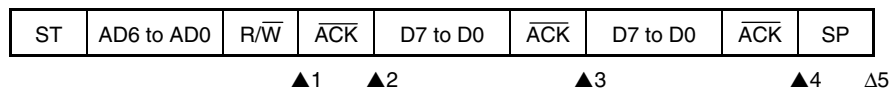
▲3: IICSn register = 0010X000B

Δ 4: IICSn register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIEn bit = 1

X: don't care

<2> When WTIMn bit = 1

▲1: IICSn register = 0110X010B (Example: When ALDn bit is read during interrupt servicing)

▲2: IICSn register = 0010X110B

▲3: IICSn register = 0010X100B

▲4: IICSn register = 0010XX00B

Δ 5: IICSn register = 00000001B

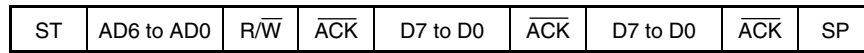
Remark ▲: Always generated

Δ: Generated only when SPIEn bit = 1

X: don't care

17.7.6 Operation when arbitration loss occurs (no communication after arbitration loss)

(1) When arbitration loss occurs during transmission of slave address data



▲1

Δ2

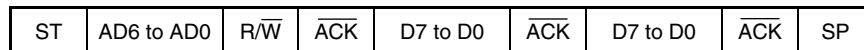
▲1: IICSn register = 01000110B (Example: When IICSn.ALDn bit is read during interrupt servicing)

Δ2: IICSn register = 00000001B

Remark ▲: Always generated

Δ: Generated only when IICn.SPIEn bit = 1

(2) When arbitration loss occurs during transmission of extension code



▲1

Δ2

▲1: IICSn register = 0110X010B (Example: When ALDn bit is read during interrupt servicing)

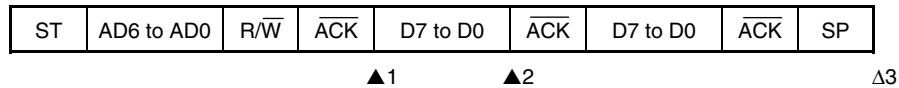
IICn.LRELn bit is set to 1 by software

Δ2: IICSn register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIEn bit = 1

X: don't care

(3) When arbitration loss occurs during data transfer**<1> When IICn.WTIMn bit = 0**

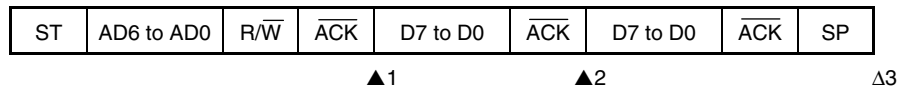
▲1: IICSn register = 10001110B

▲2: IICSn register = 01000000B (Example: When ALDn bit is read during interrupt servicing)

Δ3: IICSn register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIEn bit = 1

<2> When WTIMn bit = 1

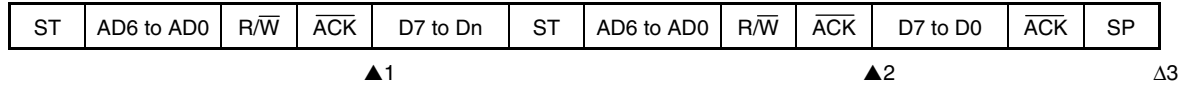
▲1: IICSn register = 10001110B

▲2: IICSn register = 01000100B (Example: When ALDn bit is read during interrupt servicing)

Δ3: IICSn register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIEn bit = 1

(4) When arbitration loss occurs due to restart condition during data transfer**<1> Not extension code (Example: Address mismatch)**

▲1: IICSn register = 1000X110B

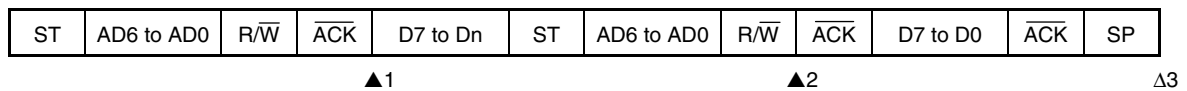
▲2: IICSn register = 01000110B (Example: When ALDn bit is read during interrupt servicing)

Δ3: IICSn register = 00000001B

Remarks 1. ▲: Always generated

Δ: Generated only when SPIEn bit = 1

X: don't care

2. Dn = D6 to D0**<2> Extension code**

▲1: IICSn register = 1000X110B

▲2: IICSn register = 0110X010B (Example: When ALDn bit is read during interrupt servicing)

IICn.LRELn bit is set to 1 by software

Δ3: IICSn register = 00000001B

Remarks 1. ▲: Always generated

Δ: Generated only when SPIEn bit = 1

X: don't care

2. Dn = D6 to D0

(5) When arbitration loss occurs due to stop condition during data transfer

ST	AD6 to AD0	R/W	\overline{ACK}	D7 to Dn	SP
----	------------	-----	------------------	----------	----

▲1

Δ2

▲1: IICSn register = 1000X110B

Δ 2: IICSn register = 01000001B

Remarks 1. ▲: Always generated

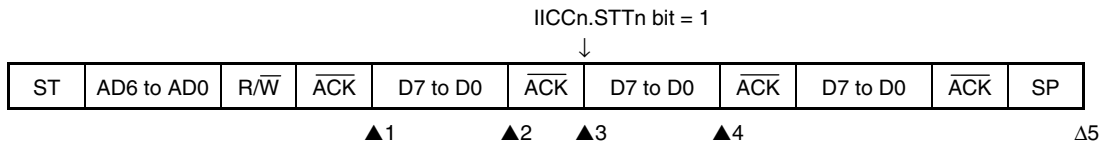
Δ: Generated only when SPIEn bit = 1

X: don't care

2. Dn = D6 to D0

(6) When arbitration loss occurs due to low level of SDA0n pin when attempting to generate a restart condition

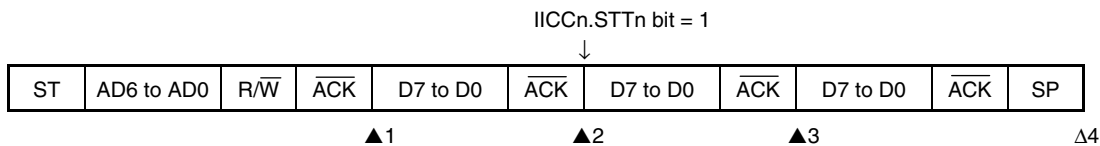
<1> When WTIMn bit = 0



- ▲1: IICSn register = 1000X110B
- ▲2: IICSn register = 1000X000B (WTIMn bit = 1)
- ▲3: IICSn register = 1000XX00B (WTIMn bit = 0)
- ▲4: IICSn register = 01000000B (Example: When ALDn bit is read during interrupt servicing)
- Δ 5: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

<2> When WTIMn bit = 1

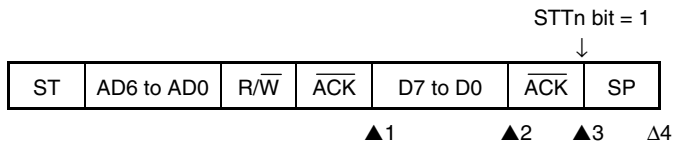


- ▲1: IICSn register = 1000X110B
- ▲2: IICSn register = 1000XX00B
- ▲3: IICSn register = 01000100B (Example: When ALDn bit is read during interrupt servicing)
- Δ 4: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

(7) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

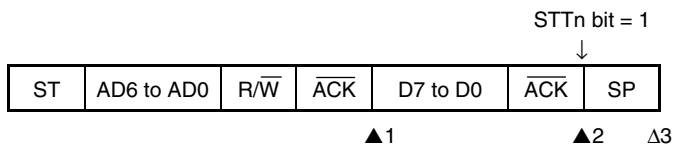
<1> When WTIMn bit = 0



- ▲1: IICSn register = 1000X110B
- ▲2: IICSn register = 1000X000B (WTIMn bit = 1)
- ▲3: IICSn register = 1000XX00B
- Δ4: IICSn register = 01000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

<2> When WTIMn bit = 1

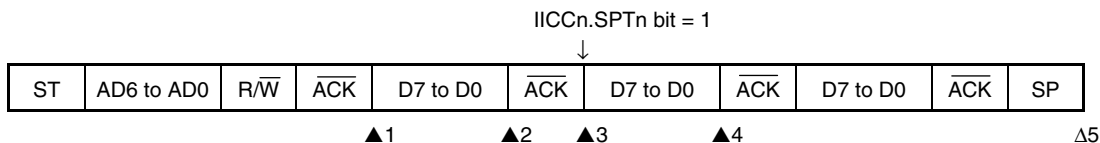


- ▲1: IICSn register = 1000X110B
- ▲2: IICSn register = 1000XX00B
- Δ3: IICSn register = 01000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

(8) When arbitration loss occurs due to low level of SDA0n pin when attempting to generate a stop condition

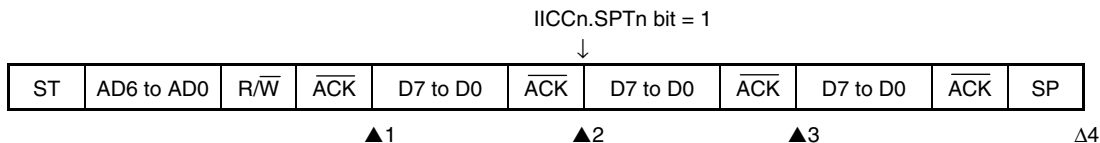
<1> When WTIMn bit = 0



- ▲1: IICSn register = 1000X110B
- ▲2: IICSn register = 1000X000B (WTIMn bit = 1)
- ▲3: IICSn register = 1000XX00B (WTIMn bit = 0)
- ▲4: IICSn register = 01000000B (Example: When ALDn bit is read during interrupt servicing)
- Δ5: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

<2> When WTIMn bit = 1



- ▲1: IICSn register = 1000X110B
- ▲2: IICSn register = 1000XX00B
- ▲3: IICSn register = 01000000B (Example: When ALDn bit is read during interrupt servicing)
- Δ4: IICSn register = 00000001B

Remark ▲: Always generated
 Δ: Generated only when SPIEn bit = 1
 X: don't care

17.8 Interrupt Request Signal (INTIICn) Generation Timing and Wait Control

The setting of the IICn.WTIMn bit determines the timing by which the INTIICn register is generated and the corresponding wait control, as shown below.

Table 17-3. INTIICn Generation Timing and Wait Control

WTIMn 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 INTIICn signal and wait period occur at the falling edge of the ninth clock only when there is a match with the address set to the SVAn register.

At this point, \overline{ACK} is generated regardless of the value set to the IICn.ACKEn bit. For a slave device that has received an extension code, the INTIICn signal occurs at the falling edge of the eighth clock.

When the address does not match after restart, the INTIICn 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 SVAn register and an extension code is not received, neither the INTIICn 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 is determined in accordance with the conditions shown in notes 1 and 2 above regardless of the WTIMn bit setting.
- Master device operation: Interrupt and wait timing occurs at the falling edge of the ninth clock regardless of the WTIMn bit setting.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing is determined according to the WTIMn bit setting.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing is determined according to the WTIMn bit setting.

(4) Wait cancellation method

The following four wait cancellation methods are available.

- By setting the IICn.WRELn bit to 1
- By writing to the IICn register
- By setting start condition (IICn.STTn bit = 1)^{Note}
- By setting stop condition (IICn.SPTn bit = 1)^{Note}

Note Master only

When an 8-clock wait has been selected (WTIMn bit = 0), whether or not $\overline{\text{ACK}}$ has been generated must be determined prior to wait cancellation.

(5) Stop condition detection

The INTIICn signal is generated when a stop condition is detected.

17.9 Address Match Detection Method

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. The INTIICn signal occurs when a local address has been set to the SVAn register and when the address set to the SVAn register matches the slave address sent by the master device, or when an extension code has been received.

17.10 Error Detection

In I²C bus mode, the status of the serial data bus pin (SDA0n) during data transmission is captured by the IICn register of the transmitting device, so the data of the IICn register prior to transmission can be compared with the transmitted IICn data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

17.11 Extension Code

- (1) When the higher 4 bits of the receive address are either 0000 or 1111, the extension code flag (IICSn.EXCn bit) is set for extension code reception and an interrupt request signal (INTIICn) is issued at the falling edge of the eighth clock.

The local address stored in the SVAn register is not affected.

- (2) If 11110xx0 is set to the SVAn register by a 10-bit address transfer and 11110xx0 is transferred from the master device, the results are as follows. Note that the INTIICn signal occurs at the falling edge of the eighth clock.

- Higher 4 bits of data match: EXCn bit = 1
- 7 bits of data match: IICSn.COIn bit = 1

- (3) Since the processing after the interrupt request signal occurs differs according to the data that follows the extension code, such processing is performed by software.

For example, when operation as a slave is not desired after the extension code is received, set the IICSn.LRELn bit to 1 and the CPU will enter the next communication wait state.

Table 17-4. Extension Code Bit Definitions

Slave Address	R/W Bit	Description
0000 000	0	General call address
0000 000	1	Start byte
0000 001	X	CBUS address
0000 010	X	Address that is reserved for different bus format
1111 0xx	X	10-bit slave address specification

17.12 Arbitration

When several master devices simultaneously generate a start condition (when the IICn.STTn bit is set to 1 before the IICn.STDn bit is set to 1), communication between the master devices is performed while 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 (IICn.ALDn bit) is set to 1 at the timing at which the arbitration loss occurred, and the SCL0n and SDA0n lines are both set to high impedance, which releases the bus.

Arbitration loss is detected based on the timing of the next interrupt request signal (INTIICn) (the eighth or ninth clock, when a stop condition is detected, etc.) and the setting of the ALDn bit to 1, which is made by software.

For details of interrupt request timing, see 17.7 I²C Interrupt Request Signals (INTIICn).

Figure 17-14. Arbitration Timing Example

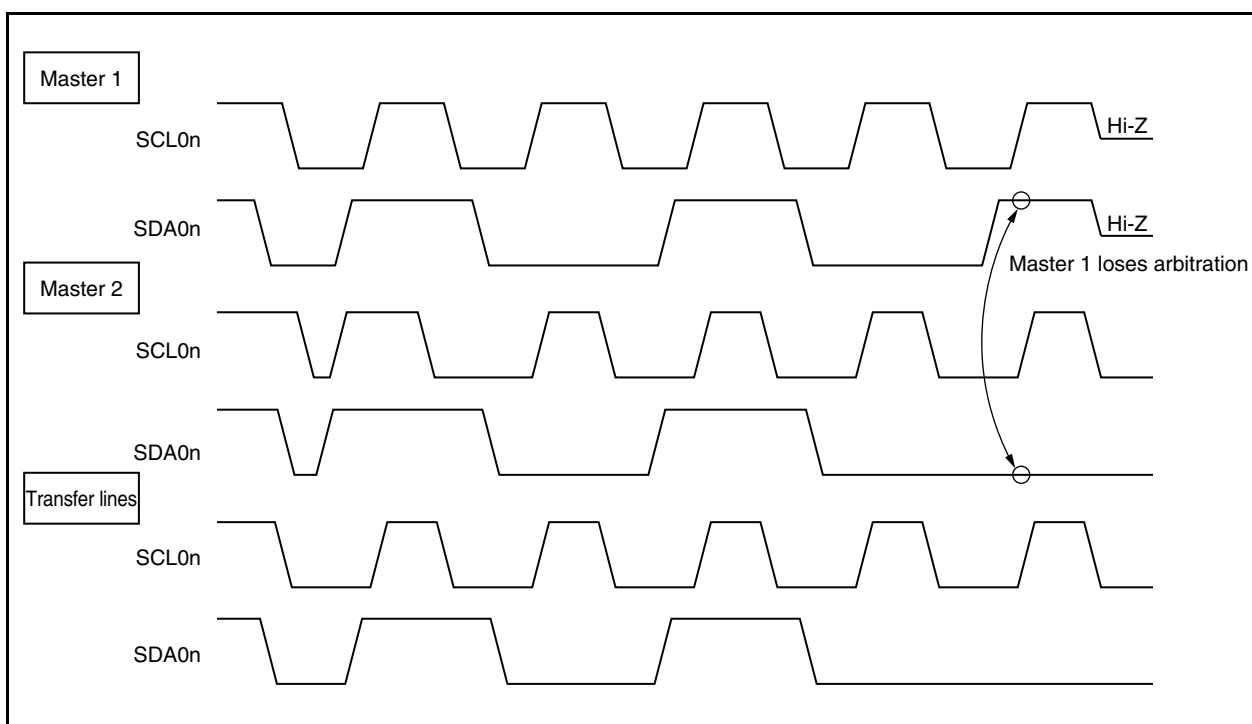


Table 17-5. Status During Arbitration and Interrupt Request Signal Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
Transmitting address transmission	At falling edge of eighth or ninth clock following byte transfer ^{Note 1}
Read/write data after address transmission	
Transmitting extension code	
Read/write data after extension code transmission	
Transmitting data	
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 IICn.SPIEn bit = 1) ^{Note 2}
When SDA0n pin is low level while attempting to generate restart condition	At falling edge of eighth or ninth clock following byte transfer ^{Note 1}
When stop condition is detected while attempting to generate restart condition	When stop condition is generated (when IICn.SPIEn bit = 1) ^{Note 2}
When DSA0n pin is low level while attempting to generate stop condition	At falling edge of eighth or ninth clock following byte transfer ^{Note 1}
When SCL0n pin is low level while attempting to generate restart condition	

- Notes 1.** When the IICn.WTIMn bit = 1, an INTIICn signal occurs at the falling edge of the ninth clock. When the WTIMn bit = 0 and the extension code's slave address is received, an INTIICn signal occurs at the falling edge of the eighth clock.
- 2.** When there is a possibility that arbitration will occur, set the SPIEn bit to 1 for master device operation.

17.13 Wakeup Function

The I²C bus slave function is a function that generates an interrupt request signal (INTIICn) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary INTIICn 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 generated a start condition) to a slave device.

However, when a stop condition is detected, the IICn.SPIEn bit is set regardless of the wakeup function, and this determines whether INTIICn signal is enabled or disabled.

17.14 Communication Reservation

17.14.1 When communication reservation function is enabled (IICFn.IICRSVn bit = 0)

To start master device communications when the V850ES/JC3-L and V850ES/JE3-L are not currently using the bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes in which the bus is not used by the V850ES/JC3-L and V850ES/JE3-L.

- When arbitration results in the V850ES/JC3-L and V850ES/JE3-L being neither the master nor a slave
- When an extension code is received and slave operation is disabled (\overline{ACK} is not returned and the bus was released when the IICn.LRELn bit was set to 1).

If the IICn.STTn bit is set to 1 while the bus is not used by the V850ES/JC3-L and V850ES/JE3-L, a start condition is automatically generated and a wait status is set after the bus is released (after a stop condition is detected).

When the bus release is detected (when a stop condition is detected), writing to the IICn register causes master address transfer to start. At this point, the IICn.SPIEn bit should be set to 1.

When STTn has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

If the bus has been releasedA 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 STTn bit, set the STTn bit to 1, wait for the wait period, then check the IICSn.MSTS_n bit.

The wait periods, which should be set via software, are listed in Table 17-6. These wait periods can be set by using the SMCn, CLn1, and CLn0 bits of the IICCLn register and the IICXn.CLXn bit.

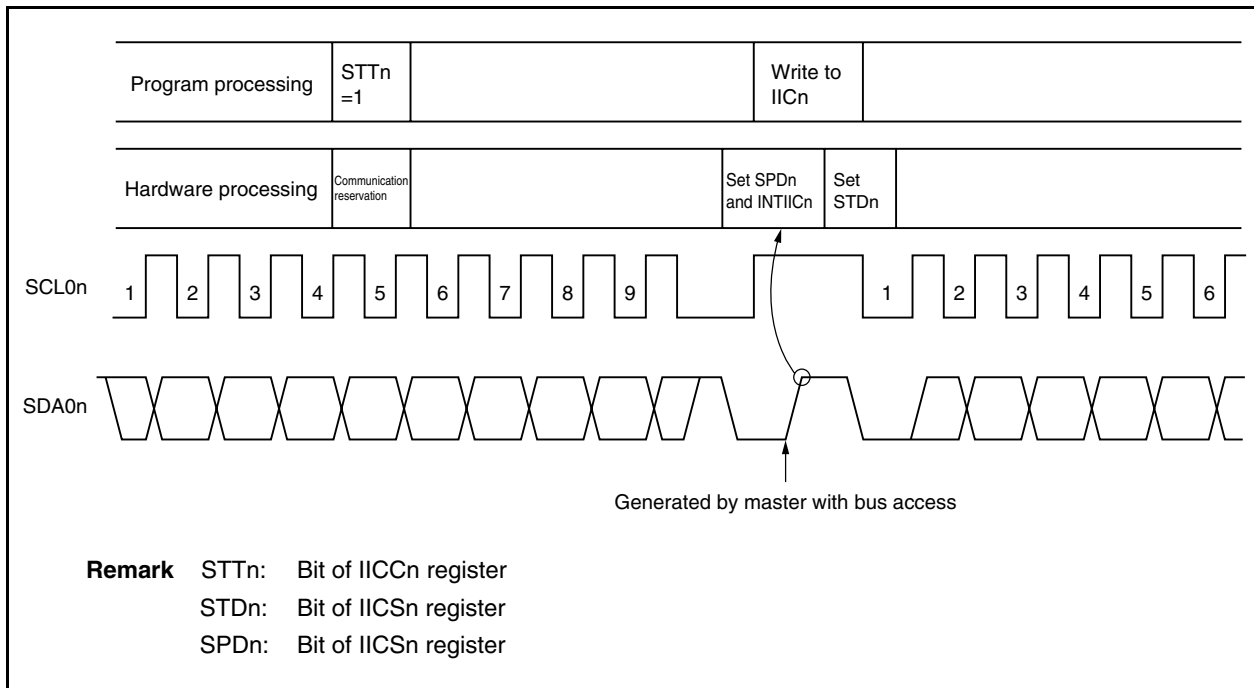
Table 17-6. Wait Periods

Clock Selection	CLXn	SMCn	CLn1	CLn0	Wait Period
f _{xx} (when OCKSm = 18H set)	0	0	0	0	26 clocks
f _{xx} /2 (when OCKSm = 10H set)	0	0	0	0	52 clocks
f _{xx} /3 (when OCKSm = 11H set)	0	0	0	0	78 clocks
f _{xx} /4 (when OCKSm = 12H set)	0	0	0	0	104 clocks
f _{xx} /5 (when OCKSm = 13H set)	0	0	0	0	130 clocks
f _{xx} (when OCKSm = 18H set)	0	0	0	1	47 clocks
f _{xx} /2 (when OCKSm = 10H set)	0	0	0	1	94 clocks
f _{xx} /3 (when OCKSm = 11H set)	0	0	0	1	141 clocks
f _{xx} /4 (when OCKSm = 12H set)	0	0	0	1	188 clocks
f _{xx}	0	0	1	0	47 clocks
f _{xx} (when OCKSm = 18H set)	0	0	1	1	37 clocks
f _{xx} /2 (when OCKSm = 10H set)	0	0	1	1	74 clocks
f _{xx} /3 (when OCKSm = 11H set)	0	0	1	1	111 clocks
f _{xx} (when OCKSm = 18H set)	0	1	0	×	16 clocks
f _{xx} /2 (when OCKSm = 10H set)	0	1	0	×	32 clocks
f _{xx} /3 (when OCKSm = 11H set)	0	1	0	×	48 clocks
f _{xx} /4 (when OCKSm = 12H set)	0	1	0	×	64 clocks
f _{xx}	0	1	1	0	16 clocks
f _{xx} (when OCKSm = 18H set)	0	1	1	1	13 clocks
f _{xx} /2 (when OCKSm = 10H set)	0	1	1	1	26 clocks
f _{xx} /3 (when OCKSm = 11H set)	0	1	1	1	39 clocks
f _{xx} (when OCKSm = 18H set)	1	1	0	×	10 clocks
f _{xx} /2 (when OCKSm = 10H set)	1	1	0	×	20 clocks
f _{xx} /3 (when OCKSm = 11H set)	1	1	0	×	30 clocks
f _{xx} /4 (when OCKSm = 12H set)	1	1	0	×	40 clocks
f _{xx} /5 (when OCKSm = 13H set)	1	1	0	×	50 clocks
f _{xx}	1	1	1	0	10 clocks

- Remarks**
1. m = 0, 1
 2. × = don't care

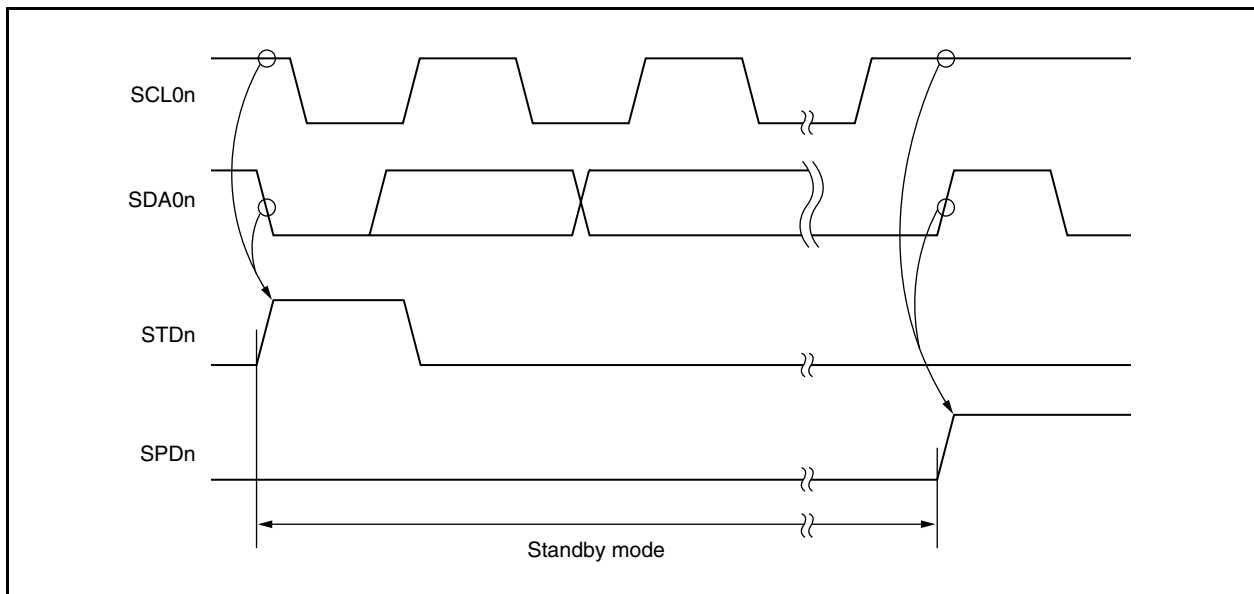
The communication reservation timing is shown below.

Figure 17-15. Communication Reservation Timing



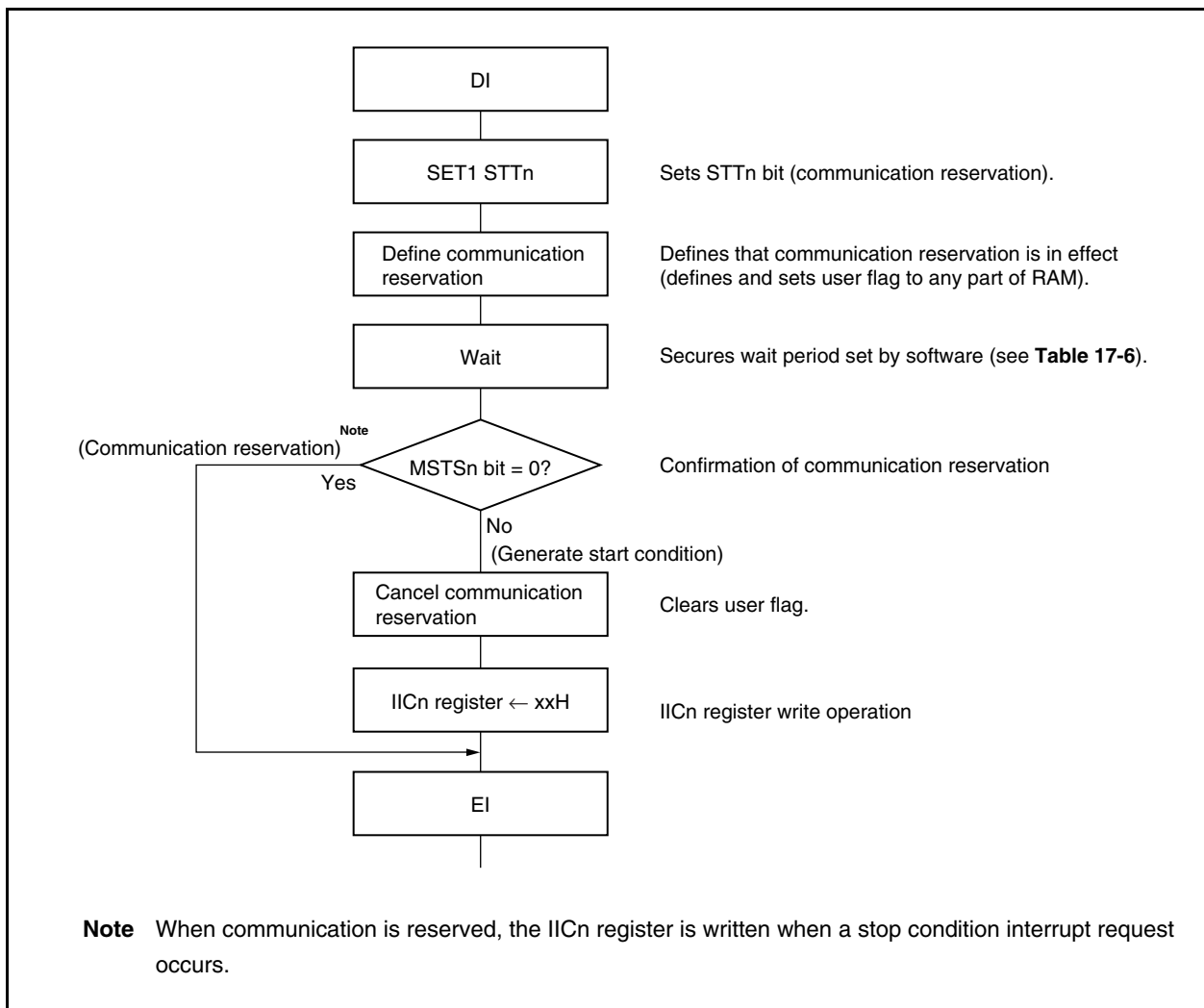
Communication reservations are accepted at the following timing. After the IICSn.STDn bit is set to 1, a communication reservation can be made by setting the IICn.STTn bit to 1 before a stop condition is detected.

Figure 17-16. Timing for Accepting Communication Reservations



The communication reservation flowchart is illustrated below.

Figure 17-17. Communication Reservation Flowchart



17.14.2 When communication reservation function is disabled (IICFn.IICRSVn bit = 1)

If the IICFn.STTn bit is set when the bus is not being used by the V850ES/JC3-L and V850ES/JE3-L in a bus communication, this request is rejected and a start condition is not generated. There are two modes in which the bus is not used by the V850ES/JC3-L and V850ES/JE3-L.

- When arbitration results in the V850ES/JC3-L and V850ES/JE3-L being neither the master nor a slave
- When an extension code is received and slave operation is disabled (\overline{ACK} is not returned and the bus was released when the IICFn.LRELn bit was set to 1).

To confirm whether the start condition was generated or request was rejected, check the IICFn.STCFn flag. The time shown in Table 17-7 is required until the STCFn flag is set after setting the STTn bit to 1. Therefore, secure the time by software.

Table 17-7. Wait Periods

OCKSENm	OCKSm1	OCKSm0	CLn1	CLn0	Wait Period
1	0	0	0	×	10 clocks
1	0	1	0	×	15 clocks
1	1	0	0	×	20 clocks
1	1	1	0	×	25 clocks
0	0	0	1	0	5 clocks

- Remarks**
1. ×: don't care
 2. m = 0, 1

17.15 Cautions

(1) When IICFn.STCENn bit = 0

Immediately after the I²C0n operation is enabled, the bus communication status (IICFn.IICBSYn 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 IICCLn register.

<2> Set the IICn.IICEn bit.

<3> Set the IICn.SPTn bit.

(2) When IICFn.STCENn bit = 1

Immediately after I²C0n operation is enabled, the bus released status (IICBSYn bit = 0) is recognized regardless of the actual bus status. To generate the first start condition (IICn.STTn bit = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) When the IICn.IICEn bit of the V850ES/JC3-L and V850ES/JE3-L 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 IICn.IICEn bit to 1 when the SCL0n and SDA0n lines are high level.

(4) Determine the operation clock frequency by the IICCLn, IICXn, and OCKSm registers before enabling the operation (IICn.IICEn bit = 1). To change the operation clock frequency, clear the IICn.IICEn bit to 0 once.

(5) After the IICn.STTn and IICn.SPTn bits have been set to 1, they must not be re-set without being cleared to 0 first.

(6) If transmission has been reserved, set the IICn.SPIEn 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 status will be released by writing communication data to I²Cn, then transferring will begin. If an interrupt is not generated by the detection of a stop condition, transmission will halt in the wait status because an interrupt request was not generated. However, it is not necessary to set the SPIEn bit to 1 for the software to detect the IICn.MSTS bit.

Remark m = 0, 1

17.16 Communication Operations

Next the following three operations are shown using flowcharts.

(1) Master operation in single master system

The flowchart when using the V850ES/JC3-L and V850ES/JE3-L 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_{0n} 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 V850ES/JC3-L and V850ES/JE3-L take part in a communication with bus released state.

This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the V850ES/JC3-L and V850ES/JE3-L lose in arbitration and are specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

(3) Slave operation

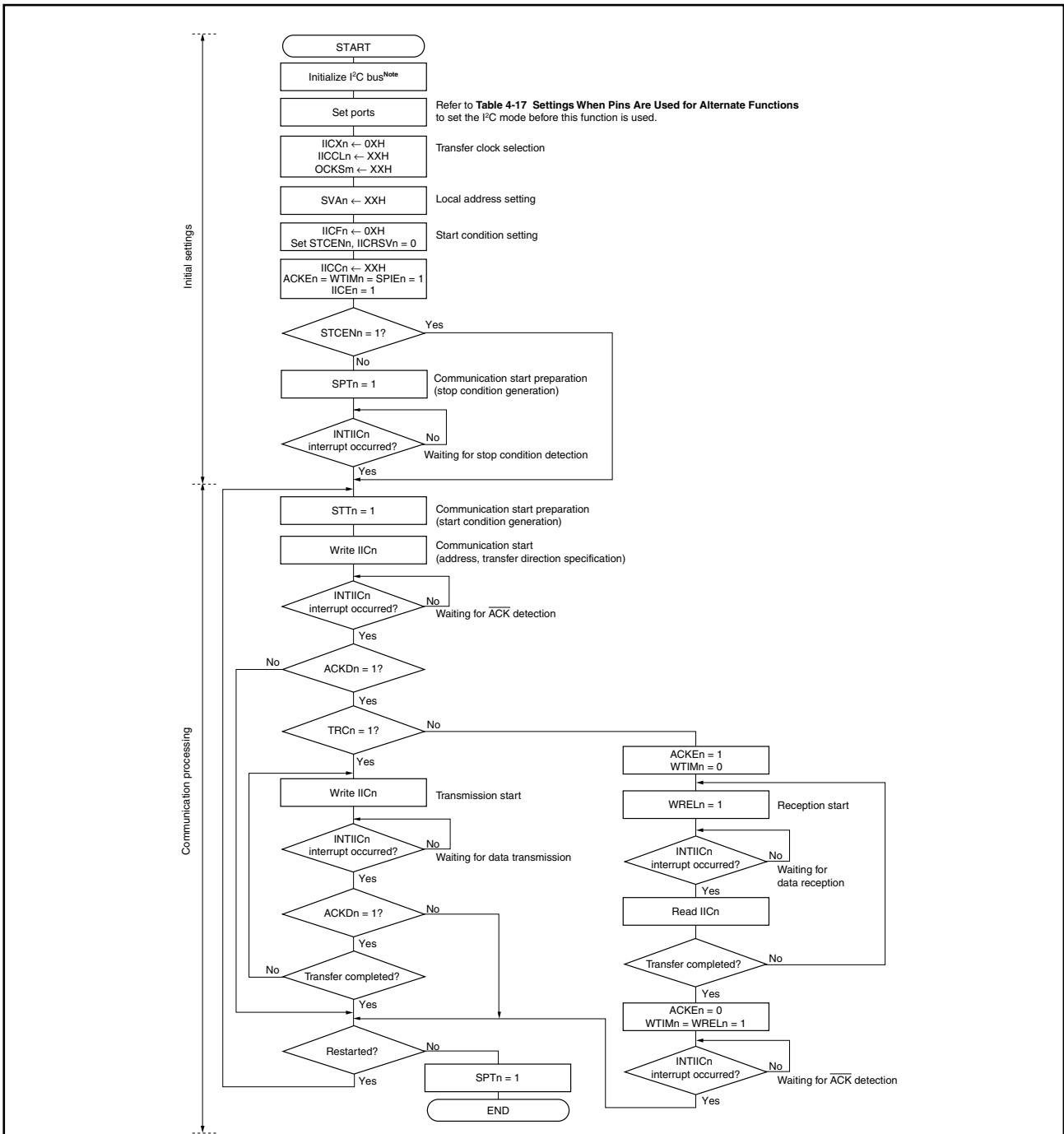
An example of when the V850ES/JC3-L and V850ES/JE3-L are used as the slave of the I²C_{0n} bus is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIIC_n interrupt occurrence (communication waiting). When the INTIIC_n interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

17.16.1 Master operation in single master system

Figure 17-18. Master Operation in Single Master System



Note Release the I²C0n bus (SCL0n, SDA0n pins = high level) in compliance with the specifications of the product involved in the communication.

For example, when the EEPROM™ outputs a low level to the SDA0n pin, set the SCL0n pin as an output pin and output clock pulses from that output pin until the SDA0n pin is constantly high level.

Remarks 1. For the transmission and reception formats, conform to the specifications of the product involved in the communication.

2. m = 0, 1

17.16.2 Master operation in multimaster system

Figure 17-19. Master Operation in Multimaster System (1/3)

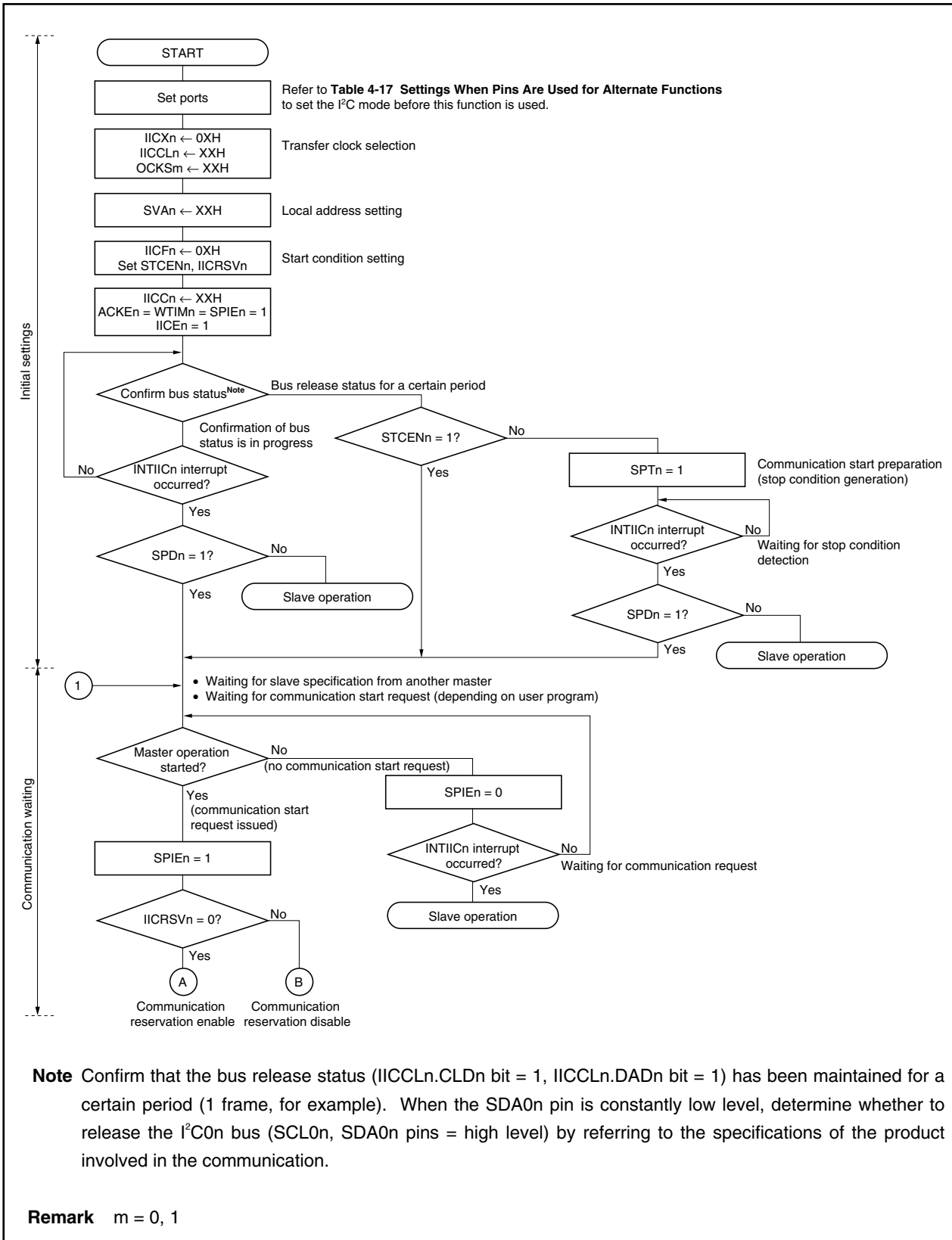


Figure 17-19. Master Operation in Multimaster System (2/3)

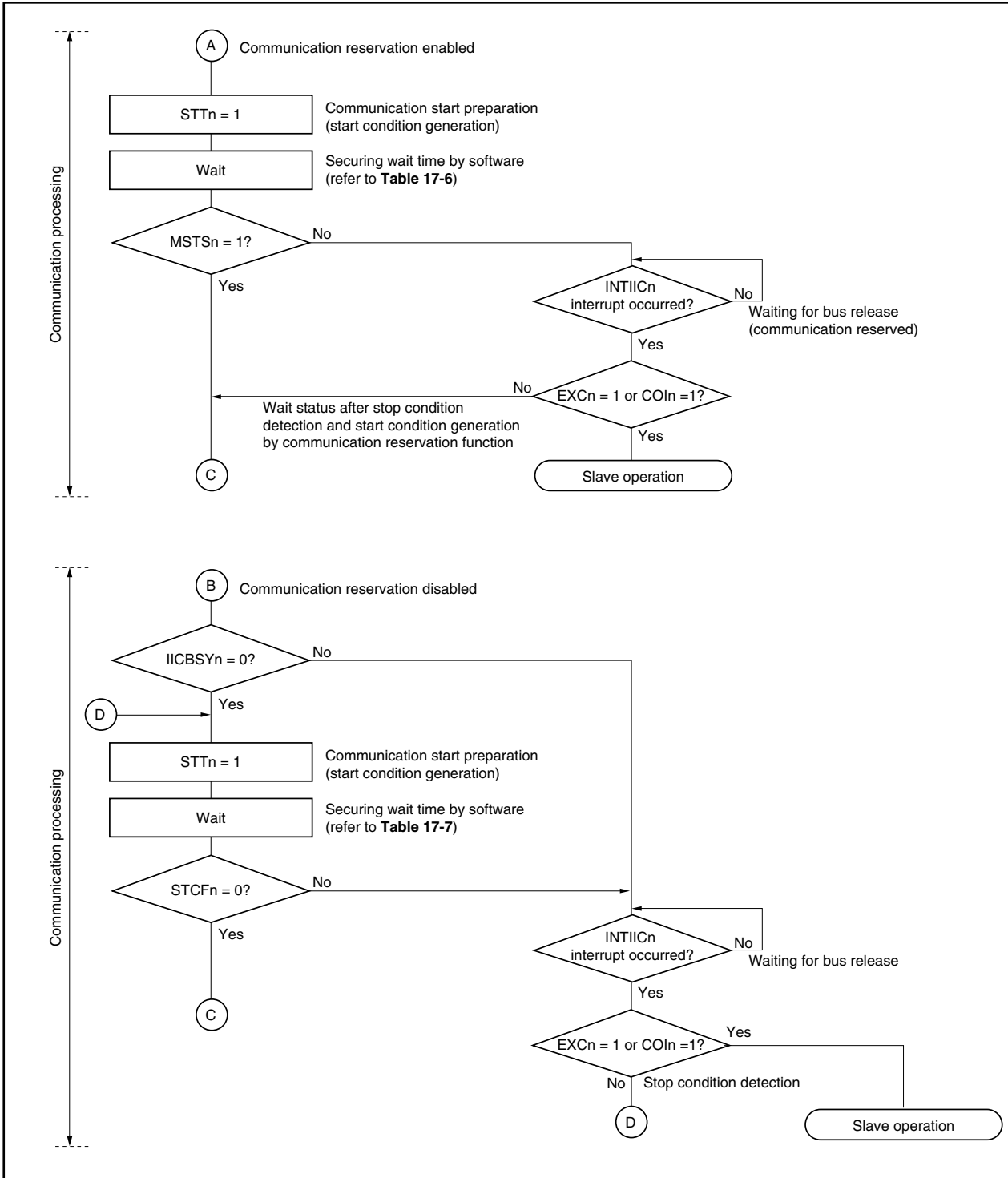
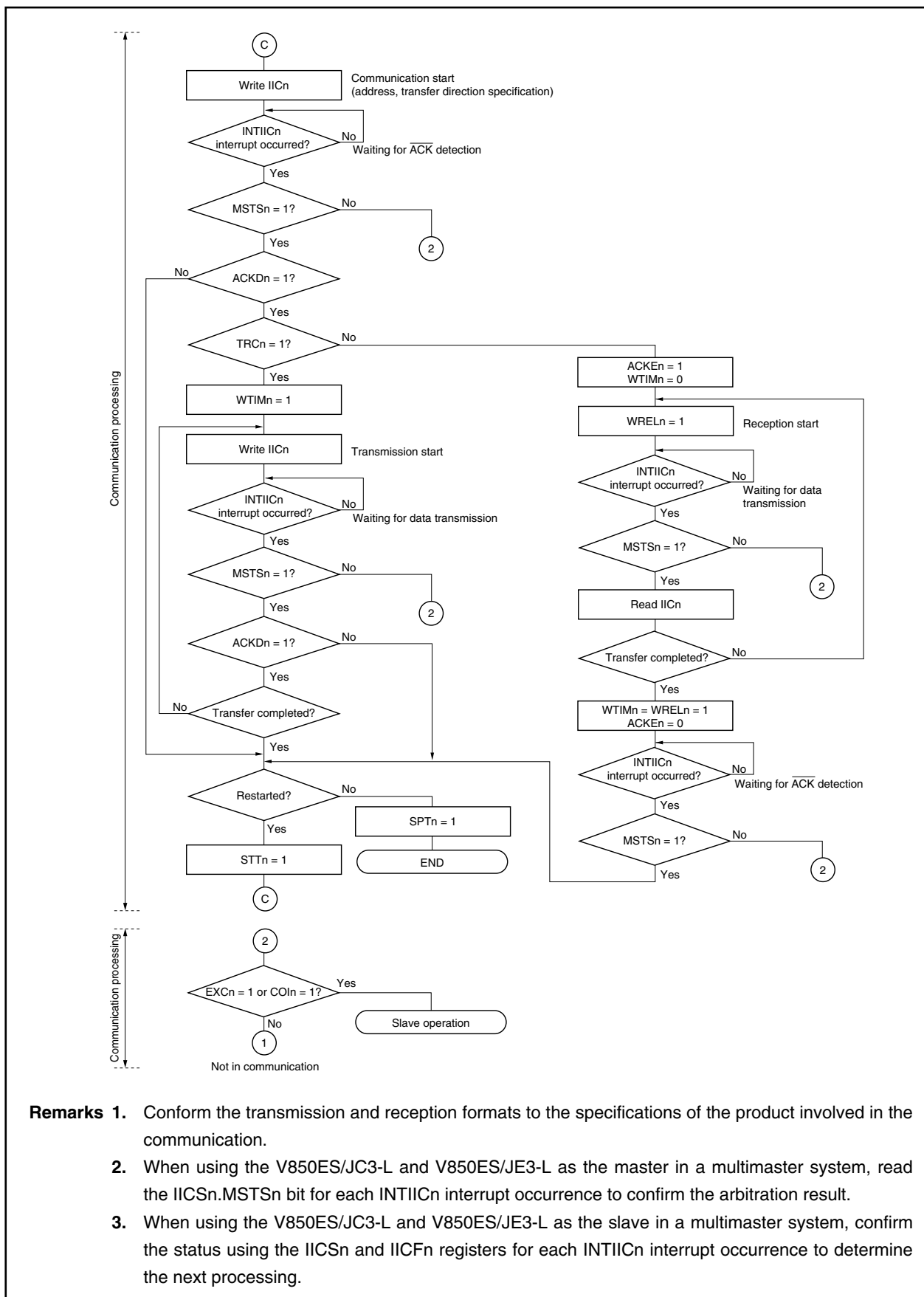


Figure 17-19. Master Operation in Multimaster System (3/3)



- Remarks 1.** Conform the transmission and reception formats to the specifications of the product involved in the communication.
- 2.** When using the V850ES/JC3-L and V850ES/JE3-L as the master in a multimaster system, read the IICSn.MSTSn bit for each INTIICn interrupt occurrence to confirm the arbitration result.
- 3.** When using the V850ES/JC3-L and V850ES/JE3-L as the slave in a multimaster system, confirm the status using the IICSn and IICFn registers for each INTIICn interrupt occurrence to determine the next processing.

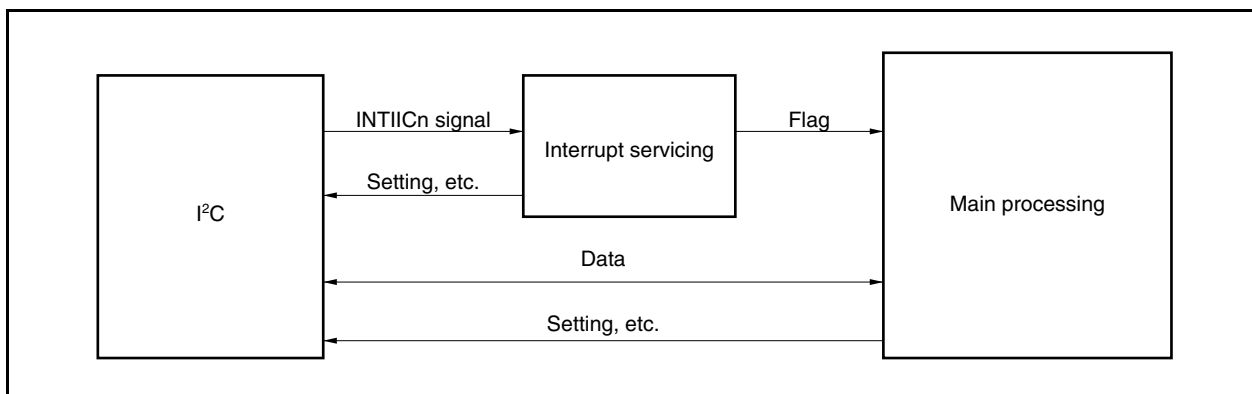
17.16.3 Slave operation

The following shows the processing procedure of the slave operation.

Basically, the operation of the slave device is event-driven. Therefore, processing by an INTIICn 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 INTIICn interrupt servicing performs only status change processing and that the actual data communication is passing during the main processing.

Figure 17-20. Outline of Software During Slave Operation



Therefore, the following three flags are prepared so that the data transfer processing can be performed by passing these flags to the main processing instead of INTIICn 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 INTIICn 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 clear 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 IICSn.TRCn bit.

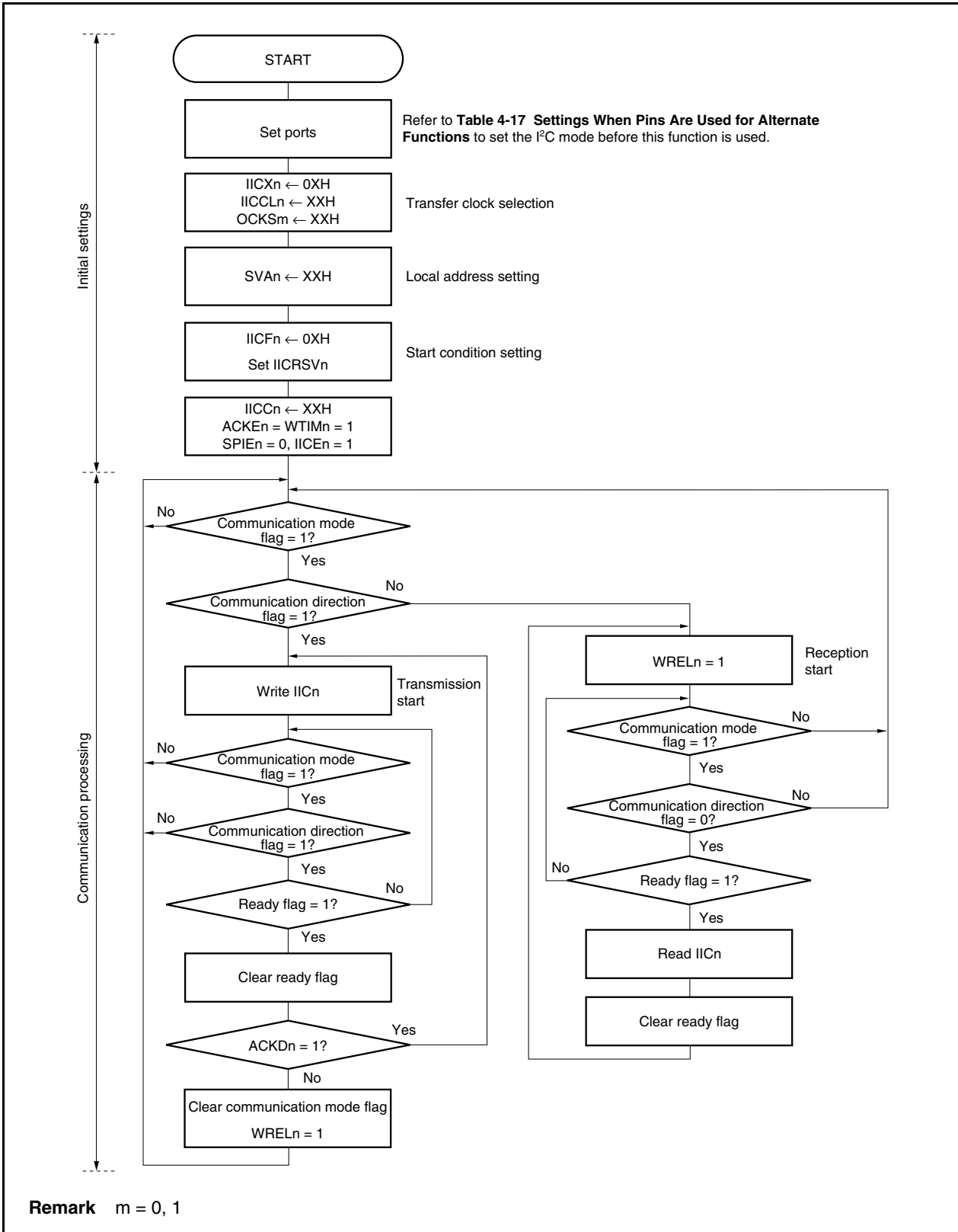
The following shows the operation of the main processing block during slave operation.

I²C0n is started and waits for the communication enabled status. When communication is enabled, transfer is executed 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, transmission is repeated until the master device stops returning $\overline{\text{ACK}}$. When the master device stops returning $\overline{\text{ACK}}$, transfer is complete.

For reception, the required number of data items are received and \overline{ACK} is not returned 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.

Figure 17-21. Slave Operation Flowchart (1)

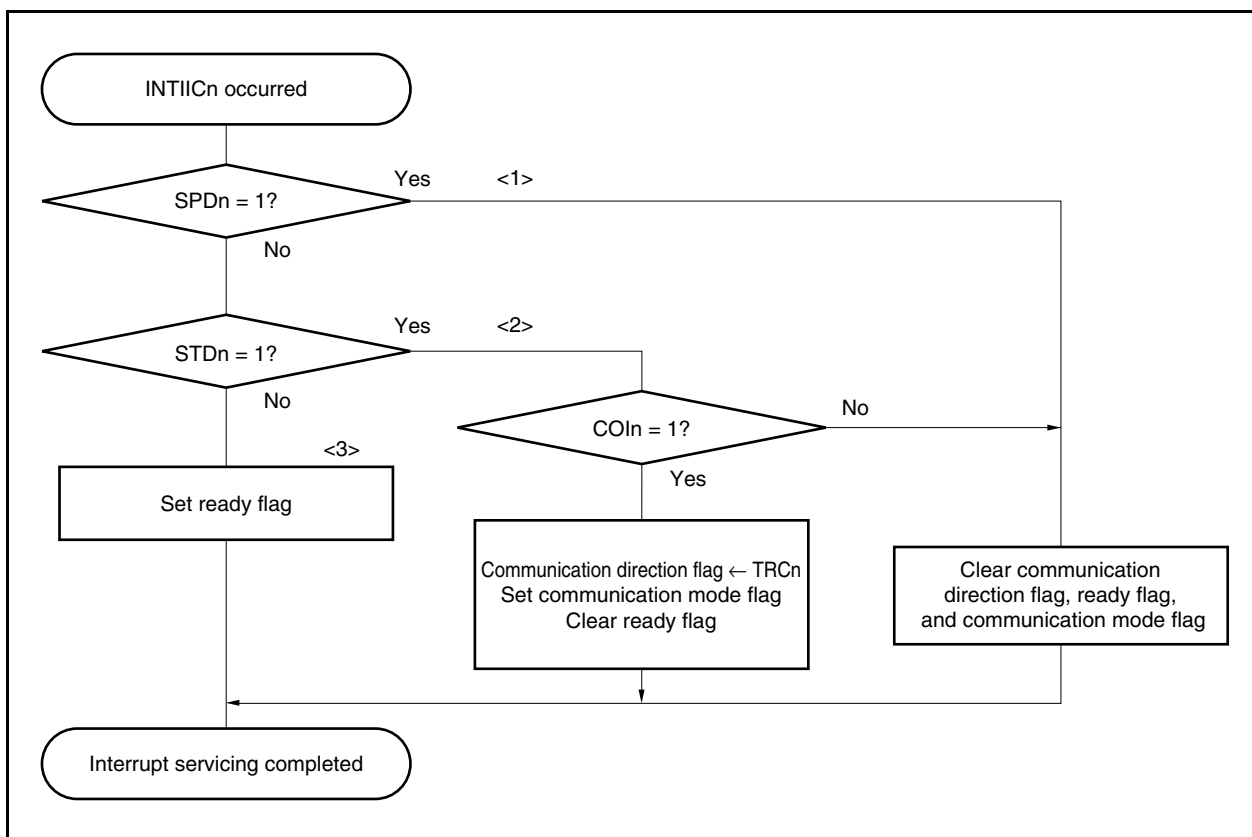


The following shows an example of the processing of the slave device by an INTIICn interrupt (it is assumed that no extension codes are used here). During an INTIICn interrupt, the status is confirmed and the following steps are executed.

- <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 I²C0n bus remains in the wait status.

Remark <1> to <3> above correspond to <1> to <3> in **Figure 17-22 Slave Operation Flowchart (2)**.

Figure 17-22. Slave Operation Flowchart (2)



17.17 Timing of Data Communication

When using I²C bus mode, the master device outputs 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 IICSn.TRCn bit value, which specifies the data transfer direction, and then starts serial communication with the slave device.

The shift operation of the IICn register is synchronized with the falling edge of the serial clock pin (SCL0n). The transmit data is transferred to the SO latch and is output (MSB first) via the SDA0n pin.

Data that is input via the SDA0n pin is captured by the IICn register at the rising edge of the SCL0n pin.

The data communication timing is shown below.

**Figure 17-23. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)**

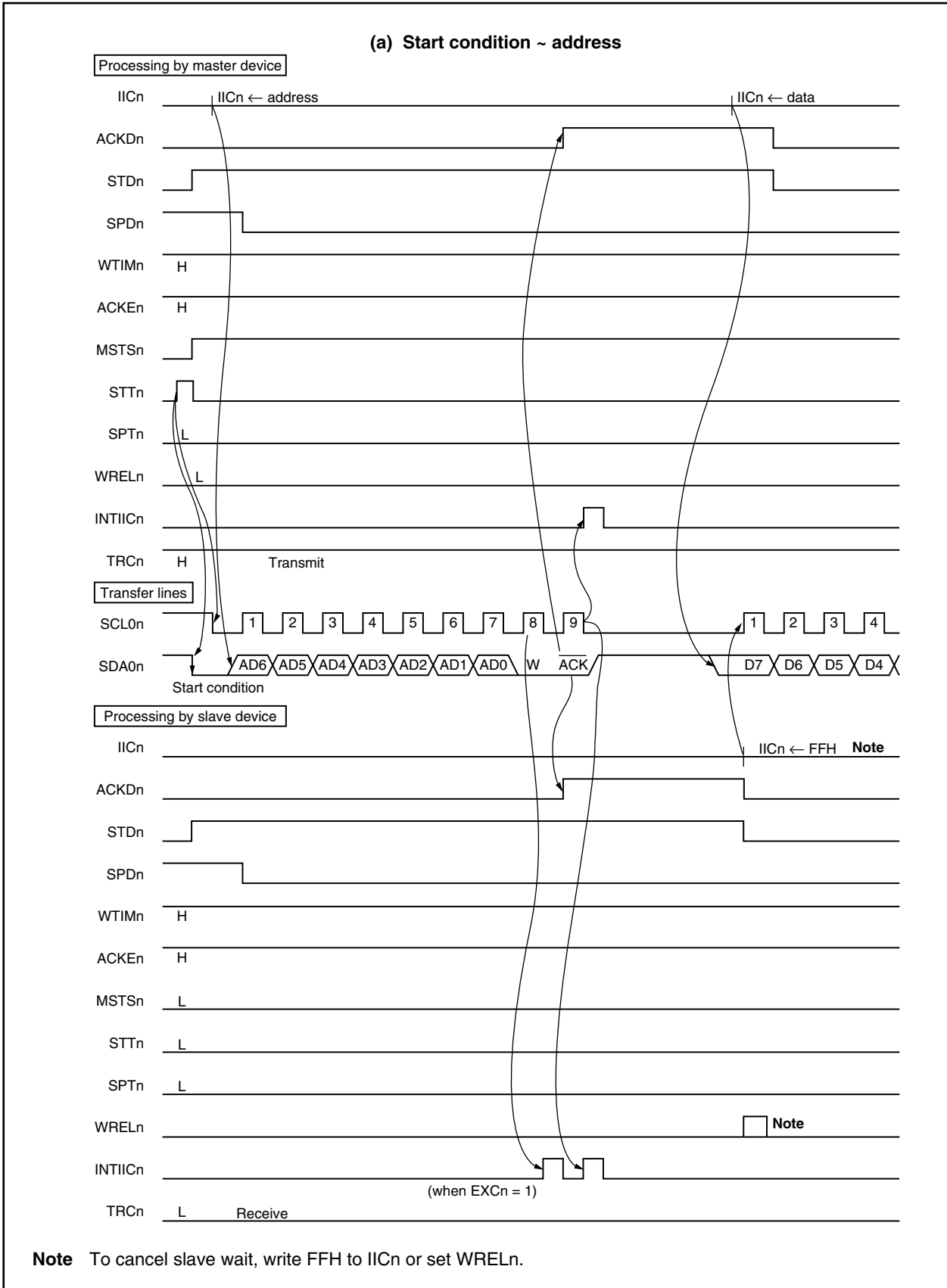


Figure 17-23. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)

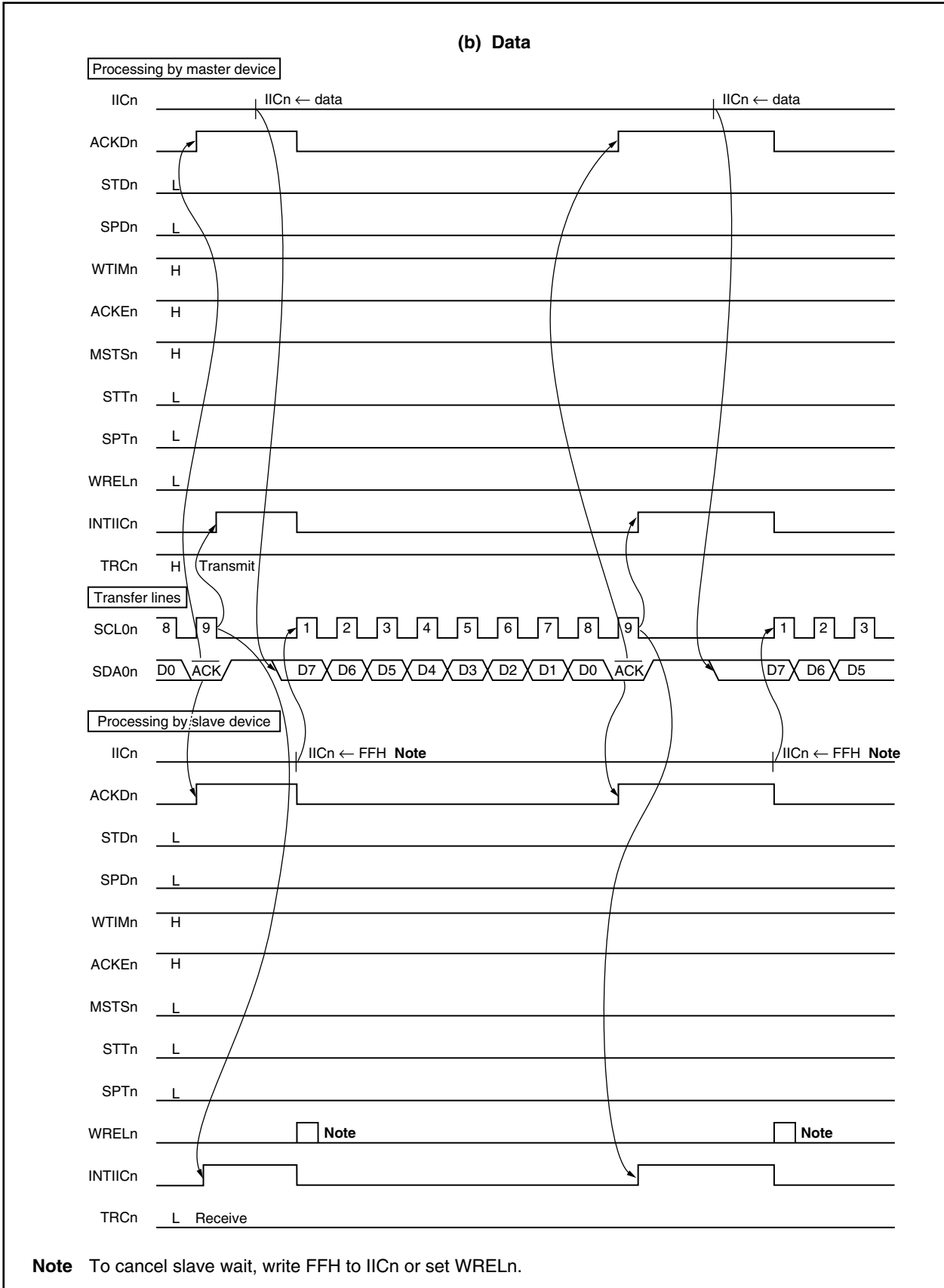


Figure 17-23. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

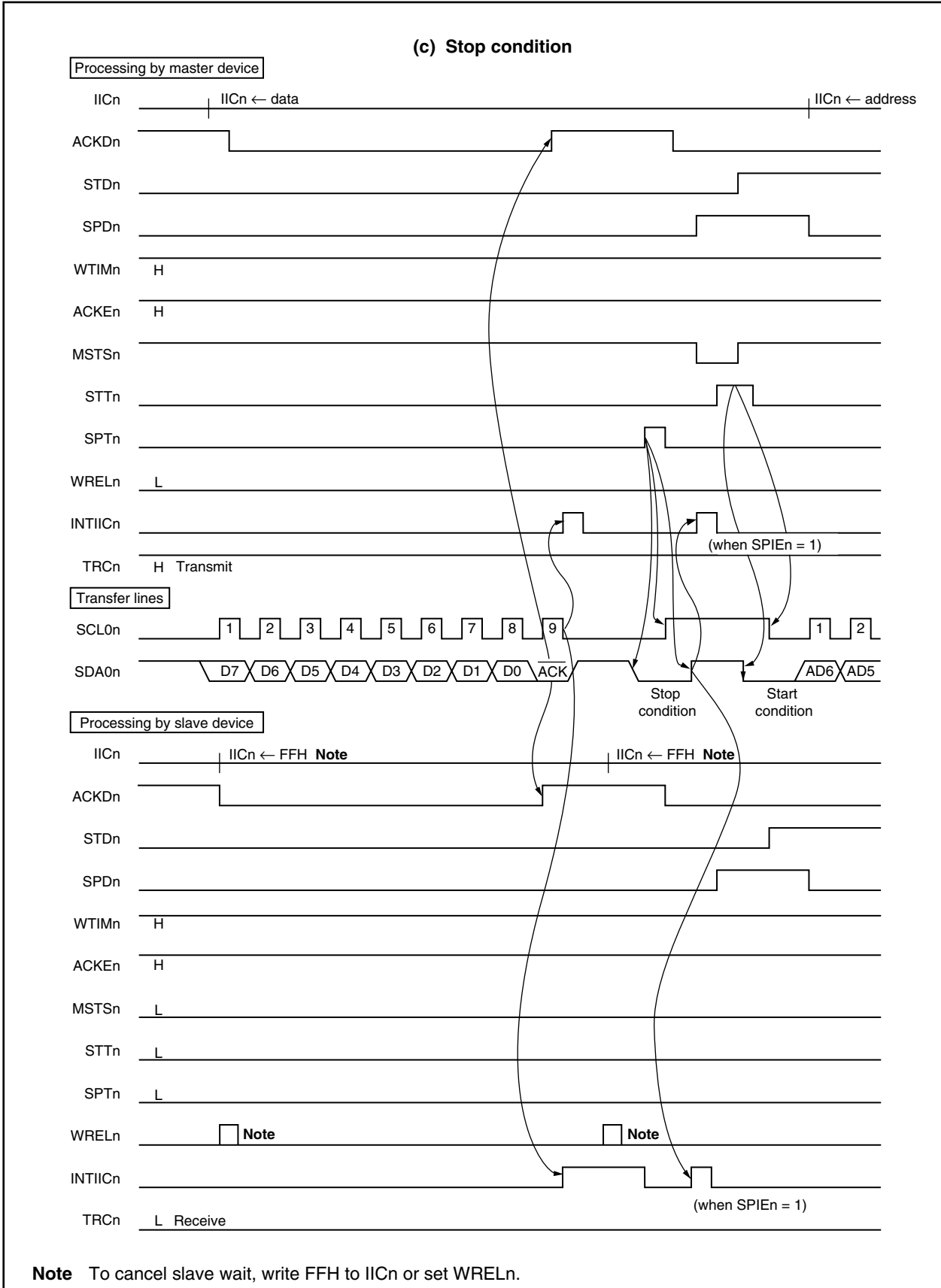


Figure 17-24. Example of Slave to Master Communication (When 8-Clock Wait Is Selected for Master and 9-Clock Wait Is Selected for Slave) (1/3)

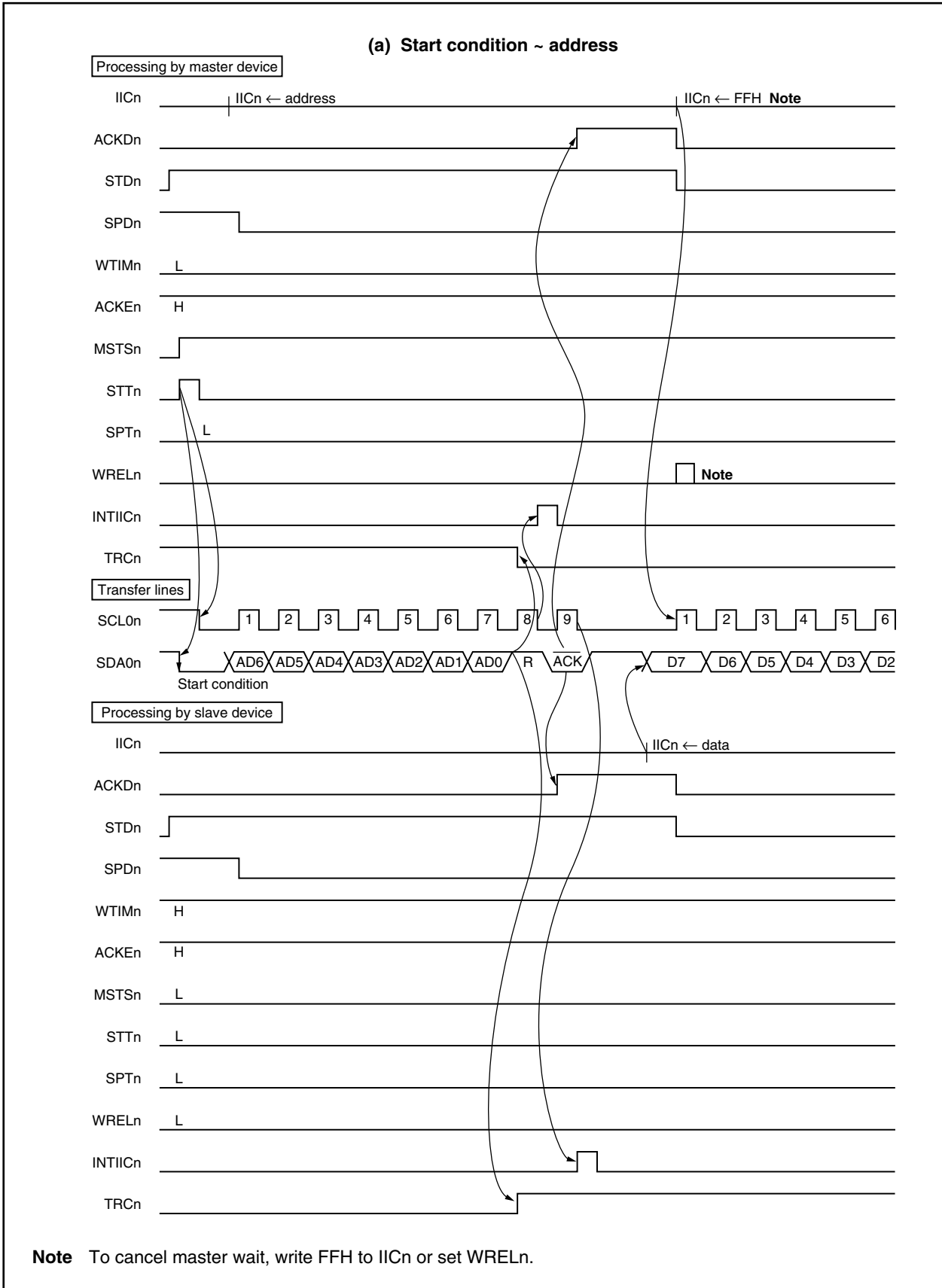


Figure 17-24. Example of Slave to Master Communication (When 8-Clock Wait Is Selected for Master and 9-Clock Wait Is Selected for Slave (2/3))

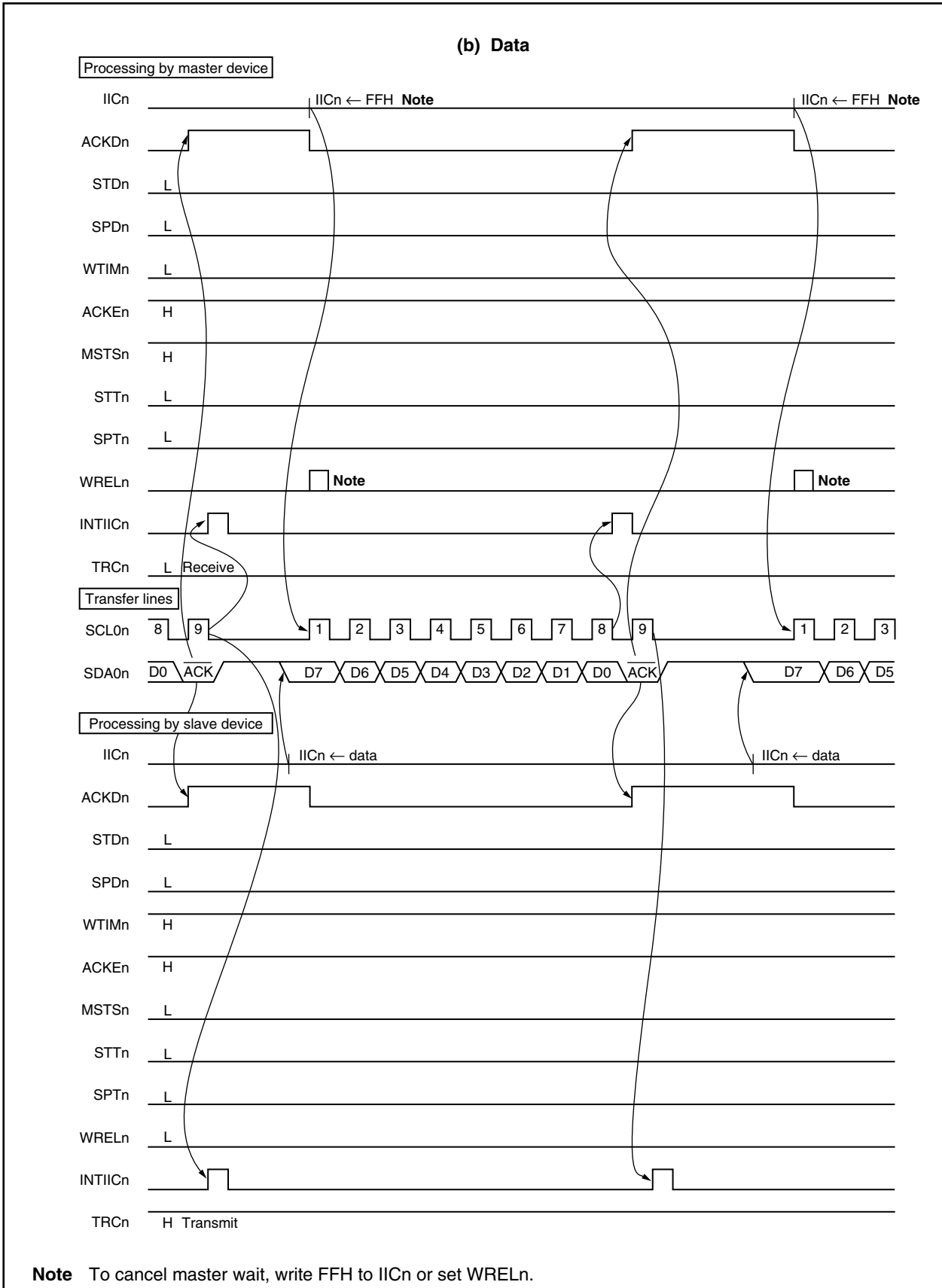
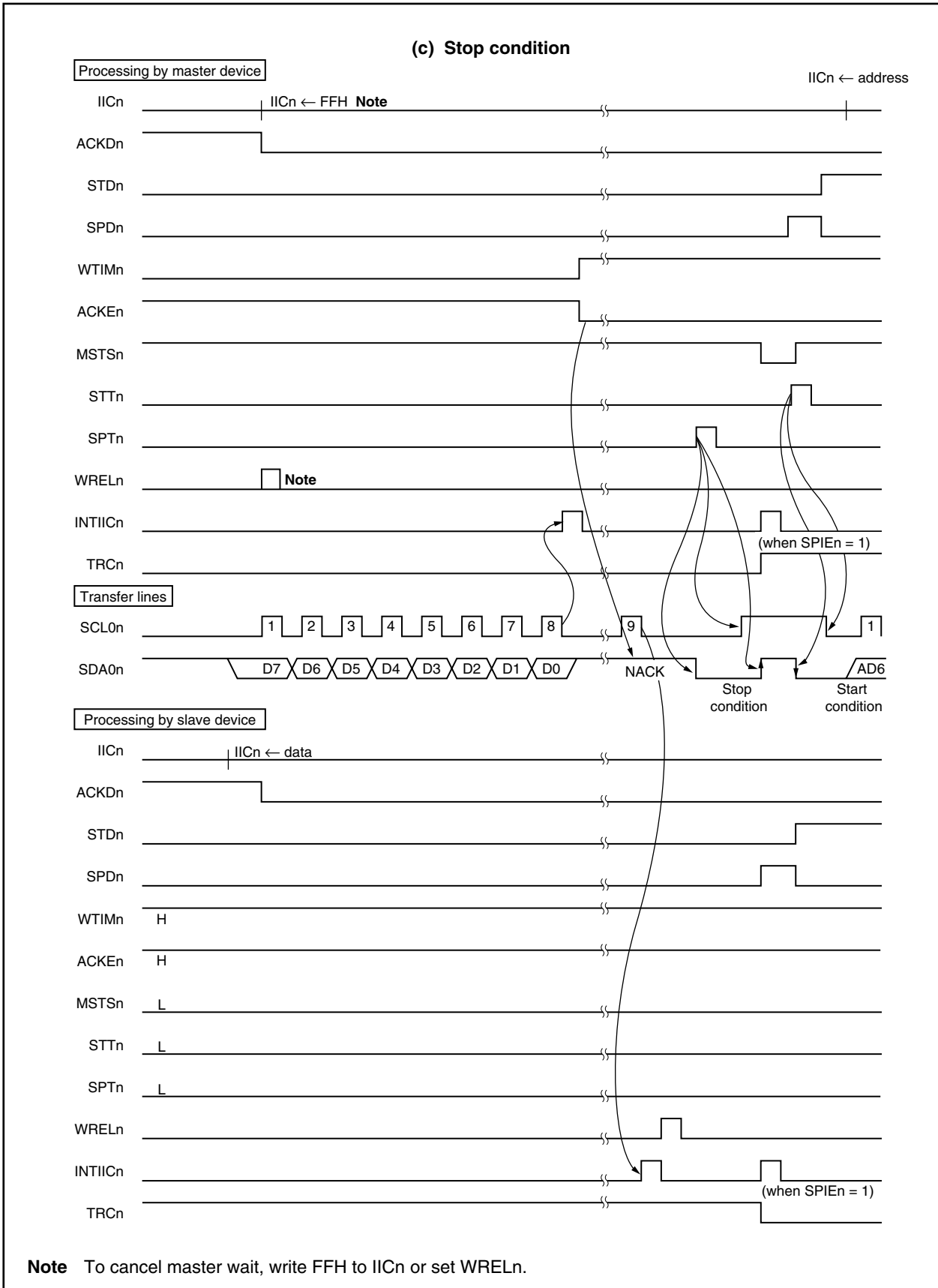


Figure 17-24. Example of Slave to Master Communication (When Wait Is Changed from 8 Clocks to 9 Clocks for Master and 9-Clock Wait Is Selected for Slave) (3/3)



CHAPTER 18 DMA FUNCTION (DMA CONTROLLER)

The V850ES/JC3-L and V850ES/JE3-L include a direct memory access (DMA) controller (DMAC) that executes and controls DMA transfer.

The DMAC controls data transfer between memory and I/Os, or between I/Os based on DMA requests issued by on-chip peripheral I/O (serial interfaces, timer/counters, and A/D converter), interrupts from external input pins, or software triggers (memory refers to internal RAM).

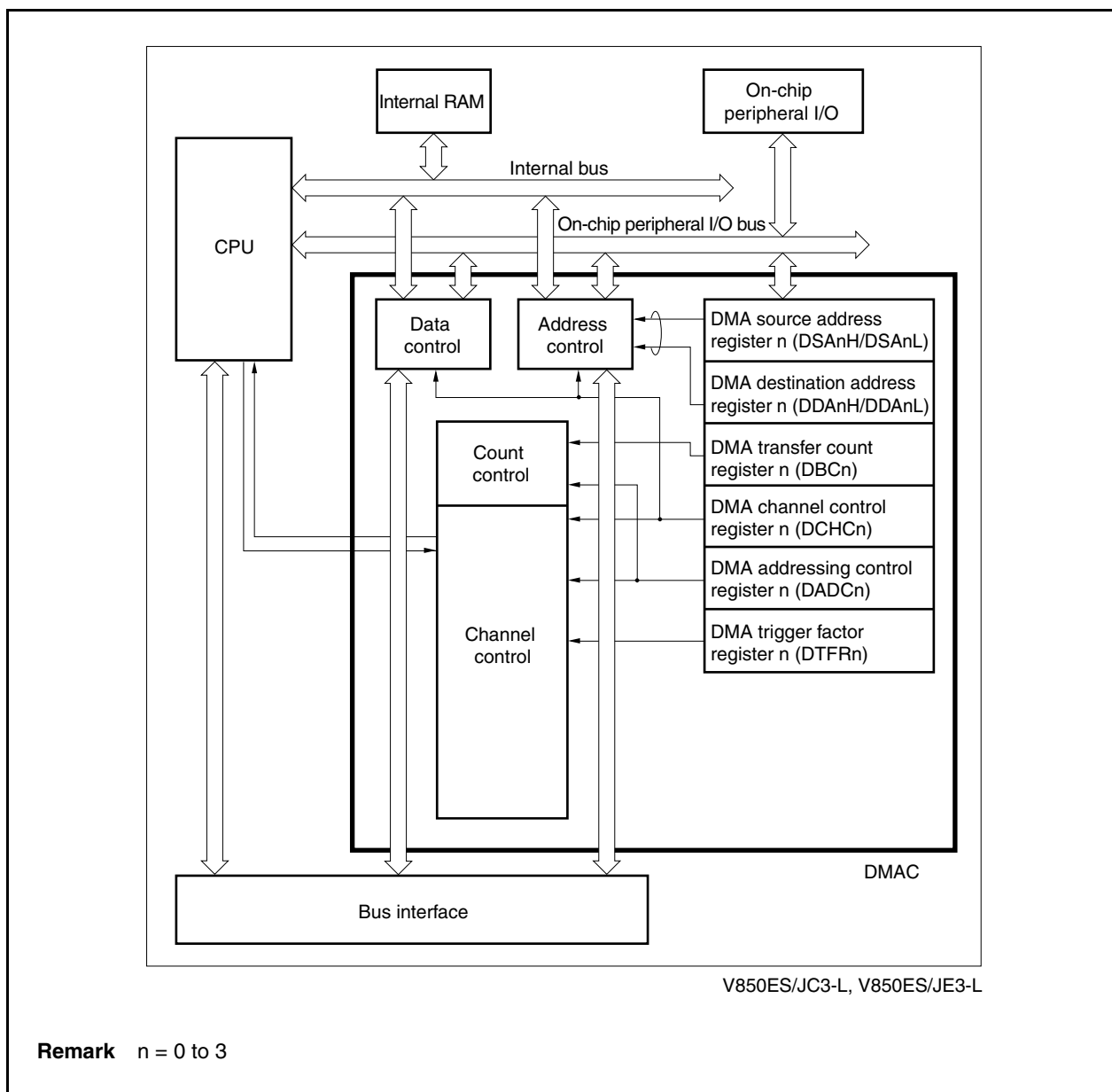
18.1 Features

- 4 independent DMA channels
- Transfer unit: 8/16 bits
- Maximum transfer count: 65,536 (2^{16})
- Program execution using internal ROM during DMA transfer
- Transfer type: Two-cycle transfer
 - Data transfer between buses that have different bus widths
- Transfer mode: Single transfer mode
- Transfer requests
 - Request by interrupts from on-chip peripheral I/Os (serial interfaces, timer/counters, A/D converter) or interrupts from external input pin
 - Requests triggered by software
- Transfer sources and destinations
 - Internal RAM ↔ On-chip peripheral I/O
 - On-chip peripheral I/O ↔ On-chip peripheral I/O

18.2 Configuration

The block diagram of the DMAC is shown below.

Figure 18-1. Block Diagram of DMAC



The DMAC includes the following hardware.

Table 18-1. Configuration of DMAC

Item	Configuration
Registers	DMA source address registers 0 to 3 (DSA0 to DSA3) DMA destination address registers 0 to 3 (DDA0 to DDA3) DMA transfer count register 0 to 3 (DBC0 to DBC3) DMA addressing control registers 0 to 3 (DADC0 to DADC3) DMA channel control registers 0 to 3 (DCHC0 to DCHC3) DMA trigger factor registers 0 to 3 (DTFR0 to DTFR3)

18.3 Registers

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

The DSA0 to DSA3 registers set the DMA source addresses (26 bits each) for DMA channel n (n = 0 to 3). These registers are divided into two 16-bit registers, DSAnH and DSAnL. These registers can be read or written in 16-bit units.

After reset: Undefined	R/W	Address: DSA0H FFFFF082H, DSA1H FFFFF08AH, DSA2H FFFFF092H, DSA3H FFFFF09AH, DSA0L FFFFF080H, DSA1L FFFFF088H, DSA2L FFFFF090H, DSA3L FFFFF098H
DSAnH (n = 0 to 3)	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	IR 0 0 0 0 0 SA25 SA24 SA23 SA22 SA21 SA20 SA19 SA18 SA17 SA16
DSAnL (n = 0 to 3)	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	SA15 SA14 SA13 SA12 SA11 SA10 SA9 SA8 SA7 SA6 SA5 SA4 SA3 SA2 SA1 SA0
IR	Specification of DMA transfer source	
0	On-chip peripheral I/O	
1	Internal RAM	
SA25 to SA16	Set the address (A25 to A16) of the DMA transfer source (the default value is undefined). During DMA transfer, the next DMA transfer source address is held. When DMA transfer is completed, the DMA address set first is held.	
SA15 to SA0	Set the address (A15 to A0) of the DMA transfer source (the default value is undefined). During DMA transfer, the next DMA transfer source address is held. When DMA transfer is completed, the DMA address set first is held.	

Cautions

1. Be sure to clear bits 14 to 10 of the DSAnH register to 0.
2. Set the DSAnH and DSAnL registers during one of the following periods in which DMA transfer is disabled (DCHCn.Enn bit = 0).
 - Period from after reset to start of first DMA transfer
 - Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer
 - Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer
3. When the value of the DSAn register is read, two 16-bit registers, DSAnH and DSAnL, are read. If reading and updating conflict, the value being updated may be read (see 18.13 Cautions).
4. DMA transfer of misaligned 16-bit data with is not supported.
If an odd address is specified as the transfer source, the least significant bit of the address is forcibly handled as being 0.
5. Following a reset, set the DSAnH, DSAnL, DDAnH, DDAnL, and DBCn registers before starting DMA transfer. If these registers are not set, DMA transfers are not guaranteed.

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

The DDA0 to DDA3 registers set the DMA destination address (26 bits each) for DMA channel n (n = 0 to 3).

These registers are divided into two 16-bit registers, DDAnH and DDAnL.

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

After reset: Undefined R/W Address: DDA0H FFFFF086H, DDA1H FFFFF08EH,
DA2H FFFFF096H, DDA3H FFFFF09EH,
DDA0L FFFFF084H, DDA1L FFFFF08CH,
DDA2L FFFFF094H, DDA3L FFFFF09CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDAnH (n = 0 to 3)	IR	0	0	0	0	0	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDAnL (n = 0 to 3)	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

IR	Specification of DMA transfer destination
0	On-chip peripheral I/O
1	Internal RAM

DA25 to DA16	Set the address (A25 to A16) of the DMA transfer destination (the default value is undefined). During DMA transfer, the next DMA transfer destination address is held. When DMA transfer is completed, the DMA transfer source address set first is held.
--------------	---

DA15 to DA0	Set the address (A15 to A0) of the DMA transfer destination (the default value is undefined). During DMA transfer, the next DMA transfer destination address is held. When DMA transfer is completed, the DMA transfer source address set first is held.
-------------	--

- Cautions**
- Be sure to clear bits 14 to 10 of the DDAnH register to 0.
 - Set the DDAnH and DDAnL registers during one of the following periods in which DMA transfer is disabled (DCHCn.Enn bit = 0).
 - Period from after reset to start of first DMA transfer
 - Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer
 - Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer
 - When the value of the DDAn register is read, two 16-bit registers, DDAnH and DDAnL, are read. If reading and updating conflict, a value being updated may be read (see 18.13 Cautions).
 - DMA transfer of misaligned 16-bit data is not supported.
If an odd address is specified as the transfer destination, the least significant bit of the address is forcibly handled as being 0.
 - Following a reset, set the DSAnH, DSAnL, DDAnH, DDAnL, and DBCn registers before starting DMA transfer. If these registers are not set, DMA transfers are not guaranteed.

(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 hold the remaining transfer count during DMA transfer.

These registers are decremented by 1 per transfer regardless of the transfer data unit (8/16 bits), and the transfer is terminated if a borrow occurs.

The number of transfers specified first is held when DMA transfer is complete.

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

After reset: Undefined R/W Address: DBC0 FFFF0C0H, DBC1 FFFF0C2H,
 DBC2 FFFF0C4H, DBC3 FFFF0C6H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0

DBCn
(n = 0 to 3)

BC15 to BC0	Byte transfer count setting or remaining byte transfer count during DMA transfer
0000H	1st byte transfer or remaining byte transfer count
0001H	2nd byte transfer or remaining byte transfer count
⋮	⋮
FFFFH	65,536 (2 ¹⁶)th byte transfer or remaining byte transfer count

Cautions 1. Set the DBCn register during one of the following periods in which DMA transfer is disabled (DCHCn.Enn bit = 0).

- Period from after reset to start of first DMA transfer
- Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer
- Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer

2. Following a reset, set the DSA nH, DSA nL, DDAnH, DDAnL, and DBCn registers before starting DMA transfer. If these registers are not set, DMA transfers are not guaranteed.

(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 can be read or written in 16-bit units.

Reset sets these registers to 0000H.

After reset: 0000H		R/W	Address: DADC0 FFFFF0D0H, DADC1 FFFFF0D2H, DADC2 FFFFF0D4H, DADC3 FFFFF0D6H							
DADCn (n = 0 to 3)	15	14	13	12	11	10	9	8		
	0	DS0	0	0	0	0	0	0		
	7	6	5	4	3	2	1	0		
	SAD1	SAD0	DAD1	DAD0	0	0	0	0		
	DS0	Setting of transfer data size								
	0	8 bits								
	1	16 bits								
	SAD1	SAD0	Setting of count direction of the transfer source address							
	0	0	Increment							
	0	1	Decrement							
	1	0	Fixed							
	1	1	Setting prohibited							
	DAD1	DAD0	Setting of count direction of the destination address							
	0	0	Increment							
	0	1	Decrement							
	1	0	Fixed							
	1	1	Setting prohibited							

Cautions

- Be sure to clear bits 15, 13 to 8, and 3 to 0 of the DADCn register to 0.
- Set the DADCn register during one of the following periods in which DMA transfer is disabled (DCHCn.Enn bit = 0).
 - Period from after reset to start of first DMA transfer
 - Period from after channel initialization by DCHCn.INITn bit to start of DMA transfer
 - Period from after completion of DMA transfer (DCHCn.TCn bit = 1) to start of the next DMA transfer
- The DS0 bit specifies the size of the transfer data, and does not control bus sizing.
- If the transfer data size is set to 16 bits (DS0 bit = 1), transfer cannot be started from an odd address. Transfer is always started from an address with the first bit of the lower address aligned to 0.
- If DMA transfer is executed on an on-chip peripheral I/O register (as the transfer source or destination), be sure to specify the same transfer size as the register size. For example, to execute DMA transfer on an 8-bit register, be sure to specify 8-bit transfer.

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

The DCHC0 to DCHC3 registers are 8-bit registers that control DMA transfer for DMA channel n.

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

Reset sets these registers to 00H.

After reset: 00H R/W Address: DCHC0 FFFFF0E0H, DCHC1 FFFFF0E2H, DCHC2 FFFFF0E4H, DCHC3 FFFFF0E6H

	<7>	6	5	4	3	<2>	<1>	<0>
DCHCn	TCn ^{Note 1}	0	0	0	0	INITn ^{Note 2}	STGn ^{Note 2}	Enn

(n = 0 to 3)

TCn^{Note 1}	Status flag indicating whether DMA transfer via DMA channel n is complete
0	DMA transfer is not complete.
1	DMA transfer is complete.
This bit is set to 1 at the last DMA transfer and cleared to 0 when it is read.	

INITn^{Note 2}	If the INITn bit is set to 1 with DMA transfer disabled (Enn bit = 0), the DMA transfer status can be initialized.
-------------------------------	--

STGn^{Note 2}	This is a software startup trigger for DMA transfer. If this bit is set to 1 in the DMA transfer enabled state (TCn bit = 0, Enn bit = 1), DMA transfer is started.
------------------------------	--

Enn	Setting of whether DMA transfer via DMA channel n is to be enabled or disabled
0	DMA transfer disabled
1	DMA transfer enabled
DMA transfer is enabled when the Enn bit is set to 1. When DMA transfer is completed (when a terminal count is generated), this bit is automatically cleared to 0. To abort DMA transfer, clear the Enn bit to 0 by software. To resume DMA transfer, set the Enn bit to 1 again. When aborting or resuming DMA transfer, be sure to follow the procedure described in 18.13 (5) Procedure for temporarily stopping DMA transfer.	

Notes

1. The TCn bit is read-only.
2. The INITn and STGn bits are write-only.

Cautions

1. Be sure to clear bits 6 to 3 of the DCHCn register to 0.
2. When DMA transfer is completed (when a terminal count is generated), the Enn bit is cleared to 0 and then the TCn bit is set to 1. If the DCHCn register is read while its bits are being updated, a value indicating “transfer not completed and transfer is disabled” (TCn bit = 0 and Enn bit = 0) may be read.

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

The DTFR0 to DTFR3 registers are 8-bit registers that control the DMA transfer start trigger via interrupt request signals from on-chip peripheral I/O.

The interrupt request signals set by these registers serve as DMA transfer start factors.

These registers can be read or written in 8-bit units. However, DF_n bit can be read or written in 1-bit units.

Reset sets these registers to 00H.

After reset: 00H R/W Address: DTFR0 FFFFF810H, DTFR1 FFFFF812H,
DTFR2 FFFFF814H, DTFR3 FFFFF816H

DTFR _n	<7>	6	5	4	3	2	1	0
	DF _n	0	IFC _{n5}	IFC _{n4}	IFC _{n3}	IFC _{n2}	IFC _{n1}	IFC _{n0}

(n = 0 to 3)

DF _n ^{Note}	DMA transfer request status flag
0	No DMA transfer request
1	DMA transfer request

Note Do not write 1 to the DF_n bit by using software. Write 0 to this bit to clear a DMA transfer request if an interrupt specified as the DMA transfer start factor occurs while DMA transfer is disabled.

Cautions

1. Set the IFC_{n5} to IFC_{n0} bits during one of the following periods in which DMA transfer is disabled (DCHC_n.Enn bit = 0).
 - Period from after reset to start of first DMA transfer
 - Period from after channel initialization by DCHC_n.INIT_n bit to start of DMA transfer
 - Period from after completion of DMA transfer (DCHC_n.TC_n bit = 1) to start of the next DMA transfer
2. Be sure to follow the steps below when changing the DTFR_n register settings. (n = 0 to 3, m = 0 to 3, n ≠ m)
 - <1> Stop the DMA_n operation of the channel to be rewritten (DCHC_n.Enn bit = 0).
 - <2> Change the DTFR_n register settings. (Be sure to set DF_n bit = 0 and change the settings in the 8-bit manipulation.)
 - <3> Confirm that DF_n bit = 0. (Stop the interrupt generation source operation beforehand.)
 - <4> Enable the DMA_n operation (Enn bit = 1).
3. An interrupt request that is generated in the standby mode (IDLE1, IDLE2, STOP, or sub-IDLE mode) does not start the DMA transfer cycle (nor is the DF_n bit set to 1).
4. If a DMA start factor is selected by the IFC_{n5} to IFC_{n0} bits, the DF_n bit is set to 1 when an interrupt from the selected on-chip peripheral I/O occurs, regardless of whether the DMA transfer is enabled. If DMA is enabled in this status, DMA transfer immediately starts.

Remark For the IFC_{n5} to IFC_{n0} bits, see **Table 18-2 DMA Start Factors**.

Table 18-2. DMA Start Factors (1/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt Source	JC3L(40)	JC3L(48)	JE3L
0	0	0	0	0	0	DMA request by interrupt disabled	✓	✓	✓
0	0	0	0	0	1	INTP0	✓	✓	✓
0	0	0	0	1	0	INTP1	–	–	✓
0	0	0	0	1	1	INTP2	✓	✓	✓
0	0	0	1	0	0	INTP3	–	–	✓
0	0	0	1	0	1	INTP4	–	–	✓
0	0	0	1	1	0	INTP5	✓	✓	✓
0	0	0	1	1	1	INTP6	✓	✓	✓
0	0	1	0	0	0	INTP7	✓	✓	✓
0	0	1	0	0	1	INTTQ0OV	✓	✓	✓
0	0	1	0	1	0	INTTQ0CC0	✓	✓	✓
0	0	1	0	1	1	INTTQ0CC1	✓	✓	✓
0	0	1	1	0	0	INTTQ0CC2	✓	✓	✓
0	0	1	1	0	1	INTTQ0CC3	✓	✓	✓
0	0	1	1	1	0	INTTP0OV	✓	✓	✓
0	0	1	1	1	1	INTTP0CC0	✓	✓	✓
0	1	0	0	0	0	INTTP0CC1	✓	✓	✓
0	1	0	0	0	1	INTTP1OV	✓	✓	✓
0	1	0	0	1	0	INTTP1CC0	✓	✓	✓
0	1	0	0	1	1	INTTP1CC1	✓	✓	✓
0	1	0	1	0	0	INTTP2OV	✓	✓	✓
0	1	0	1	0	1	INTTP2CC0	✓	✓	✓
0	1	0	1	1	0	INTTP2CC1	✓	✓	✓
0	1	0	1	1	1	INTTP3CC0	✓	✓	✓
0	1	1	0	0	0	INTTP3CC1	✓	✓	✓
0	1	1	0	0	1	INTTP4CC0	✓	✓	✓
0	1	1	0	1	0	INTTP4CC1	✓	✓	✓
0	1	1	0	1	1	INTTP5CC0	✓	✓	✓
0	1	1	1	0	0	INTTP5CC1	✓	✓	✓
0	1	1	1	0	1	INTTM0EQ0	✓	✓	✓
0	1	1	1	1	0	INTCB0R/INTIIC1	✓	✓	✓
0	1	1	1	1	1	INTCB0T	✓	✓	✓
1	0	0	0	0	0	INTCB1R	–	✓	✓
1	0	0	0	0	1	INTCB1T	–	✓	✓

Remarks1. n = 0 to 3

2. JC3L (40): V850ES/JC3-L (40-pin products)

JC3L (48): V850ES/JC3-L (48-pin products)

JE3L: V850ES/JE3-L

Table 18-2. DMA Start Factors (2/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt Source	JC3L(40)	JC3L(48)	JE3L
1	0	0	0	1	0	INTCB2R	✓	✓	✓
1	0	0	0	1	1	INTCB2T	✓	✓	✓
1	0	0	1	0	0	INTCB3R	–	–	✓
1	0	0	1	0	1	INTCB3T	–	–	✓
1	0	0	1	1	0	INTUA0R/INTCB4R	–	✓	✓
						INTUA0R	✓	–	–
1	0	0	1	1	1	INTUA0T/INTCB4T	–	✓	✓
						INTUA0T	✓	–	–
1	0	1	0	0	0	INTUA1R/INTIIC2	✓	✓	✓
1	0	1	0	0	1	INTUA1T	✓	✓	✓
1	0	1	0	1	0	INTUA2R/INTIIC0	–	✓	✓
1	0	1	0	1	1	INTUA2T	–	✓	✓
1	0	1	1	0	0	INTAD	✓	✓	✓
1	0	1	1	0	1	INTKR	✓	✓	✓
1	0	1	1	1	0	INTRTC1	✓	✓	✓

Remarks1. n = 0 to 3

2. JC3L (40): V850ES/JC3-L (40-pin products)

JC3L (48): V850ES/JC3-L (48-pin products)

JE3L: V850ES/JE3-L

18.4 Transfer Sources and Destinations

Table 18-3 shows the relationship between the transfer sources and destinations (√: Transfer enabled, ×: Transfer disabled).

Table 18-3. Relationship Between Transfer Sources and Destinations

		Transfer Destination		
		Internal ROM	On-Chip Peripheral I/O	Internal RAM
Source	On-chip peripheral I/O	×	√	√
	Internal RAM	×	√	×
	Internal ROM	×	×	×

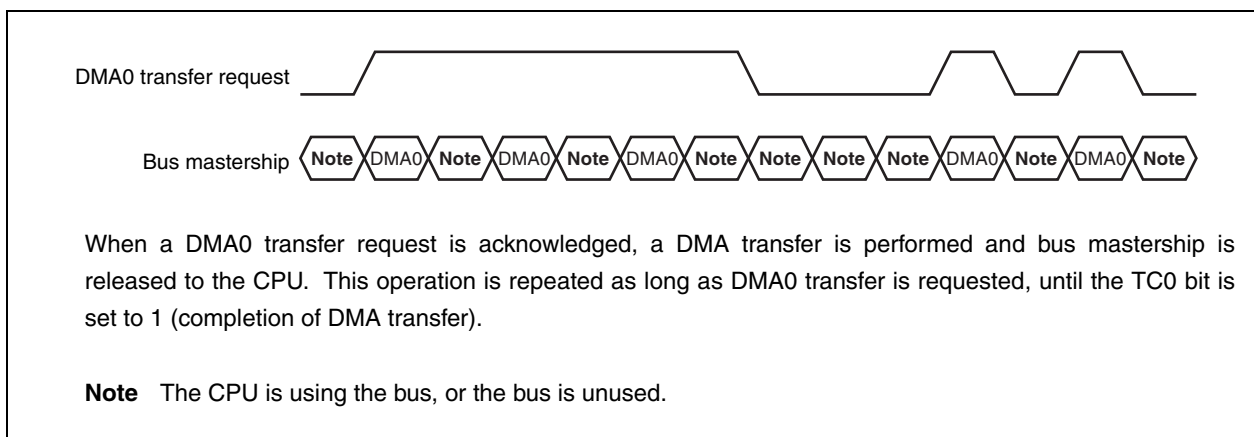
Caution The operation is not guaranteed for combinations of transfer destination and source marked with “×” in Table 18-3.

18.5 Transfer Modes

Single transfer is supported as the transfer mode.

In single transfer mode, the bus is released 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.

Figure 18-2. Single Transfer (Using Only One Channel)



18.6 Transfer Types

Two-cycle transfer is supported as the transfer type.

In two-cycle transfer, data is transferred in two cycles, a read cycle and a write cycle.

In the read cycle, the transfer source address is output and data is read from the source to the DMAC. In the write cycle, the transfer destination address is output and data is written from the DMAC to the destination.

An idle cycle of one clock is always inserted between a read cycle and a write cycle. If the data bus width differs between the transfer source and destination in two-cycle DMA transfer, the operation is performed as follows.

<16-bit data transfer (DADCn.DSn0 bit = 1)>

<1> Transfer from 32-bit bus to 16-bit bus

A read cycle (the higher or lower 16-bit data) is generated, followed by a write cycle (16 bits).

<2> Transfer from 16-/32-bit bus to 8-bit bus

A 16-bit read cycle is generated once, and then an 8-bit write cycle is generated twice.

<3> Transfer from 8-bit bus to 16-/32-bit bus

An 8-bit read cycle is generated twice, and then a 16-bit write cycle is generated once.

<4> Transfer from 16-bit bus to 32-bit bus

A 16-bit read cycle is generated once, and then a 16-bit write cycle is generated once.

For DMA transfer executed on an on-chip peripheral I/O register (transfer source/destination), be sure to specify the same transfer size as the register size. For example, to execute DMA transfer on an 8-bit register, be sure to specify byte (8-bit) transfer.

Remark The bus width of each transfer source and destination is as follows.

- On-chip peripheral I/O: 16 bits
- Internal RAM: 32 bits

18.7 DMA Channel Priorities

The DMA channel priorities are fixed as follows.

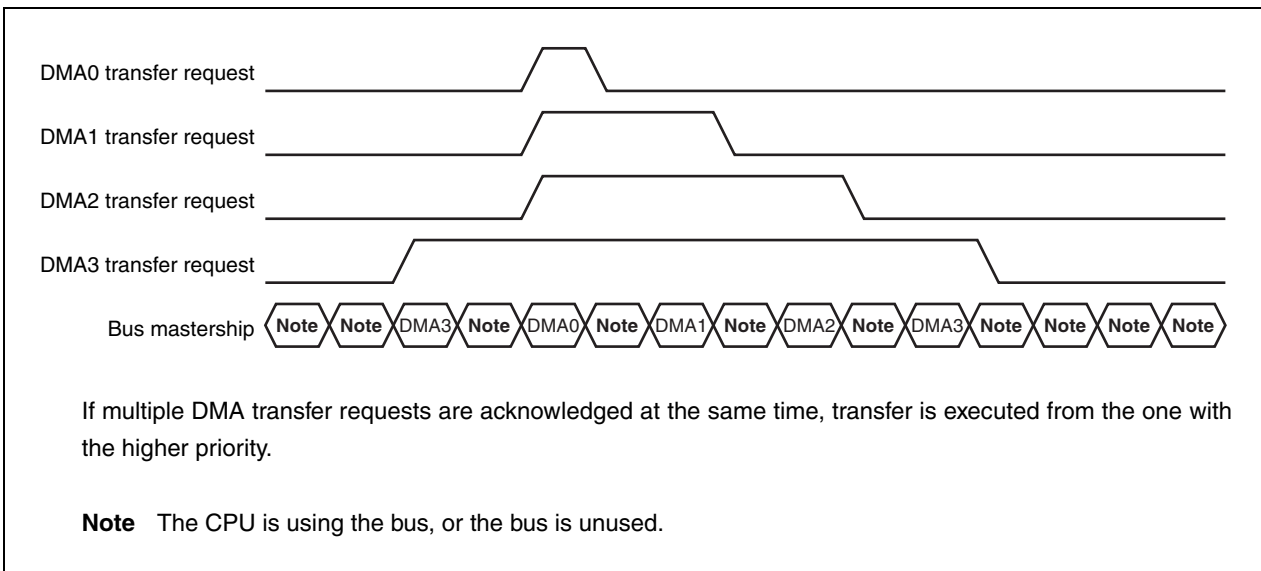
DMA channel 0 > DMA channel 1 > DMA channel 2 > DMA channel 3

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

If a new transfer request for the same channel and a transfer request for another channel with a lower priority are generated in a transfer cycle, DMA transfer on the channel with the lower priority is executed after the bus is released to the CPU (the new transfer request for the same channel is ignored in the transfer cycle).

The priorities are checked for every transfer cycle.

Figure 18-3. Single Transfer (Using Multiple Channels)



18.8 Time Related to DMA Transfer

The time required to respond to a DMA request, and the minimum number of clocks required for DMA transfer are shown below.

Single transfer: DMA response time (<1>) + Transfer source memory access (<2>) + 1^{Note 1} + Transfer destination memory access (<2>)

Table 18-4. Number of Execution Clocks During DMA Cycle

DMA Cycle		Number of Execution Clocks
<1> DMA request response time		4 clocks (MIN.) + Noise elimination time ^{Note 2}
<2> Memory access	Internal RAM access	2 clocks
	Peripheral I/O register access	3 clocks + Number of wait cycles specified by VSWC register ^{Note 3}

- Notes**
1. One clock is always inserted between a read cycle and a write cycle in DMA transfer.
 2. If an external interrupt (INTPn) is specified as the trigger to start DMA transfer, noise elimination time is added (n = 0 to 7).
 3. More wait cycles may be necessary for accessing a special register described in **3.4.9 (1)**.

18.9 DMA Transfer Start Factors

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

(1) Request by software

If the DCHCn.STGn bit is set to 1 while the DCHCn.TCn bit is 0 and DCHCn.Enn bit is 1 (DMA transfer enabled), DMA transfer starts.

To request the next DMA transfer cycle immediately after that, confirm, by using the DBCn register, that the preceding DMA transfer cycle has been completed, and set the STGn bit to 1 again (n = 0 to 3).

TCn bit = 0, Enn bit = 1

↓

STGn bit = 1 ... Starts the first DMA transfer.

↓

Confirm that the contents of the DBCn register have been updated.

STGn bit = 1 ... Starts the second DMA transfer.

↓

:

↓

Generation of terminal count ... Enn bit = 0, TCn bit = 1, and INTDMA_n signal is generated.

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

If an interrupt request is generated from the on-chip peripheral I/O set by the DTFRn register when the TCn bit is 0 and Enn bit is 1 (DMA transfer enabled), DMA transfer starts (n = 0 to 3).

Cautions 1. Two start factors (software trigger and hardware trigger) cannot be used for one DMA channel.

If two start factors are simultaneously generated for one DMA channel, only one of them is valid. However, the valid start factor cannot be identified.

- 2. A new transfer request generated for a DMA channel after the preceding DMA transfer request was generated and before the transfer is complete is ignored (cleared).**
- 3. The transfer request interval for the same DMA channel varies depending on the setting of bus waits in the DMA transfer cycle, the start status of the other channels. In particular, as described in Caution 2, a new transfer request generated for the same channel before a DMA transfer cycle starts or during a DMA transfer cycle is ignored. Therefore, the transfer request interval for the same DMA channel must be sufficiently secured by the system. When a software trigger is used, whether the preceding DMA transfer cycle has completed can be checked by reading the DBCn register.**

18.10 End of DMA Transfer

When DMA transfer has been completed the number of times set to the DBCn register and when the DCHCn.Enn bit is cleared to 0 and TCn bit is set to 1, a DMA transfer end interrupt request signal (INTDMA_n) is generated for the interrupt controller (INTC) (n = 0 to 3).

The V850ES/JC3-L and V850ES/JE3-L do not output a terminal count signal to an external device. Therefore, confirm completion of DMA transfer by using the DMA transfer end interrupt or polling the TCn bit.

18.11 Cautions

(1) VSWC register

When using the DMAC, be sure to specify an appropriate value for the VSWC register, in accordance with the operating frequency.

If an inappropriate value is specified for the VSWC register, the DMAC does not operate correctly (for details about the VSWC register, refer to **3.4.8 (1) (a) System wait control register (VSWC)**).

(2) DMA transfer executed for internal RAM

When a data access instruction located in the internal RAM is executed for a misaligned address, do not execute the instruction via DMA to transfer data to/from the internal RAM, because the CPU may not operate correctly afterward.

Similarly, when executing a DMA transfer to transfer data to/from the internal RAM, do not execute a data access instruction located in the internal RAM for a misaligned address.

(3) Reading DCHCn.TCn bit (n = 0 to 3)

The TCn bit is cleared to 0 when it is read, but not if read at a specific time. To definitely clear the TCn bit, add the following processing.

(a) When waiting for completion of DMA transfer by polling TCn bit

Confirm that the TCn bit has been set to 1 (after TCn bit = 1 is read), and then read the TCn bit three more times.

(b) When reading TCn bit in interrupt service routine

Read the TCn bit three times.

(4) DMA transfer initialization procedure (setting DCHCn.INITn bit to 1)

Even if the INITn bit is set to 1 when the channel executing DMA transfer is to be initialized, the channel may not be initialized. To definitely initialize the channel, execute either of the following two procedures.

(a) Temporarily stop transfer on all DMA channels

Initialize the channel executing DMA transfer using the procedure in <1> to <7> below.

Note, however, that TCn bit is cleared to 0 when step <5> is executed. Make sure that the other processing programs do not expect that the TCn bit is 1.

<1> Disable interrupts (DI).

<2> Read the DCHCn.Enn bit for DMA channels other than the one to be forcibly terminated, and transfer the value to a general-purpose register.

<3> Clear the Enn bit for the DMA channels used (including the channel to be forcibly terminated) to 0. To clear the Enn bit for the last DMA channel, execute the clear instruction twice. If the DMA transfer source or destination is the internal RAM, execute the instruction three times.

Example: Execute instructions in the following order if channels 0, 1, and 2 are used (if the internal RAM is not the transfer source or destination).

- Write 00H to DCHC0 (clear the E00 bit to 0).
- Write 00H to DCHC1 (clear the E11 bit to 0).
- Write 00H to DCHC2 (clear the E22 bit to 0).
- Write 00H to DCHC2 again (clear the E22 bit to 0).

<4> Write 04H to DCHCn corresponding to the channel to be forcibly terminated (set the INITn bit to 1).

<5> Read the TCn bit of each channel not to be forcibly terminated. If both the TCn bit and the Enn bit read in <2> are 1 (logical product (AND) is 1), clear the saved Enn bit to 0.

<6> After the operation in <5>, write the Enn bit value to the DCHCn register.

<7> Enable interrupts (EI).

Cautions 1. Be sure to execute step <5> above to prevent illegal setting of the Enn bit of the channels on which DMA transfer has been normally completed between <2> and <3>.

2. Clearing the Enn bit to 0 (<3>) and setting the INITn bit to 1 (<4>) by using a bit manipulation instruction clears the TCn bit, so a bit manipulation instruction must not be used.

(b) Repeatedly setting the INITn bit until transfer is forcibly terminated correctly

- <1> Before starting DMA, copy the initial number of transfers of the channel to be forcibly terminated to a general-purpose register.
- <2> Suppress a request from the DMA request source for the channel to be forcibly terminated (stop operation of the on-chip peripheral I/O).
- <3> Check that the DMA transfer request for the channel to be forcibly terminated is not held pending, by using the DTFRn.DFn bit. If a DMA transfer request is held pending, wait until execution of the pending DMA transfer request is completed.
- <4> When it has been confirmed that the DMA request for the channel to be forcibly terminated is not held pending, clear the Enn bit to 0.
- <5> Again, clear the Enn bit for the channel to be forcibly terminated to 0.
If the internal RAM is the transfer source or destination of the channel to be forcibly terminated, execute this operation again.
- <6> Set the INITn bit of the channel to be forcibly terminated to 1.
- <7> Read the value of the DBCn register corresponding to the channel to be forcibly terminated, and compare it with the value copied in <1>. If the two values do not match, repeat operations <6> and <7>.

- Remarks**
1. When the value of the DBCn register is read in <7>, the initial number of transfers is read if forced termination has been correctly completed. If not, the remaining number of transfers is read.
 2. Note that method (b) may take a long time if the application frequently uses DMA transfer for a channel other than the DMA channel to be forcibly terminated.

(5) Procedure for temporarily stopping DMA transfer (clearing Enn bit)

Stop and resume the DMA transfer under execution using the following procedure.

- <1> Suppress a transfer request from the DMA request source (stop operation of the on-chip peripheral I/O).
- <2> Check the DMA transfer request is not held pending, by using the DFn bit (check if the DFn bit = 0).
If a request is held pending, wait until execution of the pending DMA transfer request is completed.
- <3> Check the TCn bit to confirm that DMA transfer is not complete (confirm that the TCn bit is 0). If the TCn bit is 1, execute the DMA transfer completion processing.
- <4> If it has been confirmed that no DMA transfer request is held pending, clear the Enn bit to 0 (this operation suspends DMA transfer).
- <5> Set the Enn bit to 1 to resume DMA transfer.
- <6> Resume the operation of the DMA request source that has been stopped (start operation of the on-chip peripheral I/O).

(6) Memory boundary

The operation is not guaranteed if the address of the transfer source or destination exceeds the area of the DMA source or destination (internal RAM or on-chip peripheral I/O) during DMA transfer. (For details about the addresses of each area, see **Figure 3-2**.)

(7) Transferring misaligned data

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

If an odd address is specified as the transfer source or destination, the least significant bit of the address is forcibly handled as 0.

(8) Bus arbitration for CPU

Because the DMA controller is a higher priority bus master than the CPU, a CPU access that takes place during DMA transfer is held pending until the DMA transfer cycle is completed and the bus is released to the CPU.

However, the CPU can access the internal ROM and the internal RAM for which DMA transfer is not being executed.

- The CPU can access the internal ROM when DMA transfer is being executed between the on-chip peripheral I/O and the internal RAM.

(9) Registers/bits that must not be rewritten during DMA transfer

Set up the following registers during one of the periods below when a DMA transfer is not under execution (n = 0 to 3).

[Registers]

- DSAnH, DSAnL, DDAnH, DDAnL, DBCn, and DADCn registers
- DTFRn.IFCn5 to DTFRn.IFCn0 bits

[Timing of setting]

- Period from after reset to start of the first DMA transfer
- Period from after channel initialization to start of DMA transfer
- Period from after completion of DMA transfer (TCn bit = 1) to start of the next DMA transfer

(10) Be sure to set the following register bits to 0 (n = 0 to 3).

- Bits 14 to 10 of DSAnH register
- Bits 14 to 10 of DDAnH register
- Bits 15, 13 to 8, and 3 to 0 of DADCn register
- Bits 6 to 3 of DCHCn register

(11) DMA start factor

Do not start multiple DMA channels with the same start factor. If multiple channels are started with the same factor, DMA for which a channel has already been set may start or a DMA channel with a lower priority may be acknowledged before a DMA channel with a higher priority. The operation cannot be guaranteed in this case.

(12) Read values of DSAn and DDAn registers

If the DSAn and DDAn registers are read during a DMA transfer, the values before and after the registers were updated might be read.

For example, if the DSAnH register and then the DSAnL register are read when the DMA transfer source address (DSAn register) is 0000FFFFH and the count direction is incremental (DADCn.SAD1 and DADCn.SAD0 bits = 00), the value of the DSAnL register differs as follows, depending on whether DMA transfer is executed immediately after the DSAnH register is read.

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

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

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

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

(13) Setting up DMA transfer again

When re-specifying DMA settings by using the DDAnH, DDAnL, DSAnH, DSAnL, DBCn, and DADCn registers during the current DMA (the TCn bit is set to 1), be sure to initialize the DMA channels first. The DMA transfer must be initialized using the procedure described in **18.11 (4) DMA transfer initialization procedure**.

CHAPTER 19 INTERRUPT SERVICING/EXCEPTION PROCESSING FUNCTION

The V850ES/JC3-L and V850ES/JE3-L are provided with an interrupt controller dedicated to interrupt servicing (INTC) and can handle a total of 49/53/58 interrupt requests.

An interrupt is an event that occurs independently of program execution, and an exception is an event whose occurrence is dependent on program execution.

The V850ES/JC3-L and V850ES/JE3-L can handle interrupt request signals from the on-chip peripheral hardware and external sources. Moreover, exception processing can be started by a TRAP instruction (software exception) or by generation of an exception event (illegal execution of instructions) (exception trap).

19.1 Features

○ Interrupts

- Non-maskable interrupts and Maskable interrupts:

	Internal			external:		
	maskable	Non-maskable	total	maskable	Non-maskable	total
V850ES/JC3-L (40-pin)	1	42	43	1	5	6
V850ES/JC3-L (48-pin)	1	46	47	1	5	6
V850ES/JE3-L	1	48	49	1	8	9

- 8 levels of programmable priorities (maskable interrupts)
- Multiple interrupt control according to priority
- Masks can be specified for each maskable interrupt request.
- Noise elimination, edge detection, and valid edge specification for external interrupt request signals

○ Exceptions

- Software exceptions: 32 sources
- Exception trap: 2 sources (illegal opcode exception, debug trap)

The interrupt/exception sources are listed in Table 19-1.

Table 19-1. Interrupt Source List (1/3)

Type	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Interrupt Control Register
Reset	–	RESET	RESET pin input Reset input by internal source	RESET	0000H	00000000H	–
Non-maskable	–	NMI	NMI pin valid edge input	Pin	0010H	00000010H	–
	–	INTWDT2	WDT2 overflow	WDT2	0020H	00000020H	–
Software exception	–	TRAP0 ^{Note 2}	TRAP instruction	–	004nH ^{Note 2}	00000040H	–
	–	TRAP1 ^{Note 2}	TRAP instruction	–	005nH ^{Note 2}	00000050H	–
Exception trap	–	ILGOP/ DBG0	Illegal opcode/ DBTRAP instruction	–	0060H	00000060H	–
Maskable	0	INTLVI	Low voltage detection	POCLVI	0080H	00000080H	LVIIC
	1	INTP0	External interrupt pin input edge detection (INTP0)	Pin	0090H	00000090H	PIC0
	2	INTP1 ^{Note 3}	External interrupt pin input edge detection (INTP1)	Pin	00A0H	000000A0H	PIC1
	3	INTP2	External interrupt pin input edge detection (INTP2)	Pin	00B0H	000000B0H	PIC2
	4	INTP3 ^{Note 3}	External interrupt pin input edge detection (INTP3)	Pin	00C0H	000000C0H	PIC3
	5	INTP4 ^{Note 3}	External interrupt pin input edge detection (INTP4)	Pin	00D0H	000000D0H	PIC4
	6	INTP5	External interrupt pin input edge detection (INTP5)	Pin	00E0H	000000E0H	PIC5
	7	INTP6	External interrupt pin input edge detection (INTP6)	Pin	00F0H	000000F0H	PIC6
	8	INTP7	External interrupt pin input edge detection (INTP7)	Pin	0100H	00000100H	PIC7
	9	INTTQ0OV	TMQ0 overflow	TMQ0	0110H	00000110H	TQ0OVIC
	10	INTTQ0CC0	TMQ0 capture 0/compare 0 match	TMQ0	0120H	00000120H	TQ0CCIC0
	11	INTTQ0CC1	TMQ0 capture 1/compare 1 match	TMQ0	0130H	00000130H	TQ0CCIC1
	12	INTTQ0CC2	TMQ0 capture 2/compare 2 match	TMQ0	0140H	00000140H	TQ0CCIC2
	13	INTTQ0CC3	TMQ0 capture 3/compare 3 match	TMQ0	0150H	00000150H	TQ0CCIC3
	14	INTTP0OV	TMP0 overflow	TMP0	0160H	00000160H	TP0OVIC
	15	INTTP0CC0	TMP0 capture 0/compare 0 match	TMP0	0170H	00000170H	TP0CCIC0
	16	INTTP0CC1	TMP0 capture 1/compare 1 match	TMP0	0180H	00000180H	TP0CCIC1
	17	INTTP1OV	TMP1 overflow	TMP1	0190H	00000190H	TP1OVIC
	18	INTTP1CC0	TMP1 capture 0/compare 0 match	TMP1	01A0H	000001A0H	TP1CCIC0
	19	INTTP1CC1	TMP1 capture 1/compare 1 match	TMP1	01B0H	000001B0H	TP1CCIC1
	20	INTTP2OV	TMP2 overflow	TMP2	01C0H	000001C0H	TP2OVIC
	21	INTTP2CC0	TMP2 capture 0/compare 0 match	TMP2	01D0H	000001D0H	TP2CCIC0
22	INTTP2CC1	TMP2 capture 1/compare 1 match	TMP2	01E0H	000001E0H	TP2CCIC1	

Notes 1. The software that generated the exception event can be checked using the exception code set to the EICC bit of the ECR register.

2. n = 0 to FH

3. V850ES/JE3-L only

Table 19-1. Interrupt Source List (2/3)

Type	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Interrupt Control Register
Maskable	23	INTTP3OV	TMP3 overflow	TMP3	01F0H	000001F0H	TP3OVIC
	24	INTTP3CC0	TMP3 capture 0/compare 0 match	TMP3	0200H	00000200H	TP3/CCIC0
	25	INTTP3CC1	TMP3 capture 1/compare 1 match	TMP3	0210H	00000210H	TP3CCIC1
	26	INTTP4OV	TMP4 overflow	TMP4	0220H	00000220H	TP4OVIC
	27	INTTP4CC0	TMP4 capture 0/compare 0 match	TMP4	0230H	00000230H	TP4CCIC0
	28	INTTP4CC1	TMP4 capture 1/compare 1 match	TMP4	0240H	00000240H	TP4CCIC1
	29	INTTP5OV	TMP5 overflow	TMP5	0250H	00000250H	TP5OVIC
	30	INTTP5CC0	TMP5 capture 0/compare 0 match	TMP5	0260H	00000260H	TP5CCIC0
	31	INTTP5CC1	TMP5 capture 1/compare 1 match	TMP5	0270H	00000270H	TP5CCIC1
	32	INTTM0EQ0	TMM0 compare match	TMM0	0280H	00000280H	TM0EQIC0
	33	INTCB0R/ INTIIC1	CSIB0 reception completion/ CSIB0 reception error/ IIC1 transfer completion	CSIB0/ IIC1	0290H	00000290H	CB0RIC/ IIC1C1
	34	INTCB0T	CSIB0 successive transmission write enable	CSIB0	02A0H	000002A0H	CB0TIC
	35	INTCB1R ^{Note 1}	CSIB1 reception completion/ CSIB1 reception error	CSIB1	02B0H	000002B0H	CB1RIC
	36	INTCB1T ^{Note 1}	CSIB1 successive transmission write enable	CSIB1	02C0H	000002C0H	CB1TIC
	37	INTCB2R	CSIB2 reception completion/ CSIB2 reception error	CSIB2	02D0H	000002D0H	CB2RIC
	38	INTCB2T	CSIB2 successive transmission write enable	CSIB2	02E0H	000002E0H	CB2TIC
	39	INTCB3R ^{Note 2}	CSIB3 reception completion/ CSIB3 reception error	CSIB3	02F0H	000002F0H	CB3RIC
	40	INTCB3T ^{Note 2}	CSIB3 successive transmission write enable	CSIB3	0300H	00000300H	CB3TIC
	41	INTUA0R/ INTCB4R ^{Note 1}	UARTA0 reception completion/ UARTA0 reception error/ CSIB4 reception completion/ CSIB4 reception error	UARTA0/ CSIB4	0310H	00000310H	UA0RIC/ CB4RIC
	42	INTUA0T/ INTCB4T ^{Note 1}	UARTA0 successive transmission enable/CSIB4 successive transmission write enable	UARTA0/ CSIB4	0320H	00000320H	UA0TIC/ CB4TIC
	43	INTUA1R/ INTIIC2	UARTA1 reception completion/ UARTA1 reception error/ IIC2 transfer completion	UARTA1/ IIC2	0330H	00000330H	UA1RIC/ IIC2C2
	44	INTUA1T	UARTA1 successive transmission enable	UARTA1	0340H	00000340H	UA1TIC
	45	INTUA2R ^{Note 1} / INTIIC0 ^{Note 1}	UARTA2 reception completion/ UARTA2 reception error/ IIC0 transfer completion	UARTA2/ IIC0	0350H	00000350H	UA2RIC/ IIC0C0

Notes 1. V850ES/JC3-L (48-pin), V850ES/JE3-L only

2. V850ES/JE3-L only

Table 19-1. Interrupt Source List (3/3)

Type	Default Priority	Name	Trigger	Generating Unit	Exception Code	Handler Address	Interrupt Control Register
Maskable	46	INTUA2T ^{Note}	UARTA2 successive transmission enable	UARTA2	0360H	00000360H	UA2TIC
	47	INTAD	A/D conversion completion	A/D	0370H	00000370H	ADIC
	48	INTDMA0	DMA0 transfer completion	DMA	0380H	00000380H	DMAIC0
	49	INTDMA1	DMA1 transfer completion	DMA	0390H	00000390H	DMAIC1
	50	INTDMA2	DMA2 transfer completion	DMA	03A0H	000003A0H	DMAIC2
	51	INTDMA3	DMA3 transfer completion	DMA	03B0H	000003B0H	DMAIC3
	52	INTKR	Key return interrupt	KR	03C0H	000003C0H	KRIC
	53	INTWT1 /INTRTC2	Watch timer interval/ RTC interval signal	WT/ RTC	03D0H	000003D0H	WTIIC/ RTC2IC
	54	INTWT /INTRTC0	Watch timer reference time/ RTC constant cycle signal	WT/ RTC	03E0H	000003E0H	WTIC/ RTC0IC
	55	INTRTC1	RTC alarm match	RTC	03F0H	000003F0H	RTC1IC

Note V850ES/JC3-L (48-pin), V850ES/JE3-L only

- Remarks 1.** Default Priority: The priority order that is applied when multiple maskable interrupt requests having the same priority level occur simultaneously. Smaller numbers have a higher priority, with 0 given the highest priority. The priority order of non-maskable interrupts is INTWDT2 > NMI.
- 2.** The execution address of the illegal instruction when an illegal opcode exception occurs is calculated by (Restored PC – 4).

19.2 Non-Maskable Interrupts

A non-maskable interrupt request signal is acknowledged even when interrupts are disabled (DI) by the CPU. A non-maskable interrupt is not subject to priority control and takes precedence over all the other interrupt request signals.

This product has the following two non-maskable interrupt request signals.

- NMI pin input (NMI)
- Non-maskable interrupt request signal generated by overflow of watchdog timer (INTWDT2)

The valid edge of the NMI pin can be selected from four types: "rising edge", "falling edge", "both edges", and "no edge detection". "No edge detection" is selected by default. Be sure to specify the valid edge.

The non-maskable interrupt request signal generated by overflow of watchdog timer 2 (INTWDT2) functions when the WDTM2.WDM21 and WDTM2.WDM20 bits are set to "01".

If two or more non-maskable interrupt request signals occur at the same time, the interrupt with the higher priority is serviced, as follows (the interrupt request signal with the lower priority is ignored).

INTWDT2 > NMI

If a new NMI or INTWDT2 request signal is issued while a non-maskable interrupt is being serviced, it is serviced as follows.

(1) If new NMI request signal is issued while non-maskable interrupt is being serviced

The new NMI request signal is held pending, regardless of the value of the PSW.NP bit. The pending NMI request signal is acknowledged after the non-maskable interrupt currently under execution has been serviced (after the RETI instruction has been executed).

(2) If new INTWDT2 request signal is issued while non-maskable interrupt is being serviced

If the NP bit is set (1) while a non-maskable interrupt is being serviced, the new INTWDT2 request signal is held pending. The pending INTWDT2 request signal is acknowledged after the non-maskable interrupt currently under execution has been serviced (after the RETI instruction has been executed).

If the NP bit is cleared (0) while a non-maskable interrupt is being serviced, the newly generated INTWDT2 request signal is acknowledged (the current non-maskable interrupt servicing is stopped).

Caution For details about the non-maskable interrupt servicing requested by the INTWDT2 signal, see 19.2.2 (2) From INTWDT2 signal.

Figure 19-1. Non-Maskable Interrupt Request Signal Acknowledgment Operation (1/2)

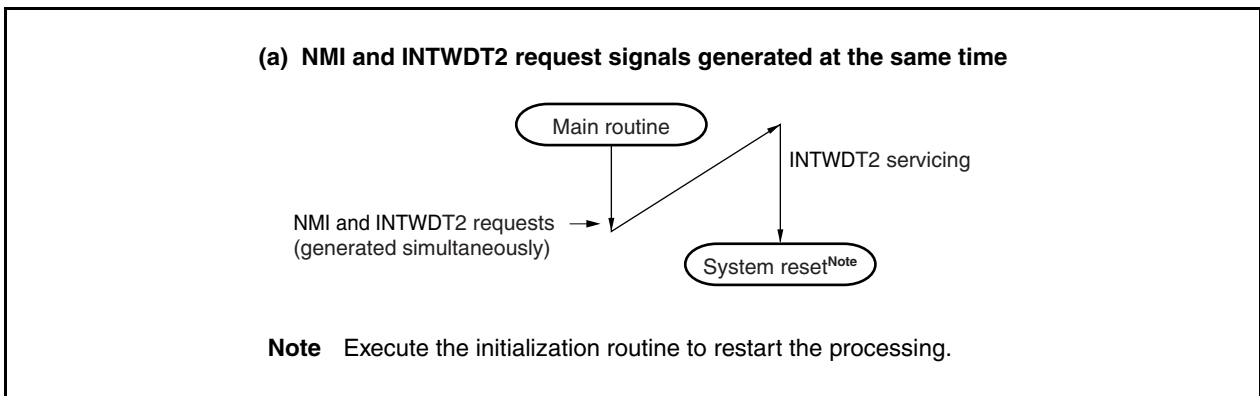
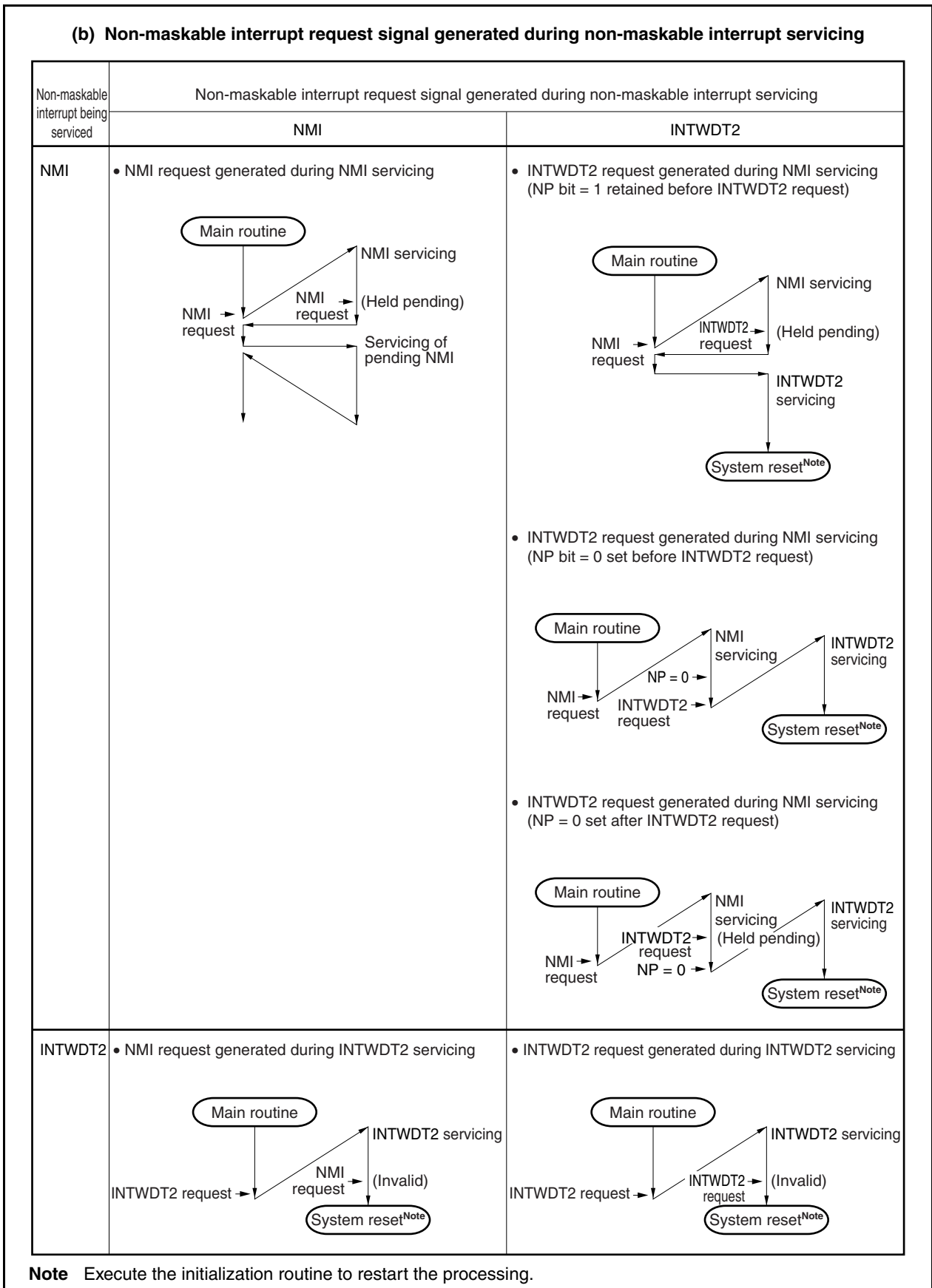


Figure 19-1. Non-Maskable Interrupt Request Signal Acknowledgment Operation (2/2)



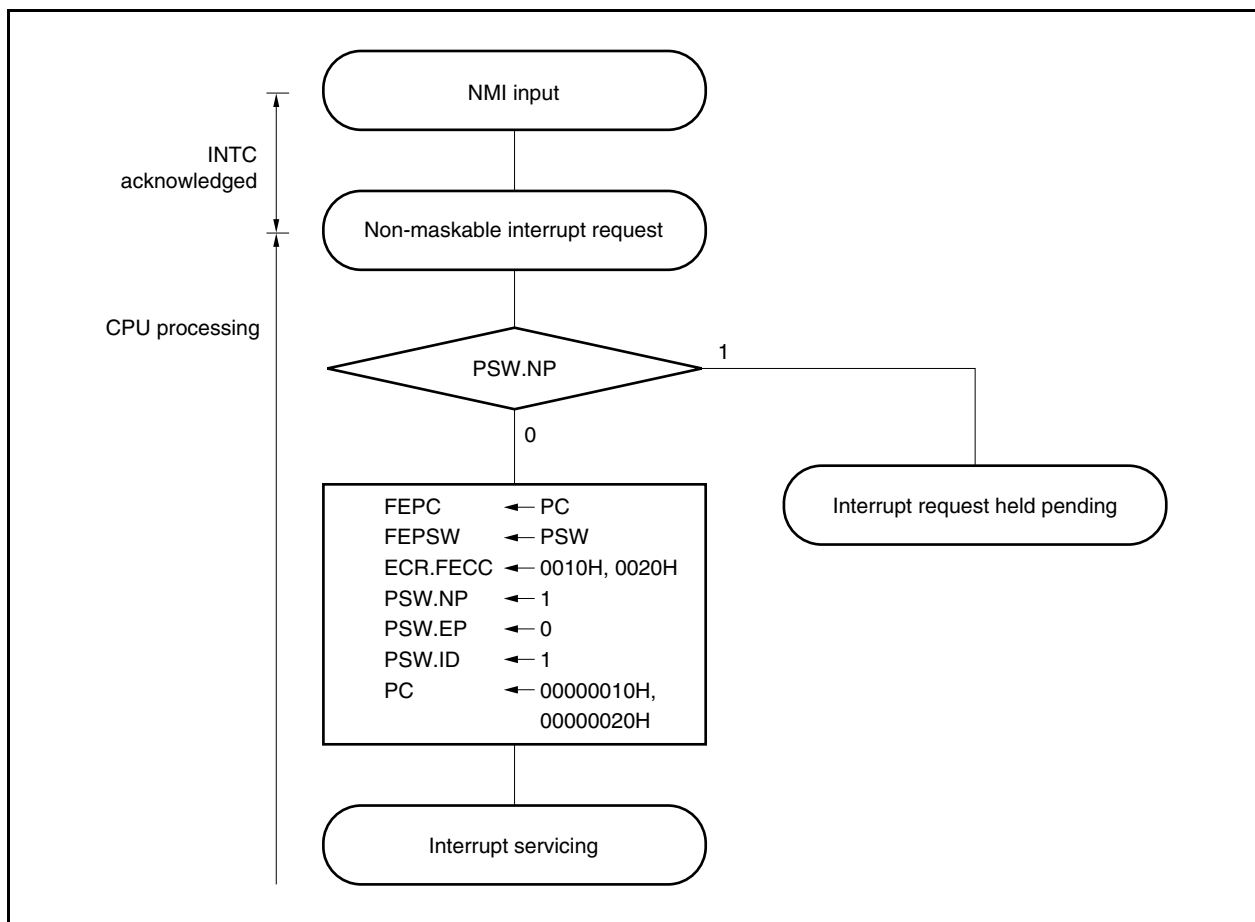
19.2.1 Operation

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 current PC to FEPC.
- <2> Saves the current PSW to FEPSW.
- <3> Writes 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> Sets the handler address (00000010H, 00000020H) corresponding to the non-maskable interrupt to the PC, and transfers control.

The servicing of a non-maskable interrupt is shown below.

Figure 19-2. Non-Maskable Interrupt Servicing



19.2.2 Restoration

(1) From NMI pin input

Execution is returned from NMI servicing by using the RETI instruction.

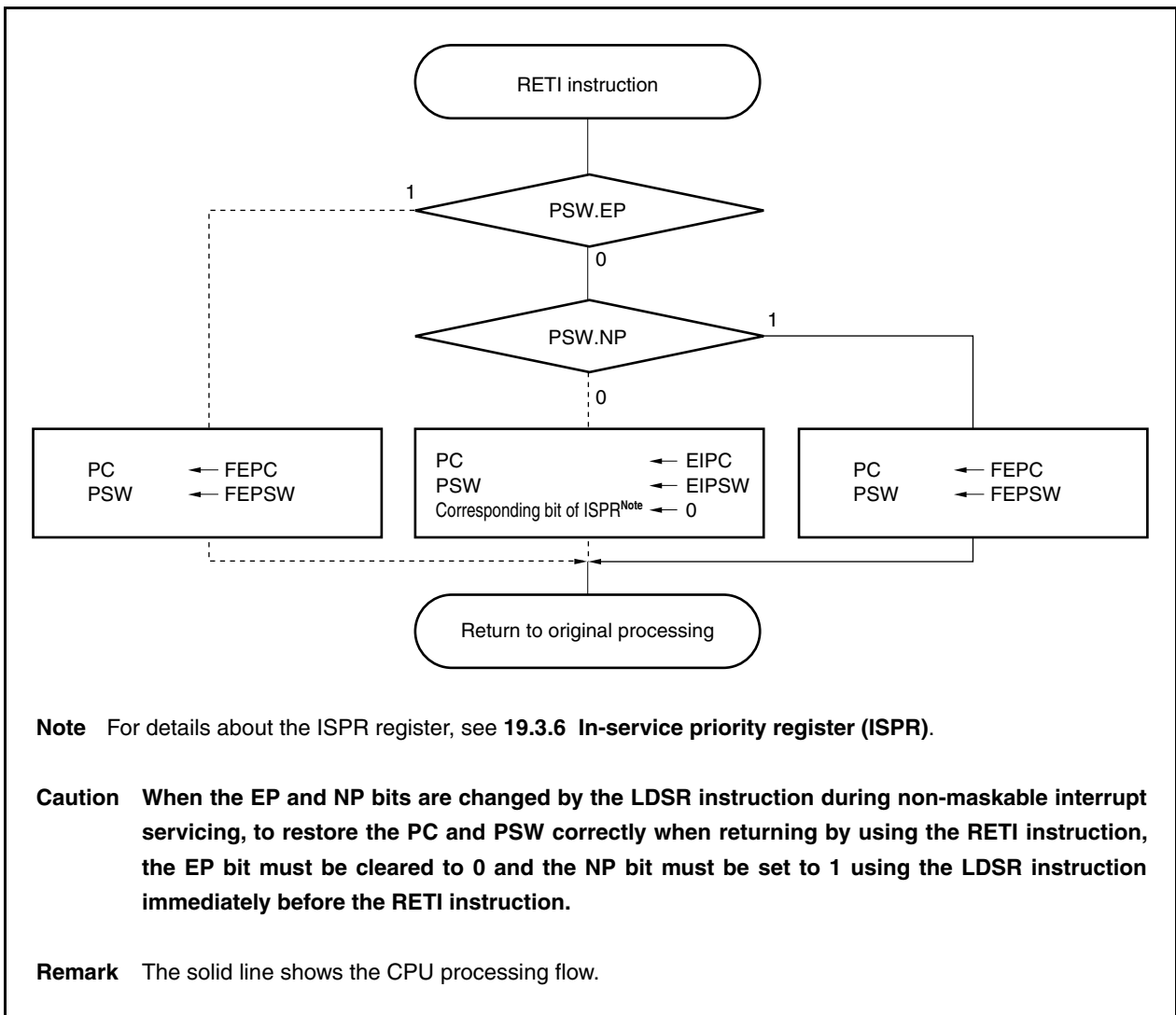
When the RETI instruction is executed, the CPU performs the following processing and transfers control to the address of the restored PC.

<1> Loads the saved PC and PSW from FEPC and FEPSW, respectively, because the PSW.EP bit is 0 and the PSW.NP bit is 1.

<2> Transfers control back to the address of the restored PC and PSW.

The processing of the RETI instruction is shown below.

Figure 19-3. RETI Instruction Processing

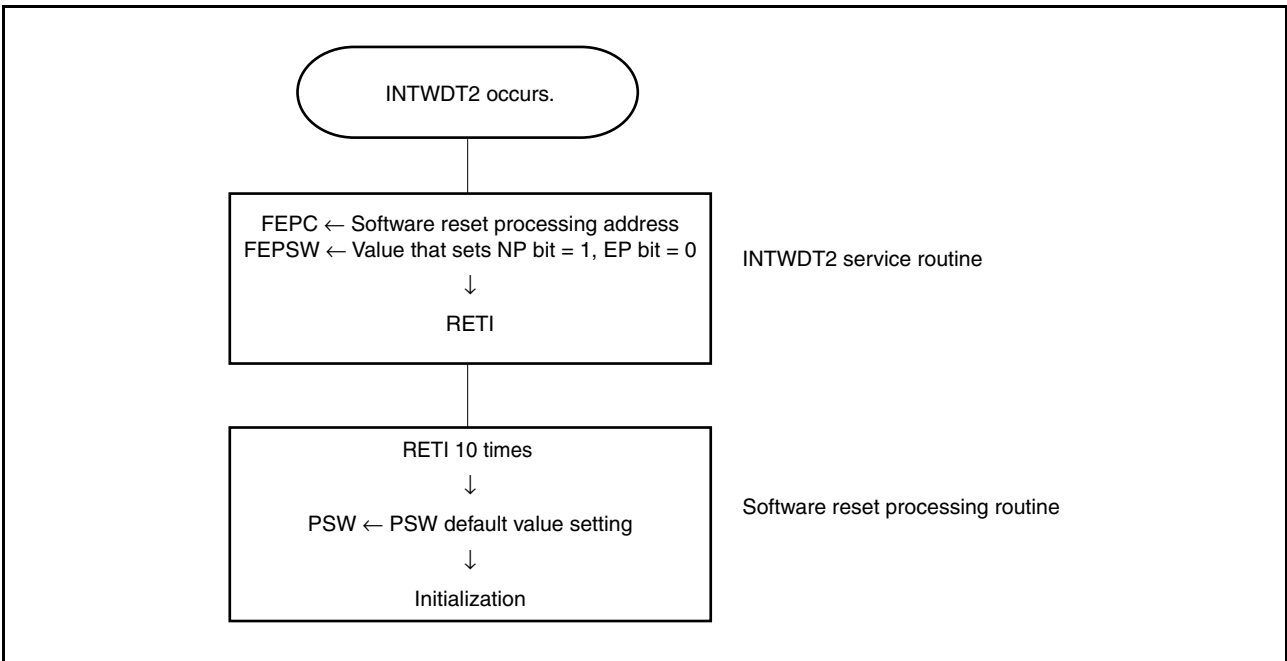


(2) From INTWDT2 signal

Non-maskable interrupt servicing executed by INTWDT2 cannot be returned from by using the RETI instruction. To return from such servicing, execute the following software reset processing.

In the software reset processing, however, the registers that can be set up only once immediately after a reset ends (such as the WDTM2 register) cannot be set up again. To reset these registers to their initial statuses, a hardware reset such as reset pin input is required.

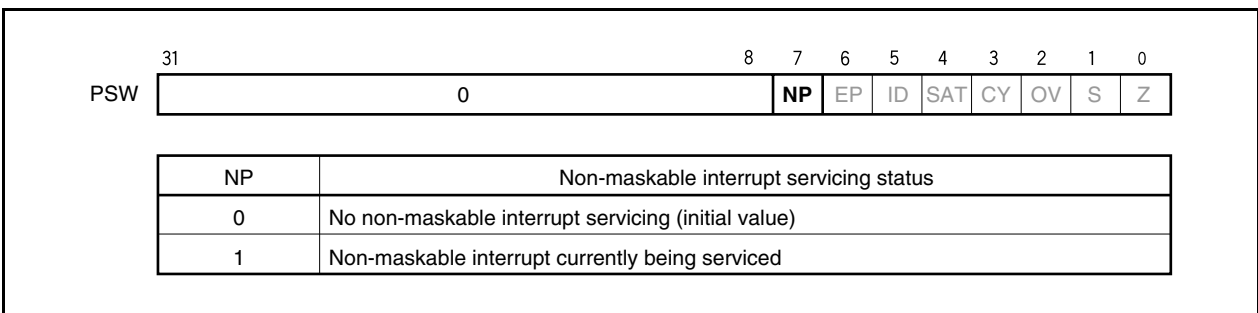
Figure 19-4. Software Reset Processing



19.2.3 NP flag

The NP flag is a status flag that indicates that a non-maskable interrupt is being serviced.

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.



19.3 Maskable Interrupts

Maskable interrupt request signals can be masked by interrupt control registers. The V850ES/JC3-L and V850ES/JE3-L have 47/51/56 maskable interrupt sources.

When an interrupt request signal has been acknowledged, interrupts are disabled (DI) and subsequent maskable interrupt request signals are not acknowledged.

When the EI instruction is executed in an interrupt service routine, interrupts are enabled (EI), which enables acknowledgment of interrupt request signals having a priority higher than that of the interrupt request signal currently being serviced. Interrupt request signals with the same priority level cannot be nested.

For details about multiple interrupts, see **19.6 Multiple Interrupt Servicing Control**.

19.3.1 Operation

If a maskable interrupt request signal is generated, the CPU performs the following processing and transfers control to the handler routine.

- <1> Saves the current 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> Loads the corresponding handler address to the PC and transfers control.

A maskable interrupt request signal masked by the interrupt controller (INTC)(xxMK bit = 1) and a maskable interrupt request signal generated while another interrupt is being serviced (while the PSW.NP bit is 1 or the PSW.ID bit is 1) are held pending in the INTC. The cause of being held pending and the workaround are described below.

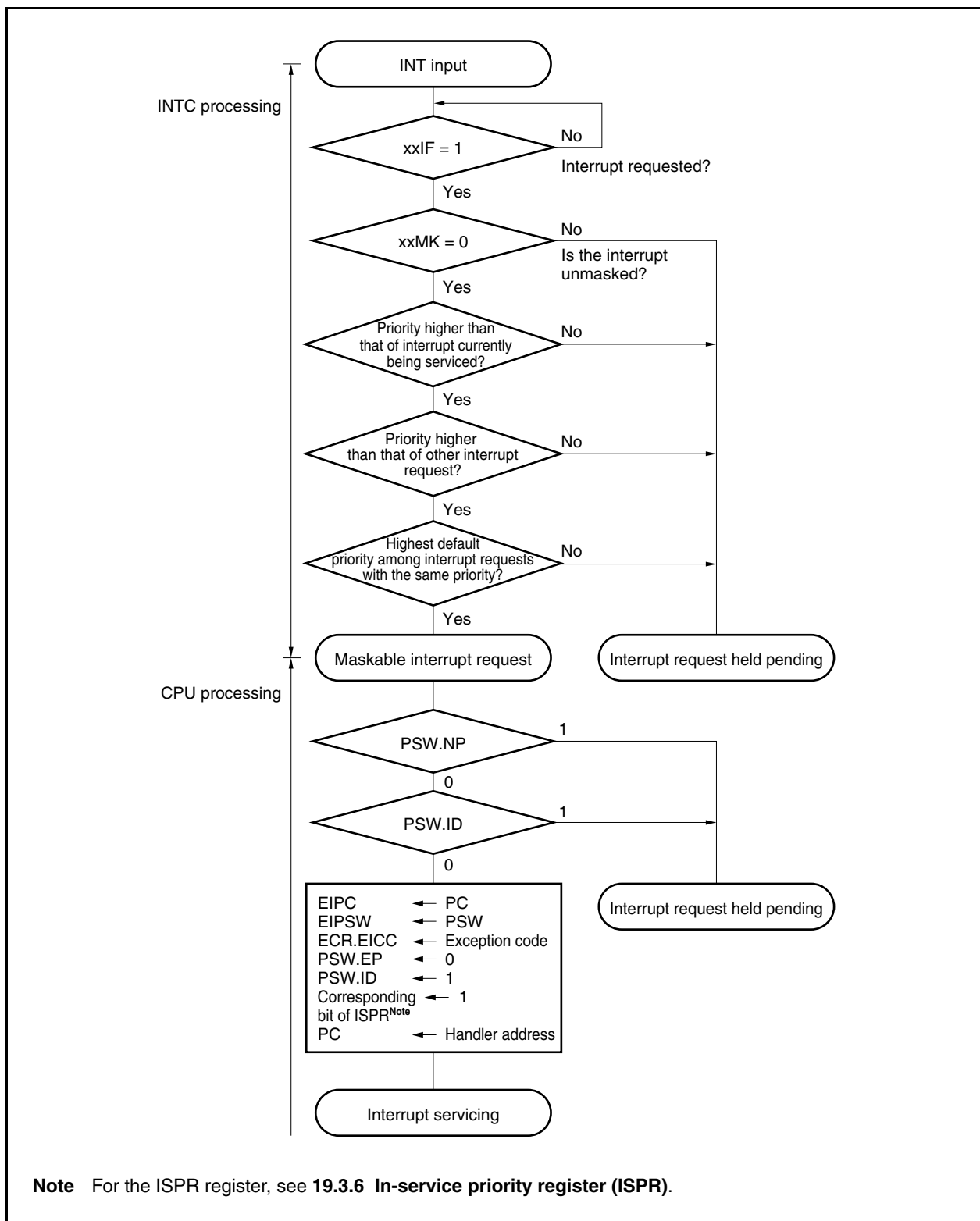
Table 19-2. Maskable Interrupts Held Pending

Cause	Workaround
xxMK bit = 1	Unmask the signal (clear xxMK bit to 0).
Another interrupt having higher priority is being held pending	Wait for the servicing of the interrupt to end.
PSW.NP bit = 1 and PSW.ID bit = 1	Set the NP bit to 0 and the ID bit to 1 by using the RETI and LDSR instructions.

Remark For details about the xxMK bit, see **19.3.4 Interrupt control register (xxICn)**.

Figure 19-5 shows the servicing of maskable interrupts.

Figure 19-5. Maskable Interrupt Servicing



19.3.2 Restoration

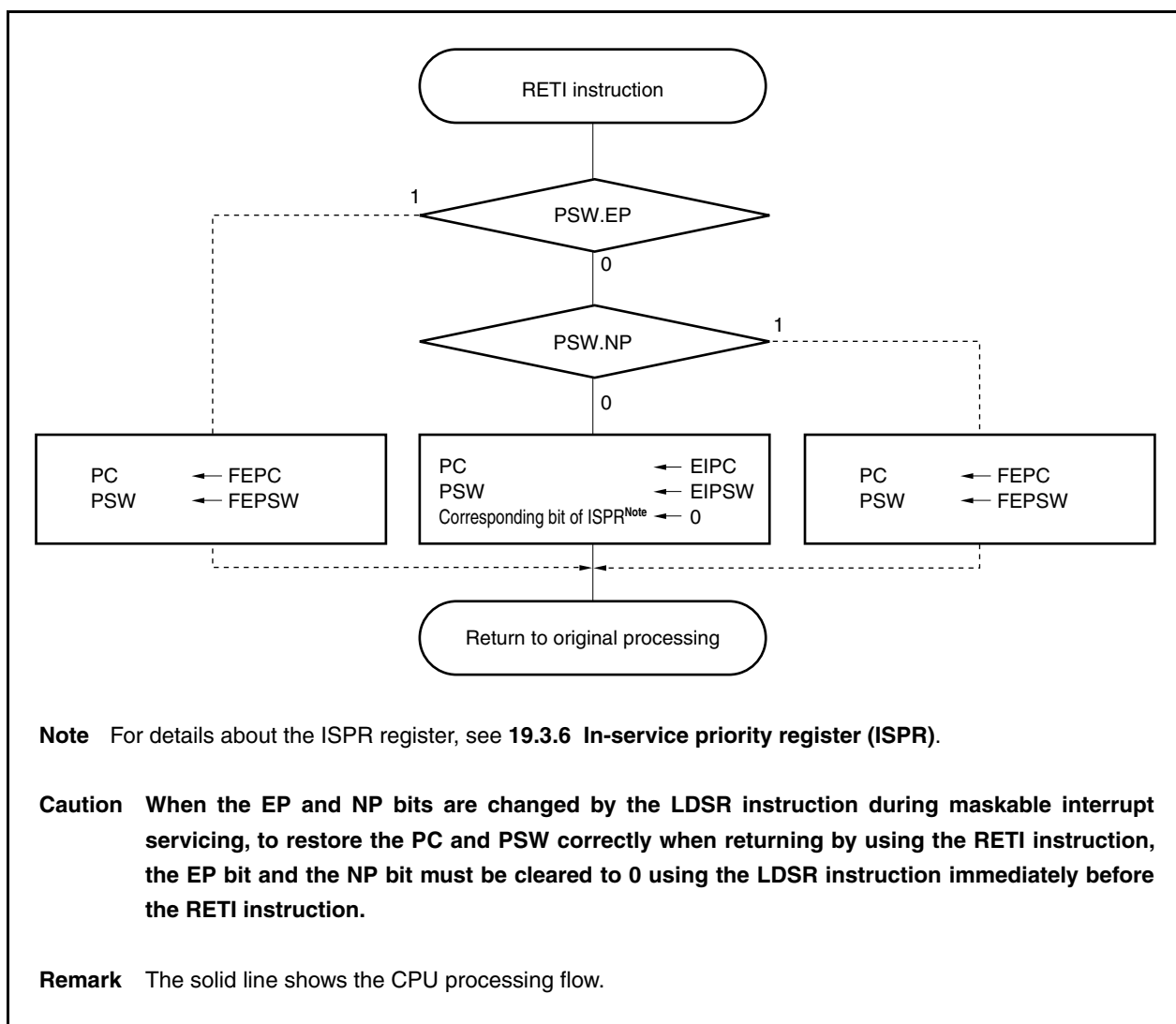
Execution is returned from maskable interrupt servicing by using the RETI instruction.

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

- <1> Loads the saved PC and PSW from EIPC and EIPSW, respectively, because the PSW.EP bit is 0 and the PSW.NP bit is 0.
- <2> Transfers control back to the address of the restored PC and PSW.

The processing of the RETI instruction is shown below.

Figure 19-6. RETI Instruction Processing



19.3.3 Priorities of maskable interrupts

The INTC can acknowledge an interrupt while servicing another. Interrupts that occur at the same time are serviced according to their priority order.

There are two types of priority level control: control based on the programmable priority levels that are specified by the interrupt priority level specification bit (xxPRn) of the interrupt control register (xxICn), and control based on the default priority levels. Programmable priority control classifies interrupt request signals into eight levels according to the setting of the xxPRn flag. When multiple interrupts having the same priority level specified by the xxPRn bit occur at the same time, the interrupts are serviced according to the priority levels assigned to the corresponding interrupt requests (default priority level) beforehand. For details, see **Table 19-1 Interrupt Source List**.

For details about multiple interrupts, see **19.7 Multiple Interrupt Servicing Control**.

Remark xx: Identification name of each peripheral unit (see **Table 19-3 Interrupt Control Registers (xxICn)**)
n: Peripheral unit number (see **Table 19-3 Interrupt Control Registers (xxICn)**).

Figure 19-7. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (1/2)

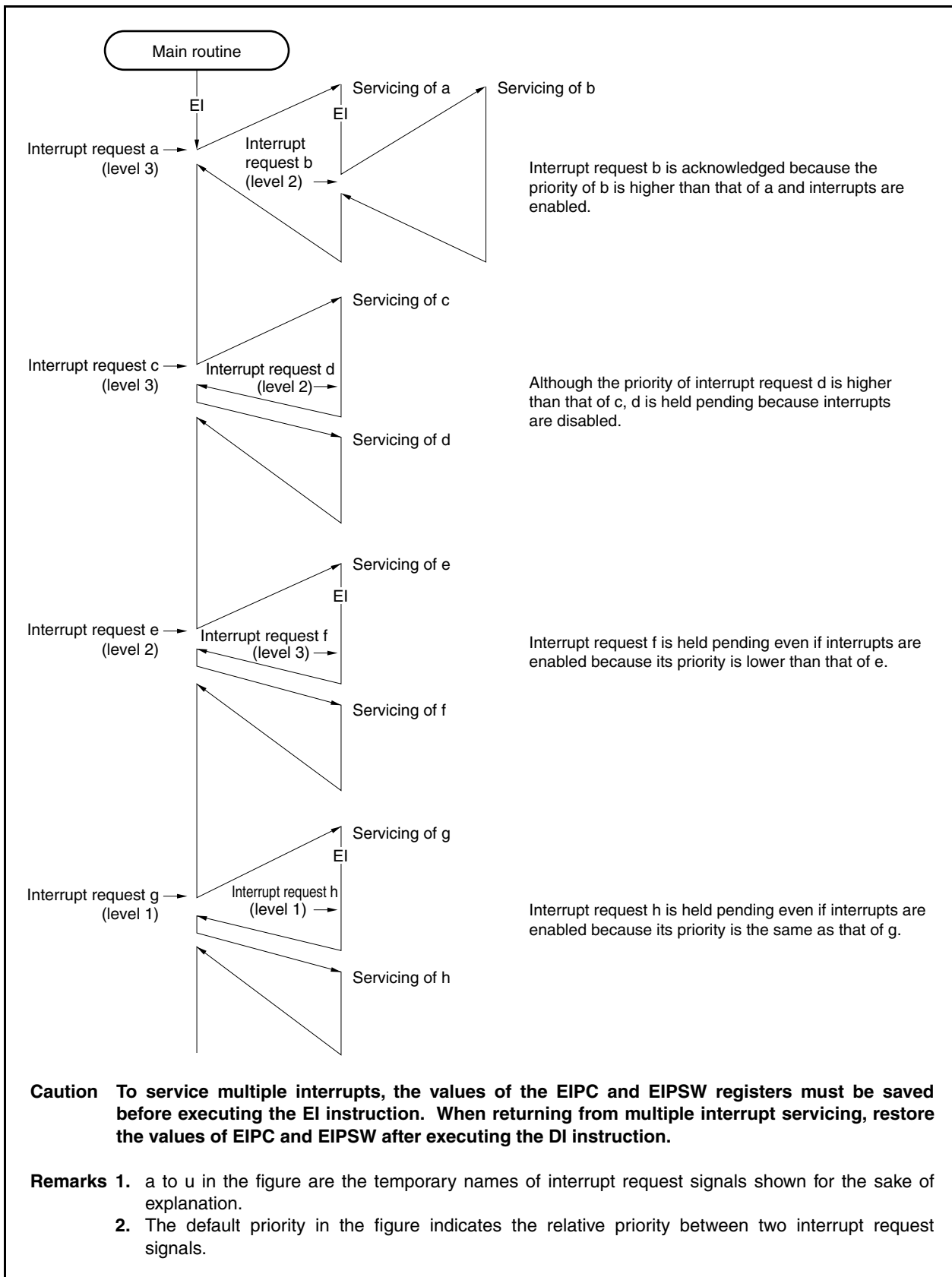


Figure 19-7. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (2/2)

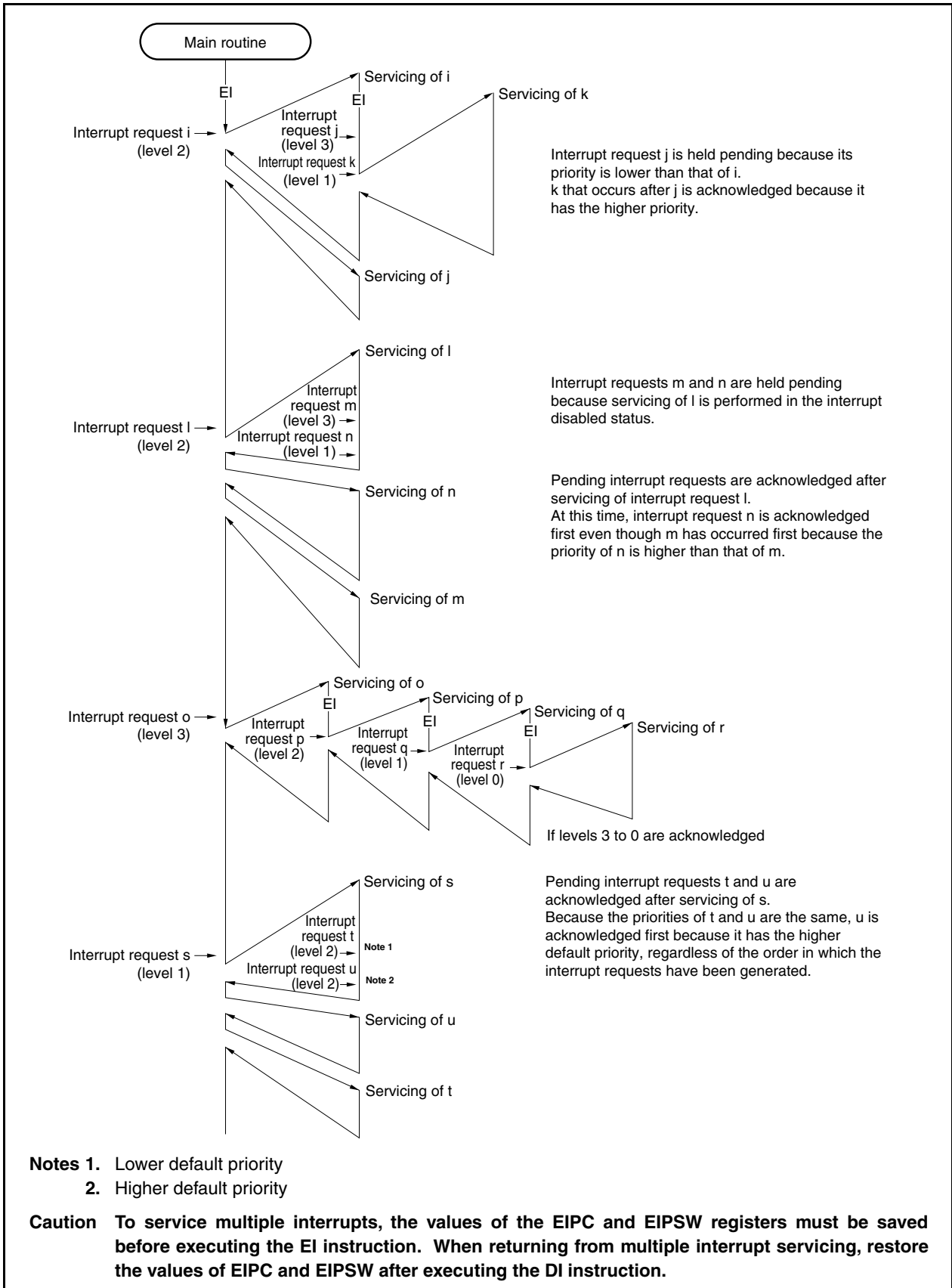
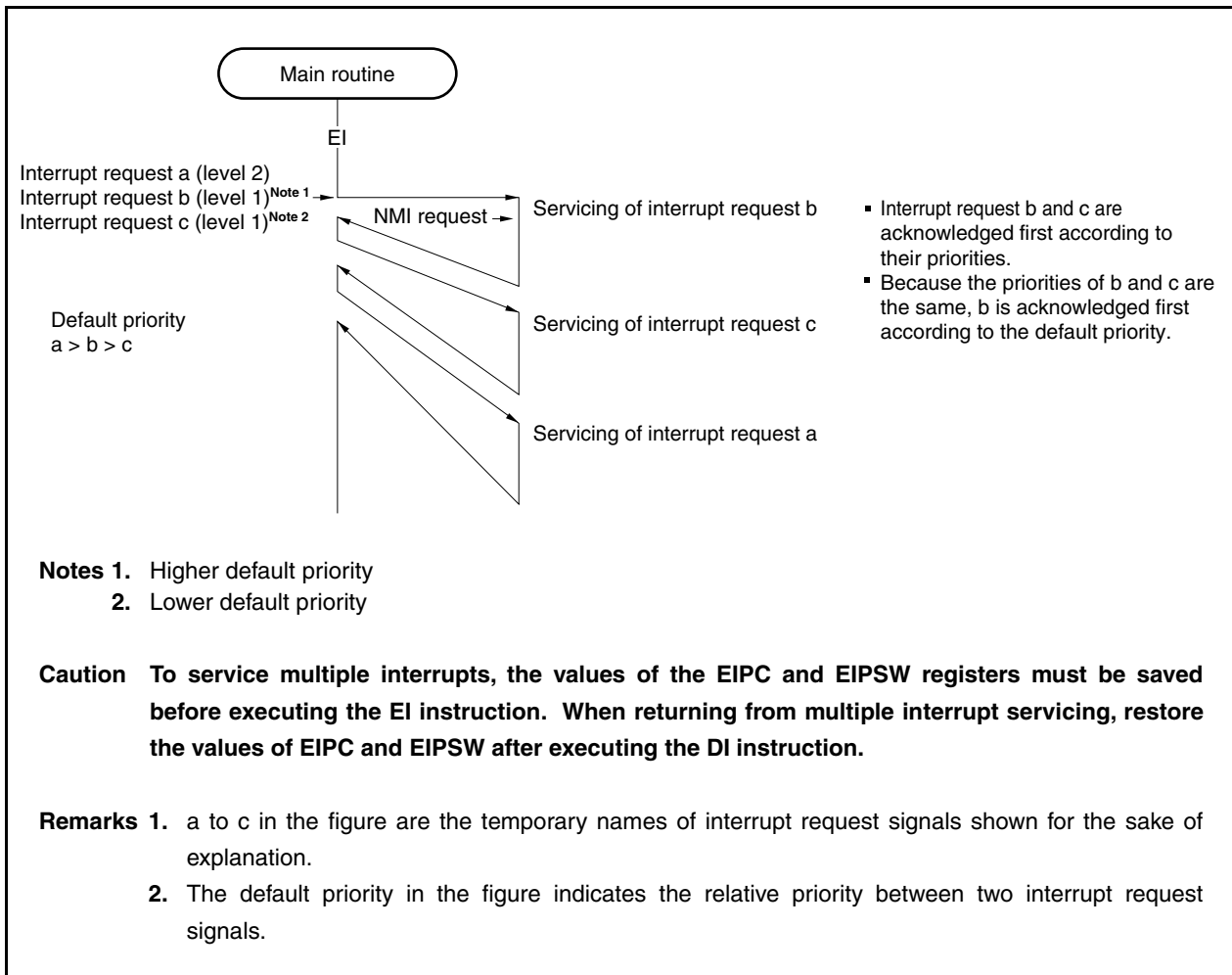


Figure 19-8. Example of Servicing Interrupt Requests Generated Simultaneously



19.3.4 Interrupt control register (xxICn)

An xxICn register is assigned to each interrupt request signal (maskable interrupt) and sets the control conditions for each maskable interrupt request.

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

Reset sets this register to 47H.

- Cautions 1. To mask interrupts, set up the IMR register or use a bit manipulation instruction. The priority levels must be specified at a time when no interrupt will occur.**
- 2. Disable interrupts (DI) before reading the xxICn.xxIFn bit. If the xxIFn bit is read while interrupts are enabled (EI), the correct value may not be read if acknowledging an interrupt and reading the bit conflict.**

After reset: 47H R/W Address: FFFFF110H to FFFFF17CH

	<7>	<6>	5	4	3	2	1	0
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0

xxIFn	Interrupt request flag^{Note}
0	Interrupt request not issued
1	Interrupt request issued

xxMKn	Interrupt mask flag
0	Interrupt servicing enabled
1	Interrupt servicing disabled (pending)

xxPRn2	xxPRn1	xxPRn0	Interrupt priority specification bit
0	0	0	Specifies level 0 (highest).
0	0	1	Specifies level 1.
0	1	0	Specifies level 2.
0	1	1	Specifies level 3.
1	0	0	Specifies level 4.
1	0	1	Specifies level 5.
1	1	0	Specifies level 6.
1	1	1	Specifies level 7 (lowest).

Note The flag xxIFn is reset automatically by the hardware if an interrupt request signal is acknowledged.

Remark xx: Identification name of each peripheral unit (see **Table 19-3 Interrupt Control Registers (xxICn)**)
n: Peripheral unit number (see **Table 19-3 Interrupt Control Registers (xxICn)**).

The addresses and bits of the interrupt control registers are as follows.

Table 19-3. Interrupt Control Registers (xxICn) (1/2)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF110H	LVIIC	LVIIIF	LVIMK	0	0	0	LVIPR2	LVIPR1	LVIPR0
FFFFF112H	PIC0	PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00
FFFFF114H	PIC1 ^{Note1}	PIF1	PMK1	0	0	0	PPR12	PPR11	PPR10
FFFFF116H	PIC2	PIF2	PMK2	0	0	0	PPR22	PPR21	PPR20
FFFFF118H	PIC3 ^{Note1}	PIF3	PMK3	0	0	0	PPR32	PPR31	PPR30
FFFFF11AH	PIC4 ^{Note1}	PIF4	PMK4	0	0	0	PPR42	PPR41	PPR40
FFFFF11CH	PIC5	PIF5	PMK5	0	0	0	PPR52	PPR51	PPR50
FFFFF11EH	PIC6	PIF6	PMK6	0	0	0	PPR62	PPR61	PPR60
FFFFF120H	PIC7	PIF7	PMK7	0	0	0	PPR72	PPR71	PPR70
FFFFF122H	TQ0OVIC	TQ0OVIF	TQ0OVMK	0	0	0	TQ0OVPR2	TQ0OVPR1	TQ0OVPR0
FFFFF124H	TQ0CCIC0	TQ0CCIF0	TQ0CCMK0	0	0	0	TQ0CCPR02	TQ0CCPR01	TQ0CCPR00
FFFFF126H	TQ0CCIC1	TQ0CCIF1	TQ0CCMK1	0	0	0	TQ0CCPR12	TQ0CCPR11	TQ0CCPR10
FFFFF128H	TQ0CCIC2	TQ0CCIF2	TQ0CCMK2	0	0	0	TQ0CCPR22	TQ0CCPR21	TQ0CCPR20
FFFFF12AH	TQ0CCIC3	TQ0CCIF3	TQ0CCMK3	0	0	0	TQ0CCPR32	TQ0CCPR31	TQ0CCPR30
FFFFF12CH	TP0OVIC	TP0OVIF	TP0OVMK	0	0	0	TP0OVPR2	TP0OVPR1	TP0OVPR0
FFFFF12EH	TP0CCIC0	TP0CCIF0	TP0CCMK0	0	0	0	TP0CCPR02	TP0CCPR01	TP0CCPR00
FFFFF130H	TP0CCIC1	TP0CCIF1	TP0CCMK1	0	0	0	TP0CCPR12	TP0CCPR11	TP0CCPR10
FFFFF132H	TP1OVIC	TP1OVIF	TP1OVMK	0	0	0	TP1OVPR2	TP1OVPR1	TP1OVPR0
FFFFF134H	TP1CCIC0	TP1CCIF0	TP1CCMK0	0	0	0	TP1CCPR02	TP1CCPR01	TP1CCPR00
FFFFF136H	TP1CCIC1	TP1CCIF1	TP1CCMK1	0	0	0	TP1CCPR12	TP1CCPR11	TP1CCPR10
FFFFF138H	TP2OVIC	TP2OVIF	TP2OVMK	0	0	0	TP2OVPR2	TP2OVPR1	TP2OVPR0
FFFFF13AH	TP2CCIC0	TP2CCIF0	TP2CCMK0	0	0	0	TP2CCPR02	TP2CCPR01	TP2CCPR00
FFFFF13CH	TP2CCIC1	TP2CCIF1	TP2CCMK1	0	0	0	TP2CCPR12	TP2CCPR11	TP2CCPR10
FFFFF13EH	TP3OVIC	TP3OVIF	TP3OVMK	0	0	0	TP3OVPR2	TP3OVPR1	TP3OVPR0
FFFFF140H	TP3CCIC0	TP3CCIF0	TP3CCMK0	0	0	0	TP3CCPR02	TP3CCPR01	TP3CCPR00
FFFFF142H	TP3CCIC1	TP3CCIF1	TP3CCMK1	0	0	0	TP3CCPR12	TP3CCPR11	TP3CCPR10
FFFFF144H	TP4OVIC	TP4OVIF	TP4OVMK	0	0	0	TP4OVPR2	TP4OVPR1	TP4OVPR0
FFFFF146H	TP4CCIC0	TP4CCIF0	TP4CCMK0	0	0	0	TP4CCPR02	TP4CCPR01	TP4CCPR00
FFFFF148H	TP4CCIC1	TP4CCIF1	TP4CCMK1	0	0	0	TP4CCPR12	TP4CCPR11	TP4CCPR10
FFFFF14AH	TP5OVIC	TP5OVIF	TP5OVMK	0	0	0	TP5OVPR2	TP5OVPR1	TP5OVPR0
FFFFF14CH	TP5CCIC0	TP5CCIF0	TP5CCMK0	0	0	0	TP5CCPR02	TP5CCPR01	TP5CCPR00
FFFFF14EH	TP5CCIC1	TP5CCIF1	TP5CCMK1	0	0	0	TP5CCPR12	TP5CCPR11	TP5CCPR10
FFFFF150H	TM0EQIC0	TM0EQIF0	TM0EQMK0	0	0	0	TM0EQPR02	TM0EQPR01	TM0EQPR00
FFFFF152H	CB0RIC/ IICIC1	CB0RIF/ IICIF1	CB0RMK/ IICMK1	0	0	0	CB0RPR2/ IICPR12	CB0RPR1/ IICPR11	CB0RPR0/ IICPR10
FFFFF154H	CB0TIC	CB0TIF	CB0TMK	0	0	0	CB0TPR2	CB0TPR1	CB0TPR0
FFFFF156H	CB1RIC ^{Note2}	CB1RIF	CB1RMK	0	0	0	CB1RPR2	CB1RPR1	CB1RPR0
FFFFF158H	CB1TIC ^{Note2}	CB1TIF	CB1TMK	0	0	0	CB1TPR2	CB1TPR1	CB1TPR0
FFFFF15AH	CB2RIC	CB2RIF	CB2RMK	0	0	0	CB2RPR2	CB2RPR1	CB2RPR0
FFFFF15CH	CB2TIC	CB2TIF	CB2TMK	0	0	0	CB2TPR2	CB2TPR1	CB2TPR0
FFFFF15EH	CB3RIC ^{Note1}	CB3RIF	CB3RMK	0	0	0	CB3RPR2	CB3RPR1	CB3RPR0

Notes 1. V850ES/JE3-L only

2. V850ES/JC3-L (48-pin), V850ES/JE3-L only

Table 19-3. Interrupt Control Registers (xxICn) (2/2)

Address	Register	Bit							
		<7>	<6>	5	4	3	2	1	0
FFFFF160H	CB3TIC ^{Note1}	CB3TIF	CB3TMK	0	0	0	CB3TPR2	CB3TPR1	CB3TPR0
FFFFF162H	UA0RIC/ CB4RIC ^{Note2}	UA0RIF/ CB4RIF	UA0RMK/ CB4RMK	0	0	0	UA0RPR2/ CB4RPR2	UA0RPR1/ CB4RPR1	UA0RPR0/ CB4RPR0
FFFFF164H	UA0TIC/ CB4TIC ^{Note2}	UA0TIF/ CB4TIF	UA0TMK/ CB4TMK	0	0	0	UA0TPR2/ CB4TPR2	UA0TPR1/ CB4TPR1	UA0TPR0/ CB4TPR0
FFFFF166H	UA1RIC/ IICIC2	UA1RIF/ IICIF2	UA1RMK/ IICMK2	0	0	0	UA1RPR2/ IICPR22	UA1RPR1/ IICPR21	UA1RPR0/ IICPR20
FFFFF168H	UA1TIC	UA1TIF	UA1TMK	0	0	0	UA1TPR2	UA1TPR1	UA1TPR0
FFFFF16AH	UA2RIC ^{Note2} / IICIC0 ^{Note2}	UA2RIF/ IICIF0	UA2RMK/ IICMK0	0	0	0	UA2RPR2/ IICPR02	UA2RPR1/ IICPR01	UA2RPR0/ IICPR00
FFFFF16CH	UA2TIC ^{Note2}	UA2TIF	UA2TMK	0	0	0	UA2TPR2	UA2TPR1	UA2TPR0
FFFFF16EH	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0
FFFFF170H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFF172H	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF174H	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFF176H	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30
FFFFF178H	KRIC	KRIF	KRMK	0	0	0	KRPR2	KRPR1	KRPR0
FFFFF17AH	WTIIC /RTC2IC	WTIIF /RTC2IF	WTIMK /RTC2MK	0	0	0	WTIPR2 /RTC2PPR2	WTIPR1 /RTC2PPR1	WTIPR0 /RTC2PPR0
FFFFF17CH	WTIC /RTC0IC	WTIF /RTC0IF	WTMK /RTC0MK	0	0	0	WTPR2 /RTC0PPR2	WTPR1 /RTC0PPR1	WTPR0 /RTC0PPR0
FFFFF17EH	RTC1IC	RTC1IF	RTC1MK	0	0	0	RTC1PPR2	RTC1PPR1	RTC1PPR0

Notes 1. V850ES/JE3-L only

2. V850ES/JC3-L (48-pin), V850ES/JE3-L only

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

The IMR0 to IMR3 registers specify masking of the maskable interrupts. The xxMKn bit of the IMR0 to IMR3 registers is equivalent to the xxICn.xxMKn bit.

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

If the higher 8 bits of each IMRm register are used as an IMRmH register and the lower 8 bits as an IMRmL register, these registers can be read or written in 8-bit or 1-bit units (m = 0 to 3).

Reset sets these registers to FFFFH.

Caution The device file defines the xxICn.xxMKn bit as a reserved word. If a bit is manipulated using the name of xxMKn, the values of the xxICn register, instead of the IMRm register, are rewritten (as a result, the values 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 ^{Note1})	1	1	1	1	1	1	1	1
	7	6	5	4	3	2	1	0
IMR3L	RTC1MK	WTMK/ RTC0MK	WTMK/ RTC2MK	KRMK	DMAMK3	DMAMK2	DMAMK1	DMAMK0

After reset: FFFFH R/W Address: IMR2 FFFFF104H, IMR2L FFFFF104H, IMR2H FFFFF105H

	15	14	13	12	11	10	9	8
IMR2 (IMR2H ^{Note1})	ADMK	UA2TMK ^{Note2}	UA2RMK ^{Note2} / IICMK0	UA1TMK	UA1RMK/ IIC2MK	UA0TMK/ CB4TMK ^{Note2}	UA0RMK/ CB4RMK ^{Note2}	CB3TMK ^{Note3}
	7	6	5	4	3	2	1	0
IMR2L	CB3RMK ^{Note3}	CB2TMK	CB2RMK	CB1TMK ^{Note2}	CB1RMK ^{Note2}	CB0TMK	CB0RMK/ IICMK1	TM0EQMK0

After reset: FFFFH R/W Address: IMR1 FFFFF102H, IMR1L FFFFF102H, IMR1H FFFFF103H

	15	14	13	12	11	10	9	8
IMR1 (IMR1H ^{Note1})	TP5CCMK1	TP5CCMK0	TP5OVMK	TP4CCMK1	TP4CCMK0	TP4OVMK	TP3CCMK1	TP3CCMK0
	7	6	5	4	3	2	1	0
IMR1L	TP3OVMK	TP2CCMK1	TP2CCMK0	TP2OVMK	TP1CCMK1	TP1CCMK0	TP1OVMK	TP0CCMK1

After reset: FFFFH R/W Address: IMR0 FFFFF100H, IMR0L FFFFF100H, IMR0H FFFFF101H

	15	14	13	12	11	10	9	8
IMR0 (IMR0H ^{Note1})	TP0CCMK0	TP0OVMK	TQ0CCMK3	TQ0CCMK2	TQ0CCMK1	TQ0CCMK0	TQ0OVMK	PMK7
	7	6	5	4	3	2	1	0
IMR0L	PMK6	PMK5	PMK4 ^{Note3}	PMK3 ^{Note3}	PMK2	PMK1 ^{Note3}	PMK0	LVIMK

xxMKn	Setting of interrupt mask flag
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Note 1. To read or write bits 8 to 15 of the IMR0 to IMR3 registers in 8-bit or 1-bit units, specify them as bits 0 to 7 of IMR0H to IMR3H registers.

2. For the V850ES/JE3-L, 0 or 1 can be specified for this bit.
For the V850ES/JC3-L, this bit must be set to 1.

3. For the V850ES/JE3-L and V850ES/JC3-L (48-pin), 0 or 1 can be specified for this bit.
For the V850ES/JC3-L (40-pin), this bit must be set to 1.

Caution Set bits 8 to 15 of the IMR3 register to 1. If the setting of these bits is changed, the operation is not guaranteed.

Remark xx: Identification name of each peripheral unit (see Table 19-3 Interrupt Control Registers (xxICn)).
n: Peripheral unit number (see Table 19-3 Interrupt Control Registers (xxICn)).

19.3.6 In-service priority register (ISPR)

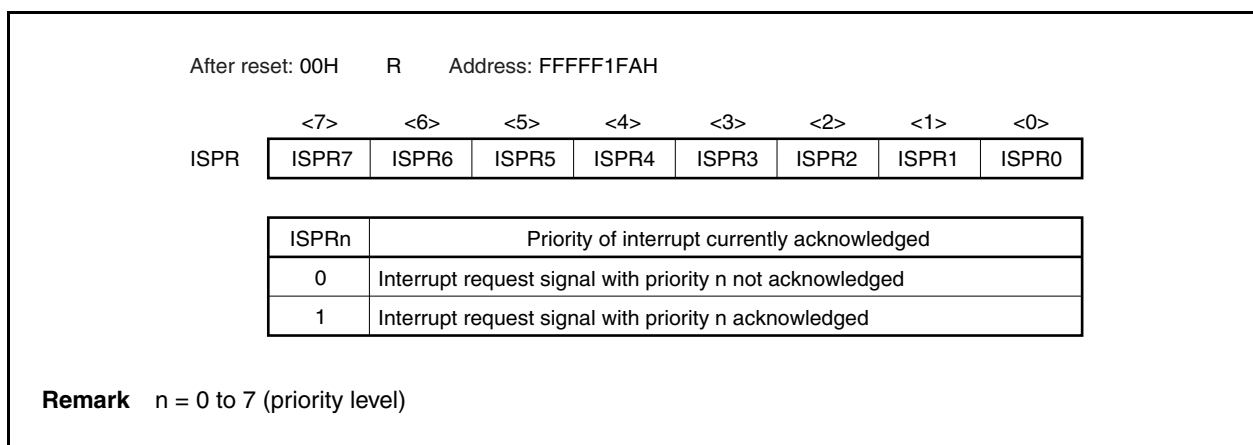
The ISPR register holds the priority level of the maskable interrupt currently acknowledged. When an interrupt request signal is acknowledged, the bit of this register corresponding to the priority level of that interrupt request signal is set to 1 and remains set while the interrupt is serviced.

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 sets this register to 00H.

Caution If an interrupt is acknowledged while the ISPR register is being read in the interrupt enabled (EI) status, the value of the ISPR register after the bits of the register have been set by acknowledging the interrupt may be read. To accurately read the value of the ISPR register before an interrupt is acknowledged, read the register while interrupts are disabled (DI).



19.3.7 ID flag

This flag stores information regarding enabling or disabling maskable interrupt request signals. The interrupt disable flag (ID) is assigned to the PSW.

Reset sets this flag to 1 and the PSW register to 00000020H.

After reset: 00000020H

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

ID	Specification of maskable interrupt servicing ^{Note}
0	Maskable interrupt request signal acknowledgment enabled
1	Maskable interrupt request signal acknowledgment disabled

Note Interrupt disable flag (ID) function
 This bit is set to 1 by the DI instruction and cleared to 0 by the EI instruction. Its value is also rewritten by the RETI instruction, or by an LDSR instruction that writes data to the PSW.
 Non-maskable interrupt request signals and exceptions are acknowledged regardless of this flag. When a maskable interrupt request signal is acknowledged, the ID flag is automatically set to 1 by hardware.
 An interrupt request signal generated during the acknowledgment disabled period (ID flag = 1) is acknowledged when the xxICn.xxIFn bit is set to 1, and the ID flag is cleared to 0.

19.3.8 Watchdog timer mode register 2 (WDTM2)

This register can be read or written in 8-bit units (for details, see CHAPTER 11 WATCHDOG TIMER 2).

Reset sets this register to 67H.

After reset: 67H R/W Address: FFFFF6D0H

	7	6	5	4	3	2	1	0
WDTM2	0	WDM21	WDM20	0	0	0	0	0

WDM21	WDM20	Selection of watchdog timer operation mode
0	0	Stops operating
0	1	Non-maskable interrupt request mode
1	×	Reset mode (initial-value)

19.4 Software Exception

A software exception occurs when the CPU executes the TRAP instruction, and can always be acknowledged.

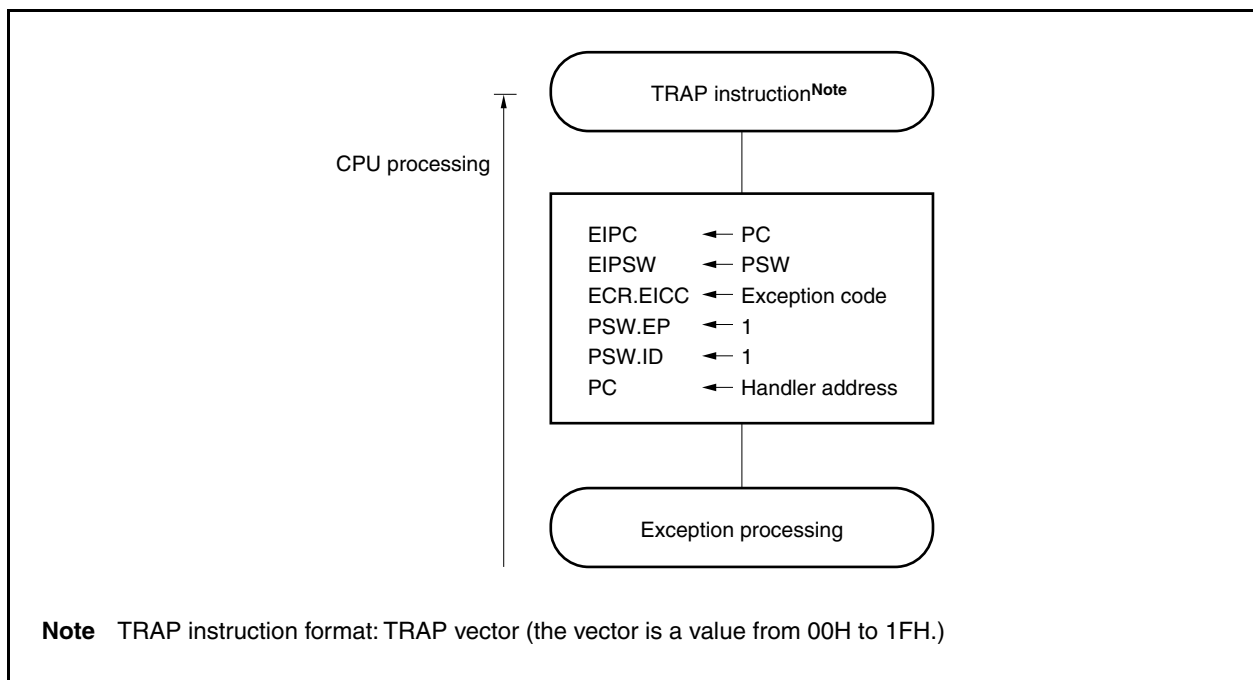
19.4.1 Operation

If a software exception occurs, the CPU performs the following processing and transfers control to the handler routine.

- <1> Saves the current 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) for the software exception to the PC and transfers control.

The processing of a software exception is shown below.

Figure 19-9. Software Exception Processing



The handler address is determined by the TRAP instruction's operand (vector). If the vector is 00H to 0FH, the handler address is 00000040H, and if the vector is 10H to 1FH, the handler address is 00000050H.

19.4.2 Restoration

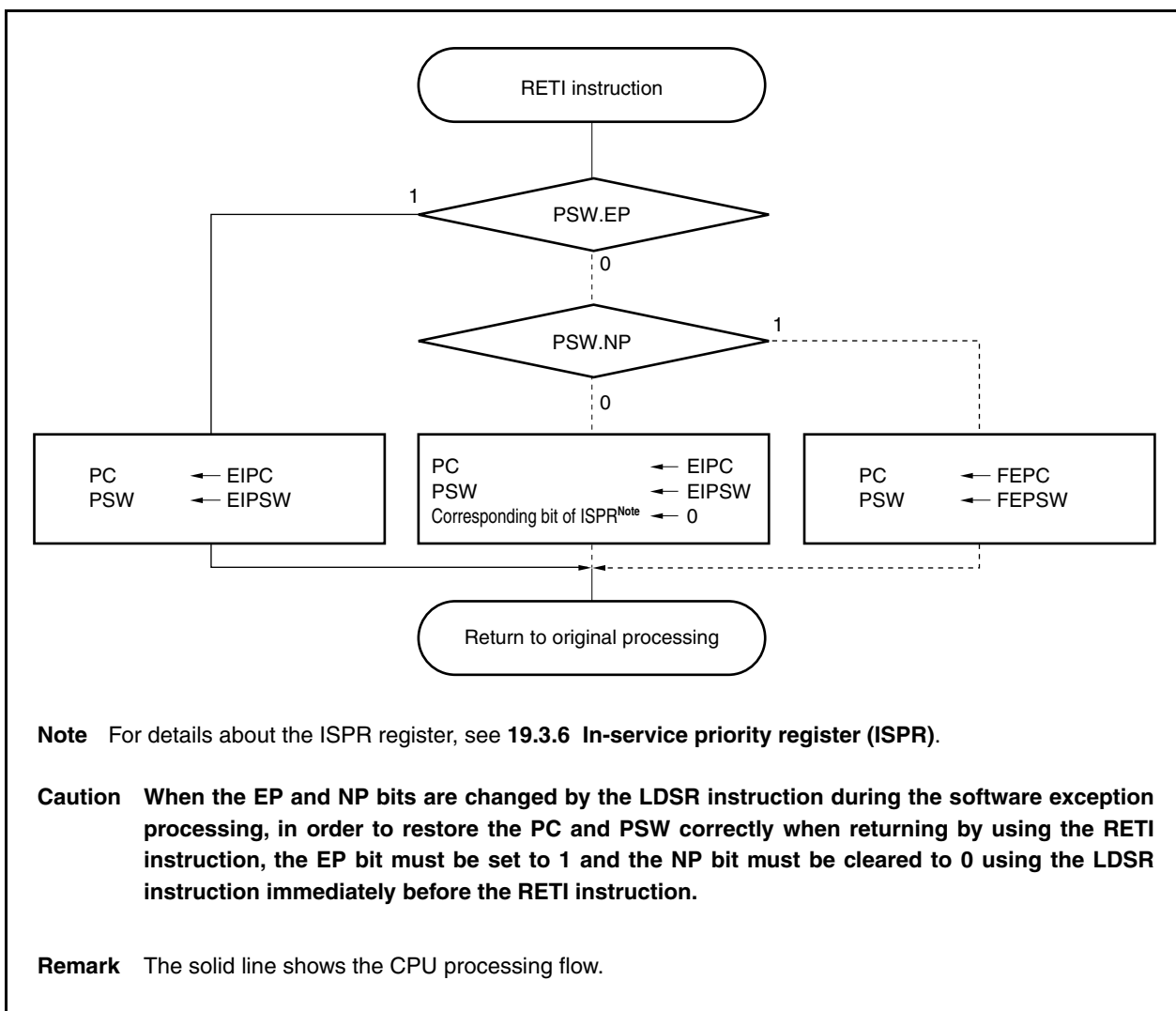
Execution is returned from software exception processing by the using RETI instruction.

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

- <1> Loads the saved PC and PSW from EIPC and EIPSW, respectively, because the PSW.EP bit is 1.
- <2> Transfers control back to the address of the restored PC and PSW.

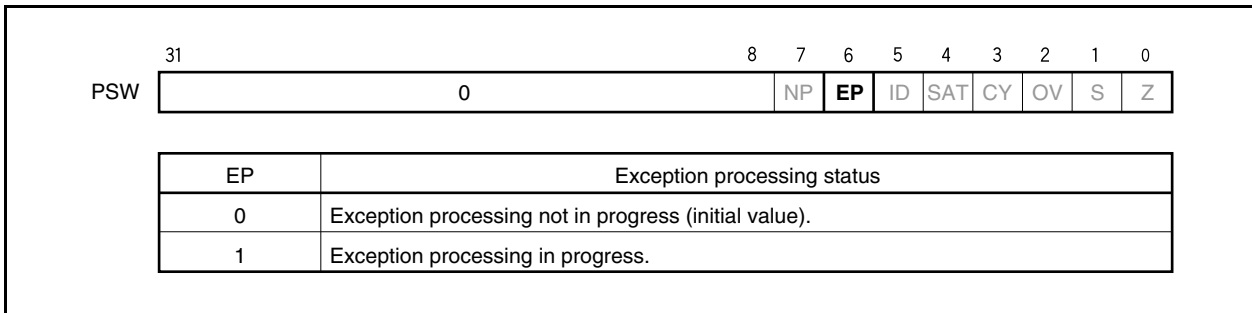
The processing of the RETI instruction is shown below.

Figure 19-10. RETI Instruction Processing



19.4.3 EP flag

The EP flag is a status flag that indicates that exception processing is in progress. This flag is set when an exception occurs.

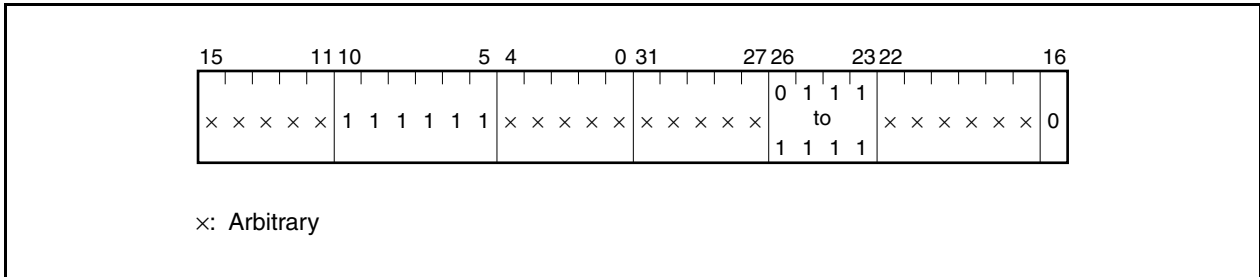


19.5 Exception Trap

An exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. In the V850ES/JC3-L and V850ES/JE3-L, an illegal opcode exception (ILGOP: Illegal Opcode Trap) is used as an exception trap.

19.5.1 Illegal opcode

An illegal opcode is defined as an instruction with instruction opcode (bits 10 to 5) = 11111B, sub-opcode (bits 26 to 23) = 0111B to 1111B, and sub-opcode (bit 16) = 0B. When such an instruction is executed, an exception trap occurs.



Caution Illegal opcodes must not be used because instructions may be newly assigned to these opcodes in the future.

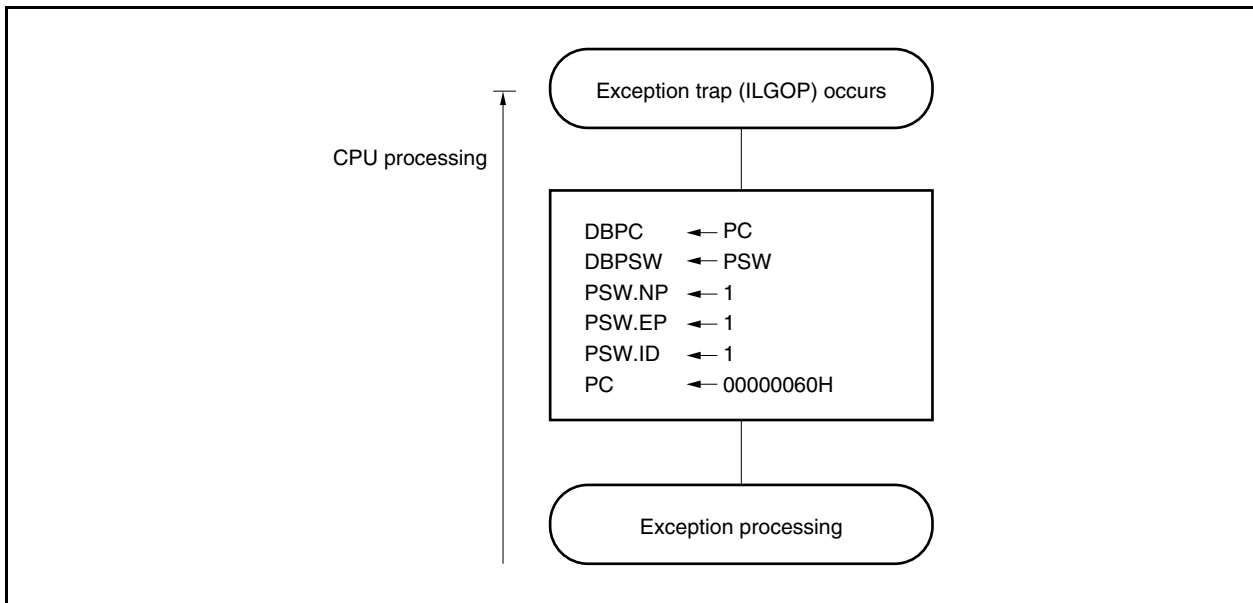
(1) Operation

If an exception trap occurs, the CPU performs the following processing and transfers control to the handler routine.

- <1> Saves the current 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) for the exception trap to the PC and transfers control.

The processing of an exception trap is shown below.

Figure 19-11. Exception Trap Processing



(2) Restoration

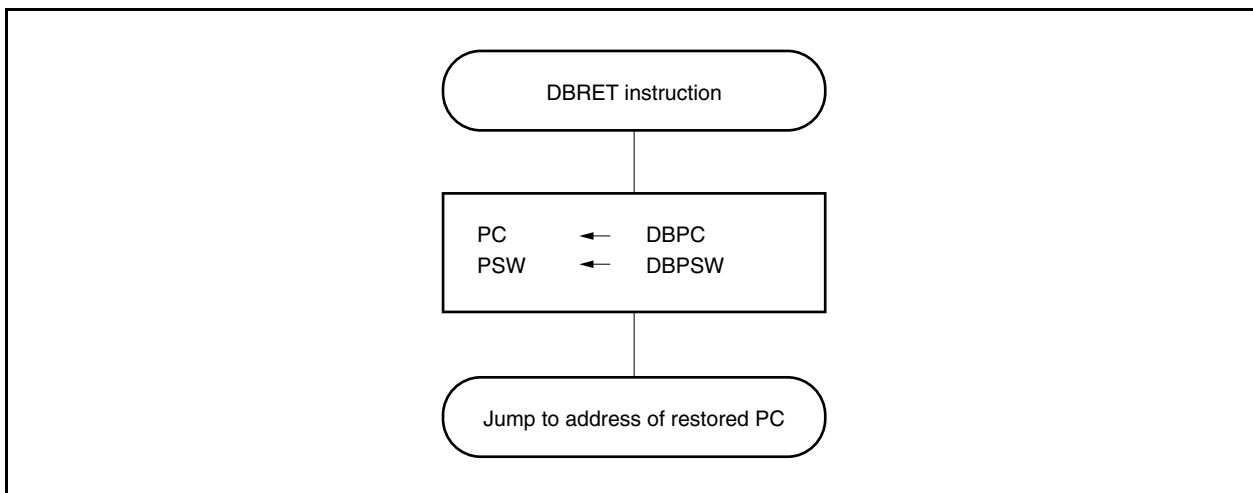
Execution is returned from an exception trap by using the DBRET instruction. When the DBRET instruction is executed, the CPU performs the following processing and transfers control to the address of the restored PC.

- <1> Loads the saved PC and PSW from DBPC and DBPSW.
- <2> Transfers control back to the address of the restored PC and PSW.

- Cautions**
1. DBPC and DBPSW can be accessed only during the interval between the execution of an illegal opcode and the DBRET instruction.
 2. If an illegal opcode is executed, specify the default settings or stop the subsequent processing.

The processing for returning from an exception trap is shown below.

Figure 19-12. Returning from Exception Trap



19.5.2 Debug trap

A debug trap is an exception that occurs when the DBTRAP instruction is executed and can always be acknowledged.

(1) Operation

If a debug trap occurs, the CPU performs the following processing.

<1> Saves the current PC to DBPC.

<2> Saves the current PSW to DBPSW.

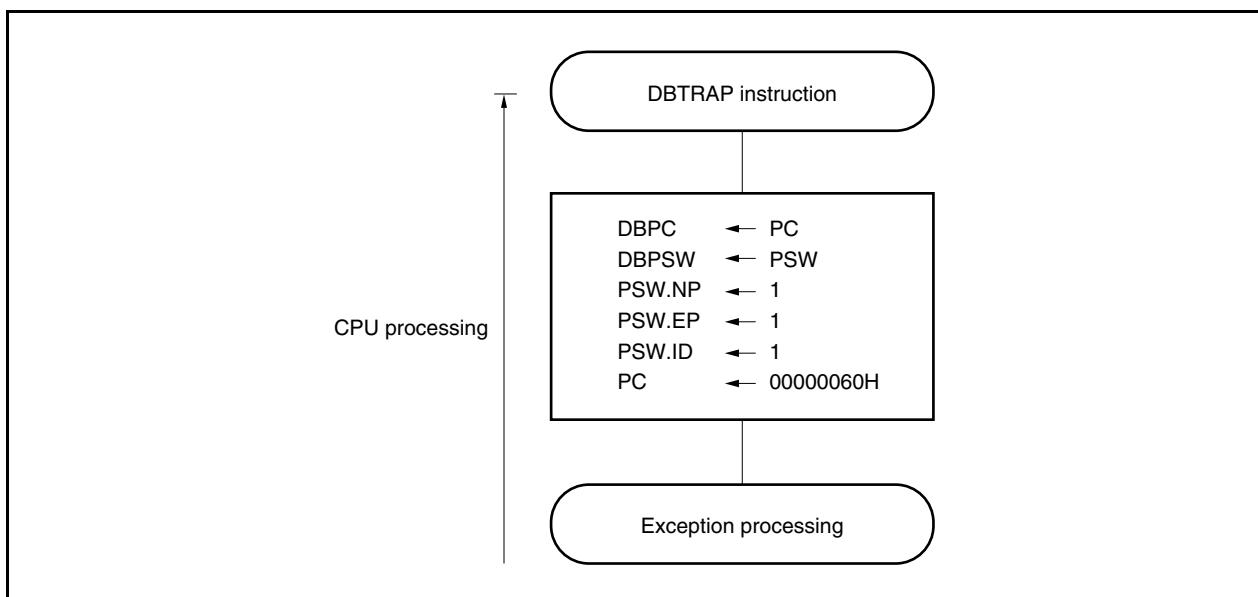
<3> Sets the PSW.NP, PSW.EP, and PSW.ID bits to 1.

<4> Sets the handler address (00000060H) for the debug trap to the PC and transfers control.

Caution The DBTRAP instruction is intended for debugging and is basically used by the debug tool. If the application uses this instruction while it is being executed by the debug tool, a malfunction might occur.

The processing of a debug trap is shown below.

Figure 19-13. Debug Trap Processing



(2) Restoration

Execution is returned from a debug trap by using the DBRET instruction.

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

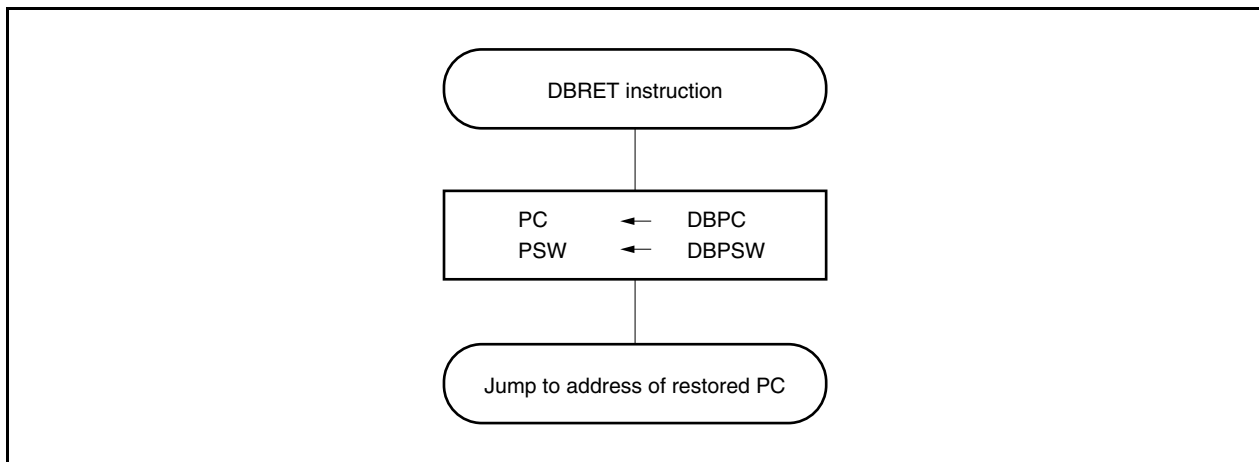
<1> Loads the saved PC and PSW from DBPC and DBPSW.

<2> Transfers control back to address of the restored PC and PSW.

Caution DBPC and DBPSW can be accessed only during the interval between the execution of the DBTRAP instruction and the DBRET instruction.

The processing for returning from a debug trap is shown below.

Figure 19-14. Returning from Debug Trap



19.6 Multiple Interrupt Servicing Control

In multiple interrupt servicing control, the servicing of an interrupt is stopped if an interrupt request signal that has a higher priority level is generated. The higher priority interrupt request signal is then acknowledged and the interrupt is serviced.

If an interrupt request signal with a lower or equal priority level is generated while an interrupt is being serviced, the newly generated interrupt request signal will be held pending.

Multiple interrupt servicing control is performed when interrupts are enabled (PSW.ID bit = 0). Even in an interrupt service routine, multiple interrupt control must be performed while interrupts are enabled (ID bit = 0). If a maskable interrupt or software exception occurs in a maskable interrupt or software exception service program, EIPC and EIPSW must be saved.

The following example shows the procedure for servicing multiple interrupts.

(1) To acknowledge maskable interrupt request signals in a service program

Service program for maskable interrupt or exception

```
...  
...  
• EIPC saved to memory or register  
• EIPSW saved to memory or register  
• EI instruction (enables interrupt acknowledgment)  
...  
...  
...  
...  
• DI instruction (disables interrupt acknowledgment)  
• Saved value restored to EIPSW  
• Saved value restored to EIPC  
• RETI instruction
```

←Acknowledges maskable interrupt

19.7 External Interrupt Request Input Pins (NMI, INTP0 to INTP7)

19.7.1 Noise elimination

(1) Noise elimination for NMI pin

The NMI pin has an internal noise eliminator that uses analog delay (several 10 ns). Therefore, a signal input to the NMI pin is not detected as an edge unless it maintains its input level for a certain period. The edge is detected only after a certain period has elapsed.

The NMI pin is used for releasing the STOP mode. In the STOP mode, noise elimination using the system clock is not performed because the internal system clock is stopped.

(2) Noise elimination for INTP0 to INTP7 pins

The INTP0 to INTP7 pins have an internal noise eliminator that uses analog delay (several 10 ns). Therefore, a signal input to each pin is not detected as an edge unless it maintains its input level for a certain period. The edge is detected only after a certain period has elapsed.

(3) Noise elimination for INTP3 pin

The INTP3 pin has an internal digital/analog noise eliminator, and digital or analog noise elimination can be selected by using the NFC.NFEN bit (analog delay: several 10 ns).

The sampling clock can be selected from $f_{xx}/64$, $f_{xx}/128$, $f_{xx}/256$, $f_{xx}/512$, $f_{xx}/1,024$, or f_{XT} by using the NFC.NFC2 to NFC.NFC0 bits. If the sampling clock is set to $f_{xx}/64$, $f_{xx}/128$, $f_{xx}/256$, $f_{xx}/512$, or $f_{xx}/1,024$, the sampling clock stops in the IDLE or STOP mode. It cannot therefore be used to release a standby mode. To release a standby mode, select f_{XT} as the sampling clock or select the analog noise eliminator.

19.7.2 Edge detection

The valid edge of each of the NMI and INTP0 to INTP7 pins can be selected from the following four.

- Rising edge
- Falling edge
- Both rising and falling edges
- No edge detected

Caution The NMI pin alternately functions as the P02 pin, and functions as a normal port pin after being reset. To enable the NMI pin function, use the PMC0 register. The initial setting of the NMI pin is “No edge detected”. Select the NMI pin valid edge by using the INTF0 and INTR0 registers.

(1) External interrupt falling, rising edge specification register 0 (INTF0, INTR0)

The INTF0 and INTR0 registers are 8-bit registers that specify detection of the falling and rising edges of the NMI pin via bit 2 and the external interrupt pins (INTP0 to INTP3) via bits 3 to 6.

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

Reset sets these registers to 00H.

Caution When switching from the port function to the external interrupt function (alternate function), an edge might be detected. Therefore, set the INTF0n and INTR0n bits to 00, and then specify the external interrupt function (PMC0.PMC0n bit = 1).

When switching from the external interrupt function to the port function, an edge might be detected as well. Therefore, set the INTF0n and INTR0n bits to 00, and then specify the port function (PMC0.PMC0n bit = 0).

After reset: 00H R/W Address: INTF0 FFFFC00H, INTR0 FFFFC20H

	7	6	5	4	3	2	1	0
INTF0	0	INTF06 ^{Note}	INTF05	INTF04 ^{Note}	INTF03	INTF02	0	0
		INTP3	INTP2	INTP1	INTP0	NMI		
	7	6	5	4	3	2	1	0
INTR0	0	INTR06 ^{Note}	INTR05	INTR04 ^{Note}	INTR03	INTR02	0	0
		INTP3	INTP2	INTP1	INTP0	NMI		

Note For the V850ES/JE3-L, 0 or 1 can be specified for this bit. For the V850ES/JC3-L, this bit must be cleared to 0.

Remark For how to specify a valid edge, see **Table 19-4**.

Table 19-4. Valid Edge Specification

INTF0n	INTR0n	Valid Edge Specification (n = 2 to 6)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution Be sure to clear the INTF0n and INTR0n bits to 00 when these registers are not used for the NMI or INTP0 to INTP3 pins.

Remark n = 2: Control of NMI pin
 n = 3 to 6: Control of INTP0 to INTP3 pins

(2) External interrupt falling, rising edge specification register 3 (INTF3, INTR3)

The INTF3 and INTR3 registers are 8-bit registers that specify detection of the falling and rising edges of the external interrupt pin (INTP7).

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

Reset sets these registers to 00H.

Cautions 1. When switching from the port function to the external interrupt function (alternate function), an edge might be detected. Therefore, set the INTF31 and INTR31 bits to 00, and then specify the external interrupt function (PMC3.PMC31 bit = 1).

When switching from the external interrupt function to the port function, an edge might be detected as well. Therefore, set the INTF31 and INTR31 bits to 00, and then specify the port function (PMC3.PMC31 bit = 0).

2. The INTP7 pin and RXDA0 pin are alternate-function pins. When using the pin as the RXDA0 pin, disable edge detection for the INTP7 alternate-function pin (clear the INTF3.INTF31 bit and the INTR3.INTR31 bit to 0). When using the pin as the INTP7 pin, stop UARTA0 reception (clear the UA0CTL0.UA0RXE bit to 0).

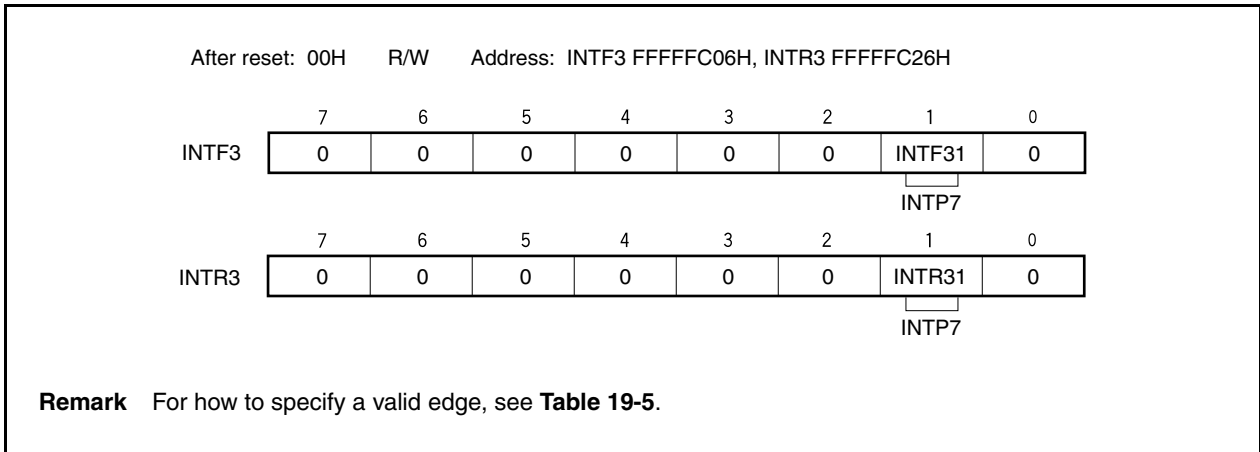


Table 19-5. Valid Edge Specification

INTF31	INTR31	Valid Edge Specification
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution Be sure to clear the INTF31 and INTR31 bits to 00 when these registers are not used for the INTP7 pin.

(3) External interrupt falling, rising edge specification register 9H (INTF9H, INTR9H)

The INTF9H and INTR9H registers are 8-bit registers that specify detection of the falling and rising edges of the external interrupt pins (INTP4 to INTP6).

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

Reset sets these registers to 00H.

Caution When switching from the port function to the external interrupt function (alternate function), an edge might be detected. Therefore, set the INTF9n and INTR9n bits to 00, and then specify the external interrupt function (PMC9.PMC9n bit = 1).

When switching from the external interrupt function to the port function, an edge might be detected as well. Therefore, set the INTF9n and INTR9n bits to 00, and then specify the port function (PMC9.PMC9n bit = 0).

After reset: 00H R/W Address: INTF9H FFFFFFFC13H, INTR9H FFFFFFFC33H

INTF9H	15	14	13	12	11	10	9	8
	INTF915	INTF914	INTF913 ^{Note}	0	0	0	0	0
	INTP6	INTP5	INTP4					
INTR9H	15	14	13	12	11	10	9	8
	INTR915	INTR914	INTR913 ^{Note}	0	0	0	0	0
	INTP6	INTP5	INTP4					

Note For the V850ES/JE3-L, 0 or 1 can be specified for this bit. For the V850ES/JC3-L, this bit must be cleared to 0.

Remark For how to specify a valid edge, see **Table 19-6**.

Table 19-6. Valid Edge Specification

INTF9n	INTR9n	Valid Edge Specification (n = 13 to 15)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution Be sure to clear the INTF9n and INTR9n bits to 00 when these registers are not used for the INTP4 to INTP6 pins.

Remark n = 13 to 15: Control of INTP4 to INTP6 pins

(4) Noise elimination control register (NFC) (V850ES/JE3-L only)

Digital noise elimination can be selected for the INTP3 pin. The noise elimination settings are specified by using the NFC register.

When digital noise elimination is selected, the sampling clock for digital sampling can be selected from $f_{xx}/64$, $f_{xx}/128$, $f_{xx}/256$, $f_{xx}/512$, $f_{xx}/1,024$, or f_{XT} . Sampling is performed three times.

Even when digital noise elimination is selected, using f_{XT} as the sampling clock makes it possible to use the INTP3 interrupt request signal to release the IDLE1, IDLE2, and STOP modes.

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

Reset sets this register to 00H.

Caution After the sampling clock has been changed, it takes 3 sampling clock cycles to initialize the digital noise eliminator. Therefore, if an INTP3 valid edge is input within these 3 sampling clock cycles after the sampling clock has been changed, an interrupt request signal may be generated. Therefore, be careful about the following points when using the interrupt and DMA functions.

- When using the interrupt function, after the 3 sampling clock cycles have elapsed, enable interrupts after the interrupt request flag (PIC3.PIF3 bit) has been cleared.
- When using the DMA function (started by INTP3), enable DMA after 3 sampling clock cycles have elapsed.

After reset: 00H R/W Address: FFFFF318H

	7	6	5	4	3	2	1	0
NFC	NFEN	0	0	0	0	NFC2	NFC1	NFC0

NFEN	Settings of INTP3 pin noise elimination
0	Analog noise elimination (60 ns (TYP.))
1	Digital noise elimination

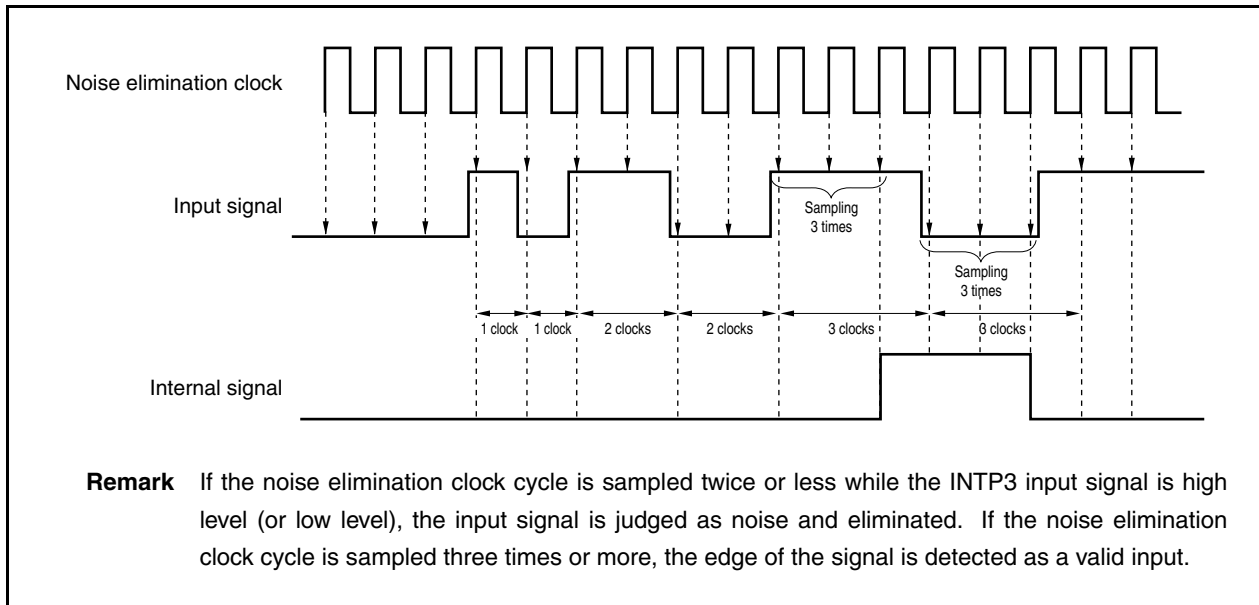
NFC2	NFC1	NFC0	Digital sampling clock
0	0	0	$f_{xx}/64$
0	0	1	$f_{xx}/128$
0	1	0	$f_{xx}/256$
0	1	1	$f_{xx}/512$
1	0	0	$f_{xx}/1,024$
1	0	1	f_{XT} (subclock)
Other than above			Setting prohibited

Remarks

1. Since sampling is performed three times, the reliably eliminated noise width is 2 sampling clock cycles.
2. In the case of noise with a width smaller than 2 sampling clock cycles, an interrupt request signal is generated if noise synchronized with the sampling clock is input.

An example of the timing of noise elimination performed by the timer T input pin digital filter is shown Figure 19-15.

Figure 19-15. Example of Digital Noise Elimination Timing

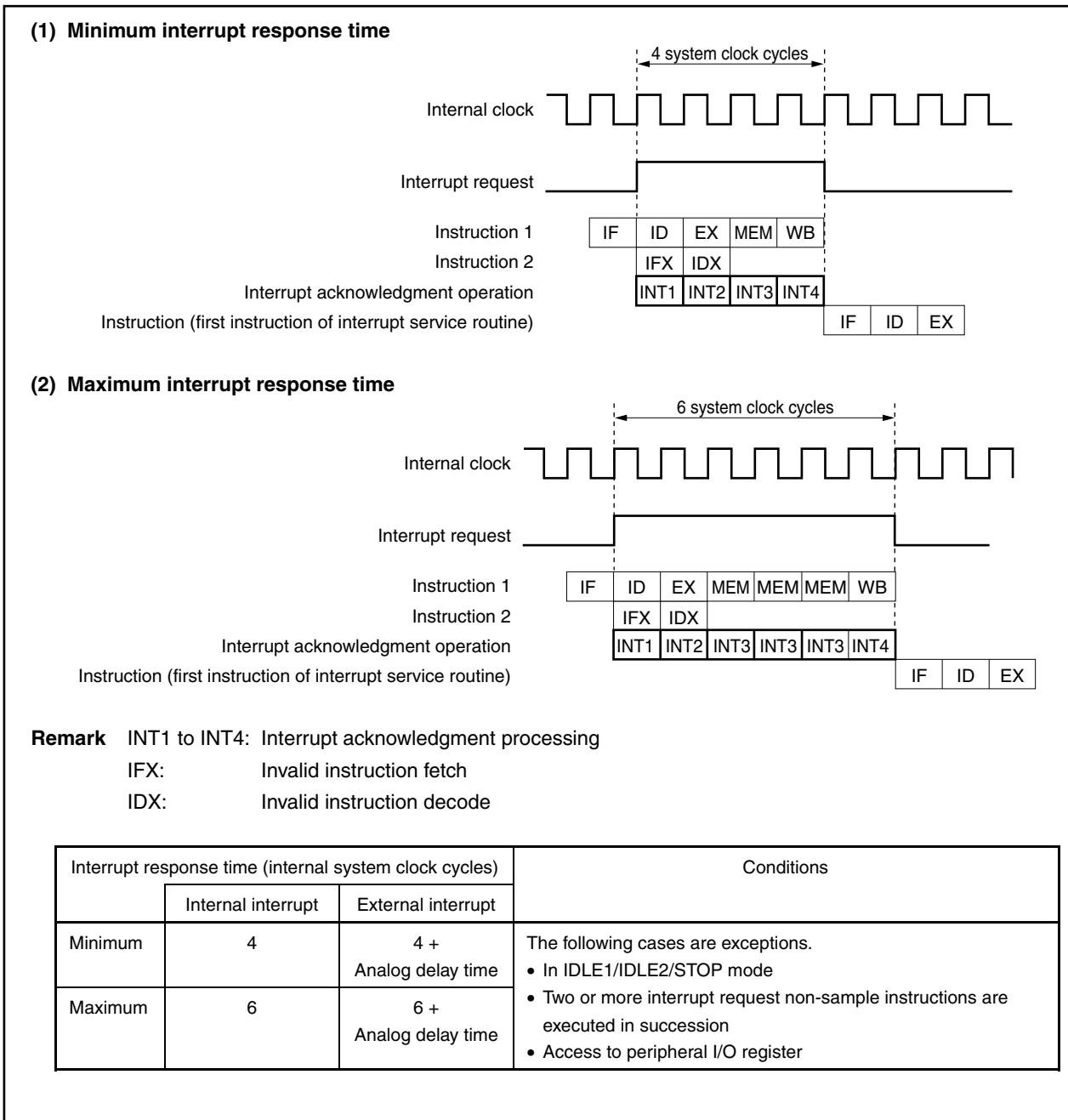


19.8 Interrupt Response Time of CPU

Except for the following cases, the interrupt response time of the CPU is at least 4 clock cycles. To input interrupt request signals successively, input the next interrupt request signal at least 5 clock cycles after the preceding interrupt.

- In IDLE1/IDLE2/STOP mode
- When interrupt request non-sample instructions are successively executed (see 19.9 Periods in Which Interrupts Are Not Acknowledged by CPU.)
- When an interrupt control register is accessed

Figure 19-16. Pipeline Operation When Interrupt Request Signal Is Acknowledged (Outline)



19.9 Periods in Which Interrupts Are Not Acknowledged by CPU

An interrupt is acknowledged by the CPU while an instruction is being executed. However, no interrupt will be acknowledged between an interrupt request non-sample instruction and the next instruction (the 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 PRCMD register
- The store, SET1, NOT1, or 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)
 - On-chip debug mode register (OCDM)

- Remarks**
1. xx: Identification name of each peripheral unit (see **Table 19-3 Interrupt Control Registers (xxICn)**)
n: Peripheral unit number (see **Table 19-3 Interrupt Control Registers (xxICn)**).
 2. For details about the operation of the pipeline, see the **V850ES Architecture User's Manual (U15943E)**.

19.10 Cautions

19.10.1 Restored PC

Restored PC is the value of the program counter (PC) saved to EIPC, FEPC, or DBPC when interrupt servicing starts. If a non-maskable or maskable interrupt is acknowledged during the execution of any of the following instructions, the execution of that instruction stops and resumes following completion of interrupt servicing.

- Load instructions (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)
- Divide instructions (DIV, DIVH, DIVU, DIVHU)
- PREPARE, DISPOSE instructions (only when an interrupt occurs before the stack pointer is updated)

CHAPTER 20 KEY INTERRUPT FUNCTION

20.1 Function

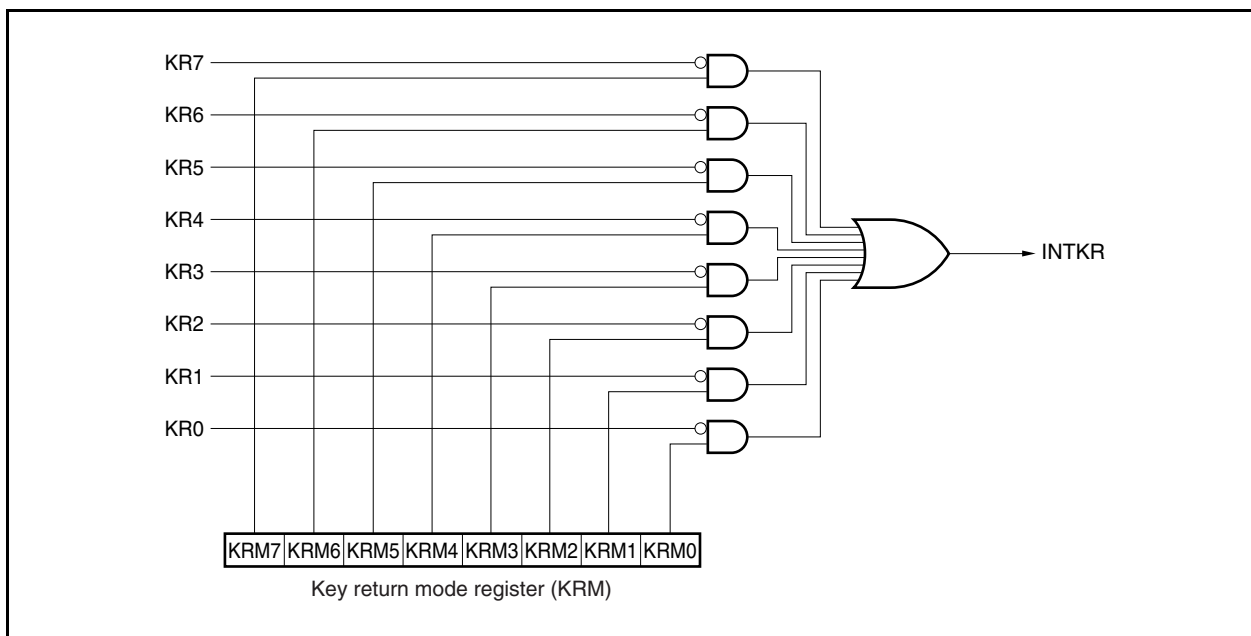
A key interrupt request signal (INTKR) can be generated by inputting a falling edge to the eight key input pins (KR0 to KR7) by setting the KRM register.

Caution If any of the KR0 to KR7 pins is at low level, the INTKR signal is not generated even if a falling edge is input to another pin.

Table 20-1. Flag Assignment

Flag	Pin Description	Alternate Function
KRM0	Controls KR0 signal.	P50
KRM1	Controls KR1 signal.	P51
KRM2	Controls KR2 signal.	P52
KRM3	Controls KR3 signal.	P53
KRM4	Controls KR4 signal.	P54
KRM5	Controls KR5 signal.	P55
KRM6	Controls KR6 signal.	P90
KRM7	Controls KR7 signal.	P91

Figure 20-1. Key Return Block Diagram



20.2 Pin Functions

The key input pins that are used as key interrupts are also used for the other functions shown in **Table 20-2**. To use these pins as key interrupts, this function must be specified by setting the relevant registers (see **Table 4-17 Settings When Pins Are Used for Alternate Functions**).

Table 20-2. Pin Functions

Pin No.	Port Function	Key Input Function	Other Functions
37	P50	KR0	P50/TIQ01/TOQ01/RTP00
38	P51	KR1	P51/TIQ02/TOQ02/RTP01
39	P52	KR2	P52/TIQ03/TOQ03/RTP02/DDI
40	P53	KR3	P53/SIB2/TIQ00/TOQ00/RTP03/DDO
41	P54	KR4	P54/SOB2/RTP04/DCK
42	P55	KR5	P55/SCKB2/RTP05/DMS
61	P90	KR6	P90/A0/TXDA1/SDA02
62	P91	KR7	P91/A1/RXDA1/SCL02

20.3 Registers

(1) Key return mode register (KRM)

The KRM register controls the KR0 to KR7 signals by using the KRM0 to KRM7 bits.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFF300H

	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	Control of key return mode
0	Do not detect key return signal
1	Detect key return signal

Caution Clear the KRM register to 00H before rewriting it.

20.4 Cautions

- (1) If a low level is input to any of the KR0 to KR7 pins, the INTKR signal is not generated even if the falling edge is input to another pin.
- (2) The RXDA1 and KR7 pins must not be used at the same time. When using the RXDA1 pin, do not use the KR7 pin. When using the KR7 pin, do not use the RXDA1 pin (it is recommended to set the PFC91 bit to 1 and clear PFCE91 bit to 0).
- (3) If the KRM register is changed, an interrupt request signal (INTKR) may be generated. To prevent this, change the KRM register after disabling (DI) or masking interrupts, then clear the interrupt request flag (KRIC.KRIF bit) to 0, and enable (EI) or unmask interrupts.
- (4) To use the key interrupt function, be sure to set the function of the port pin to “key return pin” and then enable the key interrupt function by using the KRM register. To switch the pin function from key return pin to port pin, disable the key interrupt function by using the KRM register and then set pin function to “port pin”.

CHAPTER 21 STANDBY FUNCTION

21.1 Overview

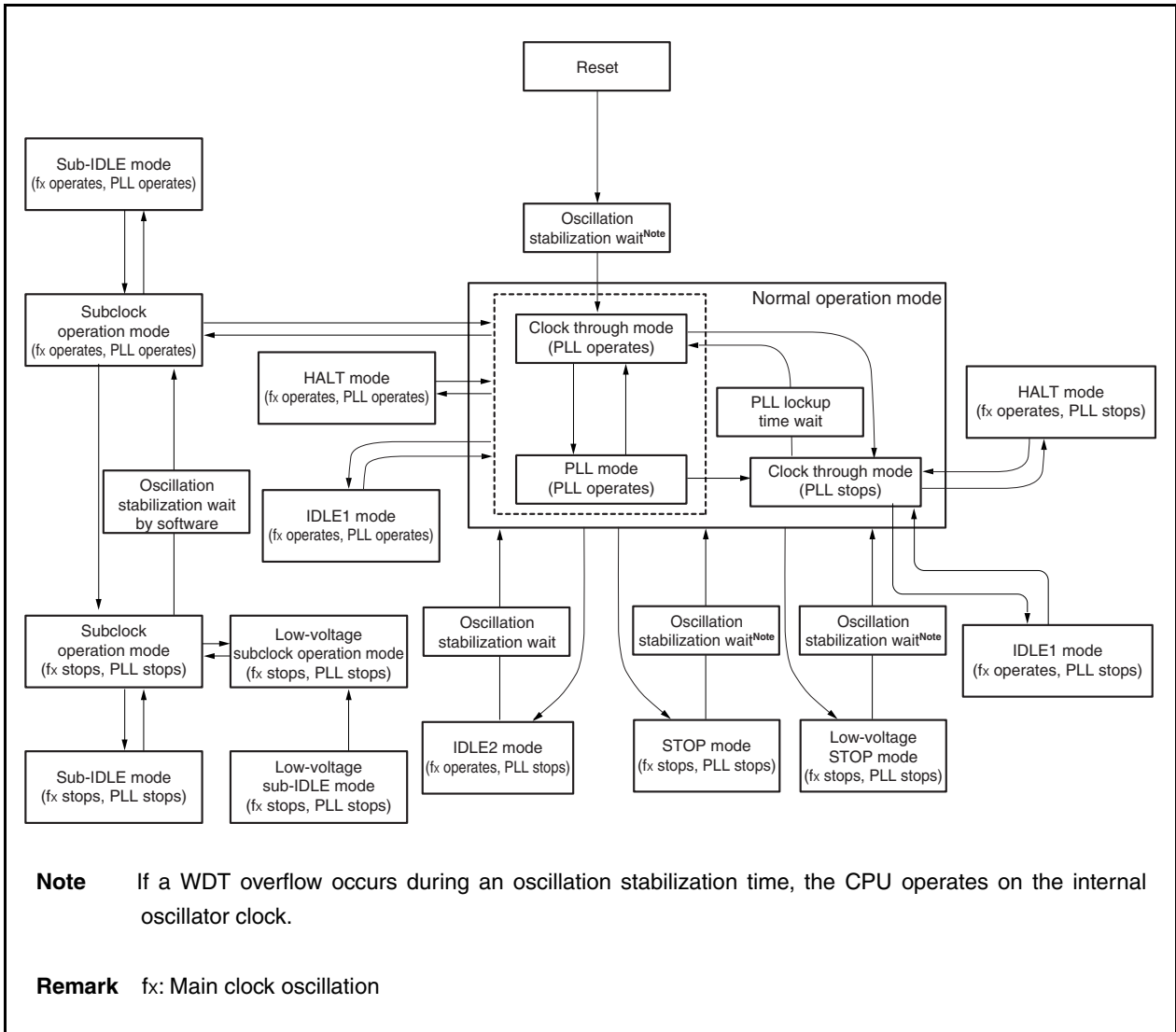
The power consumption of the system can be effectively reduced by using the standby modes in combination and selecting the appropriate mode for the application. The available standby modes are listed in Table 21-1.

Table 21-1. Standby Modes

Mode	Functional Outline
HALT mode	Mode in which only the operating clock of the CPU is stopped. The total current consumption of the system can be reduced by using this mode in combination with the normal operation mode for intermittent operation.
IDLE1 mode	Mode in which all the operations of the internal circuits except the oscillator, PLL ^{Note} , and flash memory are stopped. This mode can reduce the power consumption to a level lower than the HALT mode because it stops the operation of the on-chip peripheral functions.
IDLE2 mode	Mode in which all the operations of the internal circuits except the oscillator are stopped. This mode can reduce the power consumption to a level lower than the IDLE1 mode because it stops the operations of the on-chip peripheral functions, PLL, and flash memory.
STOP mode	Mode in which all the operations of the internal circuits except the subclock oscillator are stopped. This mode can reduce the power consumption to a level lower than the IDLE2 mode. Two modes are available: STOP mode and low-voltage STOP mode. The power consumption decreases further in the low-voltage STOP mode because the voltage of the regulator is lowered.
Subclock operation mode	Mode in which the subclock is used as the internal system clock. This mode can reduce the power consumption to a level lower than the normal operation mode. Two modes are available: subclock operation mode and low-voltage subclock operation mode. The power consumption decreases further in the low-voltage subclock operation mode because the voltage of the regulator is lowered.
Sub-IDLE mode	Mode in which all the operations of the internal circuits except the oscillator, PLL operation ^{Note} , and flash memory are stopped, in the subclock operation mode. This mode can reduce the power consumption to a level lower than the subclock operation mode. Two modes are available: sub-IDLE mode and low-voltage sub-IDLE mode. The power consumption decreases further in the low-voltage sub-IDLE mode because the voltage of the regulator is lowered.

Note In the IDLE1 or sub-IDLE mode, the PLL retains the operating status immediately before mode transition. If the PLL operation is not necessary, stop the PLL to lower the power consumption. In the IDLE2 mode, mode transition causes the PLL to stop automatically.

Figure 21-1. Status Transition



21.2 Registers

(1) Power save control register (PSC)

The PSC register is an 8-bit register that controls the standby function. The STP bit of this register is used to specify the standby mode. This register is a special register that can only be written in a specific sequence (see **3.4.7 Special registers**).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF1FEH

	7	<6>	<5>	<4>	3	2	<1>	0
PSC	0	NMI1M	NMI0M	INTM	0	0	STP	0

NMI1M	Standby mode release control upon occurrence of INTWDT2 signal
0	Standby mode release by INTWDT2 signal enabled
1	Standby mode release by INTWDT2 signal disabled

NMI0M	Standby mode release control by NMI pin input
0	Standby mode release by NMI pin input enabled
1	Standby mode release by NMI pin input disabled

INTM	Standby mode release control via maskable interrupt request signal
0	Standby mode release by maskable interrupt request signal enabled
1	Standby mode release by maskable interrupt request signal disabled

STP	Standby mode ^{Note} setting
0	Normal operation mode
1	Standby mode

Note Standby mode set by STP bit: IDLE1, IDLE2, STOP, or sub-IDLE mode

- Cautions**
1. Before setting one of the standby modes (excluding the HALT mode), specify the mode by using the PSMR.PSM1 and PSMR.PSM0 bits and then set the STP bit.
 2. The settings of the NMI1M, NMI0M, and INTM bits are invalid when HALT mode is released.
 3. If the NMI1M, NMI0M, or INTM bit is set to 1 at the same time the STP bit is set to 1, the setting of NMI1M, NMI0M, or INTM bit becomes invalid. If there is an unmasked interrupt request signal being held pending when the IDLE1/IDLE2/STOP mode is set, set the bit corresponding to the interrupt request signal (NMI1M, NMI0M, or INTM) to 1, and then set the STP bit to 1.

(2) 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 sets this register to 00H.

After reset: 00H R/W Address: FFFFF820H

	7	6	5	4	3	2	<1>	<0>
PSMR	0	0	0	0	0	0	PSM1	PSM0

PSM1	PSM0	Specification of operation in software standby mode
0	0	IDLE1, sub-IDLE modes
0	1	STOP mode
1	0	IDLE2, sub-IDLE modes
1	1	STOP mode

Cautions 1. Be sure to clear bits 2 to 7 to “0”.

2. The PSM0 and PSM1 bits are valid only when the PSC.STP bit is 1.

Remark	IDLE1:	In this mode, all operations except the oscillator operation and some other circuits (flash memory and PLL) are stopped. After the IDLE1 mode is released, the normal operation mode is restored without needing to secure the oscillation stabilization time, like the HALT mode.
	IDLE2:	In this mode, all operations except the oscillator operation are stopped. After the IDLE2 mode is released, the normal operation mode is restored following the lapse of the setup time specified by the OSTS register (flash memory and PLL).
	STOP:	In this mode, all operations except the subclock oscillator operation are stopped. After the STOP mode is released, the normal operation mode is restored following the lapse of the oscillation stabilization time specified by the OSTS register.
	Sub-IDLE:	In this mode, all other operations are halted except for the oscillator. After the IDLE mode has been released by the interrupt request signal, the subclock operation mode will be restored after 12 cycles of the subclock have been secured.

(3) Oscillation stabilization time select register (OSTS)

The wait time until the oscillation stabilizes after the STOP mode is released or the setup time until the internal flash memory stabilizes after the IDLE2 mode is released is controlled by the OSTS register.

The OSTS register can be read or written 8-bit units.

Reset sets this register to 06H.

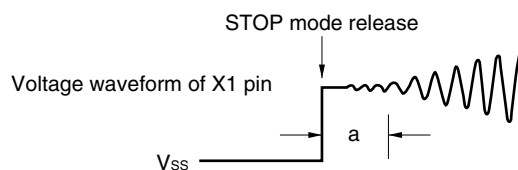
After reset: 06H R/W Address: FFFFF6C0H

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

OSTS2	OSTS1	OSTS0	Selection of oscillation stabilization time/setup time ^{Note}	fx	
				4 MHz	5 MHz
				0	0
0	0	1	$2^{11}/f_x$	0.512 ms	0.410 ms
0	1	0	$2^{12}/f_x$	1.024 ms	0.819 ms
0	1	1	$2^{13}/f_x$	2.048 ms	1.638 ms
1	0	0	$2^{14}/f_x$	4.096 ms	3.277 ms
1	0	1	$2^{15}/f_x$	8.192 ms	6.554 ms
1	1	0	$2^{16}/f_x$	16.38 ms	13.107 ms
1	1	1	Setting prohibited		

Note The oscillation stabilization time and setup time are required when the STOP mode and IDLE2 mode are released, respectively.

Cautions 1. The wait time following the release of STOP mode does not include the time until the clock oscillation starts (“a” in the figure below, regardless of whether the STOP mode is released by a reset or an interrupt).



2. Be sure to clear bits 3 to 7 to “0”.

3. The oscillation stabilization time following reset release differs depending on the option byte. For details, see CHAPTER 27 OPTION BYTE.

Remark fx = Main clock oscillation frequency

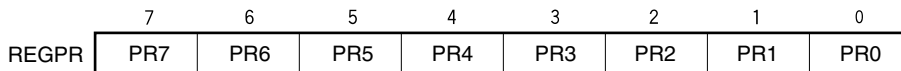
(4) Regulator protection register (REGPR)

The REGPR register is used to protect the regulator output voltage level control register 0 (REGOVL0) so that illegal data is not written to REGOVL0. Data cannot be written to the REGOVL0 register unless enabling data (C9H) is written to the REGPR register. Only two types of data, C9H (enabling data) and 00H (protection data), can be written to the REGPR register. Writing any other value is prohibited. (If a value other than C9H or 00H is written to the REGPR register, the written value is set to prohibit a write access to the REGOVL0 register, but the operation is not guaranteed.)

This register can be read or written only in 8-bit units (accessing it in 1-bit units is prohibited).

Reset sets this register to 00H (protection data status).

After reset: 00H R/W Address: FFFFF331H



• **Protection data status: REGPR = 00H**

In this status, the REGOVL0 register is protected from an illegal write access. In the protection data status, a value is not written to the REGOVL0 register even if an attempt is made to write it, and the REGOVL0 register holds the previous value.

Be sure to set REGPR to 00H, except when changing the value of the REGOVL0 register, in order to avoid unexpected malfunction.

• **Enabling data status: REGPR = C9H**

In this status, a write access to the REGOVL0 register is enabled.

- Transition from normal mode → low-voltage STOP mode
See **21.6.1 Setting and operation status.**
- Transition of subclock operation mode → low-voltage subclock operation mode
See **21.7.1 Setting and operation status.**
- Transition of subclock operation mode → low-voltage sub-IDLE mode
See **21.8.1 Setting and operation status.**

(5) Regulator output voltage level control register 0 (REGOVL0)

This register is used to select the low-voltage STOP mode, low-voltage subclock operation mode, or low-voltage sub-IDLE mode. The power consumption can be reduced by lowering the output voltage of the regulator.

This register can be read or written only in 8-bit units (accessing it in 1-bit units is prohibited).

Reset sets this register to 00H.

This register must be always written in pairs with the regulator protection register (REGPR).

After reset: 00H R/W Address: FFFFF332H

	7	6	5	4	3	2	1	0
REGOVL0	0	0	0	0	0	0	SUBMD	STPMD

SUBMD	Output mode selection of regulator in subclock operation mode/sub-IDLE mode
0	Subclock operation mode/sub-IDLE mode
1	Low-voltage subclock operation mode/low-voltage sub-IDLE mode

STPMD	Output mode selection of regulator in STOP mode
0	STOP mode
1	Low-voltage STOP mode

- **Write operation of REGOVL0 register**

Writing the REGOVL0 register is enabled only when C9H is written to the REGPR register (see 21.2 (4) **Regulator protection register (REGPR)**).

This register can be set only to 00H, 01H, and 02H.

Setting 03H is prohibited. If 03H is set, the operation is not guaranteed.

- **Read operation of REGOVL0 register**

The default value of the REGOVL0 register is 00H. After a value has been written to this register in the correct procedure^{Note}, the written value is read. The procedure for reading this register is not restricted.

Note • Transition from normal mode → low-voltage STOP mode

See 21.6.1 **Setting and operation status.**

- Transition of subclock operation mode → low-voltage subclock operation mode

See 21.7.1 **Setting and operation status.**

- Transition of subclock operation mode → low-voltage sub-IDLE mode

See 21.8.1 **Setting and operation status.**

Caution Be sure to stop the main clock and PLL when setting the low-voltage subclock mode and low-voltage sub-IDLE mode.

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 oscillator continues operating. Only clock supply to the CPU is stopped; clock supply to the other on-chip peripheral functions continues.

As a result, program execution is stopped, and the internal RAM retains the contents before the HALT mode was set. The on-chip peripheral functions that are independent of instruction processing by the CPU continue operating.

Table 21-3 shows the operating status in the HALT mode.

The average current consumption of the system can be reduced by using the HALT mode in combination with the normal operation mode.

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 status shifts to HALT mode, but the HALT mode is then released immediately by the pending interrupt request.**

21.3.2 Releasing HALT mode

The HALT mode is released by a non-maskable interrupt request signal (NMI pin input, INTWDT2 signal), unmasked external interrupt request signal (INTP0 to INTP7 pin input), unmasked internal interrupt request signal from a peripheral function operable in the HALT mode, or reset signal (reset by RESET pin input, $\overline{\text{WDT2RES}}$ signal, low-voltage detector (LVI), or clock monitor (CLM)).

After the HALT mode has been released, the normal operation mode is restored.

(1) Releasing HALT mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The HALT mode is released by a non-maskable interrupt request signal or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request. If the HALT mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

Table 21-2. Releasing HALT Mode and Operation After Release

Release Source	Interrupt Acknowledgment Status	Status After Release	Operation After Release
Reset	Disabled (DI)	–	Normal reset operation
	Enabled (EI)		
Non-maskable interrupt request signal (excluding multiple interrupts)	Disabled (DI)	–	The interrupt request is acknowledged when the HALT mode is released.
	Enabled (EI)		
Maskable interrupt request signal	Disabled (DI)	–	The HALT mode is released but the interrupt request that is the release source is not acknowledged. The interrupt request itself is retained. The processing that was being executed before shifting to the HALT mode is executed.
	Enabled (EI)	• An interrupt request with a priority higher than that of the release source is being serviced.	The HALT mode is released but the interrupt request that is the release source is not acknowledged. The interrupt request itself is retained. The interrupt that was being serviced before shifting to the HALT mode is serviced.
		• An interrupt request with a priority lower than that of the release source is being serviced.	The interrupt request is acknowledged when the HALT mode is released.

Table 21-3. Operating Status in HALT Mode

Setting of HALT Mode Item		Operating Status	
		When Subclock Is Not Used	When Subclock Is Used
LVI		Operable	
Main clock oscillator		Oscillates	
Subclock oscillator		–	Oscillates
Internal oscillator		Oscillation enabled	
PLL		Operable	
CPU		Stops operation	
DMA		Operable	
Interrupt controller		Operable	
Timer P (TMP0 to TMP5)		Operable	
Timer Q (TMQ0)		Operable	
Timer M (TMM0)		Operable when a clock other than f_{XT} is selected as the count clock	Operable
Watch timer/RTC		Operable when f_x (divided BRG) is selected as the count clock	Operable
Watchdog timer 2		Operable when a clock other than f_{XT} is selected as the count clock	Operable
Serial interface	CSIB0 to CSIB4	Operable	
	I ² C00 to I ² C02	Operable	
	UARTA0 to UARTA2	Operable	
A/D converter		Operable	
D/A converter		Operable	
Real-time output function (RTO)		Operable	
Key interrupt function (KR)		Operable	
CRC operation circuit		Operable (in the status in which data is not input to the CRCIN register to stop the CPU)	
Port function		Retains status before HALT mode was set	
CPU register set		Retains status before HALT mode was set	
Internal RAM			

21.4 IDLE1 Mode

21.4.1 Setting and operation status

The IDLE1 mode is set by clearing the PSMR.PSM1 and PSMR.PSM0 bits to 00 and setting the PSC.STP bit to 1 in the normal operation mode.

In the IDLE1 mode, the clock oscillator, PLL, and flash memory continue operating 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 IDLE1 mode was set are retained. The CPU and other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate with the subclock or an external clock continue operating.

Table 21-5 shows the operating status in the IDLE1 mode.

The IDLE1 mode can reduce the power consumption more than the HALT mode because it stops the operation of the on-chip peripheral functions. The main clock oscillator does not stop, so the normal operation mode can be restored without waiting for the oscillation stabilization time after the IDLE1 mode has been released, in the same manner as when the HALT mode is released.

- Cautions 1. Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the IDLE1 mode.**
- 2. If the IDLE1 mode is set while an unmasked interrupt request signal is being held pending, the CPU does not shift to the IDLE1 mode but executes the next instruction.**

21.4.2 Releasing IDLE1 mode

The IDLE1 mode is released by a non-maskable interrupt request signal (NMI pin input, INTWDT2 signal), unmasked external interrupt request signal (INTP0 to INTP7 pin input), unmasked internal interrupt request signal from a peripheral function operable in the IDLE1 mode, or reset signal (reset by RESET pin input, WDT2RES signal, low-voltage detector (LVI), or clock monitor (CLM)).

After the IDLE1 mode has been released, the normal operation mode is restored.

(1) Releasing IDLE mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The IDLE 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. If the IDLE1 mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is processed as follows.

Table 21-4. Operation After Releasing IDLE1 Mode by Interrupt Request Signal

Release Source	Interrupt Acknowledgment Status	Status After Release	Operation After Release
Reset	Disabled (DI)	–	Normal reset operation
	Enabled (EI)		
Non-maskable interrupt request signal (excluding multiple interrupts)	Disabled (DI)	–	The interrupt request is acknowledged when the IDLE1 mode is released.
	Enabled (EI)		
Maskable interrupt request signal	Disabled (DI)	–	The IDLE1 mode is released but the interrupt request that is the release source is not acknowledged. The interrupt request itself is retained. The processing that was being executed before shifting to the IDLE1 mode is executed.
	Enabled (EI)	• An interrupt request with a priority higher than that of the release source is being serviced.	The IDLE1 mode is released but the interrupt request that is the release source is not acknowledged. The interrupt request itself is retained. The interrupt that was being serviced before shifting to the IDLE1 mode is serviced.
		• An interrupt request with a priority lower than that of the release source is being serviced.	The interrupt request is acknowledged when the IDLE1 mode is released.

Caution An interrupt request signal that is disabled by setting the PSC.NMI2M, PSC.NMI0M, and PSC.INTM bits to 1 (interrupt disabled) is invalid and cannot release the IDLE1 mode.

Table 21-5. Operating Status in IDLE1 Mode

Setting of IDLE1 Mode Item		Operating Status	
		When Subclock Is Not Used	When Subclock Is Used
LVI		Operable	
Main clock oscillator		Oscillates	
Subclock oscillator		–	Oscillates
Internal oscillator		Oscillation enabled	
PLL		Operable	
CPU		Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation (but standby mode release enabled)	
Timer P (TMP0 to TMP5)		Stops operation	
Timer Q (TMQ0)		Stops operation	
Timer M (TMM0)		Operable when $f_{R/8}$ is selected as the count clock	Operable when $f_{R/8}$ or f_{XT} is selected as the count clock
Watch timer/RTC		Operable when f_x (divided BRG) is selected as the count clock	Operable
Watchdog timer 2		Operable when $f_{R/8}$ is selected as the count clock	Operable when $f_{R/8}$ or f_{XT} is selected as the count clock
Serial interface	CSIB0 to CSIB4	Operable when the $\overline{\text{SCKBn}}$ input clock is selected as the count clock ($n = 0$ to 4)	
	I ² C00 to I ² C02	Stops operation	
	UARTA0 to UARTA2	Stops operation (but UARTA0 is operable when the ASCKA0 input clock is selected)	
A/D converter		Holds operation (conversion result held) ^{Note}	
D/A converter		Holds operation (output held) ^{Note}	
Real-time output function (RTO)		Stops operation (output held)	
Key interrupt function (KR)		Operable	
CRC operation circuit		Stops operation	
Port function		Retains status before IDLE1 mode was set	
CPU register set		Retains status before IDLE1 mode was set	
Internal RAM			

Note To realize low power consumption, stop the A/D converter and D/A converter before shifting to the IDLE1 mode.

21.5 IDLE2 Mode

21.5.1 Setting and operation status

The IDLE2 mode is set by setting the PSMR.PSM1 and PSMR.PSM0 bits to 10 and setting the PSC.STP bit to 1 in the normal operation mode.

In the IDLE2 mode, the clock oscillator continues operation but clock supply to the CPU, PLL, flash memory, and other on-chip peripheral functions stops.

As a result, program execution stops and the contents of the internal RAM before the IDLE2 mode was set are retained. The CPU, PLL, and other on-chip peripheral functions stop operating. However, the on-chip peripheral functions that can operate with the subclock or an external clock continue operating.

Table 21-7 shows the operating status in the IDLE2 mode.

The IDLE2 mode can reduce the power consumption more than the IDLE1 mode because it stops the operations of the on-chip peripheral functions, PLL, and flash memory. However, because the PLL and flash memory are stopped, a setup time for the PLL and flash memory is required when IDLE2 mode is released.

- Cautions**
- 1. Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the IDLE2 mode.**
 - 2. If the IDLE2 mode is set while an unmasked interrupt request signal is being held pending, the CPU does not shift to the IDLE2 mode but executes the next instruction.**

21.5.2 Releasing IDLE2 mode

The IDLE2 mode is released by a non-maskable interrupt request signal (NMI pin input, INTWDT2 signal), unmasked external interrupt request signal (INTP0 to INTP7 pin input), unmasked internal interrupt request signal from the peripheral functions operable in the IDLE2 mode, or reset signal (reset by RESET pin input, WDT2RES signal, low-voltage detector (LVI), or clock monitor (CLM)). The PLL returns to the operating status it was in before the IDLE2 mode was set.

After the IDLE2 mode has been released, the normal operation mode is restored.

(1) Releasing IDLE2 mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The IDLE 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. If the IDLE2 mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is processed as follows.

Table 21-6. Operation After Releasing IDLE2 Mode by Interrupt Request Signal

Release Source	Interrupt Acknowledgment Status	Status After Release	Operation After Release
Reset	Disabled (DI)	–	Normal reset operation
	Enabled (EI)		
Non-maskable interrupt request signal (excluding multiple interrupts)	Disabled (DI)	–	The IDLE2 mode is released, and after securing the specified setup time, the interrupt request is acknowledged.
	Enabled (EI)		
Maskable interrupt request signal	Disabled (DI)	–	The IDLE2 mode is released but the interrupt request that is the release source is not acknowledged. The interrupt request itself is retained. After securing the specified setup time, the interrupt that was being serviced before shifting to the IDLE2 mode is serviced.
	Enabled (EI)	• An interrupt request with a priority higher than that of the release source is being serviced.	The IDLE2 mode is released but the interrupt request that is the release source, is not acknowledged. The interrupt request itself is retained. After securing the specified setup time, the processing that was being executed before shifting to the IDLE2 mode is executed.
		• An interrupt request with a priority lower than that of the release source is being serviced.	The IDLE2 mode is released, and after securing the specified setup time, the interrupt request is acknowledged.

Caution An interrupt request signal that is disabled by setting the PSC.NMI2M, PSC.NMI0M, and PSC.INTM bits to 1 (interrupt disabled) is invalid and cannot release the IDLE2 mode.

Table 21-7. Operating Status in IDLE2 Mode

Setting of IDLE2 Mode		Operating Status	
		When Subclock Is Not Used	When Subclock Is Used
LVI		Operable	
Main clock oscillator		Oscillates	
Subclock oscillator		–	Oscillates
Internal oscillator		Oscillation enabled	
PLL		Stops operation	
CPU		Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation (but standby mode release is possible)	
Timer P (TMP0 to TMP5)		Stops operation	
Timer Q (TMQ0)		Stops operation	
Timer M (TMM0)		Operable when $f_{R/8}$ is selected as the count clock	Operable when $f_{R/8}$ or f_{XT} is selected as the count clock
Watch timer/RTC		Operable when f_x (divided BRG) is selected as the count clock	Operable
Watchdog timer 2		Operable when $f_{R/8}$ is selected as the count clock	Operable when $f_{R/8}$ or f_{XT} is selected as the count clock
Serial interface	CSIB0 to CSIB4	Operable when the SCKBn input clock is selected as the count clock (n = 0 to 4)	
	I ² C00 to I ² C02	Stops operation	
	UARTA0 to UARTA2	Stops operation (but UARTA0 is operable when the ASCKA0 input clock is selected)	
A/D converter		Holds operation (conversion result held) ^{Note}	
D/A converter		Holds operation (output held) ^{Note}	
Real-time output function (RTO)		Stops operation (output held)	
Key interrupt function (KR)		Operable	
CRC operation circuit		Stops operation	
Port function		Retains status before IDLE2 mode was set	
CPU register set		Retains status before IDLE2 mode was set	
Internal RAM			

Note To realize low power consumption, stop the A/D and D/A converters before shifting to the IDLE2 mode.

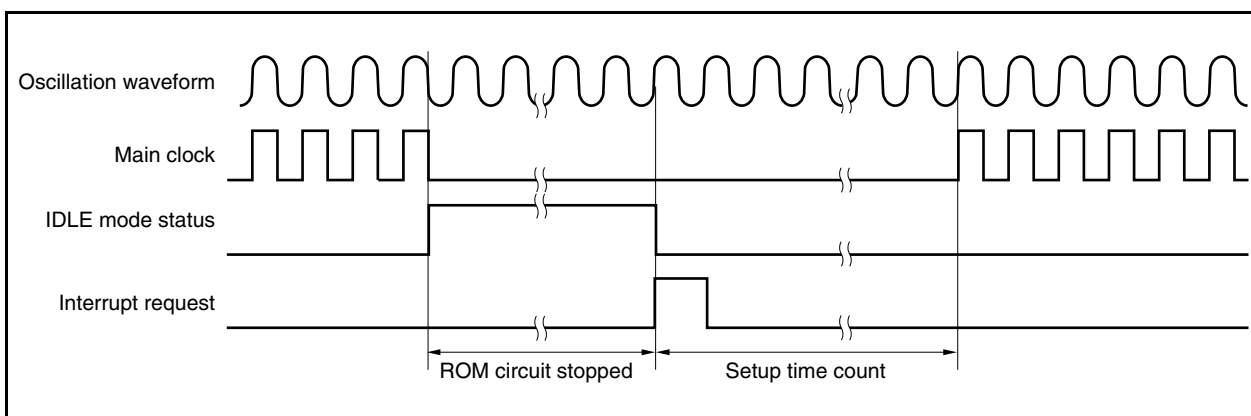
21.5.3 Securing setup time when releasing IDLE2 mode

Setting the IDLE2 mode stops the operation of blocks other than the main clock oscillator, so the setup time specified by the OSTS register for the PLL or the flash memory is automatically secured after the IDLE2 mode is released.

(1) Releasing IDLE2 mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

Secure the specified setup time by setting the OSTS register.

When the releasing source is generated, the dedicated internal timer starts counting according to the OSTS register setting. When it overflows, the normal operation mode is restored.



(2) Release by reset ($\overline{\text{RESET}}$ pin input, WDT2RES generation)

This operation is the same as that of a normal reset.

The oscillation stabilization time differs depending on the option byte. For details, see **CHAPTER 27 OPTION BYTE**.

21.6 STOP Mode/Low-Voltage STOP Mode

21.6.1 Setting and operation status

The STOP mode is set by setting the PSMR.PSM1 and PSMR.PSM0 bits to 01 or 11 and setting the PSC.STP bit to 1 in the normal operation mode. The low-voltage STOP mode is set by setting the PSMR.PSM1 and PSMR.PSM0 bits to 01 or 11 and setting the PSC.STP bit to 1 after setting the REGOVL0 register to 01H in normal operation mode.

In the STOP mode, the subclock oscillator continues operating but the main clock oscillator stops. Clock supply to the CPU and the on-chip peripheral functions is stopped.

As a result, program execution stops, and the contents of the internal RAM before the STOP mode was set are retained. Clock supply to the CPU and the on-chip peripheral functions is stopped, but the subclock oscillator continues operating. In the STOP mode, CSIBn and UARTA0, which can operate on the external clock, also continue operating. In the low-voltage STOP mode, however, stop supplying the external clock to CSIBn and UARTA0 (n = 0 to 4) because these blocks cannot continue operating.

Table 21-8 shows the operating status in the STOP mode and Table 21-9 shows the operating status in the low-voltage STOP mode.

Because the STOP mode stops operation of the main clock oscillator, it reduces the power consumption to a level lower than the IDLE2 mode. If the subclock oscillator, internal oscillator, low-voltage detector (LVI), and external clock are not used, the power consumption can be minimized with only leakage current flowing.

The power consumption decreases further in the low-voltage STOP mode because the voltage of the regulator is lowered.

Be sure to set the low-voltage STOP mode using the following procedure.

(1) Procedure for switching from normal mode to low-voltage STOP mode

Specify the following settings in the normal operation mode (while the main clock is operating). In addition, set up the OSTS register as necessary.

<1> Stop the functions whose operation is specified as stopped in **Table 21-9 Operating Status in Low-Voltage STOP Mode**.

Be especially sure to stop the following, because they are signals from external sources.

- Stop the \overline{SCKBn} input clock when the \overline{SCKBn} input clock to CSIBn is selected (n = 0 to 4).
- Stop the ASCKA0 input clock when the ASCKA0 input clock to UARTA0 is selected.

<2> Disable DMA.

<3> • Disable maskable interrupts by using the DI instruction.

- Disable the NMI interrupt (INTF02 = 0, INTR02 = 0).

- Create a status in which the INTWDT2 signal is not generated (create a status in which the INTWDT2 signal is not generated immediately after watchdog timer 2 has been cleared).

<4> Write C9H (enabling data) to the REGPR register.

<5> Write 01H to the REGOVL0 register.

At this time, the output voltage of the regulator is at the normal level.

<6> Write 00H (protection data) to the REGPR register.

<7> As necessary, enable maskable interrupts, the NMI interrupt, or the INTWDT2 interrupt by using the EI instruction (restore the settings in <2> and <3> above).

<8> Set the STOP mode.

$$\left(\begin{array}{l} \text{PSMR.PSM1, PSMR.PSM0 bits} = 01 \text{ or } 11 \\ \text{PSC.STP bit} = 1 \end{array} \right)$$

In the STOP mode, the output voltage of the regulator drops, decreasing the current consumption to an extremely low level.

Be sure to observe the above sequence.

Note, however, that step <7> may be performed at any time as long as it is done after step <6>. (The setting in step <7> may be made without problem, even after the low-voltage STOP mode has been released.)

Cautions 1. Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the STOP mode/low-voltage STOP mode.

2. If the STOP mode/low-voltage STOP mode is set while an unmasked interrupt request signal is being held pending, the CPU does not shift to the STOP mode/low-voltage STOP mode but executes the next instruction.

Table 21-8. Operating Status in STOP Mode

Setting of STOP Mode		Operating Status	
		When Subclock Is Not Used	When Subclock Is Used
LVI		Operable	
Main clock oscillator		Stops oscillation	
Subclock oscillator		–	Oscillates
Internal oscillator		Oscillation enabled	
PLL		Stops operation	
CPU		Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation (but standby mode release is possible)	
Timer P (TMP0 to TMP5)		Stops operation	
Timer Q (TMQ0)		Stops operation	
Timer M (TMM0)		Operable when $f_{R/8}$ is selected as the count clock	Operable when $f_{R/8}$ or f_{XT} is selected as the count clock
Watch timer/RTC		Stops operation	Operable when f_{XT} is selected as the count clock
Watchdog timer 2		Operable when $f_{R/8}$ is selected as the count clock	Operable when $f_{R/8}$ or f_{XT} is selected as the count clock
Serial interface	CSIB0 to CSIB4	Operable when the SCKBn input clock is selected as the count clock (n = 0 to 4)	
	I ² C00 to I ² C02	Stops operation	
	UARTA0 to UARTA2	Stops operation (but UARTA0 is operable when the ASCKA0 input clock is selected)	
A/D converter		Stops operation (conversion result undefined) ^{Notes 1, 2}	
D/A converter		Stops operation ^{Notes 3, 4} (high impedance is output)	
Real-time output function (RTO)		Stops operation (output held)	
Key interrupt function (KR)		Operable	
CRC operation circuit		Stops operation	
Port function		Retains status before STOP mode was set	
CPU register set		Retains status before STOP mode was set	
Internal RAM			

Notes1. If the STOP mode is set while the A/D converter is operating, the A/D converter is automatically stopped and starts operating again after the STOP mode is released. However, in that case, the first A/D conversion result after the STOP mode is released is invalid. The A/D conversion result before the STOP mode is set is also invalid.

2. Even if the STOP mode is set while the A/D converter is operating, the power consumption is reduced equivalently to when the A/D converter is stopped before the STOP mode is set.
3. If the STOP mode is set while the D/A converter is operating, the D/A converter is automatically stopped and the pin status becomes high impedance. After the STOP mode is released, D/A conversion resumes, the setting time elapses, and the status returns to the output level before the STOP mode was set.
4. Even if the STOP mode is set while the D/A converter is operating, the power consumption is reduced equivalently to when the D/A converter is stopped before the STOP mode is set.

Table 21-9. Operating Status in Low-Voltage STOP Mode

Setting of Low-Voltage STOP Mode		Operating Status	
		When Subclock Is Not Used	When Subclock Is Used
LVI		Operable	
Main clock oscillator		Stops oscillation	
Subclock oscillator		–	Oscillates
Internal oscillator		Oscillation enabled	
PLL		Stops operation	
CPU		Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation (but standby mode release is possible)	
Timer P (TMP0 to TMP5)		Stops operation	
Timer Q (TMQ0)		Stops operation	
Timer M (TMM0)		Operable when $f_{R/8}$ is selected as the count clock	Operable when $f_{R/8}$ or f_{XT} is selected as the count clock
Watch timer/RTC		Stops operation	Operable when f_{XT} is selected as the count clock
Watchdog timer 2		Operable when $f_{R/8}$ is selected as the count clock	Operable when $f_{R/8}$ or f_{XT} is selected as the count clock
Serial interface	CSIB0 to CSIB4	Stops operation (When the SCKBn input clock is selected as the count clock, be sure to stop the SCKBn input clock (n = 0 to 4).)	
	I ² C00 to I ² C02	Stops operation	
	UARTA0 to UARTA2	Stops operation (When the ASCKA0 input clock to UARTA0 is selected, be sure to stop the ASCKA0 input clock.)	
A/D converter		Stops operation (conversion result undefined) ^{Notes 1, 2}	
D/A converter		Stops operation ^{Notes 3, 4} (high impedance is output)	
Real-time output function (RTO)		Stops operation (output held)	
Key interrupt function (KR)		Operable	
CRC operation circuit		Stops operation	
Port function		Retains status before low-voltage STOP mode was set	
CPU register set		Retains status before low-voltage STOP mode was set	
Internal RAM			

- Notes1.** If the low-voltage STOP mode is set while the A/D converter is operating, the A/D converter is automatically stopped and starts operating again after the low-voltage STOP mode is released. However, in that case, the A/D conversion results after the low-voltage STOP mode is released are invalid. All the A/D conversion results before the low-voltage STOP mode is set are invalid.
- Even if the low-voltage STOP mode is set while the A/D converter is operating, the power consumption is reduced equivalently to when the A/D converter is stopped before the low-voltage STOP mode is set.
 - If the low-voltage STOP mode is set while the D/A converter is operating, the D/A converter is automatically stopped. After the low-voltage STOP mode is released, D/A conversion resumes, the setting time elapses, and the status returns to the output level before the low-voltage STOP mode was set.
 - Even if the low-voltage STOP mode is set while the D/A converter is operating, the power consumption is reduced equivalently to when the D/A converter is stopped before the low-voltage STOP mode is set.

21.6.2 Releasing STOP mode/low-voltage STOP mode

The STOP mode/low-voltage STOP mode is released by a non-maskable interrupt request signal (NMI pin input, INTWDT2 signal), unmasked external interrupt request signal (INTP0 to INTP7 pin input), unmasked internal interrupt request signal from the peripheral functions operable in the STOP mode/low-voltage STOP mode, or reset signal (reset by RESET pin input, WDT2RES signal, or low-voltage detector (LVI)).

After the STOP mode/low-voltage STOP mode has been released, the normal operation mode is restored after the oscillation stabilization time has been secured.

For re-set after releasing the low-voltage STOP mode, see **21.6.3 Re-setting after release of low-voltage STOP mode**.

(1) Releasing STOP mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The STOP 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. If the STOP mode/low-voltage STOP mode is set in an interrupt servicing routine, however, an interrupt request that is issued later is serviced as follows.

Table 21-10. Operation After Releasing STOP Mode/Low-Voltage STOP Mode by Interrupt Request Signal

Release Source	Interrupt Acknowledgment Status	Status After Release	Operation After Release
Reset	Disabled (DI)	–	Normal reset operation
	Enabled (EI)		
Non-maskable interrupt request signal (excluding multiple interrupts)	Disabled (DI)	–	The STOP mode/low-voltage STOP mode is released, and after securing the oscillation stabilization time, the interrupt request is acknowledged.
	Enabled (EI)		
Maskable interrupt request signal	Disabled (DI)	–	The STOP mode/low-voltage STOP mode is released but the interrupt request that is the release source is not acknowledged. The interrupt request itself is retained. After securing the oscillation stabilization time, the processing that was being executed before shifting to the STOP mode/low-voltage STOP mode is executed.
	Enabled (EI)	• An interrupt request with a priority higher than that of the release source is being serviced.	The STOP mode/low-voltage STOP mode is released but the interrupt request that is the release source is not acknowledged. The interrupt request itself is retained. After securing the oscillation stabilization time, the interrupt servicing that was being executed before shifting to the STOP mode/low-voltage STOP mode is executed.
		• An interrupt request with a priority lower than that of the release source is being serviced.	The STOP mode/low-voltage STOP mode is released, and after securing the oscillation stabilization time, the interrupt request is acknowledged.

Caution An interrupt request signal that is disabled by setting the PSC.NMI2M, PSC.NMI0M, and PSC.INTM bits to 1 (interrupt disabled) is invalid and cannot release the STOP mode/low-voltage STOP mode.

21.6.3 Re-setting after release of low-voltage STOP mode

(1) If low-voltage STOP mode is released by interrupt

The status after the low-voltage STOP mode has been released is as follows.

- Regulator: Automatically returns to the normal level.

The oscillation stabilization time specified by the OSTS register is secured.

- REGOVL0 register = 01H (low-voltage STOP mode): Value described in 21.6.1 (1) <5> is retained.
- REGPR register = 00H (protection data): Value described in 21.6.1 (1) <6> is retained.

(a) To continuously use the REGOVL0 register = 01H (low-voltage STOP mode), the other registers do not have to be set again.

(b) Follow this procedure when returning the REGOVL0 register = 00H.

<1> Disable the DMA.

<2> • Disable the maskable interrupt by the DI instruction.

- Disable the NMI interrupt (INTF02 = 0, INTR02 = 0).

- Create a status in which the INTWDT2 signal is not generated (stop watchdog timer 2 or set a mode other than the INTWDT2 mode. Create a status in which the INTWDT2 signal is not generated immediately after watchdog timer 2 has been cleared).

<3> Write C9H (enabling data) to the REGPR register.

<4> Write 00H to the REGOVL0 register.

<5> Write 00H (protection data) to the REGPR register.

<6> As necessary, enable the maskable interrupt, NMI interrupt, or INTWDT2 interrupt by enabling DMA or the EI instruction (restore the settings <1> and <2> above).

Be sure to observe the above sequence.

(2) If low-voltage STOP mode is released by reset

The CPU shifts to the normal operation mode immediately after the reset ends, and the REGOVL0 register is initialized to 00H and the REGPR register to 00H (protection data). Be sure to secure the oscillation stabilization time that follows immediately after a reset ends by setting the option byte. For details, see **CHAPTER 27 OPTION BYTE**.

Caution Interrupt requests that are set to 1 (disabled) by the PSC.NMI1M, PSC.NMI0M, and PSC.INTM bits are invalid and cannot release the low-voltage STOP mode.

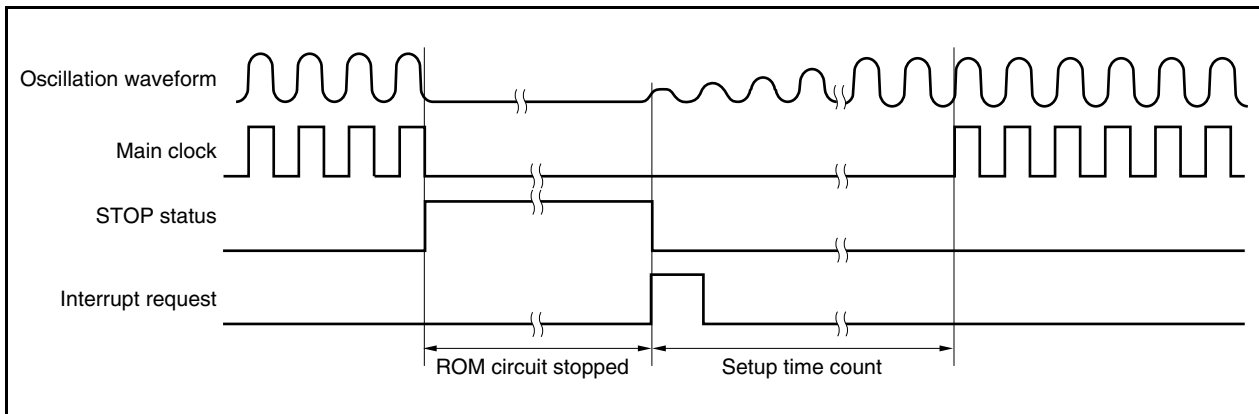
21.6.4 Securing oscillation stabilization time when releasing STOP mode

Secure the oscillation stabilization time for the main clock oscillator after releasing the STOP mode because the operation of the main clock oscillator stops after STOP mode is set.

(1) Releasing STOP mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

Secure the oscillation stabilization time by setting the OSTS register.

When the releasing source is generated, the dedicated internal timer starts counting according to the OSTS register setting. When it overflows, the normal operation mode is restored.



(2) Release by reset

This operation is the same as that of a normal reset.

The oscillation stabilization time differs depending on the option byte. For details, see **CHAPTER 27 OPTION BYTE**.

21.7 Subclock Operation Mode/Low-Voltage Subclock Operation Mode

21.7.1 Setting and operation status

The subclock operation mode is set by setting the PCC.CK3 bit to 1 in the normal operation mode. The low-voltage subclock operation mode is set by setting the REGOVL0 register to 02H in the subclock operation mode.

When the subclock operation mode is set, the internal system clock is changed from the main clock to the subclock. Check whether the clock has been switched by using the PCC.CLS bit.

When the PCC.MCK bit is set to 1, the operation of the main clock oscillator is stopped. As a result, the system operates only on the subclock.

In the subclock operation mode, power consumption can be reduced to a level lower than in the normal operation mode because the subclock is used as the internal system clock. In addition, power consumption can be further reduced to the level of the STOP mode by stopping the operation of the main clock oscillator. Power consumption decreases further in the low-voltage subclock operation mode because the voltage of the regulator is lowered.

When the main clock oscillator is stopped in the subclock operation mode, CSIBn and UARTA0, which can operate on the external clock, also continue operating. In the low-voltage subclock operation mode, however, stop supplying the external clock input to CSIBn and UARTA0 because these blocks cannot continue operating (n = 0 to 4).

Cautions 1. When manipulating the CK3 bit, do not change the set values of the PCC.CK2 to PCC.CK0 bits (using a bit manipulation instruction to manipulate the bit is recommended). For details of the PCC register, see 5.3 (1) Processor clock control register (PCC).

2. If the following conditions are not satisfied, change the CK2 to CK0 bits so that the conditions are satisfied and set the subclock operation mode.

$$\text{Internal system clock (f}_{\text{CLK}}) > \text{Subclock (f}_{\text{XT}} = 32.768 \text{ kHz)} \times 4$$

Remark Internal system clock (f_{CLK}): Clock generated from main clock (f_{xx}) in accordance with the settings of the CK2 to CK0 bits

Be sure to set the low-voltage subclock operation mode using the following procedure.

(1) Procedure for switching from subclock operation mode to low-voltage subclock operation mode

Make the following settings in the subclock operation mode.

<1> Stop the main clock and PLL.

<2> Stop the functions whose operation is specified as stopped in **Table 21-14 Operating Status in Low-Voltage Sub-IDLE Mode**.

Be especially sure to stop the following, because they are signals from external sources.

- Stop the \overline{SCKBn} input clock when the \overline{SCKBn} input clock to CSIBn is selected (n = 0 to 4).
- Stop the ASCKA0 input clock when the ASCKA0 input clock to UAR0A0 is selected.

<3> Disable DMA (if DMA is enabled).

<4> • Disable maskable interrupts by using the DI instruction.

- Disable the NMI interrupt (INTF02 = 0, INTR02 = 0).

- Create a status in which the INTWDT2 signal is not generated (create a status in which the INTWDT2 signal is not generated immediately after watchdog timer 2 has been cleared).

<5> Write C9H (enabling data) to the REGPR register.

<6> Write 02H to the REGOVL0 register.

At this time, the output voltage of the regulator is at the low level, decreasing power consumption to an extremely low level.

<7> Write 00H (protection data) to the REGPR register.

<8> As necessary, enable maskable interrupts, the NMI interrupt, or the INTWDT2 interrupt by using the EI instruction (restore the setting in <4> above).

Be sure to observe the above sequence.

For the setting of the subclock operation mode, see **21.7.1 Setting and operation status**.

Table 21-11 shows the operating status in the subclock operation mode and Table 21-12 shows the operating status in the low-voltage subclock operation mode.

Table 21-11. Operating Status in Subclock Operation Mode

Setting of Subclock Operation Mode Item		Operating Status	
		When Main Clock Is Oscillating	When Main Clock Is Stopped
LVI		Operable	
Subclock oscillator		Oscillates	
Internal oscillator		Oscillation enabled	
PLL		Operable	Stops operation ^{Note1}
CPU		Operable	
DMA		Operable	
Interrupt controller		Operable	
Timer P (TMP0 to TMP5)		Operable	Stops operation
Timer Q (TMQ0)		Operable	Stops operation
Timer M (TMM0)		Operable	Operable when f_{R8} or f_{XT} is selected as the count clock
Watch timer/RTC		Operable	Operable when f_{XT} is selected as the count clock
Watchdog timer 2		Operable	Operable when f_{R8} or f_{XT} is selected as the count clock
Serial interface	CSIB0 to CSIB4	Operable	Operable when the \overline{SCKBn} input clock is selected as the count clock (n = 0 to 4)
	I ² C00 to I ² C02	Operable	Stops operation
	UARTA0 to UARTA2	Operable	Stops operation (but UARTA0 is operable when the ASCKA0 input clock is selected)
A/D converter		Operable	Stops operation
D/A converter		Operable	
Real-time output function (RTO)		Operable	Stops operation (output held)
Key interrupt function (KR)		Operable	
CRC operation circuit		Operable	
Port function		Settable	
CPU register set		Settable	
Internal RAM			

Note Be sure to stop the PLL (PLLCTL.PLLON bit = 0) before stopping the main clock.

Caution When the CPU is operating on the subclock and main clock oscillation is stopped, a register for which a wait has been specified must not be accessed. If a wait is generated, it can only be canceled by a reset (see 3.4.8 (2)).

Table 21-12. Operating Status in Low-Voltage Subclock Operation Mode

Setting of Low-Voltage Subclock Operation Mode		Operating Status
		Main Clock Is Stopped (Must Be Stopped)
Item		
LVI		Operable
Subclock oscillator		Oscillates
Internal oscillator		Oscillation enabled
PLL		Stops operation ^{Note}
CPU		Operable
DMA		Stops operation (must stop)
Interrupt controller		Operable
Timer P (TMP0 to TMP5)		Stops operation
Timer Q (TMQ0)		Stops operation
Timer M (TMM0)		Operable when $f_{R/8}$ or f_{XT} is selected as the count clock
Watch timer/RTC		Operable when f_{XT} is selected as the count clock
Watchdog timer 2		Operable when $f_{R/8}$ or f_{XT} is selected as the count clock
Serial interface	CSIB0 to CSIB4	Stops operation (When the $SCKB_n$ input clock is selected as the count clock, be sure to stop the $SCKB_n$ input clock ($n = 0$ to 4).)
	I ² C00 to I ² C02	Stops operation
	UARTA0 to UARTA2	Stops operation (When the ASCKA0 input clock to UARTA0 is selected, be sure to stop the ASCKA0 input clock.)
A/D converter		Stops operation
D/A converter		Stops operation (must stop)
Real-time output function (RTO)		Stops operation (output held)
Key interrupt function (KR)		Operable
CRC operation circuit		Stops operation (must stop)
Port function		Settable
CPU register set		Settable
Internal RAM		

Note Be sure to stop the PLL (PLLCTL.PLLON bit = 0).

Caution When the CPU is operating on the subclock and main clock oscillation is stopped, a register for which a wait is specified must not be accessed. If a wait is generated, it can only be canceled by a reset (see 3.4.8 (2)).

21.7.2 Releasing subclock operation mode

The subclock operation mode is released by a reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low-voltage detector (LVI), or clock monitor (CLM)) when the CK3 bit is set to 0.

If the main clock is stopped (MCK bit = 1), set the MCK bit to 1, secure the oscillation stabilization time of the main clock by software, and set the CK3 bit to 0.

The normal operation mode is restored when the subclock operation mode is released.

Caution When manipulating the CK3 bit, do not change the set values of the CK2 to CK0 bits (using a bit manipulation instruction to manipulate the bit is recommended).

For details of the PCC register, see 5.3 (1) Processor clock control register (PCC).

21.7.3 Releasing low-voltage subclock operation mode

In low-voltage subclock mode, the subclock operation mode is set by setting the REGOVL0 register to 00H. After that, transit to the normal mode according to 21.7.2 Releasing subclock operation mode. Be sure to follow this procedure to transit the mode from the low-voltage subclock operation mode to the subclock operation mode.

(1) Procedure for setting “low-voltage subclock operation mode” → “subclock operation mode”

Make the following settings in the low-voltage subclock operation mode.

- <1>
 - Disable the maskable interrupt by the DI instruction.
 - Disable the NMI interrupt (INTF02 = 0, INTR02 = 0).
 - Create a status in which the INTWDT2 signal is not generated (create a status in which the INTWDT2 signal is not generated immediately after watchdog timer 2 has been cleared).
- <2> Write C9H (enabling data) to the REGPR register.
- <3> Write 00H to the REGOVL0 register (transit to the subclock operation mode).
- <4> Write 00H (protection data) to the REGPR register.
- <5> Wait for at least 800 μs by software.
- <6> As necessary, enable the maskable interrupt, NMI interrupt, or INTWDT2 interrupt by the EI instruction (restore the setting <1> above).
- <7> Enable the DMA if necessary.
- <8> Start the functions to be used, from among those that have been stopped in steps <1> and <2> in section 21.7.1 (1) Procedure for setting “subclock operation mode” → “low-voltage subclock operation mode”.

Be sure to observe the above sequence.

Note, however, that <6>, <7>, and <8> may be performed at any time as long as it is done after <5>.

(2) If low-voltage subclock operation mode is released by reset

When the low-voltage subclock operation mode is released by a reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low-voltage detector (LVI), or clock monitor (CLM)), the CPU shifts to the normal operation mode immediately after the reset ends, and the REGOVL0 register is initialized to 00H and the REGPR register to 00H (protection data). Be sure to secure the oscillation stabilization time that follows immediately after a reset ends by setting the option byte. For details, see CHAPTER 27 OPTION BYTE.

21.8 Sub-IDLE Mode/Low-Voltage Sub-IDLE Mode

21.8.1 Setting and operation status

The sub-IDLE mode is set by setting the PSMR.PSM1 and PSMR.PSM0 bits to 00 or 10 and setting the PSC.STP bit to 1 in the subclock operation mode. The low-voltage sub-IDLE mode is set by setting the PSMR.PSM1 and PSMR.PSM0 bits to 00 or 10 and setting the PSC.STP bit to 1 after setting the REGOVL0 register to 02H in the subclock operation mode.

In this mode, the clock oscillator continues operating but clock supply to the CPU, flash memory, and the other on-chip peripheral functions is stopped.

As a result, program execution stops and the contents of the internal RAM before the sub-IDLE mode was set are retained. The CPU and the other on-chip peripheral functions are stopped. However, the on-chip peripheral functions that can operate with the subclock or an external clock, continue operating. In the subclock operation mode, CSIBn and UARTA0 that can operate with the external clock also continue operating. In the low-voltage subclock operation mode, however, stop supplying the external clock input to CSIBn and UARTA0 because these blocks cannot continue operating (n = 0 to 4).

Because the sub-IDLE mode stops operation of the CPU, flash memory, and other on-chip peripheral functions, it can reduce the power consumption more than the subclock operation mode.

If the sub-IDLE mode is set after the main clock has been stopped, the current consumption can be reduced to a level as low as that in the STOP mode. The power consumption decreases further in the low-voltage sub-IDLE mode because the voltage of the regulator is lowered.

Table 21-13 shows the operating status in the sub-IDLE mode and Table 21-14 shows the operating status in the low-voltage sub-IDLE mode.

Be sure to set the low-voltage sub-IDLE mode in the following procedure.

(1) Procedure for setting “subclock operation mode” → “low-voltage subclock operation mode” → “low-voltage sub-IDLE mode”

Make the following settings in the subclock operation mode.

<1> Stop the main clock and PLL.

<2> Stop the functions that are specified to be stopped in **Table 21-14 Operating Status in Low-Voltage Sub-IDLE Mode**.

Be especially sure to stop the following functions, because they are signals from external sources.

- Stop $\overline{\text{SCKBn}}$ input clock when the $\overline{\text{SCKBn}}$ input clock to CSIBn is selected (n = 0 to 4).
- Stop ASCKA0 input clock when the ASCKA0 input clock to UARTA0 is selected.

<3> Disable the DMA operation (if the DMA operation is enabled).

<4> • Disable the maskable interrupt by the DI instruction.

- Disable the NMI interrupt (INTF02 = 0, INTR02 = 0).

- Create a status in which the INTWDT2 signal is not generated (set a status in which the INTWDT2 signal is not generated immediately after watchdog timer 2 has been cleared).

<5> Write C9H (enabling data) to the REGPR register.

<6> Write 02H to the REGOVL0 register.

At this time, the output voltage of the regulator is at the low level, decreasing the power consumption to an extremely low level.

<7> Write 00H (protection data) to the REGPR register.

<8> As necessary, enable the maskable interrupt, NMI interrupt, or INTWDT2 interrupt by the EI instruction (restore the settings in step <4>).

<9> Set the sub-IDLE mode.

PSMR.PSM1, PSMR.PSM0 bits = 00 or 10 PSC.STP bit = 1

Be sure to observe the above sequence.

For the setting of the subclock operation mode, see 21.7.1 **Setting and operation status**.

- Cautions 1.** Following the store instruction to the PSC register for setting the sub-IDLE mode/low-voltage sub-IDLE mode, insert the five or more NOP instructions.
- 2.** If the sub-IDLE mode/low-voltage sub-IDLE mode is set while an unmasked interrupt request signal is being held pending, the CPU does not shift to the sub-IDLE mode/low-voltage sub-IDLE mode but executes the next instruction.

Table 21-13. Operating Status in Sub-IDLE Mode

Setting of Sub-IDLE Mode		Operating Status	
		When Main Clock Is Oscillating	When Main Clock Is Stopped
LVI		Operable	
Subclock oscillator		Oscillates	
Internal oscillator		Oscillation enabled	
PLL		Operable	Stops operation ^{Note 1}
CPU		Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation (but standby mode release is possible)	
Timer P (TMP0 to TMP5)		Stops operation	
Timer Q (TMQ0)		Stops operation	
Timer M (TMM0)		Operable when $f_{R/8}$ or f_{XT} is selected as the count clock	
Watch timer/RTC		Operable	Operable when f_{XT} is selected as the count clock
Watchdog timer 2		Operable when $f_{R/8}$ or f_{XT} is selected as the count clock	
Serial interface	CSIB0 to CSIB4	Operable when the \overline{SCKBn} input clock is selected as the count clock ($n = 0$ to 4)	
	I ² C00 to I ² C02	Stops operation	
	UARTA0 to UARTA2	Stops operation (but UARTA0 is operable when the ASCKA0 input clock is selected)	
A/D converter		Holds operation (conversion result held) ^{Note 2}	
D/A converter		Holds operation (output held ^{Note 2})	
Real-time output function (RTO)		Stops operation (output held)	
Key interrupt function (KR)		Operable	
CRC operation circuit		Stops operation	
Port function		Retains status before sub-IDLE mode was set	
CPU register set		Retains status before sub-IDLE mode was set	
Internal RAM			

Notes 1. Be sure to stop the PLL (PLLCTL.PLLON bit = 0) before stopping the main clock.

2. To realize low power consumption, stop the A/D and D/A converters before shifting to the sub-IDLE mode.

Table 21-14. Operating Status in Low-Voltage Sub-IDLE Mode

Setting of Low-Voltage Sub-IDLE Mode		Operating Status
		Main Clock Is Stopped (Must Be Stopped)
Item		
LVI		Operable
Subclock oscillator		Oscillates
Internal oscillator		Oscillation enabled
PLL		Stops operation ^{Note}
CPU		Stops operation
DMA		Stops operation
Interrupt controller		Stops operation (but standby mode release is possible)
Timer P (TMP0 to TMP5)		Stops operation
Timer Q (TMQ0)		Stops operation
Timer M (TMM0)		Operable when $f_{R/8}$ or f_{XT} is selected as the count clock
Watch timer/RTC		Operable when f_{XT} is selected as the count clock
Watchdog timer 2		Operable when $f_{R/8}$ or f_{XT} is selected as the count clock
Serial interface	CSIB0 to CSIB4	Stops operation (When the SCKBn input clock is selected as the count clock, be sure to stop the SCKBn input clock (n = 0 to 4).)
	I ² C00 to I ² C02	Stops operation
	UARTA0 to UARTA2	Stops operation (When the ASCKA0 input clock to UARTA0 is selected, be sure to stop the ASCKA0 input clock.)
A/D converter		Stops operation
D/A converter		Stops operation (must stop)
Real-time output function (RTO)		Stops operation (output held)
Key interrupt function (KR)		Operable
CRC operation circuit		Stops operation
Port function		Retains status before low-voltage sub-IDLE mode was set
CPU register set		Retains status before low-voltage sub-IDLE mode was set
Internal RAM		

Note Be sure to stop the PLL (PLLCTL.PLLON bit = 0).

21.8.2 Releasing sub-IDLE mode/low-voltage sub-IDLE mode

The sub-IDLE mode/low-voltage sub-IDLE mode is released by a non-maskable interrupt request signal (NMI pin input, INTWDT2 signal), unmasked external interrupt request signal (INTP0 to INTP7 pin input), unmasked internal interrupt request signal from the peripheral functions operable in the sub-IDLE mode/low-voltage sub-IDLE mode, or reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, low-voltage detector (LVI), or clock monitor (CLM)). The PLL returns to the operating status it was in before the sub-IDLE mode was set. It returns to the stop status in the low-voltage sub-IDLE mode.

When the sub-IDLE mode is released by an interrupt request signal, the subclock operation mode is set.

When the low-voltage sub-IDLE mode is released by an interrupt request signal, the low-voltage subclock operation mode is set.

For releasing low-voltage subclock operation mode, see **21.7.3 Releasing low-voltage subclock operation mode**.

(1) Releasing sub-IDLE mode/low-voltage sub-IDLE mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The sub-IDLE mode/low-voltage sub-IDLE 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. If the sub-IDLE mode/low-voltage sub-IDLE mode is set in an interrupt servicing routine, however, an interrupt request signal that is issued later is serviced as follows.

Table 21-15. Operation After Releasing Sub-IDLE Mode/Low-Voltage IDLE Mode by Interrupt Request Signal

Release Source	Interrupt Acknowledgment Status	Status After Release	Operation After Release	
Reset	Disabled (DI)	–	Normal reset operation	
	Enabled (EI)			
Non-maskable interrupt request signal (excluding multiple interrupts)	Disabled (DI)	–	The interrupt request is acknowledged when the sub-IDLE mode/low-voltage sub-IDLE mode is released.	
	Enabled (EI)			
Maskable interrupt request signal	Disabled (DI)	–	The sub-IDLE mode/low-voltage sub-IDLE mode is released but the interrupt request that is the release source is not acknowledged. The interrupt request itself is retained. The processing that was being executed before shifting to the sub-IDLE mode/low-voltage sub-IDLE mode is executed.	
		Enabled (EI)	• An interrupt request with a priority higher than that of the release source is being serviced.	The sub-IDLE mode/low-voltage sub-IDLE mode is released but the interrupt request that is the release source is not acknowledged. The interrupt request itself is retained. The interrupt that was being serviced before shifting to the sub-IDLE mode/low-voltage sub-IDLE mode is serviced.
			• An interrupt request with a priority lower than that of the release source is being serviced.	The interrupt request is acknowledged when the sub-IDLE mode/low-voltage sub-IDLE mode is released.

Caution An interrupt request signal that is disabled by setting the PSC.NMI2M, PSC.NMI0M, and PSC.INTM bits to 1 (interrupt disabled) is invalid and cannot release the sub-IDLE mode/low-voltage sub-IDLE mode.

CHAPTER 22 RESET FUNCTION

22.1 Overview

The reset function is used to initialize the settings of the V850ES/JC3-L and V850ES/JE3-L functions. This function is used, for example, to stop operation at power-on until the supply voltage reaches the operation voltage level, or to initialize the settings of the V850ES/JC3-L and V850ES/JE3-L functions at any time.

The V850ES/JC3-L and V850ES/JE3-L start operating at address 00000000H immediately after a reset ends.

The following sources cause a reset:

(1) Four reset sources

- External reset input via the $\overline{\text{RESET}}$ pin
- Reset via a watchdog timer 2 (WDT2) overflow (WDT2RES)
- System reset based on comparison of the low-voltage detector (LVI) supply voltage and detected voltage
- System reset based on detecting that oscillation of clock monitor (CLM) has stopped

The source of the reset can be confirmed by using the reset source flag register (RESF) immediately after a reset ends.

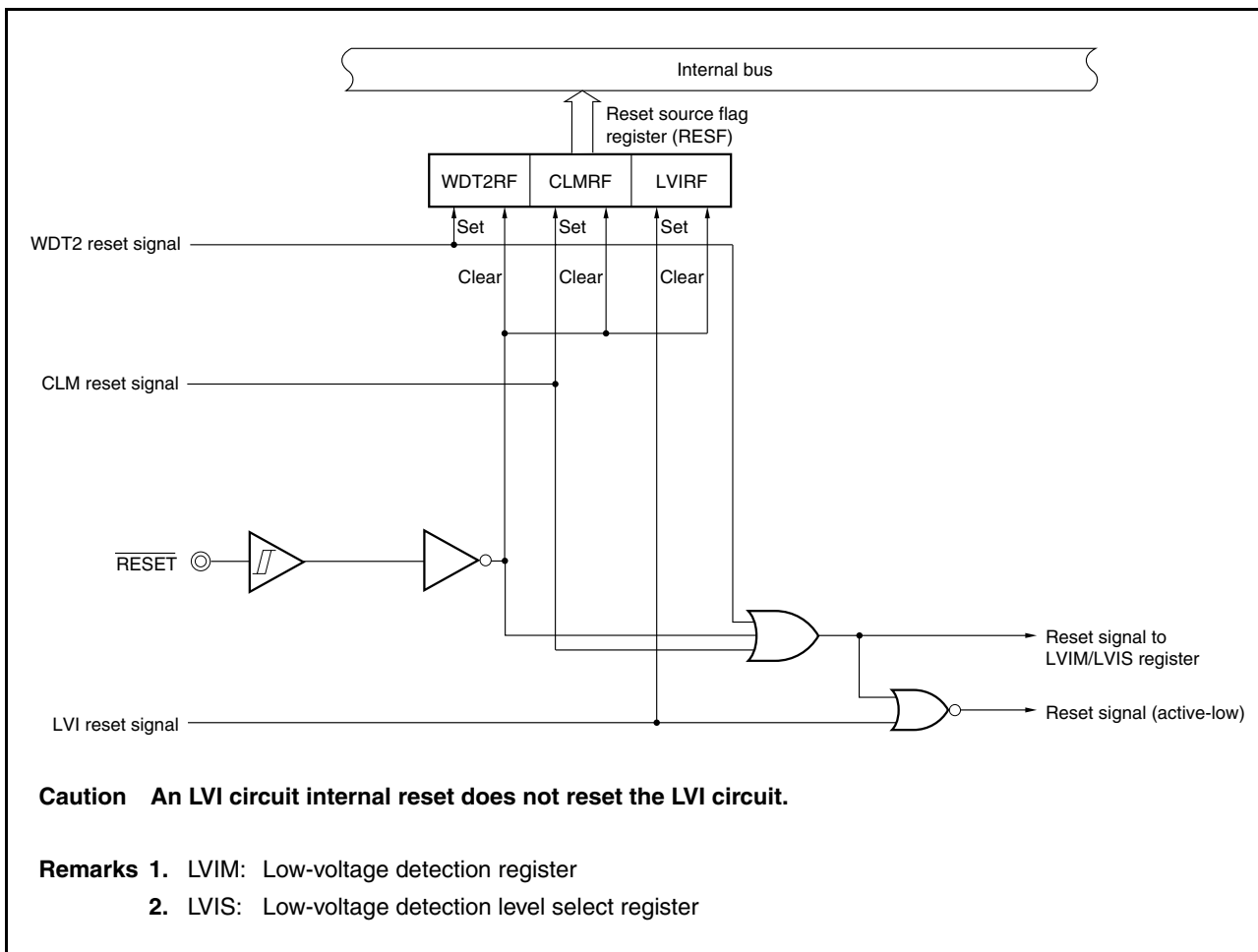
(2) Emergency operation mode

If WDT2 overflows during the main clock oscillation stabilization time inserted after a reset, the main clock oscillation is judged as abnormal and the CPU starts operating on the internal oscillator clock.

Caution In the emergency operation mode, do not access on-chip peripheral I/O registers other than those for the interrupt function, port function, WDT2, and timer M, which can operate on the internal oscillator clock. In addition, operating CSIB0 to CSIB4 and UARTA0 by using an external clock is also prohibited.

22.2 Configuration

Figure 22-1. Block Diagram of Reset Function



22.3 Register to Check Reset Source

The V850ES/JC3-L and V850ES/JE3-L have four reset sources. The source of the reset that occurred can be checked by using the reset source flag register (RESF) immediately after a reset ends.

(1) Reset source flag register (RESF)

The RESF register is a special register that can be written only in a combination of specific sequences (see **3.4.7 Special registers**).

The RESF register indicates the source that generated a reset signal.

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

$\overline{\text{RESET}}$ pin input clears this register to 00H. The default value differs if the source of the reset is other than the $\overline{\text{RESET}}$ pin signal.

After reset: 00H^{Note} R/W Address: FFFF888H

	7	6	5	4	3	2	1	0
RESF	0	0	0	WDT2RF	0	0	CLMRF	LVIRF

WDT2RF	Reset signal from WDT2
0	Not generated
1	Generated

CLMRF	Reset signal from CLM
0	Not generated
1	Generated

LVIRF	Reset signal from LVI
0	Not generated
1	Generated

Note The value of the RESF register is cleared to 00H when a reset is executed via the $\overline{\text{RESET}}$ pin. When a reset is executed by watchdog timer 2 (WDT2), the low-voltage detector (LVI), or the clock monitor (CLM), the reset flags of this register (WDT2RF bit, CLMRF bit, and LVIRF bit) are set. However, other sources are retained.

Caution Only "0" can be written to each bit of this register. If writing "0" conflicts with setting the flag (occurrence of reset), setting the flag takes precedence.

22.4 Operation

22.4.1 Reset operation via $\overline{\text{RESET}}$ pin

When a low level is input to the $\overline{\text{RESET}}$ pin, the system is reset, and each hardware unit is initialized.

When the level of the $\overline{\text{RESET}}$ pin is changed from low to high, the reset status ends.

The $\overline{\text{RESET}}$ pin has an internal noise elimination circuit that uses analog delay (60 ns (TYP.)) to prevent malfunction caused by noise.

Table 22-1. Hardware Status on $\overline{\text{RESET}}$ Pin Input

Item	During Reset	After Reset
Main clock oscillator (fx)	Oscillation stops	Oscillation starts
Subclock oscillator (fxT)	Oscillation continues	
Internal oscillator	Oscillation stops	Oscillation starts
Peripheral clock (fx to fx/1,024)	Operation stops	Operation starts after securing oscillation stabilization time
Internal system clock (fCLK), CPU clock (fCPU)	Operation stops	Operation starts after securing oscillation stabilization time (initialized to fx/8)
CPU	Initialized	Program execution starts at address 00000000H after securing oscillation stabilization time
Watchdog timer 2	Operation stops (initialized to 0)	Counts up from 0 with internal oscillator clock as source clock.
RTC	Operation continues	
Internal RAM	Undefined if power-on reset or CPU access and reset input conflict (data is damaged). Otherwise value immediately after reset input is retained.	
I/O lines (ports/alternate-function pins)	High impedance ^{Note}	
On-chip peripheral I/O registers	Initialized to specified status, OCDM register is set (01H).	
Other on-chip peripheral functions	Operation stops	Operation can be started after securing oscillation stabilization time

Note When the power is turned on, the following pins may output an undefined level temporarily even during reset.

- P10/ANO0 pin
- P53/SIB2/KR3/TIQ00/TOQ00/RTP03/DDO pin

Caution The OCDM register is initialized by the $\overline{\text{RESET}}$ pin input. Therefore, note with caution that, if a high level is input to the P05/ $\overline{\text{DRST}}$ pin immediately after a reset ends before the OCDM.OCDM0 bit is cleared, the on-chip debug mode may be entered. For details, see CHAPTER 4 PORT FUNCTIONS.

Figure 22-2. Timing of Reset Operation by RESET Pin Input

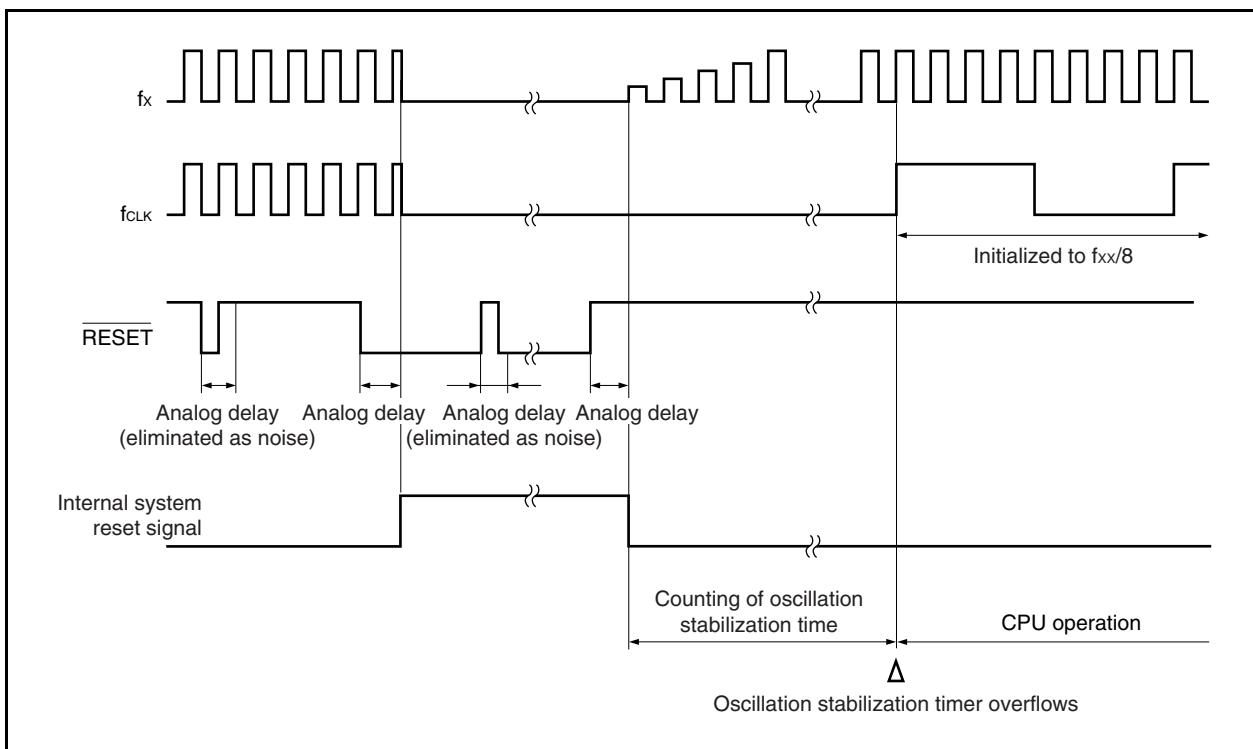
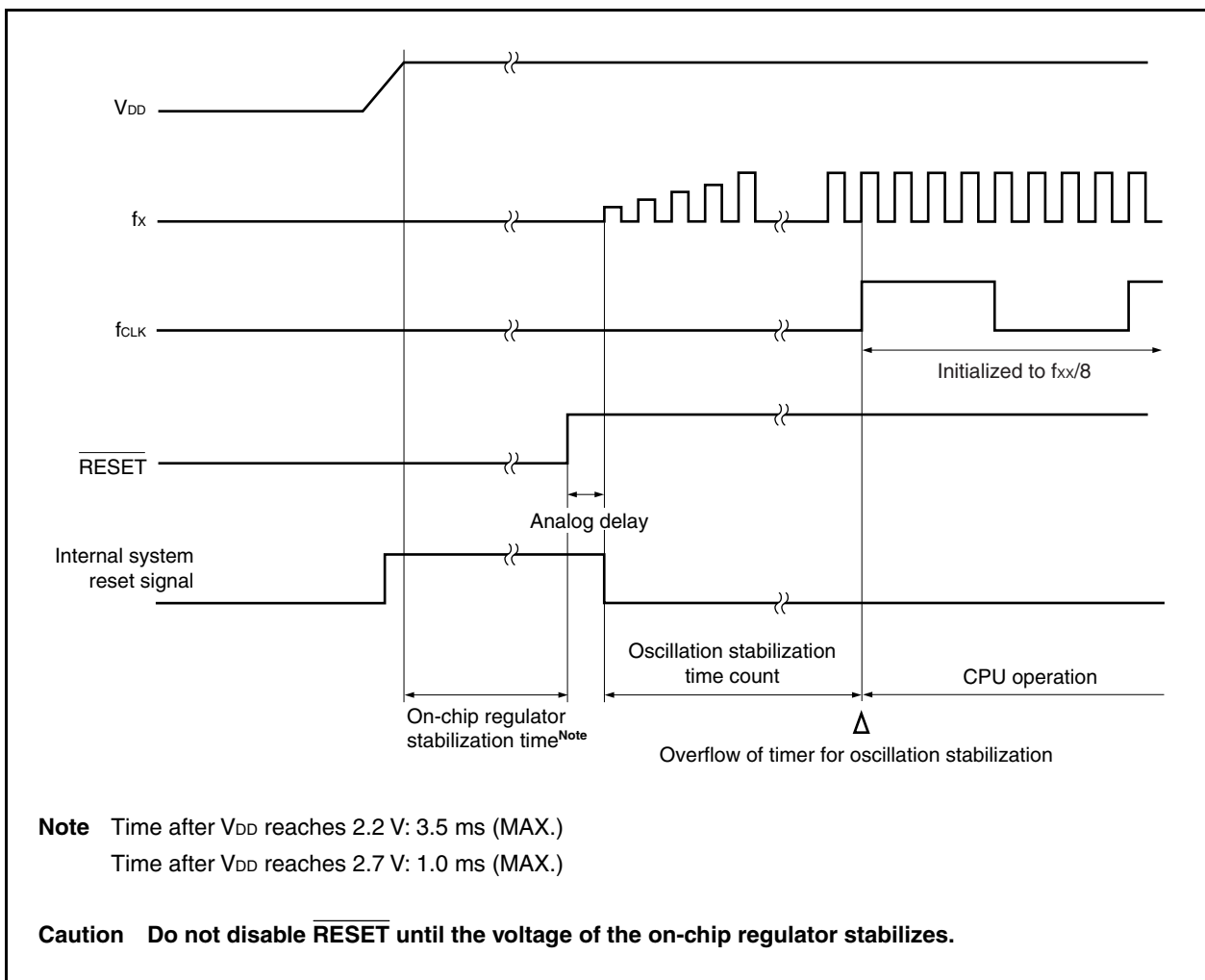


Figure 22-3. Timing of Power-on Reset Operation



22.4.2 Reset operation by watchdog timer 2

When watchdog timer 2 is set to the reset operation mode due to overflow, upon watchdog timer 2 overflow (WDT2RES signal generation), a system reset is executed and the hardware is initialized to the initial status.

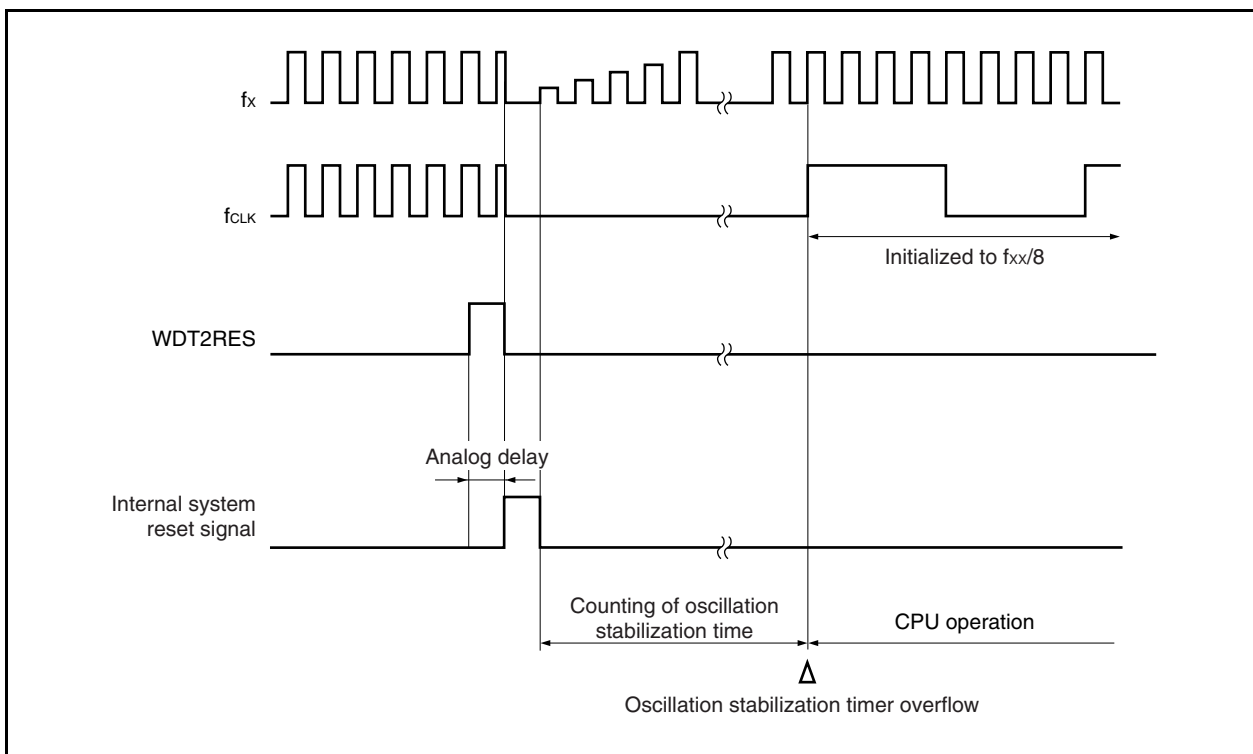
Following watchdog timer 2 overflow, the reset status is entered and lasts the predetermined time (analog delay), and then the reset status ends automatically.

The main clock oscillator is stopped during the reset period.

Table 22-2. Hardware Status During Watchdog Timer 2 Reset Operation

Item	During Reset	After Reset
Main clock oscillator (f_x)	Oscillation stops.	Oscillation starts.
Subclock oscillator (f_{XT})	Oscillation continues.	
Internal oscillator	Oscillation stops.	Oscillation starts.
Peripheral clock (f_{xx} to $f_{xx}/1,024$)	Operation stops.	Operation starts after securing oscillation stabilization time.
Internal system clock (f_{xx}), CPU clock (f_{CPU})	Operation stops.	Operation starts after securing oscillation stabilization time (initialized to $f_{xx}/8$).
CPU	Initialized	Program execution after securing oscillation stabilization time
Watchdog timer 2	Operation stops (initialized to 0).	Counts up from 0 with internal oscillator clock as source clock.
RTC	Operation continues	
Internal RAM	Undefined if power-on reset or CPU access and reset input conflict (data is damaged). Otherwise value immediately after reset input is retained.	
I/O lines (ports/alternate-function pins)	High impedance	
On-chip peripheral I/O register	Initialized to specified status, OCDM register retains its value.	
On-chip peripheral functions other than above	Operation stops.	Operation can be started after securing oscillation stabilization time.

Figure 22-4. Timing of Reset Operation by WDT2RES Signal Generation



22.4.3 Reset operation by low-voltage detector

If the supply voltage falls below the voltage detected by the low-voltage detector when LVI operation is enabled, a system reset is executed (when the LVIM.LVIMD bit is set to 1), and the hardware is initialized to the initial status.

The reset status lasts from when a supply voltage drop has been detected until the supply voltage rises above the LVI detection voltage.

The main clock oscillator is stopped during the reset period.

When the LVIMD bit is cleared to 0, an interrupt request signal (INTLVI) is generated if the supply voltage falls below or exceeds the detected voltage.

Table 22-3. Hardware Status During Reset Operation by Low-Voltage Detector

Item	During Reset	After Reset
Main clock oscillator (fx)	Oscillation stops.	Oscillation starts.
Subclock oscillator (fxT)	Oscillation continues.	
Internal oscillator	Oscillation stops.	Oscillation starts.
Peripheral clock (fx to fx/1,024)	Operation stops.	Operation starts after securing oscillation stabilization time.
Internal system clock (fxx), CPU clock (fCPU)	Operation stops	Operation starts after securing oscillation stabilization time (initialized to fxx/8).
CPU	Initialized	Program execution starts after securing oscillation stabilization time
Watchdog timer 2	Operation stops (initialized to 0).	Counts up from 0 with internal oscillator clock as source clock.
RTC	Operation continues	
Internal RAM	Undefined	
I/O lines (ports/alternate-function pins)	High impedance	
On-chip peripheral I/O register	Initialized to specified status, OCDM register retains its value.	
LVI	Operation stops.	
On-chip peripheral functions other than above	Operation stops.	Operation can be started after securing oscillation stabilization time.

Remark For the reset timing of the low-voltage detector, see **CHAPTER 24 LOW-VOLTAGE DETECTOR (LVI)**.

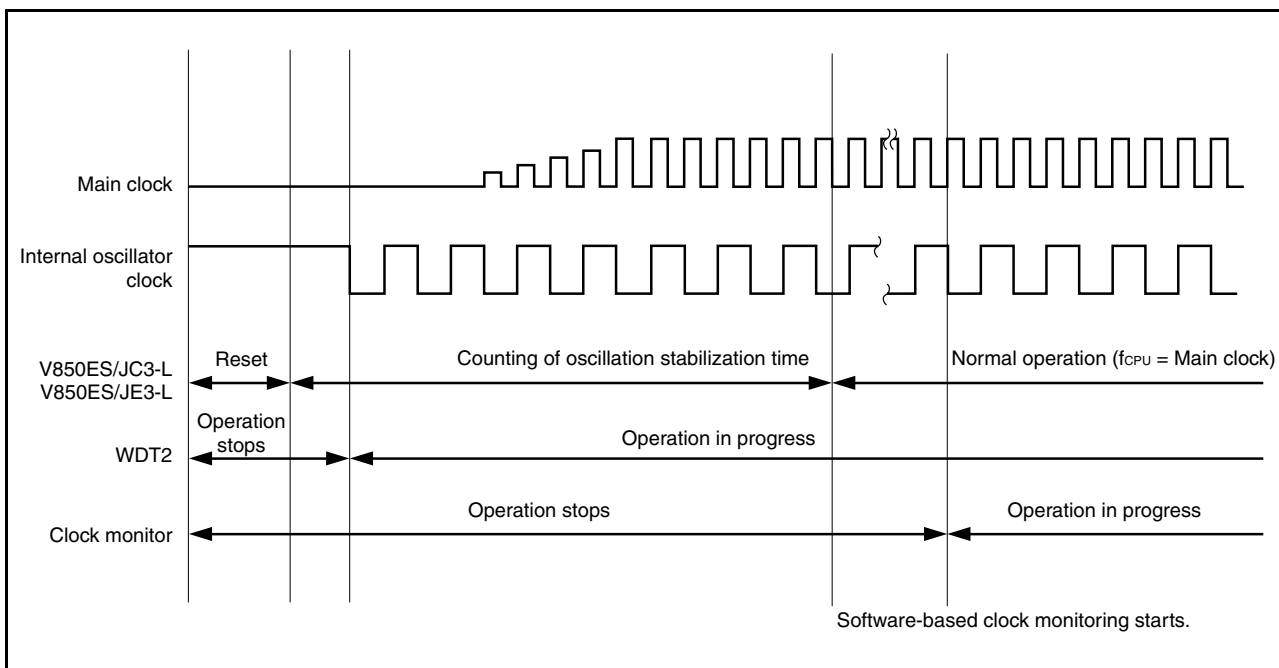
22.4.4 Operation immediately after reset ends

(1) Immediately after reset ends normally

Immediately after a reset ends, the main clock starts oscillating, the oscillation stabilization time (which differs depending on the option byte; for details, see **CHAPTER 27 OPTION BYTE**) is secured, and then the CPU starts executing the program.

WDT2 begins to operate immediately after a reset ends using the internal oscillator clock as the source clock.

Figure 22-5. Operation Immediately After Reset Ends

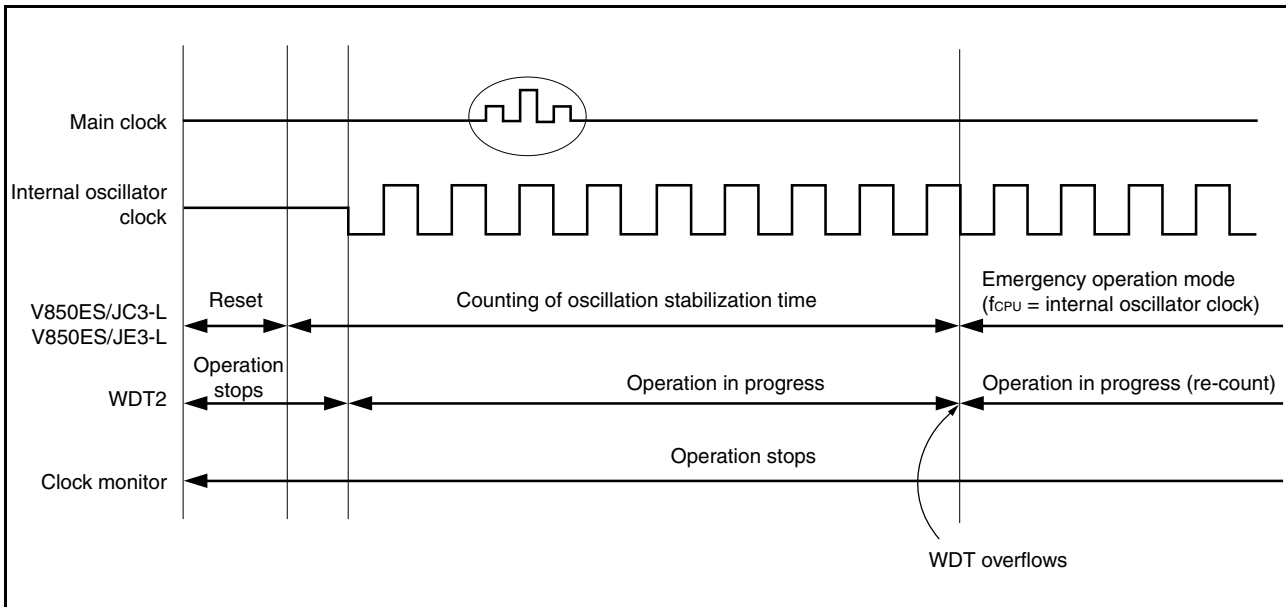


(2) Emergency operation mode

If an anomaly occurs in the main clock before the oscillation stabilization time is secured, WDT2 overflows before the CPU starts executing the program. At this time, the CPU starts executing the program by using the internal oscillator clock as the source clock.

Caution In the emergency operation mode, do not access on-chip peripheral I/O registers other than those for the interrupt function, port function, WDT2, and timer M, which can operate on the internal oscillator clock. In addition, operating CSIB0 to CSIB4 and UARTA0 by using an external clock is also prohibited.

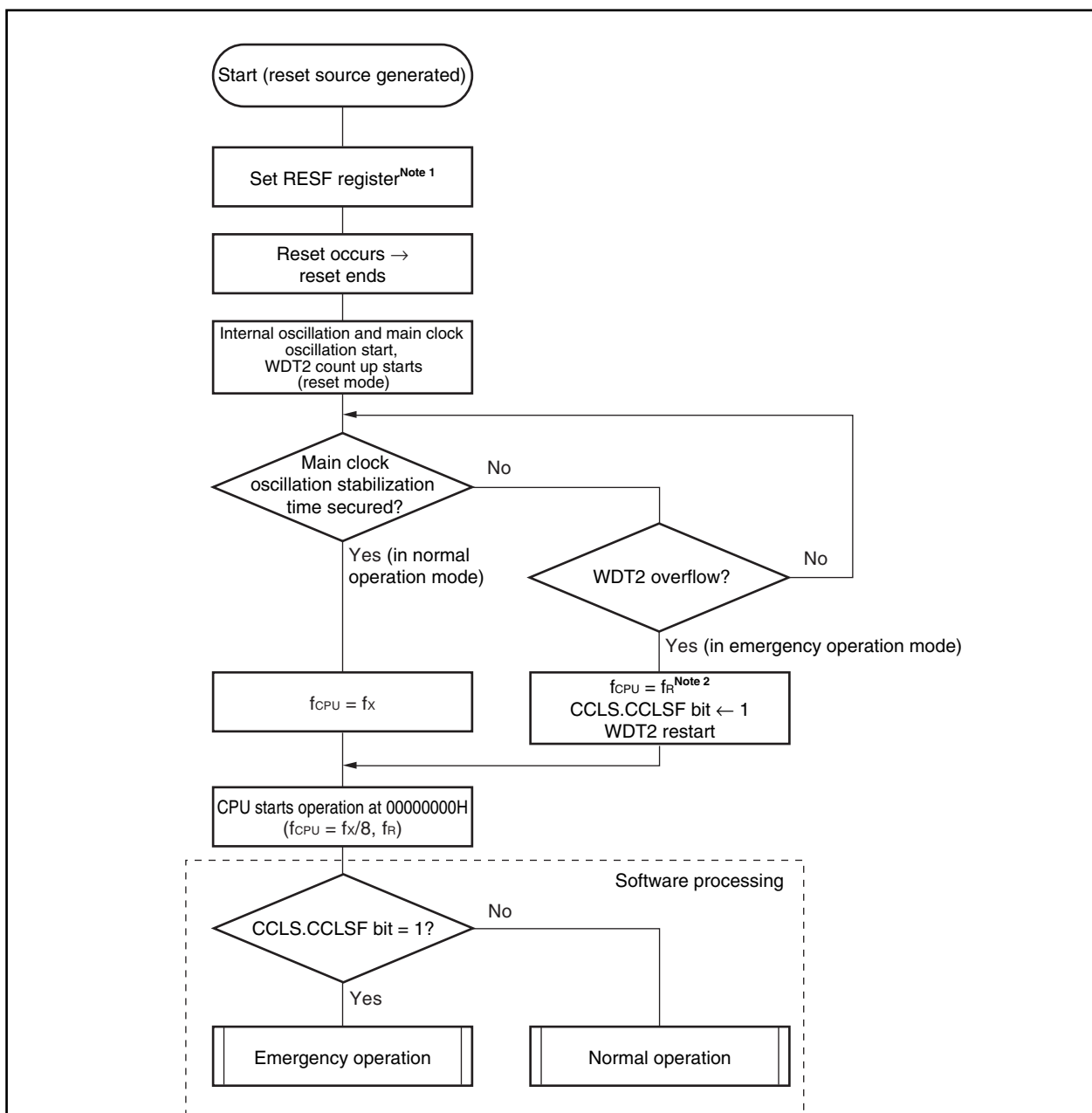
Figure 22-6. Operation Immediately After Reset Ends



The CPU operation clock states can be checked by using the CPU operation clock status register (CCLS).

22.4.5 Reset function operation

Figure 22-7. Reset Function Operation



Notes 1. The bit to be set differs depending on the reset source.

Reset Source	WDT2RF Bit	CRMRF Bit	LVIRF Bit
RESET pin	0	0	0
WDT2	1	Value before reset is retained.	Value before reset is retained.
CLM	Value before reset is retained.	1	Value before reset is retained.
LVI	Value before reset is retained.	Value before reset is retained.	1

2. The internal oscillator cannot be stopped.

22.5 Cautions

When executing a power-on reset operation, the supply voltage must be within the guaranteed operating range immediately after the reset ends. The usable range of the internal operating frequency of the V850ES/JC3-L and V850ES/JE3-L depend on the supply voltage (5 MHz (MAX.) @ 2.2 to 2.7 V or 20 MHz (MAX.) @ 2.7 to 3.6 V).

(1) At less than 2.2 V immediately after reset ends

Use prohibited

(2) At 2.2 V or more to less than 2.7 V immediately after reset ends

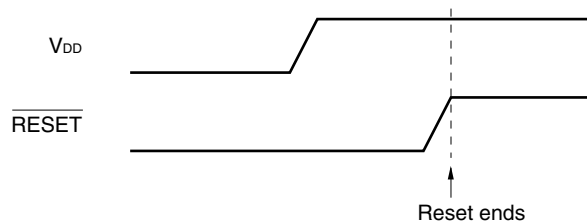
- Input $f_x = 2.5$ to 5 MHz to the main clock oscillator and set the clock-through mode (PLLCTL.SELPLL = 0).
- Inputting 5 MHz or more to the main clock oscillator is prohibited.
- Be sure to stop the PLL (PLLCTL.PLLON = 0) in the initialization routine.

(3) At 2.7 to 3.6 V immediately after reset ends

- Both the clock-through mode and PLL mode can be used.

Remarks 1. The voltage value (V) is the value of V_{DD} .

- 2.** A reset ends at the following timing. For the relationship between the rising of V_{DD} and when the reset signal generated by the $\overline{\text{RESET}}$ pin ends, see **30.7.4 or 31.7.4 or 32.7.4 Power on/power off/reset timing**.



CHAPTER 23 CLOCK MONITOR

23.1 Functions

The clock monitor monitors the main clock by using the internal oscillator clock and generates a reset request signal when oscillation of the main clock is stopped.

Once the operation of the clock monitor has been enabled by an operation enable flag, it cannot be cleared to 0 by any means other than a reset.

When a reset by the clock monitor occurs, the RESF.CLMRF bit is set. For details on the RESF register, see **22.3 Registers to Check Reset Source**.

The clock monitor automatically stops under the following conditions.

- During oscillation stabilization time after STOP mode is released
- When the main clock is stopped (from when the PCC.MCK bit = 1 during subclock operation, until the PCC.CLS bit = 0 during main clock operation)
- When the monitoring clock (internal oscillator clock) is stopped
- When the CPU operates with the internal oscillator clock

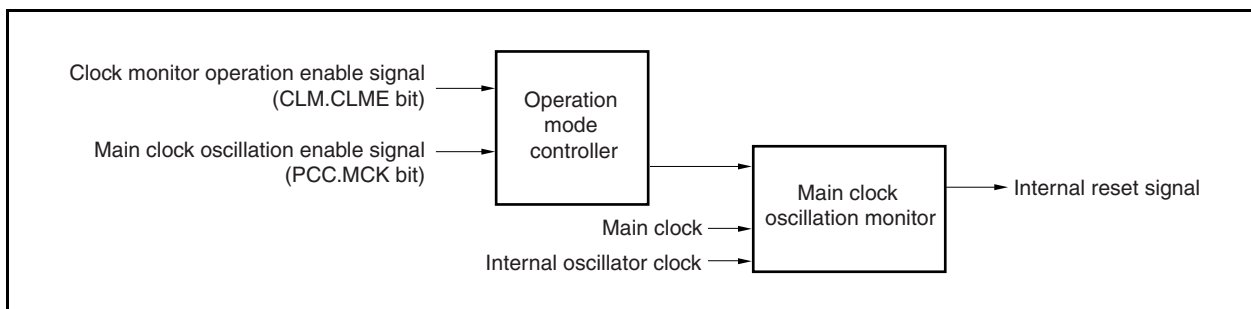
23.2 Configuration

The clock monitor includes the following hardware.

Table 23-1. Configuration of Clock Monitor

Item	Configuration
Control register	Clock monitor mode register (CLM)

Figure 23-1. Block Diagram of Clock Monitor



23.3 Registers

The clock monitor is controlled by the clock monitor mode register (CLM).

(1) Clock monitor mode register (CLM)

The CLM register is a special register that can only be written in a combination of specific sequences (see 3.4.7 **Special registers**).

This register is used to set the operation mode of the clock monitor.

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

Reset sets this register to 00H.

After reset: 00H	R/W	Address: FFFFF870H						
	7	6	5	4	3	2	1	<0>
CLM	0	0	0	0	0	0	0	CLME

CLME	Clock monitor operation enable or disable
0	Disable clock monitor operation.
1	Enable clock monitor operation.

Cautions

1. Once the CLME bit has been set to 1, it cannot be cleared to 0 by any means other than a reset.
2. When a reset by the clock monitor occurs, the CLME bit is cleared to 0 and the RESF.CLMRF bit is set to 1.

23.4 Operation

This section describes the clock monitor operation. The monitoring start and monitoring stop conditions are as follows.

<Monitoring start condition >

Enabling operation by setting the CLM.CLME bit to 1

<Monitoring stop conditions >

- While oscillation stabilization time is being counted after STOP mode is released
- When the main clock is stopped (from when PCC.MCK bit is set to 1 during subclock operation to when PCC.CLS bit is set to 0 during main clock operation)
- When the sampling clock (internal oscillator clock) is stopped
- When the CPU operates on the internal oscillator clock

Table 23-2. Operation Status of Clock Monitor (When CLM.CLME Bit = 1)

CPU Operating Clock	Operation Mode	Status of Main Clock	Status of Internal Oscillator Clock	Status of Clock Monitor
Main clock	HALT mode	Oscillates	Oscillates ^{Note 1}	Operates
	IDLE1, IDLE2 modes	Oscillates	Oscillates ^{Note 1}	Operates
	STOP mode	Stops	Oscillates ^{Note 1}	Stops
Subclock (MCK bit of PCC register = 0)	Sub-IDLE mode	Oscillates	Oscillates ^{Note 1}	Operates
Subclock (MCK bit of PCC register = 1)	Sub-IDLE mode	Stops	Oscillates ^{Note 1}	Stops
Internal oscillator clock	Emergency operation mode ^{Note 2}	Stops	Oscillates ^{Note 3}	Stops
During reset	–	Stops	Stops	Stops

Notes 1. The internal oscillator can be stopped by setting the RCM.RSTOP bit to 1.

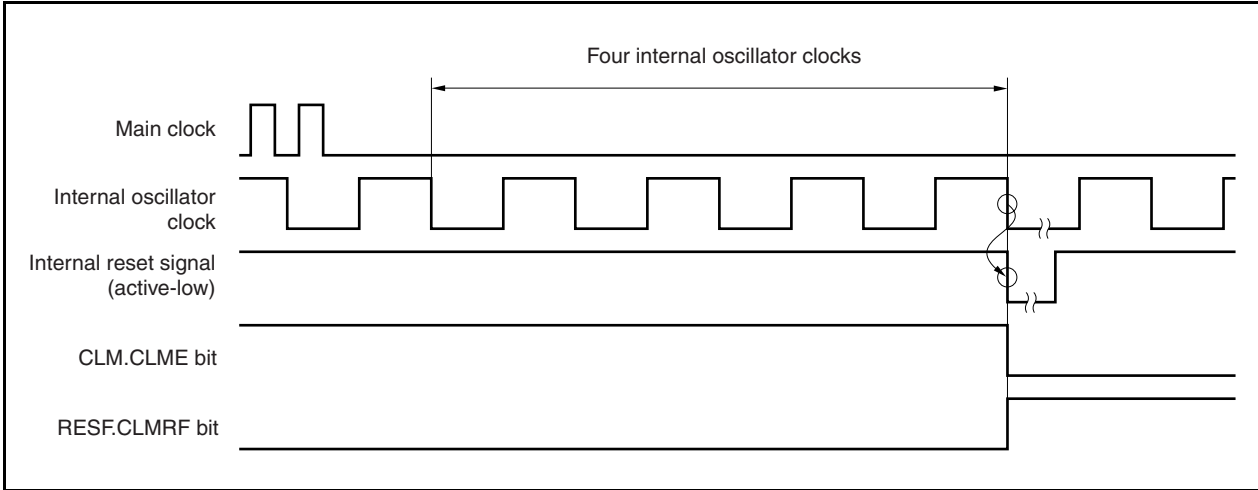
2. See **25.4.4 (2) Emergency operation mode.**

3. The internal oscillator cannot be stopped by software.

(1) Operation when main clock oscillation is stopped (CLME bit = 1)

If oscillation of the main clock is stopped when the CLME bit is 1, an internal reset signal is generated as shown in Figure 23-2.

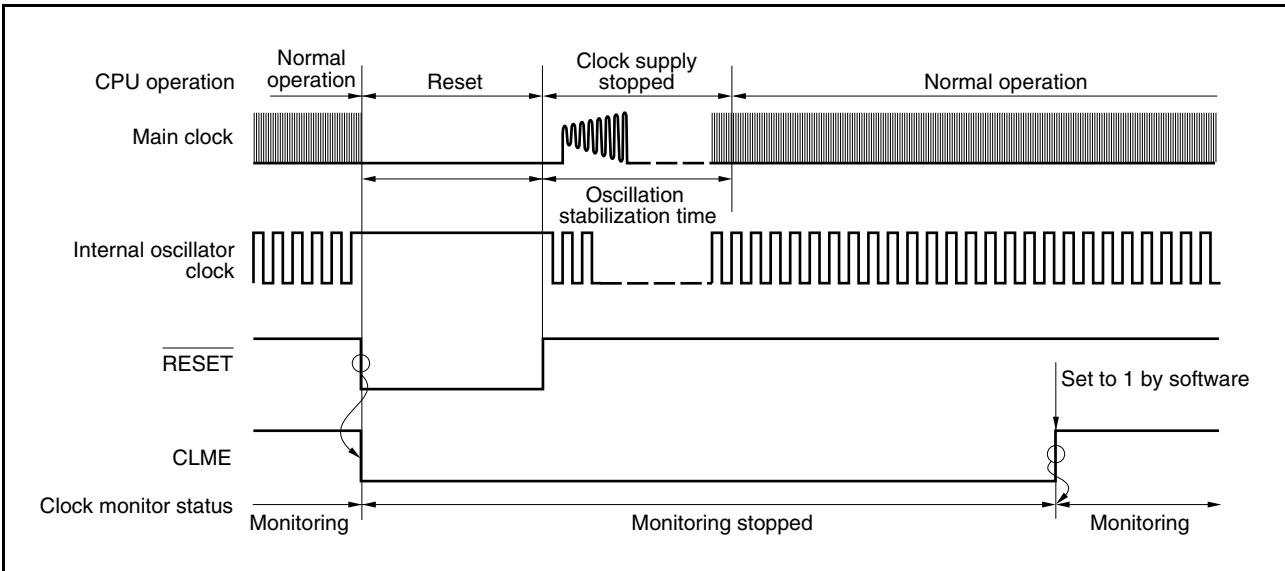
Figure 23-2. Reset Period Due to Stoppage of Main Clock Oscillation



(2) Clock monitor status after $\overline{\text{RESET}}$ input

$\overline{\text{RESET}}$ input clears the CLM.CLME bit to 0 and stops the clock monitor operation. When the CLME bit is set to 1 by software after the normal operation is started, monitoring is started.

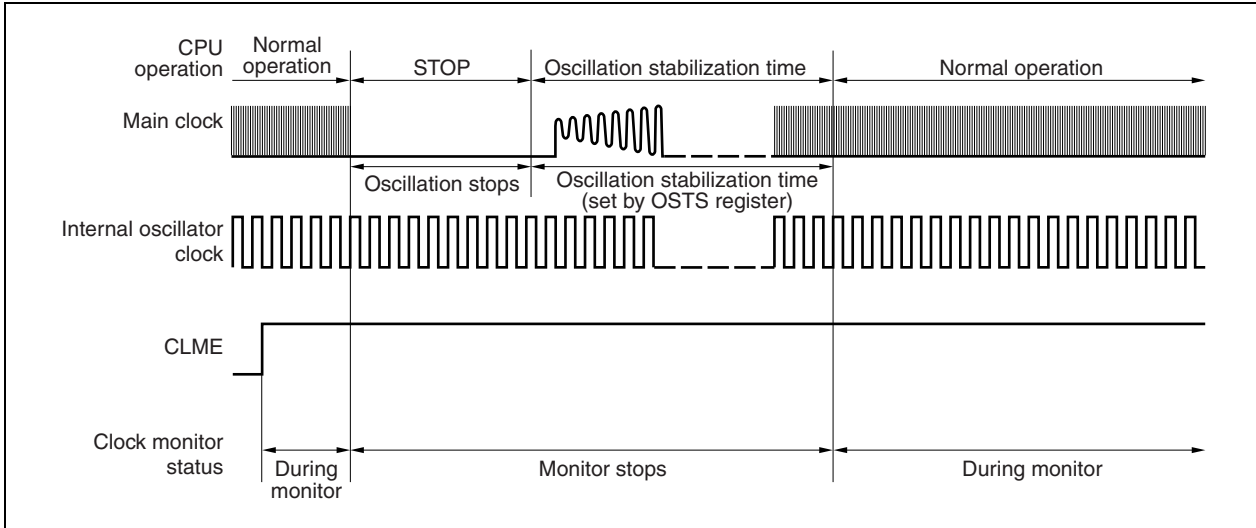
**Figure 23-3. Clock Monitor Status After $\overline{\text{RESET}}$ Input
(CLM.CLME Bit = 1 Is Set After $\overline{\text{RESET}}$ Is Input and Normal Operation Is Started)**



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

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

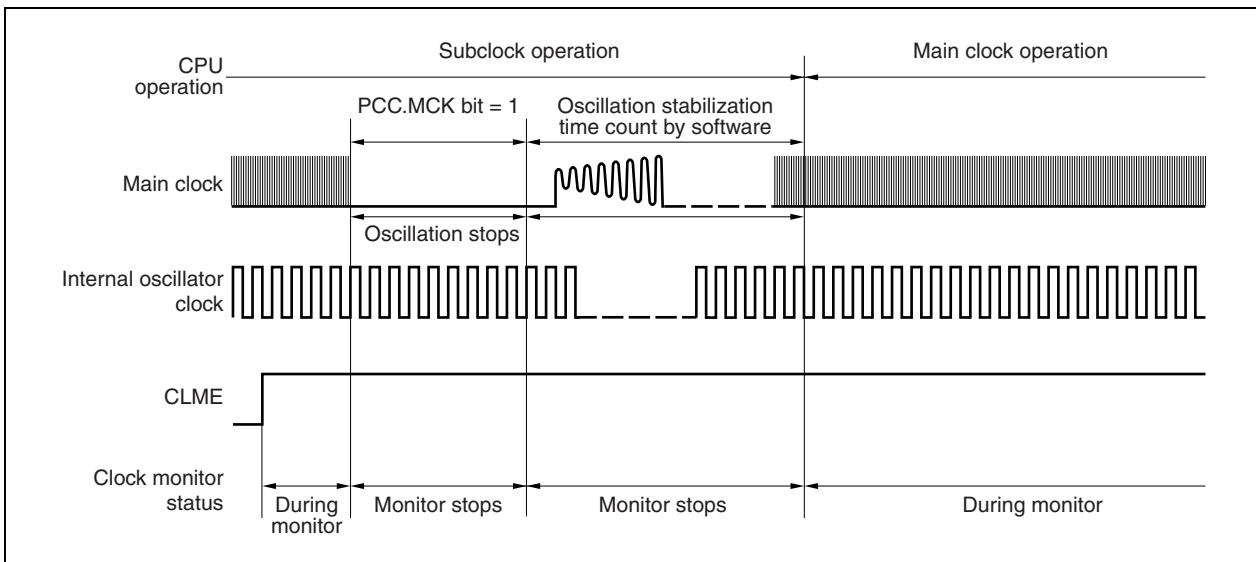
Figure 23-4. Operation in STOP Mode or After STOP Mode Is Released



(4) Operation when main clock is stopped (arbitrary)

During subclock operation (PCC.CLS bit = 1) or when the main clock is stopped by setting the PCC.MCK bit to 1, the monitor operation is stopped until the main clock operation is started (PCC.CLS bit = 0). The monitor operation is automatically started when the main clock operation is started.

Figure 23-5. Operation When Main Clock Is Stopped (Arbitrary)



(5) Operation while CPU is operating on internal oscillator clock (CCLS.CCLS bit = 1)

The monitor operation is not stopped when the CCLS bit is 1, even if the CLME bit is set to 1.

CHAPTER 24 LOW-VOLTAGE DETECTOR (LVI)

24.1 Functions

The low-voltage detector (LVI) has the following functions.

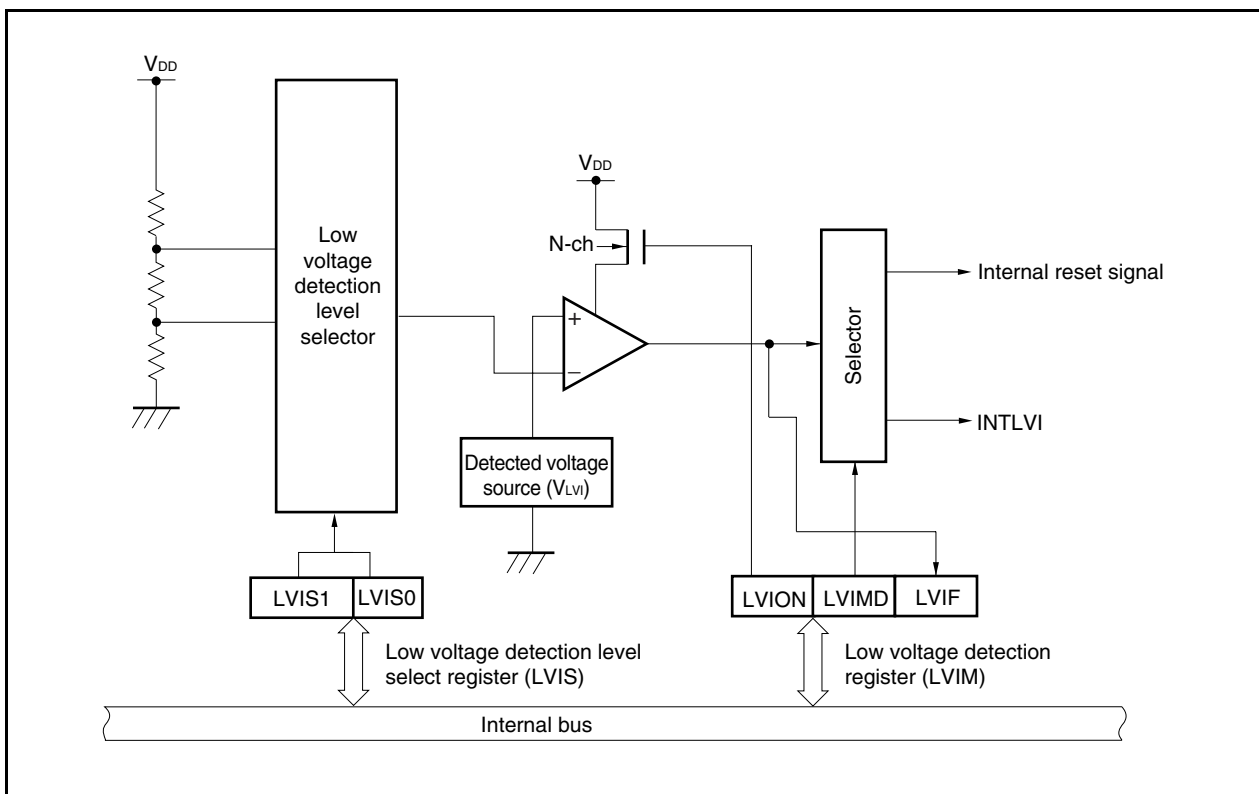
- If interrupt occurrence at low-voltage detection is selected as the operation mode, the low-voltage detector compares the supply voltage (V_{DD}) and the detection voltage (V_{LVI}), and generates an internal interrupt signal when the supply voltage drops below or rises above the detection voltage.
- If reset occurrence at low-voltage detection is selected as the operation mode, the low-voltage detector generates an internal reset signal when the supply voltage (V_{DD}) drops below the detection voltage (V_{LVI}).
- The level of the supply voltage to be detected can be changed by software.
- Interrupt or reset signal can be selected by software.
- The low-voltage detector is operable in the standby mode.

If a reset occurs when the low-voltage detector is selected to generate a reset signal, the RESF.LVIRF bit is set to 1. For details about the RESF register, see **22.3 Register to Check Reset Source**.

24.2 Configuration

The block diagram of the low-voltage detector is shown below.

Figure 24-1. Block Diagram of Low-Voltage Detector



24.3 Registers

The low-voltage detector is controlled by the following registers.

- Low voltage detection register (LVIM)
- Low voltage detection level select register (LVIS)

(1) Low voltage detection register (LVIM)

The LVIM register is a special register. This can be written only in a combination of specific sequences (see **3.4.7 Special registers**).

The LVIM register is used to enable or disable low voltage detection, and to set the operation mode of the low-voltage detector.

This register can be read or written in 8-bit or 1-bit units. However, the LVIF bit is read-only.

After reset: **Note 1** R/W Address: FFFFF890H

	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF

LVION	Low voltage detection enable or disable
0	Disable operation.
1	Enable operation.

LVIMD	Selection of operation mode of low-voltage detector
0	Generate interrupt request signal INTLVI when supply voltage drops below or rises above detection voltage.
1	Generate internal reset signal LVIRESET when supply voltage drops below detection voltage.

LVIF ^{Notes 2, 3, 4}	Low voltage detection flag
0	Supply voltage rises above detection voltage, or operation is disabled.
1	Supply voltage of connected power supply is lower than detection voltage.

Notes 1. Reset by low-voltage detection: 82H

Reset due to other source: 00H

- Do not change the LVION bit from 1 to 0 while the supply voltage (V_{DD}) is lower than the detection voltage (V_{LVI}) (LVIM.LVIF bit = 1).
- After the LVI operation has started (LVION bit = 1), check the LVIF bit.
- When the INTLVI signal is generated, check the LVIF bit to see whether the supply voltage has fallen below or exceeds the detection voltage.

Cautions 1. When the LVION and LVIMD bits are set to 1, the low-voltage detector cannot be stopped until a reset request due to other than low-voltage detection is generated.

- When the LVION bit is set to 1, the comparator in the LVI circuit starts operating. Wait at least 0.2 ms, set by software, before checking the voltage by using the LVIF bit after the LVION bit is set.
- Be sure to set bits 6 to 2 to "0".

(2) Low voltage detection level select register (LVIS)

The LVIS register is used to select the level of voltage to be detected.

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

	After reset: Note1	R/W	Address: FFFFF891H					
	7	6	5	4	3	2	1	<0>
LVIS	0	0	0	0	0	0	0	LVIS0

LVIS0	Low-voltage detection level
0	2.80 V (TYP.)
1	2.30 V (TYP.)

Note Reset by low-voltage detection: Retained
 Reset due to other source: 00H

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

24.4 Operation

Depending on the setting of the LVIM.VIMD bit, an interrupt signal (INTLVI) or an internal reset signal is generated. How to specify each operation is described below, together with timing charts.

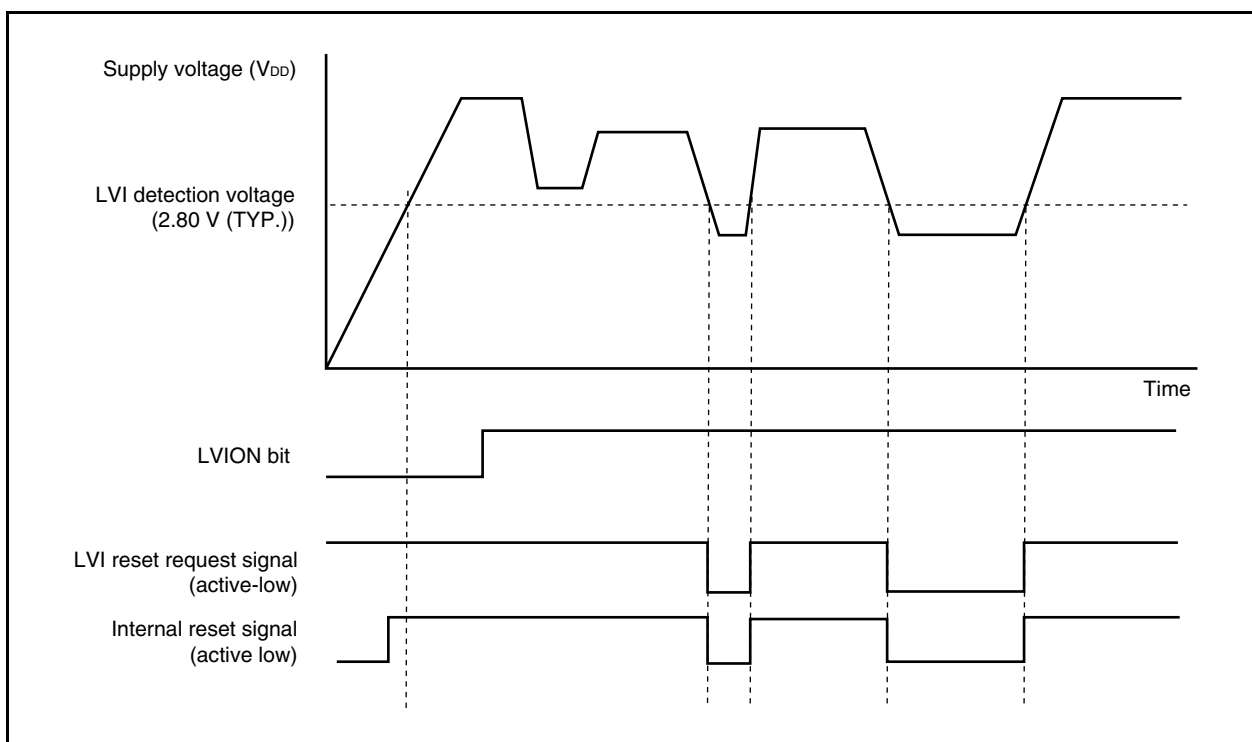
24.4.1 To use for internal reset signal

<To start operation>

- <1> Mask the interrupt of LVI.
- <2> Select the voltage to be detected by using the LVIS.LVIS0 bit.
- <3> Set the LVIM.LVION bit to 1 (to enable operation).
- <4> Insert a wait cycle of 0.2 ms or more by software.
- <5> By using the LVIM.LVIF bit, check if the supply voltage is lower than the detection voltage.
- <6> Set the LVIMD bit to 1 (to generate an internal reset signal).

Caution If the LVIMD bit is set to 1, the contents of the LVIM and LVIS registers cannot be changed until a reset request other than LVI is generated.

Figure 24-2. Timing of Low-Voltage Detector Operation (LVIMD Bit = 1, Low-Voltage Detection Level: 2.80 V)



24.4.2 To use for interrupt

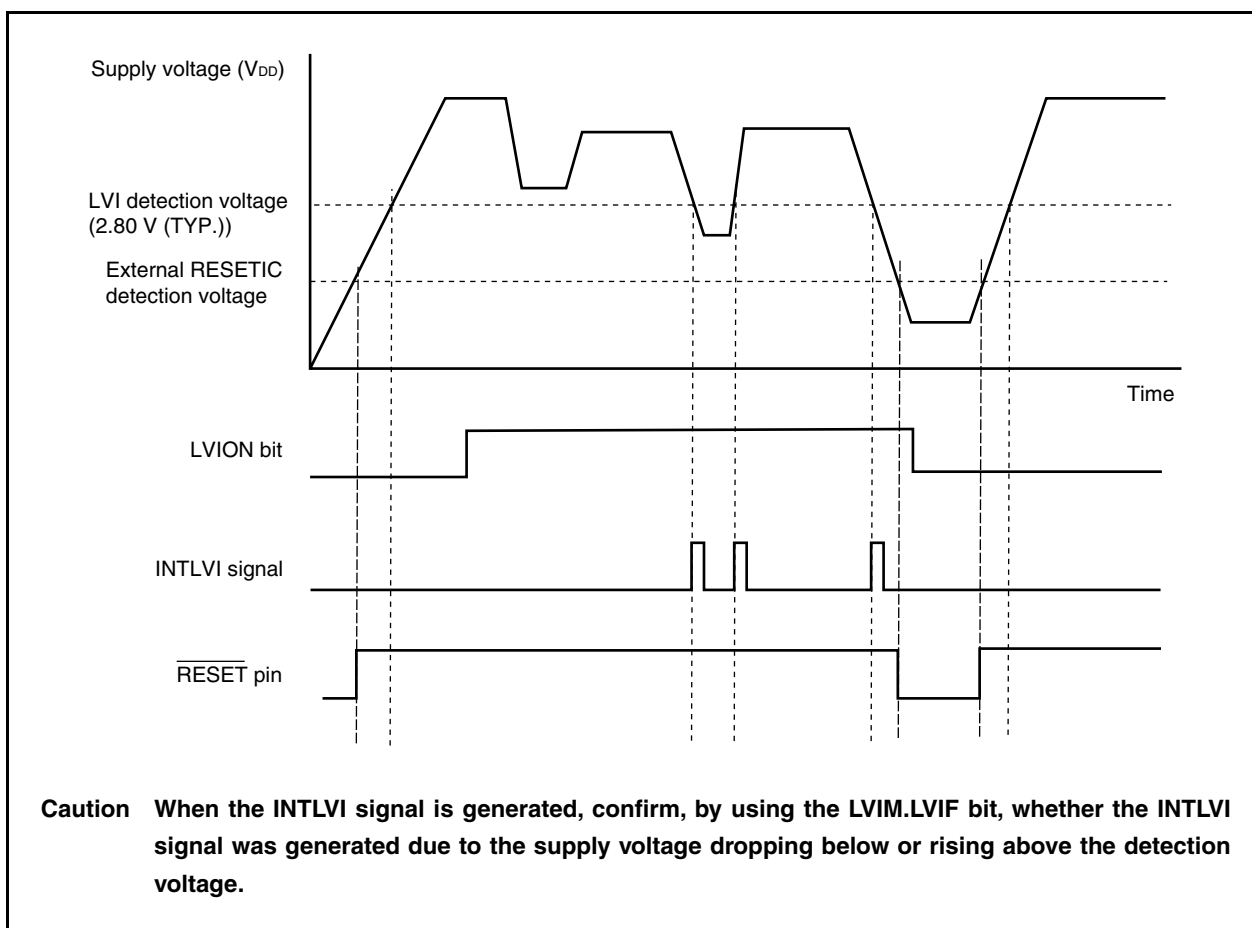
<To start operation>

- <1> Mask the interrupt of LVI.
- <2> Select the voltage to be detected by using the LVIS.LVIS0 bit.
- <3> Set the LVIM.LVION bit to 1 (to enable operation).
- <4> Insert a wait cycle of 0.2 ms (max.) or more by software.
- <5> By using the LVIM.LVIF bit, check if the supply voltage is higher than the detection voltage.
- <6> Clear the interrupt request flag of LVI.
- <7> Unmask the interrupt of LVI.

<To stop operation>

- <1> By using the LVIM.LVIF bit, check if the supply voltage is higher than the detection voltage.
- <2> Clear the LVION bit to 0.

Figure 24-3. Timing of Low-Voltage Detector Operation (LVIMD Bit = 0, Low-Voltage Detection Level: 2.80 V)



CHAPTER 25 CRC FUNCTION

25.1 Functions

- Generation of CRC (Cyclic Redundancy Check) code for detecting errors in communication data
- Generation of CRC code for detecting errors in data blocks
- Generation of 16-bit CRC code using a CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) generation polynomial for blocks of data of any length in 8-bit units
- CRC code is set to the CRC data register each time 1-byte data is transferred to the CRCIN register, after the initial value is set to the CRCD register.

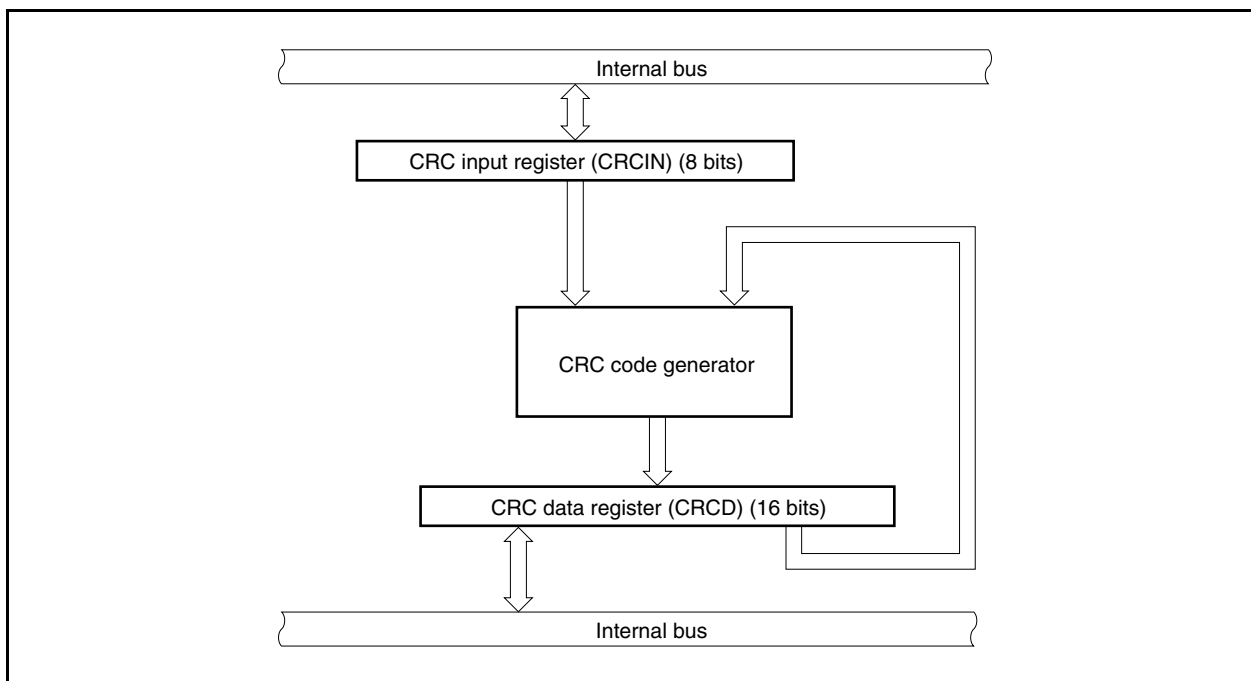
25.2 Configuration

The CRC function includes the following hardware.

Table 25-1. CRC Configuration

Item	Configuration
Control registers	CRC input register (CRCIN) CRC data register (CRCD)

Figure 25-1. Block Diagram of CRC Function



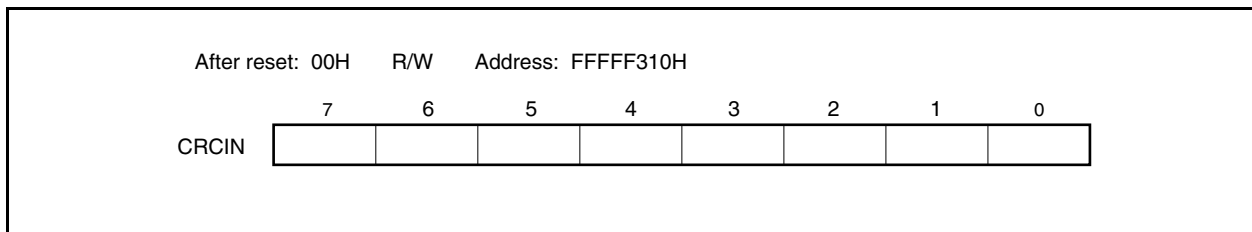
25.3 Registers

(1) CRC input register (CRCIN)

The CRCIN register is an 8-bit register for setting data.

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

Reset sets this register to 00H.



(2) CRC data register (CRCD)

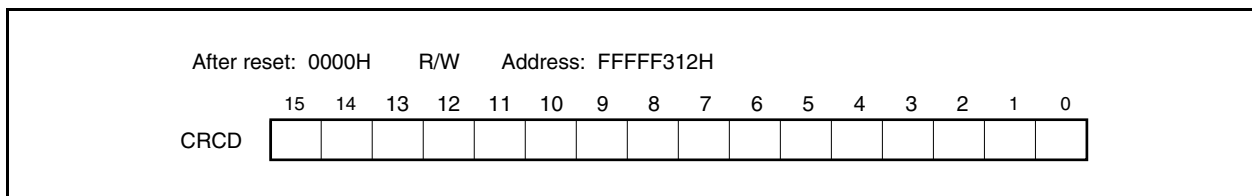
The CRCD register is a 16-bit register that stores the CRC-CCITT operation results.

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

Reset sets this register to 0000H.

Caution Accessing the CRCD register is prohibited in the following statuses. If a wait cycle is generated, it can only be cleared by a reset. For details, see 3.4.9 (1) Accessing specific on-chip peripheral I/O registers.

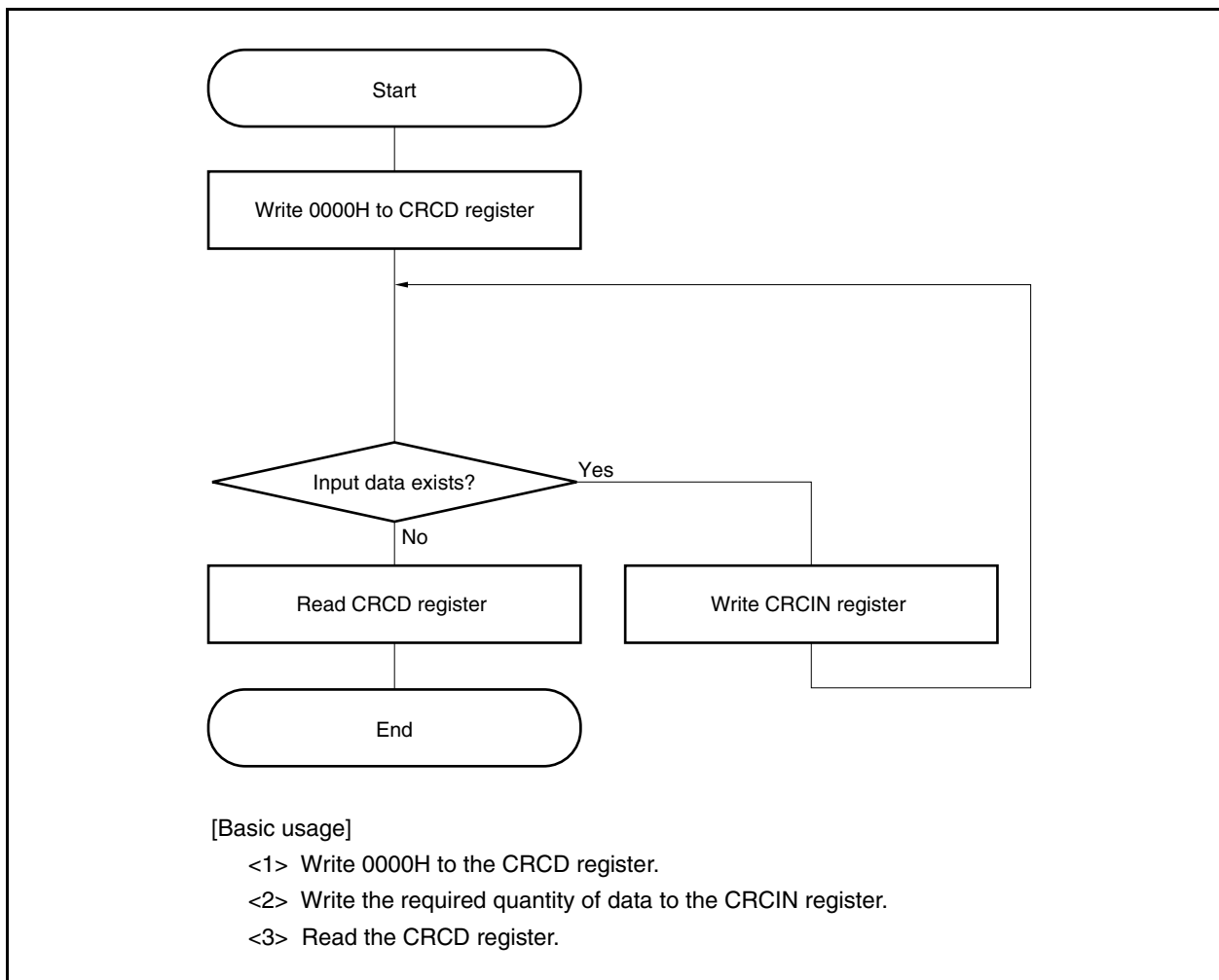
- When the CPU operates on the subclock and main clock oscillation is stopped
- When the CPU operates on the internal oscillator clock



25.5 Usage

How to use the CRC logic circuit is described below.

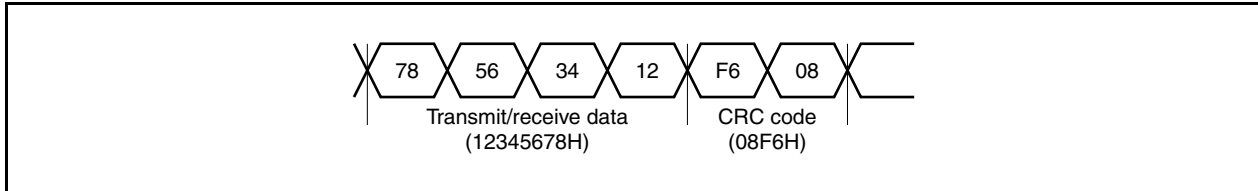
Figure 25-3. CRC Operation



Communication errors can easily be detected if the CRC code is transmitted/received along with transmit/receive data when transmitting/receiving data consisting of several bytes.

The following figure shows an example when all of the data 12345678H (0001 0010 0011 0100 0101 0110 0111 1000B) is transmitted LSB-first.

Figure 25-4. CRC Transmission Example



Processing on transmitting side

- <1> Write the initial value 0000H to the CRCD register.
- <2> Write the 1 byte of data to be transmitted first to the transmit buffer register. (At this time, also write the same data to the CRCIN register.)
- <3> When transmitting several bytes of data, write the same data to the CRCIN register each time transmit data is written to the transmit buffer register.
- <4> After all the data has been transmitted, write the contents of the CRCD register (CRC code) to the transmit buffer register and transmit them. (The data is transmitted LSB-first, starting from the lower bytes, and then the higher bytes.)
- <5> If a resend is requested by the transmitting side, resend the data.

Processing on receiving side

- <1> Write the initial value 0000H to the CRCD register.
- <2> When reception of the first 1 byte of data is complete, write that receive data to the CRCIN register.
- <3> If receiving several bytes of data, write the receive data to the CRCIN register every time reception ends. (In the case of normal reception, when all the receive data has been written to the CRCIN register, the contents of the CRCD register on the receiving side and the contents of the CRCD register on the transmitting side are the same.)
- <4> Next, the CRC code is transmitted from the transmitting side, so write this data to the CRCIN register similarly to receive data.
- <5> When reception of all the data, including the CRC code, has been completed, reception was normal if the contents of the CRCD register are 0000H. If the contents of the CRCD register are other than 0000H, this indicates a communication error, so transmit a resend request to the transmitting side.

CHAPTER 26 REGULATOR

26.1 Outline

The V850ES/JC3-L and V850ES/JE3-L include a regulator to reduce power consumption and noise.

This regulator supplies a stepped-down V_{DD} power supply voltage to the oscillator block and internal logic circuits (except the A/D converter, D/A converter, and output buffers).

Figure 26-1. Regulator (V850ES/JC3-L (40-pin))

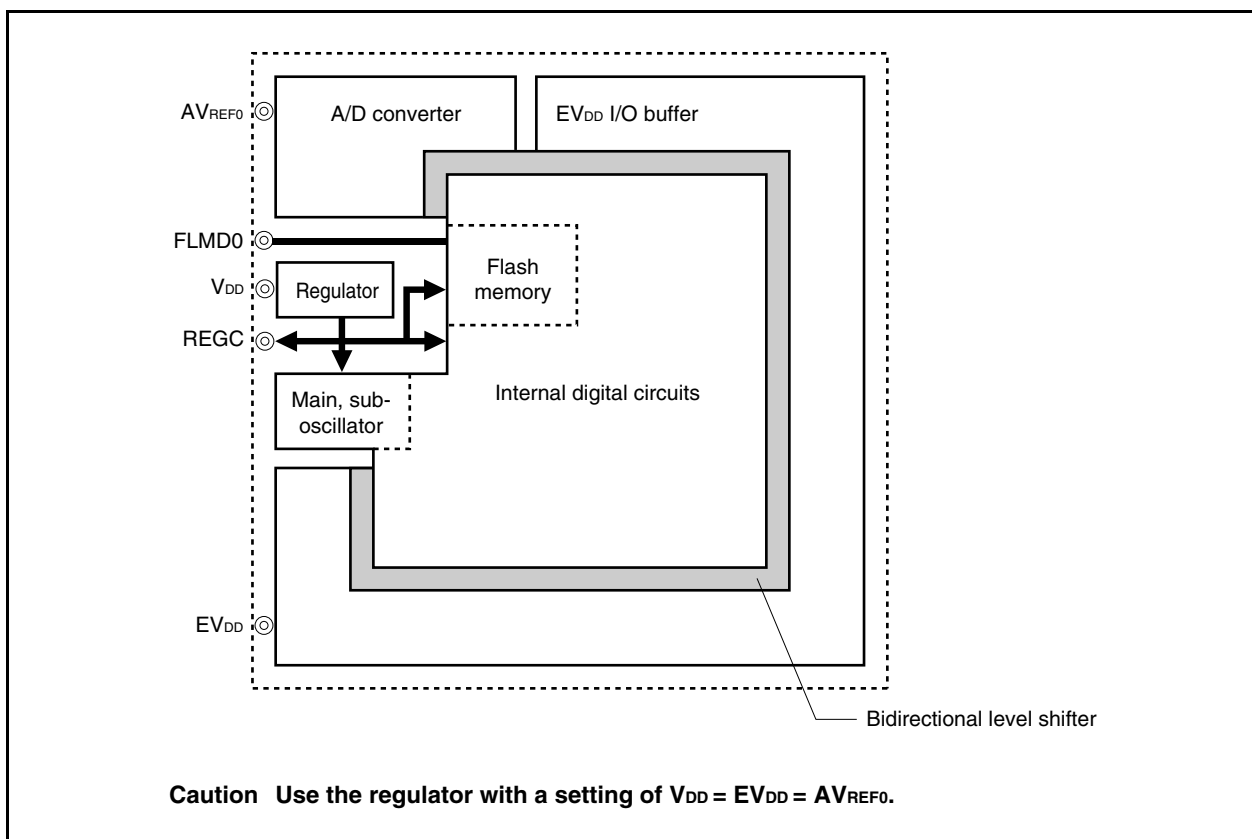
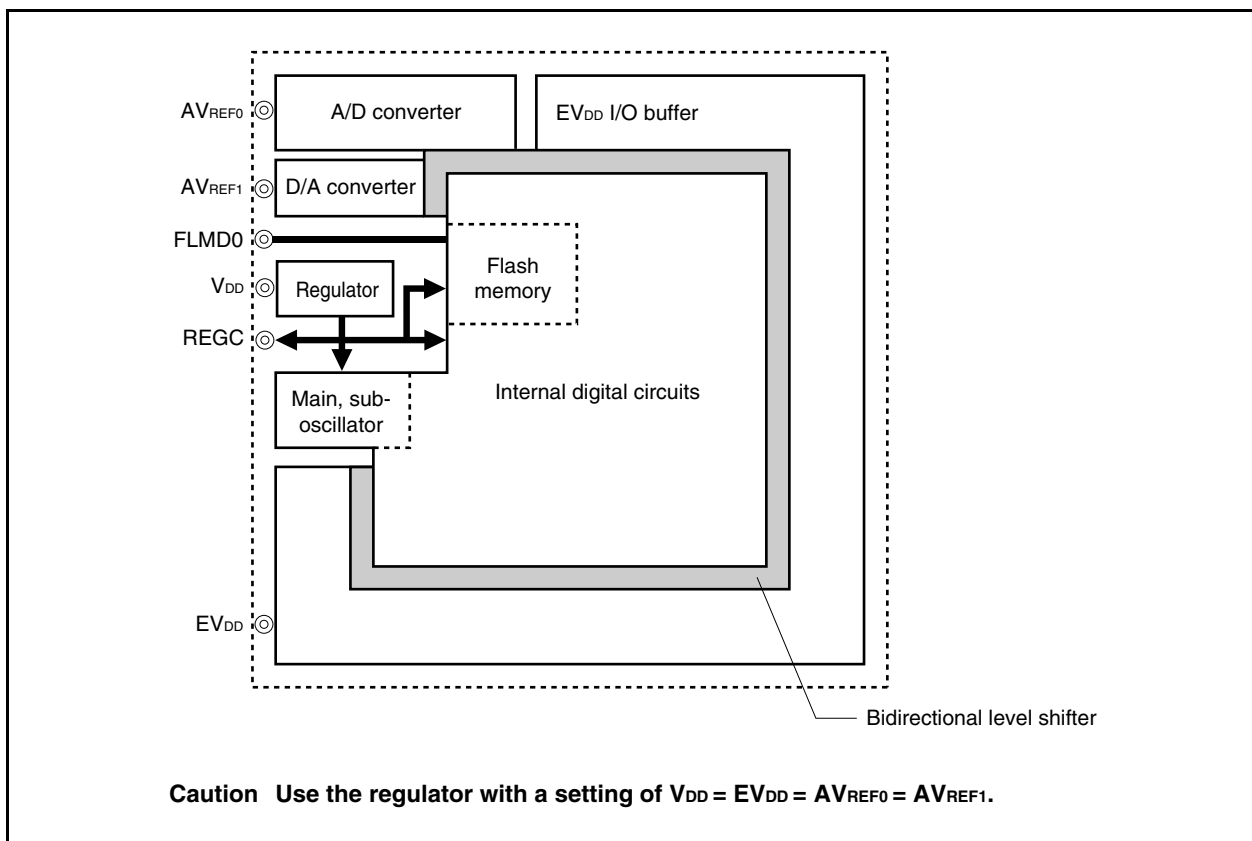


Figure 26-2. Regulator (V850ES/JC3-L (48-pin), V850ES/JE3-L)



26.2 Operation

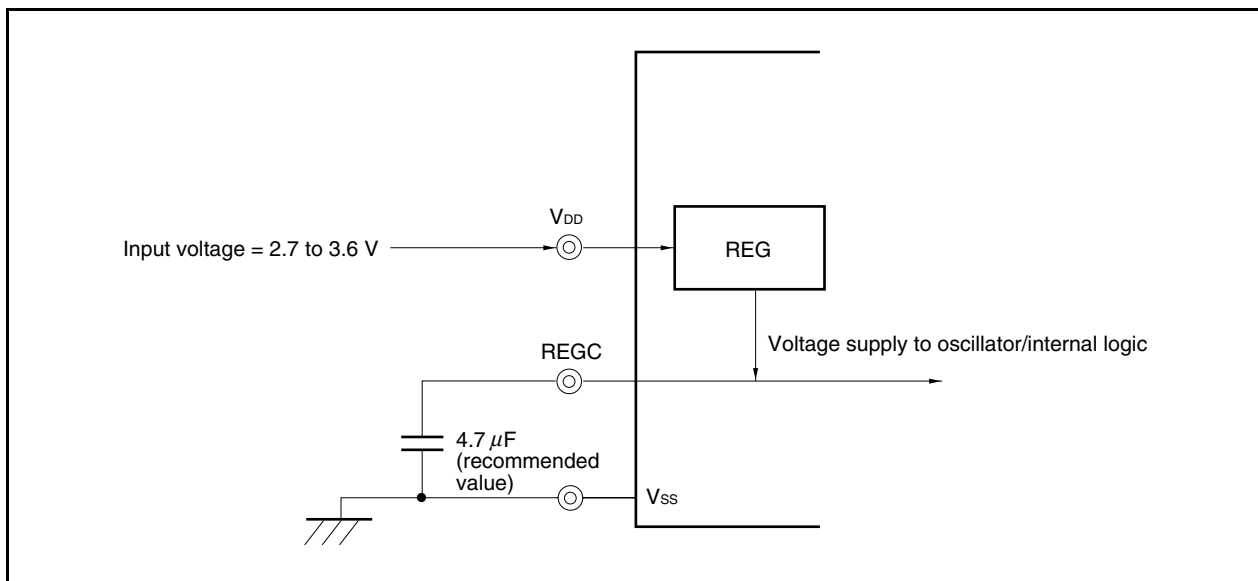
The regulator connected to V_{DD} always operates in all modes (normal operation mode, HALT mode, IDLE1 mode, IDLE2 mode, STOP mode, subclock operation mode, sub-IDLE mode, or during reset).

The output voltage of the regulator can be lowered in the STOP mode, subclock operation mode, and sub-IDLE mode to reduce the power consumption. For details, see **CHAPTER 21 STANDBY FUNCTION**.

Be sure to connect a capacitor (4.7 μF (recommended value)) to the REGC pin to stabilize the regulator output.

A diagram of the regulator pin connection method is shown below.

Figure 26-3. REGC Pin Connection



CHAPTER 27 OPTION BYTE

The option byte is stored at address 000007AH of the internal flash memory (internal ROM area) as 8-bit data. This 8-bit data is used to specify the oscillation stabilization time immediately after a reset ends. After a reset ends, the oscillation stabilization time is secured, in accordance with these set values.

When writing a program to the V850ES/JC3-L, V850ES/JE3-L, specify the option data at address 000007AH in the program, referring to **27.1 Program Example**.

The data in this area cannot be rewritten during program execution.

Address: 000007AH

7	6	5	4	3	2	1	0
WDTMD1	RMOPIIN	0	0	0	RESOSTS2	RESOSTS1	RESOSTS0

RES OSTS2	RES OSTS1	RES OSTS0	Selection of oscillation stabilization time (theoretical value)	fx		
				2.5 MHz	6 MHz	10 MHz
				0	0	0
0	0	1	$2^{11}/f_x$	819.2 μ s	Setting prohibited	Setting prohibited
0	1	0	$2^{12}/f_x$	1.638 ms	682.7 μ s	409.6 μ s
0	1	1	$2^{13}/f_x$	3.277 ms	1.365 ms	819.2 μ s
1	0	0	$2^{14}/f_x$	6.554 ms	2.731 ms	1.638 ms
1	0	1	$2^{15}/f_x$	13.11 ms	5.461 ms	3.277 ms
1	1	0	$2^{16}/f_x$	26.21 ms	10.92 ms	6.554 ms
1	1	1	$2^{16}/f_x$	26.21 ms	10.92 ms	6.554 ms

Remarks 1. The wait time after releasing the STOP mode or IDLE2 mode is set by the OSTs register. For details of the OSTs register, see **21.2 (3) Oscillation stabilization time select register (OSTS)**.

2. fx: Main clock oscillation frequency

Cautions 1. The actual oscillation stabilization time is longer than the theoretical value because the overhead time after power-on is taken into consideration. The actual oscillation stabilization time is the time shown above, plus up to 260 μ s.

2. Be sure to select an oscillation stabilization time (theoretical value) of 400 μ s or longer. If it is set to less than 400 μ s, the internal status becomes unstable and the operation cannot be guaranteed.

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

27.1 Program Example

The following shows program examples when the CA850 is used.

```
#-----  
# OPTION_BYTES  
#-----  
.section "OPTION_BYTES" //Specifies the option byte at address 0000007A.//  
.byte 0b00000001 -- 0x7a //Specifies 0b00000001 as the option byte.//  
.byte 0b00000000 -- 0x7b //Specifies 0b00000000 at address 0000007B.//  
.byte 0b00000000 -- 0x7c //Specifies 0b00000000 at address 0000007C.//  
.byte 0b00000000 -- 0x7d //Specifies 0b00000000 at address 0000007D.//  
.byte 0b00000000 -- 0x7e //Specifies 0b00000000 at address 0000007E.//  
.byte 0b00000000 -- 0x7f //Specifies 0b00000000 at address 0000007F.//
```

Caution Be sure to specify 6 option bytes in this section. If less than 6 bytes are specified, an error occurs when linking is executed.

Error message: F4112: illegal "OPTION_BYTES" section size.

Remark Set 0x00 to addresses 007BH to 007FH.

CHAPTER 28 FLASH MEMORY

The V850ES/JC3-L and V850ES/JE3-L incorporate flash memory.

- μ PD70F3797, 70F3801, 70F3805: 16 KB of flash memory
- μ PD70F3798, 70F3802, 70F3806: 32 KB of flash memory
- μ PD70F3799, 70F3803, 70F3807: 64 KB of flash memory
- μ PD70F3800, 70F3804, 70F3808: 128 KB of flash memory
- μ PD70F3838, 70F3839, 70F3807: 256 KB of flash memory

Flash memory versions offer the following advantages for development environments and mass production applications.

- For altering software after the V850ES/JC3-L and V850ES/JE3-L are soldered onto the target system.
- For data adjustment when starting mass production.
- For differentiating software according to the specification in small scale production of various models.
- For facilitating inventory management.
- For updating software after shipment.

28.1 Features

- Capacity: 256/128/64/32/16 KB
- Rewriting method
 - Rewriting by communication with dedicated flash memory programmer via serial interface (on-board/off-board programming)
 - Rewriting flash memory by user program (self programming)
- Flash memory write prohibit function supported (security function)
- Safe rewriting of entire flash memory area by self programming using boot swap function
- Interrupts can be acknowledged during self programming.
- 4-byte/1-clock access (when instruction is fetched)

28.2 Memory Configuration

The V850ES/JC3-L, V850ES/JE3-L internal flash memory area is divided into 8 or 16 or 32 or 64 or 128 blocks and can be erased in block units. All the blocks can also be erased at once.

When the boot swap function is used, the physical memory located at the addresses of blocks 0 to 7 is replaced by the physical memory located at the addresses of blocks 8 to 15. For details of the boot swap function, see **28.5 Rewriting by Self Programming**.

Figure 28-1. Flash Memory Mapping (1/2)

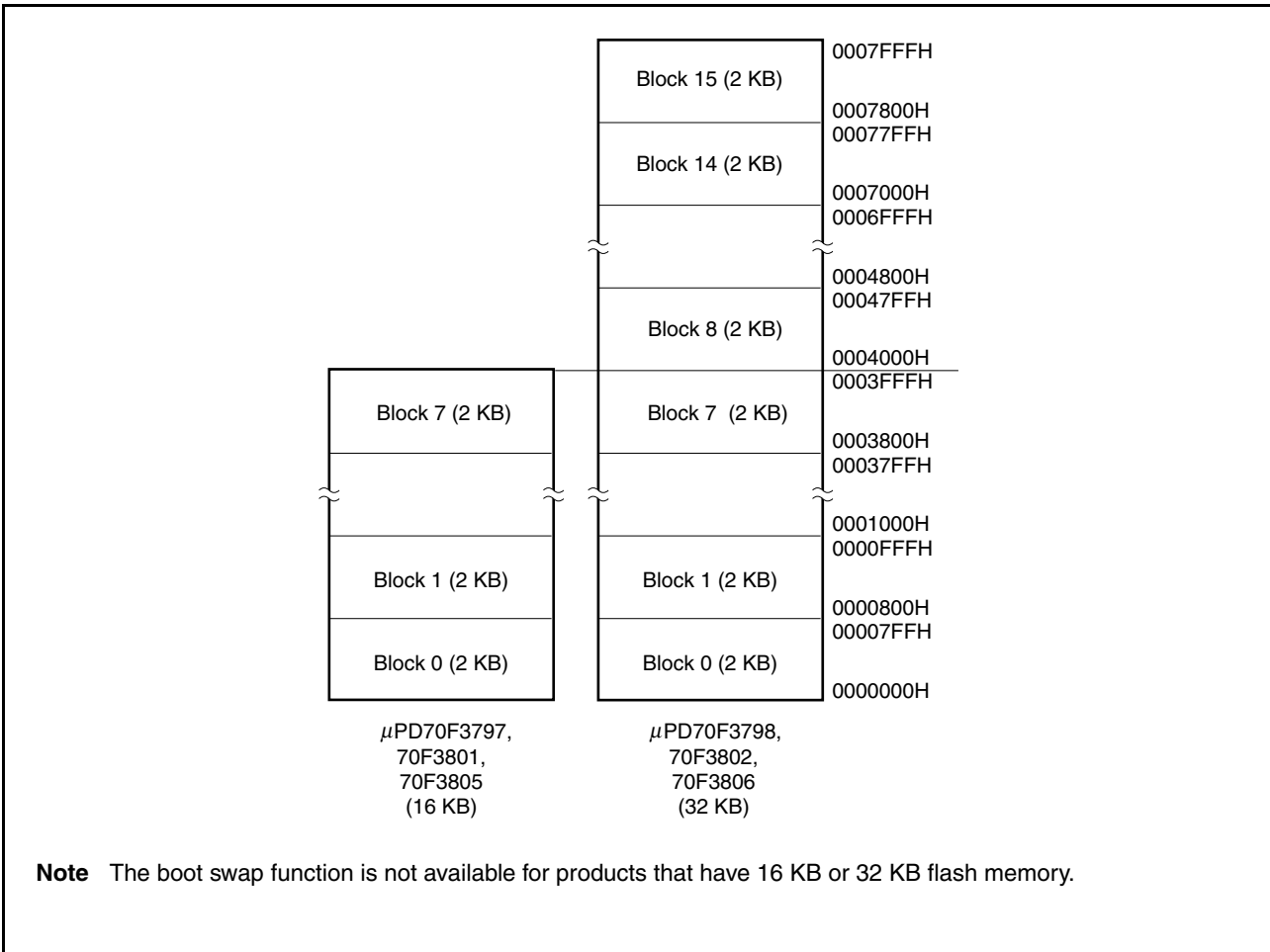
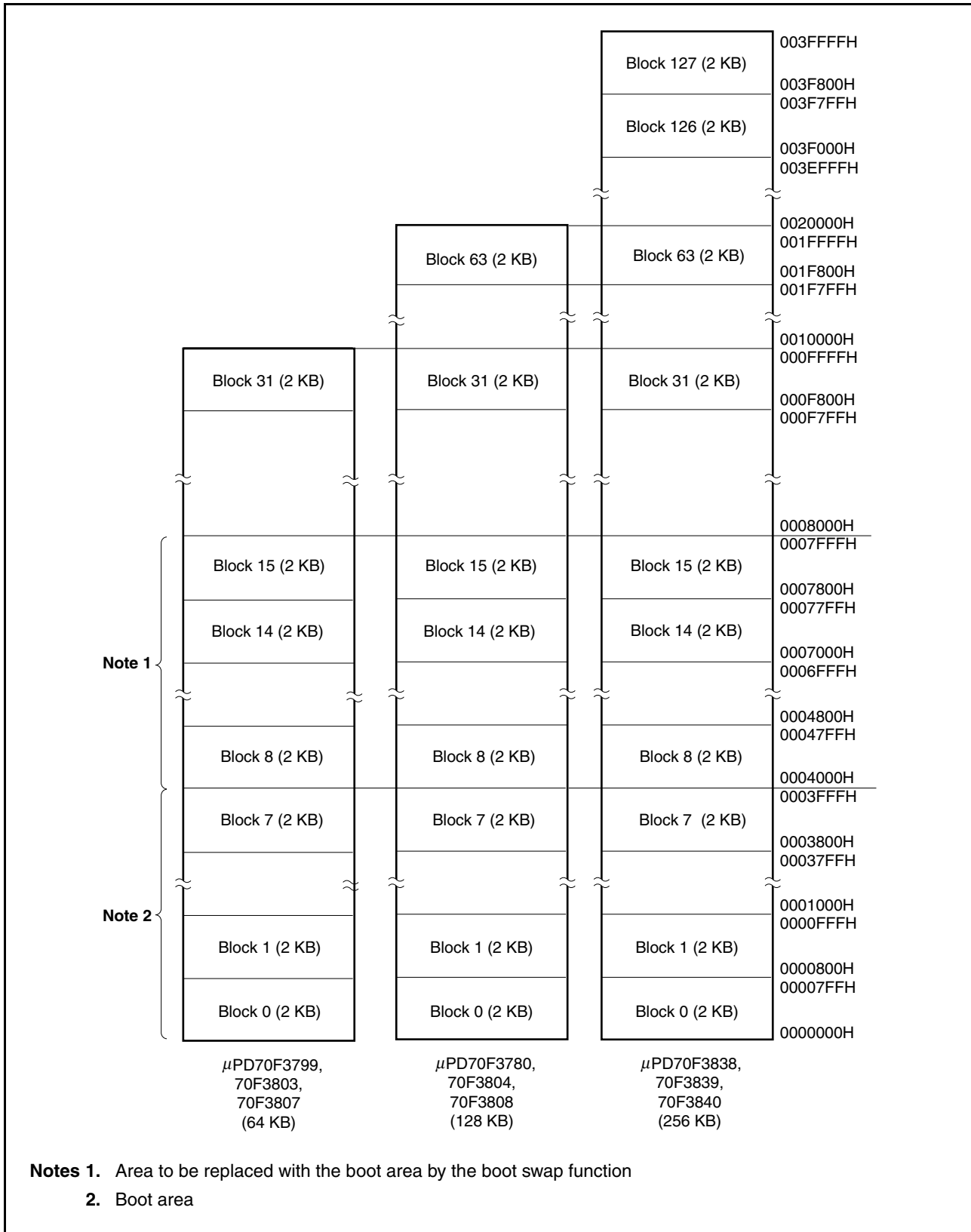


Figure 28-1. Flash Memory Mapping (2/2)



28.3 Functional Outline

The internal flash memory of the V850ES/JC3-L, V850ES/JE3-L can be rewritten by using the rewrite function of the dedicated flash memory programmer, regardless of whether the V850ES/JC3-L, V850ES/JE3-L has already been mounted on the target system or not (off-board/on-board programming).

In addition, a security function that prohibits rewriting the user program written to the internal flash memory is also supported, so that the program cannot be changed by an unauthorized person.

The rewrite function using the user program (self programming) is ideal for an application where it is assumed that the program will be changed after production/shipment of the target system. A boot swap function that rewrites the entire flash memory area safely is also supported. In addition, interrupt servicing can be executed during self programming, so that the flash memory can be rewritten under various conditions, such as while communicating with an external device.

Table 28-1. Rewrite Method

Rewrite Method	Functional Outline	Operation Mode
On-board programming	Flash memory can be rewritten after the device is mounted on the target system, by using a dedicated flash memory programmer.	Flash memory programming mode
Off-board programming	Flash memory can be rewritten before the device is mounted on the target system, by using a dedicated flash memory programmer and a dedicated program adapter board (FA series).	
Self programming	Flash memory can be rewritten by executing a user program that has been written to the flash memory in advance by means of off-board/on-board programming. (During self-programming, instructions cannot be fetched from or data access cannot be made to the internal flash memory area. Therefore, the rewrite program must be transferred to the internal RAM in advance.)	Normal operation mode

Remark The FA series is a product of Naito Densai Machida Mfg. Co., Ltd.

Table 28-2. Basic Functions

Function	Functional Outline	Support (√: Supported, ×: Not supported)	
		On-Board/Off-Board Programming	Self Programming
Blank check	The erasure status of the entire memory is checked.	√	√
Chip erasure	The contents of the entire memory area are erased all at once.	√	× ^{Note}
Block erasure	The contents of specified memory blocks are erased.	√	√
Program	Writing to specified addresses, and a verify check to see if the write level is secured, are performed.	√	√
Verify/checksum	Data read from the flash memory is compared with data transferred from the flash memory programmer.	√	× (Can be read by user program)
Read	Data written to the flash memory is read.	√	×
Security setting	Use of the block erase command, chip erase command, program command, and read command, and boot area rewrite, are prohibited.	√	× (Supported only when setting is changed from enable to disable)

Note This is possible by selecting the entire memory area for the block erase function.

The following table lists the security functions. The chip erase command prohibit, block erase command prohibit, program command prohibit, read command prohibit, and boot block cluster rewrite prohibit functions are enabled by default after shipment, and security settings can be specified only by rewriting via on-board/off-board programming. Each security function can be used in combination with the others at the same time.

Table 28-3. Security Functions

Function	Function Outline
Chip erase command prohibit	Execution of block erase and chip erase commands on all the blocks is prohibited. Once prohibition is set, setting of prohibition cannot be initialized because the chip erase command cannot be executed.
Block erase command prohibit	Execution of a block erase command on all blocks is prohibited. Setting of prohibition can be initialized by execution of a chip erase command.
Program command prohibit	Execution of program and block erase commands on all the blocks are prohibited. Setting of prohibition can be initialized by execution of the chip erase command.
Read command prohibit	Execution of read command on all the blocks is prohibited. Setting of prohibition can be initialized by execution of the chip erase command.
Boot block cluster rewrite prohibit	<p>Boot block clusters from block 0 to the specified last block can be protected. The protected boot block clusters cannot be rewritten (erased and written). Setting of prohibition cannot be initialized by execution of the chip erase command.</p> <p>The following can be specified as the last block:</p> <ul style="list-style-type: none"> μPD70F3797, 70F3801, 70F3805: block 7 μPD70F3798, 70F3802, 70F3806: block 15 μPD70F3799, 70F3803, 70F3807: block 31 μPD70F3800, 70F3804, 70F3808: block 63 μPD70F3838, 70F3839, 70F3840: block 127

Table 28-4. Security Settings

Function	Erase, Write, Read Operations When Each Security Is Set (√: Executable, ×: Not Executable, -: Not Supported)		Notes on Security Setting	
	On-Board/ Off-Board Programming	Self Programming	On-Board/ Off-Board Programming	Self Programming
Chip erase command prohibit	Chip erase command: × Block erase command: × Program command: √ ^{Note 1} Read command: √	Chip erasure: - Block erasure (FlashBlockErase): √ Write (FlashWordWrite): √ Read (FlashWordRead): √	Setting of prohibition cannot be initialized.	Supported only when setting is changed from enable to prohibit.
Block erase command prohibit	Chip erase command: √ Block erase command: × Program command: √ Read command: √	Chip erasure: - Block erasure (FlashBlockErase): √ Write (FlashWordWrite): √ Read (FlashWordRead): √	Setting of prohibition can be initialized by chip erase command.	
Program command prohibit	Chip erase command: √ Block erase command: × Program command: × Read command: √	Chip erasure: - Block erasure (FlashBlockErase): √ Write (FlashWordWrite): √ Read (FlashWordRead): √	Setting of prohibition can be initialized by chip erase command.	
Read command prohibit	Chip erase command: √ Block erase command: √ Program command: √ Read command: ×	Chip erasure: - Block erasure (FlashBlockErase): √ Write (FlashWordWrite): √ Read (FlashWordRead): √	Setting of prohibition can be initialized by chip erase command.	
Boot area rewrite prohibit	Chip erase command: × Block erase command: × ^{Note 2} Program command: × ^{Note 2} Read command: √	Chip erasure: - Block erasure (FlashBlockErase): × ^{Note 2} Write (FlashWordWrite): × ^{Note 2} Read (FlashWordRead): √	Setting of prohibition cannot be initialized.	

Notes 1. In this case, since the erase command is invalid, data that differs from the data already written in the flash memory cannot be written.

2. This can be executed for other than boot block clusters.

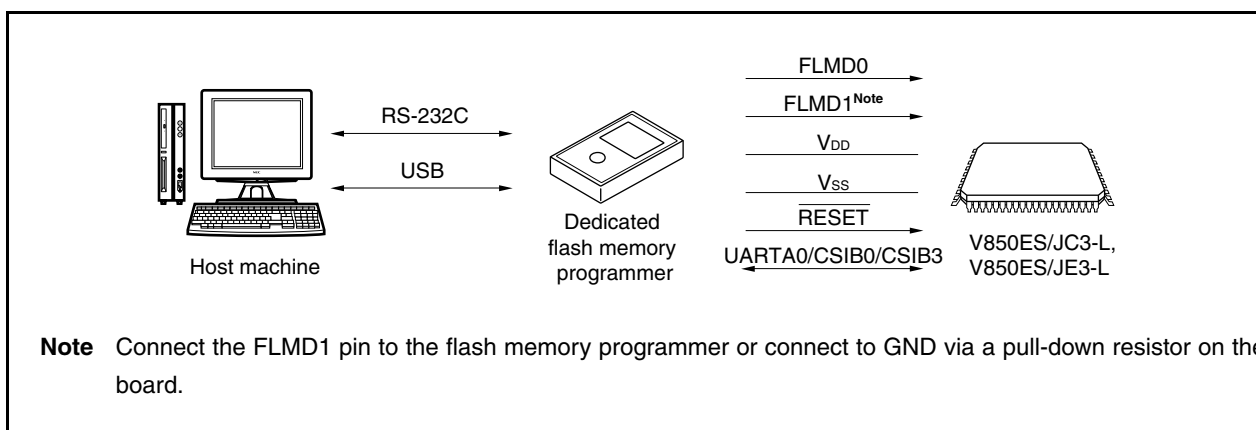
28.4 Rewriting by Dedicated Flash Memory Programmer

The flash memory can be rewritten by using a dedicated flash memory programmer after the V850ES/JC3-L, V850ES/JE3-L are mounted on the target system (on-board programming). By combining the dedicated flash memory programmer with a dedicated program adapter (FA series), the flash memory can also be rewritten before the device is mounted on the target system (off-board programming).

28.4.1 Programming environment

The following shows the environment required for writing programs to the flash memory of the V850ES/JC3-L, V850ES/JE3-L.

Figure 28-2. Environment Required for Writing Programs to Flash Memory



A host machine is required for controlling the dedicated flash memory programmer. In some cases, however, it can be used stand-alone. For details, see the user's manual of the dedicated flash memory programmer.

UARTA0, CSIB0, or CSIB3 is used for the interface between the dedicated flash memory programmer and the V850ES/JC3-L, V850ES/JE3-L to perform writing, erasing, etc. A dedicated program adapter (FA series) is required for off-board writing.

Remark The FA series is a product of Naito Densai Machida Mfg. Co., Ltd.

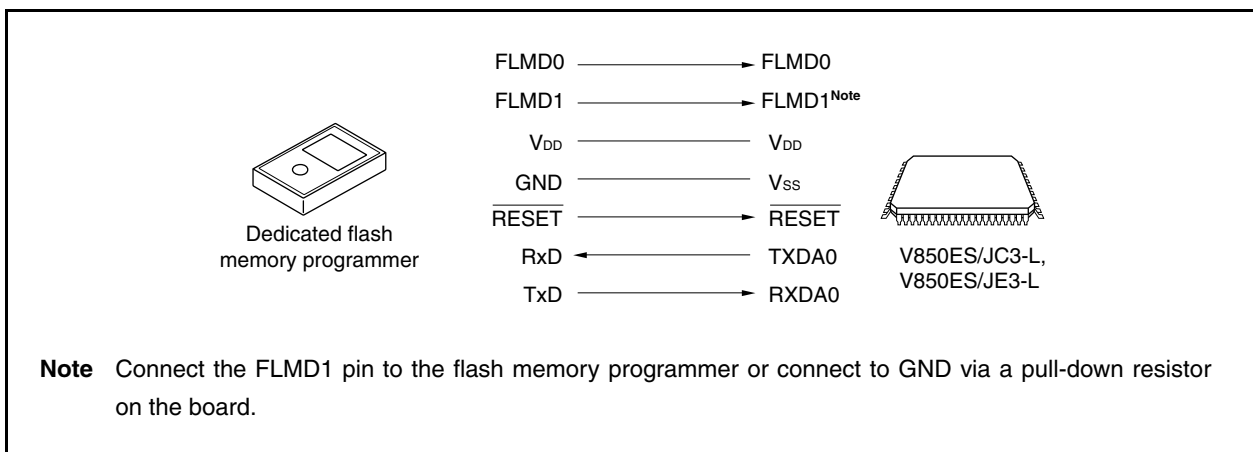
28.4.2 Communication mode

Communication between the dedicated flash memory programmer and the V850ES/JC3-L and V850ES/JE3-L are performed by serial communication using the UARTA0, CSIB0, or CSIB3 interfaces of the V850ES/JC3-L and V850ES/JE3-L.

(1) UARTA0

Transfer rate: 9,600 to 153,600 bps

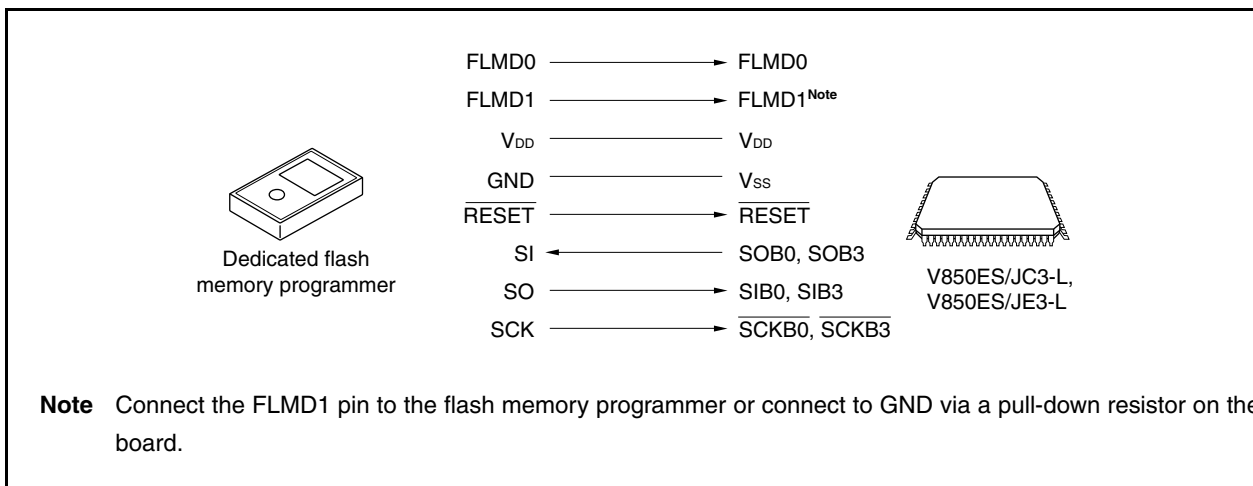
Figure 28-3. Communication with Dedicated Flash Memory Programmer (UARTA0)



(2) CSIB0, CSIB3

Serial clock: 2.4 kHz to 5 MHz (MSB first)

Figure 28-4. Communication with Dedicated Flash Memory Programmer (CSIB0, CSIB3)

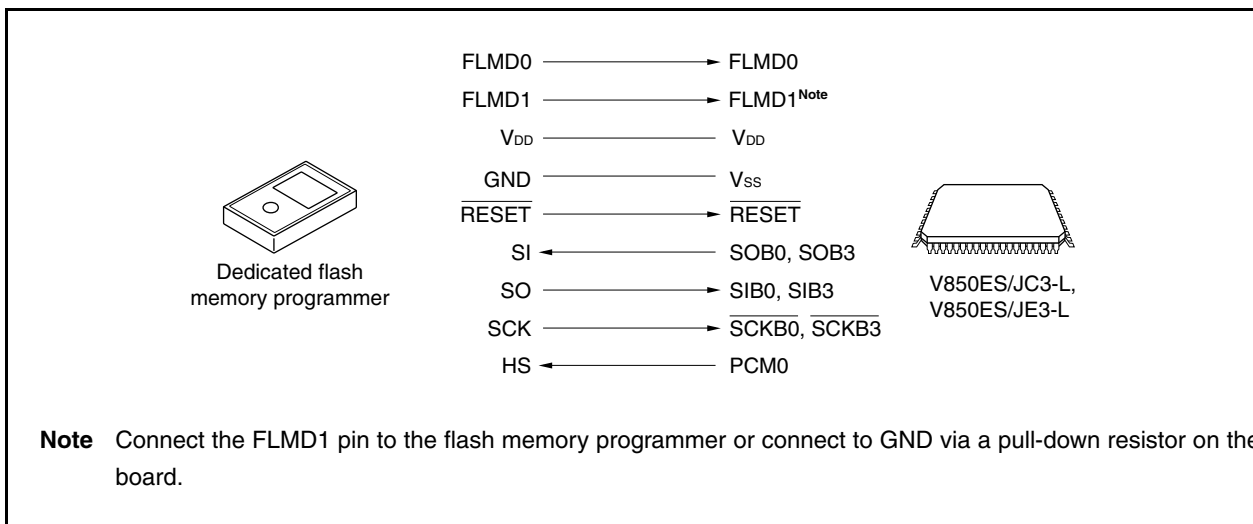


(3) CSIB0 + HS, CSIB3 + HS

Serial clock: 2.4 kHz to 5 MHz (MSB first)

The V850ES/JE3-L operates as a slave (V850ES/JC3-L does not use).

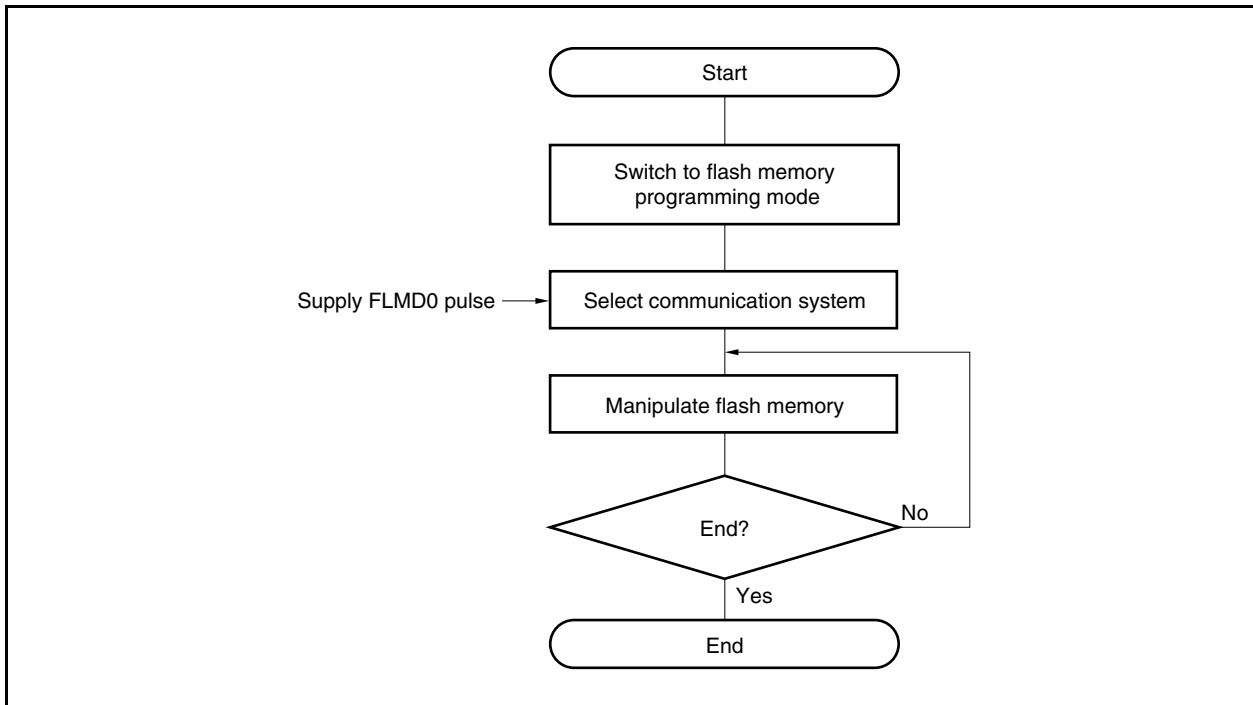
Figure 28-5. Communication with Dedicated Flash Memory Programmer (CSIB0 + HS, CSIB3 + HS)



28.4.3 Flash memory control

The following shows the procedure for manipulating the flash memory.

Figure 28-6. Procedure for Manipulating Flash Memory

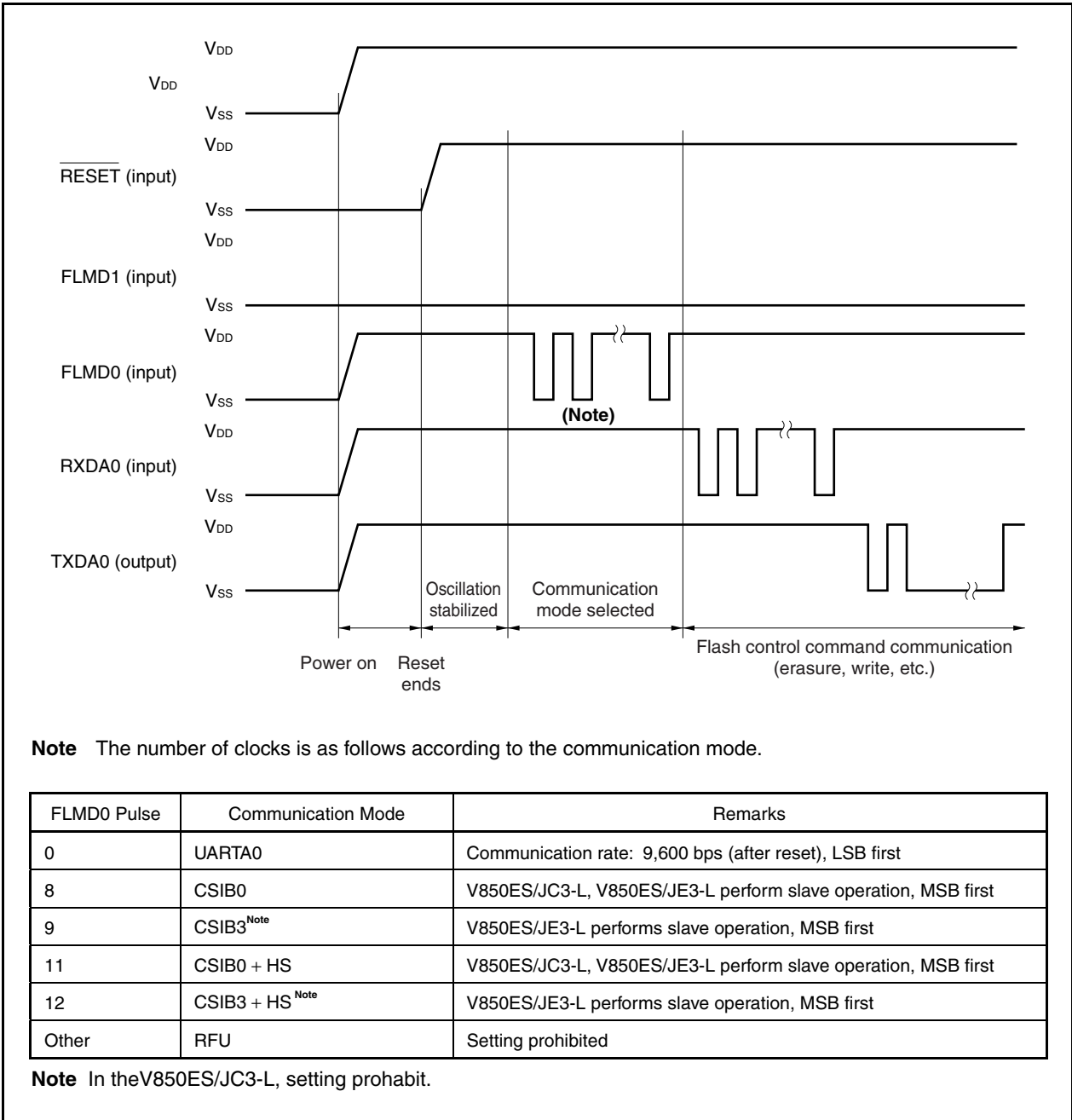


28.4.4 Selection of communication mode

In the V850ES/JC3-L and V850ES/JE3-L, the communication mode is selected by inputting pulses (12 pulses max.) to the FLMD0 pin after switching to the flash memory programming mode. The FLMD0 pulse is generated by the dedicated flash memory programmer.

The following shows the relationship between the number of pulses and the communication mode.

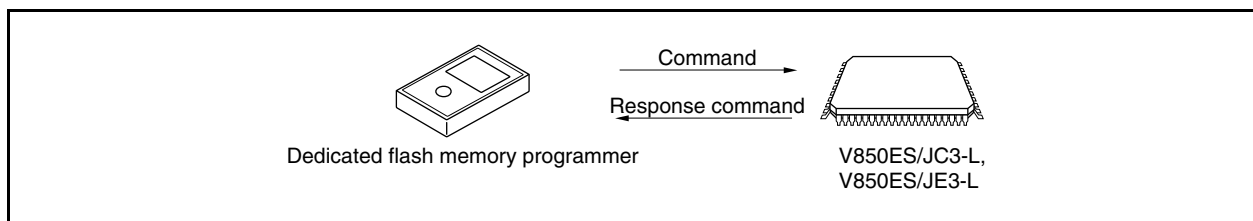
Figure 28-7. Selection of Communication Mode



28.4.5 Communication commands

The V850ES/JC3-L and V850ES/JE3-L communicate with the dedicated flash memory programmer by means of commands. The signals sent from the dedicated flash memory programmer to the V850ES/JC3-L and V850ES/JE3-L are called “commands”. The response signals sent from the V850ES/JC3-L and V850ES/JE3-L to the dedicated flash memory programmer are called “response commands”.

Figure 28-9. Communication Commands



The following shows the commands for flash memory control in the V850ES/JC3-L and V850ES/JE3-L. All of these commands are issued from the dedicated flash memory programmer, and the V850ES/JC3-L and V850ES/JE3-L perform the processing corresponding to the commands.

Table 28-7. Flash Memory Control Commands

Classification	Command Name	Support			Function
		CSIB0, CSIB3	CSIB0 + HS, CSIB3 + HS	UARTA0	
Blank check	Block blank check command	√	√	√	Checks if the contents of the memory in the specified block have been correctly erased.
Erase	Chip erase command	√	√	√	Erases the contents of the entire memory.
	Block erase command	√	√	√	Erases the contents of the memory of the specified block.
Write	Program command	√	√	√	Writes the specified address range, and executes a contents verify check.
Verify	Verify command	√	√	√	Compares the contents of memory in the specified address range with data transferred from the flash memory programmer.
	Checksum command	√	√	√	Reads the checksum in the specified address range.
Read	Read command	√	√	√	Reads the data written to flash memory.
System setting, control	Silicon signature command	√	√	√	Reads silicon signature information.
	Security setting command	√	√	√	Prohibits the chip erase command, block erase command, program command, read command, and boot area rewrite.

28.4.6 Pin connection in on-board programming

When performing on-board writing, mount a connector on the target system to connect to the dedicated flash memory programmer.

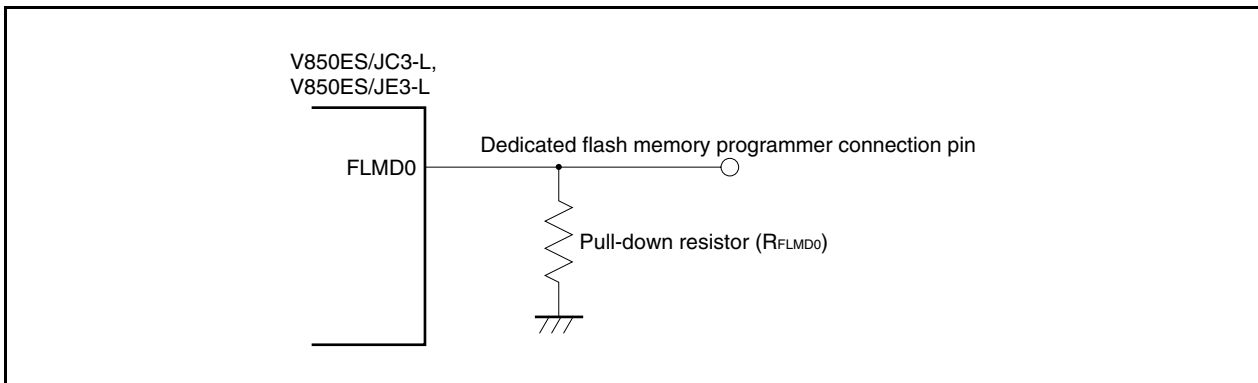
In the flash memory programming mode, all the pins not used for flash memory programming become the same status as that immediately after reset. Therefore, pin handling is required when the external device does not acknowledge the status immediately after a reset.

(1) FLMD0 pin

In the normal operation mode, input a voltage of V_{SS} level to the FLMD0 pin. In the flash memory programming mode, supply a write voltage of V_{DD} level to the FLMD0 pin.

Because the FLMD0 pin serves as a write protection pin in the self programming mode, a voltage of V_{DD} level must be supplied to the FLMD0 pin via port control, etc., before writing to the flash memory. For details, see **28.5.5 (1) FLMD0 pin**.

Figure 28-9. FLMD0 Pin Connection Example



(2) FLMD1 pin

When 0 V is input to the FLMD0 pin, the FLMD1 pin does not function. When V_{DD} is supplied to the FLMD0 pin, the flash memory programming mode is entered, so 0 V must be input to the FLMD1 pin. The following shows an example of the connection of the FLMD1 pin.

Figure 28-10. FLMD1 Pin Connection Example

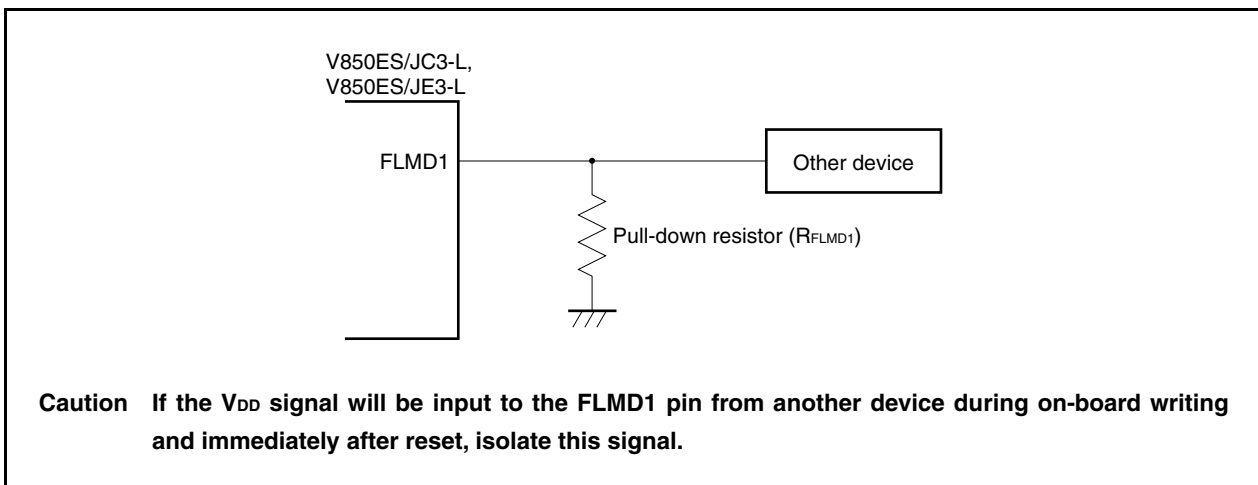


Table 28-8. Relationship Between FLMD0 and FLMD1 Pins and Operation Mode Immediately After Reset Ends

FLMD0	FLMD1	Operation Mode
0	Don't care	Normal operation mode
V _{DD}	0	Flash memory programming mode
V _{DD}	V _{DD}	Setting prohibited

(3) Serial interface pin

The following shows the pins used by each serial interface.

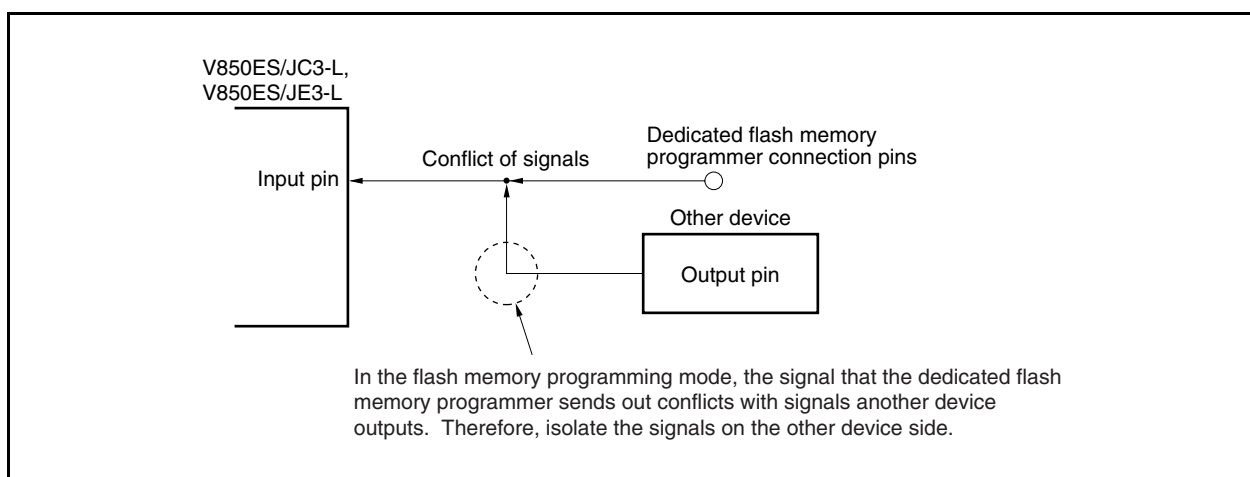
Table 28-9. Pins Used by Serial Interfaces

Serial Interface	Pins Used
UARTA0	TXDA0, RXDA0
CSIB0	SOB0, SIB0, $\overline{\text{SCKB0}}$
CSIB3	SOB3, SIB3, $\overline{\text{SCKB3}}$
CSIB0 + HS	SOB0, SIB0, $\overline{\text{SCKB0}}$, PCM0
CSIB3 + HS	SOB3, SIB3, $\overline{\text{SCKB3}}$, PCM0

When connecting a dedicated flash memory programmer to a serial interface pin that is connected to another device on-board, care should be taken to avoid conflict of signals and malfunction of the other device.

(a) Conflict of signals

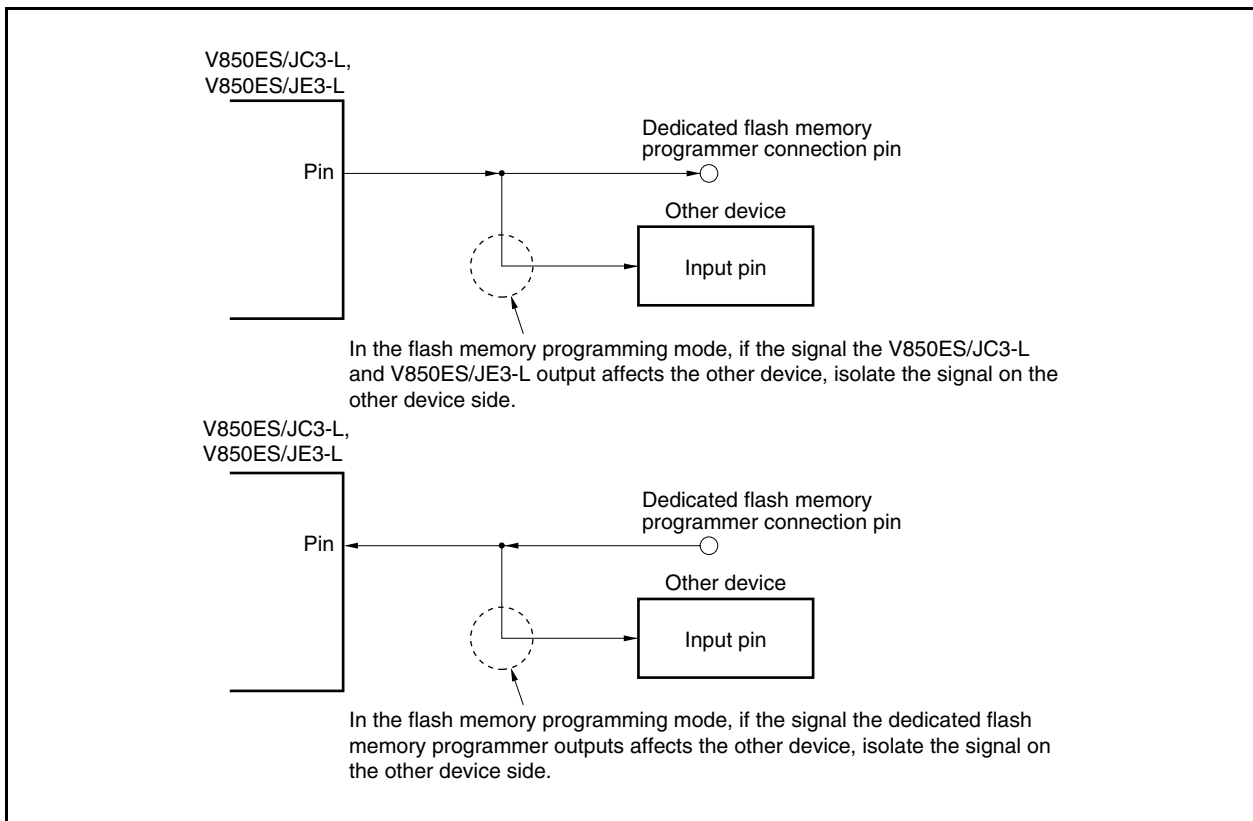
When the dedicated flash memory programmer (output) is connected to a serial interface pin (input) that is connected to another device (output), a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the other device or set the other device to the output high-impedance status.

Figure 28-11. Conflict of Signals (Serial Interface Input Pin)

(b) Malfunction of other device

When the dedicated flash memory programmer (output or input) is connected to a serial interface pin (input or output) that is connected to another device (input), the signal is output to the other device, causing the device to malfunction. To avoid this, isolate the connection to the other device.

Figure 28-12. Malfunction of Other Device

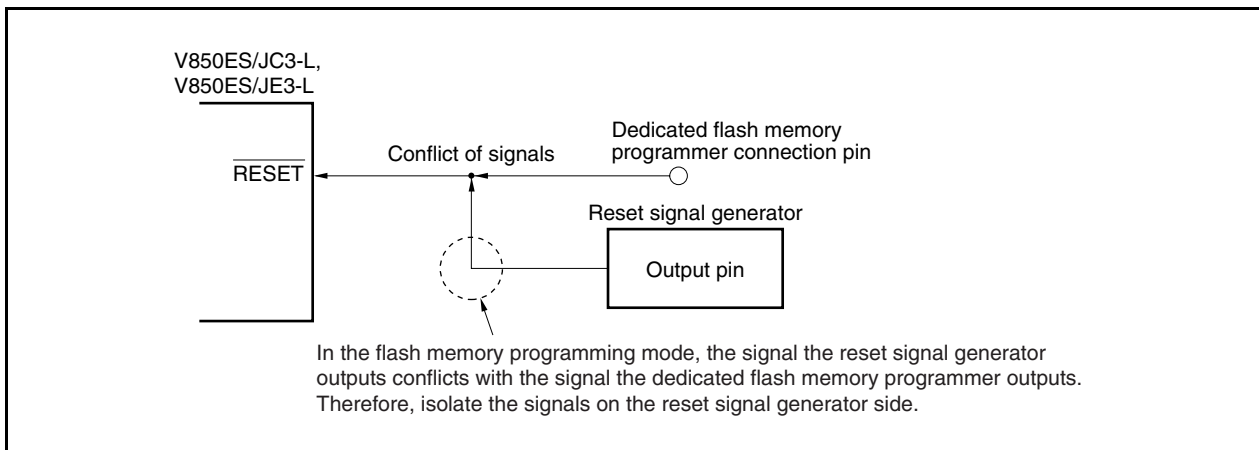


(4) RESET pin

When the reset signals of the dedicated flash memory programmer are connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on-board, a conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the reset signal generator.

When a reset signal is input from the user system in the flash memory programming mode, the programming operation will not be performed correctly. Therefore, do not input signals other than the reset signals from the dedicated flash memory programmer.

Figure 28-13. Conflict of Signals ($\overline{\text{RESET}}$ Pin)

**(5) Port pins (including NMI)**

When the system shifts to the flash memory programming mode, all the pins that are not used for flash memory programming are in the same status as that immediately after reset. If the external device connected to each port does not recognize the status of the port immediately after reset, pins require appropriate processing, such as connecting to V_{DD} via a resistor or connecting to V_{SS} via a resistor.

(6) Other signal pins

Connect X1, X2, XT1, XT2, and REGC in the same status as that in the normal operation mode.

During flash memory programming, input a low level to the $\overline{\text{DRST}}$ pin or leave it open. Do not input a high level.

(7) Power supply

Supply the same power (V_{DD} , V_{SS} , EV_{DD} , EV_{SS} , AV_{REF0} , AV_{REF1} , AV_{SS}) as in normal operation mode.

Remark For details about pin connection, see the **PG-FP5 User's Manual (U18865E)**.

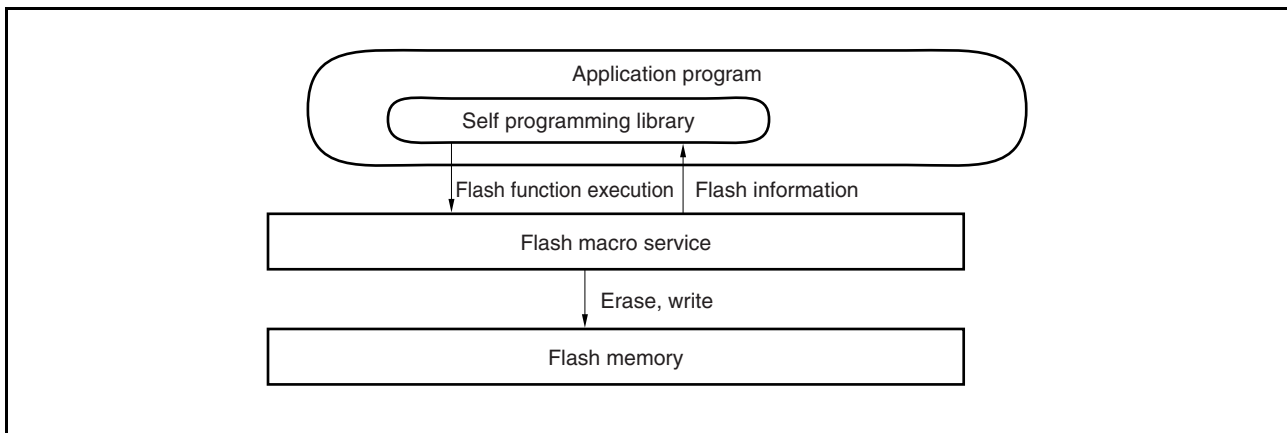
28.5 Rewriting by Self Programming

28.5.1 Overview

The V850ES/JC3-L and V850ES/JE3-L support a flash macro service that allows the user program to rewrite the internal flash memory by itself. By using this interface and a self programming library that is used to rewrite the flash memory with a user application program, the flash memory can be rewritten by a user application transferred in advance to the internal RAM. Consequently, the user program can be upgraded and constant data^{Note} can be rewritten in the field.

Note Be sure not to allocate the program code to the block where the constant data of the rewriting target is allocated. See **28.2 Memory Configuration** for the block configuration.

Figure 28-15. Concept of Self Programming



28.5.2 Features

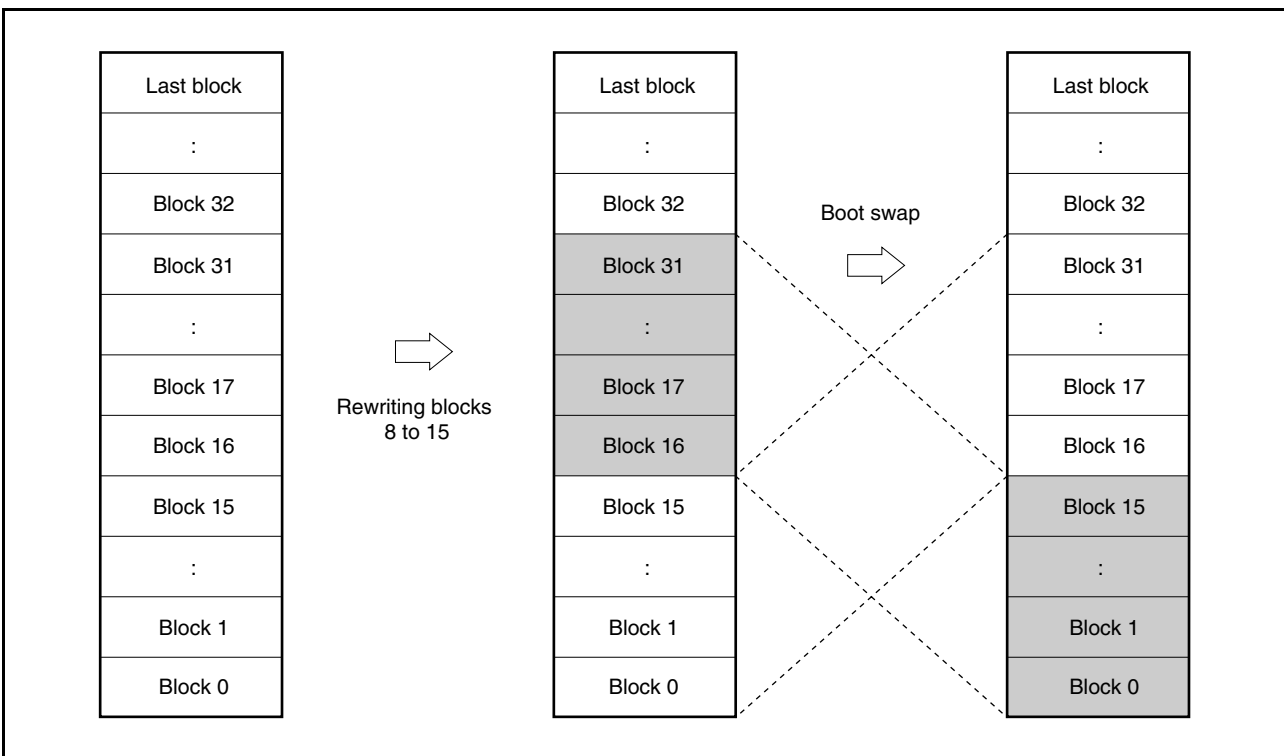
(1) Secure self programming (boot swap function)

The V850ES/JC3-L, V850ES/JE3-L supports a boot swap function that can be used to exchange the physical memory of blocks 0 to 15 with the physical memory of blocks 16 to 31. By writing the start program to be rewritten to blocks 16 to 31 in advance and then swapping the physical memory, the entire area can be safely rewritten even if a power failure occurs during rewriting because the correct user program always exists in blocks 0 to 15.

The boot swap function is not available for the following products:

- μ PD70F3797, 70F3801, 70F3805: 16 KB of flash memory
- μ PD70F3798, 70F3802, 70F3806: 32 KB of flash memory

Figure 28-15. Rewriting Entire Memory Area (Boot Swap)



(2) Interrupt support

Instructions cannot be fetched from the flash memory during self-programming. Consequently, a user handler written to the flash memory cannot be used even if an interrupt has occurred.

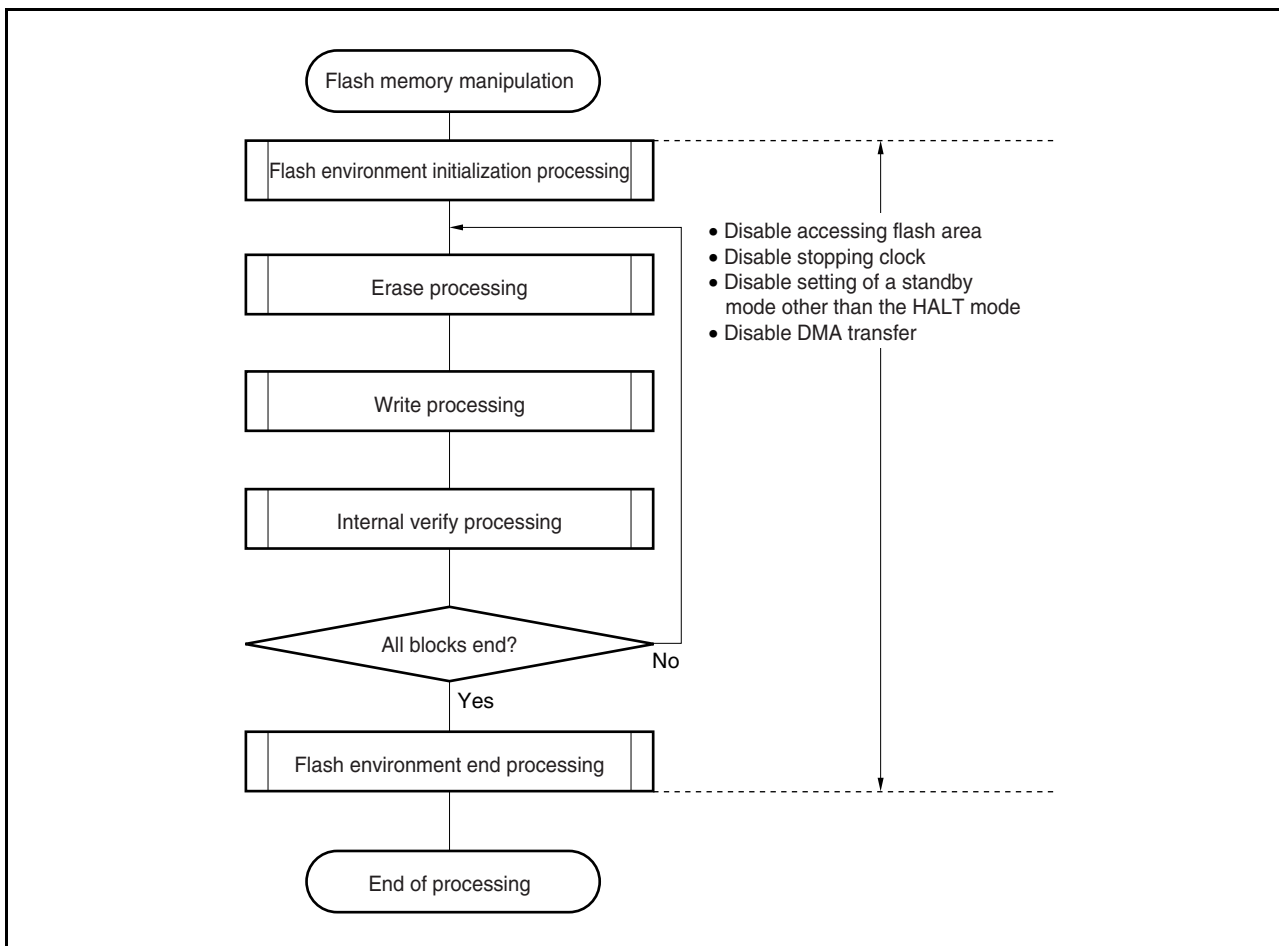
Therefore, to use an interrupt during self-programming in the V850ES/JC3-L, V850ES/JE3-L, execution must jump to a specific address^{Note} in the internal RAM. Write a branch instruction to execute the user interrupt servicing at the specific address^{Note} in the internal RAM.

- Note** NMI interrupt: Start address of internal RAM
 Maskable interrupt: Start address of internal RAM + 4 addresses

28.5.3 Standard self programming flow

The entire processing to rewrite the flash memory by flash self programming is illustrated below. For details, see the **Flash Memory Self-Programming Library User's Manual (U17819E)**.

Figure 28-16. Standard Self Programming Flowchart



28.5.4 Flash functions

Table 28-10. Flash Function List

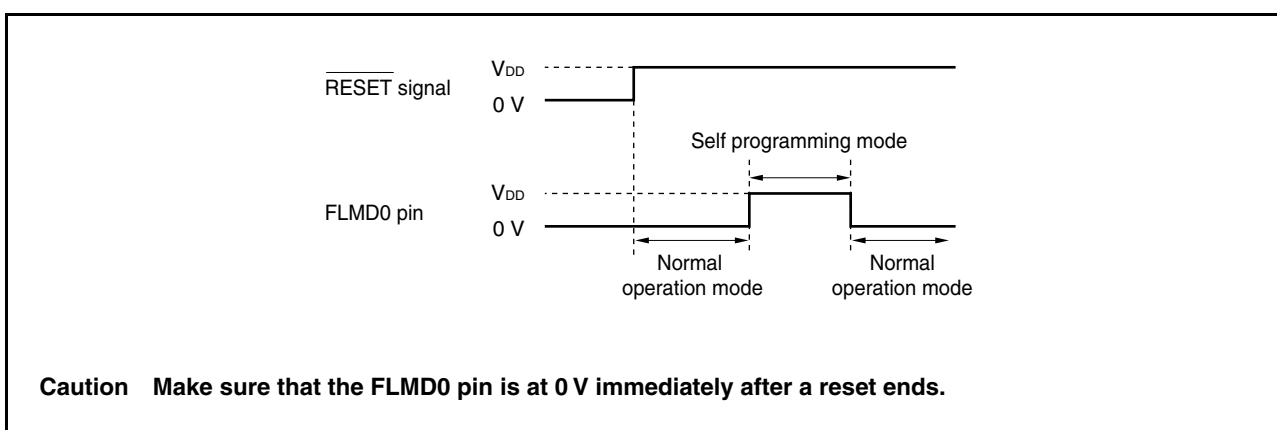
Function Name	Outline	Support
FlashInit	Self-programming library initialization	√
FlashEnv	Flash environment start/end	√
FlashFLMDCheck	FLMD pin check	√
FlashStatusCheck	Hardware processing execution status check	√
FlashBlockErase	Block erase	√
FlashWordWrite	Data write	√
FlashBlockVerify	Internal verification of block	√
FlashBlockBlankCheck	Blank check of block	√
FlashSetInfo	Flash information setting	√
FlashGetInfo	Flash information acquisition	√
FlashBootSwap	Boot swap execution	√

28.5.5 Pin processing

(1) FLMD0 pin

The FLMD0 pin is used to set the operation mode immediately after a reset ends and to protect the flash memory from being written during self rewriting. It is therefore necessary to keep the voltage applied to the FLMD0 pin at 0 V immediately after a reset ends and a normal operation is executed. It is also necessary to apply a voltage of V_{DD} level to the FLMD0 pin during the self programming mode period via port control before the memory is rewritten. When self programming has been completed, the voltage on the FLMD0 pin must be returned to 0 V.

Figure 28-17. Mode Change Timing



28.5.6 Internal resources used

The following table lists the internal resources used for self programming. These internal resources can also be used freely for purposes other than self programming.

Table 28-11. Internal Resources Used

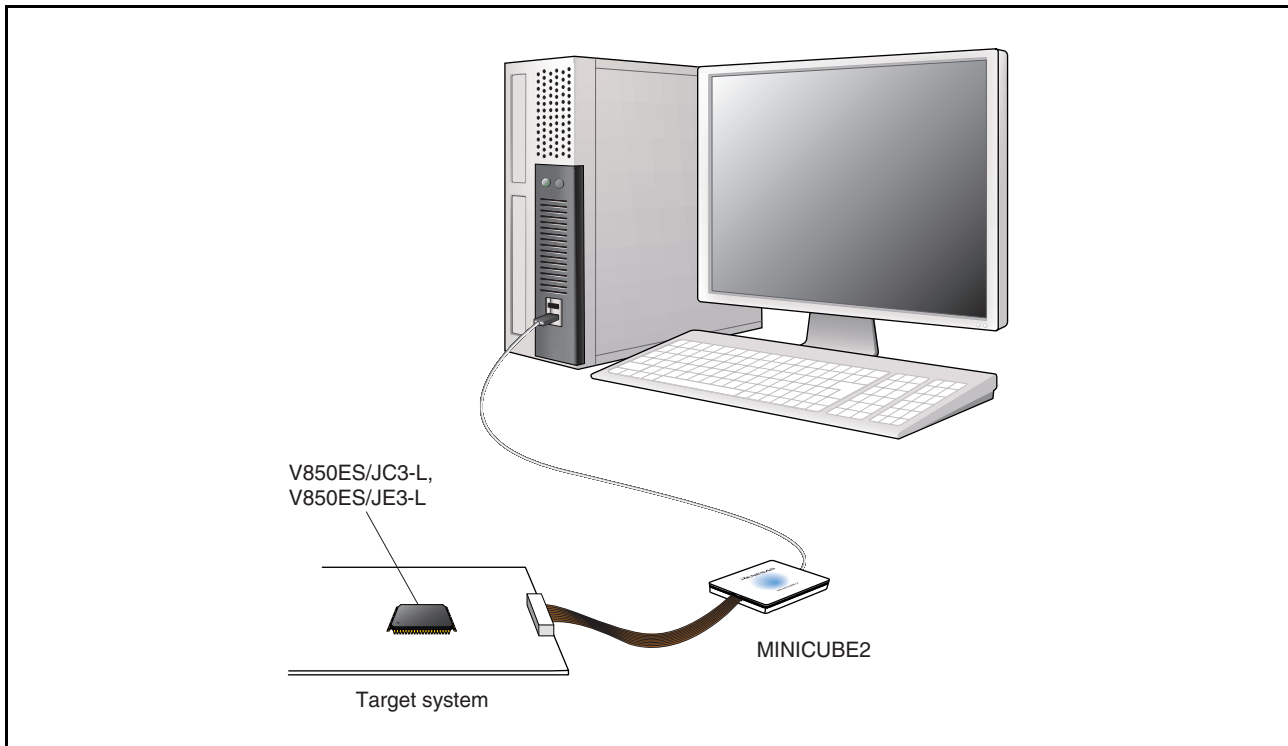
Resource Name	Description
Stack area	An extension of the stack used by the user is used by the library (can be used in both the internal RAM and external RAM).
Library code ^{Note}	Program entity of library (can be used anywhere other than the flash memory block to be manipulated).
Application program	Executed as user application. Calls flash functions.
Maskable interrupt	Can be used in the user application execution status or self-programming status. To use this interrupt in the self-programming status, since the processing transits to the address of the internal RAM start address + 4 addresses, allocate the branch instruction that shifts the processing to the user interrupt servicing at the address of the internal RAM start address + 4 addresses in advance.
NMI	Can be used in the user application execution status or self-programming status. To use this interrupt in the self-programming status, since the processing transits to the address of the internal RAM start address, allocate the branch instruction that shifts the processing to the user interrupt servicing at the internal RAM start address in advance.

Note For details about the resources used, see the **Flash Memory Self-Programming Library User's Manual**.

CHAPTER 29 ON-CHIP DEBUG FUNCTION

On-chip debugging is debugging executed with the V850ES/JC3-L, V850ES/JE3-L mounted on the target system. By using MINICUBE2, on-chip debugging can be performed with a simple interface.

Figure 29-1. On-Chip Debugging Using MINICUBE2



On-chip debugging can be performed for the V850ES/JC3-L and V850ES/JE3-L by using the following two methods.

- Using the DCU (debug control unit)
On-chip debugging is performed by the on-chip DCU in the V850ES/JC3-L and V850ES/JE3-L, with the $\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO pins used as the debug interface pins.
- Not using the DCU
On-chip debugging can be performed by MINICUBE2, using the user resources instead of the DCU.

The following table shows the features of the two on-chip debugging methods.

Table 29-1. Overview of On-Chip Debugging

		Debugging Using DCU	Debugging Without Using DCU
Debug interface pins		$\overline{\text{DRST}}$, DCK, DMS, DDI, DDO	<ul style="list-style-type: none"> When UARTA0 is used RXD0, TXD0 When CSIB0 is used SIB0, SOB0, SCKB0, HS (PCM0) When CSIB3^{Note3} is used SIB3, SOB3, SCKB3, HS (PCM0)
Allocating user resources		Not required	Required
Hardware break function		2 points	2 points
Software break function	Internal ROM area	4 points	4 points
	Internal RAM area	2000 points	2000 points
Real-time RAM monitor function ^{Note 1}		Available	Available
Dynamic memory modification (DMM) function ^{Note 2}		Available	Available
Mask function		$\overline{\text{Reset}}$, NMI, INTWDT2, $\overline{\text{HLDRQ}}$, $\overline{\text{WAIT}}$	$\overline{\text{RESET}}$ pin
ROM security function		10-byte ID code authentication	10-byte ID code authentication
Hardware used		MINICUBE, etc.	MINICUBE2, etc.
Trace function		Not supported.	Not supported.
Debug interrupt interface function (DBINT)		Not supported.	Not supported.

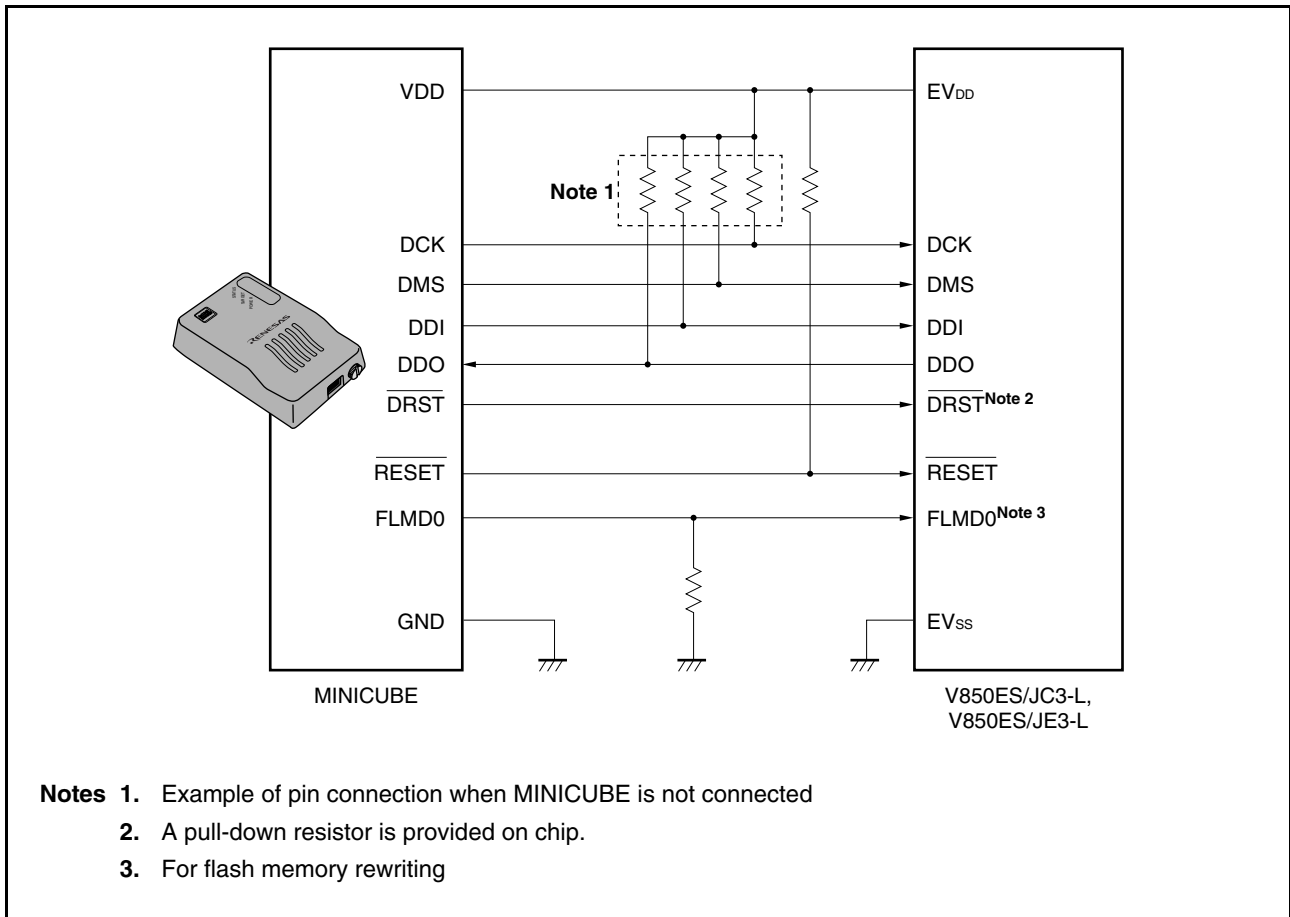
- Notes**
1. This is a function which reads out memory contents during program execution.
 2. This is a function which rewrites RAM contents during program execution.
 3. CSIB3 does not exist in the V850ES/JC3-L.

29.1 Debugging with DCU

By using the debug interface pins (\overline{DRST} , DCK, DMS, DDI, and DDO) to connect the on-chip emulator (MINICUBE), programs can be debugged without using user resources other than these pins.

29.1.1 Connection circuit example

Figure 29-2. Circuit Connection Example When Debug Interface Pins Are Used for Communication Interface



29.1.2 Interface signals

The interface signals are described below.

(1) $\overline{\text{DRST}}$

This is a reset input signal for the on-chip debug unit. It is a negative-logic signal that asynchronously initializes the debug control unit.

MINICUBE raises the $\overline{\text{DRST}}$ signal when it detects V_{DD} of the target system after the integrated debugger is started, and starts the on-chip debug unit of the device.

When the $\overline{\text{DRST}}$ signal goes high, a reset signal is also generated in the CPU.

When starting debugging by starting the integrated debugger, a CPU reset is always generated.

(2) DCK

This is a clock input signal. It supplies a 20 MHz or 10 MHz clock from MINICUBE. In the on-chip debug unit, the DMS and DDI signals are sampled at the rising edge of the DCK signal, and the data DDO is output at its falling edge.

(3) DMS

This is a transfer mode select signal. The transfer status in the debug unit changes depending on the level of the DMS signal.

(4) DDI

This is a data input signal. It is sampled in the on-chip debug unit at the rising edge of DCK.

(5) DDO

This is a data output signal. It is output from the on-chip debug unit at the falling edge of the DCK signal.

(6) EV_{DD}

This signal is used to detect V_{DD} of the target system.

If V_{DD} of the target system cannot be detected, MINICUBE makes its output signals ($\overline{\text{DRST}}$, DCK, DMS, DDI, FLMD0, and $\overline{\text{RESET}}$) high-impedance.

(7) FLMD0

The flash self programming function is used to download data to the flash memory via the integrated debugger (ID850QB). During flash self programming, the FLMD0 pin must be kept high. In addition, connect a pull-down resistor to the FLMD0 pin.

The FLMD0 pin can be controlled in either of the following two ways.

<1> To control from MINICUBE

Connect the FLMD0 signal of MINICUBE to the FLMD0 pin.

In the normal mode, nothing is driven by MINICUBE (high impedance).

During a break, MINICUBE raises the FLMD0 pin to the high level when the download function of the integrated debugger is executed. In other cases, the FLMD0 pin is in a high-impedance state.

<2> To control from port

Use this method when executing self-programming.

Connect any port of the device to the FLMD0 pin.

The same port as the one used by the user program to realize the flash self programming function may be used.

Before executing a download, set the port pin connected to the FLMD0 pin to high level on the console of the integrated debugger. Upon completion of the download, reset the port pin to low level.

For details, refer to the **ID850QB Ver. 3.40 Integrated Debugger Operation User's Manual (U18604E)**.

(8) RESET

This is a system reset input pin. If the \overline{DRST} pin is made invalid by the value of the OCDM.OCDM0 bit set by the user program, on-chip debugging cannot be executed. Therefore, a reset is executed by MINICUBE, using the \overline{RESET} pin, to make the \overline{DRST} pin valid (initialization).

29.1.3 Mask function

The reset, NMI, INTWDT2, \overline{WAIT} , and \overline{HLDRQ} signals can be masked.

The maskable signals in the integrated debugger (ID850QB) and the corresponding V850ES/JC3-L and V850ES/JE3-L functions are listed below.

Table 29-2. Mask Functions

Maskable Signals in Debugger (ID850QB)	Corresponding V850ES/JC3-L, V850ES/JE3-L Functions
NMI0	NMI pin input
NMI2	Non-maskable interrupt request signal (INTWDT2) generation
STOP	Non-maskable
HOLD	\overline{HLDRQ} pin input
RESET	Reset signal generation by \overline{RESET} pin input, low-voltage detector, clock monitor, or watchdog timer (WDT2) overflow
WAIT	\overline{WAIT} pin input

29.1.4 Registers

(1) On-chip debug mode register (OCDM)

This register is used to specify whether a pin provided with an on-chip debug function is used as an on-chip debug pin or as an ordinary port/peripheral function pin. It also is used to disconnect the internal pull-down resistor of the P05/INTP2/ $\overline{\text{DRST}}$ pin.

This register is a special register and can be written only in a combination of specific sequences (see 3.4.7 Special registers).

The OCDM register can be written only while a low level is input to the P05/INTP2/ $\overline{\text{DRST}}$ pin.

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

After reset: 01H^{Note} R/W Address: FFFF9FCH

	7	6	5	4	3	2	1	<0>
OCDM	0	0	0	0	0	0	0	OCDM0

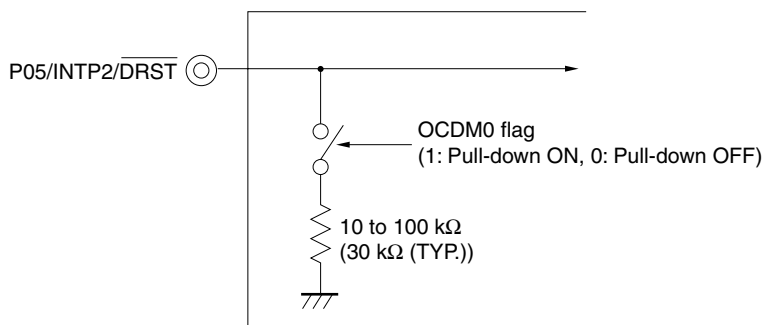
OCDM0	Operation mode
0	Selects normal operation mode (in which a pin that functions alternately as on-chip debug function pin is used as a port/peripheral function pin) and disconnects the on-chip pull-down resistor of the P05/INTP2/ $\overline{\text{DRST}}$ pin.
1	When P05/INTP2/ $\overline{\text{DRST}}$ pin is low: Normal operation mode (in which a pin that functions alternately as an on-chip debug function pin is used as a port/peripheral function pin) When P05/INTP2/ $\overline{\text{DRST}}$ pin is high: On-chip debug mode (in which a pin that functions alternately as an on-chip debug function pin is used as an on-chip debug mode pin)

Note $\overline{\text{RESET}}$ input sets this register to 01H. After reset by the WDT2RES signal, clock monitor (CLM), or low-voltage detector (LVI), however, the value of the OCDM register is retained.

Cautions 1. When using the DDI, DDO, DCK, and DMS pins not as on-chip debug pins but as port pins after external reset, one of the following actions must be taken.

- Input a low level to the P05/INTP2/ $\overline{\text{DRST}}$ pin.
- Set the OCDM0 bit. In this case, take the following actions.
 - <1> Clear the OCDM0 bit to 0.
 - <2> Fix the P05/INTP2/ $\overline{\text{DRST}}$ pin to low level until <1> is completed.

2. The P05/INTP2/ $\overline{\text{DRST}}$ pin has an on-chip pull-down resistor. This resistor is disconnected when the OCDM0 flag is cleared to 0.



29.1.5 Operation

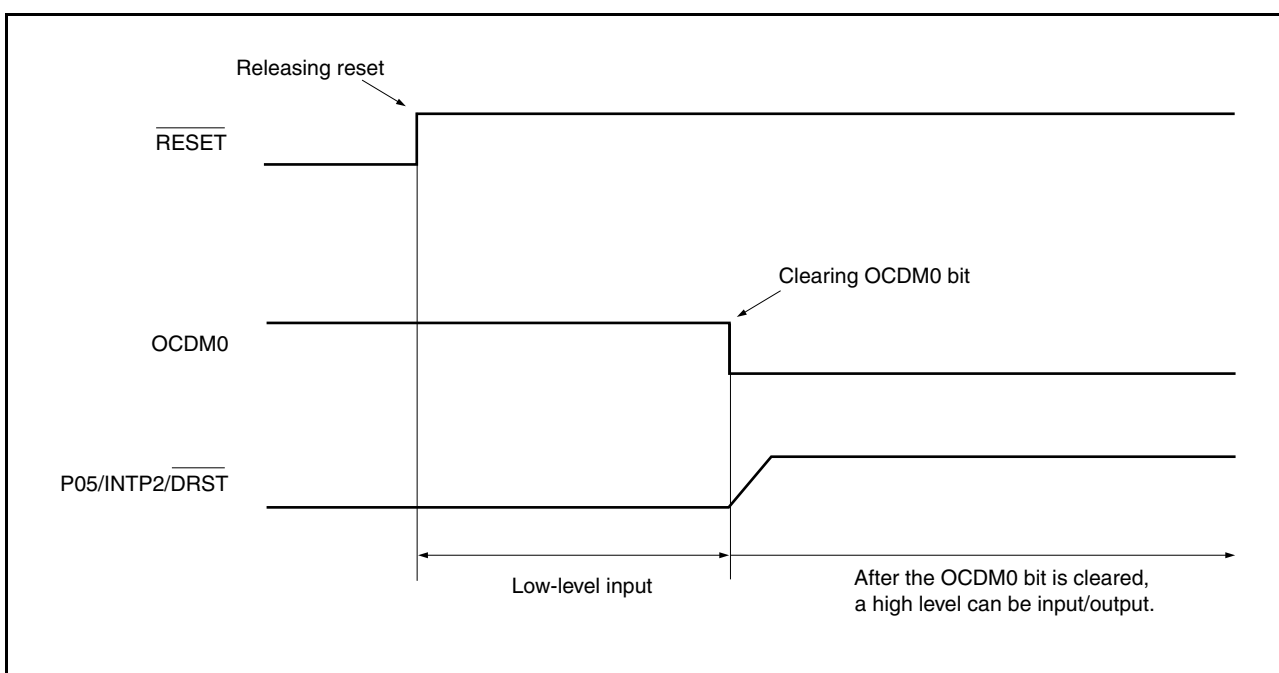
The on-chip debug function is made invalid under the conditions shown in the table below.

When this function is not used, keep the $\overline{\text{DRST}}$ pin low until the OCDM.OCDM0 flag is cleared to 0.

OCDM0 Flag $\overline{\text{DRST}}$ Pin	0	1
L	Invalid	Invalid
H	Invalid	Valid

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

Figure 29-3. Timing When On-Chip Debug Function Is Not Used



29.1.6 Cautions

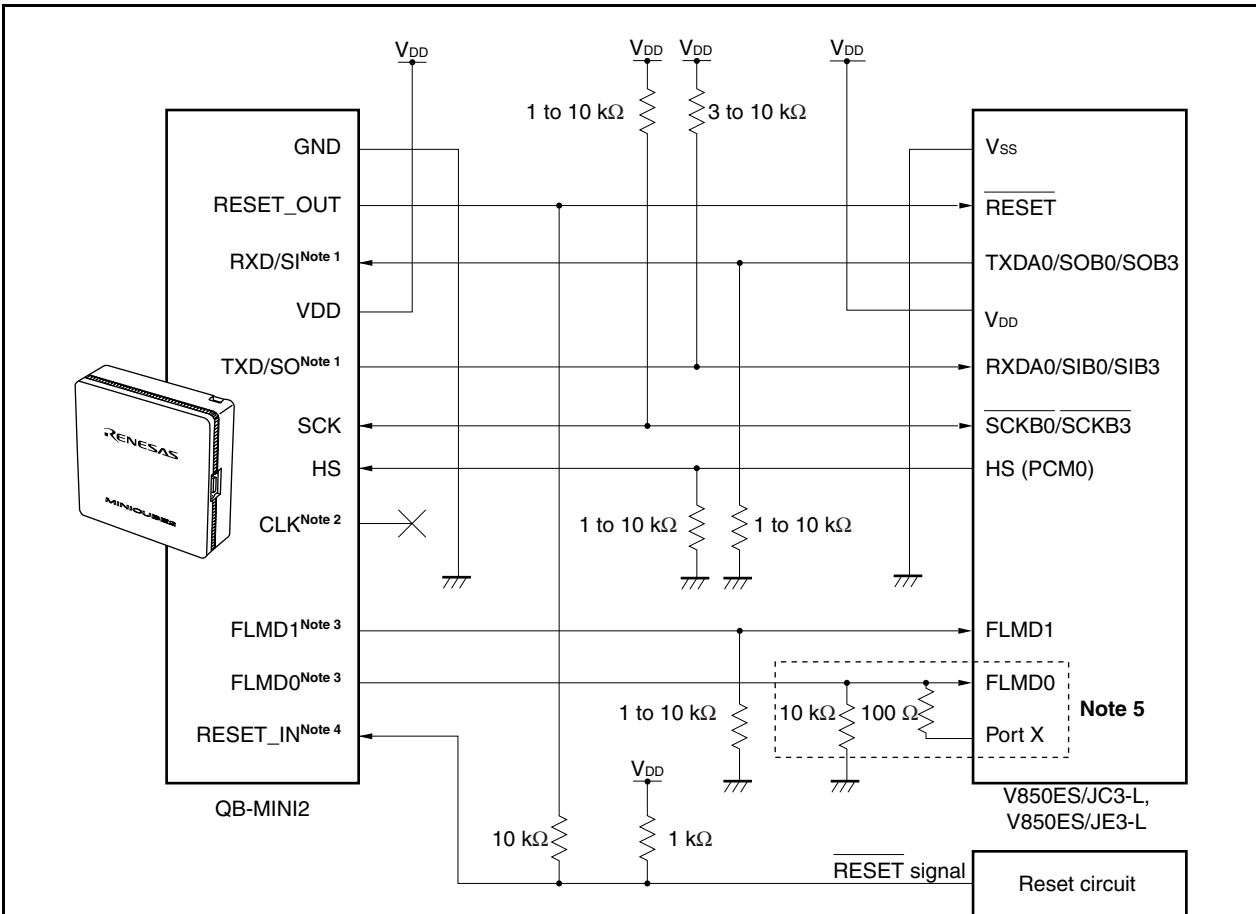
- (1) If a reset signal is input (from the target system or from an internal reset source) during program execution, the break function may malfunction.
- (2) Even if the reset signal is masked by the mask function, the I/O buffer (port pin) may be reset if a reset signal is input from a pin.
- (3) The reset signal from a pin for which a break is specified is masked and the CPU and peripheral I/O are not reset. If a pin-based reset or internal reset is generated as soon as the flash memory is rewritten by DMM or read by the RAM monitor function while the user program is being executed, the CPU and peripheral I/O may not be correctly reset.
- (4) In the on-chip debug mode, the DDO pin is forcibly set to high-level output.
- (5) On-chip debugging can be used when the supply voltage (V_{DD}) is in a range of 2.7 to 3.6 V. It cannot be used at less than 2.7 V.
- (6) In the on-chip debug mode, the output voltage of the regulator does not decrease even in the low-voltage STOP mode, low-voltage subclock mode, or low-voltage sub-IDLE mode

29.2 Debugging Without Using DCU

The following describes how to implement an on-chip debug function using MINICUBE2 with pins for UARTA0 (RXDA0 and TXDA0), pins for CSIB0 (SIB0, SOB0, SCKB0, and HS (PCM0)), or pins for CSIB3 (SIB3, SOB3, SCKB3, and HS (PCM0)) as the debug interface, without using the DCU.

29.2.1 Circuit connection examples

Figure 29-4. Circuit Connection Example When UARTA0/CSIB0/CSIB3 Is Used for Communication Interface



- Notes**
1. Connect TXDA0/SOB0/SOB3 (transmit side) of the V850ES/JC3-L and V850ES/JE3-L to RXD/SI (receive side) of the target connector, and TXD/SO (transmit side) of the target connector to RXDA0/SIB0/SIB3 (receive side) of the V850ES/JC3-L and V850ES/JE3-L.
 2. This pin is not used during on-chip debugging.
 3. During debugging, this pin is used as an input (unused) pin and can be used for its alternate functions. A pull-down resistor of 100kΩ is connected to this pin in MINICUBE2.
 4. This connection is designed assuming that the $\overline{\text{RESET}}$ signal is output from an N-ch open-drain buffer (output resistance: 100 Ω or less).
 5. The circuit enclosed by a dashed line is designed for flash self programming and controls the FLMD0 pin via a port. Use a port for inputting or outputting the high level. When flash self programming is not performed, the pull-down resistance for the FLMD0 pin can be 1 to 10 kΩ.

Remark See Table 29-3 for the pins used when UARTA0, CSIB0, or CSIB3 is used for communication interface.

Table 29-3. Wiring Between V850ES/JC3-L and MINICUBE2

Pin Configuration of MINICUBE2 (QB-MINI2)			With CSIB0-HS			With UARTA0		
Signal Name	I/O	Pin Function	Pin Name	Pin No.		Pin Name	Pin No.	
				(40)	(48)		(40)	(48)
SI/RxD	Input	Pin to receive commands and data from V850ES/JC3-L	P41/SOB0	17	21	P30/TXD A0	29	33
SO/TxD	Output	Pin to transmit commands and data to V850ES/JC3-L	P40/SIB0	16	20	P31/RXD A0	30	34
SCK	Output	Clock output pin for 3-wire serial communication	P42/ $\overline{\text{SCKB0}}$	18	22	Not needed	–	–
CLK	Output	Pin outputting clock signal to V850ES/JC3-L	Not needed	–	–	Not needed	–	–
RESET_OUT	Output	Pin outputting reset signal to V850ES/JC3-L	$\overline{\text{RESET}}$	8	10	$\overline{\text{RESET}}$	8	10
FLMD0	Output	Output pin to set V850ES/JC3-L to debug mode or programming mode	FLMD0	28	32	FLMD0	28	32
FLMD1	Output	Output pin to set programming mode	PDL5/FLMD1	31	37	PDL5/FLMD1	31	37
HS	Input	Handshake signal for CSIO + HS communication	PCM0	32	35	Not needed	–	–
GND	–	Ground	V _{SS}	5	7	V _{SS}	5	7
			AV _{SS}	2	2	AV _{SS}	2	2
			EV _{SS}	15	17	EV _{SS}	15	17
RESET_IN	Input	Reset input pin on the target system						

Remark (40): 40-pin plastic WQFN (6 × 6)

(48): 48-pin plastic WQFN (7 × 7), 48-pin plastic LQFP (7 × 7)

Table 29-4. Wiring Between V850ES/JE3-L and MINICUBE2

Pin Configuration of MINICUBE2 (QB-MINI2)			With CSIB0-HS		With CSIB3-HS		With UARTA0	
Signal Name	I/O	Pin Function	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Pin to receive commands and data from V850ES/JE3-L	P41/SOB0	29	P911/SOB3	15	P30/TXDA0	45
SO/TxD	Output	Pin to transmit commands and data to V850ES/JE3-L	P40/SIB0	28	P910/SIB3	16	P31/RXDA0	32
SCK	Output	Clock output pin for 3-wire serial communication	P42/SCKB0	30	P912/SCKB3	14	Not needed	–
CLK	Output	Pin outputting clock signal to V850ES/JE3-L	Not needed	–	Not needed	–	Not needed	–
RESET_OUT	Output	Pin outputting reset signal to V850ES/JE3-L	RESET	10	RESET	10	RESET	10
FLMD0	Output	Output pin to set V850ES/JE3-L to debug mode or programming mode	FLMD0	42	FLMD0	42	FLMD0	42
FLMD1	Output	Output pin to set programming mode	PDL5/FLMD1	49	PDL5/FLMD1	49	PDL5/FLMD1	49
HS	Input	Handshake signal for CSIO + HS communication	PCM0	54	PCM0	54	Not needed	–
GND	–	Ground	V _{SS}	7	V _{SS}	7	V _{SS}	7
			AV _{SS}	2	AV _{SS}	2	AV _{SS}	2
			EV _{SS}	25	EV _{SS}	25	EV _{SS}	25
RESET_IN	Input	Reset input pin on the target system						

29.2.2 Mask function

Only reset signals can be masked.

The maskable signals in the debugger (ID850QB) and the corresponding V850ES/JC3-L and V850ES/JE3-L functions are listed below.

Table 29-5. Mask Functions

Maskable Signals in ID850QB	Corresponding V850ES/JC3-L and V850ES/JE3-L Functions
NMI0	–
NMI1	–
NMI2	–
STOP	–
HOLD	–
RESET	Reset signal generation by $\overline{\text{RESET}}$ pin input
WAIT	–

29.2.3 Allocation of user resources

The user must prepare the following resources to perform communication between MINICUBE2 and the target device and implement each debug function. These items need to be set in the user program or using the compiler options.

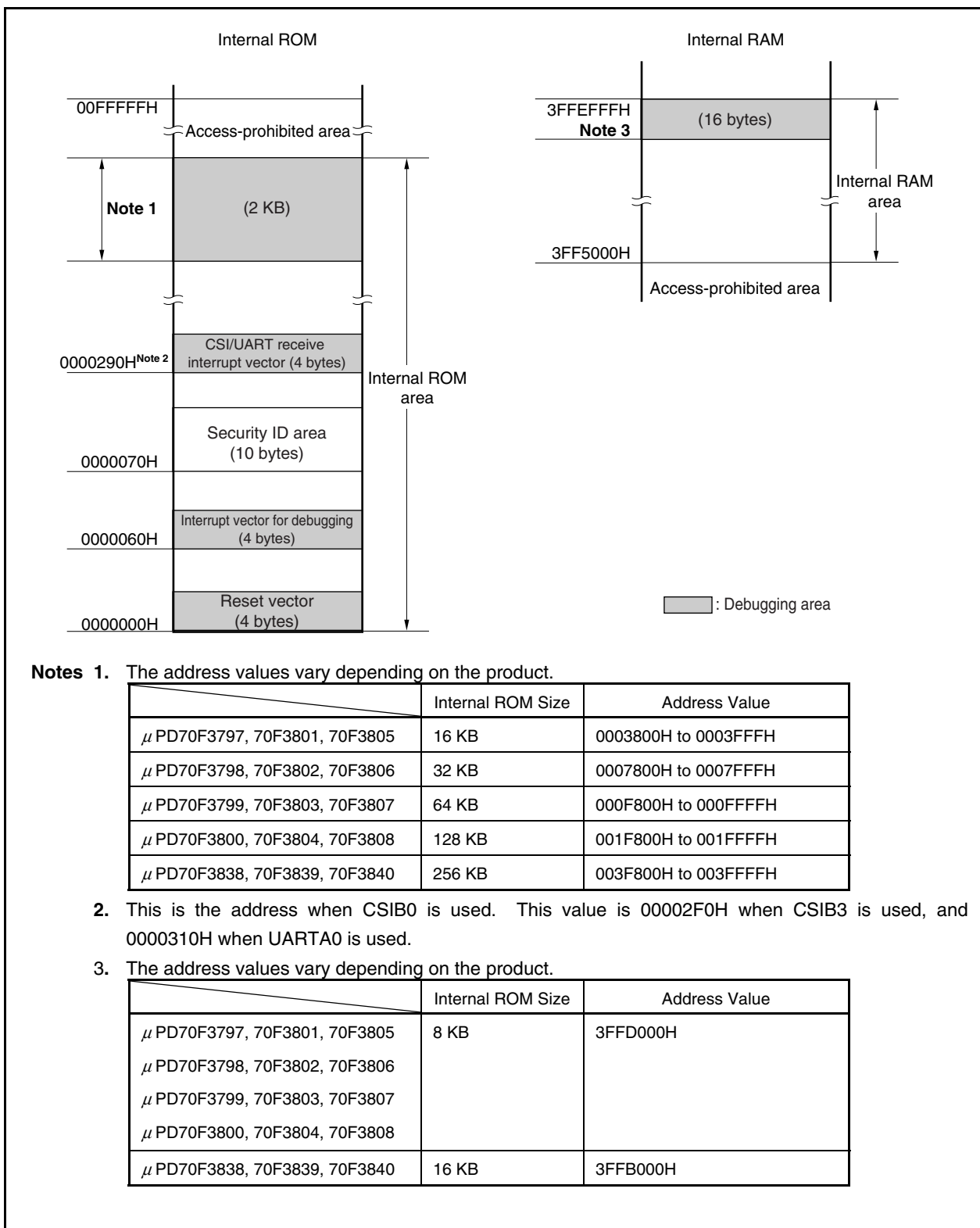
(1) Allocation of memory space

The shaded portions in Figure 29-5 are the areas reserved for placing the debug monitor program, so user programs and data cannot be allocated to these spaces. These spaces must be secured so as not to be used by the user program.

(2) Security ID setting

The ID code must be embedded in the area between 0000070H and 0000079H in Figure 29-5, to prevent the memory from being read by an unauthorized person. For details, see **29.3 ROM Security Function**.

Figure 29-5. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. The address values vary depending on the product.

	Internal ROM Size	Address Value
μ PD70F3797, 70F3801, 70F3805	16 KB	0003800H to 0003FFFH
μ PD70F3798, 70F3802, 70F3806	32 KB	0007800H to 0007FFFH
μ PD70F3799, 70F3803, 70F3807	64 KB	000F800H to 000FFFFH
μ PD70F3800, 70F3804, 70F3808	128 KB	001F800H to 001FFFFH
μ PD70F3838, 70F3839, 70F3840	256 KB	003F800H to 003FFFFH

2. This is the address when CSIB0 is used. This value is 00002F0H when CSIB3 is used, and 0000310H when UARTA0 is used.

3. The address values vary depending on the product.

	Internal ROM Size	Address Value
μ PD70F3797, 70F3801, 70F3805	8 KB	3FFD000H
μ PD70F3798, 70F3802, 70F3806		
μ PD70F3799, 70F3803, 70F3807		
μ PD70F3800, 70F3804, 70F3808		
μ PD70F3838, 70F3839, 70F3840	16 KB	3FFB000H

(3) Reset vector

The reset vector includes the jump instruction for the debug monitor program.

[How to secure the reset vector]

It is not necessary to secure this area intentionally. When downloading a program, however, the debugger rewrites the reset vector in accordance with the following cases. If the rewritten pattern does not match the following cases, the debugger generates an error (FOC34 when using the ID850QB).

(a) When two nop instructions are placed in succession from address 0

Before rewriting		After rewriting
0x0 nop	→	Jumps to debug monitor program at 0x0
0x2 nop		0x4 xxxx
0x4 xxxx		

(b) When two 0xFFFF values are successively placed from address 0 (already erased device)

Before rewriting		After rewriting
0x0 0xFFFF	→	Jumps to debug monitor program at 0x0
0x2 0xFFFF		0x4 xxxx
0x4 xxxx		

(c) The jr instruction is placed at address 0 (when using CA850)

Before rewriting		After rewriting
0x0 jr disp22	→	Jumps to debug monitor program at 0x0
		0x4 jr disp22 - 4

(d) mov32 and jmp are placed in succession from address 0 (when using IAR compiler ICCV850)

Before rewriting		After rewriting
0x0 mov imm32,reg1	→	Jumps to debug monitor program at 0x0
0x6 jmp [reg1]		0x4 mov imm32,reg1
		0xa jmp [reg1]

(e) The jump instruction for the debug monitor program is placed at address 0

Before rewriting		After rewriting
Jumps to debug monitor program at 0x0	→	No change

(4) Allocation of area for debug monitor program

The shaded portions in Figure 29-5 are the areas where the debug monitor program is allocated. The monitor program performs initialization processing for the debug communication interface and run or break processing for the CPU. The internal ROM area must be filled with 0xFF. This area must not be rewritten by the user program.

[How to secure area for the debug monitor program]

It is not necessarily required to secure this area if the user program does not use this area.

To avoid problems that may occur during debugger startup, however, it is recommended to secure this area in advance, using the compiler.

The following shows examples of securing the area, using the Renesas Electronics compiler CA850. Add the assemble source file and link directive code, as shown below.

- Assemble source (Add the following code as an assemble source file.)

```
-- Secures 2 KB space for monitor ROM section
.section "MonitorROM", const
.space 0x800, 0xff

-- Secures interrupt vector for debugging
.section "DBG0"
.space 4, 0xff

-- Secures interrupt vector for serial communication
-- Change the section name according to the serial communication mode used
.section "INTCB0R"
.space 4, 0xff

-- Secures 16-byte space for monitor RAM section
.section "MonitorRAM", bss
.lcomm monitorramsym, 16, 4 -- defines symbol monitorramsym
```

- Link directive (Add the following code to the link directive file.)

The following shows an example when using the μ PD70F3840 (the internal ROM is 256 KB (end address is 003FFFFH) and internal RAM is 16 KB (end address is 3FFEFFH)).

```
MROMSEG      : !LOAD ?R V0x03f800{
              MonitorROM = $PROGBITS ?A MonitorROM;
};

MRAMSEG      : !LOAD ?RW V0x03ffeff0{
              MonitorRAM = $NOBITS ?AW MonitorRAM;
};
```

(5) Allocation of communication serial interface

UARTA0, CSIB0, or CSIB3 is used for communication between MINICUBE2 and the target system. The settings related to the serial interface modes are performed by the debug monitor program, but if the setting is changed by the user program, a communication error may occur.

To prevent such a problem from occurring, the communication serial interface must be secured in the user program.

[How to secure the communication serial interface]

- On-chip debug mode register (OCDM)

For the on-chip debug function using the UARTA0, CSIB0, or CSIB3, set the OCDM register to normal mode. Be sure to set as follows.

- Input low level to the P05/INTP2/ $\overline{\text{DRST}}$ pin.
- Set the OCDM0 bit as shown below.
 - <1> Clear the OCDM0 bit to 0.
 - <2> Fix the P05/INTP2/ $\overline{\text{DRST}}$ pin input to low level until the processing of <1> is complete.

- Serial interface registers

Do not set the registers related to CSIB0, CSIB3, or UARTA0 in the user program.

- Interrupt mask register

When CSIB0 is used, do not mask the transmit end interrupt (INTCB0R). When CSIB3 is used, do not mask the transmit end interrupt (INTCB3R). When UARTA0 is used, do not mask the reception complete interrupt (INTUA0R).

(a) When CSIB0 is used

	7	6	5	4	3	2	1	0
CB0RIC	×	0	×	×	×	×	×	×

(b) When CSIB3 is used

	7	6	5	4	3	2	1	0
CB3RIC	×	0	×	×	×	×	×	×

(c) When UARTA0 is used

	7	6	5	4	3	2	1	0
UA0RIC	×	0	×	×	×	×	×	×

Remark ×: don't care

- Port registers when UARTA0 is used

When UARTA0 is used, port registers are set by the debug monitor program to make the TXDA0 and RXDA0 pins valid. Do not change the following register settings in the user program during debugging. (The same value can be written again.)

	7	6	5	4	3	2	1	0
PFC3	×	×	×	×	×	×	0	0
PMC3L	×	×	×	×	×	×	1	1

Remark ×: don't care

- Port registers when CSIB0 is used

When CSIB0 is used, port registers are set by the debug monitor program to make the SIB0, SOB0, $\overline{\text{SCKB0}}$, and HS (PCM0) pins valid. Do not change the following register settings in the user program during debugging. (The same value can be written again.)

(a) SIB0, SOB0, and $\overline{\text{SCKB0}}$ settings

	7	6	5	4	3	2	1	0
PFC4	×	×	×	×	×	×	0	0
PMC4	×	×	×	×	×	1	1	1

(b) HS (PCM0 pin) settings

	7	6	5	4	3	2	1	0
PMCM	×	×	×	×	×	×	×	0
PCM	×	×	×	×	×	×	×	Note

Note Writing to this bit is prohibited.
The port values corresponding to the HS pin are changed by the monitor program according to the debugger status. To specify port register settings in 8-bit units, read-modify-write can usually be used in the user program. If an interrupt for debugging occurs before writing, however, an unexpected operation may be performed.

Remark ×: don't care

- Port registers when CSIB3 is used

When CSIB3 is used, port registers are set by the debug monitor program to make the SIB3, SOB3, $\overline{\text{SCKB3}}$, and HS (PCM0) pins valid. Do not change the following register settings in the user program during debugging. (The same value can be written again.)

(a) SIB3, SOB3, and $\overline{\text{SCKB3}}$ settings

	7	6	5	4	3	2	1	0
PFC9H	×	×	×	1	1	1	×	×

	7	6	5	4	3	2	1	0
PMC9H	×	×	×	1	1	1	×	×

(b) HS (PCM0 pin) settings

	7	6	5	4	3	2	1	0
PMCM	×	×	×	×	×	×	×	0

	7	6	5	4	3	2	1	0
PCM	×	×	×	×	×	×	×	Note

Note Writing to this bit is prohibited.
 The port values corresponding to the HS pin are changed by the monitor program according to the debugger status. To specify port register settings in 8-bit units, read-modify-write can usually be used in the user program. If an interrupt for debugging occurs before writing, however, an unexpected operation may be performed.

Remark ×: don't care

29.2.4 Cautions

(1) Handling of device that was used for debugging

Do not mount a device that was used for debugging on a mass-produced product, because the flash memory was rewritten during debugging and the number of rewrites of the flash memory cannot be guaranteed. Moreover, do not embed the debug monitor program into mass-produced products.

(2) When breaks cannot be executed

Forced breaks cannot be executed if one of the following conditions is satisfied.

- Interrupts are disabled (DI)
- Interrupts issued for the serial interface, which is used for communication between MINICUBE2 and the target device, are masked
- Standby mode is entered while standby release by a maskable interrupt is prohibited
- Mode for communication between MINICUBE2 and the target device is UARTA0, and the main clock has been stopped

(3) When pseudo real-time RAM monitor (RRM) function and DMM function do not operate

The pseudo RRM function and DMM function do not operate if one of the following conditions is satisfied.

- Interrupts are disabled (DI)
- Interrupts issued for the serial interface, which is used for communication between MINICUBE2 and the target device, are masked
- Standby mode is entered while standby release by a maskable interrupt is prohibited
- Mode for communication between MINICUBE2 and the target device is UARTA0, and the main clock has been stopped
- Mode for communication between MINICUBE2 and the target device is UARTA0, and a clock different from the one specified in the debugger is used for communication

(4) Standby release with pseudo RRM and DMM functions enabled

The standby mode is released by the pseudo RRM function and DMM function if one of the following conditions is satisfied.

- Mode for communication between MINICUBE2 and the target device is CSIB0 or CSIB3
- Mode for communication between MINICUBE2 and the target device is UARTA0, and the main clock has been supplied.

(5) Writing to peripheral I/O registers that require a specific sequence, using DMM function

Peripheral I/O registers that require a specific sequence cannot be written by using the DMM function.

(6) Flash self programming

If the space where the debug monitor program is allocated is rewritten by flash self programming, the debugger can no longer operate normally.

(7) On-chip debugging can be used when the supply voltage (V_{DD}) is in a range of 2.7 to 3.6 V. It cannot be used at less than 2.7 V.

29.3 ROM Security Function

29.3.1 Security ID

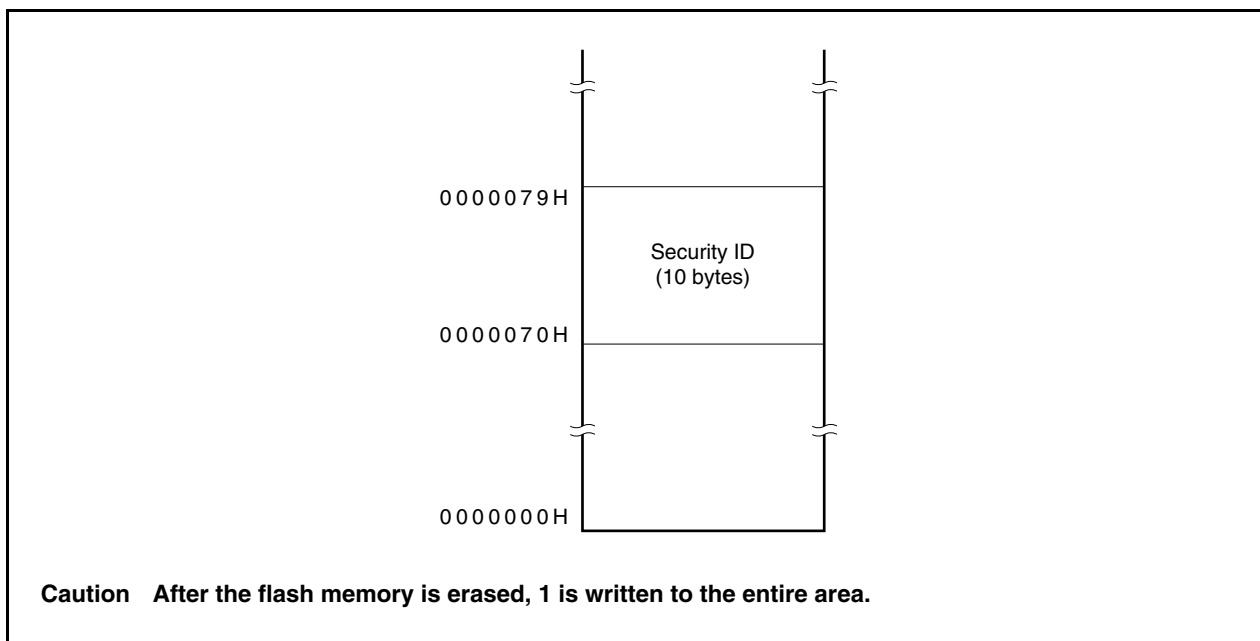
The flash memory versions of the V850ES/JC3-L and V850ES/JE3-L perform authentication using a 10-byte ID code to prevent the contents of the flash memory from being read by an unauthorized person during on-chip debugging by the on-chip debug emulator.

Set the ID code in the 10-byte internal flash memory area from 0000070H to 0000079H to allow the debugger perform ID authentication.

If the IDs match, the security is released and reading the flash memory and using the on-chip debug emulator are enabled.

- Set the 10-byte ID code to 0000070H to 0000079H.
- Bit 7 of 0000079H is the on-chip debug emulator enable flag.
(0: Disable, 1: Enable)
- When the on-chip debug emulator is started, the debugger requests ID input. When the ID code input to the debugger and the ID code set in 0000070H to 0000079H match, the debugger starts.
- Debugging cannot be performed if the on-chip debug emulator enable flag is 0, even if the ID codes match.

Figure 29-6. Security ID Area



29.3.2 Setting

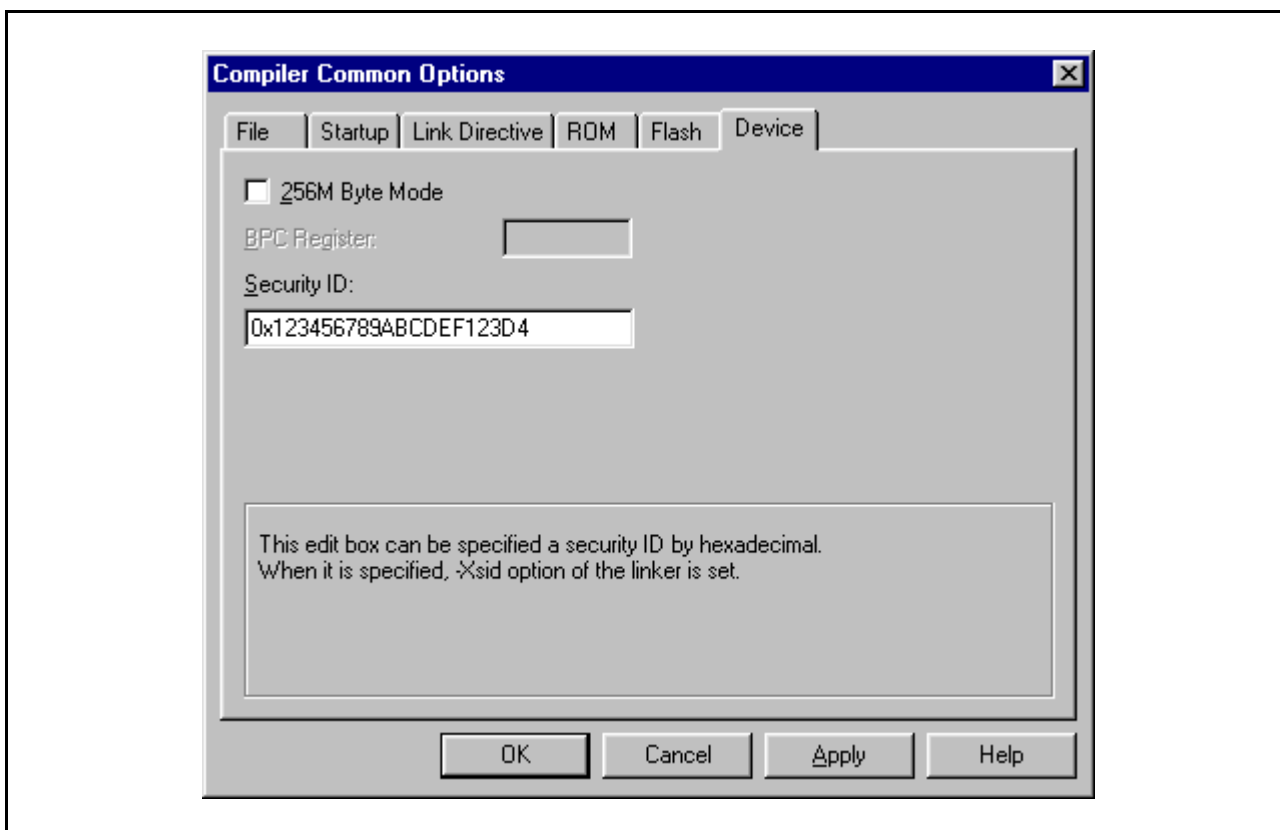
The following shows how to set the ID code as shown in Table 29-6.

When the ID code is set as shown in Table 29-6, the ID code input in the configuration dialog box of the ID850QB is "123456789ABCDEF123D4" (the ID code is not case-sensitive).

Table 29-6. ID Code

Address	Value
0x70	0x12
0x71	0x34
0x72	0x56
0x73	0x78
0x74	0x9A
0x75	0xBC
0x76	0xDE
0x77	0XF1
0x78	0x23
0x79	0xD4

The ID code can be specified in the Compiler Common Options dialog box in PM+.



CHAPTER 30 ELECTRICAL SPECIFICATIONS (V850ES/JC3-L (40-pin)) (Target)

30.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V _{DD}	V _{DD} = EV _{DD} = AV _{REF0}	-0.5 to +4.6	V
	EV _{DD}	V _{DD} = EV _{DD} = AV _{REF0}	-0.5 to +4.6	V
	AV _{REF0}	V _{DD} = EV _{DD} = AV _{REF0}	-0.5 to +4.6	V
	V _{SS}	V _{SS} = EV _{SS} = AV _{SS}	-0.5 to +0.5	V
	AV _{SS}	V _{SS} = EV _{SS} = AV _{SS}	-0.5 to +0.5	V
	EV _{SS}	V _{SS} = EV _{SS} = AV _{SS}	-0.5 to +0.5	V
Input voltage	V _{I1}	P97, P914, P915, PCM0, PDL5, RESET, FLMD0	-0.5 to EV _{DD} + 0.5 ^{Note 1}	V
	V _{I3}	X1	-0.5 to V _{DD} + 0.5 ^{Note 1}	V
		X2	-0.5 to V _{RO} ^{Note 2} + 0.5 ^{Note 1}	
	V _{I4}	P02, P03, P05, P30, P31, P40 to P42, P50 to P55, P90, P91, P96	-0.5 to +6.0	V
	V _{I5}	XT1, XT2	-0.5 to V _{RO} ^{Note 2} + 0.5	V
Analog input voltage	V _{IAN}	P70 to P74	-0.5 to AV _{REF0} + 0.5 ^{Note 1}	V

- Notes**
1. Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.
 2. On-chip regulator output voltage

- Cautions**
1. Do not directly connect the output (or I/O) pins of IC products to each other, or to V_{DD}, V_{CC}, and GND. Open-drain pins or open-collector pins, however, can be directly connected to each other. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.
 2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.
The ratings and conditions indicated for DC characteristics, AC characteristics, and operating conditions represent the quality assurance range during normal operation.

Remark Unless specified otherwise, the ratings of alternate-function pins are the same as those of port pins.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbol	Conditions	Ratings	Unit	
Output current, low	I _{OL}	P02, P03, P05, P30, P31, P40 to P42, P50 to P55, P90, P91, P96, P97, P914, P915	Per pin	4	mA
			Total of all pins	50	mA
		PCM0, PDL5	Per pin	4	mA
			Total of all pins	8	mA
		P70 to P74	Per pin	4	mA
			Total of all pins	20	mA
Output current, high	I _{OH}	P02, P03, P05, P30, P31, P40 to P42, P50 to P55, P90, P91, P96, P97, P914, P915	Per pin	-4	mA
			Total of all pins	-50	mA
		PCM0, PDL5	Per pin	-4	mA
			Total of all pins	-8	mA
		P70 to P74	Per pin	-4	mA
			Total of all pins	-20	mA
Operating ambient temperature	T _A	Normal operation mode	-40 to +85	°C	
		Flash memory programming mode	-40 to +85	°C	
Storage temperature	T _{stg}		-40 to +125	°C	

- Cautions**
1. Do not directly connect the output (or I/O) pins of IC products to each other, or to V_{DD}, V_{CC}, and GND. Open-drain pins or open-collector pins, however, can be directly connected to each other. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.
 2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded. The ratings and conditions indicated for DC characteristics, AC characteristics, and operating conditions represent the quality assurance range during normal operation.

Remark Unless specified otherwise, the ratings of alternate-function pins are the same as those of port pins.

30.2 Capacitance**Capacitance (T_A = 25°C, V_{DD} = EV_{DD} = AV_{REF0} = V_{SS} = EV_{SS} = AV_{SS} = 0 V)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
I/O capacitance	C _{IO}	f _x = 1 MHz Unmeasured pins returned to 0 V			10	pF

30.3 Operating Conditions

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

Internal System Clock	Conditions	Supply Voltage			Unit
		V_{DD}	EV_{DD}	AV_{REF0}	
$f_{xx} = 2.5$ to 20 MHz	$C = 4.7\ \mu\text{F}$, A/D converter operating	2.7 to 3.6	2.7 to 3.6	2.7 to 3.6	V
$f_{xx} = 2.5$ to 5 MHz	$C = 4.7\ \mu\text{F}$, A/D converter stopped	2.2 to 3.6	2.2 to 3.6	2.2 to 3.6	V
$f_{XT} = 32.768\text{ kHz}$	$C = 4.7\ \mu\text{F}$, A/D converter stopped	2.2 to 3.6	2.2 to 3.6	2.2 to 3.6	V

30.4 Oscillator Characteristics

30.4.1 Main clock oscillator characteristics

(1) oscillator characteristics

Main clock oscillator characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

Resonator	Circuit Example	Parameter	Conditions	MIN.	TYP.	MAX.	Unit		
Ceramic resonator/ Crystal resonator		Oscillation frequency (f_x) ^{Note 1}	Clock through mode	$V_{DD} = 2.2$ to 3.6 V	2.5		5	MHz	
				$V_{DD} = 2.7$ to 3.6 V	2.5		10	MHz	
				$V_{DD} = 2.7$ to 3.6 V in PLL mode	2.5		5	MHz	
			Oscillation stabilization time ^{Note 2}		$V_{DD} = 2.2$ to 3.6 V immediately after reset ends	Note 3	Note 4		s
		After STOP mode is released		$V_{DD} = 2.2$ to 3.6 V in clock through mode	1 ^{Note 5}	Note 6		ms	
				$V_{DD} = 2.7$ to 3.6 V in PLL mode	1 ^{Note 7}	Note 6		ms	
After IDLE2 mode is released	$V_{DD} = 2.2$ to 3.6 V in clock through mode	350 ^{Note 5}		Note 6		μs			
	$V_{DD} = 2.7$ to 3.6 V in PLL mode	800 ^{Note 7}	Note 6		μs				

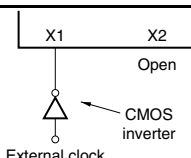
- Notes**
- The oscillation frequency shown above indicates only oscillator characteristics. Use the V850ES/JC3-L (40-pin) so that the internal operating conditions do not exceed the ratings shown in **AC Characteristics**, **DC Characteristics**, and **Operating Conditions**.
 - The wait time required from when the resonator starts oscillating until oscillation stabilizes.
 - The oscillation stabilization time after reset release is restricted in accordance with the set value of the option byte. For details, see **CHAPTER 27 OPTION BYTE**.
 - The oscillation stabilization time differs depending on the set value of the option byte. For details, see **CHAPTER 27 OPTION BYTE**.
 - Time required to set up the regulator and flash memory. Secure the setup time using the OSTS register.
 - The value varies depending on the setting of the OSTS register.
 - Time required to set up the regulator, flash memory, and PLL. Secure the setup time using the OSTS register.

Caution 1. When using the main clock oscillator, wire as follows in the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.

- Keep the wiring length as short as possible.
- Do not cross the wiring with the other signal lines.
- Do not route the wiring near a signal line through which a high fluctuating current flows.
- Always make the ground point of the oscillator capacitor the same potential as V_{SS} .
- Do not ground the capacitor to a ground pattern through which a high current flows.
- Do not fetch signals from the oscillator.

Caution 2. When the main clock is stopped and the device is operating on the subclock, wait until the oscillation stabilization time has been secured by the program before switching back to the main clock.

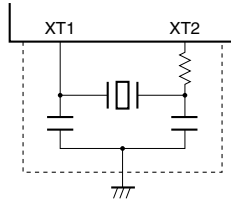
(2) External clock**($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = 2.7$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)**

Resonator	Circuit Example	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
External clock		Input frequency (f_x) ^{Note}	Clock through mode	2.5		5	MHz
			PLL mode	2.5		5	MHz
		V_{IH}	X1	2.3		V_{DD}	V
		V_{IL}	X1	V_{SS}		0.4	V

Note Keep the duty factor of the input waveform to within 45% to 55%.

- Cautions**
1. Be sure to disconnect the internal feedback resistor after reset (set PCC.MFRC = 1).
 2. Leave the X2 pin open.
 3. Make sure that the CMOS inverter is as close to the X1 pin as possible.

30.4.2 Subclock oscillator characteristics**($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)**

Resonator	Circuit Example	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		Oscillation frequency (f_{XT}) ^{Note 1}		32	32.768	35	kHz
		Oscillation stabilization time ^{Note 2}				10	s

- Notes**
1. The oscillation frequency shown above indicates only oscillator characteristics. Use the V850ES/JC3-L (40-pin) so that the internal operation conditions do not exceed the ratings shown in **AC Characteristics**, **DC Characteristics**, and operating conditions.
 2. Time required from when V_{DD} reaches the oscillation voltage range (2.2 V (MIN.)) to when the crystal resonator stabilizes.

Cautions 1. When using the subclock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- Keep the wiring length as short as possible.
 - Do not cross the wiring with the other signal lines.
 - Do not route the wiring near a signal line through which a high fluctuating current flows.
 - Always make the ground point of the oscillator capacitor the same potential as V_{SS} .
 - Do not ground the capacitor to a ground pattern through which a high current flows.
 - Do not fetch signals from the oscillator.
2. The subclock oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the main clock oscillator. Particular care is therefore required with the wiring method when the subclock is used.
 3. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

30.4.3 PLL characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input frequency	f_x		2.5		5	MHz
Output frequency	f_{xx}		10		20	MHz
Lock time	t_{PLL}	After V_{DD} reaches 2.7 V (MIN.)			800	μs

30.4.4 Internal oscillator characteristics

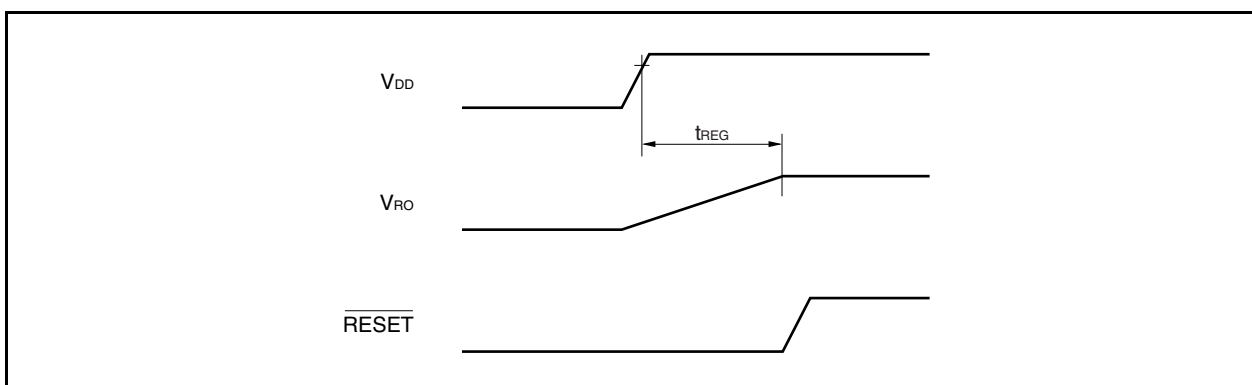
($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output frequency	f_R		100	220	400	kHz

30.5 Regulator Characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage	V_{DD}	$f_{xx} = 20\text{ MHz (MAX.)}$	2.7		3.6	V
		$f_{xx} = 5\text{ MHz (MAX.)}$	2.2		3.6	V
		Data retained (STOP mode)	1.9		3.6	V
Output voltage	V_{RO}	$V_{DD} = 2.7$ to 3.6 V		2.5		V
Regulator output stabilization time	t_{REG}	After V_{DD} reaches 2.7 V (MIN.), stabilization capacitance $C = 4.7\ \mu\text{F}$ (recommended value) connected to REGC pin			1	ms
		After V_{DD} reaches 2.2 V (MIN.), stabilization capacitance $C = 4.7\ \mu\text{F}$ (recommended value) connected to REGC pin			3.5	ms
External capacitance	REGC	Permissible error of external capacitance to be connected to REGC pin	3.76	4.70	5.64	μF



30.6 DC Characteristics

30.6.1 Pin characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$) (1/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	$\overline{\text{RESET}}$, FLMD0, P97, P914, P915	$0.8EV_{DD}$		EV_{DD}	V
	V_{IH2}	P02, P03, P05, P30, P31, P42, P50 to P55, P96	$0.8EV_{DD}$		5.5	V
	V_{IH3}	P40, P41, P90, P91	$0.7EV_{DD}$		5.5	V
	V_{IH4}	PCM0, PDL5	$0.7EV_{DD}$		EV_{DD}	V
	V_{IH5}	P70 to P74	$0.7AV_{REF0}$		AV_{REF0}	V
Input voltage, low	V_{IL1}	$\overline{\text{RESET}}$, FLMD0, P97, P914, P915	EV_{SS}		$0.2EV_{DD}$	V
	V_{IL2}	P02, P03, P05, P30, P31, P42, P50 to P55, P96	EV_{SS}		$0.2EV_{DD}$	V
	V_{IL3}	P40, P41, P90, P91	EV_{SS}		$0.3EV_{DD}$	V
	V_{IL4}	PCM0, PDL5	EV_{SS}		$0.3EV_{DD}$	V
	V_{IL5}	P70 to P74	AV_{SS}		$0.3AV_{REF0}$	V
Input leakage current, high	I_{LIH}	$V_I = V_{DD} = EV_{DD} = AV_{REF0}$			5	μA
Input leakage current, low	I_{LIL}	$V_I = 0\text{ V}$			-5	μA
Output leakage current, high	I_{LOH}	$V_O = V_{DD} = EV_{DD} = AV_{REF0}$			5	μA
Output leakage current, low	I_{LOL}	$V_O = 0\text{ V}$			-5	μA

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

(T_A = -40 to +85°C, V_{DD} = EV_{DD} = AV_{REF0}, V_{SS} = EV_{SS} = AV_{SS} = 0 V) (2/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	
Output voltage, high	V _{OH1}	P02, P03, P05, P30, P31, P40 to P42, P50 to P55, P90, P91, P96, P97, P914, P915	Per pin I _{OH} = -1.0 mA	Total of all pins -20 mA	EV _{DD} - 1.0		EV _{DD}	V
			Per pin I _{OH} = -100 μA	Total of all pins -2.0 mA	EV _{DD} - 0.5		EV _{DD}	V
	V _{OH2}	PCM0, PDL5	Per pin I _{OH} = -1.0 mA	Total of all pins -2 mA	EV _{DD} - 1.0		EV _{DD}	V
			Per pin I _{OH} = -100 μA	Total of all pins -0.2 mA	EV _{DD} - 0.5		EV _{DD}	V
	V _{OH3}	P70 to P74	Per pin I _{OH} = -0.4 mA	Total of all pins -2.0 mA	AV _{REF0} - 1.0		AV _{REF0}	V
			Per pin I _{OH} = -100 μA	Total of all pins -0.5 mA	AV _{REF0} - 0.5		AV _{REF0}	V
Output voltage, low	V _{OL1}	P02, P03, P05, P30, P31, P42, P50 to P55, P96, P97, P914, P915	Per pin I _{OL} = 1.0 mA	Total of all pins 20 mA	0		0.4	V
	V _{OL2}	P40, P41, P90, P91	Per pin I _{OL} = 3.0 mA		0		0.4	V
	V _{OL3}	PCM0, PDL5	Per pin I _{OL} = 1.0 mA	Total of all pins 2 mA	0		0.4	V
	V _{OL4}	P70 to P74	Per pin I _{OL} = 0.4 mA	Total of all pins 2.0 mA	0		0.4	V
Software pull-down resistor ^{Note}	R ₁	P05	V _I = V _{DD}		10	20	100	kΩ

Note $\overline{\text{DRST}}$ pin only (controlled by OCDM register)

- Remarks**
1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.
 2. When the I_{OH} and I_{OL} conditions are not satisfied for a pin but the total value of all pins is satisfied, only that pin does not satisfy the DC characteristics.

30.6.2 Supply current characteristics

(T_A = -40 to +85°C, V_{DD} = EV_{DD} = AV_{REF0}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP. ^{Note 1}	MAX. ^{Note 2}	Unit
Supply current ^{Note 3}	I _{DD1}	Normal operation	f _{XX} = 20 MHz (f _X = 5 MHz) ^{Note 4}	12 ^{Note 5}	20	mA
			f _{XX} = 10 MHz (f _X = 10 MHz), PLL off ^{Note 4}	6 ^{Note 5}	10	mA
	I _{DD2}	HALT mode	f _{XX} = 20 MHz (f _X = 5 MHz) ^{Note 4}	7.5	14	mA
	I _{DD3}	IDLE1 mode	f _{XX} = 5 MHz (f _X = 5 MHz), PLL off ^{Note 4}	0.6	1	mA
	I _{DD4}	IDLE2 mode	f _{XX} = 5 MHz (f _X = 5 MHz), PLL off ^{Note 4}	0.28	0.5	mA
	I _{DD5}	Subclock operation mode	f _{XT} = 32.768 kHz, main clock stopped, internal oscillator stopped, PLL off REGOVL0 = 02H (low-voltage subclock operation mode)	18		μA
	I _{DD6}	Sub-IDLE mode	f _{XT} = 32.768 kHz, main clock stopped, internal oscillator stopped, PLL off REGOVL0 = 02H (low-voltage sub-IDLE mode)	2.5	50	μA
	I _{DD7}	STOP mode	Subclock stopped, internal oscillator stopped REGOVL0 = 01H (low-voltage STOP mode) T _A = 25°C	1.5	3.0	μA
			Subclock stopped, internal oscillator stopped REGOVL0 = 01H (low-voltage STOP mode) T _A = 85°C		45	μA
			Subclock operating, internal oscillator stopped REGOVL0 = 01H (low-voltage STOP mode)	2.5	50	μA
I _{DD8}	Self programming mode	f _{XX} = 20 MHz (f _X = 5 MHz)	14	24	mA	
LVI current	I _{LVI}			1.2	3	μA
WDT, internal oscillation current	I _{WDT}			5		μA

- Notes**
1. TYP. current is a value at V_{DD} = EV_{DD} = 3.3 V, T_A = 25°C.
The TYP. value is not a value guaranteed for each device.
 2. MAX. current is a value at which the characteristic in question is at the worst-case value at V_{DD} = EV_{DD} = 3.6 V, T_A = -40 to +85°C.
 3. Total of V_{DD} and EV_{DD} currents. Currents I_{LVI} and I_{WDT} flowing through the output buffers, A/D converter, and on-chip pull-down resistor are not included.
 4. TYP. value indicates the current value when watch timer + TMM (count by watch timer interrupt) operate as peripheral functions.
MAX. value indicates the current value when all the functions operable in a range in which the pin status is not changed operate as peripheral functions.
However, I_{LVI} and I_{WDT} are excluded.
 5. TYP. value of I_{DD1} is a value when all instructions are executed + RAM access 15%.

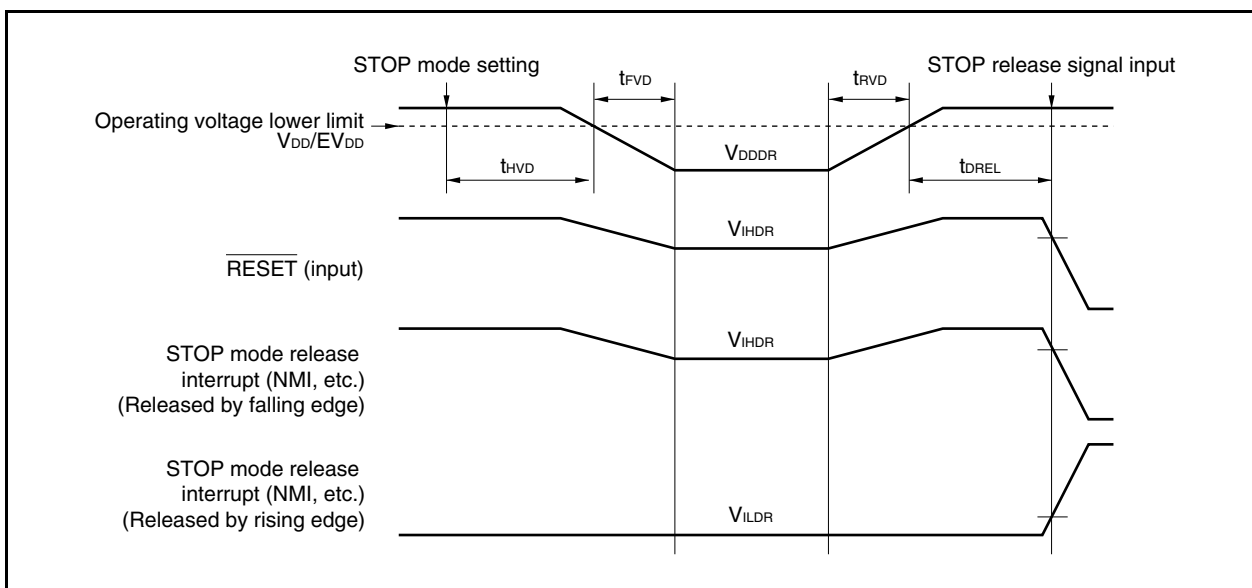
Remark For details about the operating voltage, see 30.3 Operating Conditions.

30.6.3 Data retention characteristics (in STOP mode)

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention voltage	V_{DDDR}	STOP mode (all functions stopped)	1.9		3.6	V
Data retention current	I_{DDDR}	Subclock stopped, internal oscillator stopped $T_A = 85^\circ\text{C}$			45	μA
Supply voltage rise time	t_{RVD}		200			μs
Supply voltage fall time	t_{FVD}		200			μs
Supply voltage retention time	t_{HVD}	After STOP mode setting	0			ms
STOP release signal input time	t_{DREL}	After V_{DD} reaches the operating voltage MIN. (see 30.3 Operating Conditions)	0			ms
Data retention input voltage, high	V_{IHDR}	$V_{DD} = EV_{DD} = V_{DDDR}$	$0.9V_{DDDR}$		V_{DDDR}	V
Data retention input voltage, low	V_{ILDR}	$V_{DD} = EV_{DD} = V_{DDDR}$	0		$0.1V_{DDDR}$	V

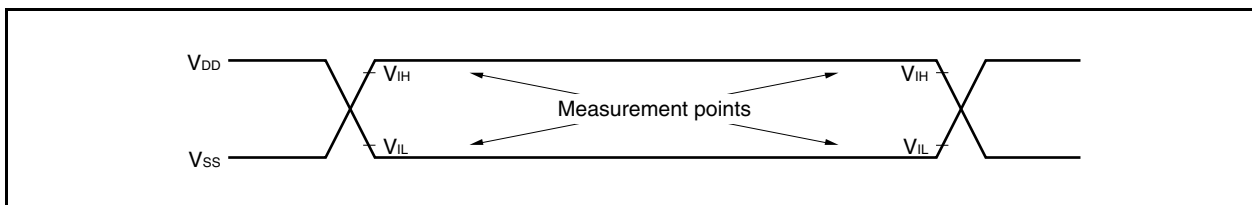
Caution Shifting to STOP mode and restoring from STOP mode must be performed within the rated operating range.



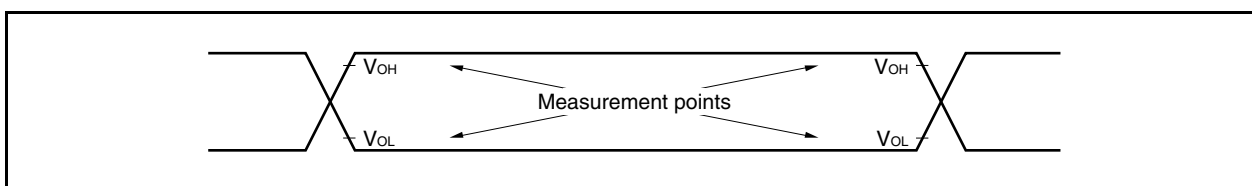
30.7 AC Characteristics

30.7.1 Measurement conditions

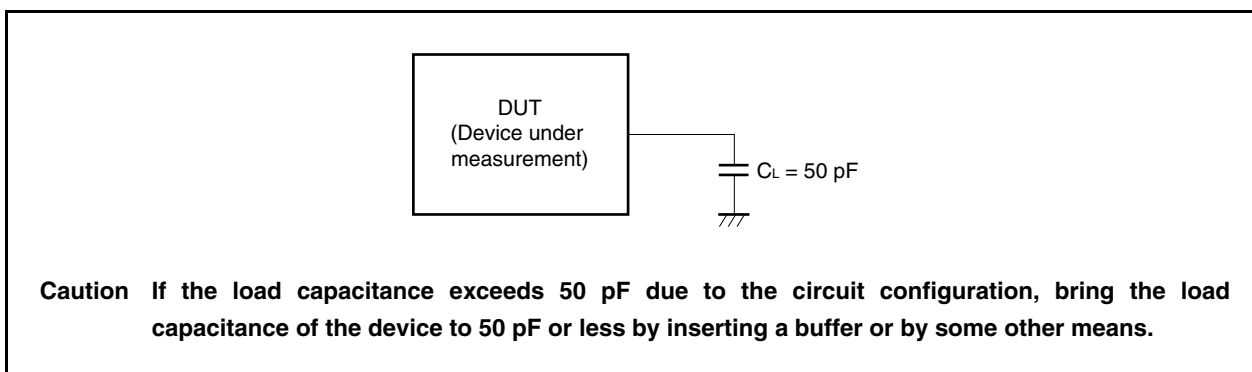
(1) AC test input measurement points



(2) AC test output measurement points



(3) Load conditions



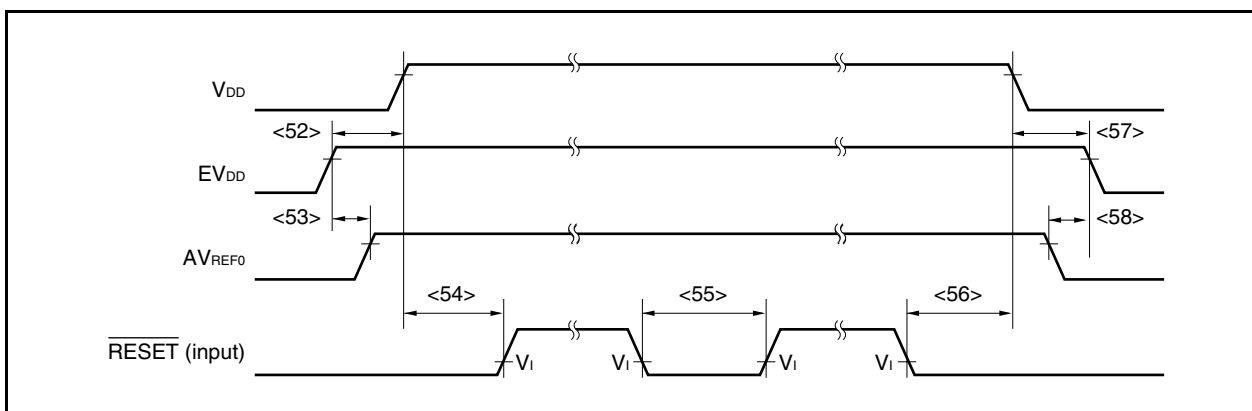
30.7.2 Power on/power off/reset timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
$EV_{DD}\uparrow \rightarrow V_{DD}\uparrow$	t_{REL}	<52>	0		ns
$EV_{DD}\uparrow \rightarrow AV_{REF0}\uparrow$	t_{REA}	<53>	0	t_{REL}	ns
$V_{DD}\uparrow \rightarrow \overline{RESET}\uparrow$	t_{RER}	<54>	$500 + t_{REG}^{\text{Note}}$		ns
\overline{RESET} low-level width	t_{WRSL}	<55>	500		ns
$\overline{RESET}\downarrow \rightarrow V_{DD}\downarrow$	t_{FRE}	<56>	500		ns
$V_{DD}\downarrow \rightarrow EV_{DD}\downarrow$	t_{FEL}	<57>	0		ns
$AV_{REF0}\downarrow \rightarrow EV_{DD}\downarrow$	t_{FEA}	<58>	0	t_{FEL}	ns

Note See 30.5 Regulator Characteristics.

Remark The \overline{RESET} pin has an analog noise elimination function.



30.8 Peripheral Function Characteristics

30.8.1 Interrupt timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
NMI high-level width	t_{WNIH}		500		ns
NMI low-level width	t_{WNIL}		500		ns
INTPn ^{Note} high-level width	t_{WITH}	$n = 0, 2, 5$ to 7	500		ns
INTPn ^{Note} low-level width	t_{WITL}	$n = 0, 2, 5$ to 7	500		ns

Note The characteristics of INTPn is the same as the \overline{DRST} pin (P05/INTP2/ \overline{DRST}).

Remark The NMI and INTPn pins have the analog noise elimination function ($n = 0, 2, 5$ to 7).

30.8.2 Key return timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
KRn high-level width	t_{WKRH}		500		ns
KRn low-level width	t_{WKRL}		500		ns

Remarks 1. $n = 0$ to 7

2. The KRn pin has an analog noise elimination function.

30.8.3 Timer timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
TI high-level width	t_{TIH}	TIP20, TIP21, TIP50, TIP51,	$2T + 20$		ns
TI low-level width	t_{TIL}	TIQ00 to TIQ03	$2T + 20$		ns

Remark $T = 1/f_{xx}$

30.8.4 UART timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0}$, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Transmit rate		$V_{DD} = 2.2$ to 3.6 V		625	kbps
ASCK0 frequency		$V_{DD} = 2.2$ to 3.6 V		5	MHz
		$V_{DD} = 2.7$ to 3.6 V		10	MHz

30.8.5 CSIB timing

(1) Master mode

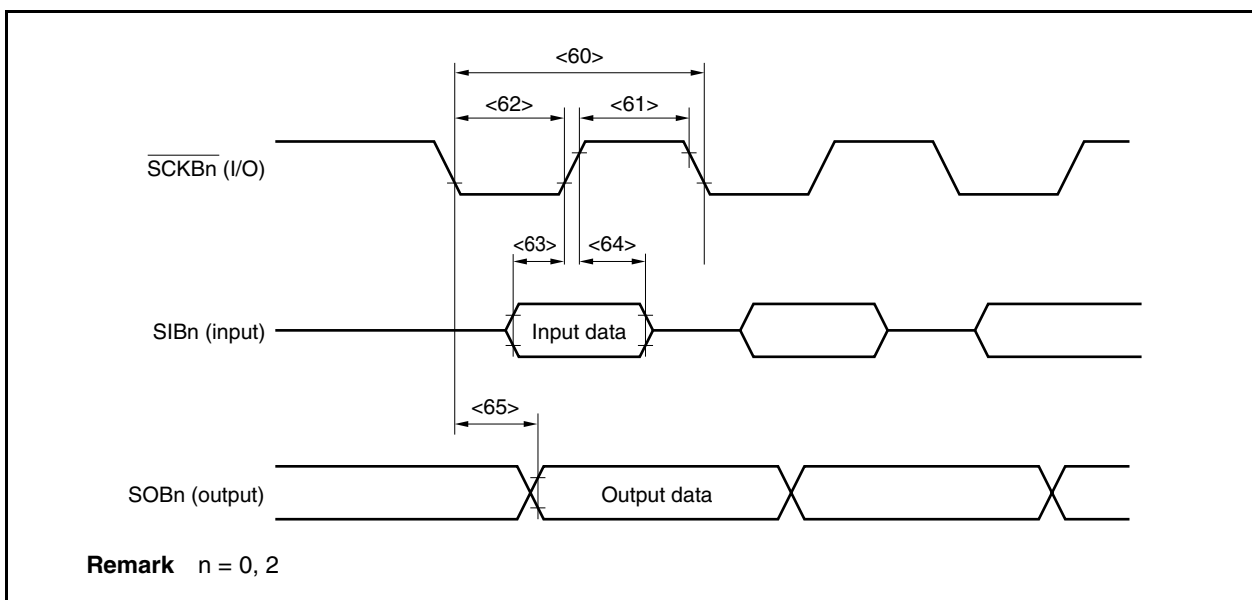
($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0}$, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKBn}}$ cycle time	t_{KY1}	$\langle 60 \rangle$ 2.7 V $\leq V_{DD} \leq 3.6$ V	125		ns
		2.2 V $\leq V_{DD} < 2.7$ V	800		ns
$\overline{\text{SCKBn}}$ high-level width	t_{KH1}	$\langle 61 \rangle$ 2.7 V $\leq V_{DD} \leq 3.6$ V	$t_{KY1}/2 - 8$		ns
		2.2 V $\leq V_{DD} < 2.7$ V	$t_{KY1}/2 - 80$		ns
$\overline{\text{SCKBn}}$ low-level width	t_{KL1}	$\langle 62 \rangle$ 2.7 V $\leq V_{DD} \leq 3.6$ V	$t_{KY1}/2 - 8$		ns
		2.2 V $\leq V_{DD} < 2.7$ V	$t_{KY1}/2 - 80$		ns
SIBn setup time (to $\overline{\text{SCKBn}}\uparrow$)	t_{SIK1}	$\langle 63 \rangle$ 2.7 V $\leq V_{DD} \leq 3.6$ V	27		ns
		2.2 V $\leq V_{DD} < 2.7$ V	100		ns
SIBn hold time (from $\overline{\text{SCKBn}}\uparrow$)	t_{KSH1}	$\langle 64 \rangle$ 2.7 V $\leq V_{DD} \leq 3.6$ V	27		ns
		2.2 V $\leq V_{DD} < 2.7$ V	100		ns
Delay time from $\overline{\text{SCKBn}}\downarrow$ to SOBn output	t_{KSO1}	$\langle 65 \rangle$ 2.7 V $\leq V_{DD} \leq 3.6$ V		27	ns
		2.2 V $\leq V_{DD} < 2.7$ V		95	ns

Remark n = 0, 2

(2) Slave mode**($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)**

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKBn}}$ cycle time	$t_{\text{KCY}2}$	$2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	125		ns
		$2.2\text{ V} \leq V_{DD} < 2.7\text{ V}$	800		ns
$\overline{\text{SCKBn}}$ high-level width	$t_{\text{KH}2}$	$2.2\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	54.5		ns
$\overline{\text{SCKBn}}$ low-level width	$t_{\text{KL}2}$	$2.2\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	54.5		ns
SIBn setup time (to $\overline{\text{SCKBn}}\uparrow$)	$t_{\text{SIK}2}$	$2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	27		ns
		$2.2\text{ V} \leq V_{DD} < 2.7\text{ V}$	100		ns
SIBn hold time (from $\overline{\text{SCKBn}}\uparrow$)	$t_{\text{KSI}2}$	$2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	27		ns
		$2.2\text{ V} \leq V_{DD} < 2.7\text{ V}$	100		ns
Delay time from $\overline{\text{SCKBn}}\downarrow$ to SOBn output	$t_{\text{KSO}2}$	$2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$		27	ns
		$2.2\text{ V} \leq V_{DD} < 2.7\text{ V}$		95	ns

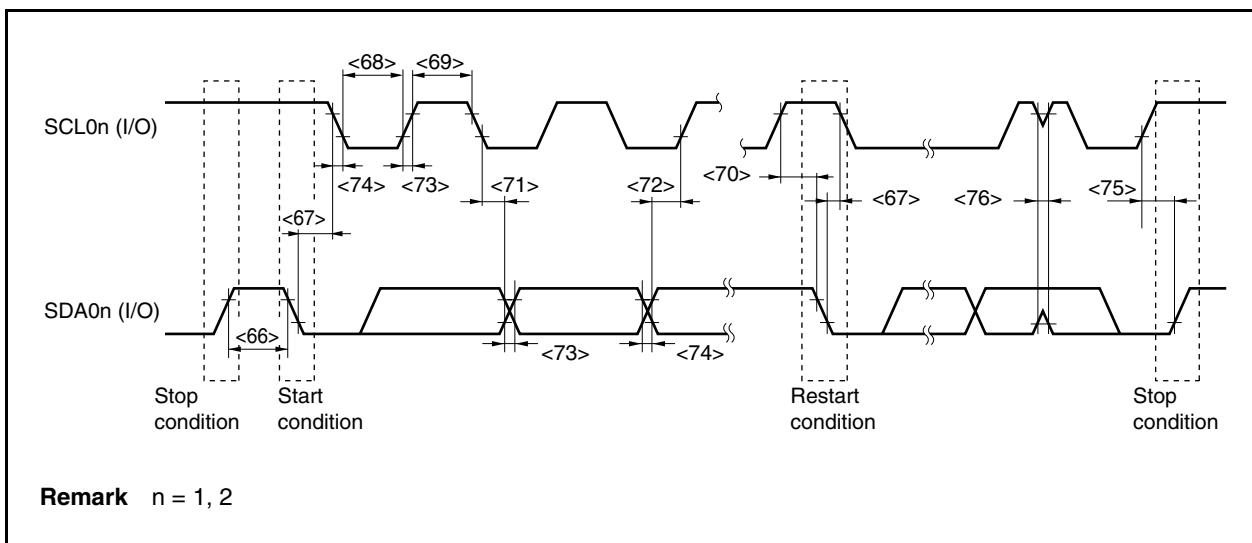
Remark n = 0, 2

30.8.6 I²C bus mode(T_A = -40 to +85°C, V_{DD} = EV_{DD} = AV_{REF0} = 2.2 to 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol		Normal Mode		High-Speed Mode		Unit	
			MIN.	MAX.	MIN.	MAX.		
SCL0n clock frequency	f _{CLK}		0	100	0	400	kHz	
Bus free time (Between start and stop conditions)	t _{BUF}	<66>	4.7	–	1.3	–	μs	
Hold time ^{Note 1}	t _{HD:STA}	<67>	4.0	–	0.6	–	μs	
SCL0n clock low-level width	t _{LOW}	<68>	4.7	–	1.3	–	μs	
SCL0n clock high-level width	t _{HIGH}	<69>	4.0	–	0.6	–	μs	
Setup time for start/restart conditions	t _{SU:STA}	<70>	4.7	–	0.6	–	μs	
Data hold time	CBUS compatible master	t _{HD:DAT}	<71>	5.0	–	–	–	μs
				0 ^{Note 2}	–	0 ^{Note 2}	0.9 ^{Note 3}	μs
Data setup time	I ² C mode	t _{SU:DAT}	<72>	250	–	100 ^{Note 4}	–	ns
				–	–	–	–	–
SDA0n and SCL0n signal rise time	t _R	<73>	–	1000	20 + 0.1Cb ^{Note 5}	300	ns	
SDA0n and SCL0n signal fall time	t _F	<74>	–	300	20 + 0.1Cb ^{Note 5}	300	ns	
Stop condition setup time	t _{SU:STO}	<75>	4.0	–	0.6	–	μs	
Pulse width of spike suppressed by input filter	t _{SP}	<76>	–	–	0	50	ns	
Capacitance load of each bus line	Cb		–	400	–	400	pF	

- Notes**
- At the start condition, the first clock pulse is generated after the hold time.
 - The system requires a minimum of 300 ns hold time internally for the SDA0n signal (at V_{IHmin.} of SCL0n signal) in order to occupy the undefined area at the falling edge of SCL0n.
 - If the system does not extend the SCL0n signal low hold time (t_{LOW}), only the maximum data hold time (t_{HD:DAT}) needs to be satisfied.
 - The high-speed mode I²C bus can be used in the normal-mode I²C bus system. In this case, set the high-speed mode I²C bus so that it meets the following conditions.
 - If the system does not extend the SCL0n signal's low state hold time:
t_{SU:DAT} ≥ 250 ns
 - If the system extends the SCL0n signal's low state hold time:
Transmit the following data bit to the SDA0n line prior to the SCL0n line release (t_{Rmax.} + t_{SU:DAT} = 1,000 + 250 = 1,250 ns: Normal mode I²C bus specification).
 - Cb: Total capacitance of one bus line (unit: pF)

Remark n = 1, 2

I²C Bus Timing

30.8.7 A/D converter

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0}$, $2.7\text{ V} \leq AV_{REF0} \leq 3.6\text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution					10	bit
Overall error ^{Note}		$2.7\text{ V} \leq AV_{REF0} \leq 3.6\text{ V}$			± 0.6	%FSR
A/D conversion time	t_{CONV}	$3.0\text{ V} \leq AV_{REF0} \leq 3.6\text{ V}$	2.6		24	μs
		$2.7\text{ V} \leq AV_{REF0} \leq 3.0\text{ V}$	3.9		24	μs
Zero scale error					± 0.5	%FSR
Full scale error					± 0.5	%FSR
Non-linearity error					± 4.0	LSB
Differential linearity error					± 4.0	LSB
Analog input voltage	V_{IAN}		AV_{SS}		AV_{REF0}	V
Reference voltage	AV_{REF0}		2.7		3.6	V
AV_{REF0} current	AI_{REF0}	Normal conversion mode		3	6.5	mA
		High-speed conversion mode		4	10	mA
		When A/D converter unused			5	μA

Note Excluding quantization error ($\pm 0.05\%$ FSR).

Caution Do not set (read/write) alternate-function ports during A/D conversion; otherwise the conversion resolution may be degraded.

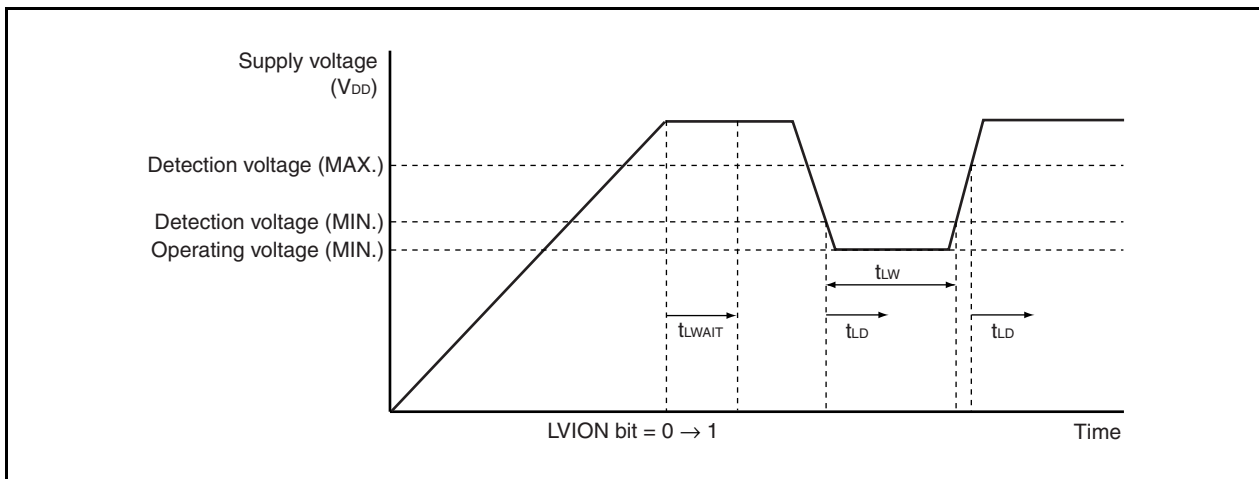
Remark LSB: Least Significant Bit
FSR: Full Scale Range

30.8.8 LVI circuit characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LV10}		2.7	2.8	2.9	V
	V_{LV11}		2.2	2.3	2.4	V
Response time ^{Note}	t_{LD}	At rising edge: After V_{DD} reaches V_{LV10}/V_{LV11} (MAX.) At falling edge: After V_{DD} has dropped to V_{LV10}/V_{LV11} (MIN.)		0.2	2.0	ms
Minimum pulse width	t_{LW}	$V_{DD} = V_{LV10}/V_{LV11}$ (MIN.)	0.2			ms
Reference voltage stabilization wait time	t_{LWAIT}	After V_{DD} reaches V_{LV10}/V_{LV11} (MAX.)		0.1	0.2	ms

Note Time required to detect the detection voltage and output an interrupt or reset signal.



30.9 Flash Memory Programming Characteristics

(1) Basic characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = 2.7$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

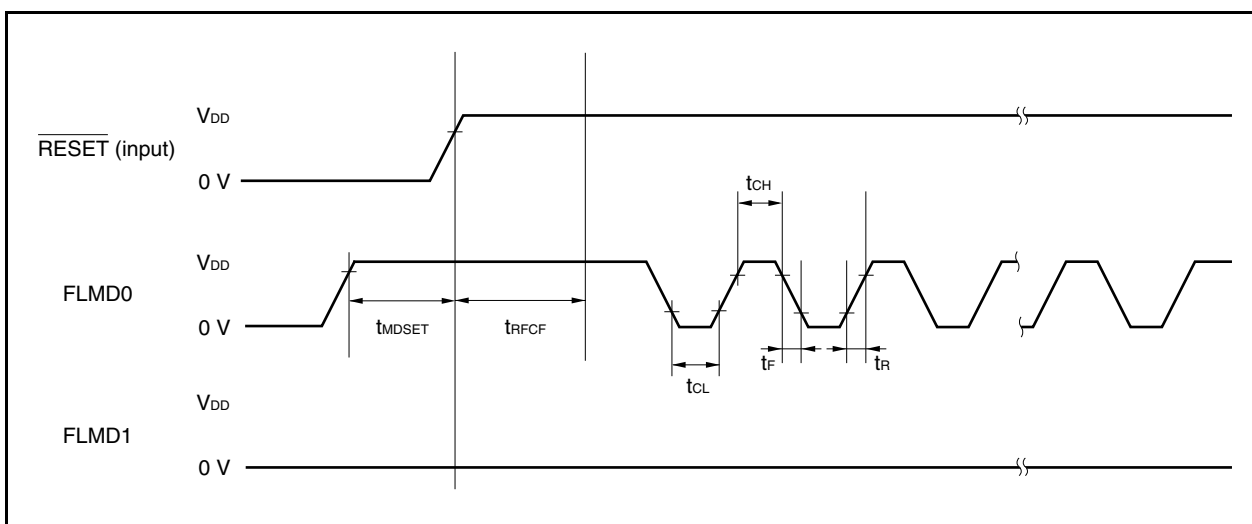
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Operating frequency	f_{CPU}		2.5		20	MHz	
Supply voltage	V_{DD}	$2.5 \text{ MHz} \leq f_{xx} \leq 20 \text{ MHz}$	2.7		3.6	V	
Number of rewrites	C_{WRT}	Used for updating programs When using flash memory programmer and Renesas Electronics self programming library	Retained for 15 years	1,000			times
		Used for updating data When using Renesas Electronics EEPROM emulation library (usable ROM size: 12 KB of 6 consecutive blocks, or 6 KB of 3 consecutive blocks)	Retained for 5 years	10,000			times
Programming temperature	t_{PRG}		-40		+85	$^\circ\text{C}$	

(2) Serial write operation characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = 2.7$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
FLMD0, FLMD1 setup time	t_{MDSET}		2		3000	ms
FLMD0 count start time from $\overline{\text{RESET}}\uparrow$	t_{RFCF}	$f_x = 2.5$ to 10 MHz	800			μs
FLMD0 counter high-level width/ low-level width	t_{CH}/t_{CL}		10		100	μs
FLMD0 counter rise time/fall time	t_r/t_f				1	μs

Flash write mode setup timing



(3) Programming characteristics**(T_A = -40 to +85°C, V_{DD} = EV_{DD} = AV_{REF0} = 2.7 to 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Chip erase time		f _{xx} = 20 MHz (when the chip erase command is executed)		105		ms
Write time per 256 bytes		f _{xx} = 20 MHz		2.0		ms
Block internal verify time		f _{xx} = 20 MHz		10		ms
Block blank check time		f _{xx} = 20 MHz		0.5		ms
Flash memory information setting time		f _{xx} = 20 MHz		30		ms

Remark Block size = 2 KB

CHAPTER 31 ELECTRICAL SPECIFICATIONS (V850ES/JC3-L (48-pin)) (Target)

31.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V _{DD}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 to +4.6	V
	EV _{DD}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 to +4.6	V
	AV _{REF0}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 to +4.6	V
	AV _{REF1}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 to +4.6	V
	V _{SS}	V _{SS} = EV _{SS} = AV _{SS}	-0.5 to +0.5	V
	AV _{SS}	V _{SS} = EV _{SS} = AV _{SS}	-0.5 to +0.5	V
	EV _{SS}	V _{SS} = EV _{SS} = AV _{SS}	-0.5 to +0.5	V
Input voltage	V _{I1}	P97 to P99, P914, P915, PCM0, PDL5, $\overline{\text{RESET}}$, FLMD0	-0.5 to EV _{DD} + 0.5 ^{Note 1}	V
	V _{I2}	P10	-0.5 to AV _{REF1} + 0.5 ^{Note 1}	V
	V _{I3}	X1	-0.5 to V _{DD} + 0.5 ^{Note 1}	V
		X2	-0.5 to V _{RO} ^{Note 2} + 0.5 ^{Note 1}	
	V _{I4}	P02, P03, P05, P30 to P32, P38, P39, P40 to P42, P50 to P55, P90, P91, P96	-0.5 to +6.0	V
V _{I5}	XT1, XT2	-0.5 to V _{RO} ^{Note 2} + 0.5	V	
Analog input voltage	V _{IAN}	P70 to P75	-0.5 to AV _{REF0} + 0.5 ^{Note 1}	V

Notes 1. Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

2. On-chip regulator output voltage

Cautions 1. Do not directly connect the output (or I/O) pins of IC products to each other, or to V_{DD}, V_{CC}, and GND. Open-drain pins or open-collector pins, however, can be directly connected to each other. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

The ratings and conditions indicated for DC characteristics, AC characteristics, and operating conditions represent the quality assurance range during normal operation.

Remark Unless specified otherwise, the ratings of alternate-function pins are the same as those of port pins.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbol	Conditions	Ratings	Unit	
Output current, low	I _{OL}	P02, P03, P05, P30 to P32, P38, P39, P40 to P42, P50 to P55, P90, P91, P96 to P99, P914, P915	Per pin	4	mA
			Total of all pins	50	mA
		PCM0, PDL5	Per pin	4	mA
			Total of all pins	8	mA
		P10		4	mA
		P70 to P75	Per pin	4	mA
Total of all pins	20		mA		
Output current, high	I _{OH}	P02, P03, P05, P30 to P32, P38, P39, P40 to P42, P50 to P55, P90, P91, P96 to P99, P914, P915	Per pin	-4	mA
			Total of all pins	-50	mA
		PCM0, PDL5	Per pin	-4	mA
			Total of all pins	-8	mA
		P10		-4	mA
		P70 to P75	Per pin	-4	mA
Total of all pins	-20		mA		
Operating ambient temperature	T _A	Normal operation mode	-40 to +85	°C	
		Flash memory programming mode	-40 to +85	°C	
Storage temperature	T _{stg}		-40 to +125	°C	

Cautions 1. Do not directly connect the output (or I/O) pins of IC products to each other, or to V_{DD}, V_{CC}, and GND. Open-drain pins or open-collector pins, however, can be directly connected to each other. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

The ratings and conditions indicated for DC characteristics, AC characteristics, and operating conditions represent the quality assurance range during normal operation.

Remark Unless specified otherwise, the ratings of alternate-function pins are the same as those of port pins.

31.2 Capacitance**Capacitance (T_A = 25°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = V_{SS} = EV_{SS} = AV_{SS} = 0 V)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
I/O capacitance	C _{IO}	f _x = 1 MHz Unmeasured pins returned to 0 V			10	pF

31.3 Operating Conditions

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

Internal System Clock	Conditions	Supply Voltage			Unit
		V_{DD}	EV_{DD}	AV_{REF0} , AV_{REF1}	
$f_{xx} = 2.5$ to 20 MHz	$C = 4.7\ \mu\text{F}$, A/D converter operating, D/A converter operating	2.7 to 3.6	2.7 to 3.6	2.7 to 3.6	V
$f_{xx} = 2.5$ to 5 MHz	$C = 4.7\ \mu\text{F}$, A/D converter stopped, D/A converter stopped	2.2 to 3.6	2.2 to 3.6	2.2 to 3.6	V
$f_{XT} = 32.768\text{ kHz}$	$C = 4.7\ \mu\text{F}$, A/D converter stopped, D/A converter stopped	2.2 to 3.6	2.2 to 3.6	2.2 to 3.6	V

31.4 Oscillator Characteristics

31.4.1 Main clock oscillator characteristics

(1) oscillator characteristics

Main clock oscillator characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

Resonator	Circuit Example	Parameter	Conditions	MIN.	TYP.	MAX.	Unit		
Ceramic resonator/ Crystal resonator		Oscillation frequency (f_x) ^{Note 1}	Clock through mode	$V_{DD} = 2.2$ to 3.6 V	2.5		5	MHz	
				$V_{DD} = 2.7$ to 3.6 V	2.5		10	MHz	
			$V_{DD} = 2.7$ to 3.6 V in PLL mode		2.5		5	MHz	
		Oscillation stabilization time ^{Note 2}	$V_{DD} = 2.2$ to 3.6 V immediately after reset ends		Note 3	Note 4			s
			After STOP mode is released	$V_{DD} = 2.2$ to 3.6 V in clock through mode	1 ^{Note 5}	Note 6			ms
				$V_{DD} = 2.7$ to 3.6 V in PLL mode	1 ^{Note 7}	Note 6			ms
After IDLE2 mode is released	$V_{DD} = 2.2$ to 3.6 V in clock through mode		350 ^{Note 5}	Note 6			μs		
	$V_{DD} = 2.7$ to 3.6 V in PLL mode	800 ^{Note 7}	Note 6			μs			

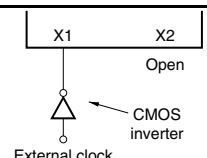
- Notes**
- The oscillation frequency shown above indicates only oscillator characteristics. Use the V850ES/JC3-L (48-pin) so that the internal operating conditions do not exceed the ratings shown in **AC Characteristics**, **DC Characteristics**, and **Operating Conditions**.
 - The wait time required from when the resonator starts oscillating until oscillation stabilizes.
 - The oscillation stabilization time after reset release is restricted in accordance with the set value of the option byte. For details, see **CHAPTER 27 OPTION BYTE**.
 - The oscillation stabilization time differs depending on the set value of the option byte. For details, see **CHAPTER 27 OPTION BYTE**.
 - Time required to set up the regulator and flash memory. Secure the setup time using the OSTS register.
 - The value varies depending on the setting of the OSTS register.
 - Time required to set up the regulator, flash memory, and PLL. Secure the setup time using the OSTS register.

Caution 1. When using the main clock oscillator, wire as follows in the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.

- Keep the wiring length as short as possible.
- Do not cross the wiring with the other signal lines.
- Do not route the wiring near a signal line through which a high fluctuating current flows.
- Always make the ground point of the oscillator capacitor the same potential as V_{SS} .
- Do not ground the capacitor to a ground pattern through which a high current flows.
- Do not fetch signals from the oscillator.

Caution 2. When the main clock is stopped and the device is operating on the subclock, wait until the oscillation stabilization time has been secured by the program before switching back to the main clock.

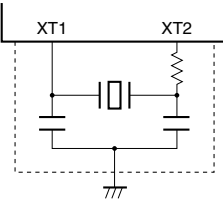
(2) External clock**($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.7$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)**

Resonator	Circuit Example	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
External clock		Input frequency (f_x) ^{Note}	Clock through mode	2.5		5	MHz
			PLL mode	2.5		5	MHz
		V_{IH}	X1	2.3		V_{DD}	V
		V_{IL}	X1	V_{SS}		0.4	V

Note Keep the duty factor of the input waveform to within 45% to 55%.

- Cautions**
1. Be sure to disconnect the internal feedback resistor after reset (set PCC.MFRC = 1).
 2. Leave the X2 pin open.
 3. Make sure that the CMOS inverter is as close to the X1 pin as possible.

31.4.2 Subclock oscillator characteristics**($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)**

Resonator	Circuit Example	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		Oscillation frequency (f_{XT}) ^{Note 1}		32	32.768	35	kHz
		Oscillation stabilization time ^{Note 2}				10	s

- Notes**
1. The oscillation frequency shown above indicates only oscillator characteristics. Use the V850ES/JC3-L (48-pin) so that the internal operation conditions do not exceed the ratings shown in **AC Characteristics**, **DC Characteristics**, and operating conditions.
 2. Time required from when V_{DD} reaches the oscillation voltage range (2.2 V (MIN.)) to when the crystal resonator stabilizes.

Cautions 1. When using the subclock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- Keep the wiring length as short as possible.
 - Do not cross the wiring with the other signal lines.
 - Do not route the wiring near a signal line through which a high fluctuating current flows.
 - Always make the ground point of the oscillator capacitor the same potential as V_{SS} .
 - Do not ground the capacitor to a ground pattern through which a high current flows.
 - Do not fetch signals from the oscillator.
2. The subclock oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the main clock oscillator. Particular care is therefore required with the wiring method when the subclock is used.
 3. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

31.4.3 PLL characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input frequency	f_x		2.5		5	MHz
Output frequency	f_{xx}		10		20	MHz
Lock time	t_{PLL}	After V_{DD} reaches 2.7 V (MIN.)			800	μs

31.4.4 Internal oscillator characteristics

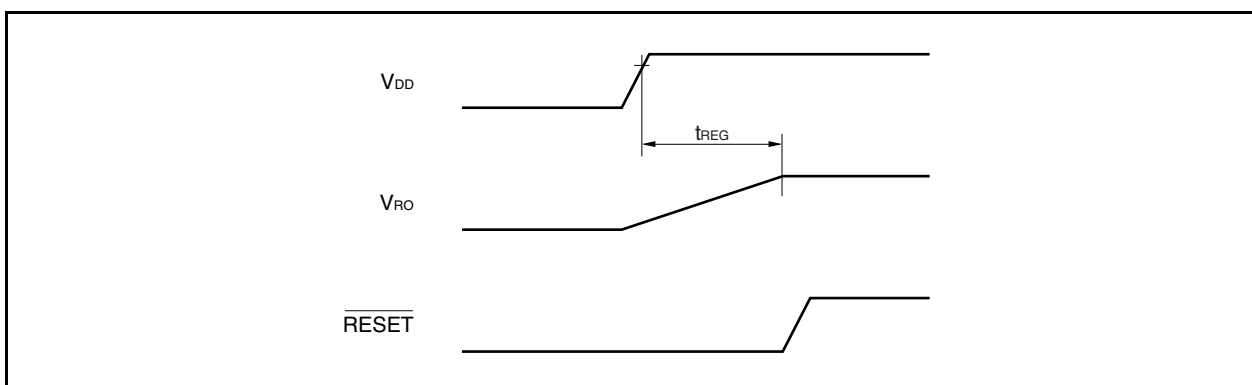
($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output frequency	f_R		100	220	400	kHz

31.5 Regulator Characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage	V_{DD}	$f_{xx} = 20$ MHz (MAX.)	2.7		3.6	V
		$f_{xx} = 5$ MHz (MAX.)	2.2		3.6	V
		Data retained (STOP mode)	1.9		3.6	V
Output voltage	V_{RO}	$V_{DD} = 2.7$ to 3.6 V		2.5		V
Regulator output stabilization time	t_{REG}	After V_{DD} reaches 2.7 V (MIN.), stabilization capacitance $C = 4.7$ μF (recommended value) connected to REGC pin			1	ms
		After V_{DD} reaches 2.2 V (MIN.), stabilization capacitance $C = 4.7$ μF (recommended value) connected to REGC pin			3.5	ms
External capacitance	REGC	Permissible error of external capacitance to be connected to REGC pin	3.76	4.70	5.64	μF



31.6 DC Characteristics

31.6.1 Pin characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$) (1/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	$\overline{\text{RESET}}$, FLMD0, P97 to P99, P914, P915	$0.8EV_{DD}$		EV_{DD}	V
	V_{IH2}	P02, P03, P05, P30 to P32, P42, P50 to P55, P96	$0.8EV_{DD}$		5.5	V
	V_{IH3}	P38, P39, P40, P41, P90, P91	$0.7EV_{DD}$		5.5	V
	V_{IH4}	PCM0, PDL5	$0.7EV_{DD}$		EV_{DD}	V
	V_{IH5}	P70 to P75	$0.7AV_{REF0}$		AV_{REF0}	V
	V_{IH6}	P10	$0.7AV_{REF1}$		AV_{REF1}	V
Input voltage, low	V_{IL1}	$\overline{\text{RESET}}$, FLMD0, P97 to P99, P914, P915	EV_{SS}		$0.2EV_{DD}$	V
	V_{IL2}	P02, P03, P05, P30 to P32, P42, P50 to P55, P96	EV_{SS}		$0.2EV_{DD}$	V
	V_{IL3}	P38, P39, P40, P41, P90, P91	EV_{SS}		$0.3EV_{DD}$	V
	V_{IL4}	PCM0, PDL5	EV_{SS}		$0.3EV_{DD}$	V
	V_{IL5}	P70 to P75	AV_{SS}		$0.3AV_{REF0}$	V
	V_{IL6}	P10	AV_{SS}		$0.3AV_{REF1}$	V
Input leakage current, high	I_{LH}	$V_I = V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$			5	μA
Input leakage current, low	I_{LL}	$V_I = 0\text{ V}$			-5	μA
Output leakage current, high	I_{LOH}	$V_O = V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$			5	μA
Output leakage current, low	I_{LOL}	$V_O = 0\text{ V}$			-5	μA

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

(T_A = -40 to +85°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V) (2/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	
Output voltage, high	V _{OH1}	P02, P03, P05, P30 to P32, P38, P39, P40 to P42, P50 to P55, P90, P91, P96 to P99, P914, P915	Per pin I _{OH} = -1.0 mA	Total of all pins -20 mA	EV _{DD} - 1.0		EV _{DD}	V
			Per pin I _{OH} = -100 μA	Total of all pins -2.5 mA	EV _{DD} - 0.5		EV _{DD}	V
	V _{OH2}	PCM0, PDL5	Per pin I _{OH} = -1.0 mA	Total of all pins -2 mA	EV _{DD} - 1.0		EV _{DD}	V
			Per pin I _{OH} = -100 μA	Total of all pins -0.2 mA	EV _{DD} - 0.5		EV _{DD}	V
	V _{OH3}	P70 to P75	Per pin I _{OH} = -0.4 mA	Total of all pins -2.4 mA	AV _{REF0} - 1.0		AV _{REF0}	V
			Per pin I _{OH} = -100 μA	Total of all pins -0.6 mA	AV _{REF0} - 0.5		AV _{REF0}	V
	V _{OH4}	P10	I _{OH} = -0.4 mA		AV _{REF1} - 1.0		AV _{REF1}	V
			I _{OH} = -100 μA		AV _{REF1} - 0.5		AV _{REF1}	V
Output voltage, low	V _{OL1}	P02, P03, P05, P30 to P32, P42, P50 to P55, P96 to P99, P914, P915	Per pin I _{OL} = 1.0 mA	Total of all pins 20 mA	0		0.4	V
	V _{OL2}	P38, P39, P40, P41, P90, P91	Per pin I _{OL} = 3.0 mA		0		0.4	V
	V _{OL3}	PCM0, PDL5	Per pin I _{OL} = 1.0 mA	Total of all pins 2 mA	0		0.4	V
	V _{OL4}	P10, P70 to P75	Per pin I _{OL} = 0.4 mA	Total of all pins 2.8 mA	0		0.4	V
Software pull-down resistor ^{Note}	R ₁	P05	V _I = V _{DD}		10	20	100	kΩ

Note $\overline{\text{DRST}}$ pin only (controlled by OCDM register)

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

2. When the I_{OH} and I_{OL} conditions are not satisfied for a pin but the total value of all pins is satisfied, only that pin does not satisfy the DC characteristics.

31.6.2 Supply current characteristics

(T_A = -40 to +85°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP. ^{Note 1}	MAX. ^{Note 2}	Unit
Supply current ^{Note 3}	I _{DD1}	Normal operation	f _{XX} = 20 MHz (f _X = 5 MHz) ^{Note 4}	12 ^{Note 5}	20	mA
			f _{XX} = 10 MHz (f _X = 10 MHz), PLL off ^{Note 4}	6 ^{Note 5}	10	mA
	I _{DD2}	HALT mode	f _{XX} = 20 MHz (f _X = 5 MHz) ^{Note 4}	7.5	14	mA
	I _{DD3}	IDLE1 mode	f _{XX} = 5 MHz (f _X = 5 MHz), PLL off ^{Note 4}	0.6	1	mA
	I _{DD4}	IDLE2 mode	f _{XX} = 5 MHz (f _X = 5 MHz), PLL off ^{Note 4}	0.28	0.5	mA
	I _{DD5}	Subclock operation mode	f _{XT} = 32.768 kHz, main clock stopped, internal oscillator stopped, PLL off REGOVL0 = 02H (low-voltage subclock operation mode)	18		μA
	I _{DD6}	Sub-IDLE mode	f _{XT} = 32.768 kHz, main clock stopped, internal oscillator stopped, PLL off REGOVL0 = 02H (low-voltage sub-IDLE mode)	2.5	50	μA
	I _{DD7}	STOP mode	Subclock stopped, internal oscillator stopped REGOVL0 = 01H (low-voltage STOP mode) T _A = 25°C	1.5	3.0	μA
			Subclock stopped, internal oscillator stopped REGOVL0 = 01H (low-voltage STOP mode) T _A = 85°C		45	μA
			Subclock operating, internal oscillator stopped REGOVL0 = 01H (low-voltage STOP mode)	2.5	50	μA
I _{DD8}	Self programming mode	f _{XX} = 20 MHz (f _X = 5 MHz)	14	24	mA	
LVI current	I _{LVI}			1.2	3	μA
WDT, internal oscillation current	I _{WDT}			5		μA

- Notes**
1. TYP. current is a value at V_{DD} = EV_{DD} = 3.3 V, T_A = 25°C.
The TYP. value is not a value guaranteed for each device.
 2. MAX. current is a value at which the characteristic in question is at the worst-case value at V_{DD} = EV_{DD} = 3.6 V, T_A = -40 to +85°C.
 3. Total of V_{DD} and EV_{DD} currents. Currents I_{LVI} and I_{WDT} flowing through the output buffers, A/D converter, D/A converter, and on-chip pull-down resistor are not included.
 4. TYP. value indicates the current value when watch timer + TMM (count by watch timer interrupt) operate as peripheral functions.
MAX. value indicates the current value when all the functions operable in a range in which the pin status is not changed operate as peripheral functions.
However, I_{LVI} and I_{WDT} are excluded.
 5. TYP. value of I_{DD1} is a value when all instructions are executed + RAM access 15%.

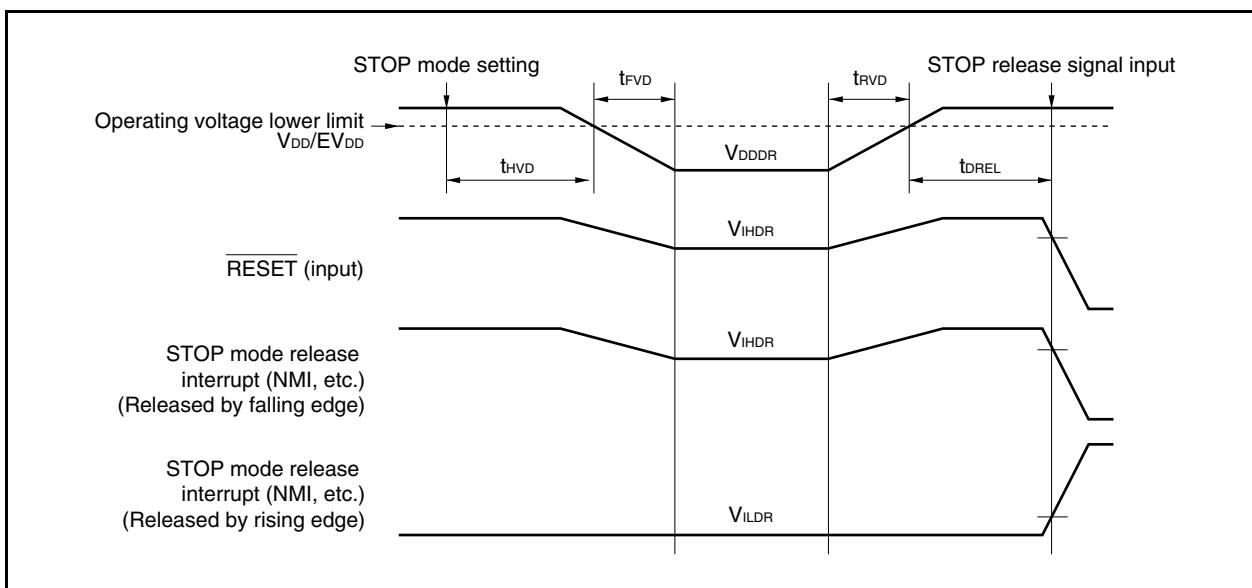
Remark For details about the operating voltage, see 31.3 Operating Conditions.

31.6.3 Data retention characteristics (in STOP mode)

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention voltage	V_{DDDR}	STOP mode (all functions stopped)	1.9		3.6	V
Data retention current	I_{DDDR}	Subclock stopped, internal oscillator stopped $T_A = 85^\circ\text{C}$			45	μA
Supply voltage rise time	t_{RVD}		200			μs
Supply voltage fall time	t_{FVD}		200			μs
Supply voltage retention time	t_{HVD}	After STOP mode setting	0			ms
STOP release signal input time	t_{DREL}	After V_{DD} reaches the operating voltage MIN. (see 31.3 Operating Conditions)	0			ms
Data retention input voltage, high	V_{IHDR}	$V_{DD} = EV_{DD} = V_{DDDR}$	$0.9V_{DDDR}$		V_{DDDR}	V
Data retention input voltage, low	V_{ILDR}	$V_{DD} = EV_{DD} = V_{DDDR}$	0		$0.1V_{DDDR}$	V

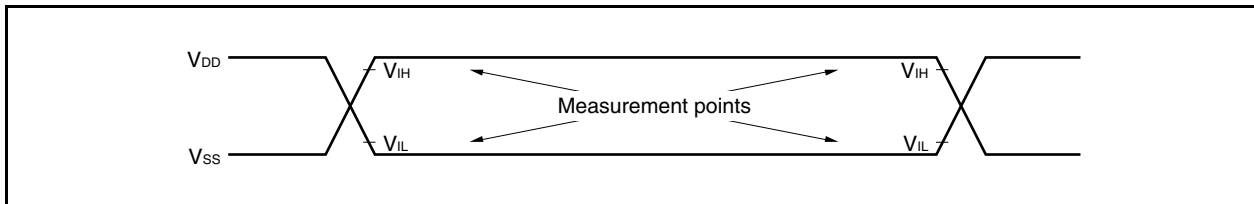
Caution Shifting to STOP mode and restoring from STOP mode must be performed within the rated operating range.



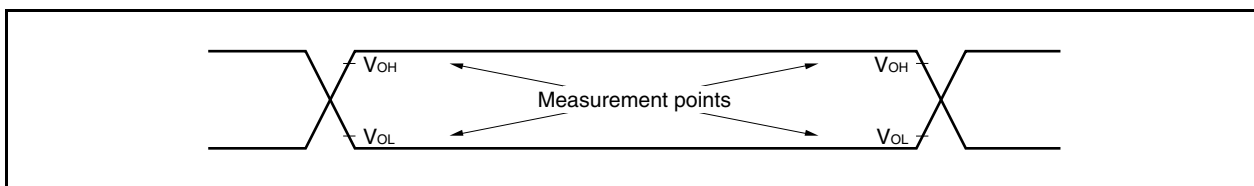
31.7 AC Characteristics

31.7.1 Measurement conditions

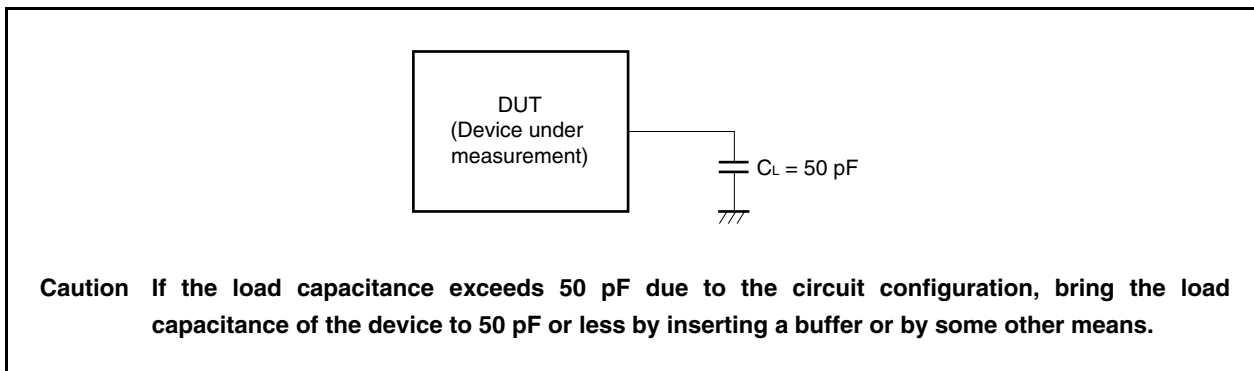
(1) AC test input measurement points



(2) AC test output measurement points



(3) Load conditions



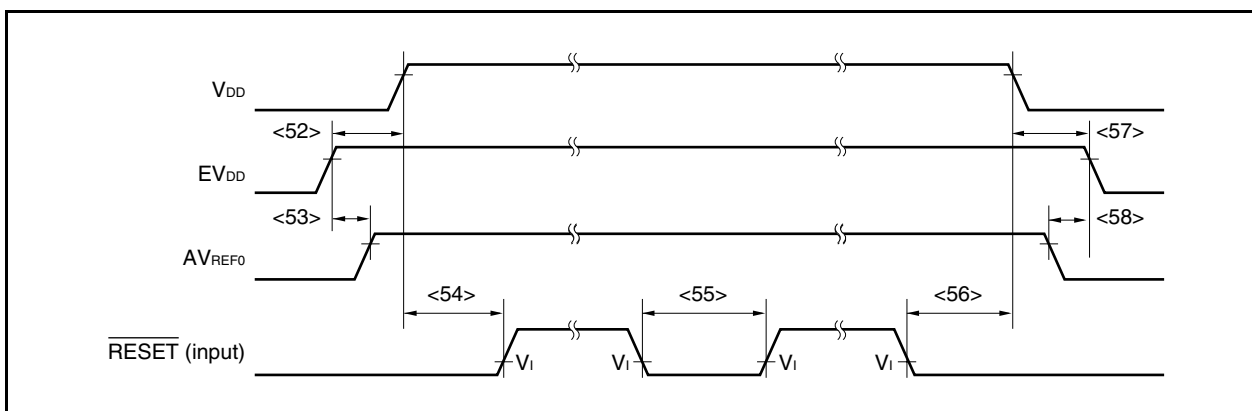
31.7.2 Power on/power off/reset timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
$EV_{DD}\uparrow \rightarrow V_{DD}\uparrow$	t_{REL}	<52>	0		ns
$EV_{DD}\uparrow \rightarrow AV_{REF0}, AV_{REF1}\uparrow$	t_{REA}	<53>	0	t_{REL}	ns
$V_{DD}\uparrow \rightarrow \overline{RESET}\uparrow$	t_{RER}	<54>	$500 + t_{REG}^{\text{Note}}$		ns
\overline{RESET} low-level width	t_{WRSL}	<55>	500		ns
$\overline{RESET}\downarrow \rightarrow V_{DD}\downarrow$	t_{FRE}	<56>	500		ns
$V_{DD}\downarrow \rightarrow EV_{DD}\downarrow$	t_{FEL}	<57>	0		ns
$AV_{REF0}\downarrow \rightarrow EV_{DD}\downarrow$	t_{FEA}	<58>	0	t_{FEL}	ns

Note See 31.5 Regulator Characteristics.

Remark The \overline{RESET} pin has an analog noise elimination function.



31.8 Peripheral Function Characteristics

31.8.1 Interrupt timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
NMI high-level width	t_{WNIH}		500		ns
NMI low-level width	t_{WNIL}		500		ns
INTPn ^{Note} high-level width	t_{WITH}	$n = 0, 2, 5$ to 7	500		ns
INTPn ^{Note} low-level width	t_{WITL}	$n = 0, 2, 5$ to 7	500		ns

Note The characteristics of INTPn is the same as the \overline{DRST} pin (P05/INTP2/ \overline{DRST}).

Remark The NMI and INTPn pins have the analog noise elimination function ($n = 0, 2, 5$ to 7).

31.8.2 Key return timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
KRn high-level width	t_{WKRH}		500		ns
KRn low-level width	t_{WKRL}		500		ns

Remarks 1. $n = 0$ to 7

2. The KRn pin has an analog noise elimination function.

31.8.3 Timer timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
TI high-level width	t_{TIH}	TIP00, TIP20, TIP21, TIP50, TIP51,	$2T + 20$		ns
TI low-level width	t_{TIL}	TIQ00 to TIQ03	$2T + 20$		ns

Remark $T = 1/f_{xx}$

31.8.4 UART timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Transmit rate		$V_{DD} = 2.2$ to 3.6 V		625	kbps
ASCK0 frequency		$V_{DD} = 2.2$ to 3.6 V		5	MHz
		$V_{DD} = 2.7$ to 3.6 V		10	MHz

31.8.5 CSIB timing

(1) Master mode

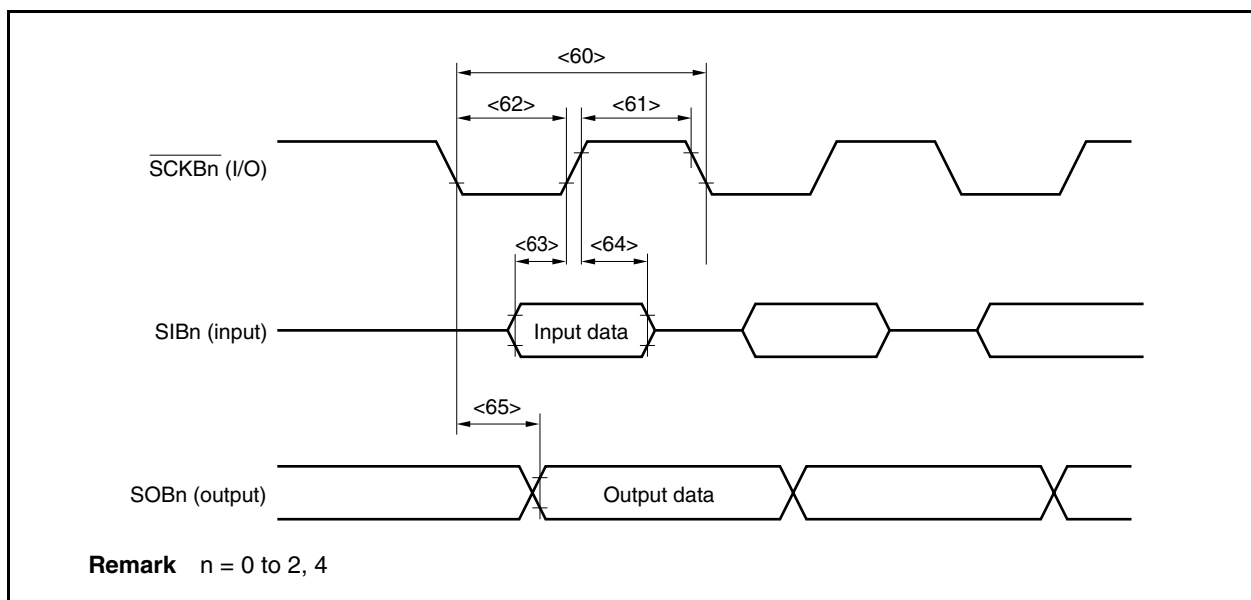
($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKBn}}$ cycle time	t_{KCY1}	<60> 2.7 V \leq $V_{DD} \leq$ 3.6 V	125		ns
			800		ns
$\overline{\text{SCKBn}}$ high-level width	t_{KH1}	<61> 2.7 V \leq $V_{DD} \leq$ 3.6 V	$t_{KCY1}/2 - 8$		ns
			$t_{KCY1}/2 - 80$		ns
$\overline{\text{SCKBn}}$ low-level width	t_{KL1}	<62> 2.7 V \leq $V_{DD} \leq$ 3.6 V	$t_{KCY1}/2 - 8$		ns
			$t_{KCY1}/2 - 80$		ns
SIBn setup time (to $\overline{\text{SCKBn}}\uparrow$)	t_{SIK1}	<63> 2.7 V \leq $V_{DD} \leq$ 3.6 V	27		ns
			100		ns
SIBn hold time (from $\overline{\text{SCKBn}}\uparrow$)	t_{KSH1}	<64> 2.7 V \leq $V_{DD} \leq$ 3.6 V	27		ns
			100		ns
Delay time from $\overline{\text{SCKBn}}\downarrow$ to SOBn output	t_{KSO1}	<65> 2.7 V \leq $V_{DD} \leq$ 3.6 V		27	ns
				95	ns

Remark n = 0 to 2, 4

(2) Slave mode**($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)**

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKBn}}$ cycle time	$t_{\text{KCY}2}$	$2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	125		ns
		$2.2\text{ V} \leq V_{DD} < 2.7\text{ V}$	800		ns
$\overline{\text{SCKBn}}$ high-level width	$t_{\text{KH}2}$	$2.2\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	54.5		ns
$\overline{\text{SCKBn}}$ low-level width	$t_{\text{KL}2}$	$2.2\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	54.5		ns
SIBn setup time (to $\overline{\text{SCKBn}}\uparrow$)	$t_{\text{SIK}2}$	$2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	27		ns
		$2.2\text{ V} \leq V_{DD} < 2.7\text{ V}$	100		ns
SIBn hold time (from $\overline{\text{SCKBn}}\uparrow$)	$t_{\text{KSI}2}$	$2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	27		ns
		$2.2\text{ V} \leq V_{DD} < 2.7\text{ V}$	100		ns
Delay time from $\overline{\text{SCKBn}}\downarrow$ to SOBn output	$t_{\text{KSO}2}$	$2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$		27	ns
		$2.2\text{ V} \leq V_{DD} < 2.7\text{ V}$		95	ns

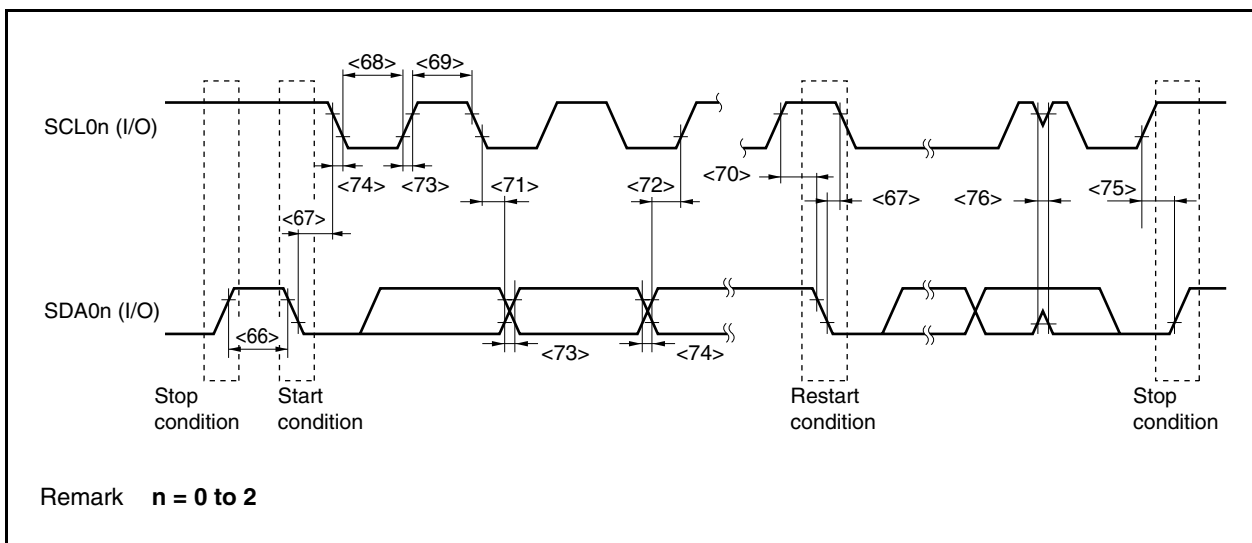
Remark n = 0 to 2, 4

31.8.6 I²C bus mode(T_A = -40 to +85°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.2 to 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol		Normal Mode		High-Speed Mode		Unit	
			MIN.	MAX.	MIN.	MAX.		
SCL0n clock frequency	f _{CLK}		0	100	0	400	kHz	
Bus free time (Between start and stop conditions)	t _{BUF}	<66>	4.7	–	1.3	–	μs	
Hold time ^{Note 1}	t _{HD:STA}	<67>	4.0	–	0.6	–	μs	
SCL0n clock low-level width	t _{LOW}	<68>	4.7	–	1.3	–	μs	
SCL0n clock high-level width	t _{HIGH}	<69>	4.0	–	0.6	–	μs	
Setup time for start/restart conditions	t _{SU:STA}	<70>	4.7	–	0.6	–	μs	
Data hold time	CBUS compatible master	t _{HD:DAT}	<71>	5.0	–	–	–	μs
	I ² C mode			0 ^{Note 2}	–	0 ^{Note 2}	0.9 ^{Note 3}	μs
Data setup time	t _{SU:DAT}	<72>	250	–	100 ^{Note 4}	–	ns	
SDA0n and SCL0n signal rise time	t _R	<73>	–	1000	20 + 0.1Cb ^{Note 5}	300	ns	
SDA0n and SCL0n signal fall time	t _F	<74>	–	300	20 + 0.1Cb ^{Note 5}	300	ns	
Stop condition setup time	t _{SU:STO}	<75>	4.0	–	0.6	–	μs	
Pulse width of spike suppressed by input filter	t _{SP}	<76>	–	–	0	50	ns	
Capacitance load of each bus line	Cb		–	400	–	400	pF	

- Notes**
- At the start condition, the first clock pulse is generated after the hold time.
 - The system requires a minimum of 300 ns hold time internally for the SDA0n signal (at V_{IHmin.} of SCL0n signal) in order to occupy the undefined area at the falling edge of SCL0n.
 - If the system does not extend the SCL0n signal low hold time (t_{LOW}), only the maximum data hold time (t_{HD:DAT}) needs to be satisfied.
 - The high-speed mode I²C bus can be used in the normal-mode I²C bus system. In this case, set the high-speed mode I²C bus so that it meets the following conditions.
 - If the system does not extend the SCL0n signal's low state hold time:
t_{SU:DAT} ≥ 250 ns
 - If the system extends the SCL0n signal's low state hold time:
Transmit the following data bit to the SDA0n line prior to the SCL0n line release (t_{Rmax.} + t_{SU:DAT} = 1,000 + 250 = 1,250 ns: Normal mode I²C bus specification).
 - Cb: Total capacitance of one bus line (unit: pF)

Remark n = 0 to 2

I²C Bus Timing

31.8.7 A/D converter

($T_A = -40 \text{ to } +85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $2.7 \text{ V} \leq AV_{REF0} = AV_{REF1} \leq 3.6 \text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0 \text{ V}$, $C_L = 50 \text{ pF}$)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution					10	bit
Overall error ^{Note}		$2.7 \text{ V} \leq AV_{REF0} \leq 3.6 \text{ V}$			± 0.6	%FSR
A/D conversion time	t_{CONV}	$3.0 \text{ V} \leq AV_{REF0} \leq 3.6 \text{ V}$	2.6		24	μs
		$2.7 \text{ V} \leq AV_{REF0} \leq 3.0 \text{ V}$	3.9		24	μs
Zero scale error					± 0.5	%FSR
Full scale error					± 0.5	%FSR
Non-linearity error					± 4.0	LSB
Differential linearity error					± 4.0	LSB
Analog input voltage	V_{IAN}		AV_{SS}		AV_{REF0}	V
Reference voltage	AV_{REF0}		2.7		3.6	V
AV_{REF0} current	AI_{REF0}	Normal conversion mode		3	6.5	mA
		High-speed conversion mode		4	10	mA
		When A/D converter unused			5	μA

Note Excluding quantization error (± 0.05 %FSR).

Caution Do not set (read/write) alternate-function ports during A/D conversion; otherwise the conversion resolution may be degraded.

Remark LSB: Least Significant Bit
FSR: Full Scale Range

31.8.8 D/A converter

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution					8	bit
Overall error		$R = 2$ M Ω			± 1.2	%FSR
Settling time		$C = 20$ pF			3	μs
Output resistor	R_o	Output data 55H		6.42		k Ω
Reference voltage	AV_{REF1}		2.7		3.6	V
AV_{REF1} current	AI_{REF1}	D/A conversion operating		1	2.5	mA
		D/A conversion stopped			5	μA

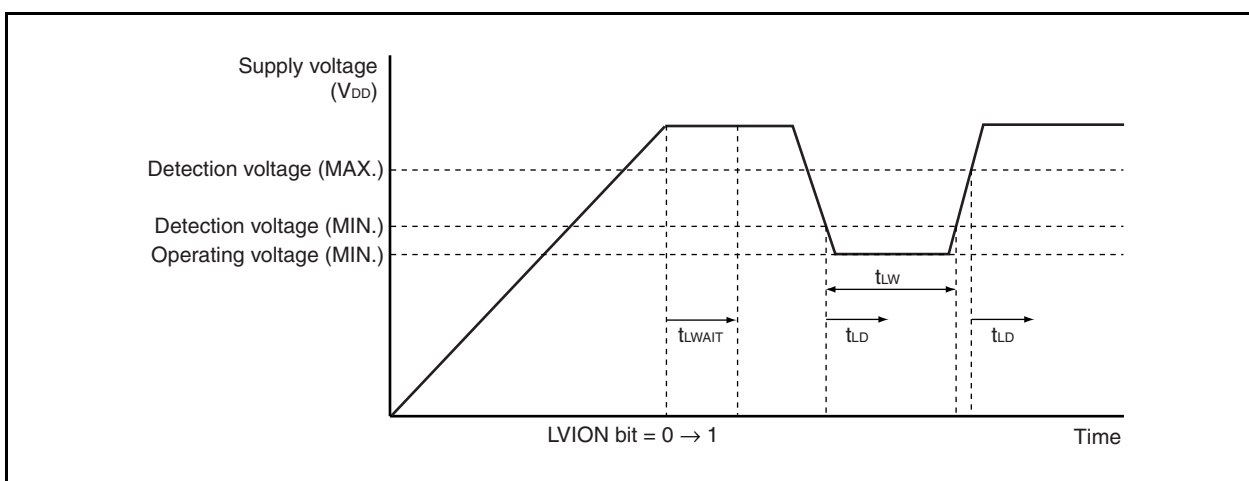
Remark R is the output pin load resistance and C is the output pin load capacitance.

31.8.9 LVI circuit characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LV10}		2.7	2.8	2.9	V
	V_{LV11}		2.2	2.3	2.4	V
Response time ^{Note}	t_{LD}	At rising edge: After V_{DD} reaches V_{LV10}/V_{LV11} (MAX.) At falling edge: After V_{DD} has dropped to V_{LV10}/V_{LV11} (MIN.)		0.2	2.0	ms
Minimum pulse width	t_{LW}	$V_{DD} = V_{LV10}/V_{LV11}$ (MIN.)	0.2			ms
Reference voltage stabilization wait time	t_{LWAIT}	After V_{DD} reaches V_{LV10}/V_{LV11} (MAX.)		0.1	0.2	ms

Note Time required to detect the detection voltage and output an interrupt or reset signal.



31.9 Flash Memory Programming Characteristics

(1) Basic characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.7$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

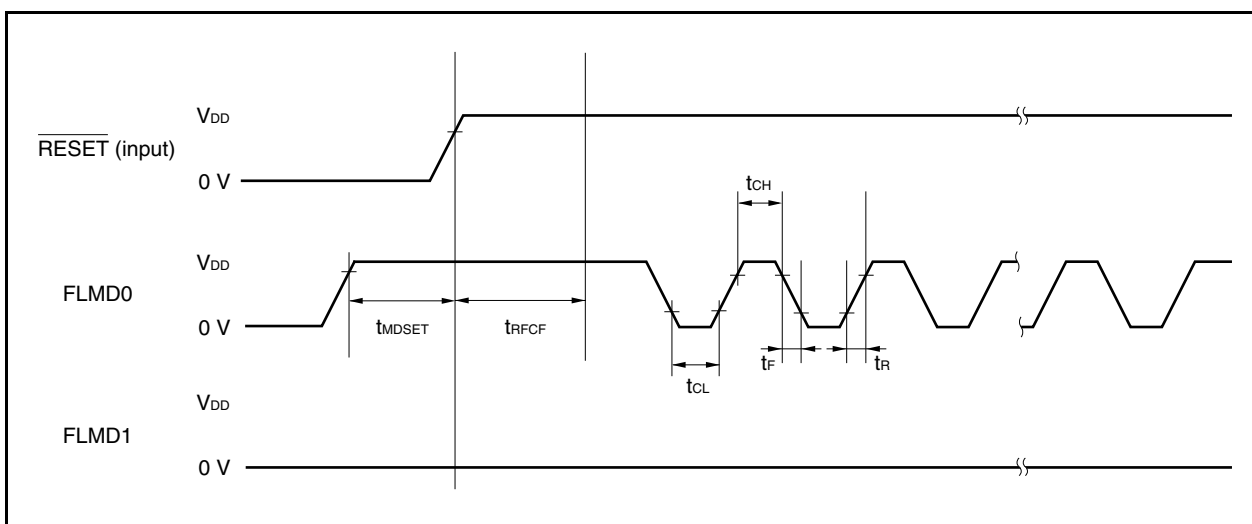
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Operating frequency	f_{CPU}		2.5		20	MHz	
Supply voltage	V_{DD}	$2.5 \text{ MHz} \leq f_{xx} \leq 20 \text{ MHz}$	2.7		3.6	V	
Number of rewrites	C_{WRT}	Used for updating programs When using flash memory programmer and Renesas Electronics self programming library	Retained for 15 years	1,000			times
		Used for updating data When using Renesas Electronics EEPROM emulation library (usable ROM size: 12 KB of 6 consecutive blocks, or 6 KB of 3 consecutive blocks)	Retained for 5 years	10,000			times
Programming temperature	t_{PRG}		-40		+85	$^\circ\text{C}$	

(2) Serial write operation characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.7$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
FLMD0, FLMD1 setup time	t_{MDSET}		2		3000	ms
FLMD0 count start time from $\overline{\text{RESET}}\uparrow$	t_{RFCF}	$f_x = 2.5$ to 10 MHz	800			μs
FLMD0 counter high-level width/ low-level width	t_{CH}/t_{CL}		10		100	μs
FLMD0 counter rise time/fall time	t_r/t_f				1	μs

Flash write mode setup timing



(3) Programming characteristics**(T_A = -40 to +85°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.7 to 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Chip erase time		f _{xx} = 20 MHz (when the chip erase command is executed)		105		ms
Write time per 256 bytes		f _{xx} = 20 MHz		2.0		ms
Block internal verify time		f _{xx} = 20 MHz		10		ms
Block blank check time		f _{xx} = 20 MHz		0.5		ms
Flash memory information setting time		f _{xx} = 20 MHz		30		ms

Remark Block size = 2 KB

CHAPTER 32 ELECTRICAL SPECIFICATIONS (V850ES/JE3-L) (Target)

32.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V _{DD}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 to +4.6	V
	EV _{DD}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 to +4.6	V
	AV _{REF0}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 to +4.6	V
	AV _{REF1}	V _{DD} = EV _{DD} = AV _{REF0} = AV _{REF1}	-0.5 to +4.6	V
	V _{SS}	V _{SS} = EV _{SS} = AV _{SS}	-0.5 to +0.5	V
	AV _{SS}	V _{SS} = EV _{SS} = AV _{SS}	-0.5 to +0.5	V
	EV _{SS}	V _{SS} = EV _{SS} = AV _{SS}	-0.5 to +0.5	V
Input voltage	V _{I1}	P97 to P915, PCM0, PDL5, $\overline{\text{RESET}}$, FLMD0	-0.5 to EV _{DD} + 0.5 ^{Note 1}	V
	V _{I2}	P10	-0.5 to AV _{REF1} + 0.5 ^{Note 1}	V
	V _{I3}	X1	-0.5 to V _{DD} + 0.5 ^{Note 1}	V
		X2	-0.5 to V _{RO} ^{Note 2} + 0.5 ^{Note 1}	
	V _{I4}	P02 to P06, P30 to P35, P38, P39, P40 to P42, P50 to P55, P90 to P94, P96	-0.5 to +6.0	V
	V _{I5}	XT1, XT2	-0.5 to V _{RO} ^{Note 2} + 0.5	V
Analog input voltage	V _{IAN}	P70 to P79	-0.5 to AV _{REF0} + 0.5 ^{Note 1}	V

- Notes**
1. Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.
 2. On-chip regulator output voltage

- Cautions**
1. Do not directly connect the output (or I/O) pins of IC products to each other, or to V_{DD}, V_{CC}, and GND. Open-drain pins or open-collector pins, however, can be directly connected to each other. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.
 2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.
The ratings and conditions indicated for DC characteristics, AC characteristics, and operating conditions represent the quality assurance range during normal operation.

Remark Unless specified otherwise, the ratings of alternate-function pins are the same as those of port pins.

Absolute Maximum Ratings (T_A = 25°C) (2/2)

Parameter	Symbol	Conditions	Ratings	Unit	
Output current, low	I _{OL}	P02 to P06, P30 to P35, P38, P39, P40 to P42, P50 to P55, P90 to P94, P96 to P915	Per pin	4	mA
			Total of all pins	50	mA
		PCM0, PDL5	Per pin	4	mA
			Total of all pins	8	mA
		P10		4	mA
		P70 to P79	Per pin	4	mA
			Total of all pins	20	mA
		Output current, high	I _{OH}	P02 to P06, P30 to P35, P38, P39, P40 to P42, P50 to P55, P90 to P94, P96 to P915	Per pin
Total of all pins	-50				mA
PCM0, PDL5	Per pin			-4	mA
	Total of all pins			-8	mA
P10				-4	mA
P70 to P79	Per pin			-4	mA
	Total of all pins			-20	mA
Operating ambient temperature	T _A			Normal operation mode	-40 to +85
		Flash memory programming mode	-40 to +85	°C	
Storage temperature	T _{stg}		-40 to +125	°C	

Cautions 1. Do not directly connect the output (or I/O) pins of IC products to each other, or to V_{DD}, V_{CC}, and GND. Open-drain pins or open-collector pins, however, can be directly connected to each other. Direct connection of the output pins between an IC product and an external circuit is possible, if the output pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.

2. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

The ratings and conditions indicated for DC characteristics, AC characteristics, and operating conditions represent the quality assurance range during normal operation.

Remark Unless specified otherwise, the ratings of alternate-function pins are the same as those of port pins.

32.2 Capacitance**Capacitance (T_A = 25°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = V_{SS} = EV_{SS} = AV_{SS} = 0 V)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
I/O capacitance	C _{IO}	f _x = 1 MHz Unmeasured pins returned to 0 V			10	pF

32.3 Operating Conditions

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

Internal System Clock	Conditions	Supply Voltage			Unit
		V_{DD}	EV_{DD}	AV_{REF0} , AV_{REF1}	
$f_{xx} = 2.5$ to 20 MHz	$C = 4.7\ \mu\text{F}$, A/D converter operating, D/A converter operating	2.7 to 3.6	2.7 to 3.6	2.7 to 3.6	V
$f_{xx} = 2.5$ to 5 MHz	$C = 4.7\ \mu\text{F}$, A/D converter stopped, D/A converter stopped	2.2 to 3.6	2.2 to 3.6	2.2 to 3.6	V
$f_{XT} = 32.768$ kHz	$C = 4.7\ \mu\text{F}$, A/D converter stopped, D/A converter stopped	2.2 to 3.6	2.2 to 3.6	2.2 to 3.6	V

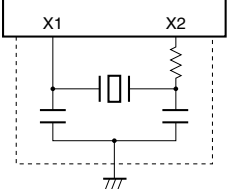
32.4 Oscillator Characteristics

32.4.1 Main clock oscillator characteristics

(1) oscillator characteristics

Main clock oscillator characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

Resonator	Circuit Example	Parameter	Conditions	MIN.	TYP.	MAX.	Unit		
Ceramic resonator/ Crystal resonator		Oscillation frequency (f_x) ^{Note 1}	Clock through mode	$V_{DD} = 2.2$ to 3.6 V	2.5		5	MHz	
				$V_{DD} = 2.7$ to 3.6 V	2.5		10	MHz	
				$V_{DD} = 2.7$ to 3.6 V in PLL mode	2.5		5	MHz	
				Oscillation stabilization time ^{Note 2}	$V_{DD} = 2.2$ to 3.6 V immediately after reset ends	Note 3	Note 4		s
				After STOP mode is released	$V_{DD} = 2.2$ to 3.6 V in clock through mode	1 ^{Note 5}	Note 6		ms
			$V_{DD} = 2.7$ to 3.6 V in PLL mode		1 ^{Note 7}	Note 6		ms	
		After IDLE2 mode is released	$V_{DD} = 2.2$ to 3.6 V in clock through mode	350 ^{Note 5}	Note 6		μs		
	$V_{DD} = 2.7$ to 3.6 V in PLL mode		800 ^{Note 7}	Note 6		μs			

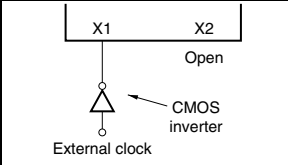
- Notes**
- The oscillation frequency shown above indicates only oscillator characteristics. Use the V850ES/JE3-L so that the internal operating conditions do not exceed the ratings shown in **AC Characteristics**, **DC Characteristics**, and **Operating Conditions**.
 - The wait time required from when the resonator starts oscillating until oscillation stabilizes.
 - The oscillation stabilization time after reset release is restricted in accordance with the set value of the option byte. For details, see **CHAPTER 27 OPTION BYTE**.
 - The oscillation stabilization time differs depending on the set value of the option byte. For details, see **CHAPTER 27 OPTION BYTE**.
 - Time required to set up the regulator and flash memory. Secure the setup time using the OSTS register.
 - The value varies depending on the setting of the OSTS register.
 - Time required to set up the regulator, flash memory, and PLL. Secure the setup time using the OSTS register.

Caution 1. When using the main clock oscillator, wire as follows in the area enclosed by the broken lines in the above figure to avoid an adverse effect from wiring capacitance.

- Keep the wiring length as short as possible.
- Do not cross the wiring with the other signal lines.
- Do not route the wiring near a signal line through which a high fluctuating current flows.
- Always make the ground point of the oscillator capacitor the same potential as V_{SS} .
- Do not ground the capacitor to a ground pattern through which a high current flows.
- Do not fetch signals from the oscillator.

Caution 2. When the main clock is stopped and the device is operating on the subclock, wait until the oscillation stabilization time has been secured by the program before switching back to the main clock.

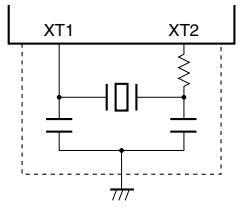
(2) External clock**($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.7$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)**

Resonator	Circuit Example	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
External clock		Input frequency (f_x) ^{Note}	Clock through mode	2.5		5	MHz
			PLL mode	2.5		5	MHz
		V_{IH}	X1	2.3		V_{DD}	V
		V_{IL}	X1	V_{SS}		0.4	V

Note Keep the duty factor of the input waveform to within 45% to 55%.

- Cautions**
1. Be sure to disconnect the internal feedback resistor after reset (set PCC.MFRC = 1).
 2. Leave the X2 pin open.
 3. Make sure that the CMOS inverter is as close to the X1 pin as possible.

32.4.2 Subclock oscillator characteristics**($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)**

Resonator	Circuit Example	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		Oscillation frequency (f_{XT}) ^{Note 1}		32	32.768	35	kHz
		Oscillation stabilization time ^{Note 2}				10	s

- Notes**
1. The oscillation frequency shown above indicates only oscillator characteristics. Use the V850ES/JE3-L so that the internal operation conditions do not exceed the ratings shown in **AC Characteristics**, **DC Characteristics**, and operating conditions.
 2. Time required from when V_{DD} reaches the oscillation voltage range (2.2 V (MIN.)) to when the crystal resonator stabilizes.

Cautions 1. When using the subclock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- Keep the wiring length as short as possible.
 - Do not cross the wiring with the other signal lines.
 - Do not route the wiring near a signal line through which a high fluctuating current flows.
 - Always make the ground point of the oscillator capacitor the same potential as V_{SS} .
 - Do not ground the capacitor to a ground pattern through which a high current flows.
 - Do not fetch signals from the oscillator.
2. The subclock oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the main clock oscillator. Particular care is therefore required with the wiring method when the subclock is used.
 3. For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

32.4.3 PLL characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input frequency	f_x		2.5		5	MHz
Output frequency	f_{xx}		10		20	MHz
Lock time	t_{PLL}	After V_{DD} reaches 2.7 V (MIN.)			800	μs

32.4.4 Internal oscillator characteristics

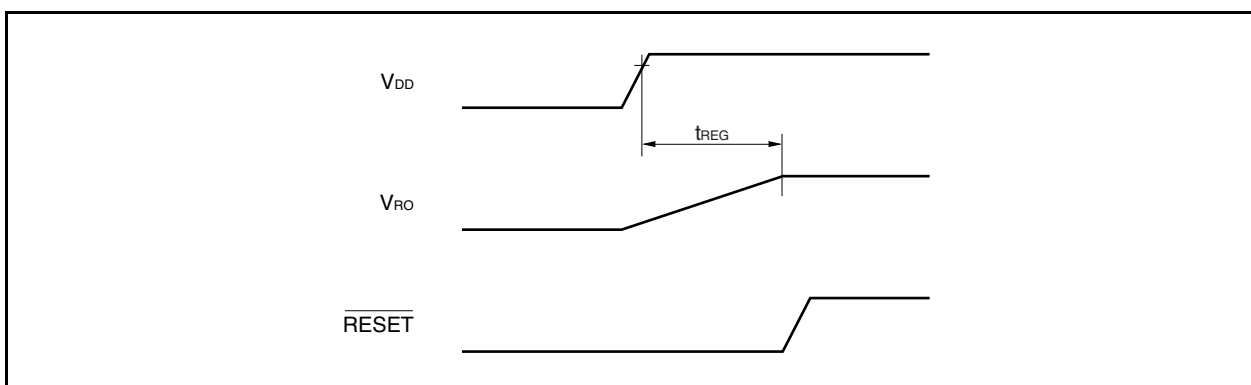
($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output frequency	f_R		100	220	400	kHz

32.5 Regulator Characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage	V_{DD}	$f_{xx} = 20$ MHz (MAX.)	2.7		3.6	V
		$f_{xx} = 5$ MHz (MAX.)	2.2		3.6	V
		Data retained (STOP mode)	1.9		3.6	V
Output voltage	V_{RO}	$V_{DD} = 2.7$ to 3.6 V		2.5		V
Regulator output stabilization time	t_{REG}	After V_{DD} reaches 2.7 V (MIN.), stabilization capacitance $C = 4.7$ μF (recommended value) connected to REGC pin			1	ms
		After V_{DD} reaches 2.2 V (MIN.), stabilization capacitance $C = 4.7$ μF (recommended value) connected to REGC pin			3.5	ms
External capacitance	REGC	Permissible error of external capacitance to be connected to REGC pin	3.76	4.70	5.64	μF



32.6 DC Characteristics

32.6.1 Pin characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$) (1/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	$\overline{\text{RESET}}$, FLMD0, P97 to P915	$0.8EV_{DD}$		EV_{DD}	V
	V_{IH2}	P02 to P06, P30 to P35, P42, P50 to P55, P92 to P94, P96	$0.8EV_{DD}$		5.5	V
	V_{IH3}	P38, P39, P40, P41, P90, P91	$0.7EV_{DD}$		5.5	V
	V_{IH4}	PCM0, PDL5	$0.7EV_{DD}$		EV_{DD}	V
	V_{IH5}	P70 to P79	$0.7AV_{REF0}$		AV_{REF0}	V
	V_{IH6}	P10	$0.7AV_{REF1}$		AV_{REF1}	V
Input voltage, low	V_{IL1}	$\overline{\text{RESET}}$, FLMD0, P97 to P915	EV_{SS}		$0.2EV_{DD}$	V
	V_{IL2}	P02 to P06, P30 to P35, P42, P50 to P55, P92 to P94, P96	EV_{SS}		$0.2EV_{DD}$	V
	V_{IL3}	P38, P39, P40, P41, P90, P91	EV_{SS}		$0.3EV_{DD}$	V
	V_{IL4}	PCM0, PDL5	EV_{SS}		$0.3EV_{DD}$	V
	V_{IL5}	P70 to P79	AV_{SS}		$0.3AV_{REF0}$	V
	V_{IL6}	P10	AV_{SS}		$0.3AV_{REF1}$	V
Input leakage current, high	I_{LH}	$V_I = V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$			5	μA
Input leakage current, low	I_{LL}	$V_I = 0\text{ V}$			-5	μA
Output leakage current, high	I_{LOH}	$V_O = V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$			5	μA
Output leakage current, low	I_{LOL}	$V_O = 0\text{ V}$			-5	μA

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

(T_A = -40 to +85°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V) (2/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit	
Output voltage, high	V _{OH1}	P02 to P06, P30 to P35, P38, P39, P40 to P42, P50 to P55, P90 to P94, P96 to P915	Per pin I _{OH} = -1.0 mA	Total of all pins -20 mA	EV _{DD} - 1.0		EV _{DD}	V
			Per pin I _{OH} = -100 μA	Total of all pins -3.7 mA	EV _{DD} - 0.5		EV _{DD}	V
	V _{OH2}	PCM0, PDL5	Per pin I _{OH} = -1.0 mA	Total of all pins -2 mA	EV _{DD} - 1.0		EV _{DD}	V
			Per pin I _{OH} = -100 μA	Total of all pins -0.2 mA	EV _{DD} - 0.5		EV _{DD}	V
	V _{OH3}	P70 to P79	Per pin I _{OH} = -0.4 mA	Total of all pins -4.0 mA	AV _{REF0} - 1.0		AV _{REF0}	V
			Per pin I _{OH} = -100 μA	Total of all pins -1.0 mA	AV _{REF0} - 0.5		AV _{REF0}	V
	V _{OH4}	P10	I _{OH} = -0.4 mA		AV _{REF1} - 1.0		AV _{REF1}	V
			I _{OH} = -100 μA		AV _{REF1} - 0.5		AV _{REF1}	V
Output voltage, low	V _{OL1}	P02 to P06, P30 to P35, P42, P50 to P55, P92 to P94, P96 to P915	Per pin I _{OL} = 1.0 mA	Total of all pins 20 mA	0		0.4	V
			Per pin I _{OL} = 3.0 mA		0		0.4	V
	V _{OL3}	PCM0, PDL5	Per pin I _{OL} = 1.0 mA	Total of all pins 2 mA	0		0.4	V
	V _{OL4}	P10, P70 to P79	Per pin I _{OL} = 0.4 mA	Total of all pins 4.4 mA	0		0.4	V
Software pull-down resistor ^{Note}	R ₁	P05	V _I = V _{DD}		10	20	100	kΩ

Note $\overline{\text{DRST}}$ pin only (controlled by OCDM register)

Remarks 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

2. When the I_{OH} and I_{OL} conditions are not satisfied for a pin but the total value of all pins is satisfied, only that pin does not satisfy the DC characteristics.

32.6.2 Supply current characteristics

(T_A = -40 to +85°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP. ^{Note 1}	MAX. ^{Note 2}	Unit
Supply current ^{Note 3}	I _{DD1}	Normal operation	f _{xx} = 20 MHz (f _x = 5 MHz) ^{Note 4}	12 ^{Note 5}	20	mA
			f _{xx} = 10 MHz (f _x = 10 MHz), PLL off ^{Note 4}	6 ^{Note 5}	10	mA
	I _{DD2}	HALT mode	f _{xx} = 20 MHz (f _x = 5 MHz) ^{Note 4}	7.5	14	mA
	I _{DD3}	IDLE1 mode	f _{xx} = 5 MHz (f _x = 5 MHz), PLL off ^{Note 4}	0.6	1	mA
	I _{DD4}	IDLE2 mode	f _{xx} = 5 MHz (f _x = 5 MHz), PLL off ^{Note 4}	0.28	0.5	mA
	I _{DD5}	Subclock operation mode	f _{XT} = 32.768 kHz, main clock stopped, internal oscillator stopped, PLL off REGOVL0 = 02H (low-voltage subclock operation mode)	18		μA
	I _{DD6}	Sub-IDLE mode	f _{XT} = 32.768 kHz, main clock stopped, internal oscillator stopped, PLL off REGOVL0 = 02H (low-voltage sub-IDLE mode)	2.5	50	μA
	I _{DD7}	STOP mode	Subclock stopped, internal oscillator stopped REGOVL0 = 01H (low-voltage STOP mode) T _A = 25°C	1.5	3.0	μA
			Subclock stopped, internal oscillator stopped REGOVL0 = 01H (low-voltage STOP mode) T _A = 85°C		45	μA
			Subclock operating, internal oscillator stopped REGOVL0 = 01H (low-voltage STOP mode)	2.5	50	μA
I _{DD8}	Self programming mode	f _{xx} = 20 MHz (f _x = 5 MHz)	14	24	mA	
LVI current	I _{LVI}			1.2	3	μA
WDT, internal oscillation current	I _{WDT}			5		μA

- Notes**
1. TYP. current is a value at V_{DD} = EV_{DD} = 3.3 V, T_A = 25°C.
The TYP. value is not a value guaranteed for each device.
 2. MAX. current is a value at which the characteristic in question is at the worst-case value at V_{DD} = EV_{DD} = 3.6 V, T_A = -40 to +85°C.
 3. Total of V_{DD} and EV_{DD} currents. Currents I_{LVI} and I_{WDT} flowing through the output buffers, A/D converter, D/A converter, and on-chip pull-down resistor are not included.
 4. TYP. value indicates the current value when watch timer + TMM (count by watch timer interrupt) operate as peripheral functions.
MAX. value indicates the current value when all the functions operable in a range in which the pin status is not changed operate as peripheral functions.
However, I_{LVI} and I_{WDT} are excluded.
 5. TYP. value of I_{DD1} is a value when all instructions are executed + RAM access 15%.

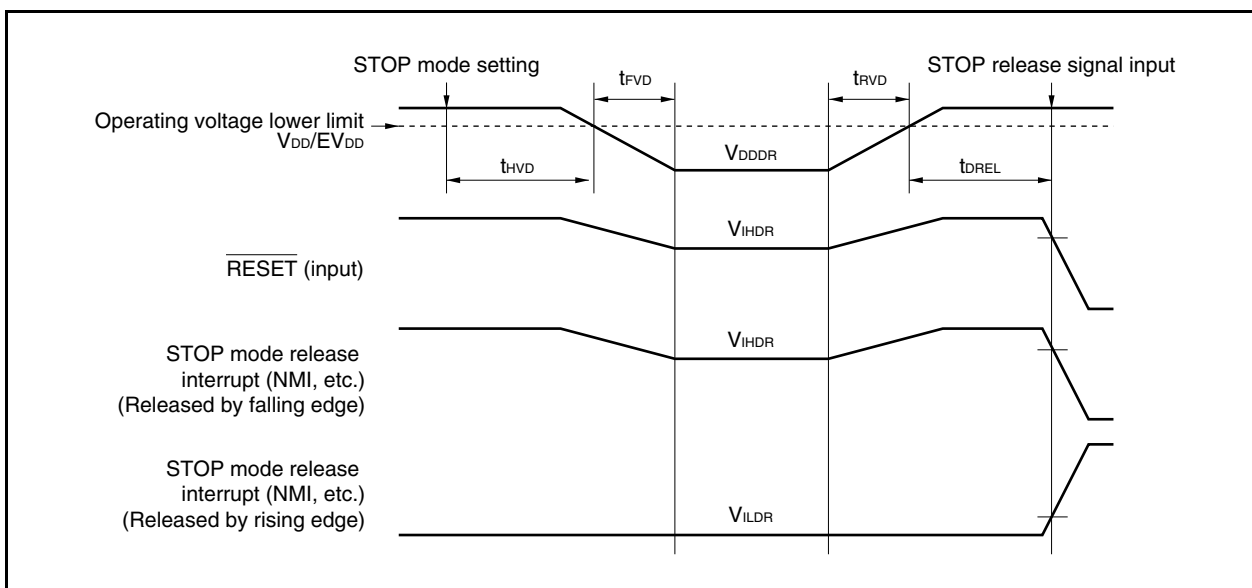
Remark For details about the operating voltage, see 32.3 Operating Conditions.

32.6.3 Data retention characteristics (in STOP mode)

(T_A = -40 to +85°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention voltage	V _{DDDR}	STOP mode (all functions stopped)	1.9		3.6	V
Data retention current	I _{DDDR}	Subclock stopped, internal oscillator stopped T _A = 85°C			45	μA
Supply voltage rise time	t _{RVD}		200			μs
Supply voltage fall time	t _{FVD}		200			μs
Supply voltage retention time	t _{HVD}	After STOP mode setting	0			ms
STOP release signal input time	t _{DREL}	After V _{DD} reaches the operating voltage MIN. (see 32.3 Operating Conditions)	0			ms
Data retention input voltage, high	V _{IHDR}	V _{DD} = EV _{DD} = V _{DDDR}	0.9V _{DDDR}		V _{DDDR}	V
Data retention input voltage, low	V _{ILDR}	V _{DD} = EV _{DD} = V _{DDDR}	0		0.1V _{DDDR}	V

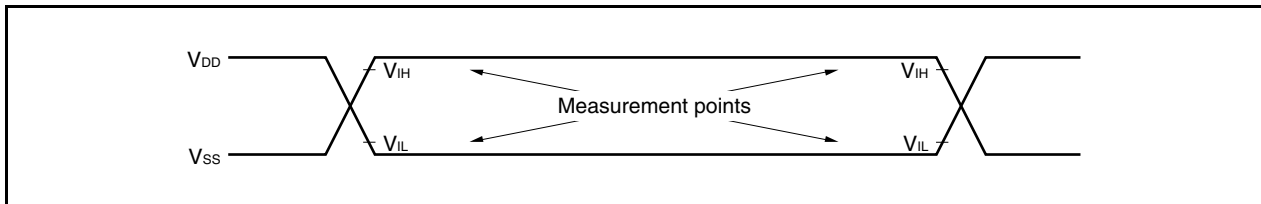
Caution Shifting to STOP mode and restoring from STOP mode must be performed within the rated operating range.



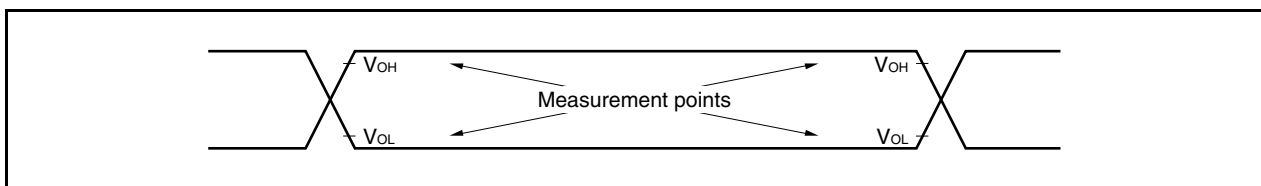
32.7 AC Characteristics

32.7.1 Measurement conditions

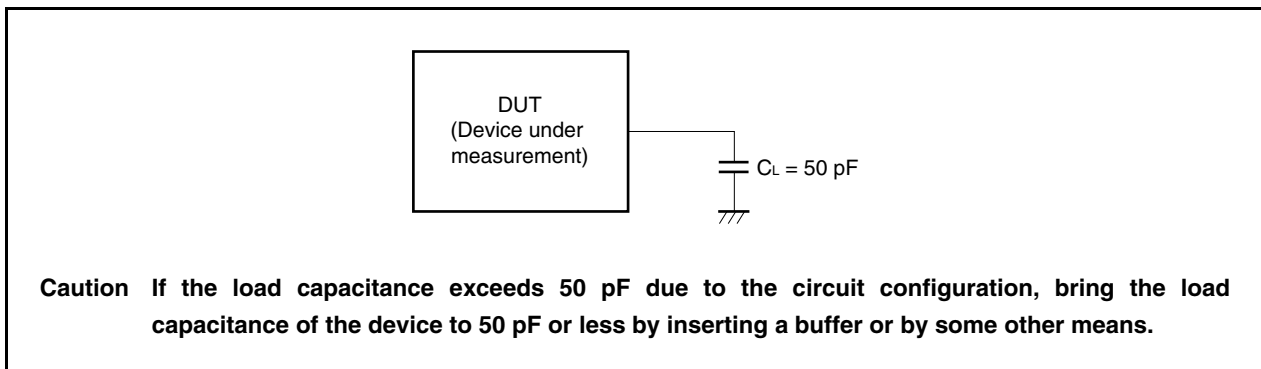
(1) AC test input measurement points



(2) AC test output measurement points



(3) Load conditions



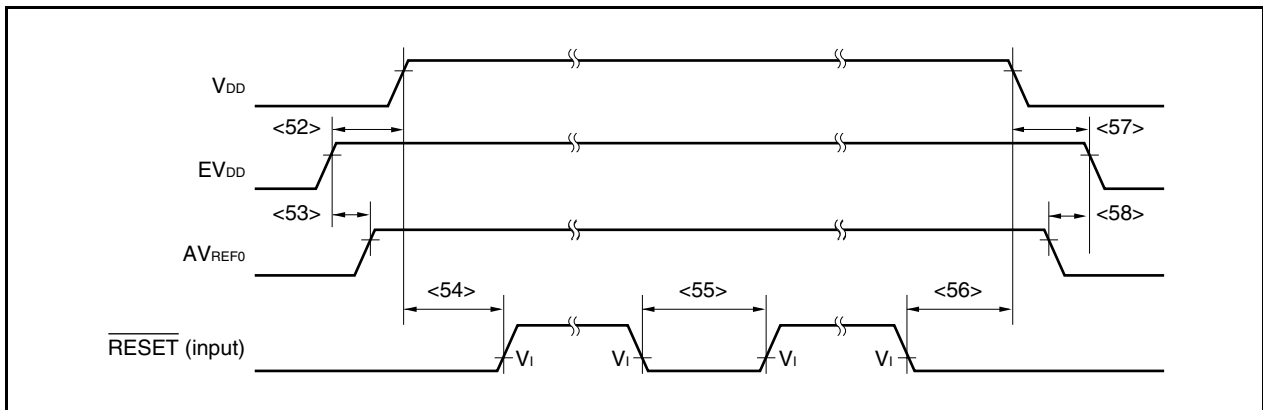
32.7.2 Power on/power off/reset timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
$EV_{DD}\uparrow \rightarrow V_{DD}\uparrow$	t_{REL}	<52>	0		ns
$EV_{DD}\uparrow \rightarrow AV_{REF0}, AV_{REF1}\uparrow$	t_{REA}	<53>	0	t_{REL}	ns
$V_{DD}\uparrow \rightarrow \overline{RESET}\uparrow$	t_{RER}	<54>	$500 + t_{REG}^{Note}$		ns
\overline{RESET} low-level width	t_{WRSL}	<55>	500		ns
$\overline{RESET}\downarrow \rightarrow V_{DD}\downarrow$	t_{FRE}	<56>	500		ns
$V_{DD}\downarrow \rightarrow EV_{DD}\downarrow$	t_{FEL}	<57>	0		ns
$AV_{REF0}\downarrow \rightarrow EV_{DD}\downarrow$	t_{FEA}	<58>	0	t_{FEL}	ns

Note See 32.5 Regulator Characteristics.

Remark The \overline{RESET} pin has an analog noise elimination function.



32.8 Peripheral Function Characteristics

32.8.1 Interrupt timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
NMI high-level width	t_{WNIH}		500		ns
NMI low-level width	t_{WNIL}		500		ns
INTPn ^{Note} high-level width	t_{WITH}	n = 0 to 7 (Analog noise elimination)	500		ns
		n = 3 (Digital noise elimination)	$3T_{SMP} + 20$		ns
INTPn ^{Note} low-level width	t_{WITL}	n = 0 to 7 (Analog noise elimination)	500		ns
		n = 3 (Digital noise elimination)	$3T_{SMP} + 20$		ns

Note The characteristics of INTPn is the same as the \overline{DRST} pin (P05/INTP2/ \overline{DRST}).

Remarks 1. T_{SMP} : Noise elimination sampling clock cycle

2. The NMI and INTPn pins have the analog noise elimination function (n = 0 to 7).

32.8.2 Key return timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
KRn high-level width	t_{WKRH}		500		ns
KRn low-level width	t_{WKRL}		500		ns

Remarks 1. n = 0 to 7

2. The KRn pin has an analog noise elimination function.

32.8.3 Timer timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.2$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
TI high-level width	t_{TIH}	TIP00, TIP01, TIP10, TIP11, TIP20, TIP21,	$2T + 20$		ns
TI low-level width	t_{TIL}	TIP31, TIP40, TIP41, TIP50, TIP51, TIQ00 to TIQ03	$2T + 20$		ns

Remark $T = 1/f_{xx}$

32.8.4 UART timing

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Transmit rate		$V_{DD} = 2.2$ to 3.6 V		625	kbps
ASCK0 frequency		$V_{DD} = 2.2$ to 3.6 V		5	MHz
		$V_{DD} = 2.7$ to 3.6 V		10	MHz

32.8.5 CSIB timing

(1) Master mode

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKBn}}$ cycle time	t_{KCY1}	$\langle 60 \rangle$ 2.7 V $\leq V_{DD} \leq 3.6$ V	125		ns
		2.2 V $\leq V_{DD} < 2.7$ V	800		ns
$\overline{\text{SCKBn}}$ high-level width	t_{KH1}	$\langle 61 \rangle$ 2.7 V $\leq V_{DD} \leq 3.6$ V	$t_{KCY1}/2 - 8$		ns
		2.2 V $\leq V_{DD} < 2.7$ V	$t_{KCY1}/2 - 80$		ns
$\overline{\text{SCKBn}}$ low-level width	t_{KL1}	$\langle 62 \rangle$ 2.7 V $\leq V_{DD} \leq 3.6$ V	$t_{KCY1}/2 - 8$		ns
		2.2 V $\leq V_{DD} < 2.7$ V	$t_{KCY1}/2 - 80$		ns
SIBn setup time (to $\overline{\text{SCKBn}}\uparrow$)	t_{SIK1}	$\langle 63 \rangle$ 2.7 V $\leq V_{DD} \leq 3.6$ V	27		ns
		2.2 V $\leq V_{DD} < 2.7$ V	100		ns
SIBn hold time (from $\overline{\text{SCKBn}}\uparrow$)	t_{KSH1}	$\langle 64 \rangle$ 2.7 V $\leq V_{DD} \leq 3.6$ V	27		ns
		2.2 V $\leq V_{DD} < 2.7$ V	100		ns
Delay time from $\overline{\text{SCKBn}}\downarrow$ to SOBn output	t_{KSO1}	$\langle 65 \rangle$ 2.7 V $\leq V_{DD} \leq 3.6$ V		27	ns
		2.2 V $\leq V_{DD} < 2.7$ V		95	ns

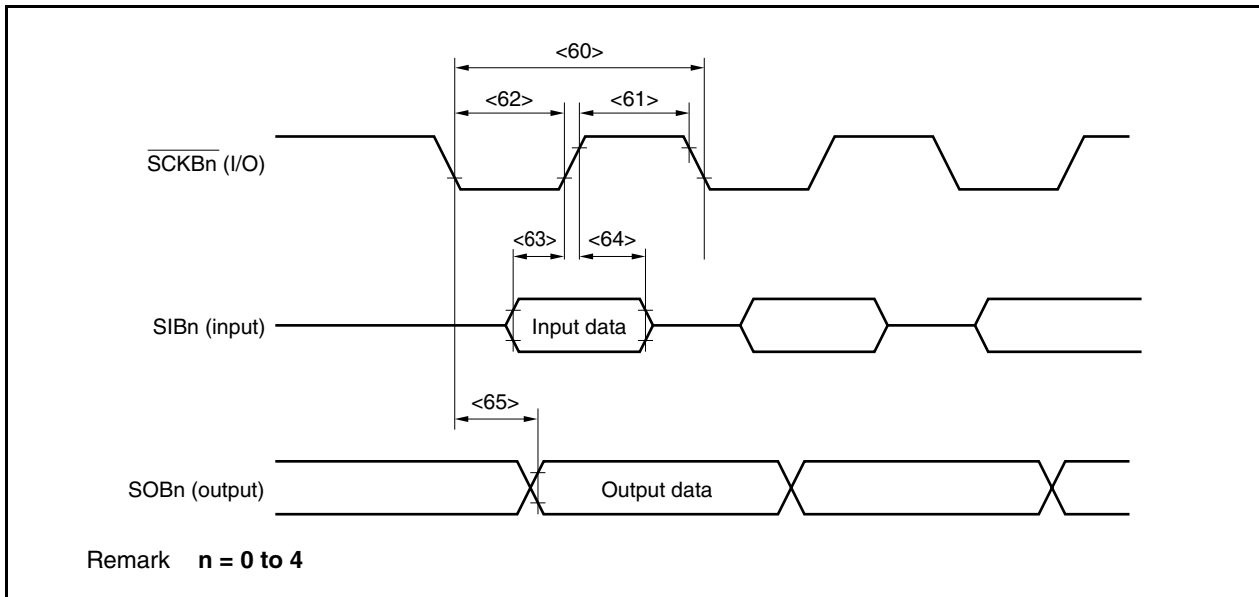
Remark n = 0 to 4

(2) Slave mode

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
$\overline{\text{SCKBn}}$ cycle time	$t_{\text{CY}2}$	$2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	125		ns
		$2.2\text{ V} \leq V_{DD} < 2.7\text{ V}$	800		ns
$\overline{\text{SCKBn}}$ high-level width	$t_{\text{KH}2}$	$2.2\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	54.5		ns
$\overline{\text{SCKBn}}$ low-level width	$t_{\text{KL}2}$	$2.2\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	54.5		ns
SIBn setup time (to $\overline{\text{SCKBn}}\uparrow$)	$t_{\text{SIK}2}$	$2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	27		ns
		$2.2\text{ V} \leq V_{DD} < 2.7\text{ V}$	100		ns
SIBn hold time (from $\overline{\text{SCKBn}}\uparrow$)	$t_{\text{SH}2}$	$2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$	27		ns
		$2.2\text{ V} \leq V_{DD} < 2.7\text{ V}$	100		ns
Delay time from $\overline{\text{SCKBn}}\downarrow$ to SOBn output	$t_{\text{KS}02}$	$2.7\text{ V} \leq V_{DD} \leq 3.6\text{ V}$		27	ns
		$2.2\text{ V} \leq V_{DD} < 2.7\text{ V}$		95	ns

Remark n = 0 to 4

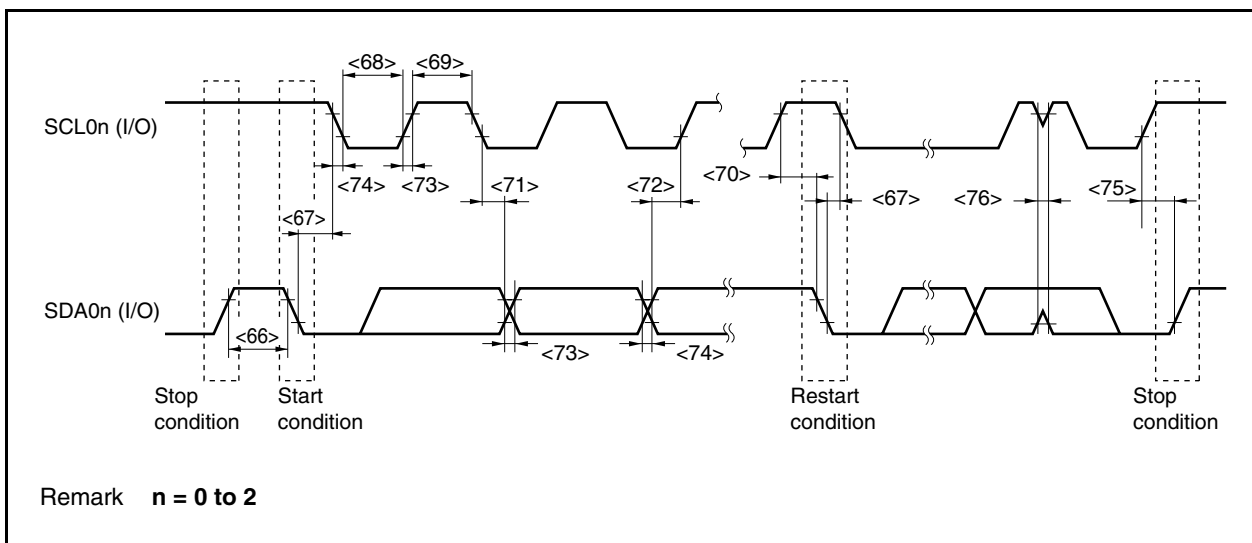


32.8.6 I²C bus mode(T_A = -40 to +85°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.2 to 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol		Normal Mode		High-Speed Mode		Unit	
			MIN.	MAX.	MIN.	MAX.		
SCL0n clock frequency	f _{CLK}		0	100	0	400	kHz	
Bus free time (Between start and stop conditions)	t _{BUF}	<66>	4.7	–	1.3	–	μs	
Hold time ^{Note 1}	t _{HD:STA}	<67>	4.0	–	0.6	–	μs	
SCL0n clock low-level width	t _{LOW}	<68>	4.7	–	1.3	–	μs	
SCL0n clock high-level width	t _{HIGH}	<69>	4.0	–	0.6	–	μs	
Setup time for start/restart conditions	t _{SU:STA}	<70>	4.7	–	0.6	–	μs	
Data hold time	CBUS compatible master	t _{HD:DAT}	<71>	5.0	–	–	–	μs
	I ² C mode			0 ^{Note 2}	–	0 ^{Note 2}	0.9 ^{Note 3}	μs
Data setup time	t _{SU:DAT}	<72>	250	–	100 ^{Note 4}	–	ns	
SDA0n and SCL0n signal rise time	t _R	<73>	–	1000	20 + 0.1Cb ^{Note 5}	300	ns	
SDA0n and SCL0n signal fall time	t _F	<74>	–	300	20 + 0.1Cb ^{Note 5}	300	ns	
Stop condition setup time	t _{SU:STO}	<75>	4.0	–	0.6	–	μs	
Pulse width of spike suppressed by input filter	t _{SP}	<76>	–	–	0	50	ns	
Capacitance load of each bus line	Cb		–	400	–	400	pF	

- Notes**
- At the start condition, the first clock pulse is generated after the hold time.
 - The system requires a minimum of 300 ns hold time internally for the SDA0n signal (at V_{IHmin.} of SCL0n signal) in order to occupy the undefined area at the falling edge of SCL0n.
 - If the system does not extend the SCL0n signal low hold time (t_{LOW}), only the maximum data hold time (t_{HD:DAT}) needs to be satisfied.
 - The high-speed mode I²C bus can be used in the normal-mode I²C bus system. In this case, set the high-speed mode I²C bus so that it meets the following conditions.
 - If the system does not extend the SCL0n signal's low state hold time:
t_{SU:DAT} ≥ 250 ns
 - If the system extends the SCL0n signal's low state hold time:
Transmit the following data bit to the SDA0n line prior to the SCL0n line release (t_{Rmax.} + t_{SU:DAT} = 1,000 + 250 = 1,250 ns: Normal mode I²C bus specification).
 - Cb: Total capacitance of one bus line (unit: pF)

Remark n = 0 to 2

I²C Bus Timing

32.8.7 A/D converter

($T_A = -40 \text{ to } +85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $2.7 \text{ V} \leq AV_{REF0} = AV_{REF1} \leq 3.6 \text{ V}$, $V_{SS} = EV_{SS} = AV_{SS} = 0 \text{ V}$, $C_L = 50 \text{ pF}$)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution					10	bit
Overall error ^{Note}		$2.7 \text{ V} \leq AV_{REF0} \leq 3.6 \text{ V}$			± 0.6	%FSR
A/D conversion time	t_{CONV}	$3.0 \text{ V} \leq AV_{REF0} \leq 3.6 \text{ V}$	2.6		24	μs
		$2.7 \text{ V} \leq AV_{REF0} \leq 3.0 \text{ V}$	3.9		24	μs
Zero scale error					± 0.5	%FSR
Full scale error					± 0.5	%FSR
Non-linearity error					± 4.0	LSB
Differential linearity error					± 4.0	LSB
Analog input voltage	V_{IAN}		AV_{SS}		AV_{REF0}	V
Reference voltage	AV_{REF0}		2.7		3.6	V
AV_{REF0} current	AI_{REF0}	Normal conversion mode		3	6.5	mA
		High-speed conversion mode		4	10	mA
		When A/D converter unused			5	μA

Note Excluding quantization error (± 0.05 %FSR).

Caution Do not set (read/write) alternate-function ports during A/D conversion; otherwise the conversion resolution may be degraded.

Remark LSB: Least Significant Bit
FSR: Full Scale Range

32.8.8 D/A converter

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1}$, $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution					8	bit
Overall error		$R = 2\text{ M}\Omega$			± 1.2	%FSR
Settling time		$C = 20\text{ pF}$			3	μs
Output resistor	R_o	Output data 55H		6.42		$\text{k}\Omega$
Reference voltage	AV_{REF1}		2.7		3.6	V
AV_{REF1} current	AI_{REF1}	D/A conversion operating		1	2.5	mA
		D/A conversion stopped			5	μA

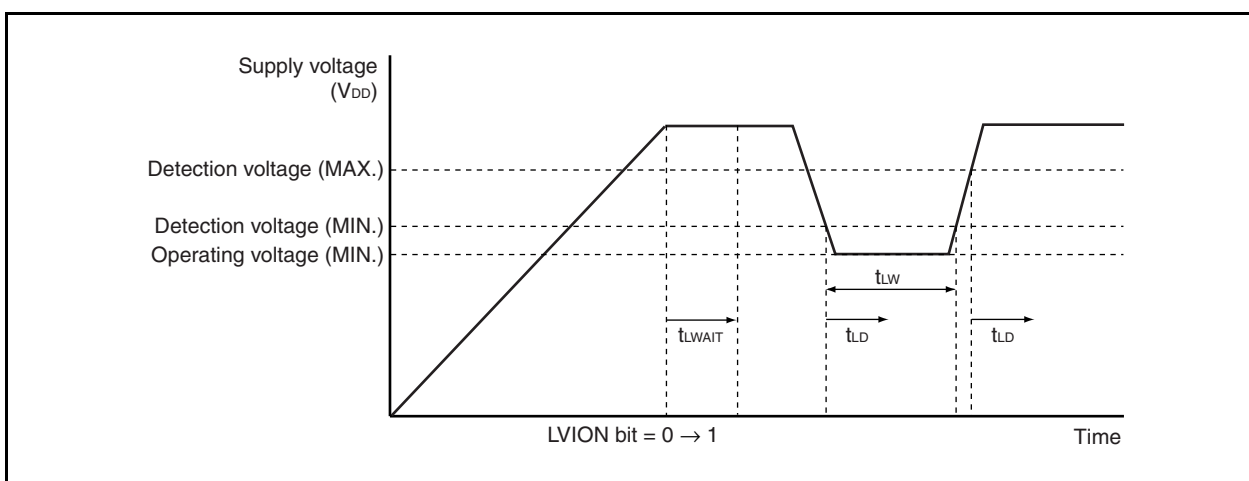
Remark R is the output pin load resistance and C is the output pin load capacitance.

32.8.9 LVI circuit characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.2$ to 3.6 V , $V_{SS} = EV_{SS} = AV_{SS} = 0\text{ V}$, $C_L = 50\text{ pF}$)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LV10}		2.7	2.8	2.9	V
	V_{LV11}		2.2	2.3	2.4	V
Response time ^{Note}	t_{LD}	At rising edge: After V_{DD} reaches V_{LV10}/V_{LV11} (MAX.) At falling edge: After V_{DD} has dropped to V_{LV10}/V_{LV11} (MIN.)		0.2	2.0	ms
Minimum pulse width	t_{LW}	$V_{DD} = V_{LV10}/V_{LV11}$ (MIN.)	0.2			ms
Reference voltage stabilization wait time	t_{LWAIT}	After V_{DD} reaches V_{LV10}/V_{LV11} (MAX.)		0.1	0.2	ms

Note Time required to detect the detection voltage and output an interrupt or reset signal.



32.9 Flash Memory Programming Characteristics

(1) Basic characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.7$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

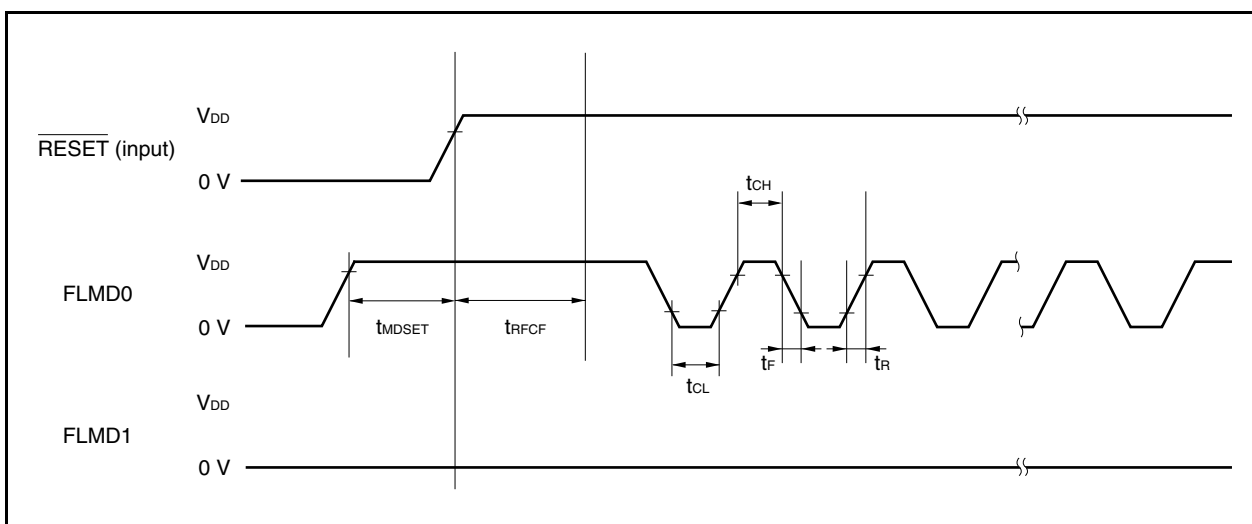
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit	
Operating frequency	f_{CPU}		2.5		20	MHz	
Supply voltage	V_{DD}	$2.5 \text{ MHz} \leq f_{xx} \leq 20 \text{ MHz}$	2.7		3.6	V	
Number of rewrites	C_{WRT}	Used for updating programs When using flash memory programmer and Renesas Electronics self programming library	Retained for 15 years	1,000			times
		Used for updating data When using Renesas Electronics EEPROM emulation library (usable ROM size: 12 KB of 6 consecutive blocks, or 6 KB of 3 consecutive blocks)	Retained for 5 years	10,000			times
Programming temperature	t_{PRG}		-40		+85	$^\circ\text{C}$	

(2) Serial write operation characteristics

($T_A = -40$ to $+85^\circ\text{C}$, $V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.7$ to 3.6 V, $V_{SS} = EV_{SS} = AV_{SS} = 0$ V, $C_L = 50$ pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
FLMD0, FLMD1 setup time	t_{MDSET}		2		3000	ms
FLMD0 count start time from $\overline{\text{RESET}}\uparrow$	t_{RFCF}	$f_x = 2.5$ to 10 MHz	800			μs
FLMD0 counter high-level width/ low-level width	t_{CH}/t_{CL}		10		100	μs
FLMD0 counter rise time/fall time	t_r/t_f				1	μs

Flash write mode setup timing



(3) Programming characteristics**(T_A = -40 to +85°C, V_{DD} = EV_{DD} = AV_{REF0} = AV_{REF1} = 2.7 to 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)**

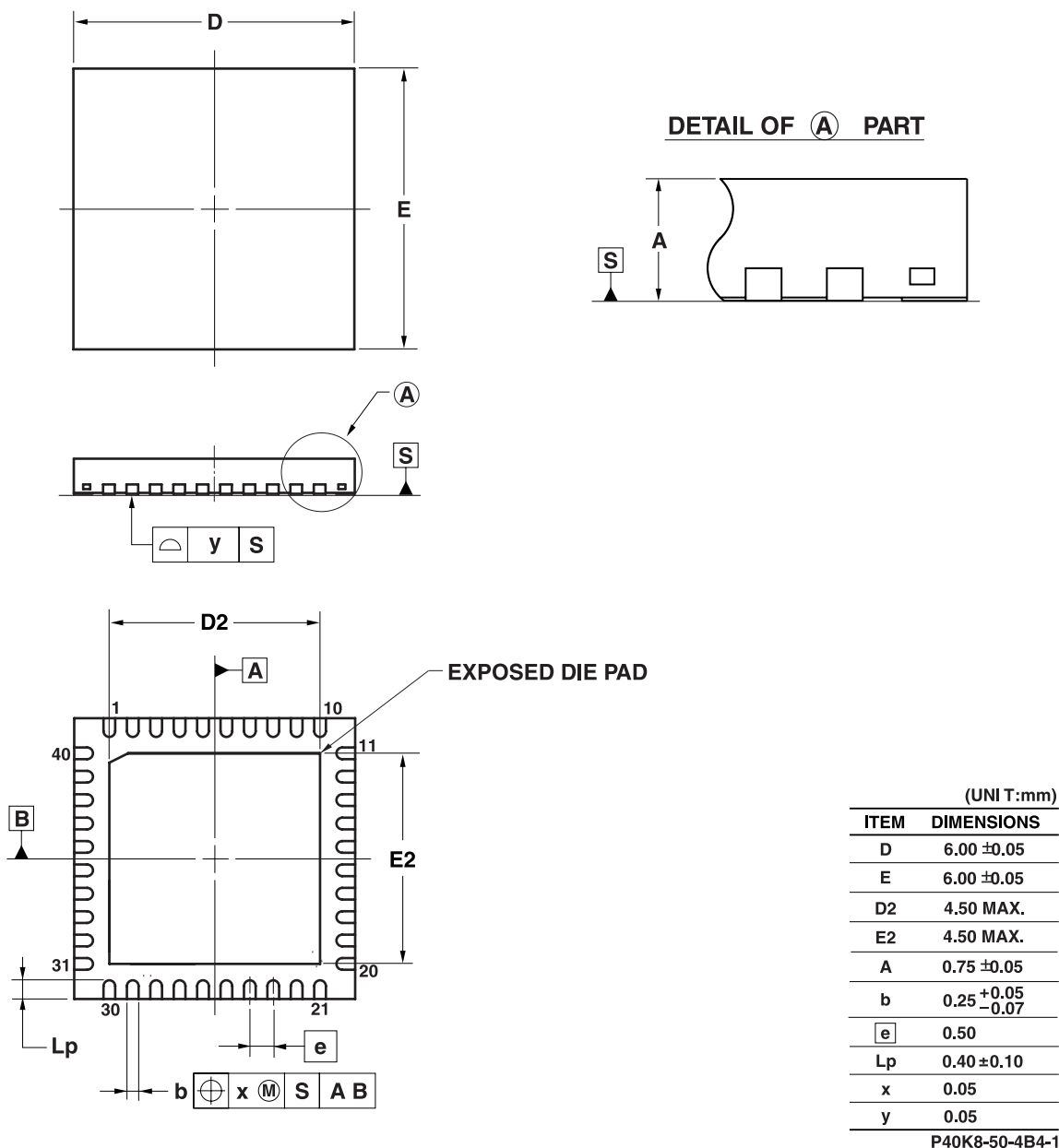
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Chip erase time		f _{xx} = 20 MHz (when the chip erase command is executed)		105		ms
Write time per 256 bytes		f _{xx} = 20 MHz		2.0		ms
Block internal verify time		f _{xx} = 20 MHz		10		ms
Block blank check time		f _{xx} = 20 MHz		0.5		ms
Flash memory information setting time		f _{xx} = 20 MHz		30		ms

Remark Block size = 2 KB

CHAPTER 33 PACKAGE DRAWINGS

• V850ES/JC3-L

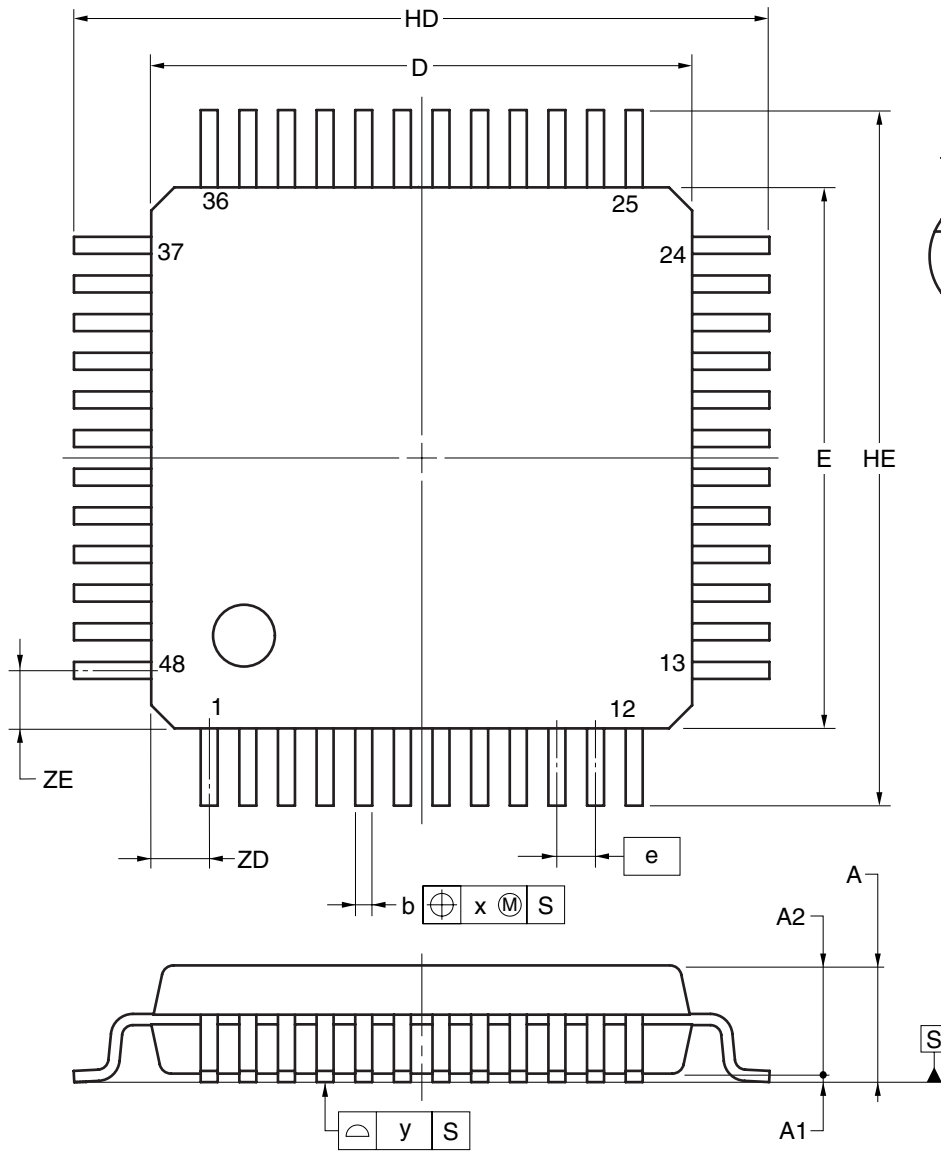
40-PIN PLASTIC WQFN(6x6)



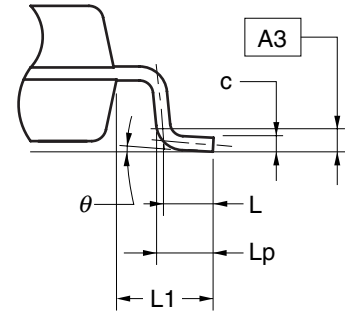
NOTE
D2 and E2 value varies depending on die size.

• V850ES/JC3-L

48-PIN PLASTIC LQFP (FINE PITCH) (7x7)



detail of lead end



(UNIT:mm)

ITEM	DIMENSIONS
D	7.00±0.20
E	7.00±0.20
HD	9.00±0.20
HE	9.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	0.20 ^{+0.07} _{-0.03}
c	0.125 ^{+0.075} _{-0.025}
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	3° ^{+5°} _{-3°}
e	0.50
x	0.08
y	0.08
ZD	0.75
ZE	0.75

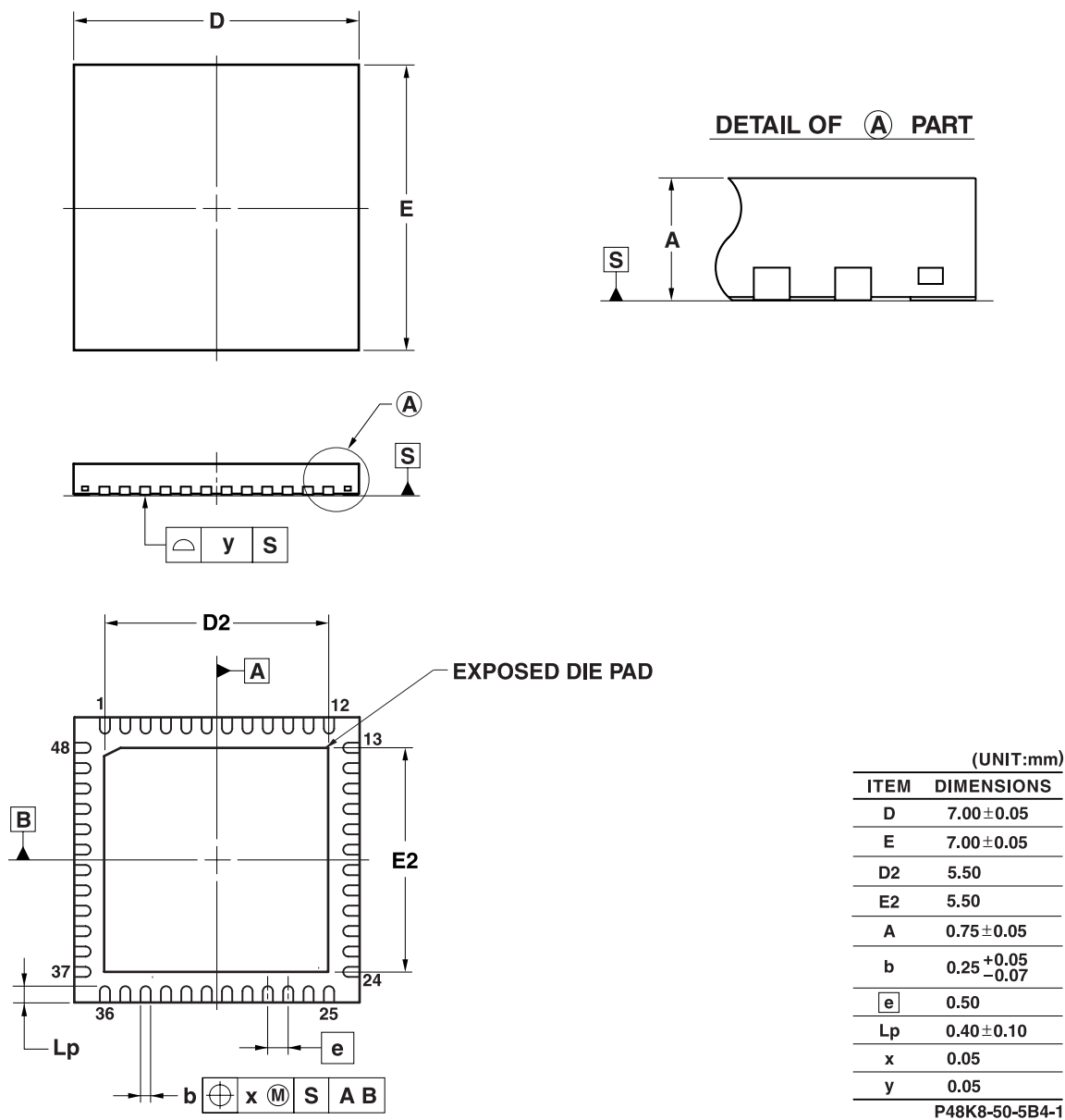
P48GA-50-GAM

NOTE

Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

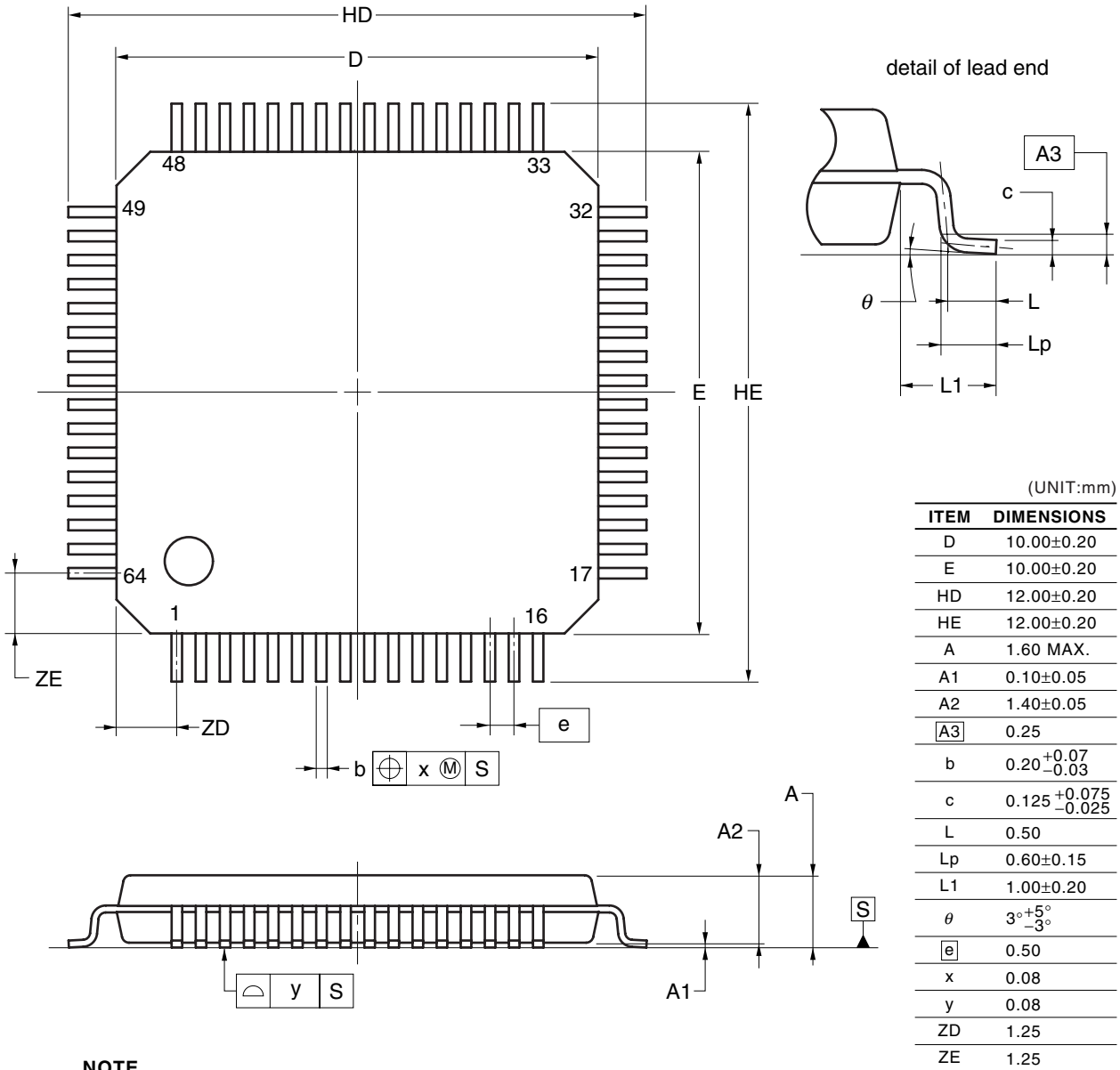
• V850ES/JC3-L

48-PIN PLASTIC WQFN(7x7)



• V850ES/JE3-L

64-PIN PLASTIC LQFP(FINE PITCH)(10x10)



NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

P64GB-50-GAH

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the V850ES/JC3-L, V850ES/JE3-L.

Figure A-1 shows the development tool configuration.

- **Support for PC98-NX series**

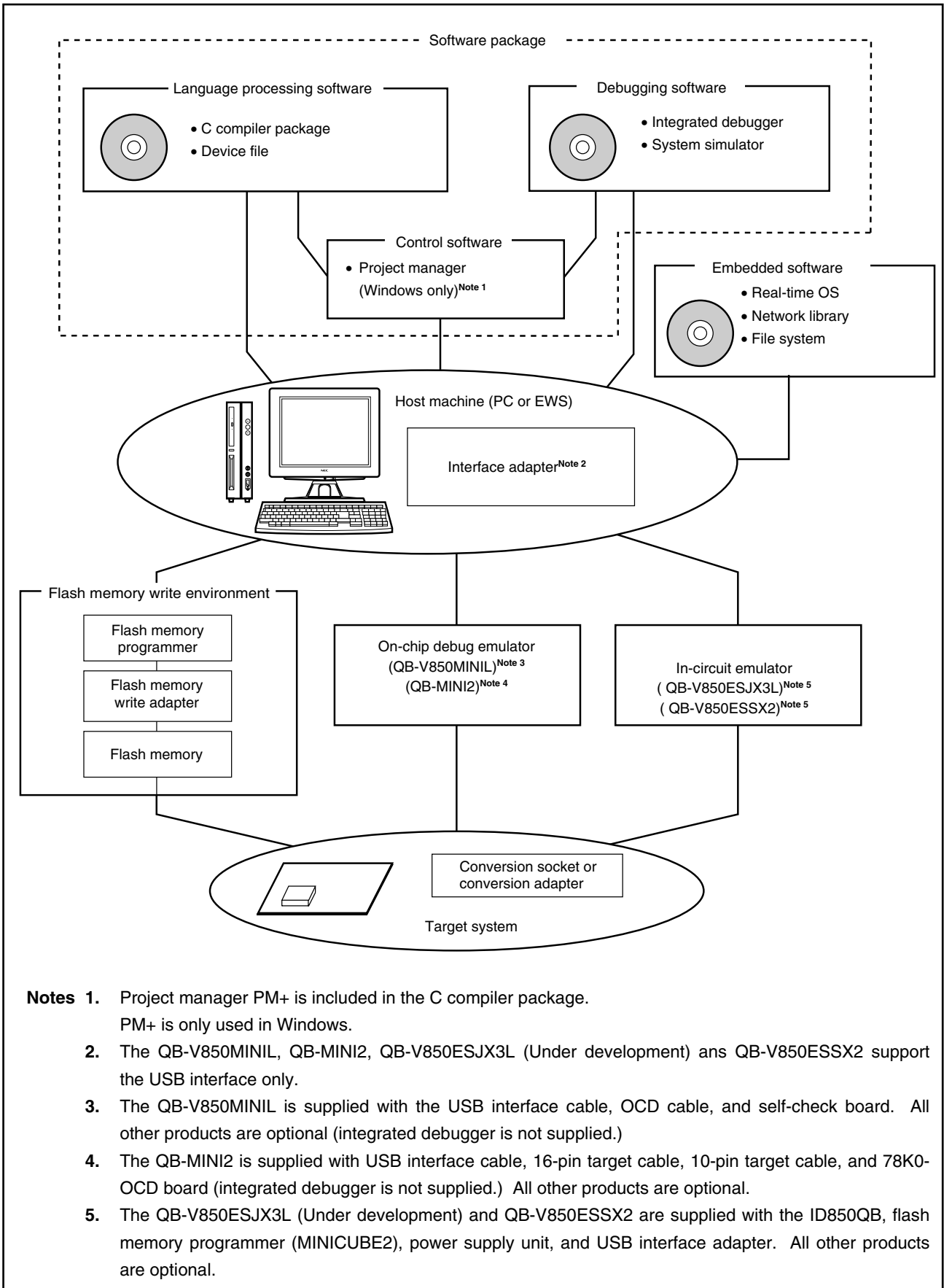
Unless otherwise specified, products supported by IBM PC/AT™ compatibles are compatible with PC98-NX series computers. When using PC98-NX series computers, refer to the explanation for IBM PC/AT compatibles.

- **Windows®**

Unless otherwise specified, "Windows" means the following OSs.

- Windows 98, 2000
- Windows Me
- Windows XP
- Windows NT® Ver. 4.0

Figure A-1. Development Tool Configuration



- Notes**
- Project manager PM+ is included in the C compiler package. PM+ is only used in Windows.
 - The QB-V850MINIL, QB-MINI2, QB-V850ESJX3L (Under development) and QB-V850ESSX2 support the USB interface only.
 - The QB-V850MINIL is supplied with the USB interface cable, OCD cable, and self-check board. All other products are optional (integrated debugger is not supplied.)
 - The QB-MINI2 is supplied with USB interface cable, 16-pin target cable, 10-pin target cable, and 78K0-OCD board (integrated debugger is not supplied.) All other products are optional.
 - The QB-V850ESJX3L (Under development) and QB-V850ESSX2 are supplied with the ID850QB, flash memory programmer (MINICUBE2), power supply unit, and USB interface adapter. All other products are optional.

A.1 Software Package

SP850 Software package for V850 microcontrollers	Development tools (software) commonly used with V850 microcontrollers are included this package.
	Part number: $\mu S_{xxxx}SP850$

Remark xxxx in the part number differs depending on the host machine and OS used.

$\mu S_{xxxx}SP850$

xxxx	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

A.2 Language Processing Software

CA850 C compiler package	This compiler converts programs written in C into object codes executable with a microcontroller. This compiler is started from project manager PM+.
	Part number: $\mu S_{xxxx}CA703000$
DF703738 Device file	This file contains information peculiar to the device. This device file should be used in combination with a tool (CA850 or ID850QB). The corresponding OS and host machine differ depending on the tool to be used.

Remark xxxx in the part number differs depending on the host machine and OS used.

$\mu S_{xxxx}CA703000$

xxxx	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	
3K17	SPARCstation™	SunOS™ (Rel. 4.1.4), Solaris™ (Rel. 2.5.1)	

A.3 Control Software

PM+ Project manager	This is control software designed to enable efficient user program development in the Windows environment. All operations used in development of a user program, such as starting the editor, building, and starting the debugger, can be performed from PM+. <Caution> PM+ is included in C compiler package CA850. It can only be used in Windows.
------------------------	--

A.4 Debugging Tools (Hardware)

A.4.1 When using IECUBE® QB-V850ESJX3L, QB-V850ESSX2

The system configuration when connecting the QB-V850ESJX3L, QB-V850ESSX2 to the host machine (PC-9821 series, PC/AT compatible) is shown below. Even if optional products are not prepared, connection is possible.

Figure A-2. System Configuration (When Using QB-V850ESJX3L, QB-V850ESSX2) (1/2)

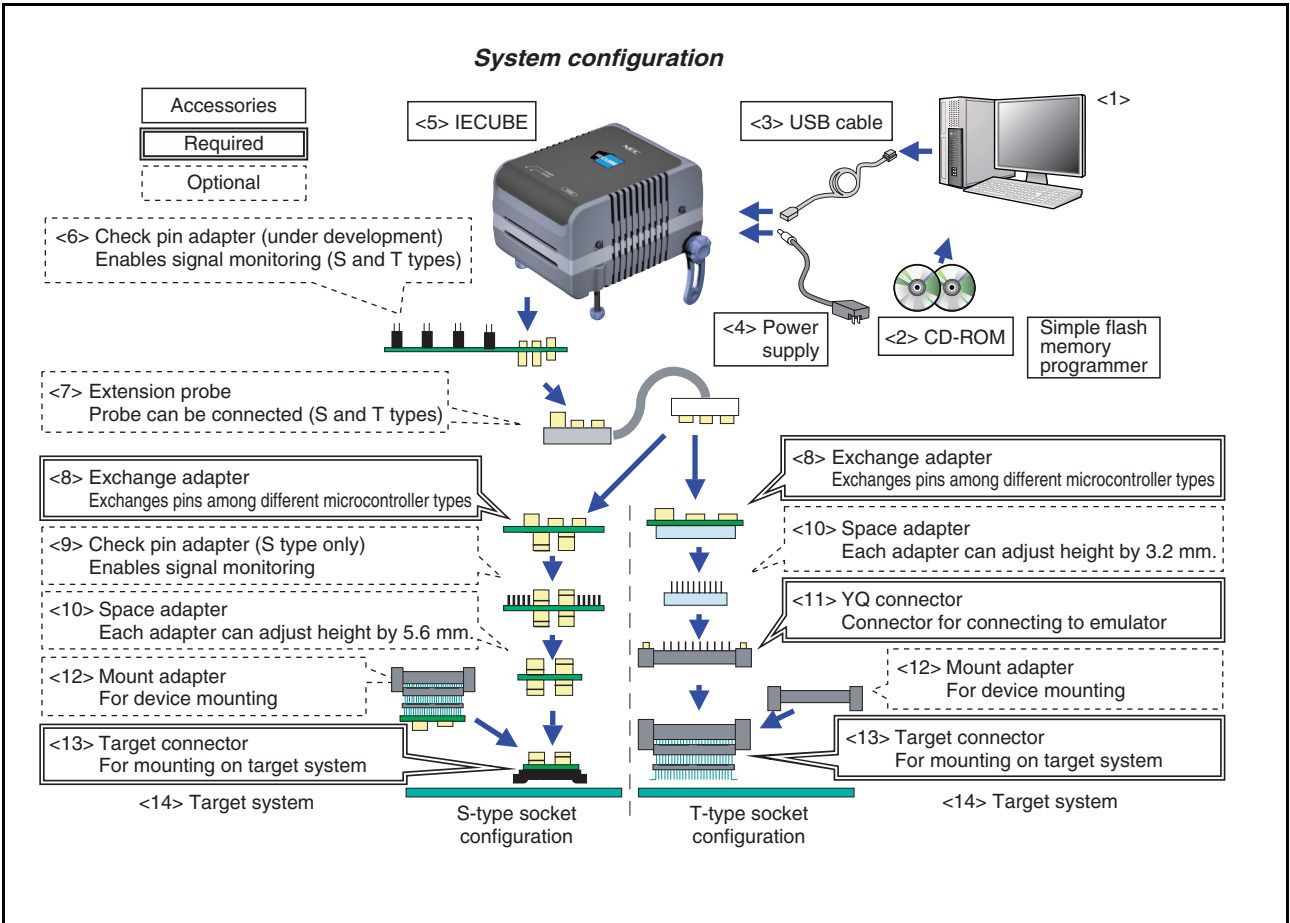


Figure A-2. System Configuration (When Using QB-V850ESJX3L, QB-V850ESSX2) (2/2)

- <1> Host machine (PC-9821 series, IBM-PC/AT compatibles)
- <2> Debugger, USB driver, manuals, etc. (ID850QB Disk, Accessory Disk^{Note 1})
- <3> USB interface cable
- <4> AC adapter
- <5> In-circuit emulator (QB-V850ESJX3L^{Note 2}, QB-V850ESSX2)
- <6> Check pin adapter (S and T types) (QB-144-CA-01^{Note 2}) (optional)
- <7> Extension probe (S and T types) (QB-144-EP-01S) (optional)
- <8> Exchange adapter^{Note 2}
- <9> Check pin adapter^{Notes 2,3} (optional)
- <10> Space adapter^{Notes 2,3} (optional)
- <11> YQ connector^{Note 2} (optional)
- <12> Mount adapter^{Note 2} (optional)
- <13> Target connector^{Note 2} (optional)
- <14> Target system

- Notes**
1. Download the device file from the Renesas Electronics website.
<http://www2.renesas.com/micro/en/ods/index.html>
 2. Under development
 3. When using both <9> and <10>, the order between <9> and <10> is not cared.

<5> QB-V850ESJX3L ^{Note1,2} QB-V850ESSX2 ^{Note1} In-circuit emulator	The in-circuit emulator serves to debug hardware and software when developing application systems using the V850ES/JC3-L, V850ES/JE3-L. It supports the integrated debugger ID850QB. This emulator should be used in combination with a power supply unit and emulation probe. Use the USB interface cable to connect this emulator to the host machine.
<3> USB interface cable	Cable to connect the host machine and the QB-V850ESJX3L ^{Note1,2} , QB-V850ESSX2 ^{Note1} .
<4> AC adapter	100 to 240 V can be supported by replacing the AC plug.
<8> Exchange adapter	Adapter to perform pin conversion.
<9> Check pin adapter	Adapter used in waveform monitoring using the oscilloscope, etc.
<10> Space adapter	Adapter to adjust the height.
<11> YQ connector	Conversion adapter to connect target connector and exchange adapter
<12> Mount adapter	Adapter to mount the V850ES/JC3-L, V850ES/JE3-L with socket.
<13> Target connector	Connector to solder on the target system.

Notes1. The QB-V850ESJX3L^{Note2}, QB-V850ESSX2 are supplied with a power supply unit, USB interface cable, and flash memory programmer (MINICUBE2). It is also supplied with integrated debugger ID850QB as control software.

2. Under development

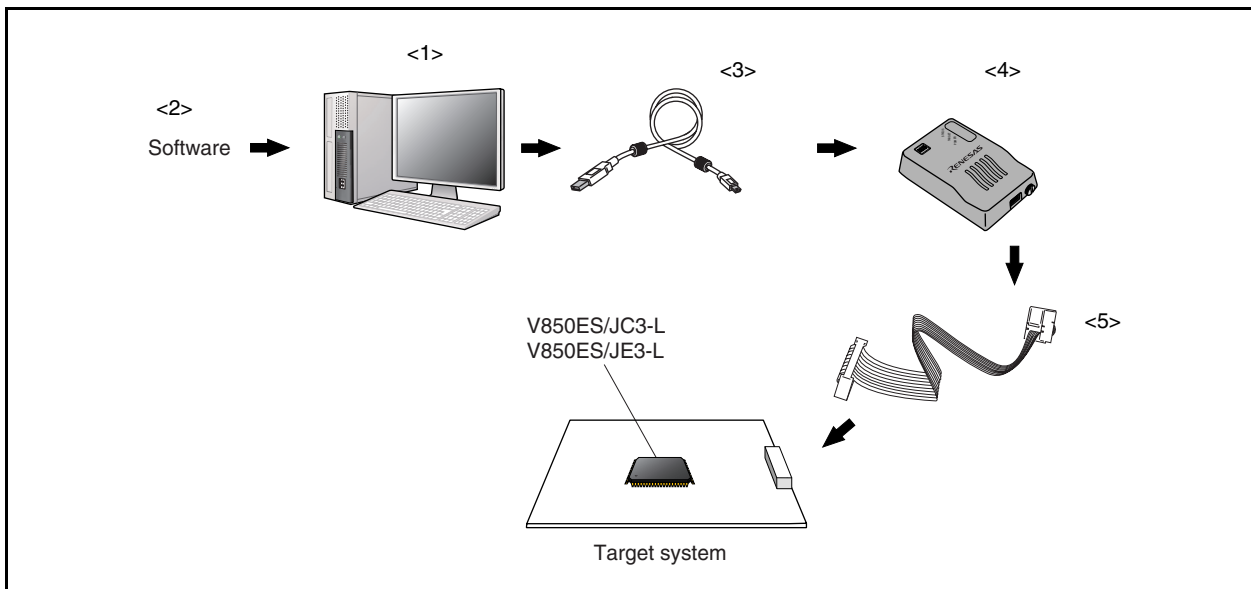
Remark The numbers in the angle brackets correspond to the numbers in Figure A-2.

A.4.2 When using MINICUBE QB-V850MINIL

(1) On-chip emulation using MINICUBE

The system configuration when connecting MINICUBE to the host machine (PC-9821 series, PC/AT compatible) is shown below.

Figure A-3. On-Chip Emulation System Configuration



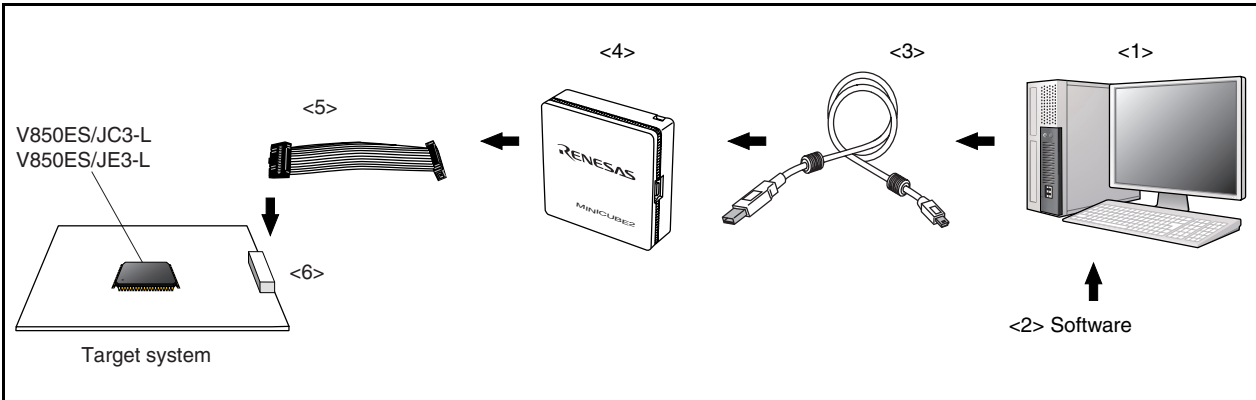
<1> Host machine	PC with USB ports
<2> Software	The integrated debugger ID850QB, device file, etc. Download the device file from the Renesas Electronics website. http://www2.renesas.com/micro/en/ods/index.html
<3> USB interface cable	USB cable to connect the host machine and MINICUBE. It is supplied with MINICUBE. The cable length is approximately 2 m.
<4> MINICUBE On-chip debug emulator	This on-chip debug emulator serves to debug hardware and software when developing application systems using the V850ES/JC3-L, V850ES/JE3-L. It supports integrated debugger ID850QB.
<5> OCD cable	Cable to connect MINICUBE and the target system. It is supplied with MINICUBE. The cable length is approximately 20 cm.

Remark The numbers in the angular brackets correspond to the numbers in Figure A-3.

A.4.3 When using MINICUBE2 QB-MINI2

The system configuration when connecting MINICUBE2 to the host machine (PC-9821 series, PC/AT compatible) is shown below.

Figure A-4. System Configuration of On-Chip Emulation System



<1> Host machine	PC with USB ports
<2> Software	The integrated debugger ID850QB, device file, etc. Download the device file from the Renesas Electronics website. http://www2.renesas.com/micro/en/ods/index.html
<3> USB interface cable	USB cable to connect the host machine and MINICUBE. It is supplied with MINICUBE. The cable length is approximately 2 m.
<4> MINICUBE2 On-chip debug emulator	This on-chip debug emulator serves to debug hardware and software when developing application systems using the V850ES/JC3-L, V850ES/JE3-L. It supports integrated debugger ID850QB.
<5> 16-pin target cable	Cable to connect MINICUBE2 and the target system. It is supplied with MINICUBE. The cable length is approximately 15 cm.
<6> Target connector (sold separately)	Use a 16-pin general-purpose connector with 2.54 mm pitch.

Remark The numbers in the angular brackets correspond to the numbers in Figure A-4.

A.5 Debugging Tools (Software)

ID850QB Integrated debugger	This debugger supports the in-circuit emulators for V850 microcontrollers. The ID850QB is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. It should be used in combination with the device file.
--------------------------------	--

A.6 Embedded Software

RX850, RX850 Pro Real-time OS	The RX850 and RX850 Pro are real-time OSs conforming to μ ITRON 3.0 specifications. A tool (configurator) for generating multiple information tables is supplied. RX850 Pro has more functions than the RX850.
	Part number: μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$ (RX850) μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$ (RX850 Pro)
Applilet [®] (under development)	This is a driver configurator that automatically generates sample programs for the V850ES/JC3-L, V850ES/JE3-L.
RX-FS850 (File system)	This is a FAT file system function. It is a file system that supports the CD-ROM file system function. This file system is used with the real-time OS RX850 Pro.

Caution To purchase the RX850 or RX850 Pro, first fill in the purchase application form and sign the license agreement.

Remark xxxx and $\Delta\Delta\Delta\Delta$ in the part number differ depending on the host machine and OS used.

μ SxxxxRX703000- $\Delta\Delta\Delta\Delta$

μ SxxxxRX703100- $\Delta\Delta\Delta\Delta$

$\Delta\Delta\Delta\Delta$	Product Outline	Maximum Number for Use in Mass Production
001	Evaluation object	Do not use for mass-produced product.
100K	Mass-production object	0.1 million units
001M		1 million units
010M		10 million units
S01	Source program	Object source program for mass production

xxxx	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	
3K17	SPARCstation	Solaris (Rel. 2.5.1)	

A.7 Flash Memory Writing Tools

Flashpro V (part number: PG-FP5) Flash memory programmer	Flash memory programmer dedicated to microcontrollers with internal flash memory.
QB-MINI2 (MINICUBE2)	On-chip debug emulator with programming function.
FA-100GC-UEU-B FA-121F1-CAH-B Flash memory writing adapter	Flash memory writing adapter used connected to the Flashpro IV, Flashpro V, etc. (not wired). • FA-100GC-UEU-B: 100-pin plastic LQFP (GC-UEU type) • FA-121F1-GAH-B: 121-pin plastic FBGA (F1-CAH type)
FA-70F3796GC-UEU-RX Flash memory writing adapter	Flash memory writing adapter used connected to the Flashpro IV, Flashpro V, etc. (already wired). • FA-70F3796GC-UEU-RX: 100-pin plastic LQFP

Remark FA-100GC-UEU-B, FA-70F3796F1-CAH-RX are products of Naito Densai Machida Mfg. Co., Ltd.
TEL: +81-42-750-4172

APPENDIX B MAJOR DIFFERENCES BETWEEN PRODUCTS

Table B-1. Major Differences between V850ES/Jx3-L products

Major Differences		V850ES/JC3-L	V850ES/JE3-L	V850ES/JF3-L	V850ES/JG3-L			
Device name		μPD70F3797, 70F3798, 70F3799, 70F3800, 70F3838	μPD70F3801, 70F3802, 70F3803, 70F3804, 70F3839	μPD70F3805, 70F3806, 70F3807, 70F3808, 70F3840	μPD70F3735, 70F3736	μPD70F3737, 70F3738	μPD70F3792, 70F3793	μPD70F3794, 70F3795, 70F3796
Pin		40	48	64	80	100/121	100/121	100/121
memory	Internal flash memory	16/32/64/128/ 256 KB	16/32/64/128/ 256 KB	16/32/64/128/ 256 KB	128/256 KB	128/256 KB	384/512 KB	256/384/512 KB
	Internal RAM	8/16 KB	8/16 KB	8/16 KB	8/16 KB	8/16 KB	32/40 KB	40 KB
External bus	Bus type	–	–	–	Multiplexed bus	Separate bus /multiplexed bus	Separate bus /multiplexed bus	Separate bus /multiplexed bus
	Address bus	–	–	–	18	22	22	22
	Address data bus	–	–	–	16	16	16	16
I/O port		27	34	50	66	84	83	80
Timer	16-bit TMP	6 ch	6 ch	6 ch	4 ch	6 ch	6 ch	6 ch
	16-bit TMQ	1 ch	1 ch	1 ch	1 ch	1 ch	1 ch	1 ch
	16-bit TMM	1 ch	1 ch	1 ch	1 ch	1 ch	1 ch	1 ch
	Watch timer	1 ch	1 ch	1 ch	1 ch	1 ch	1 ch	1 ch
	RTC	1 ch	1 ch	1 ch	–	–	1 ch	1 ch
	WDT	1 ch	1 ch	1 ch	1 ch	1 ch	1 ch	1 ch
10-bit A/D converter		5 ch	6 ch	10 ch	8 ch	12 ch	12 ch	12 ch
8-bit D/A converter		–	1 ch	1 ch	1 ch	2 ch	2 ch	2 ch
Serial interface	CSIB	2 ch	4 ch	5 ch	3 ch	5 ch	5 ch	5 ch
	UARTA	2 ch	3 ch	3 ch	3 ch	3 ch	6 ch	6 ch
	UARTC	–	–	–	–	–	1 ch	1 ch
	I ² C	2 ch	3 ch	3 ch	2 ch	3 ch	3 ch	3 ch
	USBF	–	–	–	–	–	–	1 ch
DMA controller		4 ch	4 ch	4 ch	4 ch	4 ch	4 ch	4 ch
Interrupt source	External	6	6	9	9	9	9	9
	Internal	42	46	48	40	48	55	55
RTC backup mode		None	None	None	None	None	Available	Available
Operating power supply voltage		2.2 to 3.6V	2.2 to 3.6V	2.2 to 3.6V	2.2 to 3.6V	2.2 to 3.6V	2.0 to 3.6V	2.0 to 3.6V
Package		40-pinWQFN (6×6 mm)	48-pinWQFN (7×7 mm), 48-pinLQFP (7×7 mm)	64-pinLQFP (10×10 mm)	80-pinLQFP (14×14 mm), 80-pinLQFP (12×12 mm)	100-pinLQFP (14×14 mm), 100-pinLQFP (14×20 mm), 121-pinFBGA (8×8 mm)	100-pinLQFP (14×14 mm), 121-pinFBGA (8×8 mm)	100-pinLQFP (14×14 mm), 121-pinFBGA (8×8 mm)

APPENDIX C REGISTER INDEX

(1/10)

Symbol	Name	Unit	Page
ADA0CR0	A/D conversion result register 0	ADC	484
ADA0CR0H	A/D conversion result register 0H	ADC	484
ADA0CR1	A/D conversion result register 1	ADC	484
ADA0CR1H	A/D conversion result register 1H	ADC	484
ADA0CR2	A/D conversion result register 2	ADC	484
ADA0CR2H	A/D conversion result register 2H	ADC	484
ADA0CR3	A/D conversion result register 3	ADC	484
ADA0CR3H	A/D conversion result register 3H	ADC	484
ADA0CR4	A/D conversion result register 4	ADC	484
ADA0CR4H	A/D conversion result register 4H	ADC	484
ADA0CR5	A/D conversion result register 5	ADC	484
ADA0CR5H	A/D conversion result register 5H	ADC	484
ADA0CR6	A/D conversion result register 6	ADC	484
ADA0CR6H	A/D conversion result register 6H	ADC	484
ADA0CR7	A/D conversion result register 7	ADC	484
ADA0CR7H	A/D conversion result register 7H	ADC	484
ADA0CR8	A/D conversion result register 8	ADC	484
ADA0CR8H	A/D conversion result register 8H	ADC	484
ADA0CR9	A/D conversion result register 9	ADC	484
ADA0CR9H	A/D conversion result register 9H	ADC	484
ADA0M0	A/D converter mode register 0	ADC	477
ADA0M1	A/D converter mode register 1	ADC	479
ADA0M2	A/D converter mode register 2	ADC	482
ADA0PFM	Power fail compare mode register	ADC	486
ADA0PFT	Power fail compare threshold value register	ADC	487
ADA0S	Analog input channel specification register	ADC	483
ADIC	Interrupt control register	INTC	728
CB0CTL0	CSIB0 control register 0	CSIB	566
CB0CTL1	CSIB0 control register 1	CSIB	569
CB0CTL2	CSIB0 control register 2	CSIB	570
CB0RIC	Interrupt control register	INTC	728
CB0RX	CSIB0 receive data register	CSIB	564
CB0RXL	CSIB0 receive data register L	CSIB	564
CB0STR	CSIB0 status register	CSIB	572
CB0TIC	Interrupt control register	INTC	728
CB0TX	CSIB0 transmit data register	CSIB	565
CB0TXL	CSIB0 transmit data register L	CSIB	565
CB1CTL0	CSIB1 control register 0	CSIB	566
CB1CTL1	CSIB1 control register 1	CSIB	569

(2/10)

Symbol	Name	Unit	Page
CB1CTL2	CSIB1 control register 2	CSIB	570
CB1RIC	Interrupt control register	INTC	728
CB1RX	CSIB1 receive data register	CSIB	564
CB1RXL	CSIB1 receive data register L	CSIB	564
CB1STR	CSIB1 status register	CSIB	572
CB1TIC	Interrupt control register	INTC	728
CB1TX	CSIB1 transmit data register	CSIB	565
CB1TXL	CSIB1 transmit data register L	CSIB	565
CB2CTL0	CSIB2 control register 0	CSIB	566
CB2CTL1	CSIB2 control register 1	CSIB	569
CB2CTL2	CSIB2 control register 2	CSIB	570
CB2RIC	Interrupt control register	INTC	728
CB2RX	CSIB2 receive data register	CSIB	564
CB2RXL	CSIB2 receive data register L	CSIB	564
CB2STR	CSIB2 status register	CSIB	572
CB2TIC	Interrupt control register	INTC	728
CB2TX	CSIB2 transmit data register	CSIB	565
CB2TXL	CSIB2 transmit data register L	CSIB	565
CB3CTL0	CSIB3 control register 0	CSIB	566
CB3CTL1	CSIB3 control register 1	CSIB	569
CB3CTL2	CSIB3 control register 2	CSIB	570
CB3RIC	Interrupt control register	INTC	728
CB3RX	CSIB3 receive data register	CSIB	564
CB3RXL	CSIB3 receive data register L	CSIB	564
CB3STR	CSIB3 status register	CSIB	572
CB3TIC	Interrupt control register	INTC	728
CB3TX	CSIB3 transmit data register	CSIB	565
CB3TXL	CSIB3 transmit data register L	CSIB	565
CB4CTL0	CSIB4 control register 0	CSIB	566
CB4CTL1	CSIB4 control register 1	CSIB	569
CB4CTL2	CSIB4 control register 2	CSIB	570
CB4RIC	Interrupt control register	INTC	728
CB4RX	CSIB4 receive data register	CSIB	564
CB4RXL	CSIB4 receive data register L	CSIB	564
CB4STR	CSIB4 status register	CSIB	572
CB4TIC	Interrupt control register	INTC	728
CB4TX	CSIB4 transmit data register	CSIB	565
CB4TXL	CSIB4 transmit data register L	CSIB	565
CCLS	CPU operation clock status register	CG	189
CKC	Clock control register	CG	192
CLM	Clock monitor mode register	CLM	799
CRCD	CRC data register	CRC	809
CRCIN	CRC input register	CRC	809

(3/10)

Symbol	Name	Unit	Page
CTBP	CALLT base pointer	CPU	58
CTPC	CALLT execution status saving register	CPU	57
CTPSW	CALLT execution status saving register	CPU	57
DA0CS0	D/A conversion value setting register 0	DAC	517
DA0M	D/A converter mode register	DAC	516
DADC0	DMA addressing control register 0	DMAC	697
DADC1	DMA addressing control register 1	DMAC	697
DADC2	DMA addressing control register 2	DMAC	697
DADC3	DMA addressing control register 3	DMAC	697
DBC0	DMA transfer count register 0	DMAC	696
DBC1	DMA transfer count register 1	DMAC	696
DBC2	DMA transfer count register 2	DMAC	696
DBC3	DMA transfer count register 3	DMAC	696
DBPC	Exception/debug trap status saving register	CPU	58
DBPSW	Exception/debug trap status saving register	CPU	58
DCHC0	DMA channel control register 0	DMAC	698
DCHC1	DMA channel control register 1	DMAC	698
DCHC2	DMA channel control register 2	DMAC	698
DCHC3	DMA channel control register 3	DMAC	698
DDA0H	DMA destination address register 0H	DMAC	695
DDA0L	DMA destination address register 0L	DMAC	695
DDA1H	DMA destination address register 1H	DMAC	695
DDA1L	DMA destination address register 1L	DMAC	695
DDA2H	DMA destination address register 2H	DMAC	695
DDA2L	DMA destination address register 2L	DMAC	695
DDA3H	DMA destination address register 3H	DMAC	695
DDA3L	DMA destination address register 3L	DMAC	695
DMAIC0	Interrupt control register	INTC	728
DMAIC1	Interrupt control register	INTC	728
DMAIC2	Interrupt control register	INTC	728
DMAIC3	Interrupt control register	INTC	728
DSA0H	DMA source address register 0H	DMAC	694
DSA0L	DMA source address register 0L	DMAC	694
DSA1H	DMA source address register 1H	DMAC	694
DSA1L	DMA source address register 1L	DMAC	694
DSA2H	DMA source address register 2H	DMAC	694
DSA2L	DMA source address register 2L	DMAC	694
DSA3H	DMA source address register 3H	DMAC	694
DSA3L	DMA source address register 3L	DMAC	694
DTFR0	DMA trigger factor register 0	DMAC	699
DTFR1	DMA trigger factor register 1	DMAC	699
DTFR2	DMA trigger factor register 2	DMAC	699
DTFR3	DMA trigger factor register 3	DMAC	699

(4/10)

Symbol	Name	Unit	Page
ECR	Interrupt source register	CPU	55
EIPC	Interrupt status saving register	CPU	54
EIPSW	Interrupt status saving register	CPU	54
FEPC	NMI status saving register	CPU	55
FEPSW	NMI status saving register	CPU	55
IIC0	IIC shift register 0	I ² C	634
IIC1	IIC shift register 1	I ² C	634
IIC2	IIC shift register 2	I ² C	634
IICC0	IIC control register 0	I ² C	620
IICC1	IIC control register 1	I ² C	620
IICC2	IIC control register 2	I ² C	620
IICCL0	IIC clock select register 0	I ² C	630
IICCL1	IIC clock select register 1	I ² C	630
IICCL2	IIC clock select register 2	I ² C	630
IICF0	IIC flag register 0	I ² C	628
IICF1	IIC flag register 1	I ² C	628
IICF2	IIC flag register 2	I ² C	628
IICIC0	Interrupt control register	INTC	728
IICIC1	Interrupt control register	INTC	728
IICIC2	Interrupt control register	INTC	728
IICS0	IIC status register 0	I ² C	625
IICS1	IIC status register 1	I ² C	625
IICS2	IIC status register 2	I ² C	625
IICX0	IIC function expansion register 0	I ² C	631
IICX1	IIC function expansion register 1	I ² C	631
IICX2	IIC function expansion register 2	I ² C	631
IMR0	Interrupt mask register 0	INTC	730
IMR0H	Interrupt mask register 0H	INTC	730
IMR0L	Interrupt mask register 0L	INTC	730
IMR1	Interrupt mask register 1	INTC	730
IMR1H	Interrupt mask register 1H	INTC	730
IMR1L	Interrupt mask register 1L	INTC	730
IMR2	Interrupt mask register 2	INTC	730
IMR2H	Interrupt mask register 2H	INTC	730
IMR2L	Interrupt mask register 2L	INTC	730
IMR3	Interrupt mask register 3	INTC	730
IMR3H	Interrupt mask register 3H	INTC	730
IMR3L	Interrupt mask register 3L	INTC	730
INTF0	External interrupt falling edge specification register 0	INTC	743
INTF3	External interrupt falling edge specification register 3	INTC	744
INTF9H	External interrupt falling edge specification register 9H	INTC	745
INTR0	External interrupt rising edge specification register 0	INTC	743
INTR3	External interrupt rising edge specification register 3	INTC	744

(5/10)

Symbol	Name	Unit	Page
INTR9H	External interrupt rising edge specification register 9H	INTC	745
ISPR	In-service priority register	INTC	732
KRIC	Interrupt control register	INTC	728
KRM	Key return mode register	KR	751
LOCKR	Lock register	CG	193
LVIIC	Interrupt control register	INTC	728
LVIM	Low voltage detection register	LVI	804
LVIS	Low voltage detection level select register	LVI	805
NFC	Noise elimination control register	INTC	746
OCDM	On-chip debug mode register	DCU	844
OCKS0	IIC division clock select register 0	I ² C	634
OCKS1	IIC division clock select register 1	I ² C	634
OSTS	Oscillation stabilization time select register	Standby	757
P0	Port 0 register	Port	100
P1	Port 1 register	Port	105
P3	Port 3 register	Port	107
P3H	Port 3 register H	Port	107
P3L	Port 3 register L	Port	107
P4	Port 4 register	Port	114
P5	Port 5 register	Port	116
P7H	Port 7 register H	Port	122
P7L	Port 7 register L	Port	122
P9	Port 9 register	Port	125
P9H	Port 9 register H	Port	125
P9L	Port 9 register L	Port	125
PC	Program counter	CPU	52
PCC	Processor clock control register	CG	185
PCM	Port CM register	Port	136
PDL	Port DL register	Port	138
PDLL	Port DL register L	Port	138
PF0	Port 0 function register	Port	104
PF3	Port 3 function register	Port	113
PF3H	Port 3 function register H	Port	113
PF3L	Port 3 function register L	Port	113
PF4	Port 4 function register	Port	115
PF5	Port 5 function register	Port	120
PF9	Port 9 function register	Port	135
PF9H	Port 9 function register H	Port	135
PF9L	Port 9 function register L	Port	135
PFC0	Port 0 function control register	Port	103
PFC3	Port 3 function control register	Port	111
PFC3H	Port 3 function control register H	Port	111
PFC3L	Port 3 function control register L	Port	111

(6/10)

Symbol	Name	Unit	Page
PFC4	Port 4 function control register	Port	115
PFC5	Port 5 function control register	Port	118
PFC9	Port 9 function control register	Port	131
PFC9H	Port 9 function control register H	Port	131
PFC9L	Port 9 function control register L	Port	131
PFCE0	Port 0 function control extension register	Port	103
PFCE3L	Port 3 function control extension register L	Port	111
PFCE5	Port 5 function control extension register	Port	118
PFCE9	Port 9 function control extension register	Port	132
PFCE9H	Port 9 function control extension register H	Port	132
PFCE9L	Port 9 function control extension register L	Port	132
PIC0	Interrupt control register	INTC	728
PIC1	Interrupt control register	INTC	728
PIC2	Interrupt control register	INTC	728
PIC3	Interrupt control register	INTC	728
PIC4	Interrupt control register	INTC	728
PIC5	Interrupt control register	INTC	728
PIC6	Interrupt control register	INTC	728
PIC7	Interrupt control register	INTC	728
PLLCTL	PLL control register	CG	191
PLLS	PLL lockup time specification register	CG	194
PM0	Port 0 mode register	Port	101
PM1	Port 1 mode register	Port	105
PM3	Port 3 mode register	Port	108
PM3H	Port 3 mode register H	Port	108
PM3L	Port 3 mode register L	Port	108
PM4	Port 4 mode register	Port	114
PM5	Port 5 mode register	Port	117
PM7H	Port 7 mode register H	Port	123
PM7L	Port 7 mode register L	Port	123
PM9	Port 9 mode register	Port	126
PM9H	Port 9 mode register H	Port	126
PM9L	Port 9 mode register L	Port	126
PMC0	Port 0 mode control register	Port	102
PMC3	Port 3 mode control register	Port	109
PMC3H	Port 3 mode control register H	Port	109
PMC3L	Port 3 mode control register L	Port	109
PMC4	Port 4 mode control register	Port	115
PMC5	Port 5 mode control register	Port	117
PMC9	Port 9 mode control register	Port	127
PMC9H	Port 9 mode control register H	Port	127
PMC9L	Port 9 mode control register L	Port	127
PMCM	Port CM mode register	Port	136

(7/10)

Symbol	Name	Unit	Page
PMDL	Port DL mode register	Port	138
PMDLL	Port DL mode register L	Port	138
PRCMD	Command register	CPU	84
PRSCM0	Prescaler compare register 0	WT, RTC	423, 447
PRSCM1	Prescaler compare register 1	BRG	609
PRSCM2	Prescaler compare register 2	BRG	609
PRSCM3	Prescaler compare register 3	BRG	609
PRSM0	Prescaler mode register 0	WT, RTC	422, 448
PRSM1	Prescaler mode register 1	BRG	608
PRSM2	Prescaler mode register 2	BRG	608
PRSM3	Prescaler mode register 3	BRG	608
PSC	Power save control register	Standby	755
PSMR	Power save mode register	Standby	756
PSW	Program status word	CPU	56
r0 to r31	General-purpose registers	CPU	52
RC1ALH	Alarm minute set register	RTC	445
RC1ALM	Alarm time set register	RTC	445
RC1ALW	Alarm week set register	RTC	446
RC1CC0	RTC control register 0	RTC	434
RC1CC1	RTC control register 1	RTC	434
RC1CC2	RTC control register 2	RTC	436
RC1CC3	RTC control register 3	RTC	437
RC1DAY	Day count register	RTC	441
RC1HOUR	Hour count register	RTC	439
RC1MIN	Minute count register	RTC	439
RC1MONTH	Month count register	RTC	443
RC1SEC	Second count register	RTC	438
RC1SUBC	Sub-count register	RTC	438
RC1SUBU	Time error correction register	RTC	444
RC1WEEK	Week count register	RTC	442
RC1YEAR	Year count register	RTC	443
RCM	Internal oscillation mode register	CG	189
REGOVL0	Regulator output voltage level control register 0	Standby	755
REGPR	Regulator protection register	Standby	758
RESF	Reset source flag register	Reset	787
RTBH0	Real-time output buffer register 0H	RTP	468
RTBL0	Real-time output buffer register 0L	RTP	468
RTC0IC	Interrupt control register	INTC	728
RTC1IC	Interrupt control register	INTC	728
RTC2IC	Interrupt control register	INTC	728
RTPC0	Real-time output port control register 0	RTP	470
RTPM0	Real-time output port mode register 0	RTP	469
SELCNT0	Selector operation control register 0	Timer	297

(8/10)

Symbol	Name	Unit	Page
SVA0	Slave address register 0	I ² C	635
SVA1	Slave address register 1	I ² C	635
SVA2	Slave address register 2	I ² C	635
SYS	System status register	CPU	85
TM0CMP0	TMM0 compare register 0	Timer	413
TM0CTL0	TMM0 control register 0	Timer	412
TM0EQIC0	Interrupt control register	INTC	728
TP0CCIC0	Interrupt control register	INTC	728
TP0CCIC1	Interrupt control register	INTC	728
TP0CCR0	TMP0 capture/compare register 0	Timer	214
TP0CCR1	TMP0 capture/compare register 1	Timer	216
TP0CNT	TMP0 counter read buffer register	Timer	218
TP0CTL0	TMP0 control register 0	Timer	208
TP0CTL1	TMP0 control register 1	Timer	209
TP0IOC0	TMP0 I/O control register 0	Timer	210
TP0IOC1	TMP0 I/O control register 1	Timer	211
TP0IOC2	TMP0 I/O control register 2	Timer	212
TP0OPT0	TMP0 option register 0	Timer	213
TP0OVIC	Interrupt control register	INTC	728
TP1CCIC0	Interrupt control register	INTC	728
TP1CCIC1	Interrupt control register	INTC	728
TP1CCR0	TMP1 capture/compare register 0	Timer	214
TP1CCR1	TMP1 capture/compare register 1	Timer	216
TP1CNT	TMP1 counter read buffer register	Timer	218
TP1CTL0	TMP1 control register 0	Timer	208
TP1CTL1	TMP1 control register 1	Timer	209
TP1IOC0	TMP1 I/O control register 0	Timer	210
TP1IOC1	TMP1 I/O control register 1	Timer	211
TP1IOC2	TMP1 I/O control register 2	Timer	212
TP1OPT0	TMP1 option register 0	Timer	213
TP1OVIC	Interrupt control register	INTC	728
TP2CCIC0	Interrupt control register	INTC	728
TP2CCIC1	Interrupt control register	INTC	728
TP2CCR0	TMP2 capture/compare register 0	Timer	214
TP2CCR1	TMP2 capture/compare register 1	Timer	216
TP2CNT	TMP2 counter read buffer register	Timer	218
TP2CTL0	TMP2 control register 0	Timer	208
TP2CTL1	TMP2 control register 1	Timer	209
TP2IOC0	TMP2 I/O control register 0	Timer	210
TP2IOC1	TMP2 I/O control register 1	Timer	211
TP2IOC2	TMP2 I/O control register 2	Timer	212
TP2OPT0	TMP2 option register 0	Timer	213
TP2OVIC	Interrupt control register	INTC	728
TP3CCIC0	Interrupt control register	INTC	728

(9/10)

Symbol	Name	Unit	Page
TP3CCIC1	Interrupt control register	INTC	728
TP3CCR0	TMP3 capture/compare register 0	Timer	214
TP3CCR1	TMP3 capture/compare register 1	Timer	216
TP3CNT	TMP3 counter read buffer register	Timer	218
TP3CTL0	TMP3 control register 0	Timer	208
TP3CTL1	TMP3 control register 1	Timer	209
TP3IOC0	TMP3 I/O control register 0	Timer	210
TP3IOC1	TMP3 I/O control register 1	Timer	211
TP3OPT0	TMP3 option register 0	Timer	213
TP3OVIC	Interrupt control register	INTC	728
TP4CCIC0	Interrupt control register	INTC	728
TP4CCIC1	Interrupt control register	INTC	728
TP4CCR0	TMP4 capture/compare register 0	Timer	214
TP4CCR1	TMP4 capture/compare register 1	Timer	216
TP4CNT	TMP4 counter read buffer register	Timer	218
TP4CTL0	TMP4 control register 0	Timer	208
TP4CTL1	TMP4 control register 1	Timer	209
TP4IOC0	TMP4 I/O control register 0	Timer	210
TP4IOC1	TMP4 I/O control register 1	Timer	211
TP4IOC2	TMP4 I/O control register 2	Timer	212
TP4OPT0	TMP4 option register 0	Timer	213
TP4OVIC	Interrupt control register	INTC	728
TP5CCIC0	Interrupt control register	INTC	728
TP5CCIC1	Interrupt control register	INTC	728
TP5CCR0	TMP5 capture/compare register 0	Timer	214
TP5CCR1	TMP5 capture/compare register 1	Timer	216
TP5CNT	TMP5 counter read buffer register	Timer	218
TP5CTL0	TMP5 control register 0	Timer	208
TP5CTL1	TMP5 control register 1	Timer	209
TP5IOC0	TMP5 I/O control register 0	Timer	210
TP5IOC1	TMP5 I/O control register 1	Timer	211
TP5IOC2	TMP5 I/O control register 2	Timer	212
TP5OPT0	TMP5 option register 0	Timer	213
TP5OVIC	Interrupt control register	INTC	728
TQ0CCIC0	Interrupt control register	INTC	728
TQ0CCIC1	Interrupt control register	INTC	728
TQ0CCIC2	Interrupt control register	INTC	728
TQ0CCIC3	Interrupt control register	INTC	728
TQ0CCR0	TMQ0 capture/compare register 0	Timer	310
TQ0CCR1	TMQ0 capture/compare register 1	Timer	321
TQ0CCR2	TMQ0 capture/compare register 2	Timer	314
TQ0CCR3	TMQ0 capture/compare register 3	Timer	316
TQ0CNT	TMQ0 counter read buffer register	Timer	317

(10/10)

Symbol	Name	Unit	Page
TQ0CTL0	TMQ0 control register 0	Timer	304
TQ0CTL1	TMQ0 control register 1	Timer	305
TQ0IOC0	TMQ0 I/O control register 0	Timer	306
TQ0IOC1	TMQ0 I/O control register 1	Timer	307
TQ0IOC2	TMQ0 I/O control register 2	Timer	308
TQ0OPT0	TMQ0 option register 0	Timer	309
TQ0OVIC	Interrupt control register	INTC	728
UA0CTL0	UARTA0 control register 0	UARTA	527
UA0CTL1	UARTA0 control register 1	UARTA	551
UA0CTL2	UARTA0 control register 2	UARTA	552
UA0OPT0	UARTA0 option control register 0	UARTA	529
UA0RIC	Interrupt control register	INTC	728
UA0RX	UARTA0 receive data register	UARTA	532
UA0STR	UARTA0 status register	UARTA	530
UA0TIC	Interrupt control register	INTC	728
UA0TX	UARTA0 transmit data register	UARTA	533
UA1CTL0	UARTA1 control register 0	UARTA	527
UA1CTL1	UARTA1 control register 1	UARTA	551
UA1CTL2	UARTA1 control register 2	UARTA	552
UA1OPT0	UARTA1 option control register 0	UARTA	529
UA1RIC	Interrupt control register	INTC	728
UA1RX	UARTA1 receive data register	UARTA	532
UA1STR	UARTA1 status register	UARTA	530
UA1TIC	Interrupt control register	INTC	728
UA1TX	UARTA1 transmit data register	UARTA	533
UA2CTL0	UARTA2 control register 0	UARTA	527
UA2CTL1	UARTA2 control register 1	UARTA	551
UA2CTL2	UARTA2 control register 2	UARTA	552
UA2OPT0	UARTA2 option control register 0	UARTA	529
UA2RIC	Interrupt control register	INTC	728
UA2RX	UARTA2 receive data register	UARTA	532
UA2STR	UARTA2 status register	UARTA	530
UA2TIC	Interrupt control register	INTC	728
UA2TX	UARTA2 transmit data register	UARTA	533
VSWC	System wait control register	CPU	86
WDTE	Watchdog timer enable register	WDT	465
WDTM2	Watchdog timer mode register 2	WDT	463
WTIC	Interrupt control register	INTC	728
WTIIC	Interrupt control register	INTC	728
WTM	Watch timer operation mode register	WT	424

APPENDIX D INSTRUCTION SET LIST

D.1 Conventions

(1) Register symbols used to describe operands

Register Symbol	Explanation
reg1	General-purpose registers: Used as source registers.
reg2	General-purpose registers: Used mainly as destination registers. Also used as source register in some instructions.
reg3	General-purpose registers: Used mainly to store the remainders of division results and the higher 32 bits of multiplication results.
bit#3	3-bit data for specifying the bit number
immX	X bit immediate data
dispX	X bit displacement data
regID	System register number
vector	5-bit data that specifies the trap vector (00H to 1FH)
cccc	4-bit data that shows the conditions code
sp	Stack pointer (r3)
ep	Element pointer (r30)
listX	X item register list

(2) Register symbols used to describe opcodes

Register Symbol	Explanation
R	1-bit data of a code that specifies reg1 or regID
r	1-bit data of the code that specifies reg2
w	1-bit data of the code that specifies reg3
d	1-bit displacement data
l	1-bit immediate data (indicates the higher bits of immediate data)
i	1-bit immediate data
cccc	4-bit data that shows the condition codes
CCCC	4-bit data that shows the condition codes of Bcond instruction
bbb	3-bit data for specifying the bit number
L	1-bit data that specifies a program register in the register list

(3) Register symbols used in operations

Register Symbol	Explanation
←	Input for
GR []	General-purpose register
SR []	System register
zero-extend (n)	Expand n with zeros until word length.
sign-extend (n)	Expand n with signs until word length.
load-memory (a, b)	Read size b data from address a.
store-memory (a, b, c)	Write data b into address a in size c.
load-memory-bit (a, b)	Read bit b of address a.
store-memory-bit (a, b, c)	Write c to bit b of address a.
saturated (n)	Execute saturated processing of n (n is a 2's complement). If, as a result of calculations, n ≥ 7FFFFFFFH, let it be 7FFFFFFFH. n ≤ 80000000H, let it be 80000000H.
result	Reflects the results in a flag.
Byte	Byte (8 bits)
Halfword	Half word (16 bits)
Word	Word (32 bits)
+	Addition
−	Subtraction
	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 Code (cccc)	Condition Formula	Explanation
0 0 0 0	$OV = 1$	Overflow
1 0 0 0	$OV = 0$	No overflow
0 0 0 1	$CY = 1$	Carry Lower (Less than)
1 0 0 1	$CY = 0$	No carry Not lower (Greater than or equal)
0 0 1 0	$Z = 1$	Zero
1 0 1 0	$Z = 0$	Not zero
0 0 1 1	$(CY \text{ or } Z) = 1$	Not higher (Less than or equal)
1 0 1 1	$(CY \text{ or } Z) = 0$	Higher (Greater than)
0 1 0 0	$S = 1$	Negative
1 1 0 0	$S = 0$	Positive
0 1 0 1	–	Always (Unconditional)
1 1 0 1	$SAT = 1$	Saturated
0 1 1 0	$(S \text{ xor } OV) = 1$	Less than signed
1 1 1 0	$(S \text{ xor } OV) = 0$	Greater than or equal signed
0 1 1 1	$((S \text{ xor } OV) \text{ or } Z) = 1$	Less than or equal signed
1 1 1 1	$((S \text{ xor } OV) \text{ or } Z) = 0$	Greater than signed

D.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	rrrrr010010iiii	GR[reg2]←GR[reg2]+sign-extend(imm5)	1	1	1	×	×	×	×		
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)	1	1	1	×	×	×	×		
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]	1	1	1		0	×	×		
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND zero-extend(imm16)	1	1	1		0	×	×		
Bcond	disp9	dddd1011dddcccc Note 1	if conditions are satisfied then PC←PC+sign-extend(disp9)	When conditions are satisfied	2	2	2					
				When conditions are not satisfied	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 www01101000010	GR[reg3]←GR[reg2] (23 : 16) GR[reg2] (31 : 24) GR[reg2] (7 : 0) GR[reg2] (15 : 8)	1	1	1	×	0	×	×		
BSW	reg2,reg3	rrrrr11111100000 www01101000000	GR[reg3]←GR[reg2] (7 : 0) GR[reg2] (15 : 8) GR [reg2] (23 : 16) GR[reg2] (31 : 24)	1	1	1	×	0	×	×		
CALLT	imm6	0000001000iiii	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+zero-extend(imm6 logically shift left by 1) PC←CTBP+zero-extend(Load-memory(adr,Halfword))	4	4	4						
CLR1	bit#3,disp16[reg1]	10bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0)	3	3	3				×		
	reg2,[reg1]	rrrrr111111RRRRR 000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)	3	3	3				×		
CMOV	cccc,imm5,reg2,reg3	rrrrr111111iiii www01100cccc0	if conditions are satisfied then GR[reg3]←sign-extended(imm5) else GR[reg3]←GR[reg2]	1	1	1						
	cccc,reg1,reg2,reg3	rrrrr111111RRRR www011001cccc0	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	rrrrr010011iiii	result←GR[reg2]-sign-extend(imm5)	1	1	1	×	×	×	×		
CTRET		000011111100000 0000000101000100	PC←CTPC PSW←CTPSW	3	3	3	R	R	R	R	R	
DBRET		000001111100000 0000000101000110	PC←DBPC PSW←DBPSW	3	3	3	R	R	R	R	R	

(2/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags					
				i	r	l	CY	OV	S	Z	SAT	
DBTRAP		1111100001000000	DBPC←PC+2 (restored PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	3	3	3						
DI		0000011111100000 0000000101100000	PSW.ID←1	1	1	1						
DISPOSE	imm5,list12	0000011001iiiiL LLLLLLLLLLLL00000	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]	0000011001iiiiL LLLLLLLLLLLLRRRRR 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	rrrrr11111RRRRR www01011000000	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	rrrrr11111RRRRR www01010000000	GR[reg2]←GR[reg2]÷GR[reg1] ^{Note 6} GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×		
DIVHU	reg1,reg2,reg3	rrrrr11111RRRRR www01010000010	GR[reg2]←GR[reg2]÷GR[reg1] ^{Note 6} GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×		
DIVU	reg1,reg2,reg3	rrrrr11111RRRRR www01011000010	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 www01101000100	GR[reg3]←GR[reg2](15 : 0) GR[reg2] (31 : 16)	1	1	1	×	0	×	×		
JARL	disp22,reg2	rrrrr11110dddddd dddddddddddddd0 Note 7	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2						
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	3	3	3						
JR	disp22	0000011110dddddd dddddddddddddd0 Note 7	PC←PC+sign-extend(disp22)	2	2	2						
LD.B	disp16[reg1],reg2	rrrrr111000RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 11						
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRRR 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 dddddddddddddd0 Note 8	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	Note 11						
LDSR	reg2,regID	rrrrr111111RRRRR 000000000100000 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	rrrrr010000iiii	GR[reg2]←sign-extend(imm5)	1	1	1						
	imm32,reg1	00000110001RRRRR iiiiiiiiiiiiiiii iiiiiiiiiiiiiiii	GR[reg1]←imm32	2	2	2						
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)	1	1	1						
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+(imm16 ll 0 ¹⁶)	1	1	1						
MUL	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000 Note 14	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1]	1	4	5						
	imm9,reg2,reg3	rrrrr111111iiii wwwww01001111100 Note 13	GR[reg3] ll GR[reg2]←GR[reg2]xsign-extend(imm9)	1	4	5						
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2] ^{Note 6} xGR[reg1] ^{Note 6}	1	1	2						
	imm5,reg2	rrrrr010111iiii	GR[reg2]←GR[reg2] ^{Note 6} xsign-extend(imm5)	1	1	2						
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] ^{Note 6} ximm16	1	1	2						
MULU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010 Note 14	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1]	1	4	5						
	imm9,reg2,reg3	rrrrr111111iiii wwwww0100111110 Note 13	GR[reg3] ll GR[reg2]←GR[reg2]xzero-extend(imm9)	1	4	5						
NOP		0000000000000000	Pass at least one clock cycle doing nothing.	1	1	1						
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])	1	1	1		0	×	×		
NOT1	bit#3,disp16[reg1]	01bbb11110RRRRR 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	3	3					×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,Z flag)	3	3	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	0000011110iiiiL LLLLLLLLLLLL00001	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}	0000011110iiiiL LLLLLLLLLLLLff011 imm16/imm32 Note 16	Store-memory(sp-4,GR[reg in list12],Word) sp←sp+4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend (imm5) ep←sp/imm	n+2 Note 4	n+2 Note 4	n+2 Note 4						
RETI		000001111100000 0000000101000000	if PSW.EP=1 then PC ←EIPC PSW ←EIPSW else if PSW.NP=1 then PC ←FEPC PSW ←FEPSW else PC ←EIPC PSW ←EIPSW	3	3	3	R	R	R	R	R	R
SAR	reg1,reg2	rrrrr11111RRRRR 0000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×		
	imm5,reg2	rrrrr010101iiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend (imm5)	1	1	1	×	0	×	×		
SASF	cccc,reg2	rrrrr111110cccc 0000001000000000	if conditions are satisfied then GR[reg2]←(GR[reg2]Logically shift left by 1) OR 0000001H else GR[reg2]←(GR[reg2]Logically shift left by 1) OR 0000000H	1	1	1						
SATADD	reg1,reg2	rrrrr000110RRRRR	GR[reg2]←saturated(GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×	
	imm5,reg2	rrrrr010001iiii	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	rrrrr111110cccc 0000000000000000	If conditions are satisfied then GR[reg2]←00000001H else GR[reg2]←0000000H	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 ddddddddddddddd	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]	rrrrr11111RRRRR 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	rrrrr11111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×		
	imm5,reg2	rrrrr010110iiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×		
SHR	reg1,reg2	rrrrr11111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×		
	imm5,reg2	rrrrr010100iiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×		
SLD.B	disp7[ep],reg2	rrrrr0110dddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 9						
SLD.BU	disp4[ep],reg2	rrrrr0000110ddd Note 18	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 9						
SLD.H	disp8[ep],reg2	rrrrr1000dddddd Note 19	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	Note 9						
SLD.HU	disp5[ep],reg2	rrrrr0000111ddd Notes 18, 20	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,Halfword))	1	1	Note 9						
SLD.W	disp8[ep],reg2	rrrrr1010dddddd0 Note 21	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	Note 9						
SST.B	reg2,disp7[ep]	rrrrr0111dddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1						
SST.H	reg2,disp8[ep]	rrrrr1001dddddd Note 19	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Halfword)	1	1	1						
SST.W	reg2,disp8[ep]	rrrrr1010dddddd1 Note 21	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1						
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR ddddddddddddddd	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	rrrrr11111RRRRR 0000000010000000	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	x	x	x	x	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]-GR[reg2]	1	1	1	x	x	x	x	
SWITCH	reg1	0000000010RRRRR	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	0000000101RRRRR	GR[reg1]←sign-extend (GR[reg1] (7 : 0))	1	1	1					
SXH	reg1	0000000111RRRRR	GR[reg1]←sign-extend (GR[reg1] (15 : 0))	1	1	1					
TRAP	vector	000001111111iiii 000000100000000	EIPC ←PC+4 (Restored PC) EIPSW ←PSW ECR.EICC ←Interrupt code PSW.EP ←1 PSW.ID ←1 PC ←00000040H (when vector is 00H to 0FH) 00000050H (when vector is 10H to 1FH)	3	3	3					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1	0	x	x		
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit (adr,bit#3))	3	3	3				x	
	reg2, [reg1]	rrrrr111111RRRRR 000000011100110	adr←GR[reg1] Z flag←Not (Load-memory-bit (adr,reg2))	3	3	3				x	
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1	0	x	x		
XORI	imm16,reg1,reg2	rrrrr110101RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] XOR zero-extend (imm16)	1	1	1	0	x	x		
ZXB	reg1	0000000100RRRRR	GR[reg1]←zero-extend (GR[reg1] (7 : 0))	1	1	1					
ZXH	reg1	0000000110RRRRR	GR[reg1]←zero-extend (GR[reg1] (15 : 0))	1	1	1					

Notes 1. dddddddd: Higher 8 bits of disp9.

2. 3 if there is an instruction that rewrites the contents of the PSW immediately before.

3. If there is no wait state (3 + the number of read access wait states).

4. n is the total number of list12 load registers. (According to the number of wait states. Also, if there are no wait states, n is the total number of list12 registers. If n = 0, same operation as when n = 1)

5. RRRRRR: other than 00000.

6. The lower halfword data only are valid.

7. dddddddddddddddddddd: The higher 21 bits of disp22.

8. dddddddddddddddd: 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. iiii: Lower 5 bits of imm9.

IIII: Higher 4 bits of imm9.

14. Do not specify the same register for general-purpose registers reg1 and reg3.

15. sp/imm: specified by bits 19 and 20 of the sub-opcode.

16. ff = 00: Load sp in ep.

01: Load sign expanded 16-bit immediate data (bits 47 to 32) in ep.

10: Load 16-bit logically left shifted 16-bit immediate data (bits 47 to 32) in ep.

11: Load 32-bit immediate data (bits 63 to 32) in ep.

17. If imm = imm32, n + 3 clocks.

18. rrrrr: Other than 00000.

19. ddddddd: Higher 7 bits of disp8.

20. dddd: Higher 4 bits of disp5.

21. ddddddd: Higher 6 bits of disp8.

REVISION HISTORY	V850ES/JC3-L, V850ES/JE3-L User's Manual: Hardware
------------------	--

Rev.	Date	Description	
		Page	Summary
0.01	Jul 23, 2010	—	First Edition issued

V850ES/JC3-L, V850ES/JE3-L User's Manual: Hardware

Publication Date: Rev.0.01 Jul 23, 2010

Published by: Renesas Electronics Corporation

**SALES OFFICES****Renesas Electronics Corporation**<http://www.renesas.com>Refer to "<http://www.renesas.com/>" for the latest and detailed information.**Renesas Electronics America Inc.**2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130**Renesas Electronics Canada Limited**1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220**Renesas Electronics Europe Limited**Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900**Renesas Electronics Europe GmbH**Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327**Renesas Electronics (China) Co., Ltd.**7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679**Renesas Electronics (Shanghai) Co., Ltd.**Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898**Renesas Electronics Hong Kong Limited**Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044**Renesas Electronics Taiwan Co., Ltd.**7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670**Renesas Electronics Singapore Pte. Ltd.**1 HarbourFront Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001**Renesas Electronics Malaysia Sdn.Bhd.**Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510**Renesas Electronics Korea Co., Ltd.**11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

V850ES/JC3-L, V850ES/JE3-L



Renesas Electronics Corporation

R01UH0018EJ0001