To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

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.

RENESAS

User's Manual

V850ES/HE3, V850ES/HF3, V850ES/HG3, V850ES/HJ3

32-bit Single-Chip Microcontrollers

Hardware

V850ES/HE3:
PD70F3747
V850ES/HF3:
PD70F3750
V850ES/HG3:
PD70F3752
V850ES/HJ3:
PD70F3755
PD70F3757

Document No. U18854EJ2V0UD00 (2nd edition)
Date Published September 2008 N

[MEMO]

NOTES FOR CMOS DEVICES

① VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

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

② HANDLING OF UNUSED INPUT PINS

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

③ PRECAUTION AGAINST ESD

A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

④ STATUS BEFORE INITIALIZATION

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

⑤ POWER ON/OFF SEQUENCE

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

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

⑥ INPUT OF SIGNAL DURING POWER OFF STATE

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

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

IECUBE is a registered trademark of NEC Electronics Corporation in Japan and Germany.
MINICUBE is a registered trademark of NEC Electronics Corporation in Japan and Germany or a trademark in 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.
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 Real-Time Operating system Nucleus.
ITRON is an abbreviation of Industrial TRON.

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

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

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

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

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

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

(Note)

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

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

M8E 02. 11-1

PREFACE

Readers

This manual is intended for users who wish to understand the functions of the V850ES/Hx3 and design application systems using the V850ES/Hx3.

Purpose

This manual is intended to give users an understanding of the hardware functions of the V850ES/Hx3 shown in the **Organization** below.

Organization

This manual is divided into two parts: Hardware (this manual) and Architecture (**V850ES Architecture User's Manual**).

Hardware	Architecture
• Pin functions	• Data types
• CPU function	• Register set
• On-chip peripheral functions	• Instruction format and instruction set
• Flash memory programming	• Interrupts and exceptions
• Electrical specifications	• 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/Hx3
→Read this manual according to the **CONTENTS**.

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

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/HE3
→See **CHAPTER 29 ELECTRICAL SPECIFICATIONS (V850ES/HE3)**.

To know the electrical specifications of the V850ES/HF3
→See **CHAPTER 30 ELECTRICAL SPECIFICATIONS (V850ES/HF3)**.

To know the electrical specifications of the V850ES/HG3
→See **CHAPTER 31 ELECTRICAL SPECIFICATIONS (V850ES/HG3)**.

To know the electrical specifications of the V850ES/HJ3
→See **CHAPTER 32 ELECTRICAL SPECIFICATIONS (V850ES/HJ3)**.

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

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:	$\overline{xxx}$ (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/Hx3

Document Name	Document No.
V850ES Architecture User's Manual	U15943E
V850ES/HE3, V850ES/HF3, V850ES/HG3, V850ES/HJ3 Hardware User's Manual	This manual

Documents related to development tools (user's manuals)

Document Name		Document No.
QB-V850MINI On-Chip Debug Emulator		U17638E
QB-MINI2 On-Chip Debug Emulator with Flash Programming Function		U18371E
CA850 Ver. 3.20 C Compiler Package	Operation	U18512E
	C Language	U18513E
	Assembly Language	U18514E
	Link Directives	U18515E
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 Specification	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-FP4 Flash Memory Programmer		U15260E
PG-FP5 Flash Memory Programmer		U18865E

CONTENTS

CHAPTER 1 INTRODUCTION 21

1.1 General 21
1.2 Features 22
1.2.1 V850ES/HE3 (PD70F3747) 22
1.2.2 V850ES/HF3 (PD70F3750) 23
1.2.3 V850ES/HG3 (PD70F3752) 24
1.2.4 V850ES/HJ3 (PD70F3755, 70F3757) 25
1.3 Application Fields 26
1.4 Ordering Information 26
1.5 Pin Configuration (Top View) 27
1.5.1 V850ES/HE3 (PD70F3747) 27
1.5.2 V850ES/HF3 (PD70F3750) 28
1.5.3 V850ES/HG3 (PD70F3752) 29
1.5.4 V850ES/HJ3 (PD70F3755, 70F3757) 30
1.6 Function Block Configuration 32
1.6.1 V850ES/HE3 (PD70F3747) 32
1.6.2 V850ES/HF3 (PD70F3750) 35
1.6.3 V850ES/HG3 (PD70F3752) 38
1.6.4 V850ES/HJ3 (PD70F3755, 70F3757) 41
1.7 Overview of Functions 44

CHAPTER 2 PIN FUNCTIONS 45

2.1 Pin Function List 45
2.2 Pin Status 57
2.3 Pin I/O Circuit Types and Recommended Connection of Unused Pins 58
2.4 Pin I/O Circuits 62

CHAPTER 3 CPU FUNCTION 64

3.1 Features 64
3.2 CPU Register Set 65
3.2.1 Program register set 66
3.2.2 System register set 67
3.3 Operation Modes 73
3.4 Address Space 74
3.4.1 CPU address space 74
3.4.2 Memory map 75
3.4.3 Areas 77
3.4.4 Wraparound of data space 81
3.4.5 Recommended use of address space 81
3.4.6 Peripheral I/O registers 84
3.4.7 Special registers 97
3.4.8 Cautions 101

CHAPTER 4 PORT FUNCTIONS 104

4.1 Features 104

4.1.1 V850ES/HE3 104
4.1.2 V850ES/HF3 104
4.1.3 V850ES/HG3 104
4.1.4 V850ES/HJ3 104
4.2 Basic Configuration of Ports 105
4.2.1 V850ES/HE3 105
4.2.2 V850ES/HF3 106
4.2.3 V850ES/HG3 107
4.2.4 V850ES/HJ3 108
4.3 Port Configuration 109
4.3.1 Port 0 114
4.3.2 Port 1 118
4.3.3 Port 3 120
4.3.4 Port 4 128
4.3.5 Port 5 131
4.3.6 Port 6 (V850ES/HJ3 only) 135
4.3.7 Port 7 140
4.3.8 Port 8 (V850ES/HJ3 only) 144
4.3.9 Port 9 146
4.3.10 Port 12 (V850ES/HJ3 only) 165
4.3.11 Port CD (V850ES/HJ3 only) 167
4.3.12 Port CM 168
4.3.13 Port CS 172
4.3.14 Port CT 176
4.3.15 Port DL 180
4.4 Block Diagrams of Port 185
4.5 Port Register Settings When Alternate Function Is Used 223
4.6 Cautions 233
4.6.1 Cautions on setting port pins 233
4.6.2 Cautions on bit manipulation instruction for port n register (Pn) 236
4.6.3 Cautions on on-chip debug pins 237
4.6.4 Cautions on P05/INTP2/DRST# pin 237
4.6.5 Cautions on P53 pin when power is turned on 237
4.6.6 Hysteresis characteristics 237

CHAPTER 5 BUS CONTROL FUNCTION (V850ES/HJ3 ONLY) 238
5.1 Features 238
5.2 Bus Control Pins 239
5.2.1 Pin status when internal ROM, internal RAM, or on-chip peripheral I/O is accessed 239
5.2.2 Pin status in each operation mode 239
5.3 Memory Block Function 240
5.4 Bus Access 241
5.4.1 Number of clocks for access 241
5.4.2 Bus size setting function 242
5.4.3 Access by bus size 243
5.5 Wait Function 250
5.5.1 Programmable wait function 250
5.5.2 External wait function 251

5.5.3 Relationship between programmable wait and external wait 251
5.5.4 Programmable address wait function 252
5.6 Idle State Insertion Function 253
5.7 Bus Hold Function 254
5.7.1 Functional outline 254
5.7.2 Bus hold procedure 255
5.7.3 Operation in power save mode 255
5.8 Bus Priority 256
5.9 Bus Timing 257

CHAPTER 6 CLOCK GENERATION FUNCTION 260
6.1 Overview 260
6.2 Configuration 261
6.3 Registers 264
6.4 Operation 273
6.4.1 Setting CPU clock 273
6.4.2 Operation of each clock 274
6.4.3 Clock output function 275
6.5 PLL/SSCG Function 276
6.5.1 Overview 276
6.5.2 Registers 276
6.5.3 Using PLL 283
6.5.4 Using SSCG 285

CHAPTER 7 16-BIT TIMER/EVENT COUNTER AA (TAA) 288
7.1 Overview 288
7.2 Functions 288
7.3 Configuration 289
7.4 Registers 291
7.5 Operation 305
7.5.1 Interval timer mode (TAAnMD2 to TAAnMD0 bits = 000) 306
7.5.2 External event count mode (TAAnMD2 to TAAnMD0 bits = 001) 315
7.5.3 External trigger pulse output mode (TAAnMD2 to TAAnMD0 bits = 010) 323
7.5.4 One-shot pulse output mode (TAAnMD2 to TAAnMD0 bits = 011) 335
7.5.5 PWM output mode (TAAnMD2 to TAAnMD0 bits = 100) 342
7.5.6 Free-running timer mode (TAAnMD2 to TAAnMD0 bits = 101) 351
7.5.7 Pulse width measurement mode (TAAnMD2 to TAAnMD0 bits = 110) 368
7.5.8 Timer output operations 374
7.6 Timer-Tuned Operation Function 375
7.6.1 Free-running timer mode (during timer-tuned operation) 377
7.6.2 PWM output mode (during timer-tuned operation) 384
7.6.3 Triangular PWM output mode (during timer-tuned operation) (V850ES/HJ3 only) 386
7.7 Cascade Connection 388
7.8 Selector Function 393
7.9 Cautions 395

CHAPTER 8 16-BIT TIMER/EVENT COUNTER AB (TAB) 396
8.1 Overview 396

8.2 Functions ... 396
8.3 Configuration ... 397
8.4 Registers ... 399
8.5 Operation ... 416
8.5.1 Interval timer mode (TABnMD2 to TABnMD0 bits = 000) ... 417
8.5.2 External event count mode (TABnMD2 to TABnMD0 bits = 001) ... 426
8.5.3 External trigger pulse output mode (TABnMD2 to TABnMD0 bits = 010) ... 435
8.5.4 One-shot pulse output mode (TABnMD2 to TABnMD0 bits = 011) ... 448
8.5.5 PWM output mode (TABnMD2 to TABnMD0 bits = 100) ... 457
8.5.6 Free-running timer mode (TABnMD2 to TABnMD0 bits = 101) ... 468
8.5.7 Pulse width measurement mode (TABnMD2 to TABnMD0 bits = 110) ... 488
8.5.8 Triangular wave PWM mode (TABnMD2 to TABnMD0 = 111) ... 494
8.5.9 Timer output operations ... 495
8.6 Timer-Tuned Operation Function ... 496
8.7 Cautions ... 497

CHAPTER 9 16-BIT INTERVAL TIMER M (TMM) ... 498
9.1 Overview ... 498
9.2 Configuration ... 499
9.3 Register ... 500
9.4 Operation ... 501
9.4.1 Interval timer mode ... 501
9.4.2 Cautions ... 505

CHAPTER 10 MOTOR CONTROL FUNCTION ... 506
10.1 Functional Overview ... 506
10.2 Configuration ... 507
10.3 Control Registers ... 511
10.4 Operation ... 521
10.4.1 System outline ... 521
10.4.2 Dead-time control (generation of negative-phase wave signal) ... 526
10.4.3 Interrupt culling function ... 533
10.4.4 Operation to rewrite register with transfer function ... 540
10.4.5 TAA4 tuning operation for A/D conversion start trigger signal output ... 558
10.4.6 A/D conversion start trigger output function ... 561

CHAPTER 11 WATCH TIMER FUNCTIONS ... 566
11.1 Functions ... 566
11.2 Configuration ... 567
11.3 Registers ... 569
11.4 Operation ... 573
11.4.1 Operation as watch timer ... 573
11.4.2 Operation as interval timer ... 574
11.4.3 Cautions ... 575

CHAPTER 12 FUNCTIONS OF WATCHDOG TIMER 2 ... 576
12.1 Functions ... 576
12.2 Configuration ... 577

12.3 Registers 578
12.4 Operation 581

CHAPTER 13 A/D CONVERTER 582

13.1 Overview 582
13.2 Functions 582
13.3 Configuration 583
13.4 Registers 586
13.5 Operation 594
13.5.1 Basic operation 594
13.5.2 Trigger mode 595
13.5.3 Operation mode 597
13.5.4 Power-fail compare mode 601
13.6 Cautions 606
13.7 How to Read A/D Converter Characteristics Table 610

CHAPTER 14 ASYNCHRONOUS SERIAL INTERFACE D (UARTD) 614

14.1 Features 615
14.2 Configuration 616
14.3 Assignment of Serial Interface (PD70F3757 Only) 618
14.3.1 To use UARTD3 and CSIB0 at the same time 618
14.3.2 Mode switching between UARTD5 and I^2C00 619
14.3.3 Mode switching between UARTD5 and CSIB2 620
14.4 Registers 621
14.5 Interrupt Request Signals 631
14.6 Operation 632
14.6.1 Data format 632
14.6.2 SBF transmission/reception format 634
14.6.3 SBF transmission 636
14.6.4 SBF reception 637
14.6.5 Data consistency check function 640
14.6.6 Selecting SBF reception mode 642
14.6.7 UART transmission 650
14.6.8 Continuous transmission procedure 651
14.6.9 UART reception 653
14.6.10 Procedure of processing in case of reception interrupt 654
14.6.11 Parity types and operations 658
14.6.12 Receive data noise filter 659
14.7 Dedicated Baud Rate Generator 660
14.8 Cautions 668

CHAPTER 15 3-WIRE VARIABLE-LENGTH SERIAL I/O (CSIB) 670

15.1 Features 670
15.2 Configuration 671
15.3 Mode Switching of CSIB and Other Serial Interfaces 672
15.3.1 CSIB0 and UARTD3 mode switching 672
15.3.2 CSIB2 and UARTD5 mode switching 673
15.4 Registers 674

15.5 Interrupt Request Signals 682
15.6 Operation 683
15.6.1 Single transfer mode (master mode, transmission mode) 683
15.6.2 Single transfer mode (master mode, reception mode) 685
15.6.3 Single transfer mode (master mode, transmission/reception mode) 687
15.6.4 Single transfer mode (slave mode, transmission mode) 689
15.6.5 Single transfer mode (slave mode, reception mode) 691
15.6.6 Single transfer mode (slave mode, transmission/reception mode) 693
15.6.7 Continuous transfer mode (master mode, transmission mode) 695
15.6.8 Continuous transfer mode (master mode, reception mode) 697
15.6.9 Continuous transfer mode (master mode, transmission/reception mode) 700
15.6.10 Continuous transfer mode (slave mode, transmission mode) 703
15.6.11 Continuous transfer mode (slave mode, reception mode) 705
15.6.12 Continuous transfer mode (slave mode, transmission/reception mode) 708
15.6.13 Reception error 711
15.6.14 Clock timing 712
15.7 Output Pin Status with Operation Disabled 714
15.8 Baud Rate Generator 715
15.8.1 Baud rate generation 716
15.9 Cautions 716

CHAPTER 16 I^2C BUS 717
16.1 Mode Switching of I^2C Bus and Other Serial Interfaces 717
16.1.1 UARTD4 and I^2C00 mode switching 717
16.2 Features 718
16.3 Configuration 721
16.4 Registers 723
16.5 Functions 738
16.5.1 Pin configuration 738
16.6 I^2C Bus Definitions and Control Methods 739
16.6.1 Start condition 739
16.6.2 Addresses 740
16.6.3 Transfer direction specification 741
16.6.4 ACK# 742
16.6.5 Stop condition 743
16.6.6 Wait state 744
16.6.7 Wait state cancellation method 746
16.7 I^2C Interrupt Request Signals (INTIIC0) 747
16.7.1 Master device operation 748
16.7.2 Slave device operation (when receiving slave address data (address match)) 751
16.7.3 Slave device operation (when receiving extension code) 755
16.7.4 Operation without communication 759
16.7.5 Arbitration loss operation (operation as slave after arbitration loss) 760
16.7.6 Operation when arbitration loss occurs (no communication after arbitration loss) 762
16.8 Interrupt Request Signal (INTIIC0) Generation Timing and Wait Control 769
16.9 Address Match Detection Method 770
16.10 Error Detection 770
16.11 Extension Code 770

16.12 Arbitration 771
16.13 Wakeup Function 772
16.14 Communication Reservation 773
16.14.1 When communication reservation function is enabled (IICF0.IICRSV0 bit = 0) 773
16.14.2 When communication reservation function is disabled (IICF0.IICRSV0 bit = 1) 777
16.15 Cautions 778
16.16 Communication Operations 779
16.16.1 Master operation in single master system 780
16.16.2 Master operation in multimaster system 781
16.16.3 Slave operation 784
16.17 Timing of Data Communication 787

CHAPTER 17 DMA FUNCTION (DMA CONTROLLER) 794
17.1 Features 794
17.2 Configuration 795
17.3 Registers 796
17.4 Transfer Targets 804
17.5 Transfer Modes 804
17.6 Transfer Types 805
17.7 DMA Channel Priorities 806
17.8 Time Related to DMA Transfer 806
17.9 DMA Transfer Start Factors 807
17.10 DMA Abort Factors 808
17.11 End of DMA Transfer 808
17.12 Operation Timing 808
17.13 Cautions 813

CHAPTER 18 INTERRUPT/EXCEPTION PROCESSING FUNCTION 818
18.1 Features 818
18.2 Non-Maskable Interrupts 829
18.2.1 Operation 831
18.2.2 Restore 832
18.2.3 NP flag 833
18.3 Maskable Interrupts 834
18.3.1 Operation 834
18.3.2 Restore 836
18.3.3 Priorities of maskable interrupts 837
18.3.4 Interrupt control register (xxICn) 841
18.3.5 Interrupt mask registers 0 to 5 (IMR0 to IMR5) 844
18.3.6 In-service priority register (ISPR) 848
18.3.7 ID flag 849
18.3.8 Watchdog timer mode register 2 (WDTM2) 849
18.4 Software Exception 850
18.4.1 Operation 850
18.4.2 Restore 851
18.4.3 EP flag 852
18.5 Exception Trap 853
18.5.1 Illegal opcode definition 853

18.5.2 Debug trap 855
18.6 External Interrupt Request Input Pins (NMI and INTP0 to INTP14) 857
18.6.1 Noise elimination 857
18.6.2 Edge detection 857
18.7 Interrupt Acknowledge Time of CPU 867
18.8 Periods in Which Interrupts Are Not Acknowledged by CPU 868
18.9 Cautions 868

CHAPTER 19 KEY INTERRUPT FUNCTION 869
19.1 Function 869
19.2 Register 870
19.3 Cautions 870

CHAPTER 20 STANDBY FUNCTION 871
20.1 Overview 871
20.2 Registers 873
20.3 HALT Mode 878
20.3.1 Setting and operation status 878
20.3.2 Releasing HALT mode 878
20.4 IDLE1 Mode 880
20.4.1 Setting and operation status 880
20.4.2 Releasing IDLE1 mode 880
20.5 IDLE2 Mode 882
20.5.1 Setting and operation status 882
20.5.2 Releasing IDLE2 mode 882
20.5.3 Securing setup time when releasing IDLE2 mode 884
20.6 STOP Mode 885
20.6.1 Setting and operation status 885
20.6.2 Releasing STOP mode 885
20.6.3 Securing oscillation stabilization time when releasing STOP mode 887
20.7 Subclock Operation Mode 888
20.7.1 Setting and operation status 888
20.7.2 Releasing subclock operation mode 888
20.8 Sub-IDLE Mode 890
20.8.1 Setting and operation status 890
20.8.2 Releasing sub-IDLE mode 891

CHAPTER 21 RESET FUNCTIONS 893
21.1 Overview 893
21.2 Registers to Check Reset Source 894
21.3 Operation 895
21.3.1 Reset operation via RESET# pin 895
21.3.2 Reset operation by watchdog timer 2 897
21.3.3 Reset operation by power-on clear circuit 898
21.3.4 Reset operation by low-voltage detector 898
21.3.5 Reset operation by clock monitor 898
21.4 Operation After Reset Release 899

CHAPTER 22 CLOCK MONITOR 901

22.1 Functions 901
22.2 Configuration 901
22.3 Register 902
22.4 Operation 903

CHAPTER 23 POWER-ON CLEAR CIRCUIT 906

23.1 Function 906
23.2 Configuration 906
23.3 Operation 907

CHAPTER 24 LOW-VOLTAGE DETECTOR 908

24.1 Functions 908
24.2 Configuration 908
24.3 Registers 909
24.4 Operation 911
24.4.1 To use for internal reset signal 911
24.4.2 To use for interrupt 913
24.5 RAM Retention Voltage Detection Operation 914
24.6 Emulation Function 915

CHAPTER 25 REGULATOR 916

25.1 Overview 916
25.2 Operation 918

CHAPTER 26 FLASH MEMORY 919

26.1 Features 919
26.2 Memory Configuration 920
26.3 Functional Outline 921
26.4 Rewriting by Dedicated Flash Programmer 924
26.4.1 Programming environment 924
26.4.2 Communication mode 925
26.4.3 Flash memory control 939
26.4.4 Selection of communication mode 940
26.4.5 Communication commands 941
26.4.6 Pin connection 942
26.5 Rewriting by Self Programming 946
26.5.1 Overview 946
26.5.2 Features 947
26.5.3 Standard self programming flow 949
26.5.4 Flash functions 950
26.5.5 Pin processing 950
26.5.6 Internal resources used 951

CHAPTER 27 OPTION BYTE FUNCTION 952

CHAPTER 28 ON-CHIP DEBUG FUNCTION 954

28.1 Debugging with DCU 955

28.1.1 Connection circuit example 955
28.1.2 Interface signals 955
28.1.3 Maskable functions 957
28.1.4 Register 957
28.1.5 Operation 959
28.1.6 Cautions 959
28.2 Debugging Without Using DCU 960
28.2.1 Circuit connection examples 960
28.2.2 Maskable functions 962
28.2.3 Securement of user resources 962
28.2.4 Cautions 968
28.3 ROM Security Function 969
28.3.1 Security ID 969
28.3.2 Setting 970

CHAPTER 29 ELECTRICAL SPECIFICATIONS (V850ES/HE3) 971
29.1 Absolute Maximum Ratings 971
29.2 Capacitance 973
29.3 Operating Conditions 973
29.4 Oscillator Characteristics 974
29.4.1 Main clock oscillator characteristics 974
29.4.2 Subclock oscillator characteristics 976
29.4.3 PLL characteristics 977
29.4.4 SSCG characteristics 977
29.4.5 Low-speed internal oscillator/high-speed internal oscillator characteristics 977
29.5 Voltage Regulator Characteristics 978
29.6 DC Characteristics 979
29.6.1 I/O level 979
29.6.2 Pin leakage current 980
29.6.3 Supply current 981
29.7 AC Characteristics 982
29.7.1 CLKOUT output timing 983
29.8 Basic Operation 984
29.9 Flash Memory Programming Characteristics 992

CHAPTER 30 ELECTRICAL SPECIFICATIONS (V850ES/HF3) 993
30.1 Absolute Maximum Ratings 993
30.2 Capacitance 995
30.3 Operating Conditions 995
30.4 Oscillator Characteristics 996
30.4.1 Main clock oscillator characteristics 996
30.4.2 Subclock oscillator characteristics 998
30.4.3 PLL characteristics 999
30.4.4 SSCG characteristics 999
30.4.5 Low-speed internal oscillator/high-speed internal oscillator characteristics 999
30.5 Voltage Regulator Characteristics 1000
30.6 DC Characteristics 1001
30.6.1 I/O level 1001

30.6.2 Pin leakage current ... 1002
30.6.3 Supply current ... 1003
30.7 AC Characteristics ... 1004
30.7.1 CLKOUT output timing ... 1005
30.8 Basic Operation ... 1006
30.9 Flash Memory Programming Characteristics ... 1014

CHAPTER 31 ELECTRICAL SPECIFICATIONS (V850ES/HG3) ... 1015
31.1 Absolute Maximum Ratings ... 1015
31.2 Capacitance ... 1017
31.3 Operating Conditions ... 1017
31.4 Oscillator Characteristics ... 1018
31.4.1 Main clock oscillator characteristics ... 1018
31.4.2 Subclock oscillator characteristics ... 1020
31.4.3 PLL characteristics ... 1021
31.4.4 SSCG characteristics ... 1021
31.4.5 Low-speed internal oscillator/high-speed internal oscillator characteristics ... 1021
31.5 Voltage Regulator Characteristics ... 1022
31.6 DC Characteristics ... 1023
31.6.1 I/O level ... 1023
31.6.2 Pin leakage current ... 1024
31.6.3 Supply current ... 1025
31.7 AC Characteristics ... 1026
31.7.1 CLKOUT output timing ... 1027
31.8 Basic Operation ... 1028
31.9 Flash Memory Programming Characteristics ... 1036

CHAPTER 32 ELECTRICAL SPECIFICATIONS (V850ES/HJ3) ... 1037
32.1 Absolute Maximum Ratings ... 1037
32.2 Capacitance ... 1039
32.3 Operating Conditions ... 1039
32.4 Oscillator Characteristics ... 1040
32.4.1 Main clock oscillator characteristics ... 1040
32.4.2 Subclock oscillator characteristics ... 1042
32.4.3 PLL characteristics ... 1043
32.4.4 SSCG characteristics ... 1043
32.4.5 Low-speed internal oscillator/high-speed internal oscillator characteristics ... 1043
32.5 Voltage Regulator Characteristics ... 1044
32.6 DC Characteristics ... 1045
32.6.1 I/O level ... 1045
32.6.2 Pin leakage current ... 1046
32.6.3 Supply current ... 1047
32.7 AC Characteristics ... 1049
32.7.1 CLKOUT output timing ... 1050
32.7.2 Bus timing ... 1051
32.8 Basic Operation ... 1056
32.9 Flash Memory Programming Characteristics ... 1064

CHAPTER 33 PACKAGE DRAWINGS 1065

<R> **CHAPTER 34 RECOMMENDED SOLDERING CONDITIONS 1069**

APPENDIX A DEVELOPMENT TOOLS 1070

A.1 Software Package 1072
A.2 Language Processing Software 1072
A.3 Control Software 1072
A.4 Debugging Tools (Hardware) 1073
A.4.1 When using IECUBE QB-V850ESFX3 1073
A.4.2 When using MINICUBE QB-V850MINI 1076
A.4.3 When using MINICUBE2 QB-MINI2 1077
A.5 Debugging Tools (Software) 1078
A.6 Embedded Software 1079
A.7 Flash Memory Writing Tools 1080

APPENDIX B REGISTER INDEX 1081

APPENDIX C INSTRUCTION SET LIST 1093

C.1 Conventions 1093
C.2 Instruction Set (in Alphabetical Order) 1096

<R> **APPENDIX D REVISION HISTORY 1103**

D.1 Major Revisions in This Edition 1103

CHAPTER 1 INTRODUCTION

The V850ES/Hx3 is one of the products in the NEC Electronics V850 single-chip microcontrollers designed for real-time control applications.

1.1 General

The V850ES/Hx3 is a 32-bit single-chip microcontroller that includes the V850ES CPU core and peripheral functions such as ROM/RAM, a timer/counter, serial interfaces, and an A/D converter.

In addition to high real-time response characteristics and 1-clock-pitch basic instructions, the V850ES/Hx3 features multiply instructions realized by a hardware multiplier, saturated operation instructions, bit manipulation instructions, etc., as optimum instructions for digital servo control applications.

1.2 Features

1.2.1 V850ES/HE3 (μPD70F3747)

- ○ Minimum instruction execution time: 31.25 ns (operating with main clock (f_{XX}) of 32 MHz)
- ○ General-purpose registers: 32 bits × 32 registers
- ○ CPU features:
  - Signed multiplication (16 × 16 → 32): 1 to 2 clocks
  - Signed multiplication (32 × 32 → 64): 1 to 5 clocks
  - Saturated operations (overflow and underflow detection functions included)
  - 32-bit shift instruction: 1 clock
  - 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 KB
    - Flash memory: 128 KB
- ○ Interrupts and exceptions:
  - Non-maskable interrupts: 2 sources (external: 1, internal: 1)
  - Maskable interrupts: 50 sources (external: 8, internal: 42)
  - Software exceptions: 32 sources
  - Exception trap: 2 sources
- ○ I/O lines: I/O ports: 51
- ○ Timer function:
  - 16-bit interval timer M (TMM): 1 channel
  - 16-bit timer/event counter AA (TAA): 5 channels
  - 16-bit timer/event counter AB (TAB): 1 channel
  - Watch timer: 1 channel
  - Watchdog timer 2: 1 channel
- ○ Serial interface:
  - Asynchronous serial interface D (UARTD): 2 channels
  - 3-wire variable-length serial interface B (CSIB): 2 channels
  - I^2C bus: 1 channel
- ○ A/D converter: 10-bit resolution: 10 channels
- ○ DMA controller: 4 channels (transfer target: on-chip peripheral I/O, internal RAM)
- ○ 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 (×8)/SSCG mode selectable
- ○ Low-speed internal oscillation clock (f_{RL}): 240 kHz (TYP.)
- ○ High-speed internal oscillation clock (f_{RH}): 8 MHz (TYP.)
- ○ Power-save functions: HALT/IDLE1/IDLE2/STOP/subclock/sub-IDLE mode
- ○ Package: 64-pin plastic LQFP (fine pitch) (10 × 10)

1.2.2 V850ES/HF3 (μPD70F3750)

- ○ Minimum instruction execution time: 31.25 ns (operating with main clock (f_{XX}) of 32 MHz)
- ○ General-purpose registers: 32 bits × 32 registers
- ○ CPU features:
  - Signed multiplication (16 × 16 → 32): 1 to 2 clocks
  - Signed multiplication (32 × 32 → 64): 1 to 5 clocks
  - Saturated operations (overflow and underflow detection functions included)
  - 32-bit shift instruction: 1 clock
  - 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: 16 KB
    - Flash memory: 256 KB
- ○ Interrupts and exceptions:
  - Non-maskable interrupts: 2 sources (external: 1, internal: 1)
  - Maskable interrupts: 50 sources (external: 8, internal: 42)
  - Software exceptions: 32 sources
  - Exception trap: 2 sources
- ○ I/O lines: I/O ports: 67
- ○ Timer function:
  - 16-bit interval timer M (TMM): 1 channel
  - 16-bit timer/event counter AA (TAA): 5 channels
  - 16-bit timer/event counter AB (TAB): 1 channel
  - Watch timer: 1 channel
  - Watchdog timer 2: 1 channel
- ○ Serial interface:
  - Asynchronous serial interface D (UARTD): 2 channels
  - 3-wire variable-length serial interface B (CSIB): 2 channels
  - I^2C bus: 1 channel
- ○ A/D converter: 10-bit resolution: 12 channels
- ○ DMA controller: 4 channels (transfer target: on-chip peripheral I/O, internal RAM)
- ○ 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 (×8)/SSCG mode selectable
- ○ Low-speed internal oscillation clock (f_{RL}): 240 kHz (TYP.)
- ○ High-speed internal oscillation clock (f_{RH}): 8 MHz (TYP.)
- ○ Power-save functions: HALT/IDLE1/IDLE2/STOP/subclock/sub-IDLE mode
- ○ Package: 80-pin plastic LQFP (fine pitch) (12 × 12)

1.2.3 V850ES/HG3 (μPD70F3752)

- ○ Minimum instruction execution time: 31.25 ns (operating with main clock (f_{XX}) of 32 MHz)
- ○ General-purpose registers: 32 bits × 32 registers
- ○ CPU features:
  - Signed multiplication (16 × 16 → 32): 1 to 2 clocks
  - Signed multiplication (32 × 32 → 64): 1 to 5 clocks
  - Saturated operations (overflow and underflow detection functions included)
  - 32-bit shift instruction: 1 clock
  - 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: 16 KB
    - Flash memory: 256 KB
- ○ Interrupts and exceptions:
  - Non-maskable interrupts: 2 sources (external: 1, internal: 1)
  - Maskable interrupts: 61 sources (external: 11, internal: 50)
  - Software exceptions: 32 sources
  - Exception trap: 2 sources
- ○ I/O lines: I/O ports: 84
- ○ Timer function:
  - 16-bit interval timer M (TMM): 1 channel
  - 16-bit timer/event counter AA (TAA): 5 channels
  - 16-bit timer/event counter AB (TAB): 2 channels
  - Watch timer: 1 channel
  - Watchdog timer 2: 1 channel
- ○ Serial interface:
  - Asynchronous serial interface D (UARTD): 3 channels
  - 3-wire variable-length serial interface B (CSIB): 2 channels
  - I^2C bus: 1 channel
- ○ A/D converter: 10-bit resolution: 16 channels
- ○ DMA controller: 4 channels (transfer target: on-chip peripheral I/O, internal RAM)
- ○ 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 (×8)/SSCG mode selectable
- ○ Low-speed internal oscillation clock (f_{RL}): 240 kHz (TYP.)
- ○ High-speed internal oscillation clock (f_{RH}): 8 MHz (TYP.)
- ○ Power-save functions: HALT/IDLE1/IDLE2/STOP/subclock/sub-IDLE mode
- ○ Package: 100-pin plastic LQFP (fine pitch) (14 × 14)

1.2.4 V850ES/HJ3 (μPD70F3755, 70F3757)

- ○ Minimum instruction execution time: 31.25 ns (operating with main clock (f_{XX}) of 32 MHz)
- ○ General-purpose registers: 32 bits × 32 registers
- ○ CPU features:
  - Signed multiplication (16 × 16 → 32): 1 to 2 clocks
  - Signed multiplication (32 × 32 → 64): 1 to 5 clocks
  - Saturated operations (overflow and underflow detection functions included)
  - 32-bit shift instruction: 1 clock
  - 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: 16 KB (μPD70F3755)/32 KB (μPD70F3757)
    - Flash memory: 256 KB (μPD70F3755)/512 KB (μPD70F3757)
  - External bus interface:
    - External expansion: Up to 256 KB
      - Multiplexed bus
      - 8-/16-bit data bus sizing function
      - Wait function
        - Programmable wait function
        - External wait function
      - Idle state function
      - Bus hold function
- ○ Interrupts and exceptions:
  - Non-maskable interrupts: 2 sources (external: 1, internal: 1)
  - Maskable interrupts: 72 sources (external: 15, internal: 57) (μPD70F3755)
    78 sources (external: 15, internal: 63) (μPD70F3757)
  - Software exceptions: 32 sources
  - Exception trap: 2 sources
- ○ I/O lines: I/O ports: 128
- ○ Timer function:
  - 16-bit interval timer M (TMM): 1 channel
  - 16-bit timer/event counter AA (TAA): 5 channels
  - 16-bit timer/event counter AB (TAB): 3 channels
  - Watch timer: 1 channel
  - Watchdog timer 2: 1 channel
- ○ Serial interface:
  - Asynchronous serial interface D (UARTD): 3 channels (μPD70F3755)
    6 channels (μPD70F3757)
  - 3-wire variable-length serial interface B (CSIB): 3 channels
  - I^2C bus: 1 channel
- ○ A/D converter: 10-bit resolution: 24 channels
- ○ DMA controller: 4 channels (transfer target: on-chip peripheral I/O, internal RAM, external memory)
- ○ 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 (×8)/SSCG mode selectable
- ○ Low-speed internal oscillation clock (f_{RL}): 240 kHz (TYP.)
- ○ High-speed internal oscillation clock (f_{RH}): 8 MHz (TYP.)
- ○ Power-save functions: HALT/IDLE1/IDLE2/STOP/subclock/sub-IDLE mode
- ○ Package: 144-pin plastic LQFP (fine pitch) (20 × 20)

1.3 Application Fields

Industrial equipment, air-conditioning-/housing-related equipment, measurement equipment, and consumer electronics

1.4 Ordering Information

- V850ES/HE3

Part Number	Package	On-Chip Flash Memory
μPD70F3747GB-GAH-AX	64-pin plastic LQFP (fine pitch) (10 × 10)	128 KB

- V850ES/HF3

Part Number	Package	On-Chip Flash Memory
μPD70F3750GK-GAK-AX	80-pin plastic LQFP (fine pitch) (12 × 12)	256 KB

- V850ES/HG3

Part Number	Package	On-Chip Flash Memory
μPD70F3752GC-UEU-AX	100-pin plastic LQFP (fine pitch) (14 × 14)	256 KB

- V850ES/HJ3

Part Number	Package	On-Chip Flash Memory
μPD70F3755GJ-GAE-AX	144-pin plastic LQFP (fine pitch) (20 × 20)	256 KB
μPD70F3757GJ-GAE-AX	144-pin plastic LQFP (fine pitch) (20 × 20)	512 KB

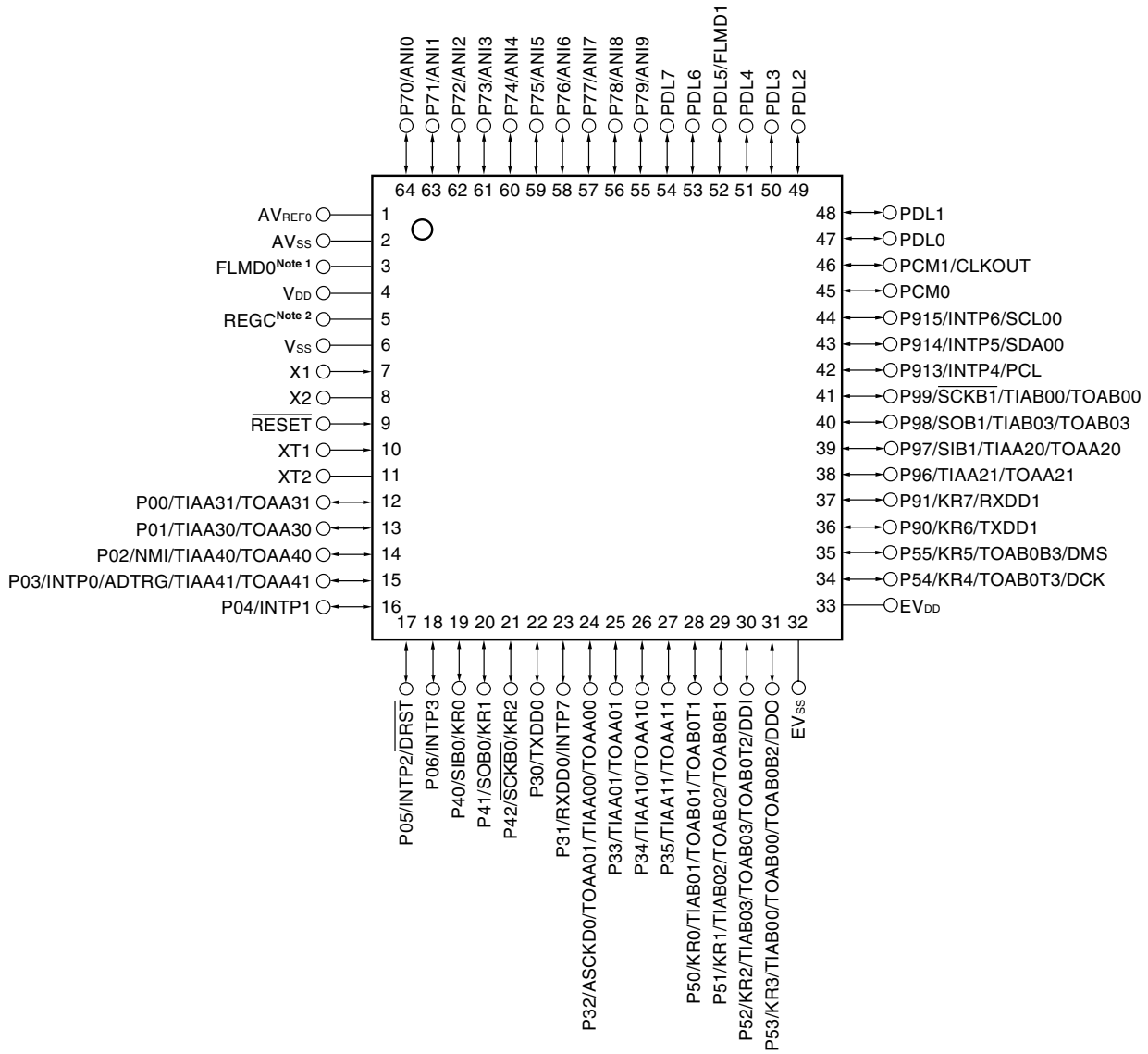
Remark Products with -AX at the end of the part number are lead-free products.

1.5 Pin Configuration (Top View)

1.5.1 V850ES/HE3 (μPD70F3747)

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

μPD70F3747GB-GAH-AX



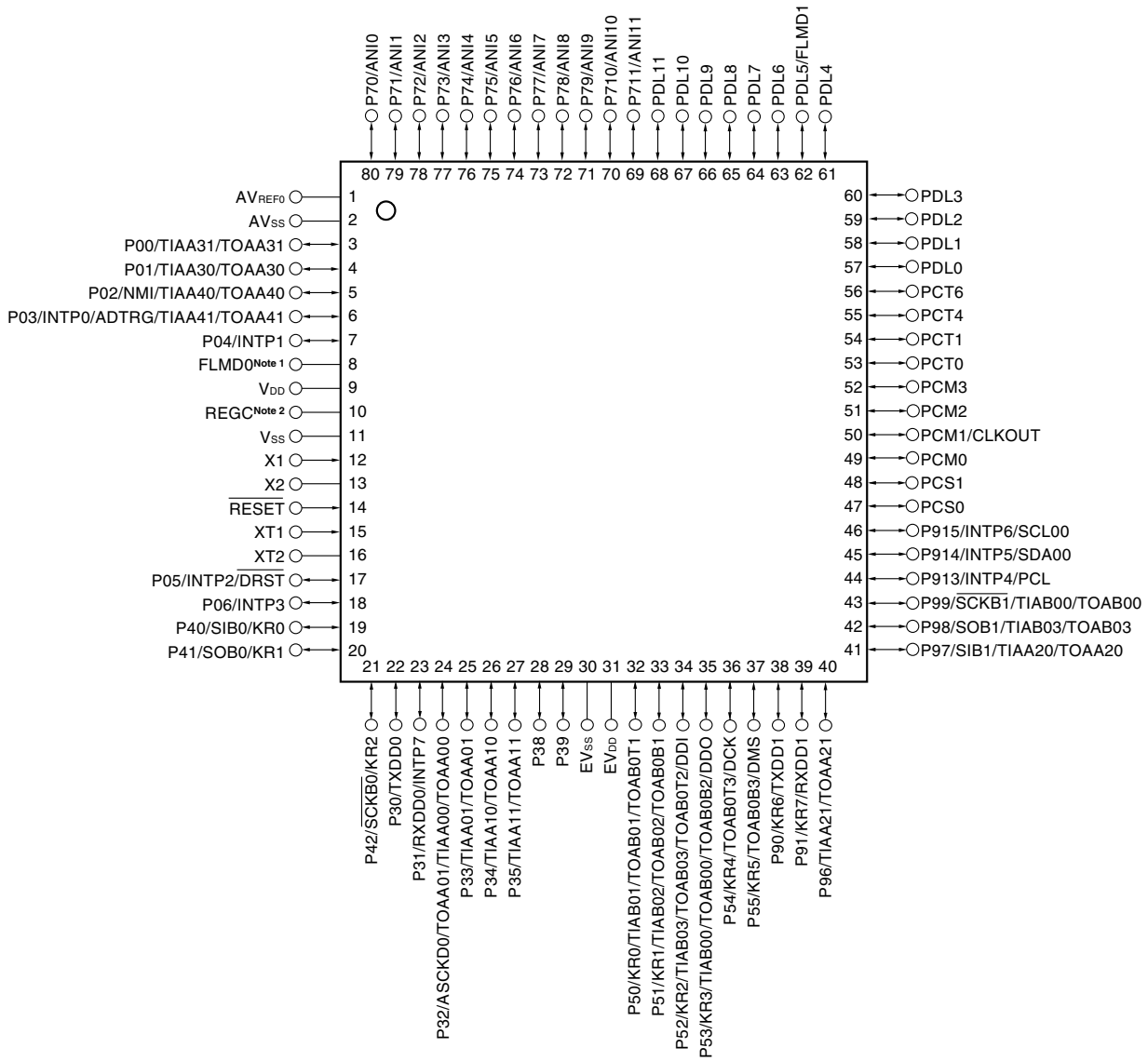
Notes 1. Connect this pin to VSS in the normal mode.

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

1.5.2 V850ES/HF3 (μPD70F3750)

80-pin plastic LQFP (fine pitch) (12 × 12)

μPD70F3750GK-GAK-AX



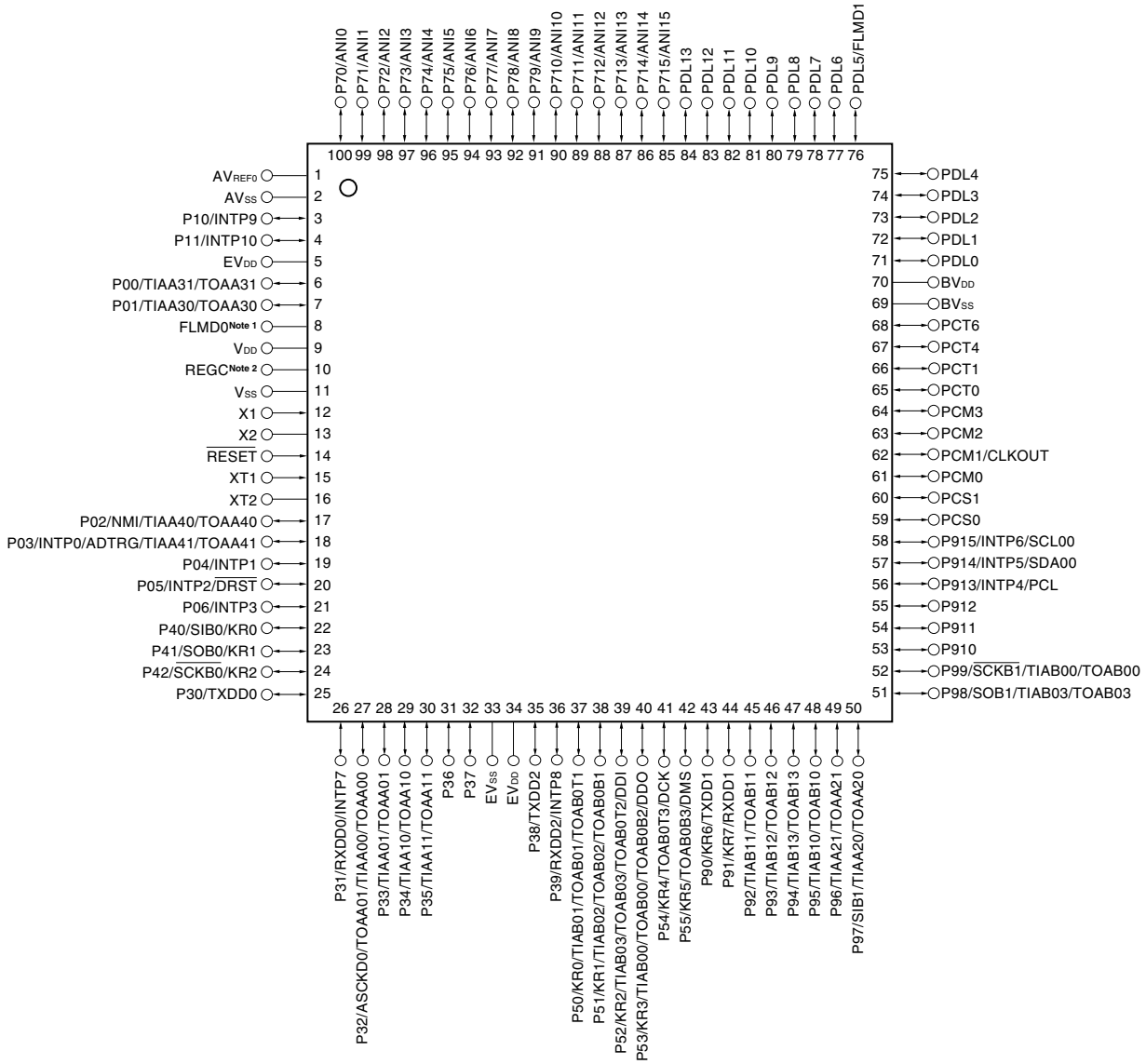
Notes 1. Connect this pin to VSS in the normal mode.

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

1.5.3 V850ES/HG3 (μPD70F3752)

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

μPD70F3752GC-UEU-AX



Notes 1. Connect this pin to V_{SS} in the normal mode.

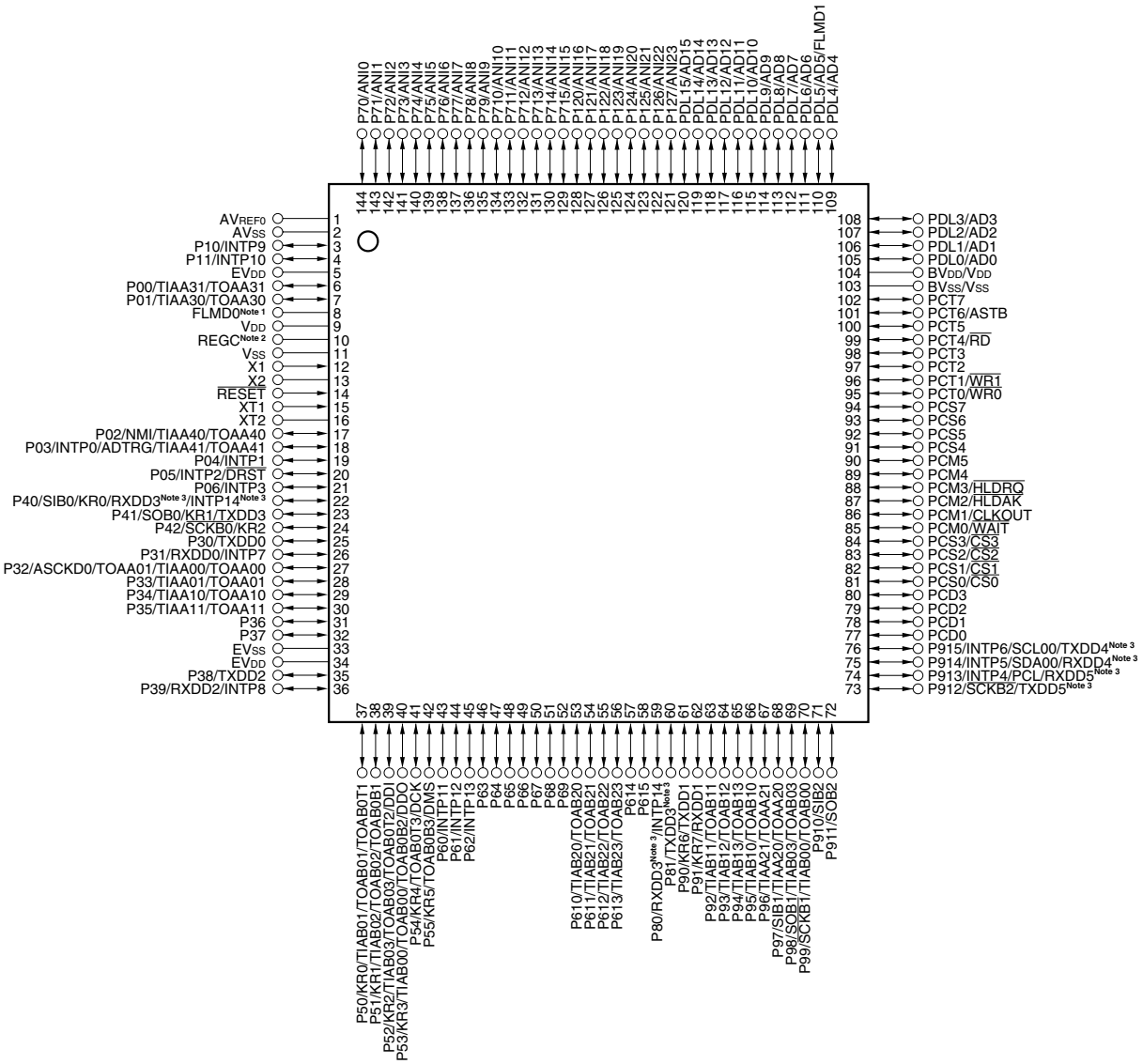
2. Connect the REGC pin to V_{SS} via a 4.7 μF (recommended value) capacitor.

1.5.4 V850ES/HJ3 (μPD70F3755, 70F3757)

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

μPD70F3755GJ-GAE-AX

μPD70F3757GJ-GAE-AX



Notes 1. Connect this pin to VSS in the normal mode.

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

3. μPD70F3757 only

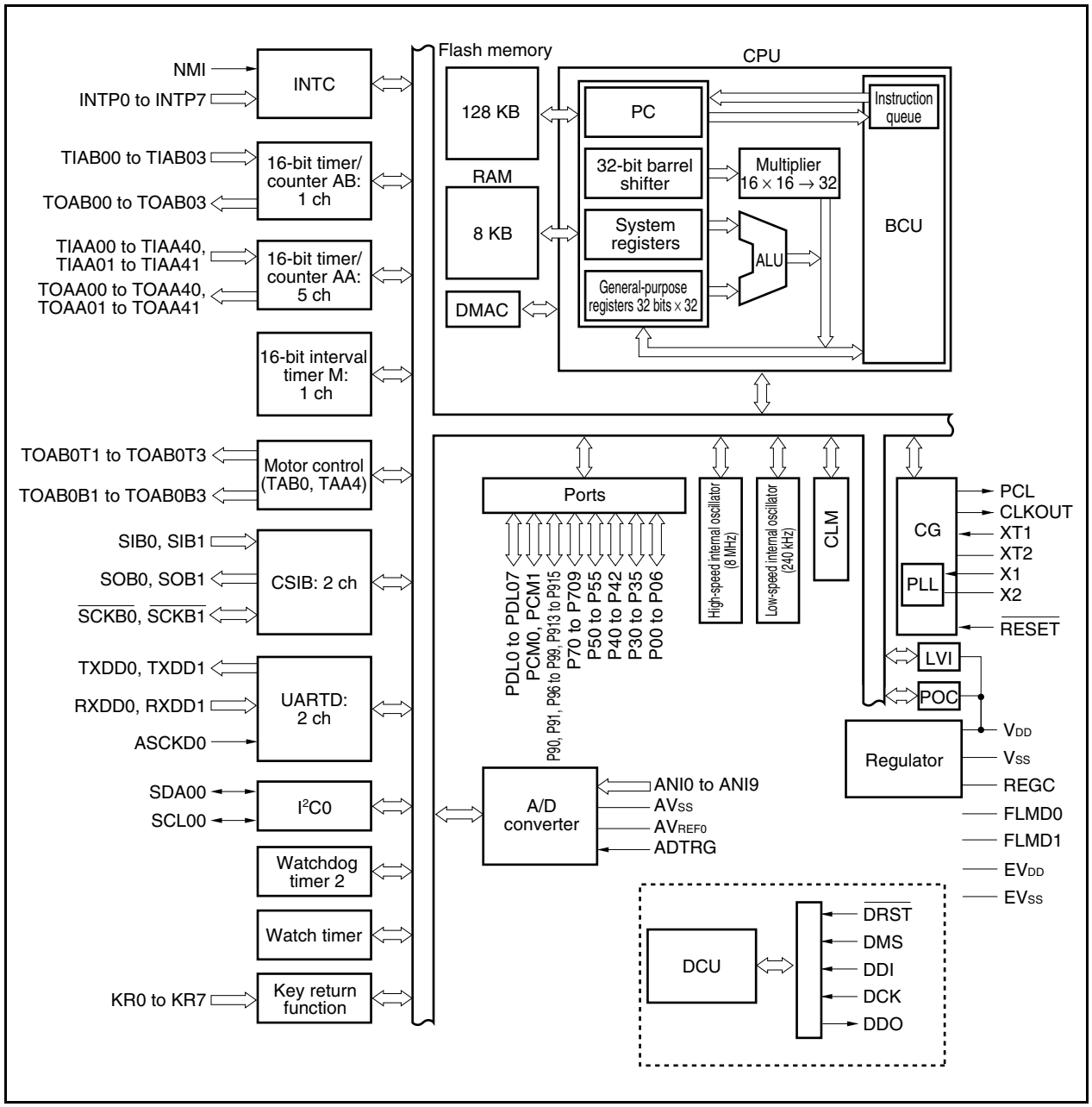
Pin identification

AD0 to AD15:	Address/data bus	$\overline{\text{RESET}}$:	Reset
ADTRG:	AD trigger input	RXDD0 to RXDD5:	Receive data
ANI0 to ANI23:	Analog input	$\overline{\text{SCKB0}}$ to $\overline{\text{SCKB2}}$:	Serial clock
ASTB:	Address strobe	SCL00:	Serial clock
AV_{REF0}:	Analog V_{DD}	SDA00:	Serial data
AV_{SS}:	Analog V_{SS}	SIB0 to SIB2:	Serial input
BV_{DD}:	Power supply for bus interface	SOB0 to SOB2:	Serial output
BV_{SS}:	Ground for bus interface	TIAA00, TIAA01,	
CLKOUT:	Clock output	TIAA10, TIAA11,	
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$:	Chip select	TIAA20, TIAA21,	
DCK:	Debug clock	TIAA30, TIAA31,	
DDI:	Debug data input	TIAA40, TIAA41,	
DDO:	Debug data output	TIAB00, TIAB01,	
DMS:	Debug mode select	TIAB02, TIAB03,	
$\overline{\text{DRST}}$:	Debug reset	TIAB10, TIAB11,	
EV_{DD}:	Power supply for external pin	TIAB12, TIAB13,	
EV_{SS}:	Ground for external pin	TIAB20, TIAB21,	
FLMD0, FLMD1:	Flash programming mode	TIAB22, TIAB23:	Timer input
$\overline{\text{HLDAK}}$:	Hold acknowledge	TOAA00, TOAA01,	
$\overline{\text{HLDRQ}}$:	Hold request	TOAA10, TOAA11,	
INTP0 to INTP14:	External interrupt input	TOAA20, TOAA21,	
KR0 to KR7:	Key return	TOAA30, TOAA31,	
NMI:	Non-maskable interrupt request	TOAA40, TOAA41,	
P00 to P06:	Port 0	TOAB00, TOAB01,	
P10, P11:	Port 1	TOAB02, TOAB03,	
P30 to P39:	Port 3	TOAB10, TOAB11,	
P40 to P42:	Port 4	TOAB12, TOAB13,	
P50 to P55:	Port 5	TOAB20, TOAB21,	
P60 to P615:	Port 6	TOAB22, TOAB23,	
P70 to P715:	Port 7	TOAB0B1, TOAB0B2,	
P80, P81:	Port 8	TOAB0B3, TOAB0T1,	
P90 to P915:	Port 9	TOAB0T2, TOAB0T3:	Timer output
P120 to P127:	Port 12	TXDD0 to TXDD5:	Transmit data
PCD0 to PCD3:	Port CD	V_{DD}:	Power supply
PCL:	Programmable clock output	V_{SS}:	Ground
PCM0 to PCM5:	Port CM	$\overline{\text{WAIT}}$:	Wait
PCS0 to PCS7:	Port CS	$\overline{\text{WR0}}$:	Write strobe low level data
PCT0 to PCT7:	Port CT	$\overline{\text{WR1}}$:	Write strobe high level data
PDL0 to PDL15:	Port DL	X1, X2:	Crystal for main clock
$\overline{\text{RD}}$	Read	XT1, XT2:	Crystal for subclock
REGC:	Regulator control		

1.6 Function Block Configuration

1.6.1 V850ES/HE3 (μPD70F3747)

(1) Internal block diagram



(2) Internal units

(a) CPU

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

Other dedicated on-chip hardware, such as a multiplier (16 bits × 16 bits → 32 bits) and a barrel shifter (32 bits) contribute to faster complex processing.

(b) Bus control unit (BCU)

The BCU controls the internal bus.

(c) ROM

This is a 128 KB flash memory mapped to addresses 0000000H to 001FFFFH. It can be accessed from the CPU in one clock during instruction fetch.

(d) RAM

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

(e) 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 multiple servicing control can be performed.

(f) Clock generator (CG)

A main clock oscillator that generates the main clock oscillation frequency (f_X) and a subclock oscillator that generates the subclock oscillation frequency (f_{XT}) are available. There are three modes, clock-through mode, PLL mode, and SSCG mode. In the clock-through mode, f_X is used as the main clock frequency (f_{XX}) as is.

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

(g) Internal oscillator

High-speed internal oscillator (f_{RH}: 8 MHz) and low-speed internal oscillator (f_{RL}: 240 kHz) are provided on chip. The high-speed internal oscillator (f_{RH}) is used as the internal system clock (f_{CLK}). The low-speed internal oscillator (f_{RL}) is used as the count clock for watchdog timer 2 and the sampling clock for clock monitor.

(h) Timer/counter

Five-channel 16-bit timer/event counter AA (TAA), one-channel 16-bit timer/event counter AB (TAB), and one-channel 16-bit interval timer M (TMM) are provided on chip. The 6-phase PWM output function can be used by using TAA and TAB in combination.

(i) Watch timer

This timer counts the reference time period (0.5 s) for counting the subclock (32.768 kHz) or the f_{BRG} (32.768 kHz) from prescaler 3. The watch timer can also be used as an interval timer for the main clock.

(j) Watchdog timer 2

A watchdog timer is provided on chip to detect inadvertent program loops, system abnormalities, etc.
Either the low-speed internal oscillation clock or the main clock 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.

(k) Serial interface

The V850ES/HE3 includes three kinds of serial interfaces: asynchronous serial interface A (UARTD), 3-wire variable-length serial interface B (CSIB), and I^2C bus interface (I^2C).
In the case of UARTD, data is transferred via the TXDD0, TXDD1, RXDD0, and RXDD1 pins.
In the case of CSIB, data is transferred via the SOB0, SOB1, SIB0, SIB1, $\overline{SCKB0}$, and $\overline{SCKB1}$ pins.
In the case of I^2C, data is transferred via the SDA00 and SCL00 pins.

(l) A/D converter

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

(m) 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.

(n) Key interrupt function

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

(o) DCU (debug control unit)

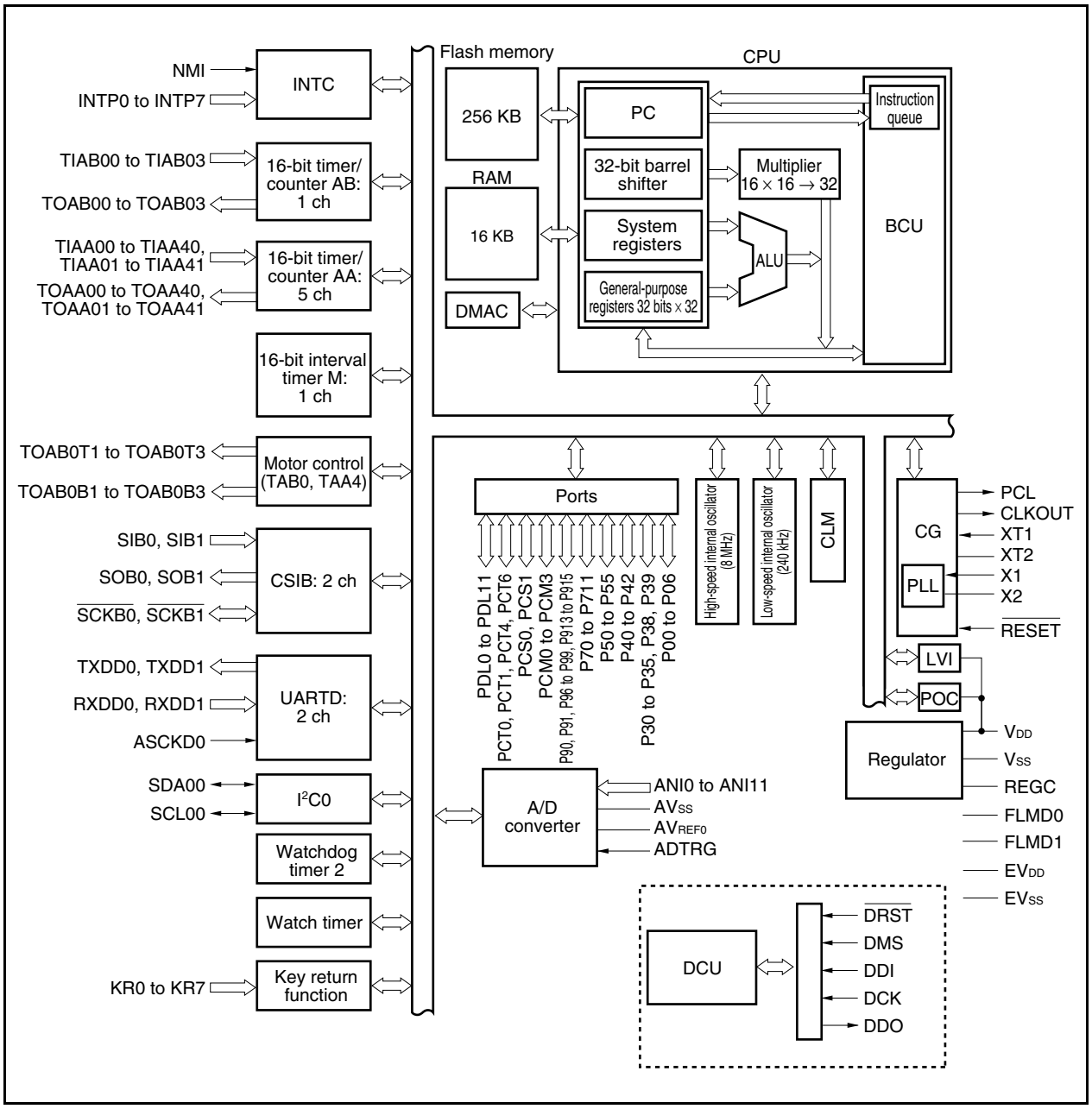
An on-chip debug function that uses the JTAG (Joint Test Action Group) communication specifications is provided. Switching between the normal port function and on-chip debugging function is done with the control pin input level and the OCDM register.

(p) Ports

The general-purpose port functions and control pin functions are provided. For details, see **CHAPTER 4 PORT FUNCTIONS**.

1.6.2 V850ES/HF3 (μPD70F3750)

(1) Internal block diagram



(2) Internal units

(a) CPU

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

Other dedicated on-chip hardware, such as a multiplier (16 bits × 16 bits → 32 bits) and a barrel shifter (32 bits) contribute to faster complex processing.

(b) Bus control unit (BCU)

The BCU controls the internal bus.

(c) ROM

This is a 256 KB flash memory mapped to addresses 0000000H to 003FFFFH. It can be accessed from the CPU in one clock during instruction fetch.

(d) RAM

This is a 16 KB RAM mapped to addresses 3FF7000H to 3FFEFFFH. It can be accessed from the CPU in one clock during data access.

(e) 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 multiple servicing control can be performed.

(f) Clock generator (CG)

A main clock oscillator that generates the main clock oscillation frequency (f_X) and a subclock oscillator that generates the subclock oscillation frequency (f_{XT}) are available. There are three modes, clock-through mode, PLL mode, and SSCG mode. In the clock-through mode, f_X is used as the main clock frequency (f_{XX}) as is.

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

(g) Internal oscillator

High-speed internal oscillator (f_{RH}: 8 MHz) and low-speed internal oscillator (f_{RL}: 240 kHz) are provided on chip. The high-speed internal oscillator (f_{RH}) is used as the internal system clock (f_{CLK}). The low-speed internal oscillator (f_{RL}) is used as the count clock for watchdog timer 2 and the sampling clock for clock monitor.

(h) Timer/counter

Five-channel 16-bit timer/event counter AA (TAA), one-channel 16-bit timer/event counter AB (TAB), and one-channel 16-bit interval timer M (TMM) are provided on chip. The 6-phase PWM output function can be used by using TAA and TAB in combination.

(i) Watch timer

This timer counts the reference time period (0.5 s) for counting the subclock (32.768 kHz) or the f_{BRG} (32.768 kHz) from prescaler 3. The watch timer can also be used as an interval timer for the main clock.

(j) Watchdog timer 2

A watchdog timer is provided on chip to detect inadvertent program loops, system abnormalities, etc.
Either the low-speed internal oscillation clock or the main clock 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.

(k) Serial interface

The V850ES/HF3 includes three kinds of serial interfaces: asynchronous serial interface A (UARTD), 3-wire variable-length serial interface B (CSIB), and I^2C bus interface (I^2C).
In the case of UARTD, data is transferred via the TXDD0, TXDD1, RXDD0, and RXDD1 pins.
In the case of CSIB, data is transferred via the SOB0, SOB1, SIB0, SIB1, $\overline{SCKB0}$, and $\overline{SCKB1}$ pins.
In the case of I^2C, data is transferred via the SDA00 and SCL00 pins.

(l) A/D converter

This 10-bit A/D converter includes 12 analog input pins. Conversion is performed using the successive approximation method.

(m) 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.

(n) Key interrupt function

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

(o) DCU (debug control unit)

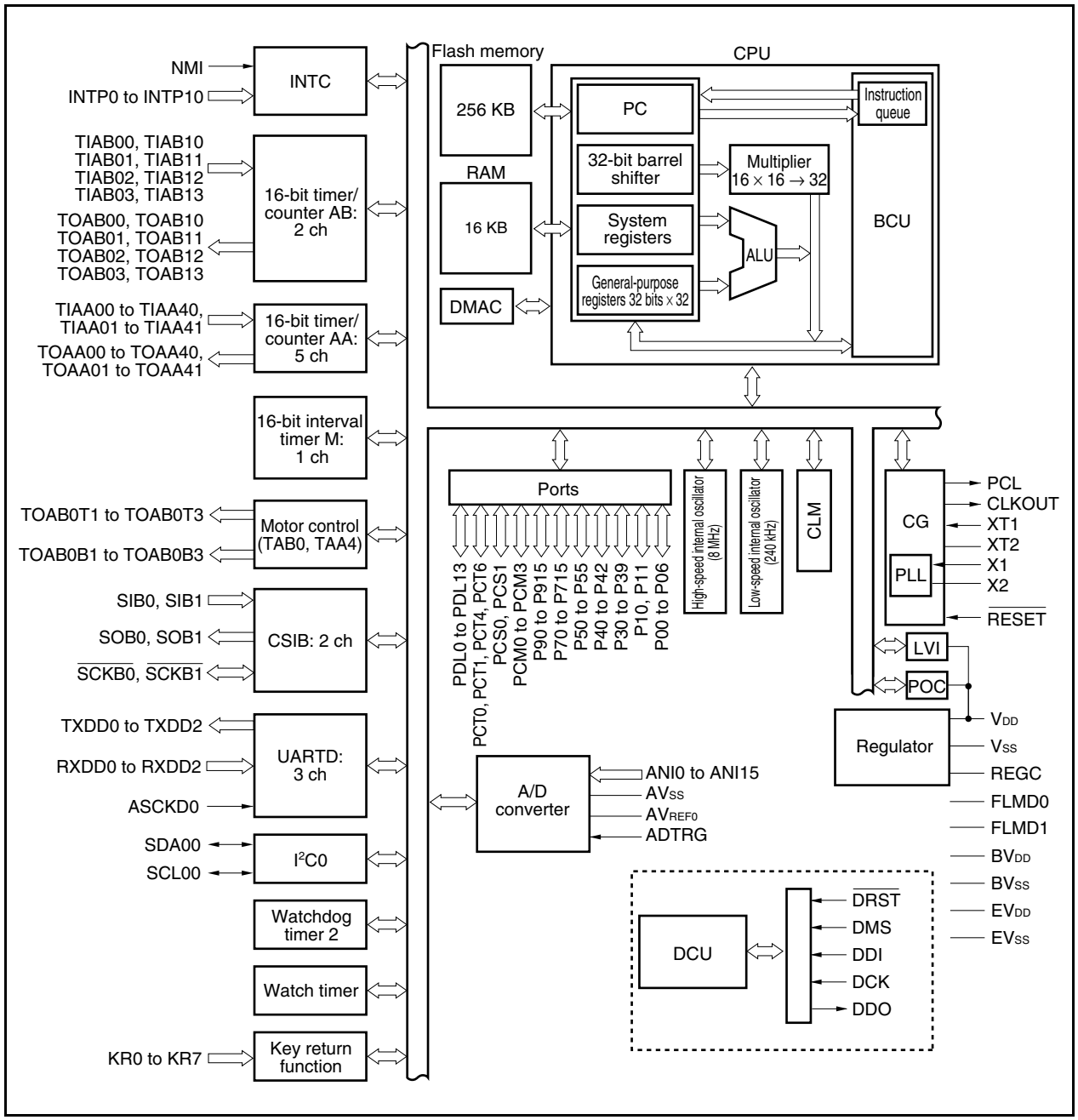
An on-chip debug function that uses the JTAG (Joint Test Action Group) communication specifications is provided. Switching between the normal port function and on-chip debugging function is done with the control pin input level and the OCDM register.

(p) Ports

The general-purpose port functions and control pin functions are provided. For details, see **CHAPTER 4 PORT FUNCTIONS**.

1.6.3 V850ES/HG3 (μPD70F3752)

(1) Internal block diagram



(2) Internal units

(a) CPU

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

Other dedicated on-chip hardware, such as a multiplier (16 bits × 16 bits → 32 bits) and a barrel shifter (32 bits) contribute to faster complex processing.

(b) Bus control unit (BCU)

The BCU controls the internal bus.

(c) ROM

This is a 256 KB flash memory mapped to addresses 0000000H to 003FFFFH. It can be accessed from the CPU in one clock during instruction fetch.

(d) RAM

This is a 16 KB RAM mapped to addresses 3FF7000H to 3FFEFFFH. It can be accessed from the CPU in one clock during data access.

(e) Interrupt controller (INTC)

This controller handles hardware interrupt requests (NMI, INTP0 to INTP10) from on-chip peripheral hardware and external hardware. Eight levels of interrupt priorities can be specified for these interrupt requests, and multiple servicing control can be performed.

(f) Clock generator (CG)

A main clock oscillator that generates the main clock oscillation frequency (f_X) and a subclock oscillator that generates the subclock oscillation frequency (f_{XT}) are available. There are three modes, clock-through mode, PLL mode, and SSCG mode. In the clock-through mode, f_X is used as the main clock frequency (f_{XX}) as is.

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

(g) Internal oscillator

High-speed internal oscillator (f_{RH}: 8 MHz) and low-speed internal oscillator (f_{RL}: 240 kHz) are provided on chip. The high-speed internal oscillator (f_{RH}) is used as the internal system clock (f_{CLK}). The low-speed internal oscillator (f_{RL}) is used as the count clock for watchdog timer 2 and the sampling clock for clock monitor.

(h) Timer/counter

Five-channel 16-bit timer/event counter AA (TAA), two-channel 16-bit timer/event counter AB (TAB), and one-channel 16-bit interval timer M (TMM) are provided on chip. The 6-phase PWM output function can be used by using TAA and TAB in combination.

(i) Watch timer

This timer counts the reference time period (0.5 s) for counting the subclock (32.768 kHz) or the f_{BRG} (32.768 kHz) from prescaler 3. The watch timer can also be used as an interval timer for the main clock.

(j) Watchdog timer 2

A watchdog timer is provided on chip to detect inadvertent program loops, system abnormalities, etc.
Either the low-speed internal oscillation clock or the main clock 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.

(k) Serial interface

The V850ES/HG3 includes three kinds of serial interfaces: asynchronous serial interface A (UARTD), 3-wire variable-length serial interface B (CSIB), and I^2C bus interface (I^2C).
In the case of UARTD, data is transferred via the TXDD0 to TXDD2 and RXDD0 to RXDD2 pins.
In the case of CSIB, data is transferred via the SOB0, SOB1, SIB0, SIB1, $\overline{SCKB0}$, and $\overline{SCKB1}$ pins.
In the case of I^2C, data is transferred via the SDA00 and SCL00 pins.

(l) A/D converter

This 10-bit A/D converter includes 16 analog input pins. Conversion is performed using the successive approximation method.

(m) 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.

(n) Key interrupt function

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

(o) DCU (debug control unit)

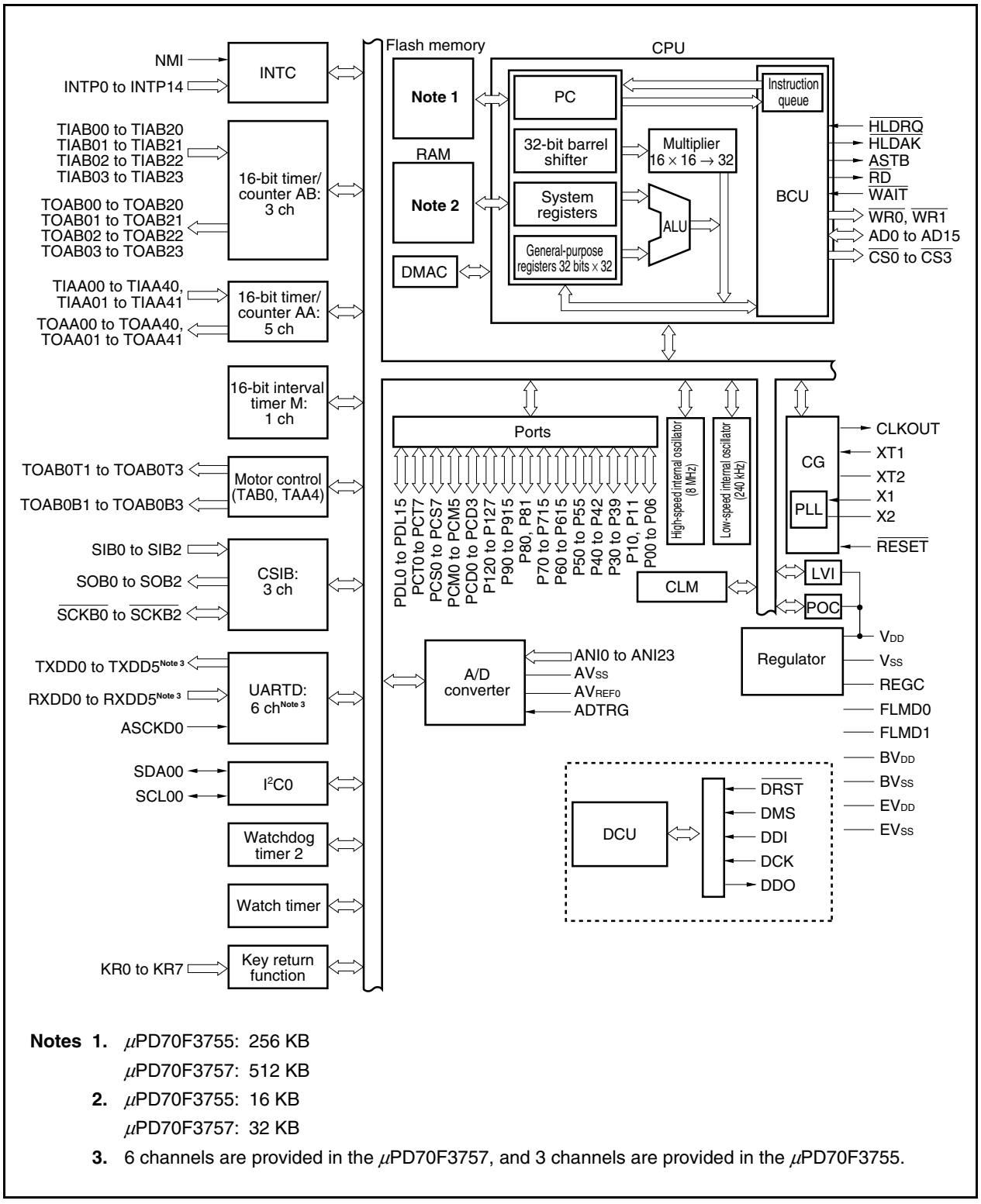
An on-chip debug function that uses the JTAG (Joint Test Action Group) communication specifications is provided. Switching between the normal port function and on-chip debugging function is done with the control pin input level and the OCDM register.

(p) Ports

The general-purpose port functions and control pin functions are provided. For details, see **CHAPTER 4 PORT FUNCTIONS**.

1.6.4 V850ES/HJ3 (μPD70F3755, 70F3757)

(1) Internal block diagram



Notes 1. μPD70F3755: 256 KB
μPD70F3757: 512 KB

2. μPD70F3755: 16 KB
μPD70F3757: 32 KB

3. 6 channels are provided in the μPD70F3757, and 3 channels are provided in the μPD70F3755.

(2) Internal units

(a) CPU

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

Other dedicated on-chip hardware, such as a multiplier (16 bits × 16 bits → 32 bits) and a barrel shifter (32 bits) contribute to faster complex processing.

(b) Bus control unit (BCU)

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

(c) ROM

This is a 512 KB/256 KB flash memory mapped to addresses 0000000H to 007FFFFH/0000000H to 003FFFFH. It can be accessed from the CPU in one clock during instruction fetch.

(d) RAM

This is a 32 KB/16 KB RAM mapped to addresses 3FF7000H to 3FFEFFFH/3FFB000H to 3FFEFFFH. It can be accessed from the CPU in one clock during data access.

(e) Interrupt controller (INTC)

This controller handles hardware interrupt requests (NMI, INTP0 to INTP14) from on-chip peripheral hardware and external hardware. Eight levels of interrupt priorities can be specified for these interrupt requests, and multiple servicing control can be performed.

(f) Clock generator (CG)

A main clock oscillator that generates the main clock oscillation frequency (f_X) and a subclock oscillator that generates the subclock oscillation frequency (f_{XT}) are available. There are three modes, clock-through mode, PLL mode, and SSCG mode. In the clock-through mode, f_X is used as the main clock frequency (f_{XX}) as is.

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

(g) Internal oscillator

High-speed internal oscillator (f_{RH}: 8 MHz) and low-speed internal oscillator (f_{RL}: 240 kHz) are provided on chip. The high-speed internal oscillator (f_{RH}) is used as the internal system clock (f_{CLK}). The low-speed internal oscillator (f_{RL}) is used as the count clock for watchdog timer 2 and the sampling clock for clock monitor.

(h) Timer/counter

Five-channel 16-bit timer/event counter AA (TAA), three-channel 16-bit timer/event counter AB (TAB), and one-channel 16-bit interval timer M (TMM) are provided on chip. The 6-phase PWM output function can be used by using TAA and TAB in combination.

(i) Watch timer

This timer counts the reference time period (0.5 s) for counting the subclock (32.768 kHz) or the f_{BRG} (32.768 kHz) from prescaler 3. The watch timer can also be used as an interval timer for the main clock.

(j) Watchdog timer 2

A watchdog timer is provided on chip to detect inadvertent program loops, system abnormalities, etc.
Either the low-speed internal oscillation clock or the main clock 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.

(k) Serial interface

The V850ES/HJ3 includes three kinds of serial interfaces: asynchronous serial interface A (UARTD), 3-wire variable-length serial interface B (CSIB), and I^2C bus interface (I^2C).
In the case of UARTD, data is transferred via the TXDDn and RXDDn pins.
(n = 0 to 5: μPD70F3757, n = 0 to 2: μPD70F3755)
In the case of CSIB, data is transferred via the SOB0 to SOB2, SIB0 to SIB2, and $\overline{\text{SCKB0}}$ to $\overline{\text{SCKB2}}$ pins.
In the case of I^2C, data is transferred via the SDA00 and SCL00 pins.

(l) A/D converter

This 10-bit A/D converter includes 24 analog input pins. Conversion is performed using the successive approximation method.

(m) DMA controller

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

(n) Key interrupt function

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

(o) 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.

(p) Ports

The general-purpose port functions and control pin functions are provided. For details, see **CHAPTER 4 PORT FUNCTIONS**.

1.7 Overview of Functions

Table 1-1. V850ES/Hx3 Function List

Generic Name		V850ES/HE3	V850ES/HF3	V850ES/HG3	V850ES/HJ3	
Product Name		μPD70F3747	μPD70F3750	μPD70F3752	μPD70F3755	μPD70F3757
Internal memory	Flash memory	128 KB	256 KB	256 KB	256 KB	512 KB
	RAM	8 KB	16 KB	16 KB	16 KB	32 KB
External bus interface		–			Address/data buses: 16 Chip select signals: 4 Multiplexed bus mode supported	
General-purpose register		32 bits × 32 registers				
Clocks	Minimum instruction execution time	31.25 ns (f_{XX} = 32 MHz)				
	Main clock oscillation	f_X = 4 to 16 MHz				
	Subclock oscillation	f_{XT} = 32.768 kHz				
	SSCG	Frequency modulation rate specification ±0.5% to ±5% (TYP.)				
	PLL multiplication	Multiplication by 8				
	High-speed internal oscillation	f_{RH} = 8 MHz (TYP.)				
	Low-speed internal oscillation	f_{RL} = 240 kHz (TYP.)				
I/O ports		I/O: 51	I/O: 67	I/O: 84	I/O: 128	
Timer	16-bit TAA	5 channels	5 channels	5 channels	5 channels	
	16-bit TAB	1 channel	1 channel	2 channels	3 channels	
	16-bit TMM	1 channel	1 channel	1 channel	1 channel	
	Motor control	1 channel	1 channel	1 channel	1 channel	
	WDT	1 channel	1 channel	1 channel	1 channel	
	Watch timer	1 channel	1 channel	1 channel	1 channel	
10-bit A/D converter		10 channels	12 channels	16 channels	24 channels	
Serial interface	CSIB	2 channels	2 channels	2 channels	3 channels	
	UARTD	2 channels	2 channels	3 channels	3 channels	6 channels
	I^2C	1 channel	1 channel	1 channel	1 channel	
DMA controller		4 channels (transfer target: on-chip peripheral I/O, internal RAM)			4 channels (transfer target: on-chip peripheral I/O, internal RAM, external memory)	
Interrupt source	External	9 (9)[Note]	9 (9)[Note]	12 (12)[Note]	16 (16)[Note]	
	Internal	43	43	51	58	64
Power-save function		HALT/IDLE1/IDLE2/STOP/subclock/sub-IDLE modes				
Reset factor		$\overline{\text{RESET}}$ pin input, watchdog timer 2 (WDT2), clock monitor (CLM), power-on-clear (POC), low-voltage detector (LVI)				
On-chip debugging		MINICUBE®, MINICUBE2 supported				
Operating supply voltage		3.7 to 5.5 V				
Package		64-pin LQFP (10 × 10 mm)	80-pin LQFP (12 × 12 mm)	100-pin LQFP (14 × 14 mm)	144-pin LQFP (20 × 20 mm)	

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

CHAPTER 2 PIN FUNCTIONS

This section explains the names and functions of the pins of the V850ES/HE3, V850ES/HF3, V850ES/HG3, and V850ES/HJ3.

2.1 Pin Function List

Three I/O buffer power supplies, AV_{REF0}, BV_{DD}, and EV_{DD}, are available (BV_{DD} is not available in the V850ES/HE3 and V850ES/HF3). The relationship between the power supplies and the pins is shown below.

Table 2-1. Pin I/O Buffer Power Supplies (V850ES/HE3)

Power Supply	Corresponding Pin
AV_{REF0}	Port 7
EV_{DD}	Port 0, port 3, port 4, port 5, port 7, port 9, port CM, port DL, $\overline{\text{RESET}}$

Table 2-2. Pin I/O Buffer Power Supplies (V850ES/HF3)

Power Supply	Corresponding Pin
AV_{REF0}	Port 7
EV_{DD}	Port 0, port 3, port 4, port 5, port 7, port 9, port CM, port CS, port CT, port DL, $\overline{\text{RESET}}$

Table 2-3. Pin I/O Buffer Power Supplies (V850ES/HG3)

Power Supply	Corresponding Pin
AV_{REF0}	Port 7
BV_{DD}	Port CM, port CS, port CT, port DL
EV_{DD}	Port 0, port 1, port 3, port 4, port 5, port 7, port 9, $\overline{\text{RESET}}$

Table 2-4. Pin I/O Buffer Power Supplies (V850ES/HJ3)

Power Supply	Corresponding Pin
AV_{REF0}	Port 7, port 12
BV_{DD}	Port CD, port CM, port CS, port CT, port DL
EV_{DD}	Port 0, port 1, port 3, port 4, port 5, port 6, port 8, port 9, $\overline{\text{RESET}}$

(1) Port pins

(1/5)

Pin Name	I/O	Function	Alternate Function	Pin No.			
				HE3	HF3	HG3	HJ3
P00	I/O	Port 0 7-bit I/O port Input/output can be specified in 1-bit units. On-chip pull-up resistor can be connected in 1-bit units.	TIAA31/TOAA31	12	3	6	6
P01			TIAA30/TOAA30	13	4	7	7
P02			NMI/TIAA40/TOAA40	14	5	17	17
P03			INTP0/ADTRG/TIAA41/TOAA41	15	6	18	18
P04			INTP1	16	7	19	19
P05			INTP2/$\overline{\text{DRST}}$	17	17	20	20
P06			INTP3	18	18	21	21
P10	I/O	Port 1 2-bit I/O port Input/output can be specified in 1-bit units. On-chip pull-up resistor can be connected in 1-bit units.	INTP9	–	–	3	3
P11			INTP10	–	–	4	4
P30	I/O	Port 3 6-bit I/O port (V850ES/HE3) 8-bit I/O port (V850ES/HF3) 10-bit I/O port (V850ES/HG3, V850ES/HJ3) Input/output can be specified in 1-bit units. On-chip pull-up resistor can be connected in 1-bit units.	TXDD0	22	22	25	25
P31			RXDD0/INTP7	23	23	26	26
P32			ASCKD0/TOAA01/TIAA00/TOAA00	24	24	27	27
P33			TIAA01/TOAA01	25	25	28	28
P34			TIAA10/TOAA10	26	26	29	29
P35			TIAA11/TOAA11	27	27	30	30
P36			–	–	–	31	31
P37			–	–	–	32	32
P38			TXDD2	–	–	35	35
			–	–	28	–	–
P39			RXDD2/INTP8	–	–	36	36
			–	–	29	–	–
P40	I/O	Port 4 3-bit I/O port Input/output can be specified in 1-bit units. On-chip pull-up resistor can be connected in 1-bit units.	SIB0/KR0/RXDD3[Note]/INTP14[Note]	–	–	–	22
			SIB0/KR0	19	19	22	–
P41			SOB0/KR1/TXDD3[Note]	–	–	–	23
			SOB0/KR1	20	20	23	–
P42			$\overline{\text{SCKB0}}$/KR2	21	21	24	24
P50	I/O	Port 5 6-bit I/O port Input/output can be specified in 1-bit units. On-chip pull-up resistor can be connected in 1-bit units.	KR0/TIAB01/TOAB01/TOAB0T1	28	32	37	37
P51			KR1/TIAB02/TOAB02/TOAB0B1	29	33	38	38
P52			KR2/TIAB03/TOAB03/TOAB0T2/DDI	30	34	39	39
P53			KR3/TIAB00/TOAB00/TOAB0B2/DDO	31	35	40	40
P54			KR4/TOAB0T3/DCK	34	36	41	41
P55			KR5/TOAB0B3/DMS	35	37	42	42

Note μPD70F3757 only

Remark HE3: V850ES/HE3, HF3: V850ES/HF3, HG3: V850ES/HG3, HJ3: V850ES/HJ3

(2/5)

Pin Name	I/O	Function	Alternate Function	Pin No,			
				HE3	HF3	HG3	HJ3
P60	I/O	Port 6 16-bit I/O port Input/output can be specified in 1-bit units. On-chip pull-up resistor can be connected in 1-bit units.	NTP11	–	–	–	43
P61			NTP12	–	–	–	44
P62			INTP13	–	–	–	45
P63			–	–	–	–	46
P64			–	–	–	–	47
P65			–	–	–	–	48
P66			–	–	–	–	49
P67			–	–	–	–	50
P68			–	–	–	–	51
P69			–	–	–	–	52
P610			TIAB20/TOAB20	–	–	–	53
P611			TIAB21/TOAB21	–	–	–	54
P612			TIAB22/TOAB22	–	–	–	55
P613			TIAB23/TOAB23	–	–	–	56
P614			–	–	–	–	57
P615			–	–	–	–	58
P70	I/O	Port 7 10-bit I/O port (V850ES/HE3) 12-bit I/O port (V850ES/HF3) 16-bit I/O port (V850ES/HG3, V850ES/HJ3) Input/output can be specified in 1-bit units.	ANI0	64	80	100	144
P71			ANI1	63	79	99	143
P72			ANI2	62	78	98	142
P73			ANI3	61	77	97	141
P74			ANI4	60	76	96	140
P75			ANI5	59	75	95	139
P76			ANI6	58	74	94	138
P77			ANI7	57	73	93	137
P78			ANI8	56	72	92	136
P79			ANI9	55	71	91	135
P710			ANI10	–	70	90	134
P711			ANI11	–	69	89	133
P712			ANI12	–	–	88	132
P713			ANI13	–	–	87	131
P714			ANI14	–	–	86	130
P715			ANI15	–	–	85	129
P80	I/O	Port 8 2-bit I/O port Input/output can be specified in 1-bit units. On-chip pull-up resistor can be connected in 1-bit units.	RXDD3[Note]/INTP14	–	–	–	59
P81			TXDD3[Note]	–	–	–	60

Note μPD70F3757 only

Remark HE3: V850ES/HE3, HF3: V850ES/HF3, HG3: V850ES/HG3, HJ3: V850ES/HJ3

(3/5)

Pin Name	I/O	Function	Alternate Function	Pin No.			
				HE3	HF3	HG3	HJ3
P90	I/O	Port 9 9-bit I/O port (V850ES/HE3, V850ES/HF3) 16-bit I/O port (V850ES/HG3, V850ES/HJ3) Input/output can be specified in 1-bit units. On-chip pull-up resistor can be connected in 1-bit units.	KR6/TXDD1	36	38	43	61
P91			KR7/RXDD1	37	39	44	62
P92			TIAB11/TOAB11	–	–	45	63
P93			TIAB12/TOAB12	–	–	46	64
P94			TIAB13/TOAB13	–	–	47	65
P95			TIAB10/TOAB10	–	–	48	66
P96			TIAA21/TOAA21	38	40	49	67
P97			SIB1/TIAA20/TOAA20	39	41	50	68
P98			SOB1/TIAB03/TOAB03	40	42	51	69
P99			$\overline{\text{SCKB1}}$/TIAB00/TOAB00	41	43	52	70
P910			SIB2	–	–	–	71
			–	–	–	53	–
P911			SOB2	–	–	–	72
			–	–	–	54	–
P912			$\overline{\text{SCKB2}}$/TXDD5[Note]	–	–	–	73
			–	–	–	55	–
P913			INTP4/PCL/RXDD5[Note]	–	–	–	74
			INTP4/PCL	42	44	56	–
P914			INTP5/SDA00/RXDD4[Note]	–	–	–	75
			INTP5/SDA00	43	45	57	–
P915			INTP6/SCL00/TXDD4[Note]	–	–	–	76
			INTP6/SCL00	44	46	58	–
P120	I/O	Port 12 8-bit I/O port Input/output can be specified in 1-bit units.	ANI16	–	–	–	128
P121			ANI17	–	–	–	127
P122			ANI18	–	–	–	126
P123			ANI19	–	–	–	125
P124			ANI20	–	–	–	124
P125			ANI21	–	–	–	123
P126			ANI22	–	–	–	122
P127			ANI23	–	–	–	121
PCD0	I/O	Port CD 4-bit I/O port Input/output can be specified in 1-bit units.	–	–	–	–	77
PCD1			–	–	–	–	78
PCD2			–	–	–	–	79
PCD3			–	–	–	–	80
PCM0	I/O	Port CM 2-bit I/O port (V850ES/HE3) 4-bit I/O port (V850ES/HF3, V850ES/HG3) 6-bit I/O port (V850ES/HJ3) Input/output can be specified in 1-bit units.	$\overline{\text{WAIT}}$	–	–	–	85
			–	45	49	61	–
PCM1			CLKOUT	46	50	62	86
PCM2			$\overline{\text{HLDAK}}$	–	–	–	87
			–	–	51	63	–
PCM3			$\overline{\text{HLDRQ}}$	–	–	–	88
			–	–	52	64	–
PCM4			–	–	–	–	89
PCM5			–	–	–	–	90

Note μPD70F3757 only

Remark HE3: V850ES/HE3, HF3: V850ES/HF3, HG3: V850ES/HG3, HJ3: V850ES/HJ3

(4/5)

Pin Name	I/O	Function	Alternate Function	Pin No.			
				HE3	HF3	HG3	HJ3
PCS0	I/O	Port CS 2-bit I/O port (V850ES/HF3, V850ES/HG3) 8-bit I/O port (V850ES/HJ3) Input/output can be specified in 1-bit units.	$\overline{CS0}$	–	–	–	81
			–	–	47	59	–
PCS1			$\overline{CS1}$	–	–	–	82
			–	–	48	60	–
PCS2			$\overline{CS2}$	–	–	–	83
PCS3			$\overline{CS3}$	–	–	–	84
PCS4			–	–	–	–	91
PCS5			–	–	–	–	92
PCS6			–	–	–	–	93
PCS7			–	–	–	–	94
PCT0	I/O	Port CT 4-bit I/O port (V850ES/HF3, V850ES/HG3) 8-bit I/O port (V850ES/HJ3) Input/output can be specified in 1-bit units.	$\overline{WR0}$	–	–	–	95
			–	–	53	65	–
PCT1			$\overline{WR1}$	–	–	–	96
			–	–	54	66	–
PCT2			–	–	–	–	97
PCT3			–	–	–	–	98
PCT4			$\overline{RD}$	–	–	–	99
			–	–	55	67	–
PCT5			–	–	–	–	100
PCT6			ASTB	–	–	–	101
			–	–	56	68	–
PCT7			–	–	–	–	102
PDL0	I/O	Port DL 8-bit I/O port (V850ES/HE3) 12-bit I/O port (V850ES/HF3) 14-bit I/O port (V850ES/HG3) 16-bit I/O port (V850ES/HJ3) Input/output can be specified in 1-bit units.	AD0	–	–	–	105
			–	47	57	71	–
PDL1			AD1	–	–	–	106
			–	48	58	72	–
PDL2			AD2	–	–	–	107
			–	49	59	73	–
PDL3			AD3	–	–	–	108
			–	50	60	74	–
PDL4			AD4	–	–	–	109
			–	51	61	75	–
PDL5			AD5/FLMD1	–	–	–	110
			FLMD1	52	62	76	–
PDL6			AD6	–	–	–	111
			–	53	63	77	–
PDL7			AD7	–	–	–	112
			–	54	64	78	–
PDL8			AD8	–	–	–	113
			–	–	65	79	–
PDL9			AD9	–	–	–	114
			–	–	66	80	–

Remark HE3: V850ES/HE3, HF3: V850ES/HF3, HG3: V850ES/HG3, HJ3: V850ES/HJ3

(5/5)

Pin Name	I/O	Function	Alternate Function	Pin No.			
				HE3	HF3	HG3	HJ3
PDL10	I/O	Port DL	AD10	–	–	–	115
		8-bit I/O port (V850ES/HE3)	–	–	67	81	–
PDL11		12-bit I/O port (V850ES/HF3)	AD11	–	–	–	116
		14-bit I/O port (V850ES/HG3)	–	–	68	82	–
PDL12		16-bit I/O port (V850ES/HJ3)	AD12	–	–	–	117
		Input/output can be specified in 1-bit units.	–	–	–	83	–
PDL13			AD13	–	–	–	118
			–	–	–	84	–
PDL14			AD14	–	–	–	119
PDL15			AD15	–	–	–	120

Remark HE3: V850ES/HE3, HF3: V850ES/HF3, HG3: V850ES/HG3, HJ3: V850ES/HJ3

(2) Non-port pins

(1/6)

Pin Name	I/O	Function	Alternate Function	Pin No.			
				HE3	HF3	HG3	HJ3
AD0	I/O	Address/data bus for external memory	PDL0	–	–	–	105
AD1			PDL1	–	–	–	106
AD2			PDL2	–	–	–	107
AD3			PDL3	–	–	–	108
AD4			PDL4	–	–	–	109
AD5			PDL5/FLMD1	–	–	–	110
AD6			PDL6	–	–	–	111
AD7			PDL7	–	–	–	112
AD8			PDL8	–	–	–	113
AD9			PDL9	–	–	–	114
AD10			PDL10	–	–	–	115
AD11			PDL11	–	–	–	116
AD12			PDL12	–	–	–	117
AD13			PDL13	–	–	–	118
AD14			PDL14	–	–	–	119
AD15			PDL15	–	–	–	120
ADTRG	Input	External trigger input for A/D converter	P03/INTP0/TIAA41/TOAA41	15	6	18	18
ANI0	Input	Analog voltage input for A/D converter	P70	64	80	100	144
ANI1			P71	63	79	99	143
ANI2			P72	62	78	98	142
ANI3			P73	61	77	97	141
ANI4			P74	60	76	96	140
ANI5			P75	59	75	95	139
ANI6			P76	58	74	94	138
ANI7			P77	57	73	93	137
ANI8			P78	56	72	92	136
ANI9			P79	55	71	91	135
ANI10			P710	–	70	90	134
ANI11			P711	–	69	89	133
ANI12			P712	–	–	88	132
ANI13			P713	–	–	87	131
ANI14			P714	–	–	86	130
ANI15			P715	–	–	85	129
ANI16			P120	–	–	–	128
ANI17			P121	–	–	–	127
ANI18			P122	–	–	–	126
ANI19			P123	–	–	–	125
ANI20			P124	–	–	–	124
ANI21			P125	–	–	–	123

Remark HE3: V850ES/HE3, HF3: V850ES/HF3, HG3: V850ES/HG3, HJ3: V850ES/HJ3

(2/6)

Pin Name	I/O	Function	Alternate Function	Pin No.			
				HE3	HF3	HG3	HJ3
ANI22	Input	Analog voltage input for A/D converter	P126	–	–	–	122
ANI23			P127	–	–	–	121
ASCKD0	Input	Serial clock input (UARTD0)	P32/TOAA01/TIAA00/TOAA00	24	24	27	27
ASTB	Output	Address strobe signal for external memory	PCT6	–	–	–	101
AV_{REF0}	–	Reference voltage for A/D converter	–	1	1	1	1
AV_{SS}	–	Ground potential for A/D converter	–	2	2	2	2
BV_{DD}	–	Positive power supply for external I/O (alphabetical ports and their alternate functions)	–	–	–	70	104
BV_{SS}	–	Ground potential for external I/O (alphabetical ports and their alternate functions)	–	–	–	69	103
CLKOUT	Output	Internal system clock output	PCM1	46	50	62	86
$\overline{CS0}$	Output	Chip select output	PCS0	–	–	–	81
$\overline{CS1}$			PCS1	–	–	–	82
$\overline{CS2}$			PCS2	–	–	–	83
$\overline{CS3}$			PCS3	–	–	–	84
DCK	Input	Clock input for on-chip debugging	P54/KR4/TOAB0T3	34	36	41	41
DDI	Input	Data input for on-chip debugging	P52/KR2/TIAB03/TOAB03 /TOAB0T2	30	34	39	39
DDO	Output	Data output for on-chip debugging	P53/KR3/TIAB00/TOAB00 /TOAB0B2	31	35	40	40
DMS	Input	Mode select signal input for on-chip debugging	P55/KR5/TOAB0B3	35	37	42	42
$\overline{DRST}$	Input	Reset signal input for on-chip debugging	P05/INTP2	17	17	20	20
EV_{DD}	–	Positive power supply for external I/O (ports (numerical ports only in V850ES/HG3, V850ES/HJ3) and their alternate functions)	–	33	31	5, 34	5, 34
EV_{SS}	–	Ground potential for external I/O (ports (numerical ports only in V850ES/HG3, V850ES/HJ3) and their alternate functions)	–	32	30	33	33
FLMD0	–	Flash programming mode setting pins	–	3	8	8	8
FLMD1			PDL5/AD5	–	–	–	110
			PDL5	52	62	76	–
$\overline{HLDAK}$	Output	Bus hold acknowledge output	PCM2	–	–	–	87
$\overline{HLDRQ}$	Input	Bus hold request input	PCM3	–	–	–	88
INTP0	Input	Maskable interrupt input	P03/ADTRG/TIAA41/TOAA41	15	6	18	18
INTP1			P04	16	7	19	19
INTP2			P05/$\overline{DRST}$	17	17	20	20
INTP3			P06	18	18	21	21
INTP4			P913/PCL/RXDD5[Note]	–	–	–	74
			P913/PCL	42	44	56	–
INTP5			P914/SDA00/RXDD4[Note]	–	–	–	75
			P914/SDA00	43	45	57	–

Note μPD70F3757 only

Remark HE3: V850ES/HE3, HF3: V850ES/HF3, HG3: V850ES/HG3, HJ3: V850ES/HJ3

(3/6)

Pin Name	I/O	Function	Alternate Function	Pin No.			
				HE3	HF3	HG3	HJ3
INTP6	Input	Maskable interrupt input	P915/SCL00/TXDD4	–	–	–	76
			P915/SCL00	44	46	58	–
INTP7			P31/RXDD0	23	23	26	26
INTP8			P39/RXDD2	–	–	36	36
INTP9			P10	–	–	3	3
INTP10			P11	–	–	4	4
INTP11			P60	–	–	–	43
INTP12			P61	–	–	–	44
INTP13			P62	–	–	–	45
INTP14			P40/SIB0/KR0/RXDD3	–	–	–	22[Note]
			P80/RXDD3[Note]	–	–	–	59
KR0	Input	Key interrupt input	P40/SIB0/RXDD3[Note] /INTP14[Note]	–	–	–	22
			P40/SIB0	19	19	22	–
			P50/TIAB01/TOAB01 /TOAB0T1	28	32	37	37
KR1			P41/SOB0/TXDD3[Note]	–	–	–	23
			P41/SOB0	20	20	23	–
			P51/TIAB02/TOAB02 /TOAB0B1	29	33	38	38
KR2			P42/$\overline{\text{SCKB0}}$	21	21	24	24
			P52/TIAB03/TOAB03 /TOAB0T2/DDI	30	34	39	39
KR3			P53/TIAB00/TOAB00 /TOAB0B2/DDO	31	35	40	40
KR4			P54/TOAB0T3/DCK	34	36	41	41
KR5			P55/TOAB0B3/DMS	35	37	42	42
KR6			P90/TXDD1	36	38	43	61
KR7			P91/RXDD1	37	39	44	62
NMI	Input	Non-maskable interrupt input	P02/TIAA40/TOAA40	14	5	17	17
PCL	Output	PCL clock output	P913/INTP4/RXDD5[Note]	–	–	–	74
			P913/INTP4	42	44	56	–
$\overline{\text{RD}}$	Output	Read strobe signal output for external memory	PCT4	–	–	–	99
REGC	–	Connecting capacitor for regulator output stabilization	–	5	10	10	10
$\overline{\text{RESET}}$	Input	External reset input	–	9	14	14	14
RXDD0	Input	Serial receive data input (UARTD0)	P31/INTP7	23	23	26	26
RXDD1		Serial receive data input (UARTD1)	P91/KR7	37	39	44	62
RXDD2		Serial receive data input (UARTD2)	P39/INTP8	–	–	36	36
RXDD3[Note]		Serial receive data input (UARTD3)	P40/SIB0/KR0/INTP14	–	–	–	22
			P80/INTP14	–	–	–	59
RXDD4[Note]		Serial receive data input (UARTD4)	P914/INTP5/SDA00	–	–	–	75
RXDD5[Note]		Serial receive data input (UARTD5)	P913/INTP4/PCL	–	–	–	74

Note μPD70F3757 only

Remark HE3: V850ES/HE3, HF3: V850ES/HF3, HG3: V850ES/HG3, HJ3: V850ES/HJ3

(4/6)

Pin Name	I/O	Function	Alternate Function	Pin No.			
				HE3	HF3	HG3	HJ3
$\overline{\text{SCKB0}}$	I/O	Serial clock I/O (CSIB0)	P42/KR2	21	21	24	24
$\overline{\text{SCKB1}}$		Serial clock I/O (CSIB1)	P99/TIAB00/TOAB00	41	43	52	70
$\overline{\text{SCKB2}}$		Serial clock I/O (CSIB2)	P912/TXDD5Note	–	–	–	73
SCL00	I/O	Serial clock I/O (I^2C00)	P915/INTP6/TXDD4Note	–	–	–	76
			P915/INTP6	44	46	58	–
SDA00	I/O	Serial transmit/receive data I/O (I^2C00)	P914/INTP5/RXDD4Note	–	–	–	75
			P914/INTP5	43	45	57	–
SIB0	Input	Serial receive data input (CSIB0)	P40/KR0/RXDD3Note /INTP14Note	–	–	–	22
			P40/KR0	19	19	22	–
SIB1		Serial receive data input (CSIB1)	P97/TIAA20/TOAA20	39	41	50	68
SIB2		Serial receive data input (CSIB2)	P910	–	–	–	71
SOB0	Output	Serial transmit data output (CSIB0)	P41/KR1/TXDD3Note	–	–	–	23
			P41/KR1	20	20	23	–
SOB1		Serial transmit data output (CSIB1)	P98/TIAB03/TOAB03	40	42	51	69
SOB2		Serial transmit data output (CSIB2)	P911	–	–	–	72
TIAA00	Input	Capture trigger input/external event input/external clock input (TAA0)	P32/ASCKD0/TOAA01/ TOAA00	24	24	27	27
TIAA01		Capture trigger input (TAA0)	P33/TOAA01	25	25	28	28
TIAA10		Capture trigger input/external event input/external clock input (TAA1)	P34/TOAA10	26	26	29	29
TIAA11		Capture trigger input (TAA1)	P35/TOAA11	27	27	30	30
TIAA20		Capture trigger input/external event input/external clock input (TAA2)	P97/SIB1/TOAA20	39	41	50	68
TIAA21		Capture trigger input (TAA2)	P96/TOAA21	38	40	49	67
TIAA30		Capture trigger input/external event input/external clock input (TAA3)	P01/TOAA30	13	4	7	7
TIAA31		Capture trigger input (TAA3)	P00/TOAA31	12	3	6	6
TIAA40		Capture trigger input/external event input/external clock input (TAA4)	P02/NMI/TOAA40	14	5	17	17
TIAA41		Capture trigger input (TAA4)	P03/INTP0/ADTRG/TOAA41	15	6	18	18
TIAB00	Input	Capture trigger input/external event input/external trigger input (TAB0)	P53/KR3/TOAB00 /TOAB0B2/DDO	31	35	40	40
			P99/$\overline{\text{SCKB1}}$/TOAB00	41	43	52	70
TIAB01		Capture trigger input (TAB0)	P50/KR0/TOAB01 /TOAB0T1	28	32	37	37
TIAB02			P51/KR1/TOAB02 /TOAB0B1	29	33	38	38
TIAB03			P52/KR2/TOAB03 /TOAB0T2/DDI	30	34	39	39
			P98/SOB1/TOAB03	40	42	51	69

Note μPD70F3757 only

Remark HE3: V850ES/HE3, HF3: V850ES/HF3, HG3: V850ES/HG3, HJ3: V850ES/HJ3

(5/6)

Pin Name	I/O	Function	Alternate Function	Pin No.			
				HE3	HF3	HG3	HJ3
TIAB10	Input	Capture trigger input/external event input/external trigger input (TAB1)	P95/TOAB10	–	–	48	66
TIAB11		Capture trigger input (TAB1)	P92/TOAB11	–	–	45	63
TIAB12			P93/TOAB12	–	–	46	64
TIAB13			P94/TOAB13	–	–	47	65
TIAB20		Capture trigger input/external event input/external trigger input (TAB2)	P610/TOAB20	–	–	–	53
TIAB21		Capture trigger input (TAB2)	P611/TOAB21	–	–	–	54
TIAB22			P612/TOAB22	–	–	–	55
TIAB23			P613/TOAB23	–	–	–	56
TOAA00	Output	Timer output (TAA0)	P32/ASCKD0/TOAA01 /TIAA00	24	24	27	27
TOAA01			P32/ASCKD0/TIAA00 /TOAA00	24	24	27	27
			P33/TIAA01	25	25	28	28
TOAA10		Timer output (TAA1)	P34/TIAA10	26	26	29	29
TOAA11			P35/TIAA11	27	27	30	30
TOAA20		Timer output (TAA2)	P97/SIB1/TIAA20	39	41	50	68
TOAA21			P96/TIAA21	38	40	49	67
TOAA30	Output	Timer output (TAA3)	P01/TIAA30	13	4	7	7
TOAA31			P00/TIAA31	12	3	6	6
TOAA40		Timer output (TAA4)	P02/NMI/TIAA40	14	5	17	17
TOAA41			P03/INTP0/ADTRG/TIAA41	15	6	18	18
TOAB00	Output	Timer output (TAB0)	P53/KR3/TIAB00 /TOAB0B2/DDO	31	35	40	40
			P99/$\overline{\text{SCKB1}}$/TIAB00	41	43	52	70
TOAB01			P50/KR0/TIAB01/TOAB0T1	28	32	37	37
TOAB02			P51/KR1/TIAB02/TOAB0B1	29	33	38	38
TOAB03			P52/KR2/TIAB03/TOAB0T2 /DDI	30	34	39	39
			P98/SOB1/TIAB03	40	42	51	69
TOAB10		Timer output (TAB1)	P95/TIAB10	–	–	48	66
TOAB11			P92/TIAB11	–	–	45	63
TOAB12			P93/TIAB12	–	–	46	64
TOAB13			P94/TIAB13	–	–	47	65
TOAB20		Timer output (TAB2)	P610/TIAB20	–	–	–	53
TOAB21			P611/TIAB21	–	–	–	54
TOAB22			P612/TIAB22	–	–	–	55
TOAB23			P613/TIAB23	–	–	–	56
TOAB0B1	Output	Motor control output	P51/KR1/TIAB02/TOAB02	29	33	38	38
TOAB0B2			P53/KR3/TIAB00/DDO	31	35	40	40
TOAB0B3			P55/KR5/DMS	35	37	42	42

Remark HE3: V850ES/HE3, HF3: V850ES/HF3, HG3: V850ES/HG3, HJ3: V850ES/HJ3

(6/6)

Pin Name	I/O	Function	Alternate Function	Pin No.			
				HE3	HF3	HG3	HJ3
TOAB0T1	Output	Motor control inverted input	P50/KR0/TIAB01/TOAB01	28	32	37	37
TOAB0T2			P52/KR2/TIAB03/TOAB03	30	34	39	39
TOAB0T3			P54/KR4/DCK	34	36	41	41
TXDD0	Output	Serial transmit data output (UARTD0)	P30	22	22	25	25
TXDD1		Serial transmit data output (UARTD1)	P90/KR6	36	38	43	61
TXDD2		Serial transmit data output (UARTD2)	P38	–	–	35	35
TXDD3[Note]		Serial transmit data output (UARTD3)	P41/SOB0/KR1	–	–	–	23
			P81	–	–	–	60
TXDD4[Note]		Serial transmit data output (UARTD4)	P915/INTP6/SCL00	–	–	–	76
TXDD5[Note]		Serial transmit data output (UARTD5)	P912/$\overline{\text{SCKB2}}$	–	–	–	73
V_{DD}	–	Positive power supply for internal circuit	–	4	9	9	9, 104
V_{SS}	–	Ground potential for internal circuit	–	6	11	11	11, 103
$\overline{\text{WAIT}}$	Input	External wait input	PCM0	–	–	–	85
$\overline{\text{WR0}}$	Output	Write strobe for external memory (lower 8 bits)	PCT0	–	–	–	95
$\overline{\text{WR1}}$		Write strobe for external memory (higher 8 bits)	PCT1	–	–	–	96
X1	Input	Connecting resonator for main clock	–	7	12	12	12
X2	–		–	8	13	13	13
XT1	Input	Connecting resonator for subclock	–	10	15	15	15
XT2	–		–	11	16	16	16

Note μPD70F3757 only

Remark HE3: V850ES/HE3, HF3: V850ES/HF3, HG3: V850ES/HG3, HJ3: V850ES/HJ3

2.2 Pin Status

The V850ES/HJ3 has an external bus interface function that enables connection of external memories, such as ROM and RAM, and I/O.

Table 2-5 shows the operating status of each external bus interface pin in each operation mode.

Table 2-5. Pin Operating Status in Each Operation Mode

Bus Control Pin	Reset	HALT Mode and DMA Transfer	IDLE1, IDLE2, and STOP Modes	Idle State[Note 2]	Bus Hold
AD0 to AD15	Hi-Z[Note 1]	Operating	Hi-Z	Held	Hi-Z
$\overline{CS0}$ to $\overline{CS3}$			H		
$\overline{WAIT}$			–	–	–
CLKOUT			L	Operating	Operating
$\overline{WR0}$, $\overline{WR1}$			H	H	Hi-Z
RD					
ASTB					
$\overline{HLDAK}$					L
$\overline{HLDRQ}$			–	–	Operating

Notes 1. The bus control pins function alternately as port pins and are initialized to the input mode (port mode).

2. Pin status in the idle state that is inserted after the T3 state.

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

2.3 Pin I/O Circuit Types and Recommended Connection of Unused Pins

(1/4)

Pin Name	Alternate Function	I/O Circuit Type	Recommended Connection of Unused Pins	Pin No.			
				HE3	HF3	HG3	HJ3
P00	TIAA31/TOAA31	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.	12	3	6	6
P01	TIAA30/TOAA30			13	4	7	7
P02	NMI/TIAA40/TOAA40			14	5	17	17
P03	INTP0/ADTRG/TIAA41/TOAA41			15	6	18	18
P04	INTP1			16	7	19	19
P05	INTP2/$\overline{\text{DRST}}$	5-AF	Input: Independently connect to EV_{SS} via a resistor. Output: Leave open.	17	17	20	20
P06	INTP3	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.	18	18	21	21
P10	INTP9			–	–	3	3
P11	INTP10			–	–	4	4
P30	TXDD0			22	22	25	25
P31	RXDD0/INTP7			23	23	26	26
P32	ASCKD0/TOAA01/TIAA00/TOAA00			24	24	27	27
P33	TIAA01/TOAA01			25	25	28	28
P34	TIAA10/TOAA10			26	26	29	29
P35	TIAA11/TOAA11			27	27	30	30
P36	–			–	–	31	31
P37	–			–	–	32	32
P38	TXDD2			–	–	35	35
	–			–	28	–	–
P39	RXDD2/INTP8			–	–	36	36
	–			–	29	–	–
P40	SIB0/KR0/RXDD3[Note]/INTP14[Note]			–	–	–	22
	SIB0/KR0			19	19	22	–
P41	SOB0/KR1/TXDD3			–	–	–	23
	SOB0/KR1			20	20	23	–
P42	$\overline{\text{SCKB0}}$/KR2			21	21	24	24
P50	KR0/TIAB01/TOAB01/TOAB0T1			28	32	37	37
P51	KR1/TIAB02/TOAB02/TOAB0B1			29	33	38	38
P52	KR2/TIAB03/TOAB03/TOAB0T2/DDI			30	34	39	39
P53	KR3/TIAB00/TOAB00/TOAB0B2/DDO			31	35	40	40
P54	KR4/TOAB0T3/DCK			34	36	41	41
P55	KR5/TOAB0B3/DMS			35	37	42	42

Note μPD70F3755 only

Remark HE3: V850ES/HE3, HF3: V850ES/HF3, HG3: V850ES/HG3, HJ3: V850ES/HJ3

(2/4)

Pin Name	Alternate Function	I/O Circuit Type	Recommended Connection of Unused Pins	Pin No.			
				HE3	HF3	HG3	HJ3
P60 to P62	NTP11 to INTP13	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.	–	–	–	43 to 45
P63 to P69	–			–	–	–	46 to 52
P610 to P613	TIAB20/TOAB20 to TIAB23/TOAB23			–	–	–	53 to 56
P614	–			–	–	–	57
P615	–			–	–	–	58
P70 to P79	ANI0 to ANI9	11-G	Input: Independently connect to AV_{REF0} or AV_{SS} via a resistor. Output: Leave open.	64 to 55	80 to 71	100 to 91	144 to 135
P710	ANI10			–	70	90	134
P711	ANI11			–	69	89	133
P712 to P715	ANI12 to ANI15			–	–	88 to 85	132 to 129
P80	RXDD3/INTP14	5-W	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.	–	–	–	59
P81	TXDD3			–	–	–	60
P90	KR6/TXDD1			36	38	43	61
P91	KR7/RXDD1			37	39	44	62
P92 to P94	TIAB11/TOAB11 to TIAB13/TOAB13			–	–	45 to 47	63 to 65
P95	TIAB10/TOAB10			–	–	48	66
P96	TIAA21/TOAA21			38	40	49	67
P97	SIB1/TIAA20/TOAA20			39	41	50	68
P98	SOB1/TIAB03/TOAB03			40	42	51	69
P99	$\overline{SCKB1}$/TIAB00/TOAB00			41	43	52	70
P910	SIB2			–	–	–	71
	–			–	–	53	–
P911	SOB2			–	–	–	72
	–			–	–	54	–
P912	$\overline{SCKB2}$/TXDD5			–	–	–	73
	–			–	–	55	–
P913	INTP4/PCL/RXDD5			–	–	–	74
	INTP4/PCL			42	44	56	–
P914	INTP5/SDA00/RXDD4			–	–	–	75
	INTP5/SDA00			43	45	57	–
P915	INTP6/SCL00/TXDD4			–	–	–	76
	INTP6/SCL00			44	46	58	–

Remark HE3: V850ES/HE3, HF3: V850ES/HF3, HG3: V850ES/HG3, HJ3: V850ES/HJ3

(3/4)

Pin Name	Alternate Function	I/O Circuit Type	Recommended Connection of Unused Pins	Pin No.			
				HE3	HF3	HG3	HJ3
P120 to P127	ANI16 to ANI23	11-G	Input: Independently connect to AV_{REF0} or AV_{SS} via a resistor. Output: Leave open.	–	–	–	128 to 121
PCD0 to PCD3	–	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor. Output: Leave open.	–	–	–	77 to 80
PCM0	$\overline{WAIT}$	5	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor (connect the V850ES/HE3 and V850ES/HF3 to EV_{DD} or EV_{SS}). Output: Leave open.	–	–	–	85
	–			45	49	61	–
PCM1	CLKOUT			46	50	62	86
PCM2	$\overline{HLDAK}$			–	–	–	87
	–			–	51	63	–
PCM3	$\overline{HLDRQ}$			–	–	–	88
	–			–	52	64	–
PCM4	–			–	–	–	89
PCM5	–			–	–	–	90
PCS0	$\overline{CS0}$		Input: Independently connect to BV_{DD} or BV_{SS} via a resistor (connect the V850ES/HF3 to EV_{DD} or EV_{SS}). Output: Leave open.	–	–	–	81
	–			–	47	59	–
PCS1	$\overline{CS1}$			–	–	–	82
	–			–	48	60	–
PCS2	$\overline{CS2}$			–	–	–	83
PCS3	$\overline{CS3}$			–	–	–	84
PCS4 to PCS7	–			–	–	–	92 to 94
PCT0	$\overline{WR0}$			–	–	–	95
	–			–	53	65	–
PCT1	$\overline{WR1}$			–	–	–	96
	–			–	54	66	–
PCT2	–			–	–	–	97
PCT3	–			–	–	–	98
PCT4	$\overline{RD}$			–	–	–	99
	–			–	55	67	–
PCT5	–			–	–	–	100
PCT6	ASTB			–	–	–	101
	–			–	56	68	–
PCT7	–			–	–	–	102

Remark HE3: V850ES/HE3, HF3: V850ES/HF3, HG3: V850ES/HG3, HJ3: V850ES/HJ3

(4/4)

Pin Name	Alternate Function	I/O Circuit Type	Recommended Connection of Unused Pins	Pin No.			
				HE3	HF3	HG3	HJ3
PDL0 to PDL4	AD0 to AD4	5-K	Input: Independently connect to BV_{DD} or BV_{SS} via a resistor (connect the V850ES/HE3 and V850ES/HF3 to EV_{DD} or EV_{SS}).	–	–	–	105 to 109
	–		Output: Leave open.	47 to 51	57 to 61	71 to 75	–
PDL5	AD5/FLMD1			–	–	–	110
	FLMD1			52	62	76	–
PDL6	AD6			–	–	–	111
	–			53	67	77	–
PDL7	AD7			–	–	–	112
	–			54	64	78	–
PDL8 to PDL11	AD8 to AD11			–	–	–	113 to 116
	–			–	65 to 68	79 to 82	–
PDL12	AD12			–	–	–	117
	–			–	–	83	–
PDL13	AD13			–	–	–	118
	–			–	–	84	–
PDL14	AD14			–	–	–	119
PDL15	AD15			–	–	–	120
AV_{REF0}	–	–	Directly connect to V_{DD}.	1	1	1	1
AV_{SS}	–	–	–	2	2	2	2
BV_{DD}	–	–	–	–	–	70	104
BV_{SS}	–	–	–	–	–	69	103
EV_{DD}	–	–	–	33	31	5, 34	5, 34
EV_{SS}	–	–	–	32	30	33	33
FLMD0	–	–	Connect to V_{SS} in other than flash mode.	3	8	8	8
REGC	–	–	Connect to regulator output stabilization capacitor.	5	10	10	10
$\overline{RESET}$	–	2	Connect to EV_{DD} via a resistor.	9	14	14	14
V_{DD}	–	–	–	4	9	9	9, 104
V_{SS}	–	–	–	6	11	11	11, 103
X1	–	–	–	7	12	12	12
X2	–	–	–	8	13	13	13
XT1	–	16	Connect to V_{SS} via a resistor.	10	15	15	15
XT2	–		Leave open.	11	16	16	16

Remark HE3: V850ES/HE3, HF3: V850ES/HF3, HG3: V850ES/HG3, HJ3: V850ES/HJ3

2.4 Pin I/O Circuits

Figure 2-1. Pin I/O Circuit Types (1/2)

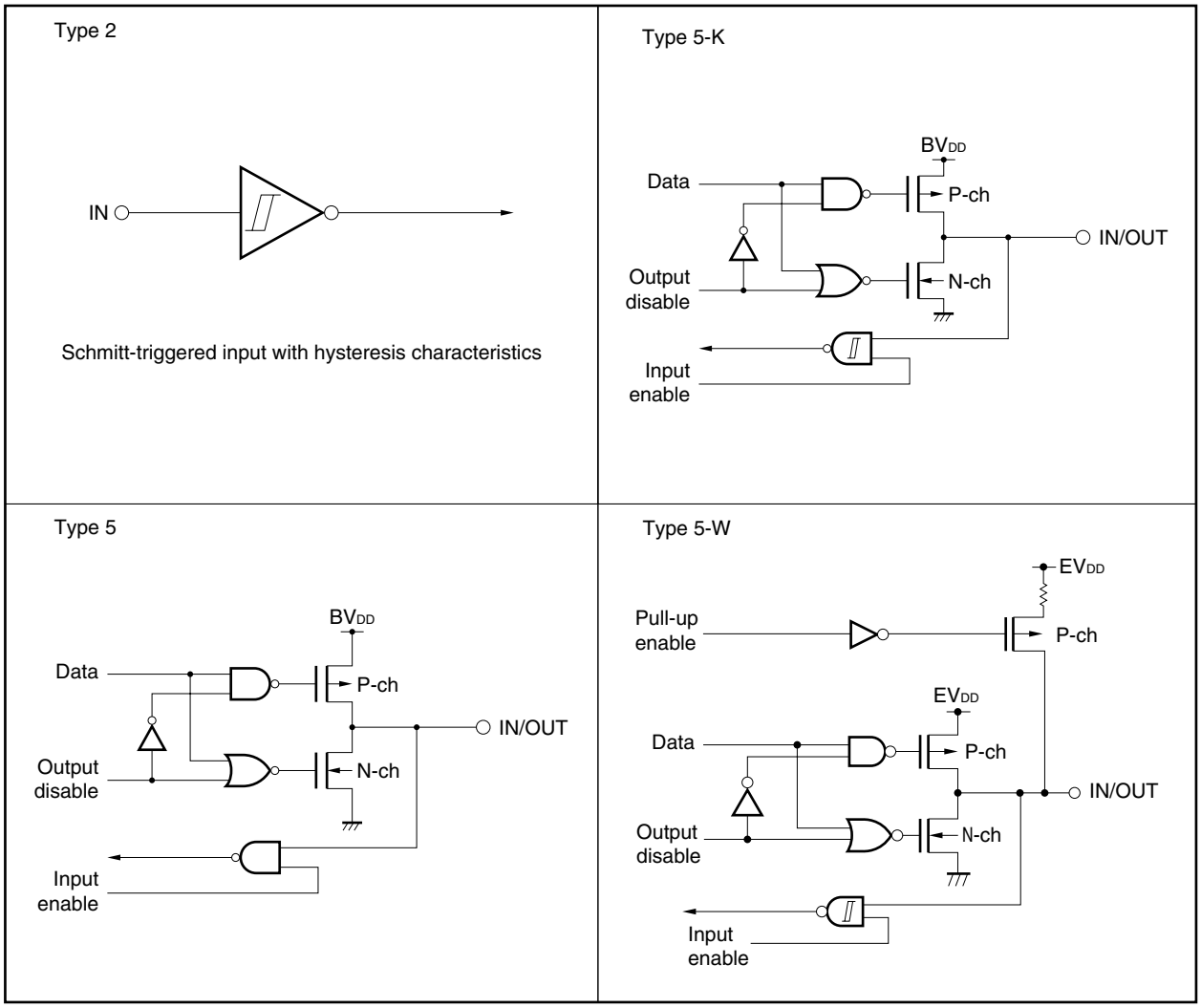
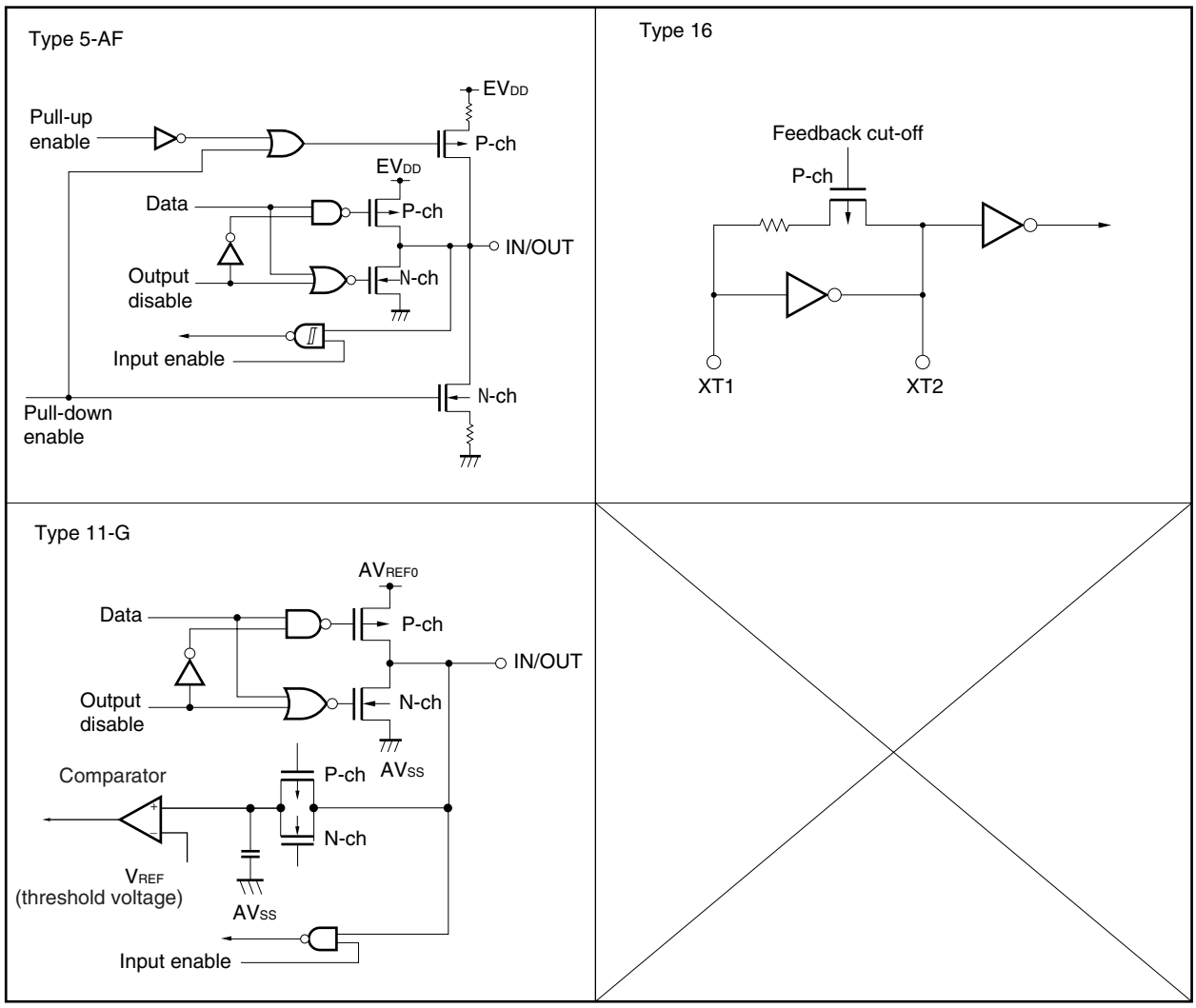


Figure 2-1. Pin I/O Circuit Types (2/2)



CHAPTER 3 CPU FUNCTION

The CPU of the V850ES/Hx3 is based on RISC architecture and executes almost all instructions with one clock by using a 5-stage pipeline.

3.1 Features

○ Minimum instruction execution time: 31.25 ns (at 32 MHz operation)
○ Variable instruction length (16-bit/32-bit length)
○ Memory space Program space: 64 MB linear
Data space: 4 GB linear
○ General-purpose registers: 32 bits × 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/Hx3 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

General-purpose registers

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

31		0
PC	(Program counter)	

(2) System register set

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

3.2.1 Program register set

The program 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

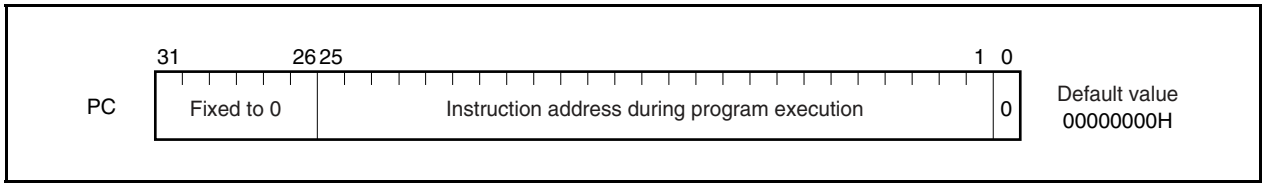
Remark For furthers details on the r1, r3 to r5, and r31 that are used in the assembler and C compiler, refer to the **CA850 (C Compiler Package) Assembly Language User's Manual**.

(2) Program counter (PC)

The program counter holds the instruction address during program execution. The lower 26 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.

Therefore, the highest address of the program space, 03FFFFFFH, and the lowest address, 00000000H, are contiguous addresses.

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 the DBRET instruction.

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).

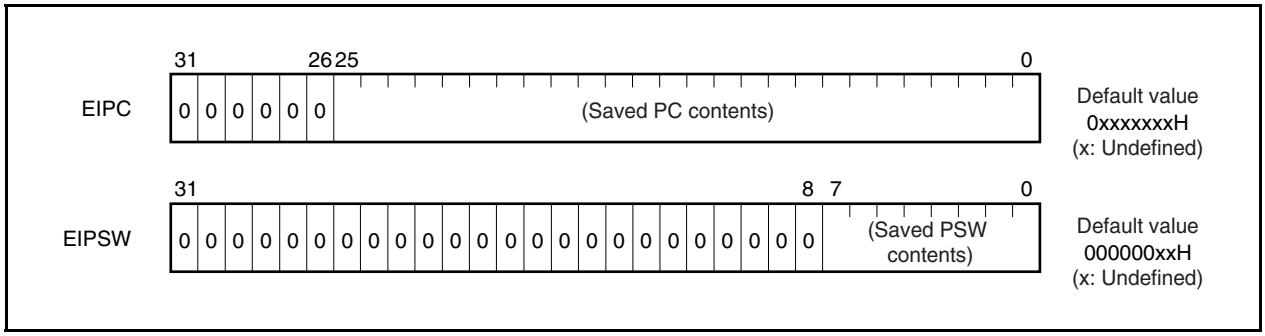
The address of the instruction next to the instruction under execution, except some instructions (see **18.8 Periods in Which Interrupts Are Not Acknowledged by CPU**), is saved to EIPC when a software exception or a maskable interrupt occurs.

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.

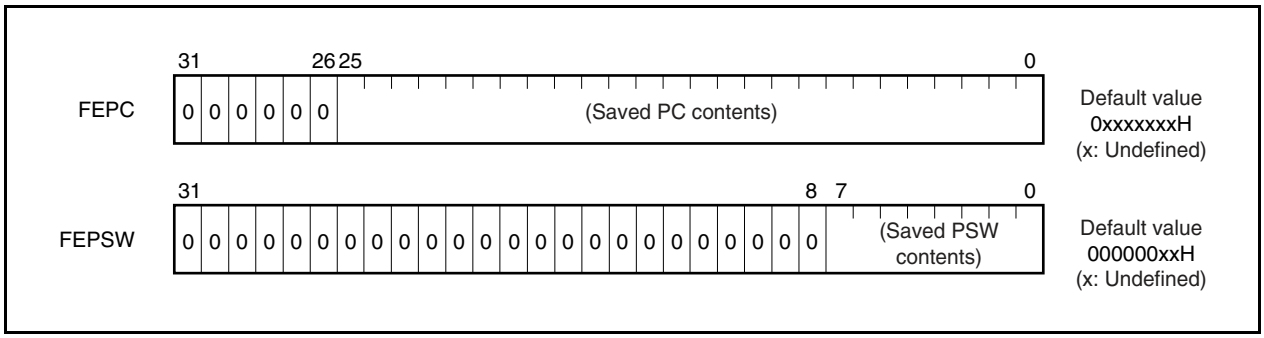
The address of the instruction next to the one of the instruction under execution, except some instructions, is saved to FEPC when an NMI occurs.

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.

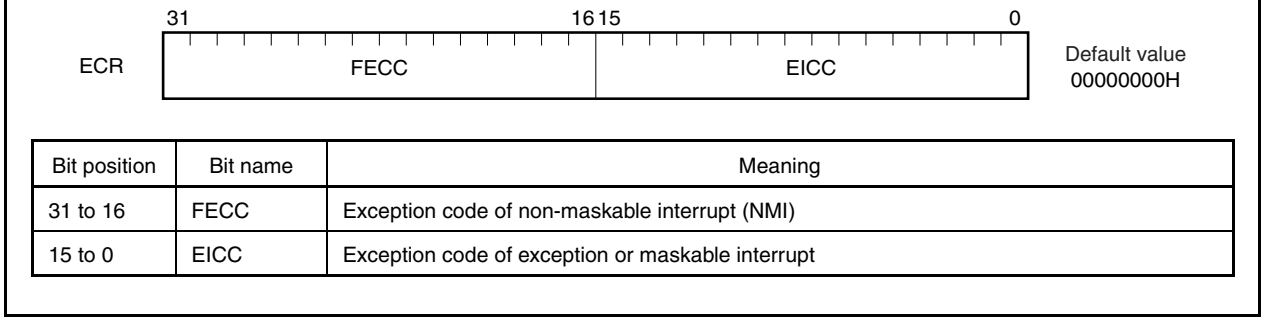
Bits 31 to 26 of FEPC and bits 31 to 8 of FEPSW are reserved for future function expansion (these bits are always fixed to 0).

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.



Bit position	Bit name	Meaning
31 to 16	FECC	Exception code of non-maskable interrupt (NMI)
15 to 0	EICC	Exception code of exception or maskable interrupt

(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)

PSW

31																							NP	EP	ID	SAT	CY	OV	S	Z
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								

8 7 6 5 4 3 2 1 0

Default value 00000020H

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)

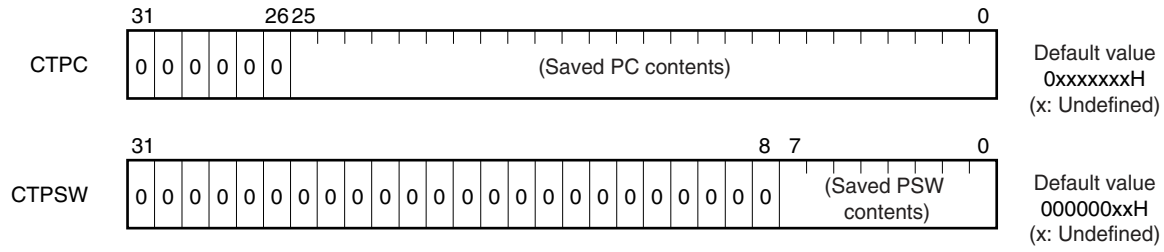
CTPC and CTPSW are CALLT execution status saving registers.

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.

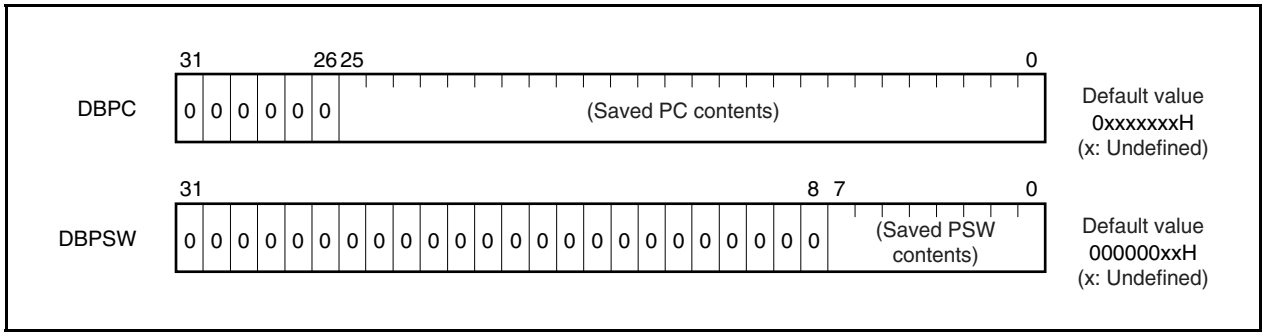
The contents to be saved to DBPC are the address of the instruction next to the one that is being executed when an exception trap or debug trap occurs.

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).

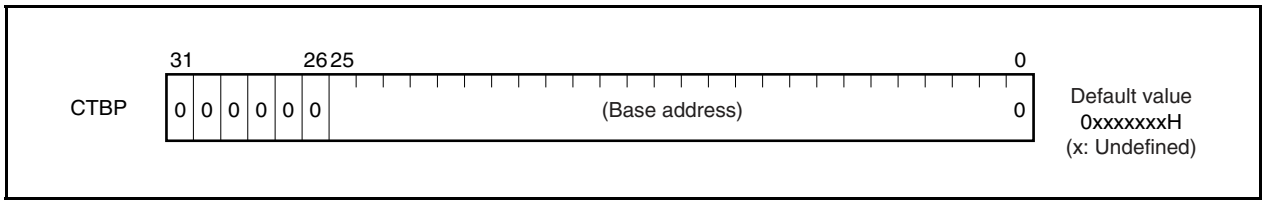
The value of DBPC is restored to the PC and the value of DBPSW to the PSW by the DBRET instruction.



(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/Hx3 has the following operation modes.

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

The normal operation mode or flash memory programming mode is specified according to the status (input level) of the FLMD0 and FLMD1 pins.

In the normal mode, input a low level to the FLMD0 pin when reset is released.

In the flash memory programming mode, a high level is input to the FLMD0 pin from the flash programmer if a flash programmer is connected. In the self-programming mode, however, input a high level to the FLMD0 pin by controlling the port before rewriting the flash memory after the flash memory has operated in the normal operation mode.

Operation When Reset Is Released		Operation Mode After Reset
FLMD0	FLMD1	
L	×	Normal operation mode
H	L	Flash memory programming mode
H	H	Setting prohibited

Remark L: Low-level input
H: High-level input
×: Don't care

(1) Normal operation mode

In this mode, each pin related to the bus interface is set to the port mode after system reset has been released. Execution branches to the reset entry address of the internal ROM, and then instruction processing is started.

(2) Flash memory programming mode

In this mode, the internal flash memory can be programmed by using a flash programmer.

(3) Self-programming mode

In this mode, the internal flash memory can be erased or written by a user application. For details, see **CHAPTER 26 FLASH MEMORY**.

(4) On-chip debug mode

The V850ES/Hx3 is provided with an on-chip debug function that employs the JTAG (Joint Test Action Group) communication specifications.

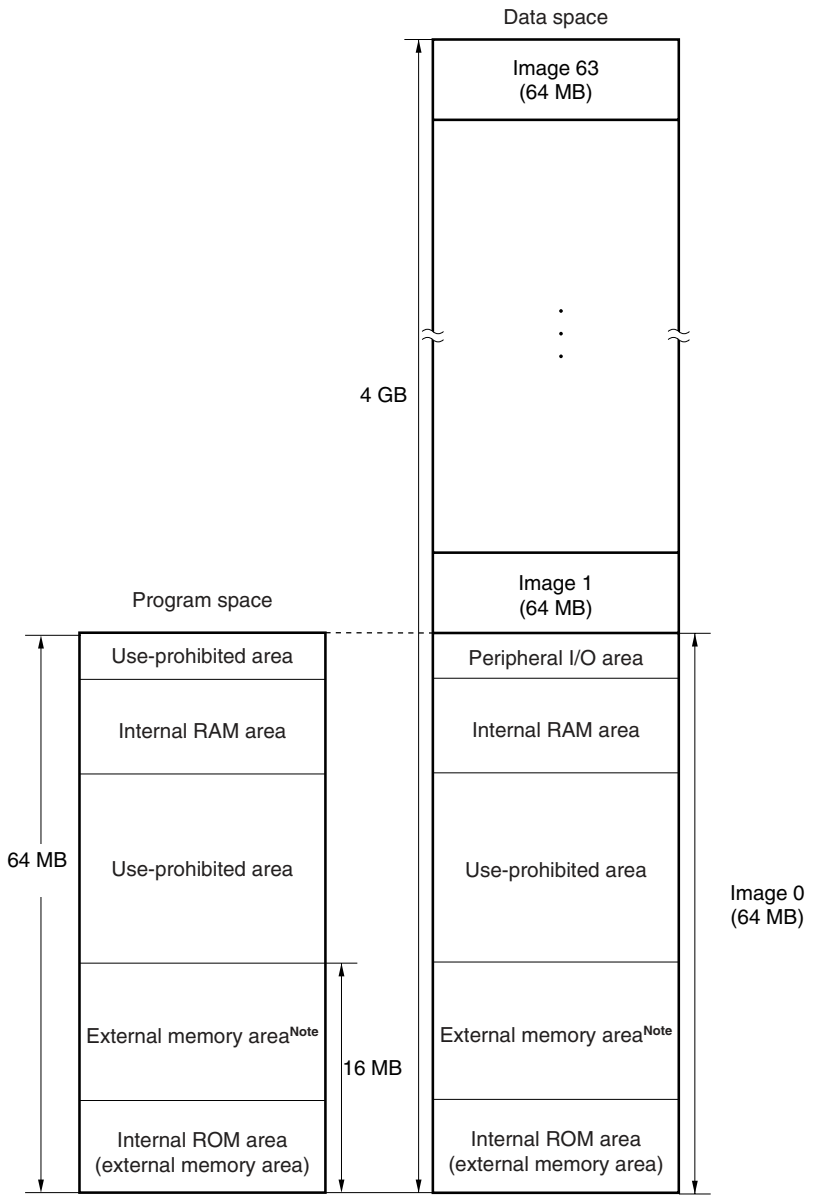
For details, see **CHAPTER 28 ON-CHIP DEBUG FUNCTION**.

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. Image on Address Space



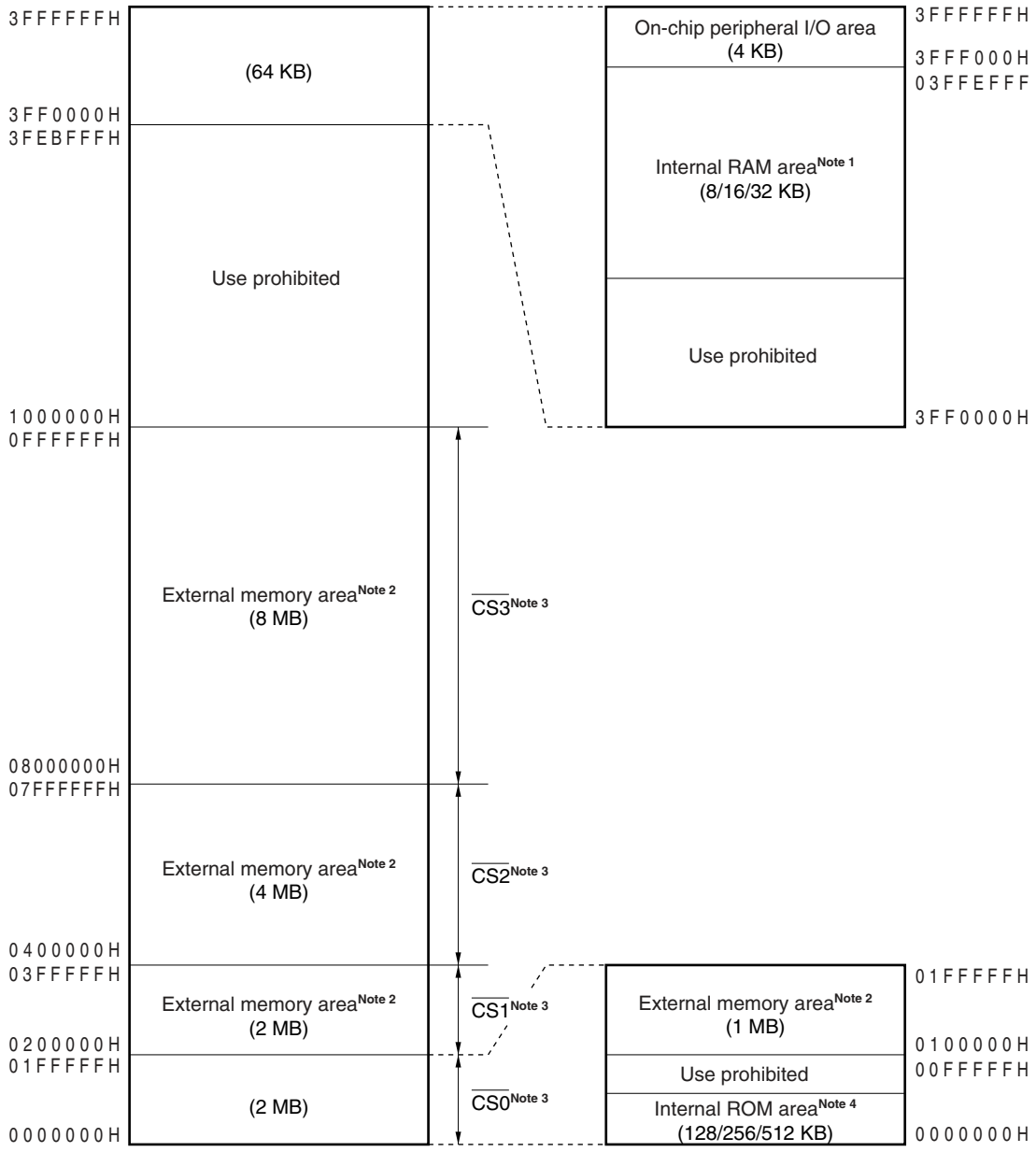
Note The external memory area is available only in the V850ES/HJ3, and is a use-prohibited area in other products

Remark The same image as image 0 appears on images 1 to 63 of the data space.

3.4.2 Memory map

The areas shown below are reserved in the V850ES/Hx3.

Figure 3-2. Data Memory Map



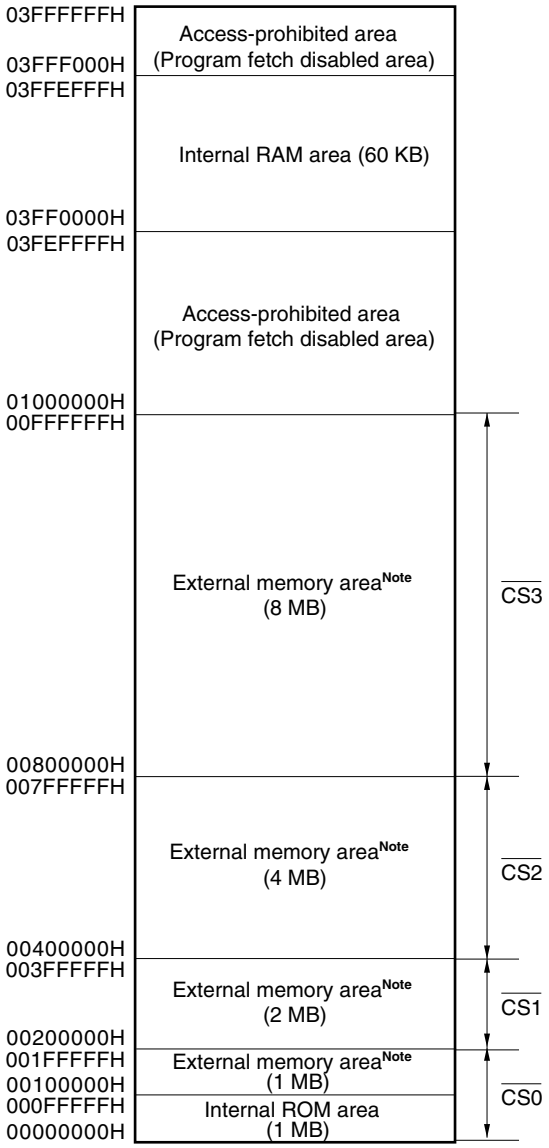
Notes 1. RAM size is different for each product (see **Table 1-1**).

2. The external memory area is available only in the V850ES/HJ3, and is a use-prohibited area in other products. However, because the number of address bus lines is 16 (AD0 to AD15), an image of 64 KB repeatedly appears if addresses of 64 KB are specified.

3. $\overline{CS0}$ to $\overline{CS3}$ are available only in the V850ES/HJ3, and are use-prohibited areas in other products.

4. ROM (flash memory) size is different for each product (see **Table 1-1**).
Fetch access and read access to addresses 00000000H to 000FFFFFH is made to the internal ROM area. However, data write access to these addresses is made to the external memory area.

Figure 3-3. Program Memory Map



Note The external memory area is available only in the V850ES/HJ3, and is a use-prohibited area in other products

3.4.3 Areas

(1) Internal ROM area

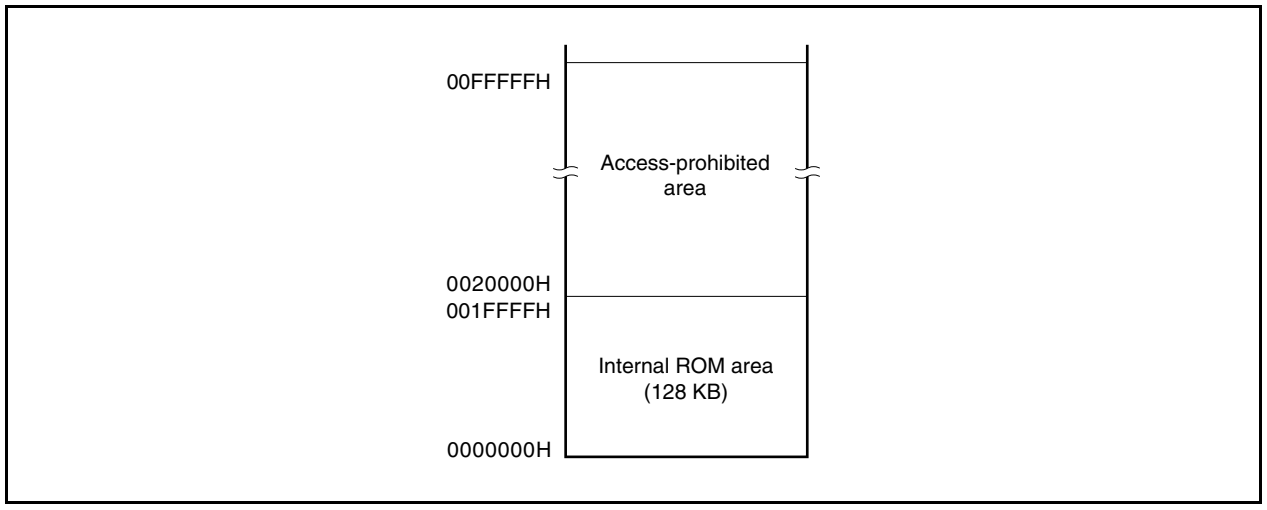
Up to 1 MB is reserved as an internal ROM area.

(a) Internal ROM (128 KB)

128 KB are allocated to addresses 0000000H to 001FFFFH in the μPD70F3747.

Accessing addresses 0020000H to 00FFFFFH is prohibited.

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

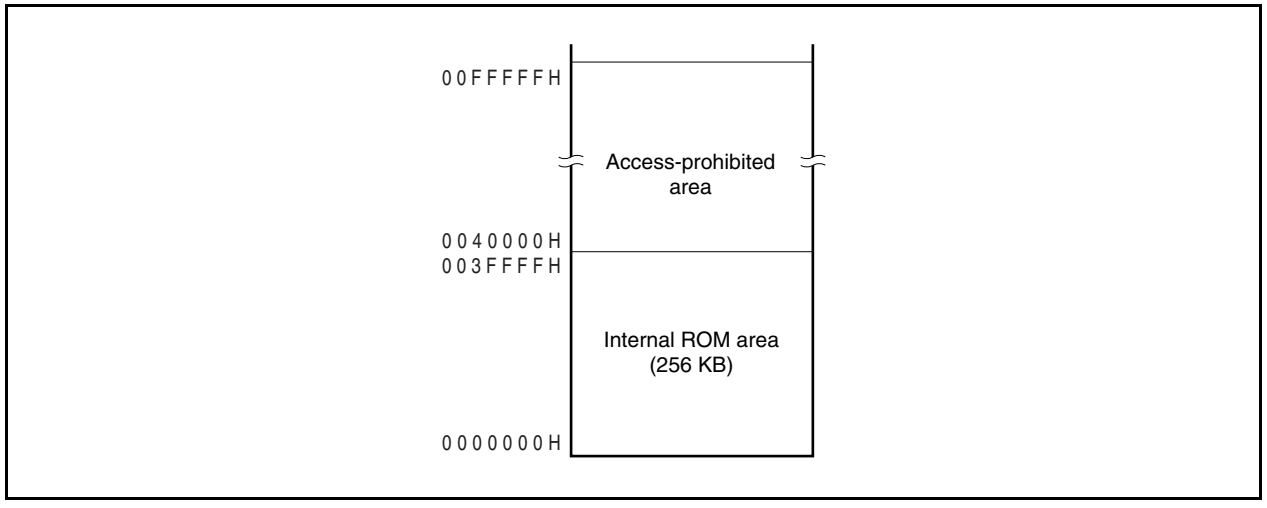


(b) Internal ROM (256 KB)

256 KB are allocated to addresses 00000000H to 0003FFFFH in the μPD70F3750, 70F3752, and 70F3755.

Accessing addresses 0040000H to 00FFFFFH is prohibited.

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

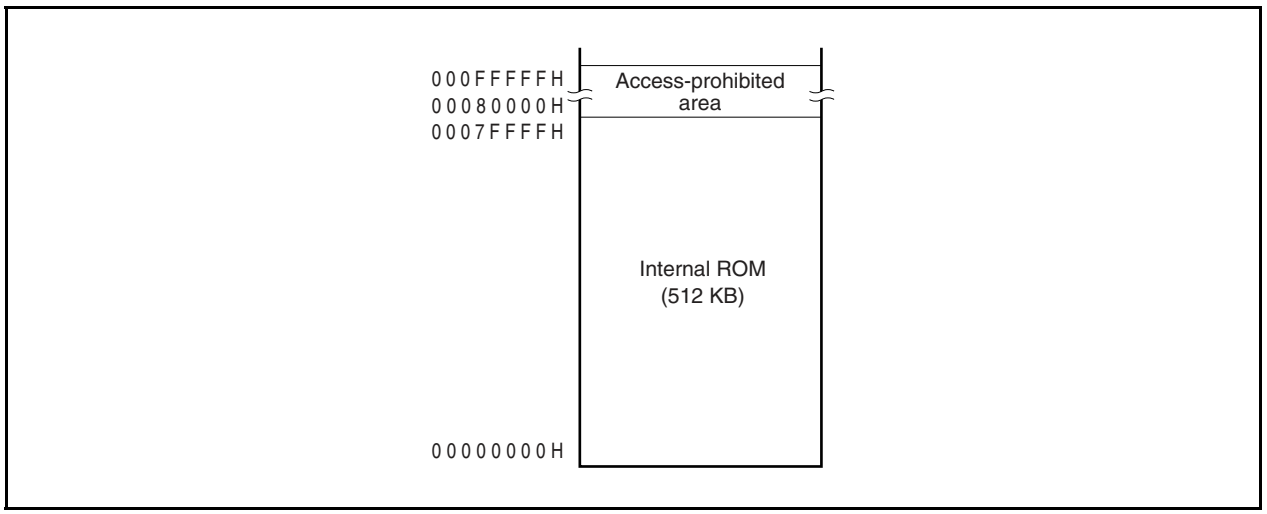


(c) ROM (512 KB)

512 KB are allocated to addresses 00000000H to 0007FFFFH in the μPD70F3757.

Accessing addresses 00080000H to 000FFFFFH is prohibited.

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



(2) Internal RAM area

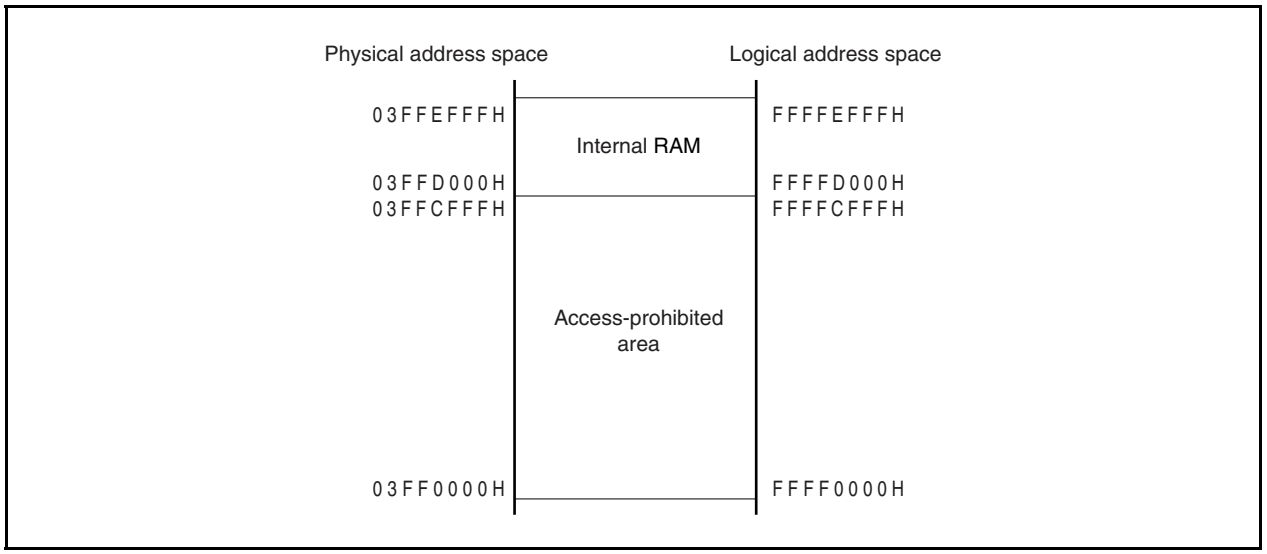
Up to 60 KB are reserved as the internal RAM area.

(a) Internal RAM (8 KB)

8 KB are allocated to addresses 03FFD000H to 03FFEFFFH in the μPD70F3747.

Accessing addresses 03FF0000H to 03FFCFFFH is prohibited.

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

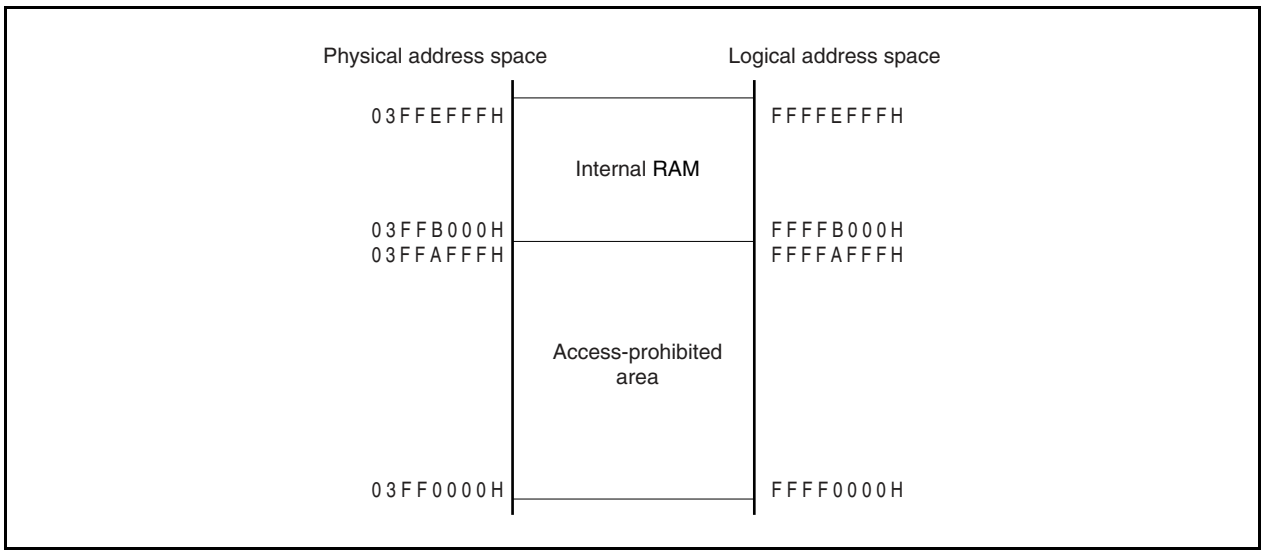


(b) Internal RAM (16 KB)

16 KB are allocated to addresses 03FFB000H to 03FFEFFFH in the following versions.
Accessing addresses 03FF0000H to 03FF9FFFH is prohibited.

- μPD70F3750, 70F3752, 70F3755

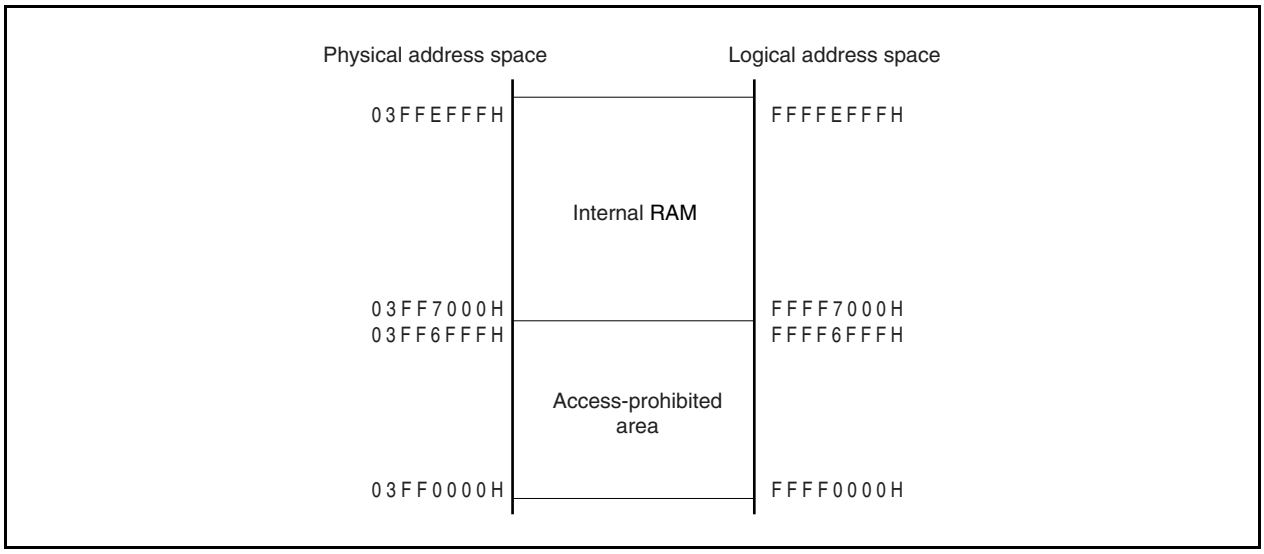
Figure 3-8. Internal RAM Area (16 KB)



(c) Internal RAM (32 KB)

32 KB are allocated to addresses 03FF7000H to 03FFEFFFH in the μPD70F3757.
Accessing addresses 03FF0000H to 03FF6FFFH is prohibited.

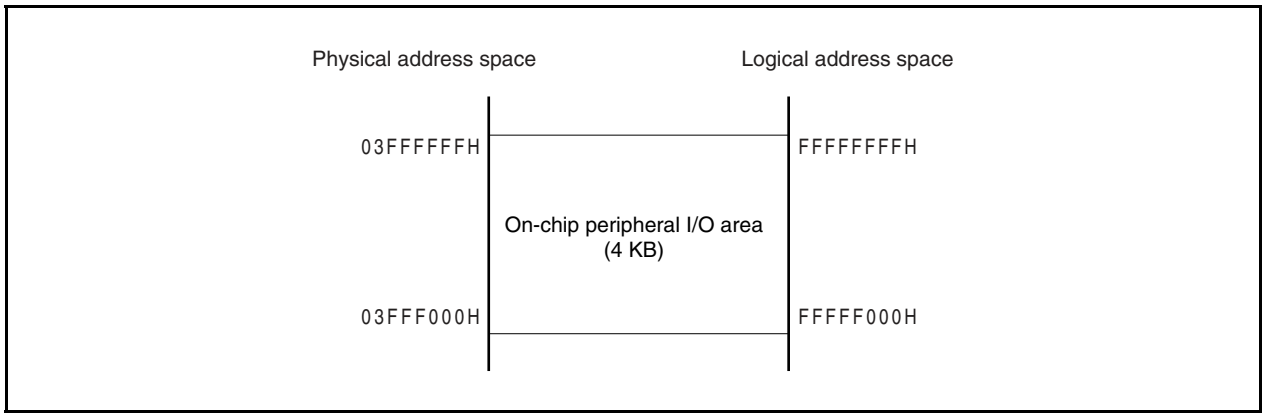
Figure 3-9. Internal RAM Area (32 KB)



(3) On-chip peripheral I/O area

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

Figure 3-10. 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 and 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 higher 8 bits are undefined when the peripheral I/O register is read, and data is written to the lower 8 bits.

3. Addresses not defined as peripheral I/O registers are reserved for future expansion. The operation is undefined and not guaranteed when these addresses are accessed.

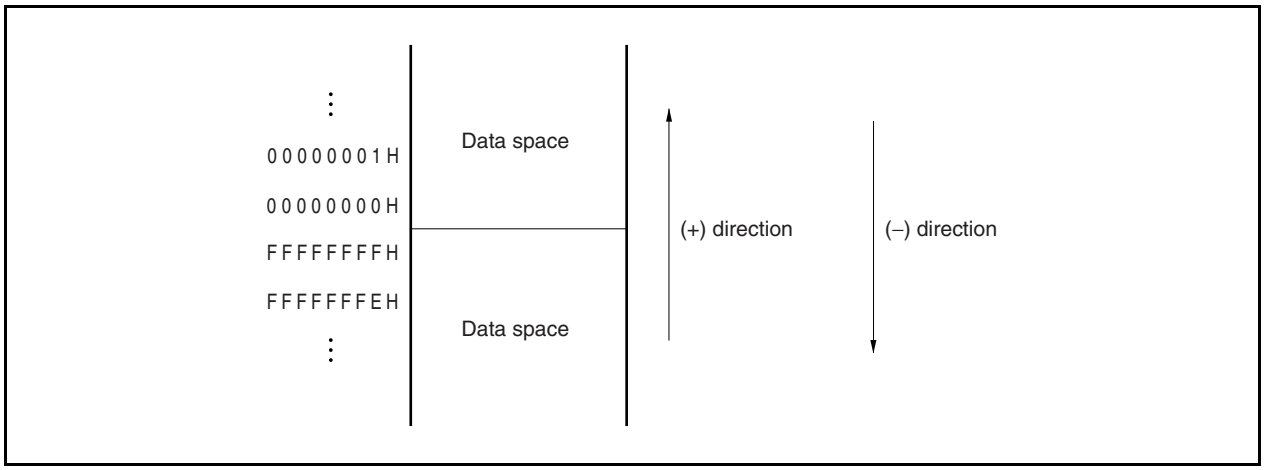
(4) External memory area (V850ES/HJ3 only)

15 MB are allocated as the external memory area. For details, see **CHAPTER 5 BUS CONTROL FUNCTION**.

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.



3.4.5 Recommended use of address space

The architecture of the V850ES/Hx3 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 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.

RAM Size	Access Address
8 KB	03FFD000H to 03FFEFFFH
16 KB	03FFB000H to 03FFEFFFH
32 KB	03FF7000H to 03FFEFFFH

(2) Data space

With the V850ES/Hx3, it seems that there are sixty-four 64 MB address spaces on the 4 GB CPU address space. Therefore, the least significant bit (bit 25) of a 26-bit address is sign-extended to 32 bits and allocated as an address.

(a) Application example of wraparound

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

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

Figure 3-11. Wraparound (μPD70F3755)

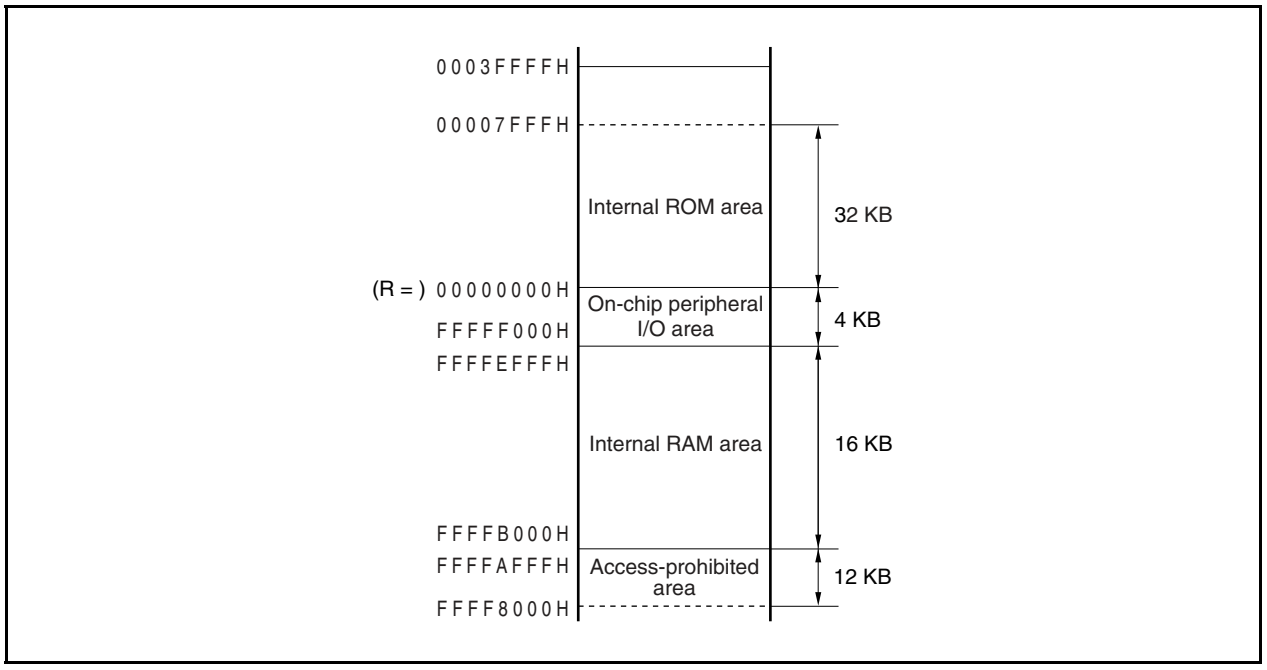
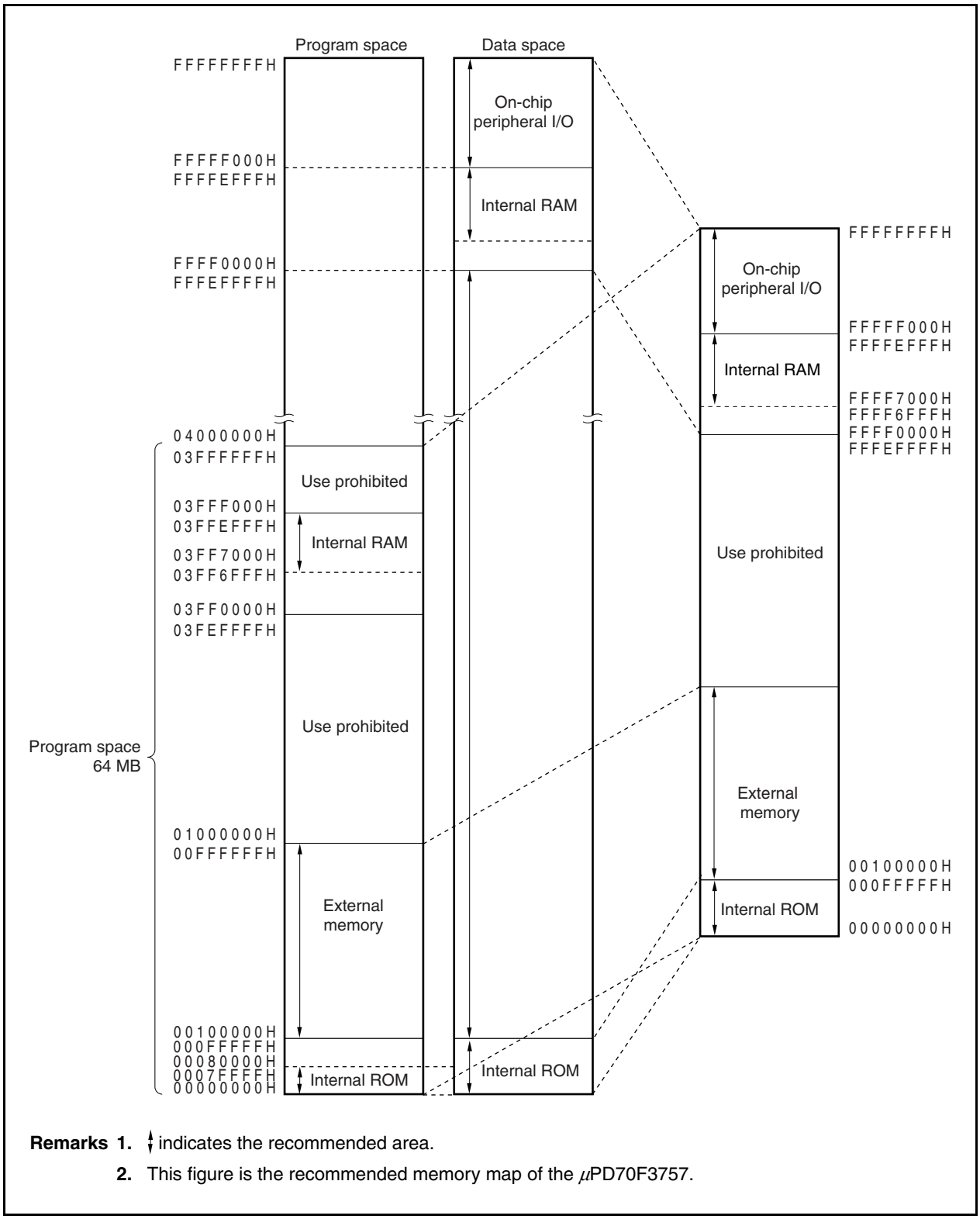


Figure 3-12. Recommended Memory Map



Remarks 1. ↕ indicates the recommended area.

2. This figure is the recommended memory map of the μPD70F3757.

3.4.6 Peripheral I/O registers

(1/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value	HE3	HF3	HG3	HJ3
				1	8	16					
FFFFF004H	Port DL	PDL	R/W			√	Undefined	–	√	√	√
FFFFF004H	Port DLL	PDLL		√	√		Undefined	√	√	√	√
FFFFF005H	Port DLH	PDLH		√	√		Undefined	–	√	√	√
FFFFF008H	Port CS	PCS		√	√		Undefined	–	√	√	√
FFFFF00AH	Port CT	PCT		√	√		Undefined	–	√	√	√
FFFFF00CH	Port CM	PCM		√	√		Undefined	√	√	√	√
FFFFF00EH	Port CD	PCD		√	√		Undefined	–	–	–	√
FFFFF024H	Port mode register DL	PMDL				√	FFFFH	–	√	√	√
FFFFF024H	Port mode register DLL	PMDLL		√	√		FFH	√	√	√	√
FFFFF025H	Port mode register DLH	PMDLH		√	√		FFH	–	√	√	√
FFFFF028H	Port mode register CS	PMCS		√	√		FFH	–	√	√	√
FFFFF02AH	Port mode register CT	PMCT		√	√		FFH	–	√	√	√
FFFFF02CH	Port mode register CM	PMCM		√	√		FFH	√	√	√	√
FFFFF02EH	Port mode register CD	PMCD		√	√		FFH	–	–	–	√
FFFFF044H	Port mode control register DL	PMCDL				√	0000H	–	√	√	√
FFFFF044H	Port mode control register DLL	PMCDLL		√	√		00H	√	√	√	√
FFFFF045H	Port mode control register DLH	PMCDLH		√	√		00H	–	√	√	√
FFFFF048H	Port mode control register CS	PMCCS		√	√		00H	–	√	√	√
FFFFF04AH	Port mode control register CT	PMCCT		√	√		00H	–	√	√	√
FFFFF04CH	Port mode control register CM	PMCCM		√	√		00H	√	√	√	√
FFFFF066H	Bus size configuration register	BSC				√	5555H	–	–	–	√
FFFFF06EH	System wait control register	VSWC			√		77H	√	√	√	√
FFFFF080H	DMA source address register 0L	DSA0L				√	Undefined	√	√	√	√
FFFFF082H	DMA source address register 0H	DSA0H				√	Undefined	√	√	√	√
FFFFF084H	DMA destination address register 0L	DDA0L				√	Undefined	√	√	√	√
FFFFF086H	DMA destination address register 0H	DDA0H				√	Undefined	√	√	√	√
FFFFF088H	DMA source address register 1L	DSA1L				√	Undefined	√	√	√	√
FFFFF08AH	DMA source address register 1H	DSA1H				√	Undefined	√	√	√	√
FFFFF08CH	DMA destination address register 1L	DDA1L				√	Undefined	√	√	√	√
FFFFF08EH	DMA destination address register 1H	DDA1H				√	Undefined	√	√	√	√
FFFFF090H	DMA source address register 2L	DSA2L				√	Undefined	√	√	√	√
FFFFF092H	DMA source address register 2H	DSA2H				√	Undefined	√	√	√	√
FFFFF094H	DMA destination address register 2L	DDA2L				√	Undefined	√	√	√	√
FFFFF096H	DMA destination address register 2H	DDA2H				√	Undefined	√	√	√	√
FFFFF098H	DMA source address register 3L	DSA3L				√	Undefined	√	√	√	√
FFFFF09AH	DMA source address register 3H	DSA3H				√	Undefined	√	√	√	√
FFFFF09CH	DMA destination address register 3L	DDA3L				√	Undefined	√	√	√	√
FFFFF09EH	DMA destination address register 3H	DDA3H				√	Undefined	√	√	√	√
FFFFF0C0H	DMA transfer count register 0	DBC0				√	Undefined	√	√	√	√

(2/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value	HE3	HF3	HG3	HJ3
				1	8	16					
FFFFF0C2H	DMA transfer count register 1	DBC1	R/W			√	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	√	√	√	√
FFFFF102H	Interrupt mask register 1	IMR1					FFFFH	√	√	√	√
FFFFF102H	Interrupt mask register 1L	IMR1L		√	√		FFH	√	√	√	√
FFFFF103H	Interrupt mask register 1H	IMR1H		√	√		FFH	√	√	√	√
FFFFF104H	Interrupt mask register 2	IMR2					FFFFH	√	√	√	√
FFFFF104H	Interrupt mask register 2L	IMR2L		√	√		FFH	√	√	√	√
FFFFF105H	Interrupt mask register 2H	IMR2H		√	√		FFH	√	√	√	√
FFFFF106H	Interrupt mask register 3	IMR3					FFFFH	√	√	√	√
FFFFF106H	Interrupt mask register 3L	IMR3L		√	√		FFH	√	√	√	√
FFFFF107H	Interrupt mask register 3H	IMR3H		√	√		FFH	√	√	√	√
FFFFF108H	Interrupt mask register 4	IMR4					FFFFH	–	–	√	√
FFFFF108H	Interrupt mask register 4L	IMR4L		√	√		FFH	–	–	√	√
FFFFF109H	Interrupt mask register 4H	IMR4H		√	√		FFH	–	–	–	√
FFFFF10AH	Interrupt mask register 5	IMR5					FFFFH	–	–	–	√
FFFFF10AH	Interrupt mask register 5L	IMR5L		√	√		FFH	–	–	–	√
FFFFF10BH	Interrupt mask register 5H	IMR5H		√	√		FFH	–	–	–	√
FFFFF110H	Interrupt control register	LVILIC		√	√		47H	√	√	√	√
FFFFF112H	Interrupt control register	LVIHIC		√	√		47H	√	√	√	√
FFFFF114H	Interrupt control register	PIC0		√	√		47H	√	√	√	√
FFFFF116H	Interrupt control register	PIC1		√	√		47H	√	√	√	√
FFFFF118H	Interrupt control register	PIC2		√	√		47H	√	√	√	√
FFFFF11AH	Interrupt control register	PIC3		√	√		47H	√	√	√	√
FFFFF11CH	Interrupt control register	PIC4		√	√		47H	√	√	√	√
FFFFF11EH	Interrupt control register	PIC5		√	√		47H	√	√	√	√
FFFFF120H	Interrupt control register	PIC6		√	√		47H	√	√	√	√
FFFFF122H	Interrupt control register	PIC7		√	√		47H	√	√	√	√
FFFFF124H	Interrupt control register	TAB0OVIC		√	√		47H	√	√	√	√
FFFFF126H	Interrupt control register	TAB0CCIC0		√	√		47H	√	√	√	√

(3/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value	HE3	HF3	HG3	HJ3
				1	8	16					
FFFFF128H	Interrupt control register	TAB0CCIC1	R/W	√	√		47H	√	√	√	√
FFFFF12AH	Interrupt control register	TAB0CCIC2		√	√		47H	√	√	√	√
FFFFF12CH	Interrupt control register	TAB0CCIC3		√	√		47H	√	√	√	√
FFFFF12EH	Interrupt control register	TAA0OVIC		√	√		47H	√	√	√	√
FFFFF130H	Interrupt control register	TAA0CCIC0		√	√		47H	√	√	√	√
FFFFF132H	Interrupt control register	TAA0CCIC1		√	√		47H	√	√	√	√
FFFFF134H	Interrupt control register	TAA1OVIC		√	√		47H	√	√	√	√
FFFFF136H	Interrupt control register	TAA1CCIC0		√	√		47H	√	√	√	√
FFFFF138H	Interrupt control register	TAA1CCIC1		√	√		47H	√	√	√	√
FFFFF13AH	Interrupt control register	TAA2OVIC		√	√		47H	√	√	√	√
FFFFF13CH	Interrupt control register	TAA2CCIC0		√	√		47H	√	√	√	√
FFFFF13EH	Interrupt control register	TAA2CCIC1		√	√		47H	√	√	√	√
FFFFF140H	Interrupt control register	TAA3OVIC		√	√		47H	√	√	√	√
FFFFF142H	Interrupt control register	TAA3CCIC0		√	√		47H	√	√	√	√
FFFFF144H	Interrupt control register	TAA3CCIC1		√	√		47H	√	√	√	√
FFFFF146H	Interrupt control register	TAA4OVIC		√	√		47H	√	√	√	√
FFFFF148H	Interrupt control register	TAA4CCIC0		√	√		47H	√	√	√	√
FFFFF14AH	Interrupt control register	TAA4CCIC1		√	√		47H	√	√	√	√
FFFFF14CH	Interrupt control register	TM0EQIC0		√	√		47H	√	√	√	√
FFFFF14EH	Interrupt control register	CB0RIC		√	√		47H	√	√	√	√
FFFFF150H	Interrupt control register	CB0TIC		√	√		47H	√	√	√	√
FFFFF152H	Interrupt control register	CB1RIC		√	√		47H	√	√	√	√
FFFFF154H	Interrupt control register	CB1TIC		√	√		47H	√	√	√	√
FFFFF156H	Interrupt control register	UD0SIC		√	√		47H	√	√	√	√
FFFFF158H	Interrupt control register	UD0RIC		√	√		47H	√	√	√	√
FFFFF15AH	Interrupt control register	UD0TIC		√	√		47H	√	√	√	√
FFFFF15CH	Interrupt control register	UD1SIC		√	√		47H	√	√	√	√
FFFFF15EH	Interrupt control register	UD1RIC		√	√		47H	√	√	√	√
FFFFF160H	Interrupt control register	UD1TIC		√	√		47H	√	√	√	√
FFFFF162H	Interrupt control register	IIC0IC/ UD4SIC[Note]		√	√		47H	√	√	√	√
FFFFF164H	Interrupt control register	ADIC		√	√		47H	√	√	√	√
FFFFF16EH	Interrupt control register	DMAIC0		√	√		47H	√	√	√	√
FFFFF170H	Interrupt control register	DMAIC1		√	√		47H	√	√	√	√
FFFFF172H	Interrupt control register	DMAIC2		√	√		47H	√	√	√	√
FFFFF174H	Interrupt control register	DMAIC3		√	√		47H	√	√	√	√
FFFFF176H	Interrupt control register	KRIC		√	√		47H	√	√	√	√
FFFFF178H	Interrupt control register	WTIIC		√	√		47H	√	√	√	√
FFFFF17AH	Interrupt control register	WTIC		√	√		47H	√	√	√	√
FFFFF180H	Interrupt control register	PIC8		√	√		47H	–	–	√	√
FFFFF182H	Interrupt control register	PIC9		√	√		47H	–	–	√	√

Note Supported only in the μPD70F3757

(4/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value	HE3	HF3	HG3	HJ3
				1	8	16					
FFFFF184H	Interrupt control register	PIC10	R/W	√	√		47H	–	–	√	√
FFFFF186H	Interrupt control register	TAB1OVIC		√	√		47H	–	–	√	√
FFFFF188H	Interrupt control register	TAB1CCIC0		√	√		47H	–	–	√	√
FFFFF18AH	Interrupt control register	TAB1CCIC1		√	√		47H	–	–	√	√
FFFFF18CH	Interrupt control register	TAB1CCIC2		√	√		47H	–	–	√	√
FFFFF18EH	Interrupt control register	TAB1CCIC3		√	√		47H	–	–	√	√
FFFFF190H	Interrupt control register	UD2SIC		√	√		47H	–	–	√	√
FFFFF192H	Interrupt control register	UD2RIC		√	√		47H	–	–	√	√
FFFFF194H	Interrupt control register	UD2TIC		√	√		47H	–	–	√	√
FFFFF19EH	Interrupt control register	PIC11		√	√		47H	–	–	–	√
FFFFF1A0H	Interrupt control register	PIC12		√	√		47H	–	–	–	√
FFFFF1A2H	Interrupt control register	PIC13		√	√		47H	–	–	–	√
FFFFF1A4H	Interrupt control register	PIC14		√	√		47H	–	–	–	√
FFFFF1A6H	Interrupt control register	UD3SIC		√	√		47H	–	–	–	**Note**
FFFFF1A8H	Interrupt control register	UD3RIC		√	√		47H	–	–	–	**Note**
FFFFF1AAH	Interrupt control register	UD3TIC		√	√		47H	–	–	–	**Note**
FFFFF1ACH	Interrupt control register	UD4RIC		√	√		47H	–	–	–	**Note**
FFFFF1AEH	Interrupt control register	UD4TIC		√	√		47H	–	–	–	**Note**
FFFFF1B0H	Interrupt control register	TAB2OVIC		√	√		47H	–	–	–	√
FFFFF1B2H	Interrupt control register	TAB2CCIC0		√	√		47H	–	–	–	√
FFFFF1B4H	Interrupt control register	TAB2CCIC1		√	√		47H	–	–	–	√
FFFFF1B6H	Interrupt control register	TAB2CCIC2		√	√		47H	–	–	–	√
FFFFF1B8H	Interrupt control register	TAB2CCIC3		√	√		47H	–	–	–	√
FFFFF1BAH	Interrupt control register	UD5SIC		√	√		47H	–	–	–	**Note**
FFFFF1BCH	Interrupt control register	CB2RIC/ UD5RIC**Note**		√	√		47H	–	–	–	√
FFFFF1BDH	Interrupt control register	CB2TIC/ UD5TIC**Note**		√	√		47H	–	–	–	√
FFFFF1FAH	In-service priority register	ISPR	R	√	√		00H	√	√	√	√
FFFFF1FCH	Command register	PRCMD	W		√		Undefined	√	√	√	√
FFFFF1FEH	Power save control register	PSC	R/W	√	√		00H	√	√	√	√
FFFFF200H	A/D converter mode register 0	ADA0M0		√	√		00H	√	√	√	√
FFFFF201H	A/D converter mode register 1	ADA0M1		√	√		00H	√	√	√	√
FFFFF202H	A/D converter channel specification register 0	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			√	0000H	√	√	√	√
FFFFF211H	A/D conversion result register 0H	ADA0CR0H			√		00H	√	√	√	√
FFFFF212H	A/D conversion result register 1	ADA0CR1				√	0000H	√	√	√	√
FFFFF213H	A/D conversion result register 1H	ADA0CR1H			√		00H	√	√	√	√

Note Supported only in the μPD70F3757

(5/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value	HE3	HF3	HG3	HJ3
				1	8	16					
FFFFF214H	A/D conversion result register 2	ADA0CR2	R			√	0000H	√	√	√	√
FFFFF215H	A/D conversion result register 2H	ADA0CR2H			√		00H	√	√	√	√
FFFFF216H	A/D conversion result register 3	ADA0CR3				√	0000H	√	√	√	√
FFFFF217H	A/D conversion result register 3H	ADA0CR3H			√		00H	√	√	√	√
FFFFF218H	A/D conversion result register 4	ADA0CR4				√	0000H	√	√	√	√
FFFFF219H	A/D conversion result register 4H	ADA0CR4H			√		00H	√	√	√	√
FFFFF21AH	A/D conversion result register 5	ADA0CR5				√	0000H	√	√	√	√
FFFFF21BH	A/D conversion result register 5H	ADA0CR5H			√		00H	√	√	√	√
FFFFF21CH	A/D conversion result register 6	ADA0CR6				√	0000H	√	√	√	√
FFFFF21DH	A/D conversion result register 6H	ADA0CR6H			√		00H	√	√	√	√
FFFFF21EH	A/D conversion result register 7	ADA0CR7				√	0000H	√	√	√	√
FFFFF21FH	A/D conversion result register 7H	ADA0CR7H			√		00H	√	√	√	√
FFFFF220H	A/D conversion result register 8	ADA0CR8				√	0000H	√	√	√	√
FFFFF221H	A/D conversion result register 8H	ADA0CR8H			√		00H	√	√	√	√
FFFFF222H	A/D conversion result register 9	ADA0CR9				√	0000H	√	√	√	√
FFFFF223H	A/D conversion result register 9H	ADA0CR9H			√		00H	√	√	√	√
FFFFF224H	A/D conversion result register 10	ADA0CR10				√	0000H	–	√	√	√
FFFFF225H	A/D conversion result register 10H	ADA0CR10H			√		00H	–	√	√	√
FFFFF226H	A/D conversion result register 11	ADA0CR11				√	0000H	–	√	√	√
FFFFF227H	A/D conversion result register 11H	ADA0CR11H			√		00H	–	√	√	√
FFFFF228H	A/D conversion result register 12	ADA0CR12				√	0000H	–	–	√	√
FFFFF229H	A/D conversion result register 12H	ADA0CR12H			√		00H	–	–	√	√
FFFFF22AH	A/D conversion result register 13	ADA0CR13				√	0000H	–	–	√	√
FFFFF22BH	A/D conversion result register 13H	ADA0CR13H			√		00H	–	–	√	√
FFFFF22CH	A/D conversion result register 14	ADA0CR14				√	0000H	–	–	√	√
FFFFF22DH	A/D conversion result register 14H	ADA0CR14H			√		00H	–	–	√	√
FFFFF22EH	A/D conversion result register 15	ADA0CR15				√	0000H	–	–	√	√
FFFFF22FH	A/D conversion result register 15H	ADA0CR15H			√		00H	–	–	√	√
FFFFF230H	A/D conversion result register 16	ADA0CR16				√	0000H	–	–	–	√
FFFFF231H	A/D conversion result register 16H	ADA0CR16H			√		00H	–	–	–	√
FFFFF232H	A/D conversion result register 17	ADA0CR17				√	0000H	–	–	–	√
FFFFF233H	A/D conversion result register 17H	ADA0CR17H			√		00H	–	–	–	√
FFFFF234H	A/D conversion result register 18	ADA0CR18				√	0000H	–	–	–	√
FFFFF235H	A/D conversion result register 18H	ADA0CR18H			√		00H	–	–	–	√
FFFFF236H	A/D conversion result register 19	ADA0CR19				√	0000H	–	–	–	√
FFFFF237H	A/D conversion result register 19H	ADA0CR19H			√		00H	–	–	–	√
FFFFF238H	A/D conversion result register 20	ADA0CR20				√	0000H	–	–	–	√
FFFFF239H	A/D conversion result register 20H	ADA0CR20H			√		00H	–	–	–	√
FFFFF23AH	A/D conversion result register 21	ADA0CR21				√	0000H	–	–	–	√
FFFFF23BH	A/D conversion result register 21H	ADA0CR21H			√		00H	–	–	–	√

`(6/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value	HE3	HF3	HG3	HJ3
				1	8	16					
FFFFF23CH	A/D conversion result register 22	ADA0CR22	R			√	0000H	–	–	–	√
FFFFF23DH	A/D conversion result register 22H	ADA0CR22H			√		00H	–	–	–	√
FFFFF23EH	A/D conversion result register 23	ADA0CR23				√	0000H	–	–	–	√
FFFFF23FH	A/D conversion result register 23H	ADA0CR23H			√		00H	–	–	–	√
FFFFF300H	Key return mode register	KRM	R/W	√	√		00H	√	√	√	√
FFFFF308H	Selector operation control register 0	SELCNT0		√	√		00H	√	√	√	√
FFFFF30AH	Selector operation control register 1	SELCNT1		√	√		00H	–	–	–	**Note**
FFFFF30EH	Selector operation control register 3	SELCNT3		√	√		00H	–	–	–	**Note**
FFFFF318H	Noise elimination control register	NFC		√	√		00H	√	√	√	√
FFFFF340H	OPS0 clock selection register	OCKS0			√		00H	√	√	√	√
FFFFF3F0H	SSCG control register	SSCGCTL		√	√		00H	√	√	√	√
FFFFF3F1H	SSCG frequency control register 0	SFC0			√		00H	√	√	√	√
FFFFF3F2H	SSCG frequency control register 1	SFC1			√		00H	√	√	√	√
FFFFF3F8H	Selector operation control register 4	SELCNT4		√	√		00H	√	√	√	√
FFFFF400H	Port 0	P0		√	√		Undefined	√	√	√	√
FFFFF402H	Port 1	P1		√	√		Undefined	–	–	√	√
FFFFF406H	Port 3	P3				√	Undefined	–	√	√	√
FFFFF406H	Port 3L	P3L		√	√		Undefined	√	√	√	√
FFFFF407H	Port 3H	P3H		√	√		Undefined	–	√	√	√
FFFFF408H	Port 4	P4		√	√		Undefined	√	√	√	√
FFFFF40AH	Port 5	P5		√	√		Undefined	√	√	√	√
FFFFF40CH	Port 6	P6				√	Undefined	–	–	–	√
FFFFF40CH	Port 6L	P6L		√	√		Undefined	–	–	–	√
FFFFF40DH	Port 6H	P6H		√	√		Undefined	–	–	–	√
FFFFF40EH	Port 7L	P7L		√	√		Undefined	√	√	√	√
FFFFF40FH	Port 7H	P7H		√	√		Undefined	√	√	√	√
FFFFF410H	Port 8	P8		√	√		Undefined	–	–	–	√
FFFFF412H	Port 9	P9				√	Undefined	√	√	√	√
FFFFF412H	Port 9L	P9L		√	√		Undefined	√	√	√	√
FFFFF413H	Port 9H	P9H		√	√		Undefined	√	√	√	√
FFFFF418H	Port 12	P12		√	√		Undefined	–	–	–	√
FFFFF420H	Port mode register 0	PM0		√	√		FFH	√	√	√	√
FFFFF422H	Port mode register 1	PM1		√	√		FFH	–	–	√	√
FFFFF426H	Port mode register 3	PM3				√	FFFFH	–	√	√	√
FFFFF426H	Port mode register 3L	PM3L		√	√		FFH	√	√	√	√
FFFFF427H	Port mode register 3H	PM3H		√	√		FFH	–	√	√	√
FFFFF428H	Port mode register 4	PM4		√	√		FFH	√	√	√	√
FFFFF42AH	Port mode register 5	PM5		√	√		FFH	√	√	√	√
FFFFF42CH	Port mode register 6	PM6				√	FFFFH	–	–	–	√
FFFFF42CH	Port mode register 6L	PM6L		√	√		FFH	–	–	–	√
FFFFF42DH	Port mode register 6H	PM6H		√	√		FFH	–	–	–	√

Note Supported only in the μPD70F3757

(7/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value	HE3	HF3	HG3	HJ3
				1	8	16					
FFFFF42EH	Port mode register 7L	PM7L	R/W	√	√		FFH	√	√	√	√
FFFFF42FH	Port mode register 7H	PM7H		√	√		FFH	√	√	√	√
FFFFF430H	Port mode register 8	PM8		√	√		FFH	–	–	–	√
FFFFF432H	Port mode register 9	PM9				√	FFFFH	√	√	√	√
FFFFF432H	Port mode register 9L	PM9L		√	√		FFH	√	√	√	√
FFFFF433H	Port mode register 9H	PM9H		√	√		FFH	√	√	√	√
FFFFF438H	Port mode register 12	PM12		√	√		FFH	–	–	–	√
FFFFF440H	Port mode control register 0	PMC0		√	√		00H	√	√	√	√
FFFFF442H	Port mode control register 1	PMC1		√	√		00H	–	–	√	√
FFFFF446H	Port mode control register 3	PMC3				√	0000H	–	–	√	√
FFFFF446H	Port mode control register 3L	PMC3L		√	√		00H	√	√	√	√
FFFFF447H	Port mode control register 3H	PMC3H		√	√		00H	–	–	√	√
FFFFF448H	Port mode control register 4	PMC4		√	√		00H	√	√	√	√
FFFFF44AH	Port mode control register 5	PMC5		√	√		00H	√	√	√	√
FFFFF44CH	Port mode control register 6	PMC6				√	0000H	–	–	–	√
FFFFF44CH	Port mode control register 6L	PMC6L		√	√		00H	–	–	–	√
FFFFF44DH	Port mode control register 6H	PMC6H		√	√		00H	–	–	–	√
FFFFF44EH	Port mode control register 7L	PMC7L		√	√		00H	√	√	√	√
FFFFF44FH	Port mode control register 7H	PMC7H		√	√		00H	√	√	√	√
FFFFF450H	Port mode control register 8	PMC8		√	√		00H	–	–	–	√
FFFFF452H	Port mode control register 9	PMC9				√	0000H	√	√	√	√
FFFFF452H	Port mode control register 9L	PMC9L		√	√		00H	√	√	√	√
FFFFF453H	Port mode control register 9H	PMC9H		√	√		00H	√	√	√	√
FFFFF458H	Port mode control register 12	PMC12		√	√		00H	–	–	–	√
FFFFF460H	Port function control register 0	PFC0		√	√		00H	√	√	√	√
FFFFF466H	Port function control register 3L	PFC3L		√	√		00H	√	√	√	√
FFFFF468H	Port function control register 4	PFC4		√	√		00H	√	√	√	√
FFFFF46AH	Port function control register 5	PFC5		√	√		00H	√	√	√	√
FFFFF46CH	Port function control register 6	PFC6				√	0000H	–	–	–	√
FFFFF46CH	Port function control register 6L	PFC6L		√	√		00H	–	–	–	√
FFFFF46DH	Port function control register 6H	PFC6H		√	√		00H	–	–	–	√
FFFFF472H	Port function control register 9	PFC9				√	0000H	√	√	√	√
FFFFF472H	Port function control register 9L	PFC9L		√	√		00H	√	√	√	√
FFFFF473H	Port function control register 9H	PFC9H		√	√		00H	√	√	√	√
FFFFF484H	Data wait control register 0	DWC0				√	7777H	–	–	–	√
FFFFF488H	Address wait control register	AWC				√	FFFFH	–	–	–	√
FFFFF48AH	Bus cycle control register	BCC				√	AAAAH	–	–	–	√
FFFFF540H	TAB0 control register 0	TAB0CTL0		√	√		00H	√	√	√	√

(8/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value	HE3	HF3	HG3	HJ3
				1	8	16					
FFFFF541H	TAB0 control register 1	TAB0CTL1	R/W	√	√		00H	√	√	√	√
FFFFF542H	TAB0 I/O control register 0	TAB0IOC0		√	√		00H	√	√	√	√
FFFFF543H	TAB0 I/O control register 1	TAB0IOC1		√	√		00H	√	√	√	√
FFFFF544H	TAB0 I/O control register 2	TAB0IOC2		√	√		00H	√	√	√	√
FFFFF545H	TAB0 option register 0	TAB0OPT0		√	√		00H	√	√	√	√
FFFFF546H	TAB0 capture/compare register 0	TAB0CCR0				√	0000H	√	√	√	√
FFFFF548H	TAB0 capture/compare register 1	TAB0CCR1				√	0000H	√	√	√	√
FFFFF54AH	TAB0 capture/compare register 2	TAB0CCR2				√	0000H	√	√	√	√
FFFFF54CH	TAB0 capture/compare register 3	TAB0CCR3				√	0000H	√	√	√	√
FFFFF54EH	TAB0 counter read buffer register	TAB0CNT	R			√	0000H	√	√	√	√
FFFFF556H	TAB0 option register 1	TAB0OPT1	R/W	√	√		00H	√	√	√	√
FFFFF561H	TAB0 option register 2	TAB0OPT2		√	√		00H	√	√	√	√
FFFFF562H	TAB0 output I/O control register 3	TAB0IOC3		√	√		00H	√	√	√	√
FFFFF564H	TAB0 deadtime compare register	TAB0DTC		√	√		00H	√	√	√	√
FFFFF570H	High-impedance output control register 0	HZA0CTL0		√	√		00H	√	√	√	√
FFFFF571H	High-impedance output control register 0	HAZ0CTL1		√	√		00H	√	√	√	√
FFFFF590H	TAA0 control register 0	TAA0CTL0		√	√		00H	√	√	√	√
FFFFF591H	TAA0 control register 1	TAA0CTL1		√	√		00H	√	√	√	√
FFFFF592H	TAA0 I/O control register 0	TAA0IOC0		√	√		00H	√	√	√	√
FFFFF593H	TAA0 I/O control register 1	TAA0IOC1		√	√		00H	√	√	√	√
FFFFF594H	TAA0 I/O control register 2	TAA0IOC2		√	√		00H	√	√	√	√
FFFFF595H	TAA0 option register 0	TAA0OPT0		√	√		00H	√	√	√	√
FFFFF596H	TAA0 capture/compare register 0	TAA0CCR0				√	0000H	√	√	√	√
FFFFF598H	TAA0 capture/compare register 1	TAA0CCR1				√	0000H	√	√	√	√
FFFFF59AH	TAA0 counter read buffer register	TAA0CNT	R			√	0000H	√	√	√	√
FFFFF5A0H	TAA1 control register 0	TAA1CTL0	R/W	√	√		00H	√	√	√	√
FFFFF5A1H	TAA1 control register 1	TAA1CTL1		√	√		00H	√	√	√	√
FFFFF5A2H	TAA1 I/O control register 0	TAA1IOC0		√	√		00H	√	√	√	√
FFFFF5A3H	TAA1 I/O control register 1	TAA1IOC1		√	√		00H	√	√	√	√
FFFFF5A4H	TAA1 I/O control register 2	TAA1IOC2		√	√		00H	√	√	√	√
FFFFF5A5H	TAA1 option register 0	TAA1OPT0		√	√		00H	√	√	√	√
FFFFF5A6H	TAA1 capture/compare register 0	TAA1CCR0				√	0000H	√	√	√	√
FFFFF5A8H	TAA1 capture/compare register 1	TAA1CCR1				√	0000H	√	√	√	√
FFFFF5AAH	TAA1 counter read buffer register	TAA1CNT	R			√	0000H	√	√	√	√
FFFFF5AdH	TAA1 option register 1	TAA1OPT1	R/W	√	√		00H	√	√	√	√
FFFFF5B0H	TAA2 control register 0	TAA2CTL0		√	√		00H	√	√	√	√
FFFFF5B1H	TAA2 control register 1	TAA2CTL1		√	√		00H	√	√	√	√
FFFFF5B2H	TAA2 I/O control register 0	TAA2IOC0		√	√		00H	√	√	√	√
FFFFF5B3H	TAA2 I/O control register 1	TAA2IOC1		√	√		00H	√	√	√	√
FFFFF5B4H	TAA2 I/O control register 2	TAA2IOC2		√	√		00H	√	√	√	√
FFFFF5B5H	TAA2 option register	TAA2OPT0		√	√		00H	√	√	√	√

(9/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value	HE3	HF3	HG3	HJ3
				1	8	16					
FFFFF5B6H	TAA2 capture/compare register 0	TAA2CCR0	R/W			√	0000H	√	√	√	√
FFFFF5B8H	TAA2 capture/compare register 1	TAA2CCR1				√	0000H	√	√	√	√
FFFFF5BAH	TAA2 counter read buffer register	TAA2CNT	R			√	0000H	√	√	√	√
FFFFF5C0H	TAA3 control register 0	TAA3CTL0	R/W	√	√		00H	√	√	√	√
FFFFF5C1H	TAA3 control register 1	TAA3CTL1		√	√		00H	√	√	√	√
FFFFF5C2H	TAA3 I/O control register 0	TAA3IOC0		√	√		00H	√	√	√	√
FFFFF5C3H	TAA3 I/O control register 1	TAA3IOC1		√	√		00H	√	√	√	√
FFFFF5C4H	TAA3 I/O control register 2	TAA3IOC2		√	√		00H	√	√	√	√
FFFFF5C5H	TAA3 option register 0	TAA3OPT0		√	√		00H	√	√	√	√
FFFFF5C6H	TAA3 capture/compare register 0	TAA3CCR0				√	0000H	√	√	√	√
FFFFF5C8H	TAA3 capture/compare register 1	TAA3CCR1				√	0000H	√	√	√	√
FFFFF5CAH	TAA3 counter read buffer register	TAA3CNT	R			√	0000H	√	√	√	√
FFFFF5CDH	TAA3 option register 1	TAA3OPT1	R/W	√	√		00H	√	√	√	√
FFFFF5D0H	TAA4 control register 0	TAA4CTL0		√	√		00H	√	√	√	√
FFFFF5D1H	TAA4 control register 1	TAA4CTL1		√	√		00H	√	√	√	√
FFFFF5D2H	TAA4 I/O control register 0	TAA4IOC0		√	√		00H	√	√	√	√
FFFFF5D3H	TAA4 I/O control register 1	TAA4IOC1		√	√		00H	√	√	√	√
FFFFF5D4H	TAA4 I/O control register 2	TAA4IOC2		√	√		00H	√	√	√	√
FFFFF5D5H	TAA4 option register 0	TAA4OPT0		√	√		00H	√	√	√	√
FFFFF5D6H	TAA4 capture/compare register 0	TAA4CCR0				√	0000H	√	√	√	√
FFFFF5D8H	TAA4 capture/compare register 1	TAA4CCR1				√	0000H	√	√	√	√
FFFFF5DAH	TAA4 counter read buffer register	TAA4CNT	R			√	0000H	√	√	√	√
FFFFF610H	TAB1 timer control register 0	TAB1CTL0	R/W	√	√		00H	–	–	√	√
FFFFF611H	TAB1 timer control register 1	TAB1CTL1		√	√		00H	–	–	√	√
FFFFF612H	TAB1 timer dedicated I/O control register 0	TAB1IOC0		√	√		00H	–	–	√	√
FFFFF613H	TAB1 timer dedicated I/O control register 1	TAB1IOC1		√	√		00H	–	–	√	√
FFFFF614H	TAB1 timer dedicated I/O control register 2	TAB1IOC2		√	√		00H	–	–	√	√
FFFFF615H	TAB1 timer option register 0	TAB1OPT0		√	√		00H	–	–	√	√
FFFFF616H	TAB1 capture/compare register 0	TAB1CCR0				√	0000H	–	–	√	√
FFFFF618H	TAB1 capture/compare register 1	TAB1CCR1				√	0000H	–	–	√	√
FFFFF61AH	TAB1 capture/compare register 2	TAB1CCR2				√	0000H	–	–	√	√
FFFFF61CH	TAB1 capture/compare register 3	TAB1CCR3				√	0000H	–	–	√	√
FFFFF61EH	TAB1 counter read buffer register	TAB1CNT	R			√	0000H	–	–	√	√
FFFFF620H	TAB2 timer control register 0	TAB2CTL0	R/W	√	√		00H	–	–	–	√
FFFFF621H	TAB2 timer control register 1	TAB2CTL1		√	√		00H	–	–	–	√
FFFFF622H	TAB2 timer dedicated I/O control register 0	TAB2IOC0		√	√		00H	–	–	–	√
FFFFF623H	TAB2 timer dedicated I/O control register 1	TAB2IOC1		√	√		00H	–	–	–	√
FFFFF624H	TAB2 timer dedicated I/O control register 2	TAB2IOC2		√	√		00H	–	–	–	√
FFFFF625H	TAB2 timer option register 0	TAB2OPT0		√	√		00H	–	–	–	√
FFFFF626H	TAB2 capture/compare register 0	TAB2CCR0				√	0000H	–	–	–	√
FFFFF628H	TAB2 capture/compare register 1	TAB2CCR1				√	0000H	–	–	–	√
FFFFF62AH	TAB2 capture/compare register 2	TAB2CCR2				√	0000H	–	–	–	√

(10/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value	HE3	HF3	HG3	HJ3
				1	8	16					
FFFFF62CH	TAB2 capture/compare register 3	TAB2CCR3	R/W			√	0000H	–	–	–	√
FFFFF62EH	TAB2 counter read buffer register	TAB2CNT	R			√	0000H	–	–	–	√
FFFFF680H	Watch timer operation mode register	WTM	R/W	√	√		00H	√	√	√	√
FFFFF690H	TMM0 timer control register 0	TM0CTL0		√	√		00H	√	√	√	√
FFFFF694H	TMM0 compare register 0	TM0CMP0				√	0000H	√	√	√	√
FFFFF6C0H	Oscillation stabilization time selection register	OSTS			√		06H	√	√	√	√
FFFFF6C1H	PLL lockup timer specification register	PLLS			√		03H	√	√	√	√
FFFFF6C2H	Oscillation stabilization time count status register	OSTC	R	√	√		00H	√	√	√	√
FFFFF6D0H	Watchdog timer mode register 2	WDTM2	R/W	√	√		67H	√	√	√	√
FFFFF6D1H	Watchdog timer enable register	WDTE			√		9AH	√	√	√	√
FFFFF700H	Port 0 function control expansion register	PFCE0		√	√		00H	√	√	√	√
FFFFF706H	Port 3 function control expansion register L	PFCE3L		√	√		00H	√	√	√	√
FFFFF708H	Port 4 function control expansion register	PFCE4		√	√		00H	–	–	–	**Note**
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		√	√		80H	√	√	√	√
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	√	√	√	√
FFFFF824H	Lock register	LOCKR	R	√	√		01H	√	√	√	√
FFFFF828H	Processor clock control register	PCC	R/W	√	√		40H	√	√	√	√
FFFFF82CH	PLL control register	PLLCTL		√	√		00H	√	√	√	√
FFFFF82EH	CPU operation clock status register	CCLS	R	√	√		00H	√	√	√	√
FFFFF82FH	Programmable clock mode register	PCLM	R/W	√	√		00H	√	√	√	√
FFFFF860H	Main system clock mode register	MCM		√	√		00H	√	√	√	√
FFFFF870H	Clock monitor mode register	CLM		√	√		00H	√	√	√	√
FFFFF888H	Reset source flag register	RESF		√	√		00H	√	√	√	√
FFFFF890H	Low-voltage detection register	LVIM		√	√		00H	√	√	√	√
FFFFF891H	Low-voltage detection level selection register	LVIS			√		00H	√	√	√	√
FFFFF892H	Internal RAM data status register	RAMS		√	√		01H	√	√	√	√
FFFFF8B0H	Prescaler mode register	PRSM0			√		00H	√	√	√	√
FFFFF8B1H	Prescaler compare register	PRSCM0			√		00H	√	√	√	√
FFFFF9FCH	On-chip debug alternate pin setting register	OCDM		√	√		01H	√	√	√	√
FFFFF9FEH	Peripheral emulation register 1	PEMU1		√	√		00H	√	√	√	√
FFFFFA00H	UARTD0 control register 0	UD0CTL0		√	√		10H	√	√	√	√
FFFFFA01H	UARTD0 control register 1	UD0CTL1			√		00H	√	√	√	√
FFFFFA02H	UARTD0 control register 2	UD0CTL2			√		FFH	√	√	√	√

Note Supported only in the μPD70F3757

(11/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value	HE3	HF3	HG3	HJ3
				1	8	16					
FFFFFA03H	UARTD0 option control register 0	UD0OPT0	R/W	√	√		14H	√	√	√	√
FFFFFA04H	UARTD0 status register	UD0STR		√	√		00H	√	√	√	√
FFFFFA05H	UARTD0 option control register 1	UD0OPT1			√		00H	√	√	√	√
FFFFFA06H	UARTD0 receive data register	UD0RX	R		√		FFH	√	√	√	√
FFFFFA07H	UARTD0 transmit data register	UD0TX	R/W		√		FFH	√	√	√	√
FFFFFA10H	UARTD1 control register 0	UD1CTL0		√	√		10H	√	√	√	√
FFFFFA11H	UARTD1 control register 1	UD1CTL1			√		00H	√	√	√	√
FFFFFA12H	UARTD1 control register 2	UD1CTL2			√		FFH	√	√	√	√
FFFFFA13H	UARTD1 option control register 0	UD1OPT0		√	√		14H	√	√	√	√
FFFFFA14H	UARTD1 status register	UD1STR		√	√		00H	√	√	√	√
FFFFFA15H	UARTD1 option control register 1	UD1OPT1			√		00H	√	√	√	√
FFFFFA16H	UARTD1 receive data register	UD1RX	R		√		FFH	√	√	√	√
FFFFFA17H	UARTD1 transmit data register	UD1TX	R/W		√		FFH	√	√	√	√
FFFFFA20H	UARTD2 control register 0	UD2CTL0		√	√		10H	–	–	√	√
FFFFFA21H	UARTD2 control register 1	UD2CTL1			√		00H	–	–	√	√
FFFFFA22H	UARTD2 control register 2	UD2CTL2			√		FFH	–	–	√	√
FFFFFA23H	UARTD2 option control register 0	UD2OPT0		√	√		14H	–	–	√	√
FFFFFA24H	UARTD2 status register	UD2STR		√	√		00H	–	–	√	√
FFFFFA25H	UARTD2 option control register 1	UD2OPT1			√		00H	–	–	√	√
FFFFFA26H	UARTD2 receive data register	UD2RX	R		√		FFH	–	–	√	√
FFFFFA27H	UARTD2 transmit data register	UD2TX	R/W		√		FFH	–	–	√	√
FFFFFA30H	UARTD3 control register 0	UD3CTL0		√	√		10H	–	–	–	**Note**
FFFFFA31H	UARTD3 control register 1	UD3CTL1			√		00H	–	–	–	**Note**
FFFFFA32H	UARTD3 control register 2	UD3CTL2			√		FFH	–	–	–	**Note**
FFFFFA33H	UARTD3 option control register 0	UD3OPT0		√	√		14H	–	–	–	**Note**
FFFFFA34H	UARTD3 status register	UD3STR		√	√		00H	–	–	–	**Note**
FFFFFA35H	UARTD3 option control register 1	UD3OPT1			√		00H	–	–	–	**Note**
FFFFFA36H	UARTD3 receive data register	UD3RX	R		√		FFH	–	–	–	**Note**
FFFFFA37H	UARTD3 transmit data register	UD3TX	R/W		√		FFH	–	–	–	**Note**
FFFFFA40H	UARTD4 control register 0	UD4CTL0		√	√		10H	–	–	–	**Note**
FFFFFA41H	UARTD4 control register 1	UD4CTL1			√		00H	–	–	–	**Note**
FFFFFA42H	UARTD4 control register 2	UD4CTL2			√		FFH	–	–	–	**Note**
FFFFFA43H	UARTD4 option control register 0	UD4OPT0		√	√		14H	–	–	–	**Note**
FFFFFA44H	UARTD4 status register	UD4STR		√	√		00H	–	–	–	**Note**
FFFFFA45H	UARTD4 option control register 1	UD4OPT1			√		00H	–	–	–	**Note**
FFFFFA46H	UARTD4 receive data register	UD4RX	R		√		FFH	–	–	–	**Note**
FFFFFA47H	UARTD4 transmit data register	UD4TX	R/W		√		FFH	–	–	–	**Note**
FFFFFA50H	UARTD5 control register 0	UD5CTL0			√		10H	–	–	–	**Note**
FFFFFA51H	UARTD5 control register 1	UD5CTL1		√	√		00H	–	–	–	**Note**
FFFFFA52H	UARTD5 control register 2	UD5CTL2			√		FFH	–	–	–	**Note**
FFFFFA53H	UARTD5 option control register 0	UD5OPT0			√		14H	–	–	–	**Note**
FFFFFA54H	UARTD5 status register	UD5STR		√	√		00H	–	–	–	**Note**

Note Supported only in the μPD70F3757

(12/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value	HE3	HF3	HG3	HJ3
				1	8	16					
FFFFFA55H	UARTD5 option control register 1	UD5OPT1	R/W		√		00H	–	–	–	**Note**
FFFFFA56H	UARTD5 receive data register	UD5RX	R		√		FFH	–	–	–	**Note**
FFFFFA57H	UARTD5 transmit data register	UD5TX	R/W		√		FFH	–	–	–	**Note**
FFFFFC00H	External interrupt falling edge specification register 0	INTF0		√	√		00H	√	√	√	√
FFFFFC02H	External interrupt falling edge specification register 1	INTF1		√	√		00H	–	–	√	√
FFFFFC06H	External interrupt falling edge specification register 3	INTF3				√	0000H	–	–	√	√
FFFFFC06H	External interrupt falling edge specification register 3L	INTF3L		√	√		00H	√	√	√	√
FFFFFC07H	External interrupt falling edge specification register 3H	INTF3H		√	√		00H	–	–	√	√
FFFFFC08H	External interrupt falling edge specification register 4	INTF4		√	√		00H	–	–	–	**Note**
FFFFFC0CH	External interrupt falling edge specification register 6L	INTF6L		√	√		00H	–	–	–	√
FFFFFC10H	External interrupt falling edge specification register 8	INTF8		√	√		00H	–	–	–	√
FFFFFC13H	External interrupt falling edge specification register 9H	INTF9H		√	√		00H	√	√	√	√
FFFFFC20H	External interrupt rising edge specification register 0	INTR0		√	√		00H	√	√	√	√
FFFFFC22H	External interrupt rising edge specification register 1	INTR1		√	√		00H	–	–	√	√
FFFFFC26H	External interrupt rising edge specification register 3	INTR3				√	0000H	–	–	√	√
FFFFFC26H	External interrupt rising edge specification register 3L	INTR3L		√	√		00H	√	√	√	√
FFFFFC27H	External interrupt rising edge specification register 3H	INTR3H		√	√		00H	–	–	√	√
FFFFFC28H	External interrupt rising edge specification register 4	INTR4		√	√		00H	–	–	–	**Note**
FFFFFC2CH	External interrupt rising edge specification register 6L	INTR6L		√	√		00H	–	–	–	√
FFFFFC30H	External interrupt rising edge specification register 8	INTR8		√	√		00H	–	–	–	√
FFFFFC33H	External interrupt rising edge specification register 9H	INTR9H		√	√		00H	√	√	√	√
FFFFFC40H	Pull-up resistor option register 0	PU0		√	√		00H	√	√	√	√
FFFFFC42H	Pull-up resistor option register 1	PU1		√	√		00H	–		√	√
FFFFFC46H	Pull-up resistor option register 3	PU3				√	0000H	–	√	√	√
FFFFFC46H	Pull-up resistor option register 3L	PU3L		√	√		00H	√	√	√	√
FFFFFC47H	Pull-up resistor option register 3H	PU3H		√	√		00H	–	√	√	√
FFFFFC48H	Pull-up resistor option register 4	PU4		√	√		00H	√	√	√	√
FFFFFC4AH	Pull-up resistor option register 5	PU5		√	√		00H	√	√	√	√
FFFFFC4CH	Pull-up resistor option register 6	PU6				√	0000H	–	–	–	√
FFFFFC4CH	Pull-up resistor option register 6L	PU6L		√	√		00H	–	–	–	√
FFFFFC4DH	Pull-up resistor option register 6H	PU6H		√	√		00H	–	–	–	√
FFFFFC50H	Pull-up resistor option register 8	PU8		√	√		00H	–	–	–	√
FFFFFC52H	Pull-up resistor option register 9	PU9				√	0000H	√	√	√	√
FFFFFC52H	Pull-up resistor option register 9L	PU9L		√	√		00H	√	√	√	√
FFFFFC53H	Pull-up resistor option register 9H	PU9H		√	√		00H	√	√	√	√
FFFFFC73H	Port 9 function control 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	√	√	√	√

Note Supported only in the μPD70F3757

(13/13)

Address	Function Register Name	Symbol	R/W	Manipulatable Bits			Default Value	HE3	HF3	HG3	HJ3
				1	8	16					
FFFFFD06H	CSIB0 transmit data register	CB0TX	R/W			√	0000H	√	√	√	√
FFFFFD06H	CSIB0 transmit data register L	CB0TXL			√		00H	√	√	√	√
FFFFFD10H	CSIB1 control register 0	CB1CTL0		√	√		01H	√	√	√	√
FFFFFD11H	CSIB1 control register 1	CB1CTL1		√	√		00H	√	√	√	√
FFFFFD12H	CSIB1 control register 2	CB1CTL2			√		00H	√	√	√	√
FFFFFD13H	CSIB1 status register	CB1STR		√	√		00H	√	√	√	√
FFFFFD14H	CSIB1 receive data register	CB1RX	R			√	0000H	√	√	√	√
FFFFFD14H	CSIB1 receive data register L	CB1RXL			√		00H	√	√	√	√
FFFFFD16H	CSIB1 transmit data register	CB1TX	R/W			√	0000H	√	√	√	√
FFFFFD16H	CSIB1 transmit data register L	CB1TXL			√		00H	√	√	√	√
FFFFFD20H	CSIB2 control register 0	CB2CTL0		√	√		01H	–	–	–	√
FFFFFD21H	CSIB2 control register 1	CB2CTL1		√	√		00H	–	–	–	√
FFFFFD22H	CSIB2 control register 2	CB2CTL2			√		00H	–	–	–	√
FFFFFD23H	CSIB2 status register	CB2STR		√	√		00H	–	–	–	√
FFFFFD24H	CSIB2 receive data register	CB2RX	R			√	0000H	–	–	–	√
FFFFFD24H	CSIB2 receive data register L	CB2RXL			√		00H	–	–	–	√
FFFFFD26H	CSIB2 transmit data register	CB2TX	R/W			√	0000H	–	–	–	√
FFFFFD26H	CSIB2 transmit data register L	CB2TXL			√		00H	–	–	–	√
FFFFFD80H	IIC0 shift register	IIC0			√		00H	√	√	√	√
FFFFFD82H	IIC0 control register	IICC0		√	√		00H	√	√	√	√
FFFFFD83H	IIC0 slave address register	SVA0			√		00H	√	√	√	√
FFFFFD84H	IIC0 clock selection register	IICCL0		√	√		00H	√	√	√	√
FFFFFD85H	IIC0 function expansion register	IICX0		√	√		00H	√	√	√	√
FFFFFD86H	IIC0 status register	IICS0	R	√	√		00H	√	√	√	√
FFFFFD8AH	IIC0 flag register	IICF0	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/Hx3 has the following ten special registers.

- Power save control register (PSC)
- Processor clock control register (PCC)
- Clock monitor mode register (CLM)
- Main system clock mode register (MCM)
- SSCG frequency control register 0 (SFC0)
- SSCG frequency control register 1 (SFC1)
- Reset source flag register (RESF)
- Low-voltage detection register (LVIM)
- Internal RAM data status register (RAMS)
- 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 (reported even when the read operation of the option data (address: 007BH) is illegal because of noise, instantaneous voltage drop, etc.).

(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.

[Example] With 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>NOP[Note]            ; Dummy instruction
<6>NOP[Note]            ; Dummy instruction
<7>NOP[Note]            ; Dummy instruction
<8>NOP[Note]            ; Dummy instruction
<9>NOP[Note]            ; Dummy instruction
<10>SET1 0, DCHCn[r0]   ; Enable DMA operation.  n = 0 to 3
(next instruction)
```

There is no special sequence to read a special register.

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

Cautions 1. When a store instruction is executed to store data in the command register, interrupts are not acknowledged. This is because it is assumed that steps <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.

2. Although dummy data is written to the PRCMD register, use the same general-purpose register used to set the special register (<4> in Example) to write data to the PRCMD register (<3> in 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).

After reset: Undefined W Address: FFFFF1FCH

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

(3) System status register (SYS)

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

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

Reset 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 Between an operation to write the PRCMD register and an operation to write a special register, even if the internal RAM is accessed, such as reading when an on-chip peripheral I/O register (except reading by a bit manipulation instruction), 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 Cautions

(1) Registers to be set first

Be sure to set the following registers first when using the V850ES/Hx3.

- 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.

When using the external bus, set each pin to the alternate-function bus control pin mode by using the port-related registers after setting the above registers.

(a) System wait control register (VSWC)

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

Three clocks are required to access an on-chip peripheral I/O register (without a wait cycle). The V850ES/Hx3 requires wait cycles according to the operating frequency. Set the following value to the VSWC register in accordance with the frequency used.

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

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 kHz ≤ f_{CLK} < 16.6 MHz	00H	0 (no waits)
16.6 MHz ≤ f_{CLK} < 25 MHz	01H	1
25 MHz ≤ f_{CLK} ≤ 32 MHz	11H	2

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

For details, see **CHAPTER 28 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. Write the WDTM2 register to activate this operation.

For details, see **CHAPTER 12 FUNCTIONS OF WATCHDOG TIMER 2**.

(2) Accessing specific 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 state. If this wait state 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 specific 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.

Peripheral Function	Register Name	Access	k
16-bit timer/event counter AA (TAA) (n = 0 to 3)	TAAnCNT	Read	1
	TAAnCCR0, TAAnCCR1	Write	• 1st access: No wait • Continuous write: 3 or 4
		Read	1
16-bit timer/event counter AB (TAB) (m = 0 to 2)	TABmCNT	Read	1
	TABmCCR0 to TABmCCR3	Write	• 1st access: No wait • Continuous write: 3 or 4
		Read	1
Motor control function	TAB0OPT1	Write	• 1st access: No wait • Continuous write: 2 to 4
	TAB0DTC	Write	• 1st access: No wait • Continuous write: 2 to 4
Watchdog timer 2 (WDT2)	WDTM2	Write (when WDT2 operating)	3
A/D converter	ADA0M0	Read	1 to 3
	ADA0CR0 to ADA0CR23	Read	1 to 3
	ADA0CR0H to ADA0CR23H	Read	1 to 3
I^2C00	IICS0	Read	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 with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the low-speed internal oscillation clock**

Remark i: Values (0 or 1) of higher 4 bits of VSWC register

j: Values (0 or 1) of lower 4 bits of VSWC register

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

(a) Description

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

Instruction <1>

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

Instruction <2>

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

<Example>

<i> ld.w [r11], r10
•
•
•
<ii> mov r10, r28
<iii> sld.w 0x28, r10

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

(b) Countermeasure

<1> When compiler (CA850) is used

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

<2> Countermeasure by 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/HE3

- O I/O ports: 51
- O Port pins function alternately as other peripheral-function I/O pins
- O Can be set in input or output mode in 1-bit units.

4.1.2 V850ES/HF3

- O I/O ports: 67
- O Port pins function alternately as other peripheral-function I/O pins
- O Can be set in input or output mode in 1-bit units.

4.1.3 V850ES/HG3

- O I/O ports: 84
- O Port pins function alternately as other peripheral-function I/O pins
- O Can be set in input or output mode in 1-bit units.

4.1.4 V850ES/HJ3

- O I/O ports: 128
- O Port pins function alternately as other peripheral-function I/O pins
- O Can be set in input or output mode in 1-bit units.

4.2 Basic Configuration of Ports

4.2.1 V850ES/HE3

The V850ES/HE3 has a total of 51 I/O ports, ports 0, 3 to 5, 7, 9, CM, and DL. The port configuration is shown below.

Figure 4-1. Port Configuration (V850ES/HE3)

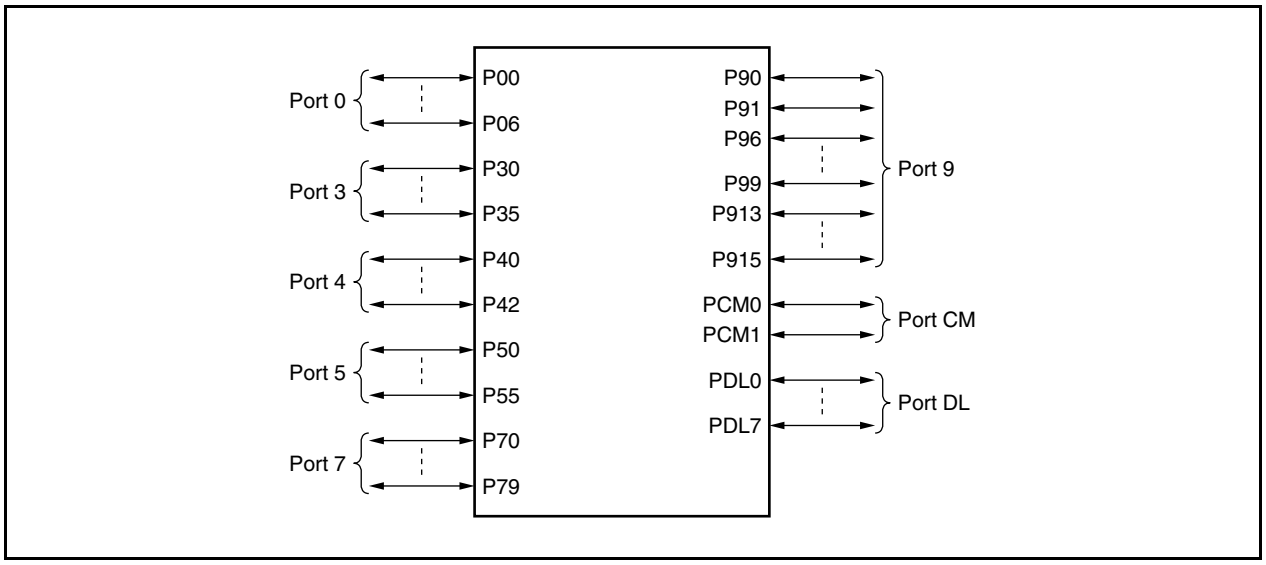


Table 4-1. Configuration of Ports (V850ES/HE3)

Item	Configuration
Control registers	Port mode register (PMn: n = 0, 3, 4, 5, 7L, 7H, 9, CM, or DL)
	Port mode control register (PMCn: n = 0, 3, 4, 5, 7L, 7H, 9, or CM)
	Port function control register (PFCn: n = 0, 3L, 4, 5, or 9)
	Port function control expansion register (PFCEn: n = 0, 3L, 5, or 9)
	Pull-up resistor option register (PUn: n = 0, 3, 4, 5, or 9)
Ports	51

Table 4-2. Pin I/O Buffer Power Supplies (V850ES/HE3)

Power Supply	Corresponding Pin
AV_{REF0}	Port 7
EV_{DD}	Port 0, port 3, port 4, port 5, port 9, port CM, port DL, $\overline{RESET}$, FLMD0

4.2.2 V850ES/HF3

The V850ES/HF3 has a total of 67 I/O ports, ports 0, 3 to 5, 7, 9, CM, CS, CT, and DL. The port configuration is shown below.

Figure 4-2. Port Configuration (V850ES/HF3)

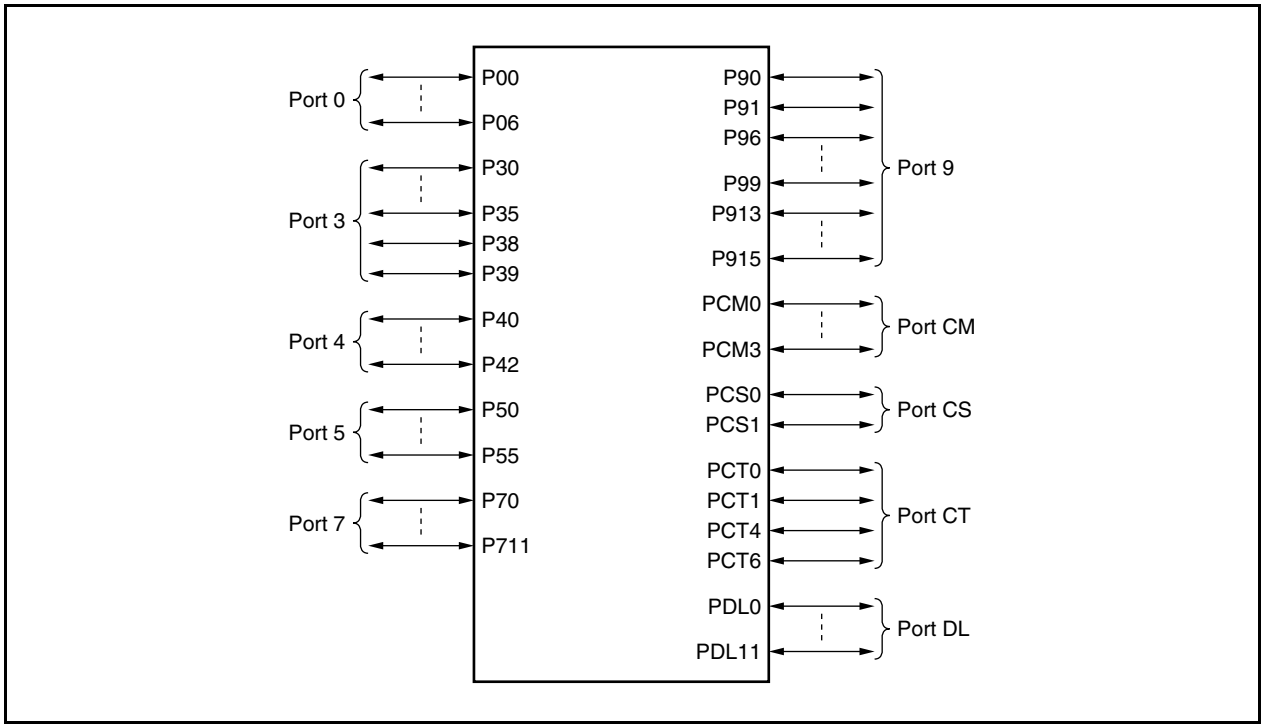


Table 4-3. Configuration of Ports (V850ES/HF3)

Item	Configuration
Control registers	Port mode register (PMn: n = 0, 3, 4, 5, 7L, 7H, 9, CM, CS, CT, or DL)
	Port mode control register (PMCn: n = 0, 3, 4, 5, 7L, 7H, 9, or CM)
	Port function control register (PFCn: n = 0, 3L, 4, 5, or 9)
	Port function control expansion register (PFCEn: n = 0, 3L, 5, or 9)
	Pull-up resistor option register (PUn: n = 0, 3, 4, 5, or 9)
Ports	67

Table 4-4. Pin I/O Buffer Power Supplies (V850ES/HF3)

Power Supply	Corresponding Pin
AV_{REF0}	Port 7
EV_{DD}	Port 0, port 3, port 4, port 5, port 9, port CM, port CS, port CT, port DL, $\overline{RESET}$, FLMD0

4.2.3 V850ES/HG3

The V850ES/HG3 has a total of 84 I/O ports, ports 0, 1, 3 to 5, 7, 9, CM, CS, CT, and DL. The port configuration is shown below.

Figure 4-3. Port Configuration (V850ES/HG3)

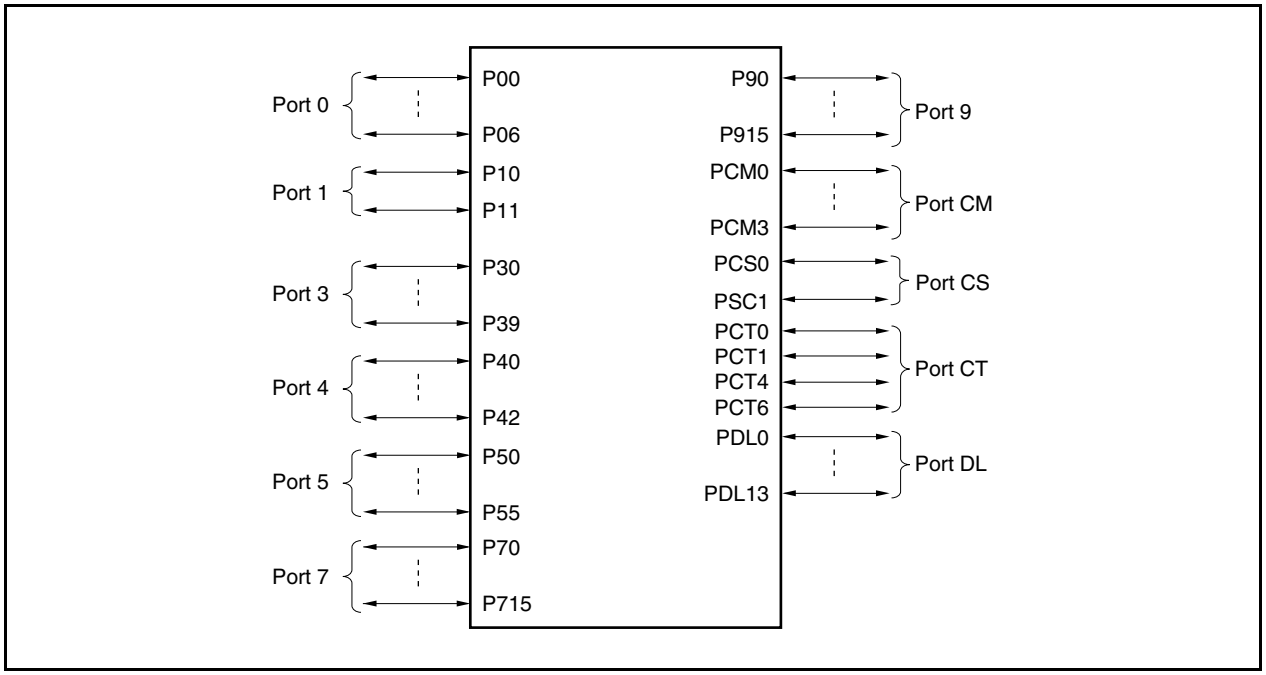


Table 4-5. Configuration of Ports (V850ES/HG3)

Item	Configuration
Control registers	Port mode register (PMn: n = 0, 1, 3, 4, 5, 7L, 7H, 9, CM, CS, CT, or DL)
	Port mode control register (PMCn: n = 0, 1, 3, 4, 5, 7L, 7H, 9, or CM)
	Port function control register (PFCn: n = 0, 3L, 4, 5, or 9)
	Port function control expansion register (PFCEn: n = 0, 3L, 5, or 9)
	Pull-up resistor option register (PUn: n = 0, 1, 3, 4, 5, or 9)
Ports	84

Table 4-6. Pin I/O Buffer Power Supplies (V850ES/HG3)

Power Supply	Corresponding Pin
AV_{REF0}	Port 7
BV_{DD}	Port CM, port CS, port CT, port DL
EV_{DD}	Port 0, port 1, port 3, port 4, port 5, port 9, $\overline{\text{RESET}}$, FLMD0

4.2.4 V850ES/HJ3

The V850ES/HJ3 has a total of 128 I/O ports, ports 0, 1, 3 to 9, 12, CD, CM, CS, CT, and DL. The port configuration is shown below.

Figure 4-4. Port Configuration (V850ES/HJ3)

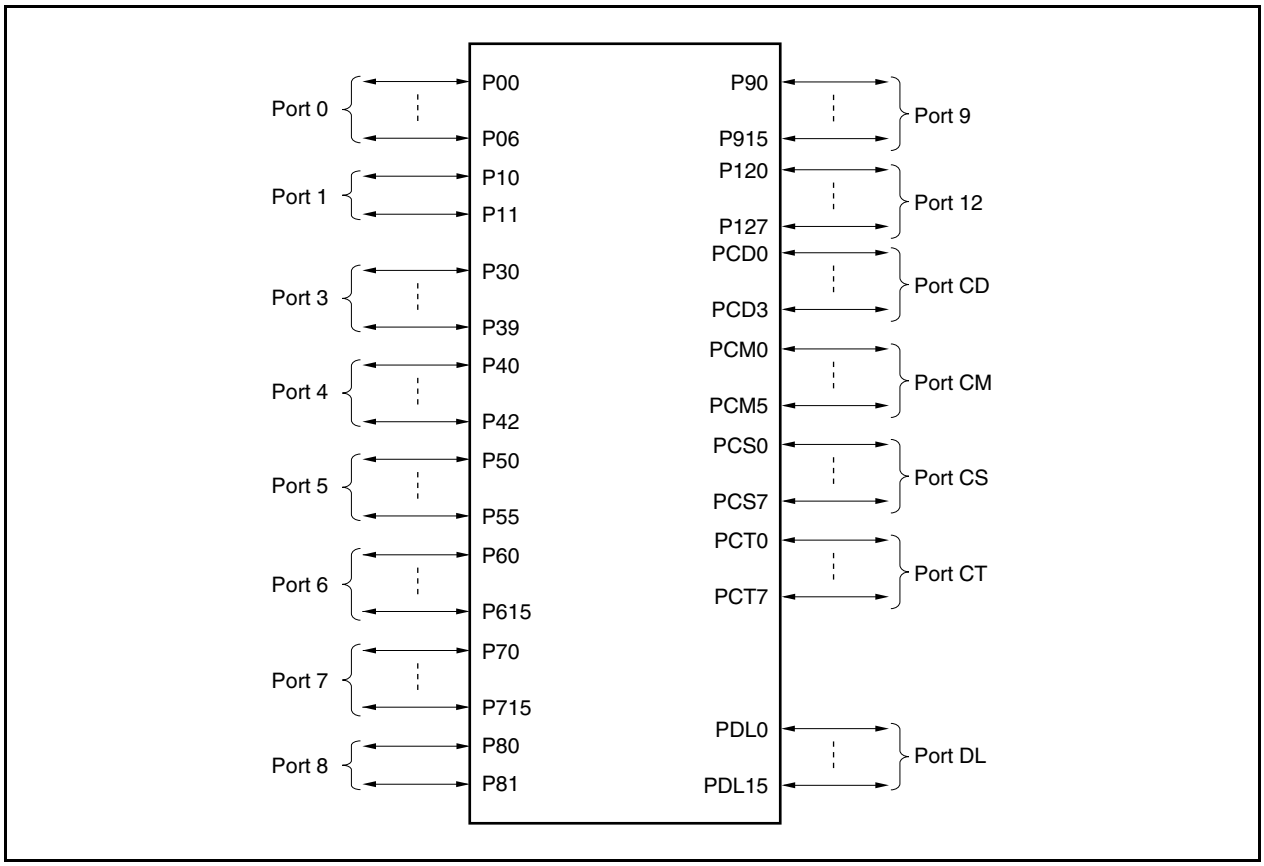


Table 4-7. Configuration of Ports (V850ES/HJ3)

Item	Configuration
Control registers	Port mode register (PMn: n = 0, 1, 3, 4, 5, 6, 7L, 7H, 8, 9, 12, CD, CM, CS, CT, or DL)
	Port mode control register (PMCn: n = 0, 1, 3, 4, 5, 6, 7L, 7H, 8, 9, 12, CM, CS, CT, or DL)
	Port function control register (PFCn: n = 0, 3L, 4, 5, 6, 9, or 12)
	Port function control expansion register (PFCEn: n = 0, 3L, 4[Note], 5, or 9)
	Port function register (PF9)
	Pull-up resistor option register (PUn: n = 0, 1, 3, 4, 5, 6, 8, or 9)
Ports	128

Note μPD70F3737 only

Table 4-8. Pin I/O Buffer Power Supplies (V850ES/HJ3)

Power Supply	Corresponding Pin
AV_{REF0}	Port 7, port 12
BV_{DD}	Port CD, port CM, port CS, port CT, port DL
EV_{DD}	Port 0, port 1, port 3, port 4, port 5, port 6, port 8, port 9, $\overline{\text{RESET}}$, FLMD0

4.3 Port Configuration

(1) Port n register (Pn)

Data is input from or output to an external device by writing or reading the Pn register. The Pn register consists of a port latch that holds output data, and a circuit that reads the status of pins.

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

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

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

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

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

Table 4-9. Writing/Reading Pn Register

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

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

(2) Port n mode register (PMn)

The PMn register specifies the I/O mode of the corresponding port pin.

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

After reset: FFH R/W

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

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

(3) Port n mode control register (PMCn)

The PMCn register specifies the port mode or alternate function.

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

After reset: 00H R/W

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

PMCnm	Specification of operation mode
0	Port mode
1	Alternate function mode

(4) Port n function control register (PFCn)

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

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

After reset: 00H R/W

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

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

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

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

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

After reset: 00H R/W

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

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

(6) Port n function register (PFn)

The PFn register specifies normal output or N-ch open-drain output.

Each bit of this register corresponds to one pin of port n, and the output mode of the port pin 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]	Control of normal output/N-ch open-drain output
0	Normal output (CMOS output)
1	N-ch open-drain output

Note The PFnm bit of the PFn register is valid only when the PMnm bit of the PMn register is 0 (when the output mode is specified) in port mode (PMCnm bit = 0). When the PMnm bit is 1 (when the input mode is specified), the set value of the PFn register is invalid.

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

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

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

After reset: 00H R/W

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

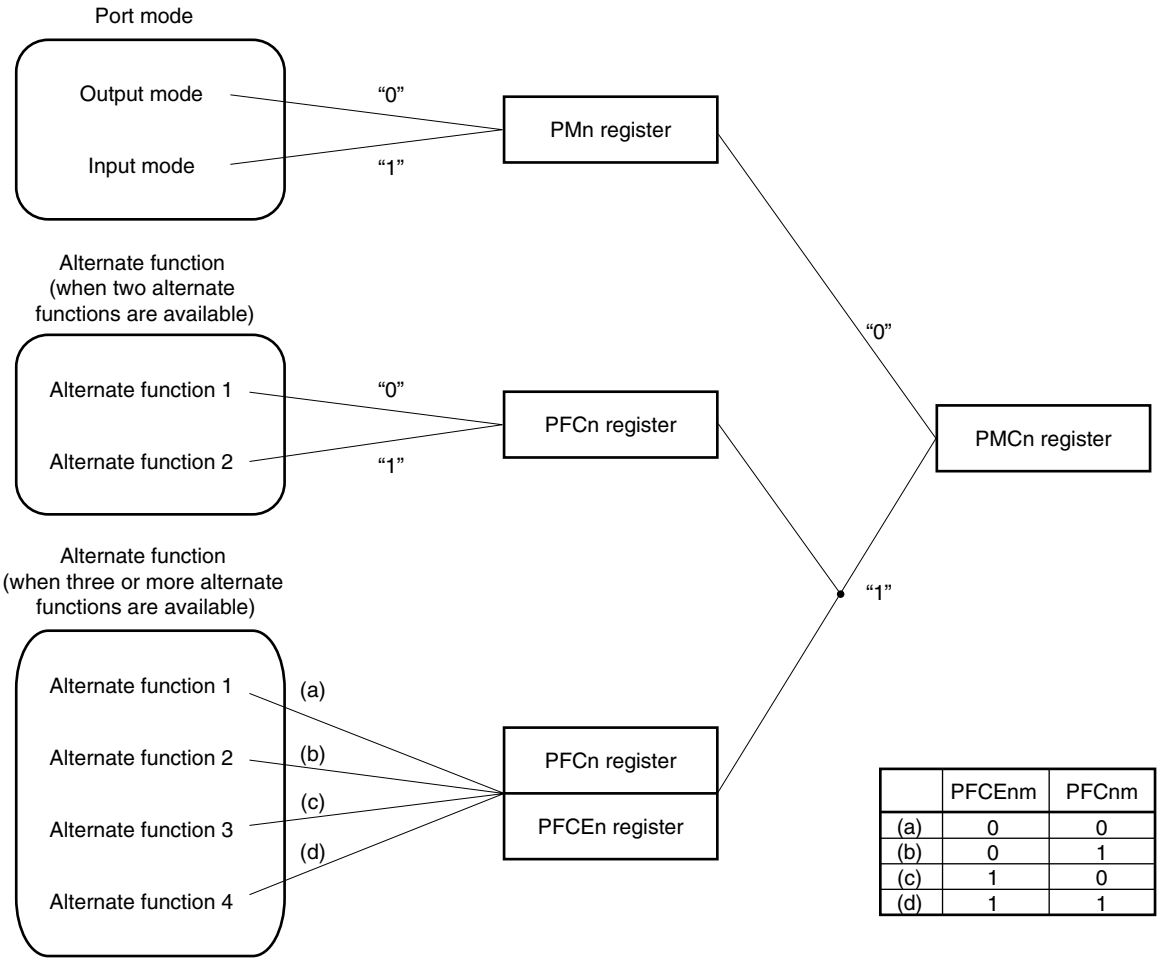
PUnm[Note]	Control of on-chip pull-up resistor connection
0	Not connected
1	Connected

Note The on-chip pull-up resistor is valid only in the input mode (PMnm bit = 1). In the output mode (PMnm bit = 0), the set value of the PUn register is invalid.

(8) Port setting

Set a port as illustrated below.

Figure 4-5. Setting of Each Register and Pin Function



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

Remark Set the alternate functions in the following sequence.

<1> Set the PFCn and PFCEn registers.
<2> Set the PMCn register.
<3> Set the INTRn or INTFn register (to specify an external interrupt pin).

If the PMCn register is set first, an unintended function may be set while the PFCn and PFCEn registers are being set.

4.3.1 Port 0

Port 0 I/O settings can be controlled in 1-bit units.

Each product has the same number of I/O ports.

Generic Name	Number of I/O Ports
V850ES/HE3	7-bit I/O port
V850ES/HF3	7-bit I/O port
V850ES/HG3	7-bit I/O port
V850ES/HJ3	7-bit I/O port

Table 4-10. Alternate-Function Pins of Port 0

Function Name	Alternate-Function Name		Pin No.				Remark	Block Type
	Name	I/O	HE3	HF3	HG3	HJ3		
P00	TIAA31/TOAA31	I/O	12	3	6	6	–	E10-U
P01	TIAA30/TOAA30	I/O	13	4	7	7		E10-U
P02	NMI[Note 1]/TIAA40/TOAA40	I/O	14	5	17	17		F1x10-UI
P03	INTP0/ADTRG/TIAA41/TOAA41	I/O	15	6	18	18		F1110-UI
P04	INTP1	Input	16	7	19	19		D1-U
P05	INTP2/$\overline{\text{DRST}}$[Note 2]	Input	17	17	20	20		D1O1-UI
P06	INTP3	Input	18	18	21	21		D1-UI

Notes 1. The NMI pin alternately functions as the P02 pin. It functions as the P02 pin after reset.
To enable the NMI pin, 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 INTF0 and INTR0 registers.

2. The alternate function of the P05 pin is the on-chip debug function. After external reset, the P05/INTP2/$\overline{\text{DRST}}$ pin is initialized as the on-chip debug pin ($\overline{\text{DRST}}$). To use the P05 pin as a port pin, not as an on-chip debug pin, the following actions must be taken.

<1> Clear the OCDM0 bit of the OCDM register (special register) to 0.
<2> Fix the P05/INTP2/$\overline{\text{DRST}}$ pin to the low level until the above action has been taken.

When the on-chip debug function is not used, inputting a high level to the $\overline{\text{DRST}}$ pin before the above actions are taken may cause a malfunction. Exercise utmost care in handling the P05 pin.
When a high level is not input to the P05/INTP2/$\overline{\text{DRST}}$ pin (when this pin is fixed to low level), it is not necessary to manipulate the OCDM0 bit of the OCDM register.
The P05/INTP2/$\overline{\text{DRST}}$ pin does not have to be externally pulled down, because it has an on-chip pull-down resistor (30 kΩ typ.). The pull-down resistor is disconnected by clearing the OCDM0 bit to 0.

Caution The P00 to P06 pins have hysteresis characteristics in the input mode of the alternate function, but do not have hysteresis characteristics in the port mode.

(1) Port register 0 (P0)

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

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

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

(2) Port mode register 0 (PM0)

After reset: FFH R/W Address: FFFFF420H

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

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

(3) Port mode control register 0 (PMC0)

After reset: 00H R/W Address: FFFFF440H

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

PMC06	Specification of P06 pin operation mode
0	I/O port
1	INTP3 input

PMC05	Specification of P05 pin operation mode
0	I/O port
1	INTP2 input

PMC04	Specification of P04 pin operation mode
0	I/O port
1	INTP1 input

PMC03	Specification of P03 pin operation mode
0	I/O port
1	INTP0 input/ADTRG input/TIAA41 input/TOAA41 output

PMC02	Specification of P02 pin operation mode
0	I/O port
1	NMI input/TIAA40 input/TOAA40 output

PMC01	Specification of P01 pin operation mode
0	I/O port
1	TIAA30 input/TOAA30 output

PMC00	Specification of P00 pin operation mode
0	I/O port
1	TIAA31 input/TOAA31 output

(4) Port function control register 0 (PFC0)

After reset: 00H R/W Address: FFFFF460H

	7	6	5	4	3	2	1	0
PFC0	0	0	0	0	PFC03	PFC02	PFC01	PFC00

Remark For the specifications of alternate functions, see **4.3.1 (6) Settings of alternate functions of port 0**.

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

After reset: 00H R/W Address: FFFFF700H

	7	6	5	4	3	2	1	0
PFCE0	0	0	0	0	PFCE03	PFCE02	0	0

Remark For the specifications of alternate functions, see **4.3.1 (6) Settings of alternate functions of port 0**.

(6) Settings of alternate functions of port 0

PFCE03	PFC03	Specification of P03 Pin Alternate Function
0	0	INTP0 input
0	1	ADTRG input
1	0	TIAA41 input
1	1	TOAA41 output

PFCE02	PFC02	Specification of P02 Pin Alternate Function
0	0	NMI input
0	1	Setting prohibited
1	0	TIAA40 input
1	1	TOAA40 output

PFC01	Specification of P01 Pin Alternate Function
0	TIAA30 input
1	TOAA30 output

PFC00	Specification of P00 Pin Alternate Function
0	TIAA31 input
1	TOAA31 output

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

After reset: 00H R/W Address: FFFFFC40H

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

PU0n	Control of on-chip pull-up resistor connection (n = 0 to 6)
0	Do not connect
1	Connect

4.3.2 Port 1

Port 1 I/O settings can be controlled in 1-bit units.

The number of I/O ports differs for each product.

Generic Name	Number of I/O Ports
V850ES/HE3	–
V850ES/HF3	–
V850ES/HG3	2-bit I/O port
V850ES/HJ3	2-bit I/O port

Table 4-11. Alternate-Function Pins of Port 1

Function Name	Alternate-Function Name		Pin No.				Remark	Block Type
	Name	I/O	HE3	HF3	HG3	HJ3		
P10	INTP9	Input	–	–	3	3	–	D1-UI
P11	INTP10	Input	–	–	4	4		D1-UI

Caution The P10 and P11 pins have hysteresis characteristics in the input mode of the alternate function, but do not have hysteresis characteristics in the port mode.

(1) Port register 1 (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	P11	P10

P1n	Output data control (in output mode) (n = 0, 1)
0	Outputs 0
1	Outputs 1

(2) Port mode register 1 (PM1)

After reset: FFH R/W Address: FFFFF422H

	7	6	5	4	3	2	1	0
PM1	1	1	1	1	1	1	PM11	PM10

PM1n	I/O mode control (n = 0, 1)
0	Output mode
1	Input mode

(3) Port mode control register 1 (PMC1)

After reset: 00H R/W Address: FFFFF442H

	7	6	5	4	3	2	1	0
PMC1	0	0	0	0	0	0	PMC11	PMC10

PMC11	Specification of P11 pin operation mode
0	I/O port
1	INTP10 input

PMC10	Specification of P10 pin operation mode
0	I/O port
1	INTP9 input

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

After reset: 00H R/W Address: FFFFFC42H

	7	6	5	4	3	2	1	0
PU1	0	0	0	0	0	0	PU11	PU10

PU1n	Control of on-chip pull-up resistor connection (n = 0, 1)
0	Do not connect
1	Connect

4.3.3 Port 3

Port 3 I/O settings can be controlled in 1-bit units.

The number of I/O ports differs for each product.

Generic Name	Number of I/O Ports
V850ES/HE3	6-bit I/O port
V850ES/HF3	8-bit I/O port
V850ES/HG3	10-bit I/O port
V850ES/HJ3	10-bit I/O port

Table 4-12. Alternate-Function Pins of Port 3

Function Name	Alternate-Function Name		Pin No.				Remark	Block Type
	Name	I/O	HE3	HF3	HG3	HJ3		
P30	TXDD0	Output	22	22	25	25	–	D0-U
P31	RXDD0/INTP7	Input	23	23	26	26		D3-UI
P32	ASCKD0/TOAA01/TIAA00/TOAA00	I/O	24	24	27	27		F1010-U
P33	TIAA01/TOAA01	I/O	25	25	28	28		E10-U
P34	TIAA10/TOAA10	I/O	26	26	29	29		E10-U
P35	TIAA11/TOAA11	I/O	27	27	30	30		E10-U
P36	–	–	–	–	31	31		C-U
P37	–	–	–	–	32	32		C-U
P38	–	–	–	28	–	–		C-U
	TXDD2	Output	–	–	35	35		D0-U
P39	–	–	–	29	–	–		C-U
	RXDD2/INTP8	Input	–	–	36	36		D3-UI

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

(1) Port register 3 (P3)

(a) V850ES/HE3

After reset: 00H (output latch) R/W Address: FFFFF406H

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

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

(b) V850ES/HF3

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

(c) V850ES/HG3, V850ES/HJ3

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)	P37	P36	P35	P34	P33	P32	P31	P30

P3n	Output data control (in output mode) (n = 0 to 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 mode register 3 (PM3)

(a) V850ES/HE3

After reset: FFH R/W Address: FFFFF426H

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

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

(b) V850ES/HF3

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	PM35	PM34	PM33	PM32	PM31	PM30

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

(c) V850ES/HG3, V850ES/HJ3

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)	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30

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

Remarks 1. 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.

2. 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 mode control register 3 (PMC3)

(a) V850ES/HE3, V850ES/HF3

After reset: 00H R/W Address: FFFFF446H

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

PMC35	Specification of P35 pin operation mode
0	I/O port
1	TIAA11 input/TOAA11 output

PMC34	Specification of P34 pin operation mode
0	I/O port
1	TIAA10 input/TOAA10 output

PMC33	Specification of P33 pin operation mode
0	I/O port
1	TIAA01 input/TOAA01 output

PMC32	Specification of P32 pin operation mode
0	I/O port
1	ASCKD0 input/TOAA01 output/TIAA00 input/TOAA00 output

PMC31	Specification of P31 pin operation mode
0	I/O port
1	RXDD0 input/INTP7 input[Note]

PMC30	Specification of P30 pin operation mode
0	I/O port
1	TXDD0 output

Note The INTP7 function and RXDD0 function are alternately used. When using as the RXDD0 function, disable edge detection for the INTP7 function (set the INTF3.INTF31 bit and the INTR3.INTR31 bit to 0). When using as the INTP7 function, stop UARTD0 reception (set the UD0CTL0.UD0RXE bit to 0).

(2/3)

(b) V850ES/HG3, V850ES/HJ3 (1/2)

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	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30

PMC39	Specification of P39 pin operation mode
0	I/O port
1	RXDD2 input/INTP8 input[Note]

PMC38	Specification of P38 pin operation mode
0	I/O port
1	TXDD2 output

PMC35	Specification of P35 pin operation mode
0	I/O port
1	TIAA11 input/TOAA11 output

PMC34	Specification of P34 pin operation mode
0	I/O port
1	TIAA10 input/TOAA10 output

PMC33	Specification of P33 pin operation mode
0	I/O port
1	TIAA01 input/TOAA01 output

PMC32	Specification of P32 pin operation mode
0	I/O port
1	ASCKD0 input/TOAA01 output/TIAA00 input/TOAA00 output

Note The INTP8 function and RXDD2 function are alternately used. When using as the RXDD2 function, disable edge detection for the INTP8 function (set the INTF3.INTF39 bit and the INTR3.INTR39 bit to 0). When using as the INTP8 function, stop UARTD2 reception (set the UD2CTL0.UD2RXE bit to 0).

Remarks 1. 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.

2. 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.

(3/3)

(b) V850ES/HG3, V850ES/HJ3 (2/2)

PMC31	Specification of P31 pin operation mode
0	I/O port
1	RXDD0 input/INTP7 input[Note]

PMC30	Specification of P30 pin operation mode
0	I/O port
1	TXDD0 output

Note The INTP7 function and RXDD0 function are alternately used. When using as the RXDD0 function, disable edge detection for the INTP7 function (set the INTF3.INTF31 bit and the INTR3.INTR31 bit to 0). When using as the INTP7 function, stop UARTD0 reception (set the UD0CTL0.UD0RXE bit to 0).

(4) Port function control register 3L (PFC3L)

After reset: 00H R/W Address: FFFFF466H

	7	6	5	4	3	2	1	0
PFC3L	0	0	PFC35	PFC34	PFC33	PFC32	0	0

Remark For the specifications of alternate functions, see **4.3.3 (6) Settings of alternate functions of port 3**.

(5) Port function control expansion register 3L (PFCE3L)

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 the specifications of alternate functions, see **4.3.3 (6) Settings of alternate functions of port 3**.

(6) Settings of alternate functions of port 3

PFC35	Specification of P35 Pin Control Mode
0	TIAA11 input
1	TOAA11 output

PFC34	Specification of P34 Pin Control Mode
0	TIAA10 input
1	TOAA10 output

PFC33	Specification of P33 Pin Control Mode
0	TIAA01 input
1	TOAA01 output

PFCE32	PFC32	Specification of P32 Pin Alternate Function
0	0	ASCKD0 input
0	1	TOAA01 output
1	0	TIAA00 input
1	1	TOAA00 output

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

(a) V850ES/HE3

After reset: 00H R/W Address: FFFFFC46H

	7	6	5	4	3	2	1	0
PU3L	0	0	PU35	PU34	PU33	PU32	PU31	PU30

PU3n	Control of on-chip pull-up resistor connection (n = 0 to 5)
0	Do not connect
1	Connect

(b) V850ES/HF3

After reset: 0000H R/W Address: PU3 FFFFFC46H,
PU3L FFFFFC46H, PU3H FFFFFC47H

	15	14	13	12	11	10	9	8
PU3 (PU3H)	0	0	0	0	0	0	PU39	PU38

	7	6	5	4	3	2	1	0
(PU3L)	0	0	PU35	PU34	PU33	PU32	PU31	PU30

PU3n	Control of on-chip pull-up resistor connection (n = 0 to 5, 8, 9)
0	Do not connect
1	Connect

(c) V850ES/HG3, V850ES/HJ3

After reset: 0000H R/W Address: PU3 FFFFFC46H,
PU3L FFFFFC46H, PU3H FFFFFC47H

	15	14	13	12	11	10	9	8
PU3 (PU3H)	0	0	0	0	0	0	PU39	PU38

	7	6	5	4	3	2	1	0
(PU3L)	PU37	PU36	PU35	PU34	PU33	PU32	PU31	PU30

PU3n	Control of on-chip pull-up resistor connection (n = 0 to 9)
0	Do not connect
1	Connect

Remarks 1. The PU3 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PU3 register as the PU3H register and the lower 8 bits as the PU3L register, PU3 can be read or written in 8-bit or 1-bit units.

2. To read/write bits 8 to 15 of the PU3 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PU3H register.

4.3.4 Port 4

Port 4 I/O settings can be controlled in 1-bit units.

Each product has the same number of I/O ports, but the alternate functions of the pins differ.

Generic Name	Number of I/O Ports
V850ES/HE3	3-bit I/O port
V850ES/HF3	3-bit I/O port
V850ES/HG3	3-bit I/O port
V850ES/HJ3	3-bit I/O port

Table 4-13. Alternate-Function Pins of Port 4

Function Name	Alternate-Function Name		Pin No.				Remark	Block Type
	Name	I/O	HE3	HF3	HG3	HJ3		
P40	SIB0/KR0	Input	19	19	22	22 Note 1	–	E11-U
	SIB0/KR0/RXDD3/INTP14	Input	–	–	–	22 Note 2		F113x-UI
P41	SOB0/KR1	I/O	20	20	23	23 Note 1		E01-U
	SOB0/KR1/TXDD3	I/O	–	–	–	23 Note 2		F010x-U
P42	$\overline{\text{SCKB0}}$/KR2	I/O	21	21	24	24		E21-U

Notes 1. μPD70F3755 only

2. μPD70F3757 only

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

(1) Port register 4 (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 mode register 4 (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 mode control register 4 (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

PMC42	Specification of P42 pin operation mode
0	I/O port
1	$\overline{\text{SCKB0}}$ I/O/KR2 input

PMC41	Specification of P41 pin operation mode
0	I/O port
1	SOB0 output/KR1 input/TXDD3 output[Note]

PMC40	Specification of P40 pin operation mode
0	I/O port
1	SIB0 input/KR0 input/RXDD3 input[Note]/INTP14 input[Note]

Note μPD70F3757 only

(4) Port function control register 4 (PFC4)

After reset: 00H R/W Address: FFFFF468H

	7	6	5	4	3	2	1	0
PFC4	0	0	0	0	0	PFC42	PFC41	PFC40

Remark For the specifications of alternate functions, see **4.3.4 (6) Settings of alternate functions of port 4**.

(5) Port function control expansion register 4 (PFCE4) (μPD70F3757 only)

After reset: 00H R/W Address: FFFFF708H

	7	6	5	4	3	2	1	0
PFCE4	0	0	0	0	0	0	PFCE41	PFCE40

Remark For the specifications of alternate functions, see **4.3.4 (6) Settings of alternate functions of port 4**.

(6) Settings of alternate functions of port 4

PFC42	Specification of P42 Pin Alternate Function
0	$\overline{\text{SCKB0}}$ I/O
1	KR2 input

PFCE41[Note 1]	PFC41	Specification of P41 Pin Alternate Function
0	0	SOB0 output
0	1	KR1 input
1	0	TXDD3 output[Note 1]
1	1	Setting prohibited[Note 1]

PFCE40[Note 1]	PFC40	Specification of P40 Pin Alternate Function
0	0	SIB0 input
0	1	KR0 input
1	0	RXDD3 input/INTP14 input[Notes 1, 2]
1	1	Setting prohibited[Note 1]

Notes 1. μPD70F3757 only

2. The INTP14 function and RXDD3 function are alternately used. When using as the RXDD3 function, disable edge detection for the INTP14 function (set the INTF4.INTF40 bit and the INTR4.INTR40 bit to 0). When using as the INTP14 function, stop UARTD3 reception (set the UD3CTL0.UD3RXE bit to 0) (μPD70F3757 only).

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

After reset: 00H R/W Address: FFFFFC48H

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

PU4n	Control of on-chip pull-up resistor connection (n = 0 to 2)
0	Do not connect
1	Connect

4.3.5 Port 5

Port 5 I/O settings can be controlled in 1-bit units.

The number of I/O ports differs for each product.

Generic Name	Number of I/O Ports
V850ES/HE3	6-bit I/O port
V850ES/HF3	6-bit I/O port
V850ES/HG3	6-bit I/O port
V850ES/HJ3	6-bit I/O port

Table 4-14. Alternate-Function Pins of Port 5

Function Name	Alternate-Function Name		Pin No.				Remark	Block Type
	Name	I/O	HE3	HF3	HG3	HJ3		
P50	KR0/TIAB01/TOAB01/TOAB0T1	I/O	28	32	37	37	–	F1100-U
P51	KR1/TIAB02/TOAB02/TOAB0B1	I/O	29	33	38	38		F1100-U
P52	KR2/TIAB03/TOAB03/TOAB0T2/DDI[Note]	I/O	30	34	39	39		F1100O1-U
P53	KR3/TIAB00/TOAB00/TOAB0B2/DDO[Note]	I/O	31	35	40	40		F1100O0-U
P54	KR4/TOAB0T3/DCK[Note]	I/O	34	36	41	41		F1xx0O1-U
P55	KR5/TOAB0B3/DMS[Note]	I/O	35	37	42	42		F1xx0O1-U

Note The DDI, DDO, DCK, and DMS pins are for the on-chip debug function. The port function of these pins must not be selected.

For details, see **CHAPTER 28 ON-CHIP DEBUG FUNCTION**.

Cautions 1. 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.

2. When the power is turned on, the P53 pin may momentarily output an undefined level.

(1) Port register 5 (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 mode register 5 (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 mode control register 5 (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 P55 pin operation mode
0	I/O port
1	KR5 input/TOAB0B3 output

PMC54	Specification of P54 pin operation mode
0	I/O port
1	KR4 input/TOAB0T3 output

PMC53	Specification of P53 pin operation mode
0	I/O port
1	KR3 input/TIAB00 input/TOAB00 output/TOAB0B2 output

PMC52	Specification of P52 pin operation mode
0	I/O port
1	KR2 input/TIAB03 input/TOAB03 output/TOAB0T2 output

PMC51	Specification of P51 pin operation mode
0	I/O port
1	KR1 input/TIAB02 input/TOAB02 output/TOAB0B1 output

PMC50	Specification of P50 pin operation mode
0	I/O port
1	KR0 input/TIAB01 input/TOAB01 output/TOAB0T1 output

(4) Port function control register 5 (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 the specifications of alternate functions, see **4.3.5 (6) Settings of alternate functions of port 5**.

(5) Port function control expansion register 5 (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 the specifications of alternate functions, see **4.3.5 (6) Settings of alternate functions of port 5**.

(6) Settings of alternate functions of port 5

PFCE55	PFC55	Specification of P55 Pin Alternate Function
0	0	KR5 input
0	1	Setting prohibited
1	0	Setting prohibited
1	1	TOAB0B3 output

PFCE54	PFC54	Specification of P54 Pin Alternate Function
0	0	KR4 input
0	1	Setting prohibited
1	0	Setting prohibited
1	1	TOAB0T3 output

PFCE53	PFC53	Specification of P53 Pin Alternate Function
0	0	KR3 input
0	1	TIAB00 input
1	0	TOAB00 output
1	1	TOAB0B2 output

PFCE52	PFC52	Specification of P52 Pin Alternate Function
0	0	KR2 input
0	1	TIAB03 input
1	0	TOAB03 output
1	1	TOAB0T2 output

PFCE51	PFC51	Specification of P51 Pin Alternate Function
0	0	KR1 input
0	1	TIAB02 input
1	0	TOAB02 output
1	1	TOAB0B1 output

PFCE50	PFC50	Specification of P50 Pin Alternate Function
0	0	KR0 input
0	1	TIAB01 input
1	0	TOAB01 output
1	1	TOAB0T1 output

(7) Pull-up resistor option register 5 (PU5)

After reset: 00H R/W Address: FFFFFC4AH

	7	6	5	4	3	2	1	0
PU5	0	0	PU55	PU54	PU53	PU52	PU51	PU50

PU5n	Control of on-chip pull-up resistor connection (n = 0 to 5)
0	Do not connect
1	Connect

4.3.6 Port 6 (V850ES/HJ3 only)

Port 6 I/O settings can be controlled in 1-bit units.

The number of I/O ports differs for each product.

Generic Name	Number of I/O Ports
V850ES/HE3	–
V850ES/HF3	–
V850ES/HG3	–
V850ES/HJ3	16-bit I/O port

Table 4-15. Alternate-Function Pins of Port 6

Function Name	Alternate-Function Name		Pin No.				Remark	Function Name
	Name	I/O	HE3	HF3	HG3	HJ3		
P60	INTP11	Input	–	–	–	43	–	Ex1-UI
P61	INTP12	Input	–	–	–	44		Ex1-UI
P62	INTP13	Input	–	–	–	45		Ex1-UI
P63	–	–	–	–	–	46		C-U
P64	–	–	–	–	–	47		C-U
P65	–	–	–	–	–	48		C-U
P66	–	–	–	–	–	49		C-U
P67	–	–	–	–	–	50		C-U
P68	–	–	–	–	–	51		C-U
P69	–	–	–	–	–	52		C-U
P610	TIAB20/TOAB20	I/O	–	–	–	53		E10-U
P611	TIAB21/TOAB21	I/O	–	–	–	54		E10-U
P612	TIAB22/TOAB22	I/O	–	–	–	55		E10-U
P613	TIAB23/TOAB23	I/O	–	–	–	56		E10-U
P614	–	–	–	–	–	57		C-U
P615	–	–	–	–	–	58		C-U

Caution The P60 to P62 and P610 to P613 pins have hysteresis characteristics in the input mode of the alternate function, but do not have hysteresis characteristics in the port mode.

(1) Port register 6 (P6) (V850ES/HJ3 only)

After reset: 0000H (output latch) R/W Address: P6 FFFFF40CH, P6L FFFFF40CH, P6H FFFFF40DH

	15	14	13	12	11	10	9	8
P6 (P6H)	P615	P614	P613	P612	P611	P610	P69	P68
	7	6	5	4	3	2	1	0
(P6L)	P67	P66	P65	P64	P63	P63	P61	P60

P6n	Output data control (in output mode) (n = 0 to 15)
0	Outputs 0
1	Outputs 1

Remarks 1. The P6 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the P6 register as the P6H register and the lower 8 bits as the P6L register, P6 can be read or written in 8-bit or 1-bit units.

2. To read/write bits 8 to 15 of the P6 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the P6H register.

(2) Port mode register 6 (PM6) (V850ES/HJ3 only)

After reset: FFFFH R/W Address: PM6 FFFFF42CH, PM6L FFFFF42CH, PM6H FFFFF42DH

	15	14	13	12	11	10	9	8
PM6 (PM6H)	PM615	PM614	PM613	PM612	PM611	PM610	PM69	PM68
	7	6	5	4	3	2	1	0
(PM6L)	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60

PM6n	I/O mode control (n = 0 to 15)
0	Output mode
1	Input mode

Remarks 1. The PM6 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PM6 register as the PM6H register and the lower 8 bits as the PM6L register, PM6 can be read or written in 8-bit or 1-bit units.

2. To read/write bits 8 to 15 of the PM6 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PM6H register.

(3) Port mode control register 6 (PMC6) (V850ES/HJ3 only)

After reset: 0000H R/W Address: PMC6 FFFFF44CH, PMC6L FFFFF44CH, PMC6H FFFFF44DH

	15	14	13	12	11	10	9	8
PMC6 (PMC6H)	0	0	PMC613	PMC612	PMC611	PMC610	0	0

	7	6	5	4	3	2	1	0
(PMC6L)	0	0	0	0	0	PMC62	PMC61	PMC60

PMC613	Specification of P613 pin operation mode
0	I/O port
1	TIAB23 input/TOAB23 output

PMC612	Specification of P612 pin operation mode
0	I/O port
1	TIAB22 input/TOAB22 output

PMC611	Specification of P611 pin operation mode
0	I/O port
1	TIAB21 input/TOAB21 output

PMC610	Specification of P610 pin operation mode
0	I/O port
1	TIAB20 input/TOAB20 output

PMC62	Specification of P62 pin operation mode
0	I/O port
1	INTP13 input

PMC61	Specification of P61 pin operation mode
0	I/O port
1	INTP12 input

PMC60	Specification of P60 pin operation mode
0	I/O port
1	INTP11 input

Remarks 1. The PMC6 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PMC6 register as the PMC6H register and the lower 8 bits as the PMC6L register, PMC6 can be read or written in 8-bit or 1-bit units.

2. To read/write bits 8 to 15 of the PMC6 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PMC6H register.

(4) Port function control register 6 (PFC6) (V850ES/HJ3 only)

After reset: 0000H R/W Address: PFC6 FFFFF46CH,
PFC6L FFFFF46CH, PFC6H FFFFF46DH

	15	14	13	12	11	10	9	8
PFC6 (PFC6H)	0	0	PFC613	PFC612	PFC611	PFC610	0	0

	7	6	5	4	3	2	1	0
(PFC6L)	0	0	0	0	0	PFC62	PFC61	PFC60

PFC613	Specification of P613 pin alternate function
0	TIAB23 input
1	TOAB23 output

PFC612	Specification of P612 pin alternate function
0	TIAB22 input
1	TOAB22 output

PFC611	Specification of P611 pin alternate function
0	TIAB21 input
1	TOAB21 output

PFC610	Specification of P610 pin alternate function
0	TIAB20 input
1	TOAB20 output

PFC62	Specification of P62 pin alternate function
0	Setting prohibited
1	INTP13 input

PFC61	Specification of P61 pin alternate function
0	Setting prohibited
1	INTP12 input

PFC60	Specification of P60 pin alternate function
0	Setting prohibited
1	INTP11 input

Remarks 1. The PFC6 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PFC6 register as the PFC6H register and the lower 8 bits as the PFC6L register, PFC6 can be read or written in 8-bit or 1-bit units.

2. To read/write bits 8 to 15 of the PFC6 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PFC6H register.

(5) Pull-up resistor option register 6 (PU6) (V850ES/HJ3 only)

After reset: 0000H R/W Address: PU6 FFFFFC4CH,
PU6L FFFFFC4CH, PU6H FFFFFC4DH

	15	14	13	12	11	10	9	8
PU6 (PU6H)	PU615	PU614	PU613	PU612	PU611	PU610	PU69	PU68

	7	6	5	4	3	2	1	0
(PU6L)	PU67	PU66	PU65	PU64	PU63	PU62	PU61	PU60

PU6n	Control of on-chip pull-up resistor connection (n = 0 to 15)
0	Do not connect
1	Connect

Remarks 1. The PU6 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PU6 register as the PU6H register and the lower 8 bits as the PU6L register, PU6 can be read or written in 8-bit or 1-bit units.

2. To read/write bits 8 to 15 of the PU6 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PU6H register.

4.3.7 Port 7

Port 7 I/O settings can be controlled in 1-bit units.

The number of I/O ports differs for each product.

Generic Name	Number of I/O Ports
V850ES/HE3	10-bit I/O port
V850ES/HF3	12-bit I/O port
V850ES/HG3	16-bit I/O port
V850ES/HJ3	16-bit I/O port

Table 4-16. Alternate-Function Pins of Port 7

Function Name	Alternate-Function Name		Pin No.				Remark	Block Type
	Name	I/O	HE3	HF3	HG3	HJ3		
P70	ANI0	Input	64	80	100	144	–	D1A
P71	ANI1	Input	63	79	99	143		D1A
P72	ANI2	Input	62	78	98	142		D1A
P73	ANI3	Input	61	77	97	141		D1A
P74	ANI4	Input	60	76	96	140		D1A
P77	ANI5	Input	59	75	95	139		D1A
P76	ANI6	Input	58	74	94	138		D1A
P77	ANI7	Input	57	73	93	137		D1A
P78	ANI8	Input	56	72	92	136		D1A
P79	ANI9	Input	55	71	91	135		D1A
P710	ANI10	Input	–	70	90	134		D1A
P711	ANI11	Input	–	69	89	133		D1A
P712	ANI12	Input	–	–	88	132		D1A
P713	ANI13	Input	–	–	87	131		D1A
P714	ANI14	Input	–	–	86	130		D1A
P715	ANI15	Input	–	–	85	129		D1A

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

(a) V850ES/HE3

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
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

(b) V850ES/HF3

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	P711	P710	P79	P78
P7L	P77	P76	P75	P74	P73	P72	P71	P70

P7n	Output data control (in output mode) (n = 0 to 11)
0	Outputs 0
1	Outputs 1

(c) V850ES/HG3, V850ES/HJ3

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

	7	6	5	4	3	2	1	0
P7H	P715	P714	P713	P712	P711	P710	P79	P78
P7L	P77	P76	P75	P74	P73	P72	P71	P70

P7n	Output data control (in output mode) (n = 0 to 15)
0	Outputs 0
1	Outputs 1

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

(a) V850ES/HE3

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
PM7L	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

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

(b) V850ES/HF3

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

	7	6	5	4	3	2	1	0
PM7H	1	1	1	1	PM711	PM710	PM79	PM78
PM7L	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	I/O mode control (n = 0 to 11)
0	Output mode
1	Input mode

(c) V850ES/HG3, V850ES/HJ3

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

	7	6	5	4	3	2	1	0
PM7H	PM715	PM714	PM713	PM712	PM711	PM710	PM79	PM78
PM7L	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70

PM7n	I/O mode control (n = 0 to 15)
0	Output mode
1	Input mode

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.

(3) Port mode control register 7H, port mode control register 7L (PMC7H, PMC7L)

(a) V850ES/HE3

After reset: 00H R/W Address: PMC7L FFFFF44EH, PMC7H FFFFF44FH

	7	6	5	4	3	2	1	0
PMC7H	0	0	0	0	0	0	PMC79	PMC78

	7	6	5	4	3	2	1	0
PMC7L	PMC77	PMC76	PMC75	PMC74	PMC73	PMC72	PMC71	PMC70

PMC7n	Specification of P7n pin operation mode (n = 0 to 9)
0	I/O port
1	ANIn input

(b) V850ES/HF3

After reset: 00H R/W Address: PMC7L FFFFF44EH, PMC7H FFFFF44FH

	7	6	5	4	3	2	1	0
PMC7H	0	0	0	0	PMC711	PMC710	PMC79	PMC78

	7	6	5	4	3	2	1	0
PMC7L	PMC77	PMC76	PMC75	PMC74	PMC73	PMC72	PMC71	PMC70

PMC7n	Specification of P7n pin operation mode (n = 0 to 11)
0	I/O port
1	ANIn input

(c) V850ES/HG3, V850ES/HJ3

After reset: 00H R/W Address: PMC7L FFFFF44EH, PMC7H FFFFF44FH

	7	6	5	4	3	2	1	0
PMC7H	PMC715	PMC714	PMC713	PMC712	PMC711	PMC710	PMC79	PMC78

	7	6	5	4	3	2	1	0
PMC7L	PMC77	PMC76	PMC75	PMC74	PMC73	PMC72	PMC71	PMC70

PMC7n	Specification of P7n pin operation mode (n = 0 to 15)
0	I/O port
1	ANIn input

Remark These registers cannot be accessed in 16-bit units as the PMC7 register. They can be read or written in 8-bit or 1-bit units as the PMC7H and PMC7L registers.

4.3.8 Port 8 (V850ES/HJ3 only)

Port 8 I/O settings can be controlled in 1-bit units.

The number of I/O ports differs for each product.

Generic Name	Number of I/O Ports
V850ES/HE3	–
V850ES/HF3	–
V850ES/HG3	–
V850ES/HJ3	2-bit I/O port[Note]

Note In the μPD70F3755, the alternate functions of the P80 and P81 pins (RXDD3 and TXDD3) are not available.

Table 4-17. Alternate-Function Pins of Port 8

Function Name	Alternate-Function Name		Pin No.				Remark	Block Type
	Name	I/O	HE3	HF3	HG3	HJ3		
P80	INTP14	Input	–	–	–	59[Note 1]	–	D1-UI
	RXDD3/INTP14	Input	–	–	–	59[Note 2]		D3-UI
P81	–	–	–	–	–	60[Note 1]		C-U
	TXDD3	Output	–	–	–	60[Note 2]		D0-U

Notes 1. μPD70F3755 only

2. μPD70F3757 only

Caution The P80 and P81 pins have hysteresis characteristics in the input mode of the alternate function, but do not have hysteresis characteristics in the port mode.

(1) Port register 8 (P8) (V850ES/HJ3 only)

After reset: 00H (output latch) R/W Address: FFFFF410H

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

P8n	Output data control (in output mode) (n = 0, 1)
0	Outputs 0
1	Outputs 1

(2) Port mode register 8 (PM8) (V850ES/HJ3 only)

After reset: FFH R/W Address: FFFFF430H

	7	6	5	4	3	2	1	0
PM8	1	1	1	1	1	1	PM81	PM80

PM8n	I/O mode control (n = 0, 1)
0	Output mode
1	Input mode

(3) Port mode control register 8 (PMC8) (V850ES/HJ3 only)

After reset: 00H R/W Address: FFFFF450H

	7	6	5	4	3	2	1	0
PMC8	0	0	0	0	0	0	PMC81	PMC80

PMC81	Specification of P81 pin operation mode
0	I/O port
1	TXDD3 output[Note 1]

PMC80	Specification of P80 pin operation mode
0	I/O port
1	RXDD3 input[Note 1]/INTP14 input[Note 2]

Notes 1. μPD70F3757 only

2. The INTP14 function and RXDD3 function are alternately used. When using as the RXDD3 function, disable edge detection for the INTP14 function (set the INTF8.INTF80 bit and the INTR8.INTR80 bit to 0). When using as the INTP14 function, stop UARTD3 reception (set the UD3CTL0.UD3RXE bit to 0).

(4) Pull-up resistor option register 8 (PU8) (V850ES/HJ3 only)

After reset: 00H R/W Address: FFFFFC50H

	7	6	5	4	3	2	1	0
PU8	0	0	0	0	0	0	PU81	PU80

PU8n	Control of on-chip pull-up resistor connection (n = 0, 1)
0	Do not connect
1	Connect

4.3.9 Port 9

Port 9 which I/O settings can be controlled in 1-bit units.

The number of I/O ports differs for each product.

Generic Name	Number of I/O Ports
V850ES/HE3	9-bit I/O port
V850ES/HF3	9-bit I/O port
V850ES/HG3	16-bit I/O port
V850ES/HJ3	16-bit I/O port

Table 4-18. Alternate-Function Pins of Port 9

Function Name	Alternate-Function Name		Pin No.				Remark	Block Type
	Name	I/O	HE3	HF3	HG3	HJ3		
P90	KR6/TXDD1	I/O	36	38	43	61	–	Fx10x-U
P91	KR7/RXDD1	Input	37	39	44	62		Fx13x-U
P92	TIAB11/TOAB11	I/O	–	–	45	63		Fx10x-U
P93	TIAB12/TOAB12	I/O	–	–	46	64		Fx10x-U
P94	TIAB13/TOAB13	I/O	–	–	47	65		Fx10x-U
P95	TIAB10/TOAB10	I/O	–	–	48	66		Fx10x-U
P96	TIAA21/TOAA21	I/O	38	40	49	67		Fxx10-U
P97	SIB1/TIAA20/TOAA20	I/O	39	41	50	68		Fx110-U
P98	SOB1/TIAB03/TOAB03	I/O	40	42	51	69		Fx010-U
P99	$\overline{\text{SCKB1}}$/TIAB00/TOAB00	I/O	41	43	52	70		Fx210-U
P910	–	–	–	–	53	–		C-U
	SIB2	Input	–	–	–	71		Ex1-U
P911	–	–	–	–	54	–		C-U
	SOB2	Output	–	–		72		Ex0-U
P912	–	–	–	–	55	–		C-U
	$\overline{\text{SCKB2}}$	I/O	–	–	–	73[Note 1]		Ex2-U
	$\overline{\text{SCKB2}}$/TXDD5	I/O	–	–	–	73[Note 2]		Fx2x0-U
P913	INTP4/PCL	I/O	42	44	56	74[Note 1]		Fx10x-UI
	INTP4/PCL/RXDD5	I/O	–	–	–	74[Note 2]		Fx103-UI
P914	INTP5/SDA00	I/O	43	45	57	75[Note 1]		Fx12x-UFI
	INTP5/SDA00/RXDD4	I/O	–	–	–	75[Note 2]		Fx123-UFI
P915	INTP6/SCL00	I/O	44	46	58	76[Note 1]		Fx12x-UFI
	INTP6/SCL00/TXDD4	I/O	–	–	–	76[Note 2]		Fx120-UFI

Notes 1. μPD70F3755 only

2. μPD70F3757 only

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

(1) Port register 9 (P9)

(a) V850ES/HE3, V850ES/HF3

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	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, 13 to 15)
0	Outputs 0
1	Outputs 1

(b) V850ES/HG3, V850ES/HJ3

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	P95	P94	P93	P92	P91	P90

P9n	Output data control (in output mode) (n = 0 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 mode register 9 (PM9)

(a) V850ES/HE3, V850ES/HF3

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	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, 13 to 15)
0	Output mode
1	Input mode

(b) V850ES/HG3, V850ES/HJ3

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	PM95	PM94	PM93	PM92	PM91	PM90

PM9n	I/O mode control (n = 0 to 15)
0	Output mode
1	Input mode

Remarks 1. 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 or 1-bit units.

2. 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 mode control register 9 (PMC9)

(a) V850ES/HE3, V850ES/HF3

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

PMC9 (PMC9H)

15	14	13	12	11	10	9	8
PMC915	PMC914	PMC913	0	0	0	PMC99	PMC98

(PMC9L)

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

PMC915	Specification of P915 pin operation mode
0	I/O port
1	INTP6 input/SCL00 I/O

PMC914	Specification of P914 pin operation mode
0	I/O port
1	INTP5 input/SDA00 I/O

PMC913	Specification of P913 pin operation mode
0	I/O port
1	INTP4 input/PCL output

PMC99	Specification of P99 pin operation mode
0	I/O port
1	$\overline{\text{SCKB1}}$ I/O/TIAB00 input/TOAB00 output

PMC98	Specification of P98 pin operation mode
0	I/O port
1	SOB1 output/TIAB03 input/TOAB03 output

PMC97	Specification of P97 pin operation mode
0	I/O port
1	SIB1 input/TIAA20 input/TOAA20 output

PMC96	Specification of P96 pin operation mode
0	I/O port
1	TIAA21 input/TOAA21 output

PMC91	Specification of P91 pin operation mode
0	I/O port
1	KR7 input/RXDD1 input

PMC90	Specification of P90 pin operation mode
0	I/O port
1	KR6 input/TXDD1 output

(2/5)

(b) V850ES/HG3

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	0	0	0	PMC99	PMC98

	7	6	5	4	3	2	1	0
(PMC9L)	PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

PMC915	Specification of P915 pin operation mode
0	I/O port
1	INTP6 input/SCL00 I/O

PMC914	Specification of P914 pin operation mode
0	I/O port
1	INTP5 input/SDA00 I/O

PMC913	Specification of P913 pin operation mode
0	I/O port
1	INTP4 input/PCL output

PMC99	Specification of P99 pin operation mode
0	I/O port
1	$\overline{\text{SCKB1}}$ I/O/TIAB00 input/TOAB00 output

PMC98	Specification of P98 pin operation mode
0	I/O port
1	SOB1 output/TIAB03 input/TOAB03 output

PMC97	Specification of P97 pin operation mode
0	I/O port
1	SIB1 input/TIAA20 input/TOAA20 output

PMC96	Specification of P96 pin operation mode
0	I/O port
1	TIAA21 input/TOAA21 output

PMC95	Specification of P95 pin operation mode
0	I/O port
1	TIAB10 input/TOAB10 output

(3/5)

PMC94	Specification of P94 pin operation mode
0	I/O port
1	TIAB13 input/TOAB13 output

PMC93	Specification of P93 pin operation mode
0	I/O port
1	TIAB12 input/TOAB12 output

PMC92	Specification of P92 pin operation mode
0	I/O port
1	TIAB11 input/TOAB11 output

PMC91	Specification of P91 pin operation mode
0	I/O port
1	KR7 input/RXDD1 input

PMC90	Specification of P90 pin operation mode
0	I/O port
1	KR6 input/TXDD1 output

(4/5)

(c) V850ES/HJ3 (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	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

PMC915	Specification of P915 pin operation mode
0	I/O port
1	INTP6 input/SCL00 I/O/TXDD4 output[Note]

PMC914	Specification of P914 pin operation mode
0	I/O port
1	INTP5 input/SDA00 I/O/RXDD4 input[Note]

PMC913	Specification of P913 pin operation mode
0	I/O port
1	INTP4 input/PCL output/RXDD5 input[Note]

PMC912	Specification of P912 pin operation mode
0	I/O port
1	$\overline{\text{SCKB2}}$ I/O/TXDD5 output[Note]

PMC911	Specification of P911 pin operation mode
0	I/O port
1	SOB2 output

PMC910	Specification of P910 pin operation mode
0	I/O port
1	SIB2 input

PMC99	Specification of P99 pin operation mode
0	I/O port
1	$\overline{\text{SCKB1}}$ I/O/TIAB00 input/TOAB00 output

Note μPD70F3757 only

(5/5)

(c) V850ES/HJ3 (2/2)

PMC98	Specification of P98 pin operation mode
0	I/O port
1	SOB1 output/TIAB03 input/TOAB03 output

PMC97	Specification of P97 pin operation mode
0	I/O port
1	SIB1 input/TIAA20 input/TOAA20 output

PMC96	Specification of P96 pin operation mode
0	I/O port
1	TIAA21 input/TOAA21 output

PMC95	Specification of P95 pin operation mode
0	I/O port
1	TIAB10 input/TOAB10 output

PMC94	Specification of P94 pin operation mode
0	I/O port
1	TIAB13 input/TOAB13 output

PMC93	Specification of P93 pin operation mode
0	I/O port
1	TIAB12 input/TOAB12 output

PMC92	Specification of P92 pin operation mode
0	I/O port
1	TIAB11 input/TOAB11 output

PMC91	Specification of P91 pin operation mode
0	I/O port
1	KR7 input/RXDD1 input

PMC90	Specification of P90 pin operation mode
0	I/O port
1	KR6 input/TXDD1 output

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 function control register 9 (PFC9)

(a) V850ES/HE3, V850ES/HF3

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	0	0	0	PFC99	PFC98

	7	6	5	4	3	2	1	0
(PFC9L)	PFC97	PFC96	0	0	0	0	PFC91	PFC90

(b) V850ES/HG3

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	0	0	0	PFC99	PFC98

	7	6	5	4	3	2	1	0
(PFC9L)	PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90

(c) V850ES/HJ3

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	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90

Remarks 1. For the specifications of alternate functions, see **4.3.9 (6) Settings of alternate functions of port 9**.

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 function control expansion register 9 (PFCE9)

(a) V850ES/HE3, V850ES/HF3

After reset: 0000H R/W Address: PFCE9 FFFFF712H,
PFCE9L FFFFF712H, PFCE9H FFFFF713H

	15	14	13	12	11	10	9	8
PFCE9 (PFCE9H)	PFCE915	PFCE914	PFCE913	0	0	0	PFCE99	PFCE98

	7	6	5	4	3	2	1	0
(PFCE9L)	PFCE97	PFCE96	0	0	0	0	PFCE91	PFCE90

(b) V850ES/HG3

After reset: 0000H R/W Address: PFCE9 FFFFF712H,
PFCE9L FFFFF712H, PFCE9H FFFFF713H

	15	14	13	12	11	10	9	8
PFCE9 (PFCE9H)	PFCE915	PFCE914	PFCE913	0	0	0	PFCE99	PFCE98

	7	6	5	4	3	2	1	0
(PFCE9L)	PFCE97	PFCE96	PFCE95	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90

(c) V850ES/HJ3

After reset: 0000H R/W Address: PFCE9 FFFFF712H,
PFCE9L FFFFF712H, PFCE9H FFFFF713H

	15	14	13	12	11	10	9	8
PFCE9 (PFCE9H)	PFCE915	PFCE914	PFCE913	PFCE912Note	0	0	PFCE99	PFCE98

	7	6	5	4	3	2	1	0
(PFCE9L)	PFCE97	PFCE96	PFCE95	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90

Note μPD70F3757 only

Remarks 1. For the specifications of alternate functions, see **4.3.9 (6) Settings of alternate functions of port 9**.

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.

(6) Settings of alternate functions of port 9

(a) V850ES/HE3, V850ES/HF3

PFCE915	PFC915	Specification of P915 Pin Alternate Function
0	0	Setting prohibited
0	1	INTP6 input
1	0	SCL00 I/O
1	1	Setting prohibited

PFCE914	PFC914	Specification of P914 Pin Alternate Function
0	0	Setting prohibited
0	1	INTP5 input
1	0	SDA00 I/O
1	1	Setting prohibited

PFCE913	PFC913	Specification of P913 Pin Alternate Function
0	0	Setting prohibited
0	1	INTP4 input
1	0	PCL output
1	1	Setting prohibited

PFCE99	PFC99	Specification of P99 Pin Alternate Function
0	0	Setting prohibited
0	1	$\overline{\text{SCKB1}}$ I/O
1	0	TIAB00 input
1	1	TOAB00 output

PFCE98	PFC98	Specification of P98 Pin Alternate Function
0	0	Setting prohibited
0	1	SOB1 output
1	0	TIAB03 input
1	1	TOAB03 output

PFCE97	PFC97	Specification of P97 Pin Alternate Function
0	0	Setting prohibited
0	1	SIB1 input
1	0	TIAA20 input
1	1	TOAA20 output

PFCE96	PFC96	Specification of P96 Pin Alternate Function
0	0	Setting prohibited
0	1	Setting prohibited
1	0	TIAA21 input
1	1	TOAA21 output

PFCE91	PFC91	Specification of P91 Pin Alternate Function
0	0	Setting prohibited
0	1	KR7 input
1	0	RXDD1 input/KR7 input[Note]
1	1	Setting prohibited

PFCE90	PFC90	Specification of P90 Pin Alternate Function
0	0	Setting prohibited
0	1	KR6 input
1	0	TXDD1 output
1	1	Setting prohibited

Note The KR7 function and RXDD1 function are alternately used.
When using as the RXDD1 function, disable KR7 function key return detection (set the KRM7 bit of the KRM register to 0). Also, when using as the KR7 function, it is recommended to set the PFC91 bit to 1 and set the PFCE91 bit to 0.

(b) V850ES/HG3

PFCE915	PFC915	Specification of P915 Pin Alternate Function
0	0	Setting prohibited
0	1	INTP6 input
1	0	SCL00 I/O
1	1	Setting prohibited

PFCE914	PFC914	Specification of P914 Pin Alternate Function
0	0	Setting prohibited
0	1	INTP5 input
1	0	SDA00 I/O
1	1	Setting prohibited

PFCE913	PFC913	Specification of P913 Pin Alternate Function
0	0	Setting prohibited
0	1	INTP4 input
1	0	PCL output
1	1	Setting prohibited

PFCE99	PFC99	Specification of P99 Pin Alternate Function
0	0	Setting prohibited
0	1	$\overline{\text{SCKB1}}$ I/O
1	0	TIAB00 input
1	1	TOAB00 output

PFCE98	PFC98	Specification of P98 Pin Alternate Function
0	0	Setting prohibited
0	1	SOB1 output
1	0	TIAB03 input
1	1	TOAB03 output

PFCE97	PFC97	Specification of P97 Pin Alternate Function
0	0	Setting prohibited
0	1	SIB1 input
1	0	TIAA20 input
1	1	TOAA20 output

PFCE96	PFC96	Specification of P96 Pin Alternate Function
0	0	Setting prohibited
0	1	Setting prohibited
1	0	TIAA21 input
1	1	TOAA21 output

PFCE95	PFC95	Specification of P95 Pin Alternate Function
0	0	Setting prohibited
0	1	TIAB10 input
1	0	TOAB10 output
1	1	Setting prohibited

PFCE94	PFC94	Specification of P94 Pin Alternate Function
0	0	Setting prohibited
0	1	TIAB13 input
1	0	TOAB13 output
1	1	Setting prohibited

PFCE93	PFC93	Specification of P93 Pin Alternate Function
0	0	Setting prohibited
0	1	TIAB12 input
1	0	TOAB12 output
1	1	Setting prohibited

PFCE92	PFC92	Specification of P92 Pin Alternate Function
0	0	Setting prohibited
0	1	TIAB11 input
1	0	TOAB11 output
1	1	Setting prohibited

PFCE91	PFC91	Specification of P91 Pin Alternate Function
0	0	Setting prohibited
0	1	KR7 input
1	0	RXDD1 input/KR7 input[Note]
1	1	Setting prohibited

PFCE90	PFC90	Specification of P90 Pin Alternate Function
0	0	Setting prohibited
0	1	KR6 input
1	0	TXDD1 output
1	1	Setting prohibited

Note The KR7 function and RXDD1 function are alternately used.
When using as the RXDD1 function, disable KR7 function key return detection (set the KRM7 bit of the KRM register to 0). Also, when using as the KR7 function, it is recommended to set the PFC91 bit to 1 and set the PFCE91 bit to 0.

(c) V850ES/HJ3

PFCE915	PFC915	Specification of P915 Pin Alternate Function
0	0	Setting prohibited
0	1	INTP6 input
1	0	SCL00 I/O
1	1	TXDD4 output[Note 1]

PFCE914	PFC914	Specification of P914 Pin Alternate Function
0	0	Setting prohibited
0	1	INTP5 input
1	0	SDA00 I/O
1	1	RXDD4 input[Note 1]/INTP5 input[Note 2]

PFCE913	PFC913	Specification of P913 Pin Alternate Function
0	0	Setting prohibited
0	1	INTP4 input
1	0	PCL output
1	1	RXDD5 input[Note 1]/INTP4 input[Note 3]

PFCE912[Note 1]	PFC912	Specification of P912 Pin Alternate Function
0	0	Setting prohibited
0	1	$\overline{\text{SCKB2}}$ I/O
1	0	Setting prohibited
1	1	TXDD5 output[Note 1]

PFC911	Specification of P911 Pin Alternate Function
0	Setting prohibited
1	SOB2 output

PFC910	Specification of P910 Pin Alternate Function
0	Setting prohibited
1	SIB2 input

Notes 1. μPD70F3757 only

2. In the μPD70F3757, the INTP5 function and RXDD4 function are alternately used.
When using as the RXDD4 function, set the valid edge specification of the INTP5 function to “No edge detected” (set the INTF9H.INTF914 bit and the INTR9H.INTR914 bit to “00”). Also, when using as the INTP5 function, it is recommended to set the PFC914 bit to 1 and set the PFCE914 bit to 0.

3. In the μPD70F3757, the INTP4 function and RXDD5 function are alternately used.
When using as the RXDD5 function, set the valid edge specification of the INTP4 function to “No edge detected” (set the INTF9H.INTF913 bit and the INTR9H.INTR913 bit to “00”). Also, when using as the INTP4 function, it is recommended to set the PFC913 bit to 1 and set the PFCE913 bit to 0.

PFCE99	PFC99	Specification of P99 Pin Alternate Function
0	0	Setting prohibited
0	1	$\overline{\text{SCKB1}}$ I/O
1	0	TIAB00 input
1	1	TOAB00 output

PFCE98	PFC98	Specification of P98 Pin Alternate Function
0	0	Setting prohibited
0	1	SOB1 output
1	0	TIAB03 input
1	1	TOAB03 output

PFCE97	PFC97	Specification of P97 Pin Alternate Function
0	0	Setting prohibited
0	1	SIB1 input
1	0	TIAA20 input
1	1	TOAA20 output

PFCE96	PFC96	Specification of P96 Pin Alternate Function
0	0	Setting prohibited
0	1	Setting prohibited
1	0	TIAA21 input
1	1	TOAA21 output

PFCE95	PFC95	Specification of P95 Pin Alternate Function
0	0	Setting prohibited
0	1	TIAB10 input
1	0	TOAB10 output
1	1	Setting prohibited

PFCE94	PFC94	Specification of P94 Pin Alternate Function
0	0	Setting prohibited
0	1	TIAB13 input
1	0	TOAB13 output
1	1	Setting prohibited

PFCE93	PFC93	Specification of P93 Pin Alternate Function
0	0	Setting prohibited
0	1	TIAB12 input
1	0	TOAB12 output
1	1	Setting prohibited

PFCE92	PFC92	Specification of P92 Pin Alternate Function
0	0	Setting prohibited
0	1	TIAB11 input
1	0	TOAB11 output
1	1	Setting prohibited

PFCE91	PFC91	Specification of P91 Pin Alternate Function
0	0	Setting prohibited
0	1	KR7 input
1	0	RXDD1 input/KR7 input[Note]
1	1	Setting prohibited

PFCE90	PFC90	Specification of P90 Pin Alternate Function
0	0	Setting prohibited
0	1	KR6 input
1	0	TXDD1 output
1	1	Setting prohibited

Note The KR7 function and RXDD1 function are alternately used.
When using as the RXDD1 function, disable KR7 function key return detection (set the KRM7 bit of the KRM register to 0). Also, when using as the KR7 function, it is recommended to set the PFC91 bit to 1 and set the PFCE91 bit to 0.

(7) Pull-up resistor option register 9 (PU9)

(a) V850ES/HE3, V850ES/HF3

After reset: 0000H R/W Address: PU9 FFFFFC52H, PU9L FFFFFC52H, PU9H FFFFFC53H

	15	14	13	12	11	10	9	8
PU9 (PU9H)	PU915	PU914	PU913	0	0	0	PU99	PU98

	7	6	5	4	3	2	1	0
(PU9L)	PU97	PU96	0	0	0	0	PU91	PU90

PU9n	Control of on-chip pull-up resistor connection (n = 0, 1, 6 to 9, 13 to 15)
0	Do not connect
1	Connect

(b) V850ES/HG3, V850ES/HJ3

After reset: 0000H R/W Address: PU9 FFFFFC52H, PU9L FFFFFC52H, PU9H FFFFFC53H

	15	14	13	12	11	10	9	8
PU9 (PU9H)	PU915	PU914	PU913	PU912	PU911	PU910	PU99	PU98

	7	6	5	4	3	2	1	0
(PU9L)	PU97	PU96	PU95	PU94	PU93	PU92	PU91	PU90

PU9n	Control of on-chip pull-up resistor connection (n = 0 to 15)
0	Do not connect
1	Connect

Remarks 1. The PU9 register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PU9 register as the PU9H register and the lower 8 bits as the PU9L register, PU9 can be read or written in 8-bit or 1-bit units.

2. To read/write bits 8 to 15 of the PU9 register in 8-bit or 1-bit units, specify them as bits 0 to 7 of the PU9H register.

(8) Port 9 function register H (PF9H)

After reset: 00H R/W Address: FFFFFC73H

	7	6	5	4	3	2	1	0
PF9H	PF915	PF914	0	0	0	0	0	0

PF9n	Control of normal output/N-ch open-drain output (n = 15, 14)
0	Normal output
1	N-ch open-drain output

Caution When using P915 and P914 as N-ch open-drain-output alternate-function pins, set in the following sequence.

Be sure to set the port latch to 1 before setting the pin to N-ch open-drain output.

P9n bit = 1 → PFC9n bit = 0/1 → PF9n bit = 1 → PMC9n bit = 1

4.3.10 Port 12 (V850ES/HJ3 only)

Port 12 I/O settings can be controlled in 1-bit units.

The number of I/O ports differs for each product.

Generic Name	Number of I/O Ports
V850ES/HE3	–
V850ES/HF3	–
V850ES/HG3	–
V850ES/HJ3	8-bit I/O port

Table 4-19. Alternate-Function Pins of Port 12

Function Name	Alternate-Function Name		Pin No.				Remark	Block Type
	Name	I/O	HE3	HF3	HG3	HJ3		
P120	ANI16	Input	–	–	–	128	–	D1A
P121	ANI17	Input	–	–	–	127		D1A
P122	ANI18	Input	–	–	–	126		D1A
P123	ANI19	Input	–	–	–	125		D1A
P124	ANI20	Input	–	–	–	124		D1A
P125	ANI21	Input	–	–	–	123		D1A
P126	ANI22	Input	–	–	–	122		D1A
P127	ANI23	Input	–	–	–	121		D1A

(1) Port register 12 (P12) (V850ES/HJ3 only)

After reset: 00H (output latch) R/W Address: FFFFF418H

	7	6	5	4	3	2	1	0
P12	P127	P126	P125	P124	P123	P122	P121	P120

P12n	Output data control (in output mode) (n = 0 to 7)
0	Outputs 0
1	Outputs 1

(2) Port mode register 12 (PM12) (V850ES/HJ3 only)

After reset: FFH R/W Address: FFFFF438H

	7	6	5	4	3	2	1	0
PM12	PM127	PM126	PM125	PM124	PM123	PM122	PM121	PM120

PM12n	I/O mode control (n = 0 to 7)
0	Output mode
1	Input mode

(3) Port mode control register 12 (PMC12) (V850ES/HJ3 only)

After reset: 00H R/W Address: FFFFF458H

	7	6	5	4	3	2	1	0
PMC12	PMC127	PMC126	PMC125	PMC124	PMC123	PMC122	PMC121	PMC120

PMC12n	Specification of P12n pin operation mode (n = 0 to 7, m = 16 to 23)
0	I/O port
1	ANIm input

4.3.11 Port CD (V850ES/HJ3 only)

Port CD I/O settings can be controlled in 1-bit units.

The number of I/O ports differs for each product.

Generic Name	Number of I/O Ports
V850ES/HE3	–
V850ES/HF3	–
V850ES/HG3	–
V850ES/HJ3	4-bit I/O port

Table 4-20. Alternate-Function Pins of Port CD

Function Name	Alternate-Function Name		Pin No.				Remark	Block Type
	Name	I/O	HE3	HF3	HG3	HJ3		
PCD0	–	–	–	–	–	77	–	C
PCD1	–	–	–	–	–	78		C
PCD2	–	–	–	–	–	79		C
PCD3	–	–	–	–	–	80		C

(1) Port register CD (PCD) (V850ES/HJ3 only)

After reset: 00H (output latch) R/W Address: FFFFF00EH

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

PCDn	Output data control (in output mode) (n = 0 to 3)
0	Outputs 0
1	Outputs 1

(2) Port mode register CD (PMCD) (V850ES/HJ3 only)

After reset: FFH R/W Address: FFFFF02EH

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

PMCDn	I/O mode control (n = 0 to 3)
0	Output mode
1	Input mode

4.3.12 Port CM

Port CM I/O settings can be controlled in 1-bit units.

The number of I/O ports differs for each product.

Generic Name	Number of I/O Ports
V850ES/HE3	2-bit I/O port
V850ES/HF3	4-bit I/O port
V850ES/HG3	4-bit I/O port
V850ES/HJ3	6-bit I/O port

Table 4-21. Alternate-Function Pins of Port CM

Function Name	Alternate-Function Name		Pin No.				Remark	Block Type
	Name	I/O	HE3	HF3	HG3	HJ3		
PCM0	–	–	45	49	61	–	–	C
	$\overline{\text{WAIT}}$	Input	–	–	–	85		D1
PCM1	CLKOUT	Output	46	50	62	86		D0
PCM2	–	–	–	51	63	–		C
	$\overline{\text{HLDAK}}$	Output	–	–	–	87		D1
PCM3	–	–	–	52	64	–		C
	$\overline{\text{HLDRQ}}$	Input	–	–	–	88		D0
PCM4	–	–	–	–	–	89		C
PCM5	–	–	–	–	–	90		C

(1) Port register CM (PCM)

(a) V850ES/HE3

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

	7	6	5	4	3	2	1	0
PCM	0	0	0	0	0	0	PCM1	PCM0

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

(b) V850ES/HF3, V850ES/HG3

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

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

PCMn	Output data control (in output mode) (n = 0 to 3)
0	Outputs 0
1	Outputs 1

(c) V850ES/HJ3

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

	7	6	5	4	3	2	1	0
PCM	0	0	PCM5	PCM4	PCM3	PCM2	PCM1	PCM0

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

(2) Port mode register CM (PMCM)

(a) V850ES/HE3

After reset: FFH R/W Address: FFFFF02CH

	7	6	5	4	3	2	1	0
PMCM	1	1	1	1	1	1	PMCM1	PMCM0

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

(b) V850ES/HF3, V850ES/HG3

After reset: FFH R/W Address: FFFFF02CH

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

PMCMn	I/O mode control (n = 0 to 3)
0	Output mode
1	Input mode

(c) V850ES/HJ3

After reset: FFH R/W Address: FFFFF02CH

	7	6	5	4	3	2	1	0
PMCM	1	1	PMCM5	PMCM4	PMCM3	PMCM2	PMCM1	PMCM0

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

(3) Port mode control register CM (PMCCM)

(a) V850ES/HE3, V850ES/HF3, V850ES/HG3

After reset: 00H R/W Address: FFFFF04CH

	7	6	5	4	3	2	1	0
PMCCM	0	0	0	0	0	0	PMCCM1	0

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

(b) V850ES/HJ3

After reset: 00H R/W Address: FFFFF04CH

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

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

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

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

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

4.3.13 Port CS

Port CSI/O settings can be controlled in 1-bit units.

The number of I/O ports differs for each product.

Generic Name	Number of I/O Ports
V850ES/HE3	–
V850ES/HF3	2-bit I/O port
V850ES/HG3	2-bit I/O port
V850ES/HJ3	8-bit I/O port

Table 4-22. Alternate-Function Pins of Port CS

Function Name	Alternate-Function Name		Pin No.				Remark	Block Type
	Name	I/O	HE3	HF3	HG3	HJ3		
PCS0	–		–	47	59	–	–	C
	$\overline{CS0}$	Output	–	–	–	81		D0
PCS1	–		–	48	60	–		C
	$\overline{CS1}$	Output	–	–	–	82		D0
PCS2	$\overline{CS2}$	Output	–	–	–	83		D0
PCS3	$\overline{CS3}$	Output	–	–	–	84		D0
PCS4	–	–	–	–	–	91		C
PCS5	–	–	–	–	–	92		C
PCS6	–	–	–	–	–	93		C
PCS7	–	–	–	–	–	94		C

(1) Port register CS (PCS)

(a) V850ES/HF3, V850ES/HG3

After reset: 00H (output latch) R/W Address: FFFFF008H

	7	6	5	4	3	2	1	0
PCS	0	0	0	0	0	0	PCS1	PCS0

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

(b) V850ES/HJ3

After reset: 00H (output latch) R/W Address: FFFFF008H

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

PCSn	Output data control (in output mode) (n = 0 to 7)
0	Outputs 0
1	Outputs 1

(2) Port mode register CS (PMCS)

(a) V850ES/HF3, V850ES/HG3

After reset: FFH R/W Address: FFFFF028H

	7	6	5	4	3	2	1	0
PMCS	0	0	0	0	0	0	PMCS1	PMCS0

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

(b) V850ES/HJ3

After reset: FFH R/W Address: FFFFF028H

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

PMCSn	I/O mode control (n = 0 to 7)
0	Output mode
1	Input mode

(3) Port mode control register CS (PMCCS) (V850ES/HJ3 only)

After reset: 00H R/W Address: FFFFF048H

	7	6	5	4	3	2	1	0
PMCCS	0	0	0	0	PMCCS3	PMCCS2	PMCCS1	PMCCS0

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

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

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

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

4.3.14 Port CT

Port CT I/O settings can be controlled in 1-bit units.

The number of I/O ports differs for each product.

Generic Name	Number of I/O Ports
V850ES/HE3	–
V850ES/HF3	4-bit I/O port
V850ES/HG3	4-bit I/O port
V850ES/HJ3	8-bit I/O port

Table 4-23. Alternate-Function Pins of Port CT

Function Name	Alternate-Function Name		Pin No.				Remark	Block Type
	Name	I/O	HE3	HF3	HG3	HJ3		
PCT0	–	–	–	53	65	–	–	C
	$\overline{WR0}$	Output	–	–	–	81		D0
PCT1	–	–	–	54	66	–		C
	$\overline{WR1}$	Output	–	–	–	82		D0
PCT2	–	–	–	–	–	83		C
PCT3	–	–	–	–	–	84		C
PCT4	–	–	–	55	67	–		C
	$\overline{RD}$	Output	–	–	–	91		D0
PCT5	–	–	–	–	–	92		C
PCT6	–	–	–	56	68	–		C
	$\overline{ASTB}$	Output	–	–	–	93		D0
PCT7	–	–	–	–	–	94		C

(1) Port register CT (PCT)

(a) V850ES/HF3, V850ES/HG3

After reset: 00H (output latch) R/W Address: FFFFF00AH

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

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

(b) V850ES/HJ3

After reset: 00H (output latch) R/W Address: FFFFF00AH

	7	6	5	4	3	2	1	0
PCT	PCT7	PCT6	PCT5	PCT4	PCT3	PCT2	PCT1	PCT0

PCTn	Output data control (in output mode) (n = 0 to 7)
0	Outputs 0
1	Outputs 1

(2) Port mode register CT (PMCT)

(a) V850ES/HF3, V850ES/HG3

After reset: FFH R/W Address: FFFFF02AH

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

PMCTn	I/O mode control (n = 0, 1, 4, 6)
0	Output mode
1	Input mode

(b) V850ES/HJ3

After reset: FFH R/W Address: FFFFF02AH

	7	6	5	4	3	2	1	0
PMCT	PMCT7	PMCT6	PMCT5	PMCT4	PMCT3	PMCT2	PMCT1	PMCT0

PMCTn	I/O mode control (n = 0 to 7)
0	Output mode
1	Input mode

(3) Port mode control register CT (PMCCT) (V850ES/HJ3 only)

After reset: 00H R/W Address: FFFFF04AH

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

PMCCT6	Specification of PCS6 pin operation mode
0	I/O port
1	$\overline{\text{ASTB}}$ output

PMCCT4	Specification of PCS4 pin operation mode
0	I/O port
1	$\overline{\text{RD}}$ output

PMCCT1	Specification of PCS1 pin operation mode
0	I/O port
1	$\overline{\text{WR1}}$ output

PMCCT0	Specification of PCS0 pin operation mode
0	I/O port
1	$\overline{\text{WR0}}$ output

4.3.15 Port DL

Port DL I/O settings can be controlled in 1-bit units.

The number of I/O ports differs for each product.

Generic Name	Number of I/O Ports
V850ES/HE3	8-bit I/O port
V850ES/HF3	12-bit I/O port
V850ES/HG3	14-bit I/O port
V850ES/HJ3	16-bit I/O port

Table 4-24. Alternate-Function Pins of Port DL

Function Name	Alternate-Function Name		Pin No.				Remark	Block Type
	Name	I/O	HE	HF	HG	HJ		
PDL0	–	–	47	57	71	–	–	C
	AD0	I/O	–	–	–	105		D2
PDL1	–	–	48	58	72	–		C
	AD1	I/O	–	–	–	106		D2
PDL2	–	–	49	59	73	–		C
	AD2	I/O	–	–	–	107		D2
PDL3	–	–	50	60	74	–		C
	AD3	I/O	–	–	–	108		D2
PDL4	–	–	51	61	75	–		C
	AD4	I/O	–	–	–	109		D2
PDL5	FLMD1[Note]	–	52	62	76	–		C
	AD5/FLMD1[Note]	I/O	–	–	–	110		D2
PDL6	–	–	53	63	77	–		C
	AD6	I/O	–	–	–	111		D2
PDL7	–	–	54	64	78	–		C
	AD7	I/O	–	–	–	112		D2
PDL8	–	–	–	65	79	–		C
	AD8	I/O	–	–	–	113		D2
PDL9	–	–	–	66	80	–		C
	AD9	I/O	–	–	–	114		D2
PDL10	–	–	–	67	81	–		C
	AD10	I/O	–	–	–	115		D2
PDL11	–	–	–	68	82	–		C
	AD11	I/O	–	–	–	116		D2
PDL12	–	–	–	–	83	–		C
	AD12	I/O	–	–	–	117		D2
PDL13	–	–	–	–	84	–		C
	AD13	I/O	–	–	–	118		D2
PDL14	AD14	I/O	–	–	–	119		D2
PDL15	AD15	I/O	–	–	–	120		D2

Note Because the FLMD1 pin is used in the flash programming mode, it does not have to be manipulated by using a port control register. For details, see **CHAPTER 26 FLASH MEMORY**.

(1) Port register DL (PDL)

(a) V850ES/HE3

After reset: 00H (output latch) R/W Address: FFFFF004H

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

PDLn	Output data control (in output mode) (n = 0 to 7)
0	Outputs 0
1	Outputs 1

(b) V850ES/HF3

After reset: 0000H (output latch) R/W Address: PDL FFFFF004H, PDLL FFFFF0004H, PDLH FFFFF005H

	15	14	13	12	11	10	9	8
PDL (PDLH)	0	0	0	0	PDL11	PDL10	PDL9	PDL8
	7	6	5	4	3	2	1	0
(PDLL)	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0

PDLn	Output data control (in output mode) (n = 0 to 11)
0	Outputs 0
1	Outputs 1

(c) V850ES/HG3

After reset: 0000H (output latch) R/W Address: PDL FFFFF004H, PDLL FFFFF0004H, PDLH FFFFF005H

	15	14	13	12	11	10	9	8
PDL (PDLH)	0	0	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8
	7	6	5	4	3	2	1	0
(PDLL)	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0

PDLn	Output data control (in output mode) (n = 0 to 13)
0	Outputs 0
1	Outputs 1

(2/2)

(d) V850ES/HJ3

After reset: 0000H (output latch) R/W Address: PDL FFFFF004H, PDLL FFFFF0004H, PDLH FFFFF005H

	15	14	13	12	11	10	9	8
PDL (PDLH)	PDL15	PDL14	PDL13	PDL12	PDL11	PDL10	PDL9	PDL8
	7	6	5	4	3	2	1	0
(PDLL)	PDL7	PDL6	PDL5	PDL4	PDL3	PDL2	PDL1	PDL0

PDLn	Output data control (in output mode) (n = 0 to 15)
0	Outputs 0
1	Outputs 1

Remarks 1. The PDL register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PDL register as the PDLH register and the lower 8 bits as the PDLL register, PDL can be read or written in 8-bit or 1-bit units.

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

(2) Port mode register DL (PMDL)

(1/2)

(a) V850ES/HE3

After reset: FFH R/W Address: FFFFF024H

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

PMDLn	I/O mode control (n = 0 to 7)
0	Output mode
1	Input mode

(2/2)

(b) V850ES/HF3

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

	15	14	13	12	11	10	9	8
PMDL (PMDLH)	1	1	1	1	PMDL11	PMDL10	PMDL9	PMDL8
	7	6	5	4	3	2	1	0
(PMDLL)	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0

PMDLn	I/O mode control (n = 0 to 11)
0	Output mode
1	Input mode

(c) V850ES/HG3

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

	15	14	13	12	11	10	9	8
PMDL (PMDLH)	1	1	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8
	7	6	5	4	3	2	1	0
(PMDLL)	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0

PMDLn	I/O mode control (n = 0 to 13)
0	Output mode
1	Input mode

(d) V850ES/HJ3

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

	15	14	13	12	11	10	9	8
PMDL (PMDLH)	PMDL15	PMDL14	PMDL13	PMDL12	PMDL11	PMDL10	PMDL9	PMDL8
	7	6	5	4	3	2	1	0
(PMDLL)	PMDL7	PMDL6	PMDL5	PMDL4	PMDL3	PMDL2	PMDL1	PMDL0

PMDLn	I/O mode control (n = 0 to 15)
0	Output mode
1	Input mode

Remarks 1. The PMDL register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PMDL register as the PMDLH register and the lower 8 bits as the PMDLL register, PMDL can be read or written in 8-bit or 1-bit units.

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

(3) Port mode control register DL (PMCDL) (V850ES/HJ3 only)

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

	15	14	13	12	11	10	9	8
PMCDL (PMCDLH)	PMCDL15	PMCDL14	PMCDL13	PMCDL12	PMCDL11	PMCDL10	PMCDL9	PMCDL8
	7	6	5	4	3	2	1	0
(PMCDLL)	PMCDL7	PMCDL6	PMCDL5	PMCDL4	PMCDL3	PMCDL2	PMCDL1	PMCDL0

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

Remarks 1. The PMCDL register can be read or written in 16-bit units.
However, when using the higher 8 bits of the PMCDL register as the PMCDLH register and the lower 8 bits as the PMCDLL register, PMCDL can be read or written in 8-bit or 1-bit units.

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

4.4 Block Diagrams of Port

Figure 4-6. Block Diagram of Type C

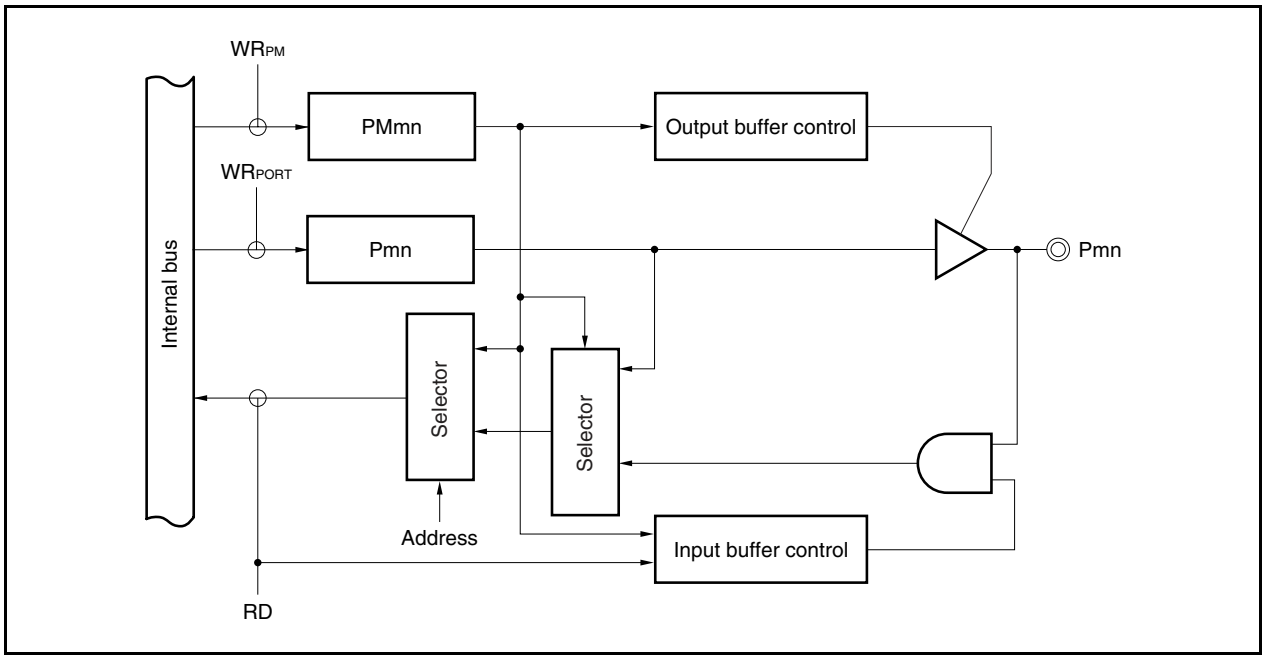


Figure 4-7. Block Diagram of Type C-U

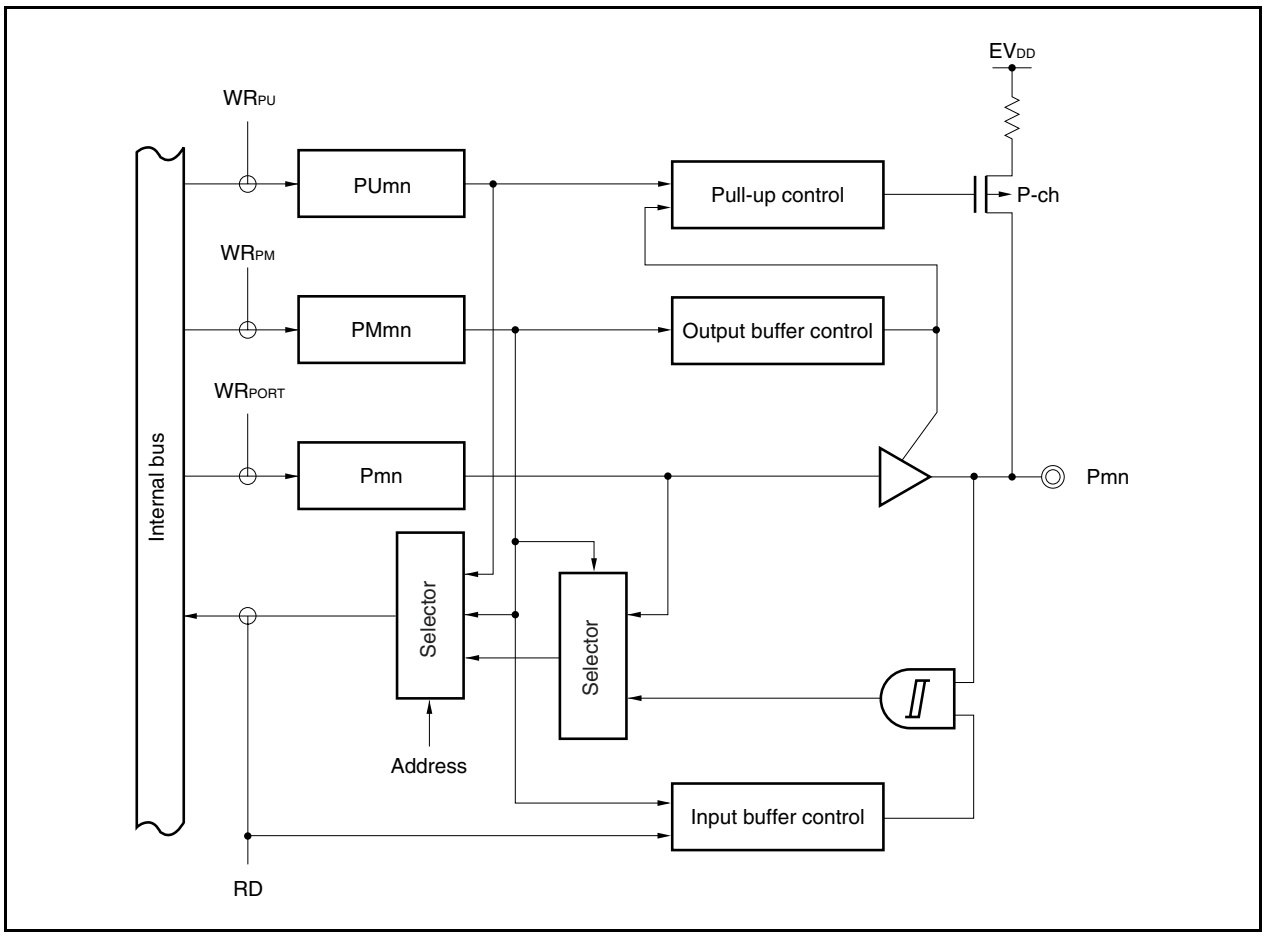


Figure 4-8. Block Diagram of Type D0

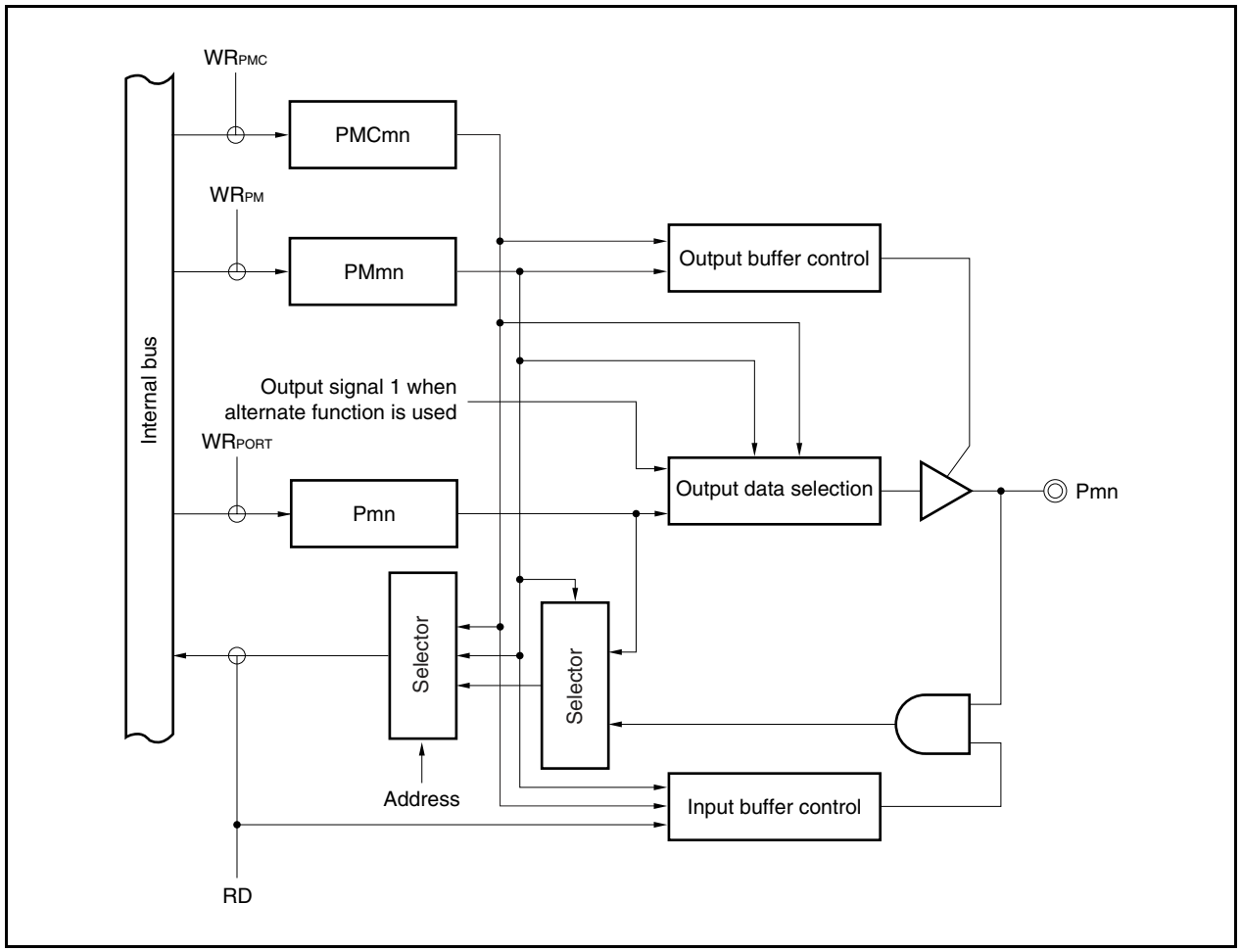


Figure 4-9. Block Diagram of Type D0-U

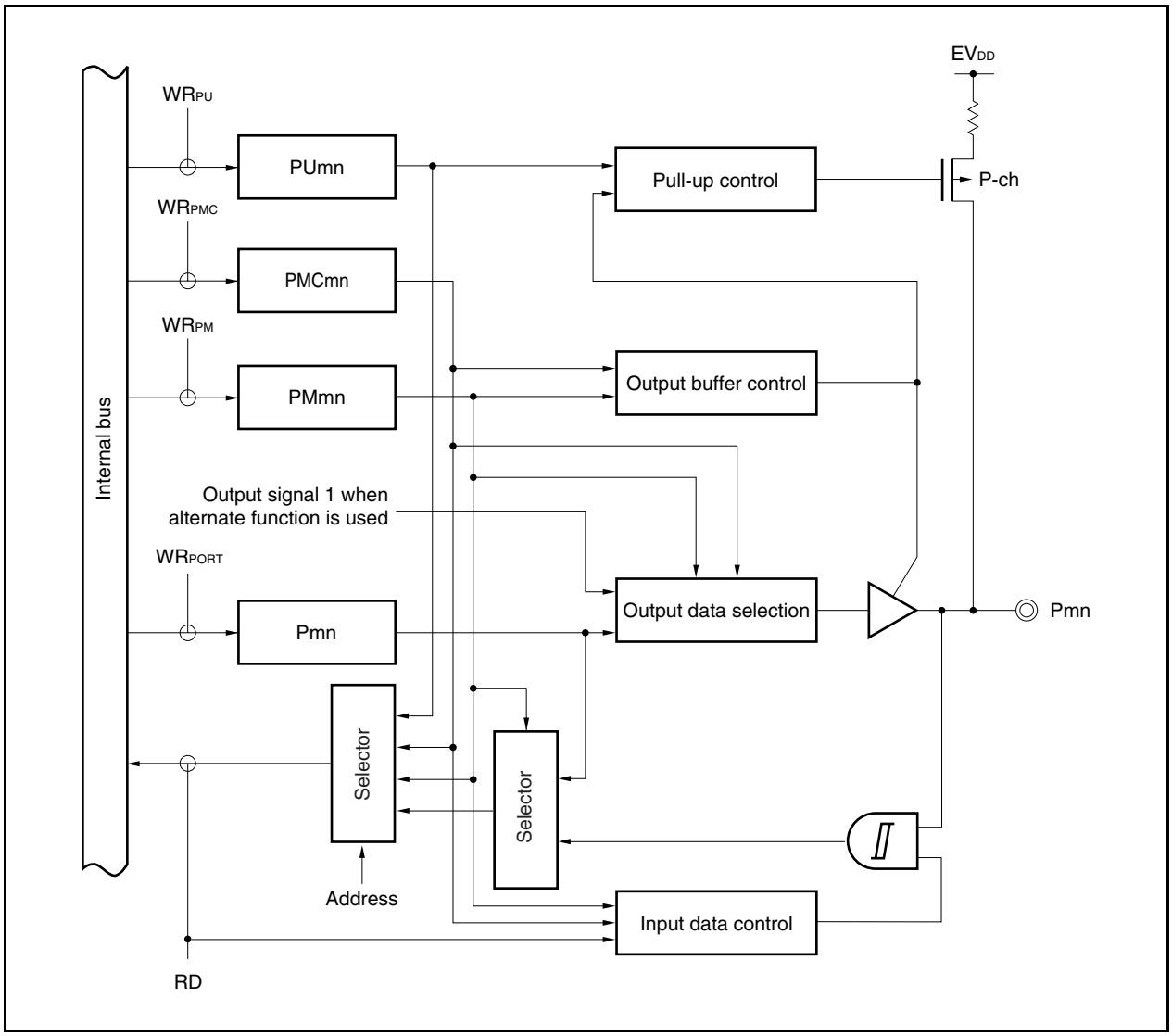


Figure 4-10. Block Diagram of Type D1

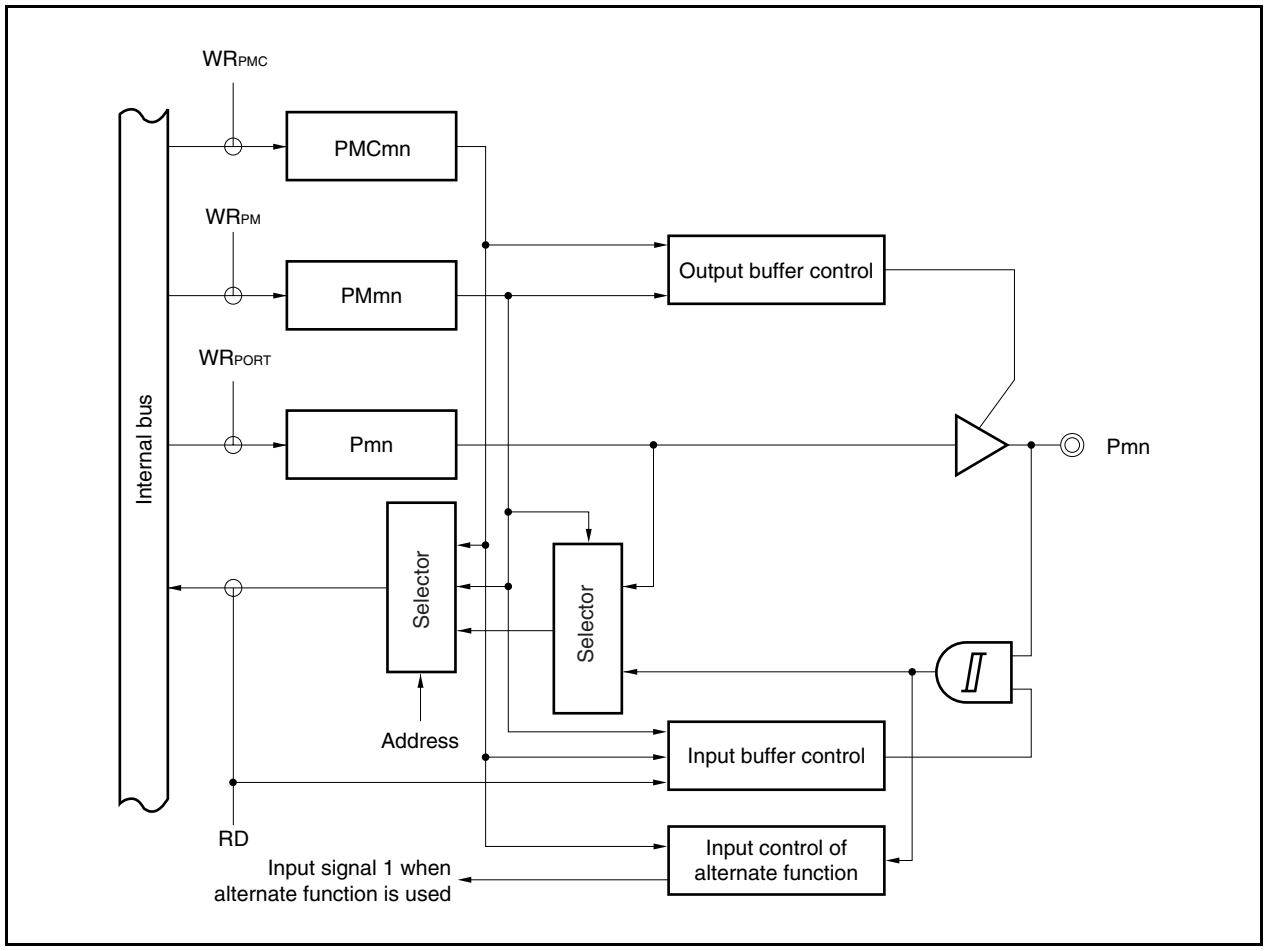


Figure 4-11. Block Diagram of Type D1-U

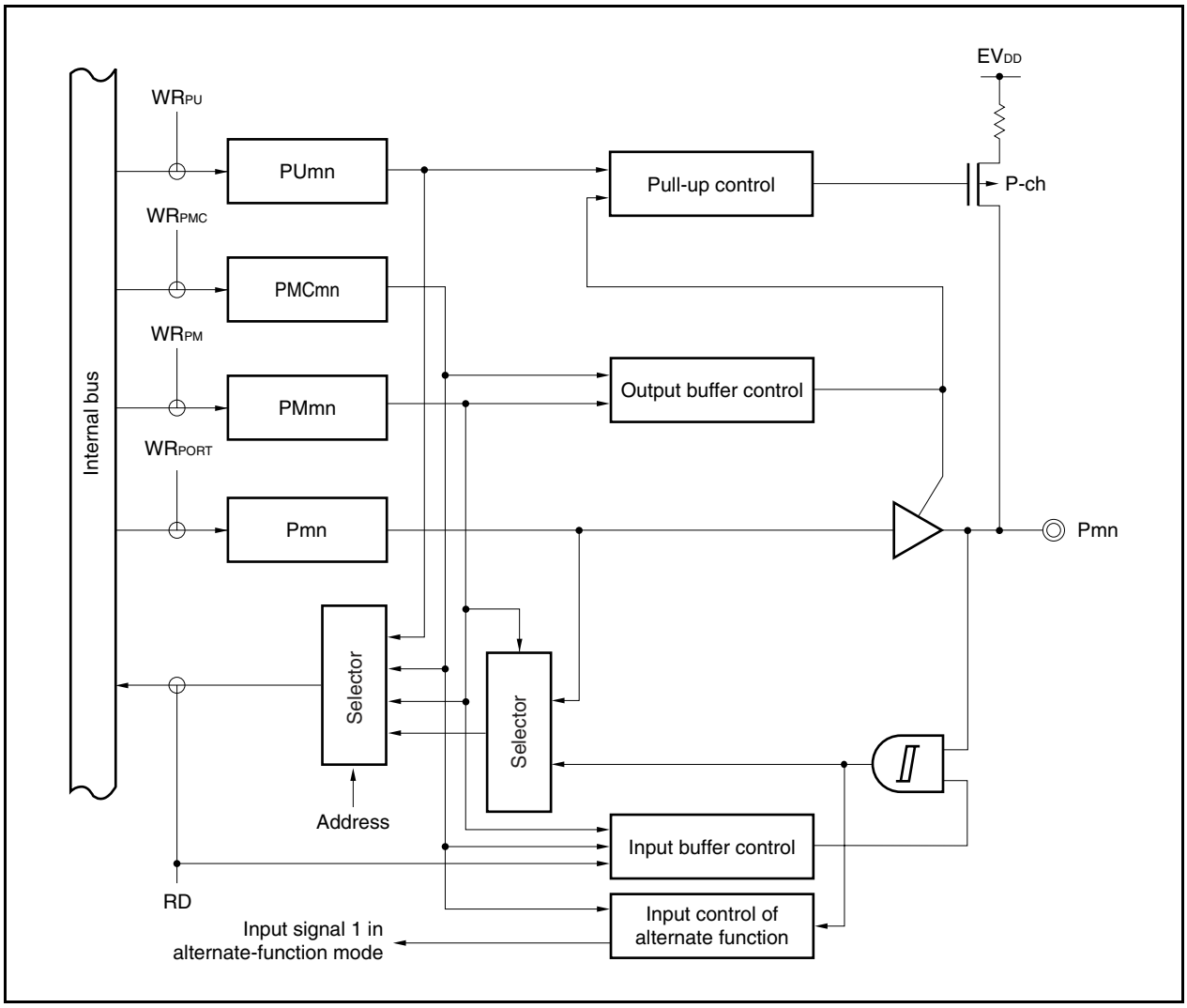


Figure 4-12. Block Diagram of Type D1-UI

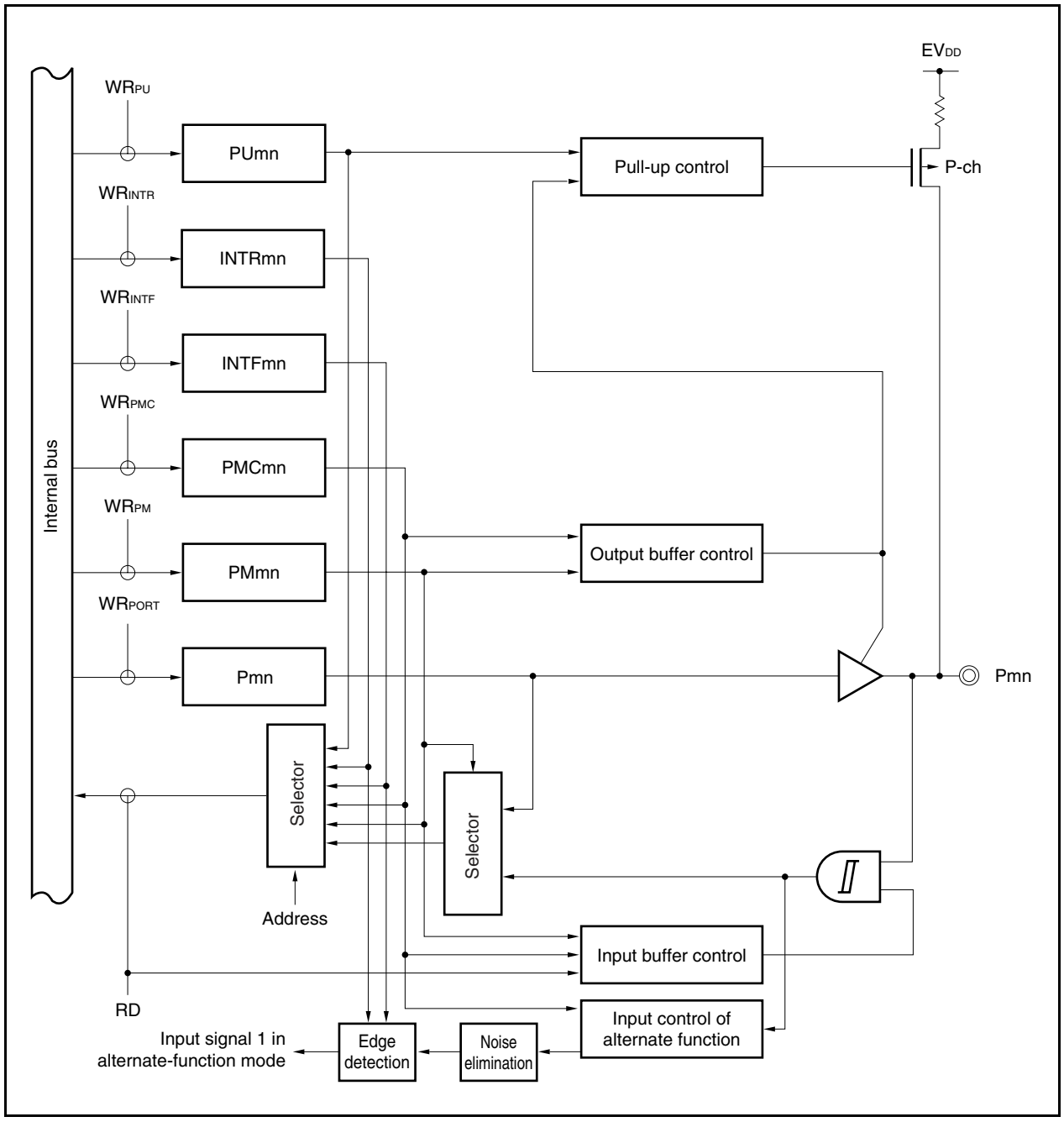


Figure 4-13. Block Diagram of Type D1A

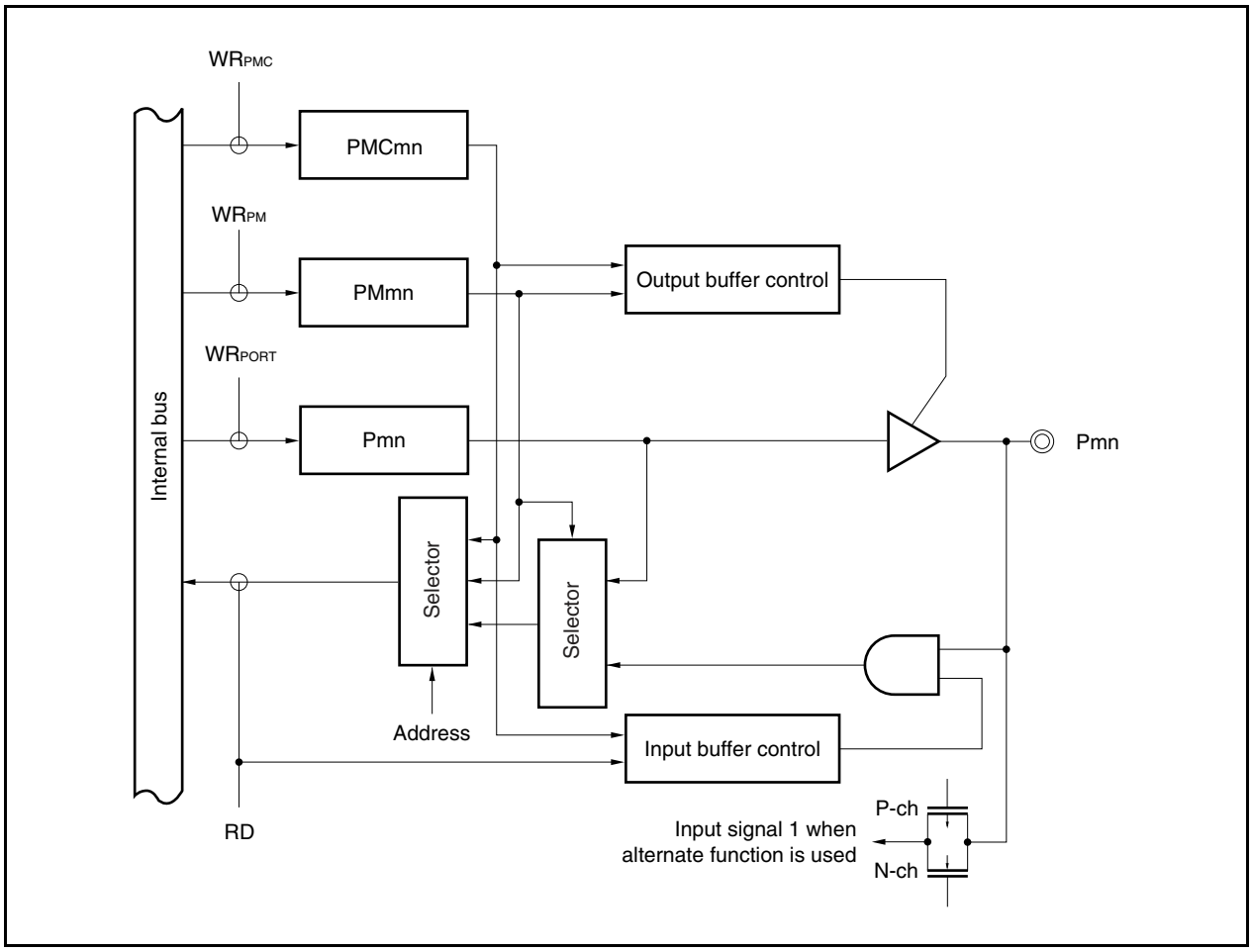


Figure 4-14. Block Diagram of Type D1O1-UI

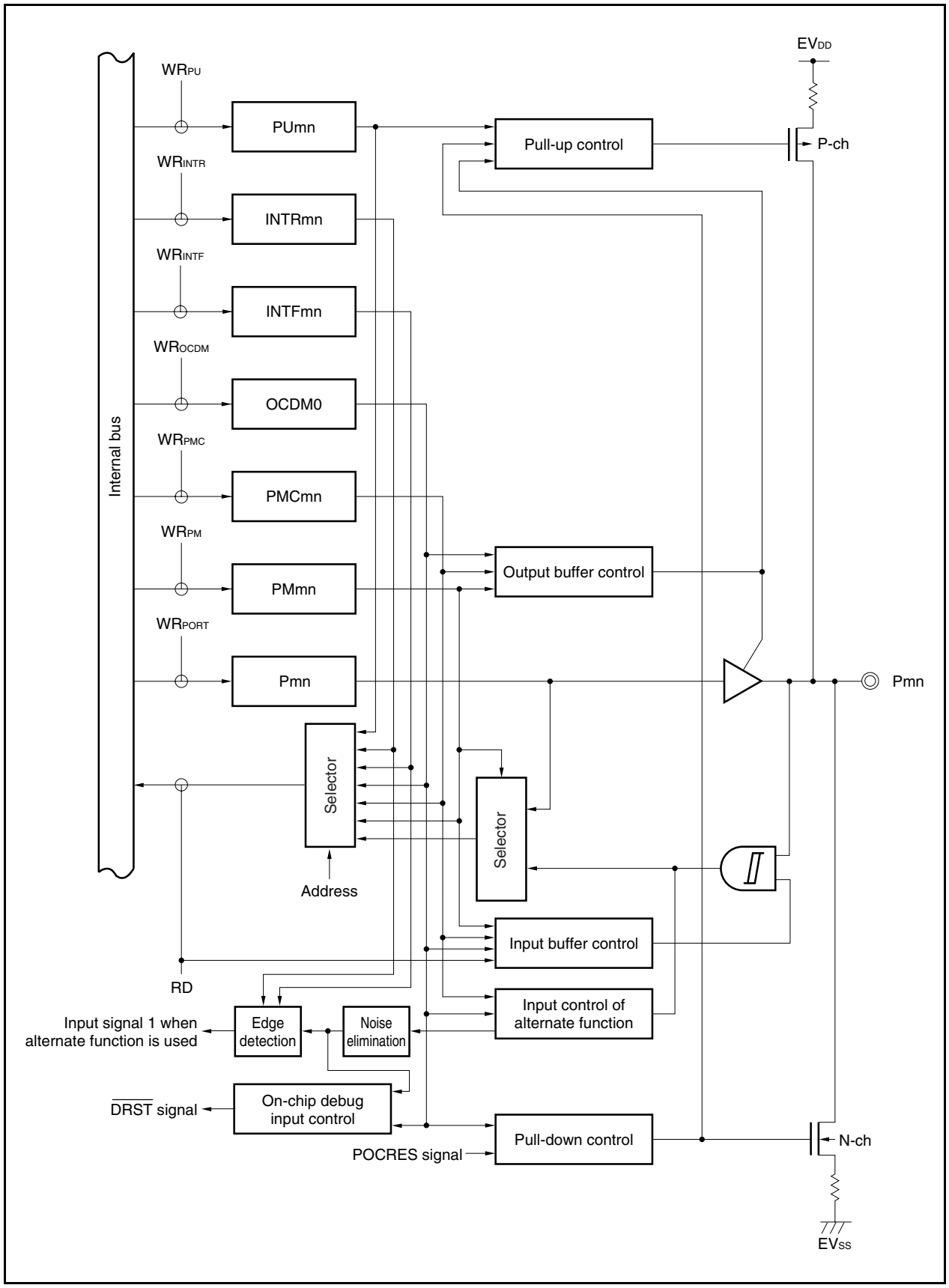


Figure 4-15. Block Diagram of Type D2

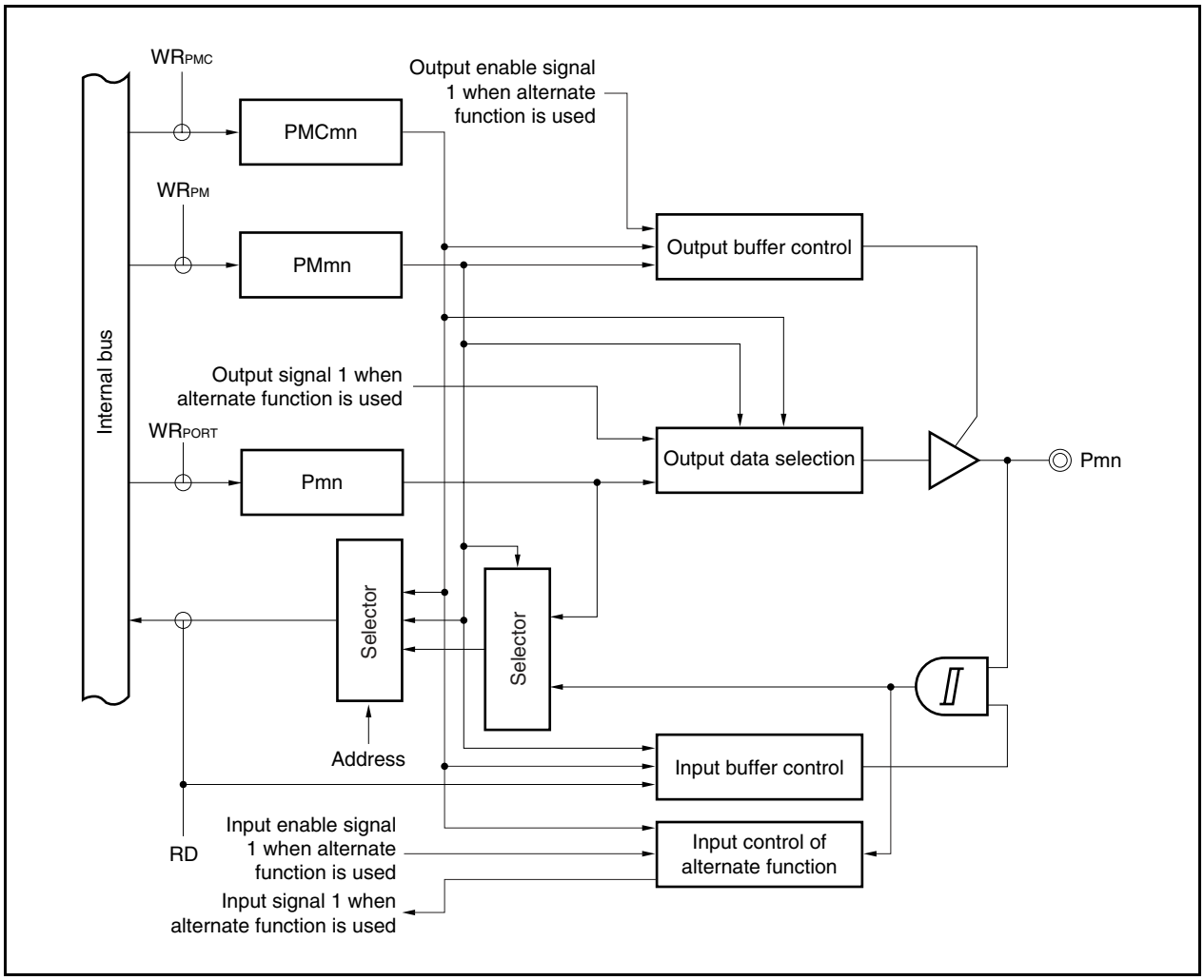


Figure 4-16. Block Diagram of Type D3-UI

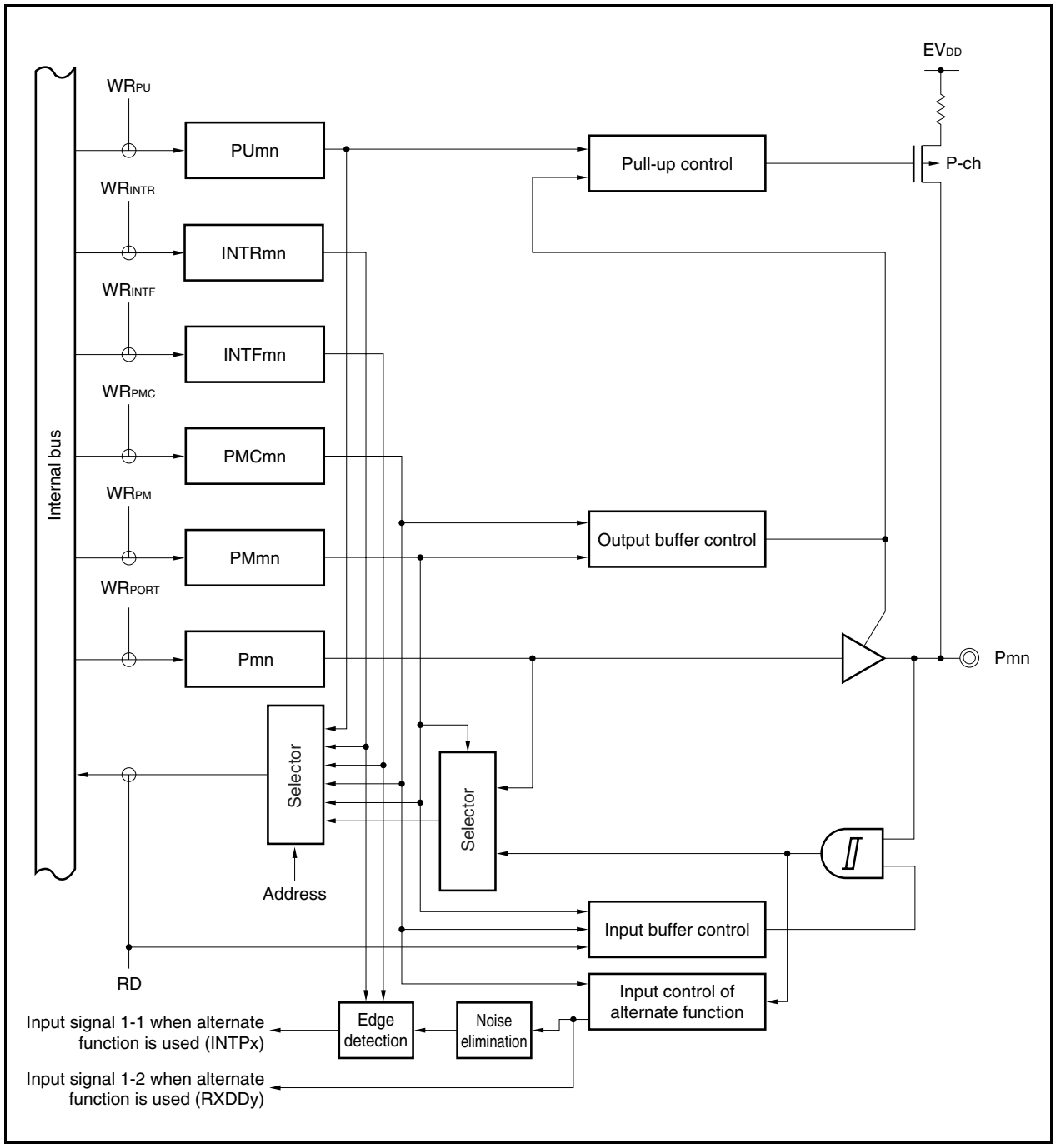


Figure 4-17. Block Diagram of Type E01-U

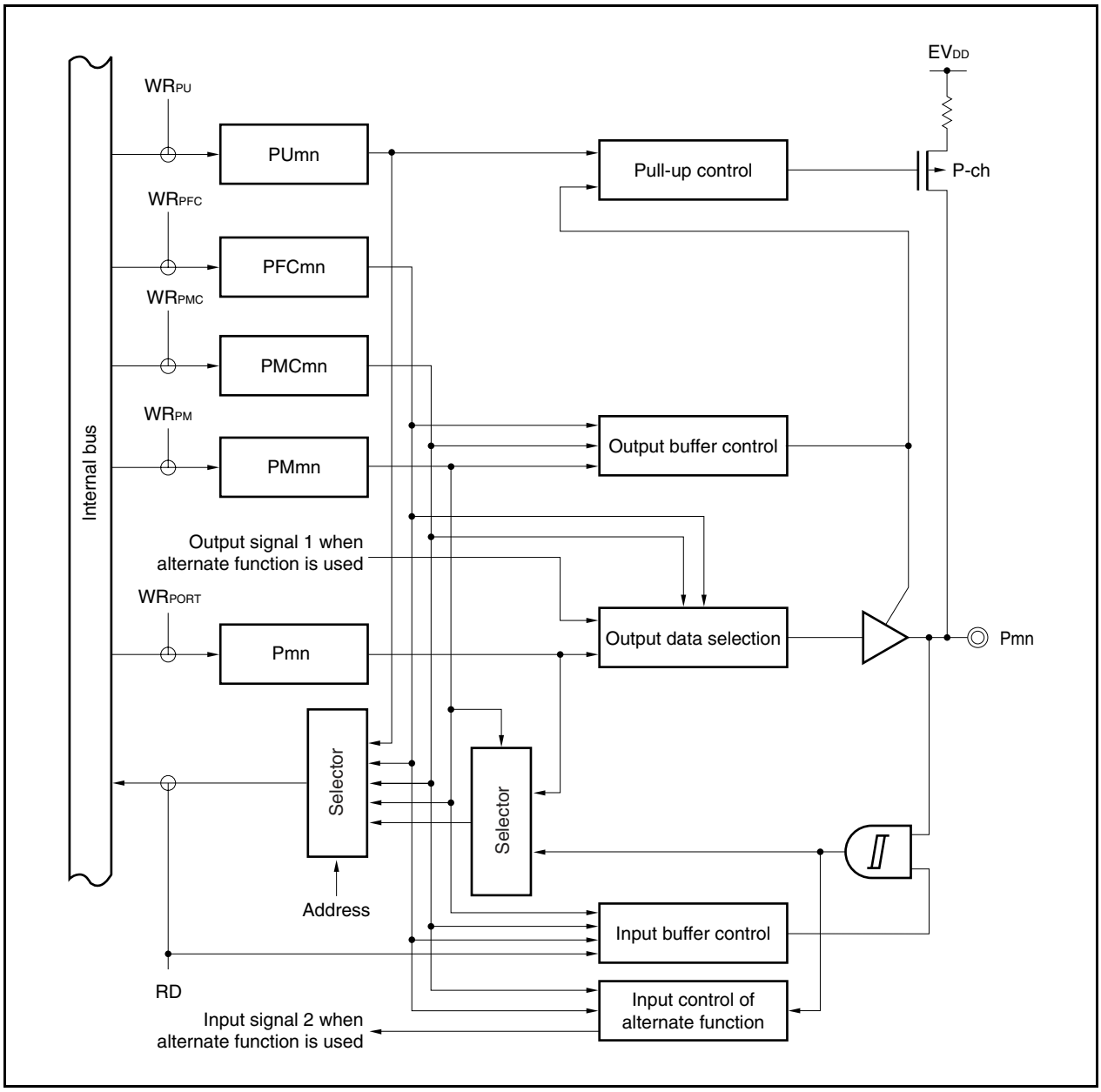


Figure 4-18. Block Diagram of Type E10-U

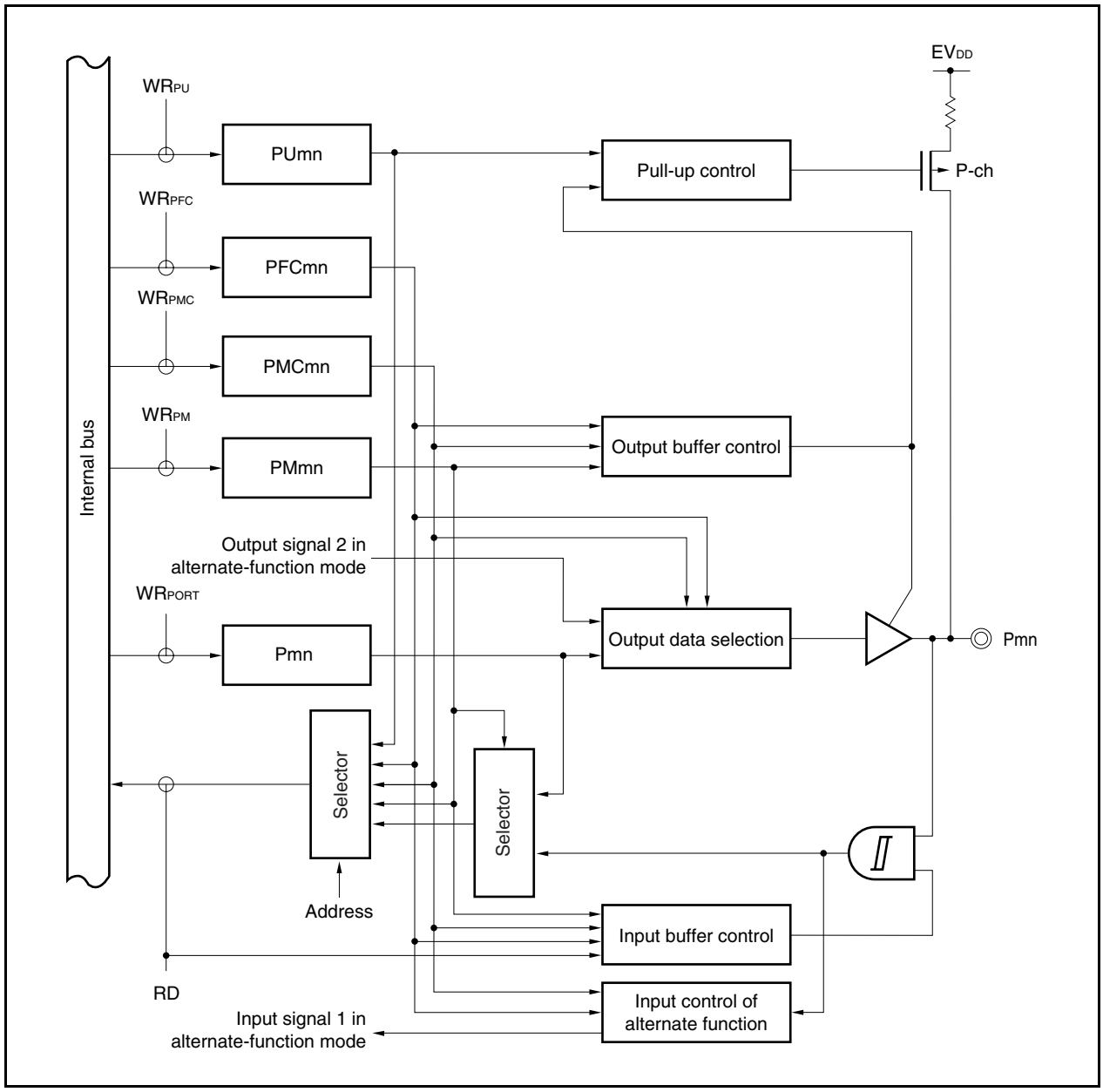


Figure 4-19. Block Diagram of Type E11-U

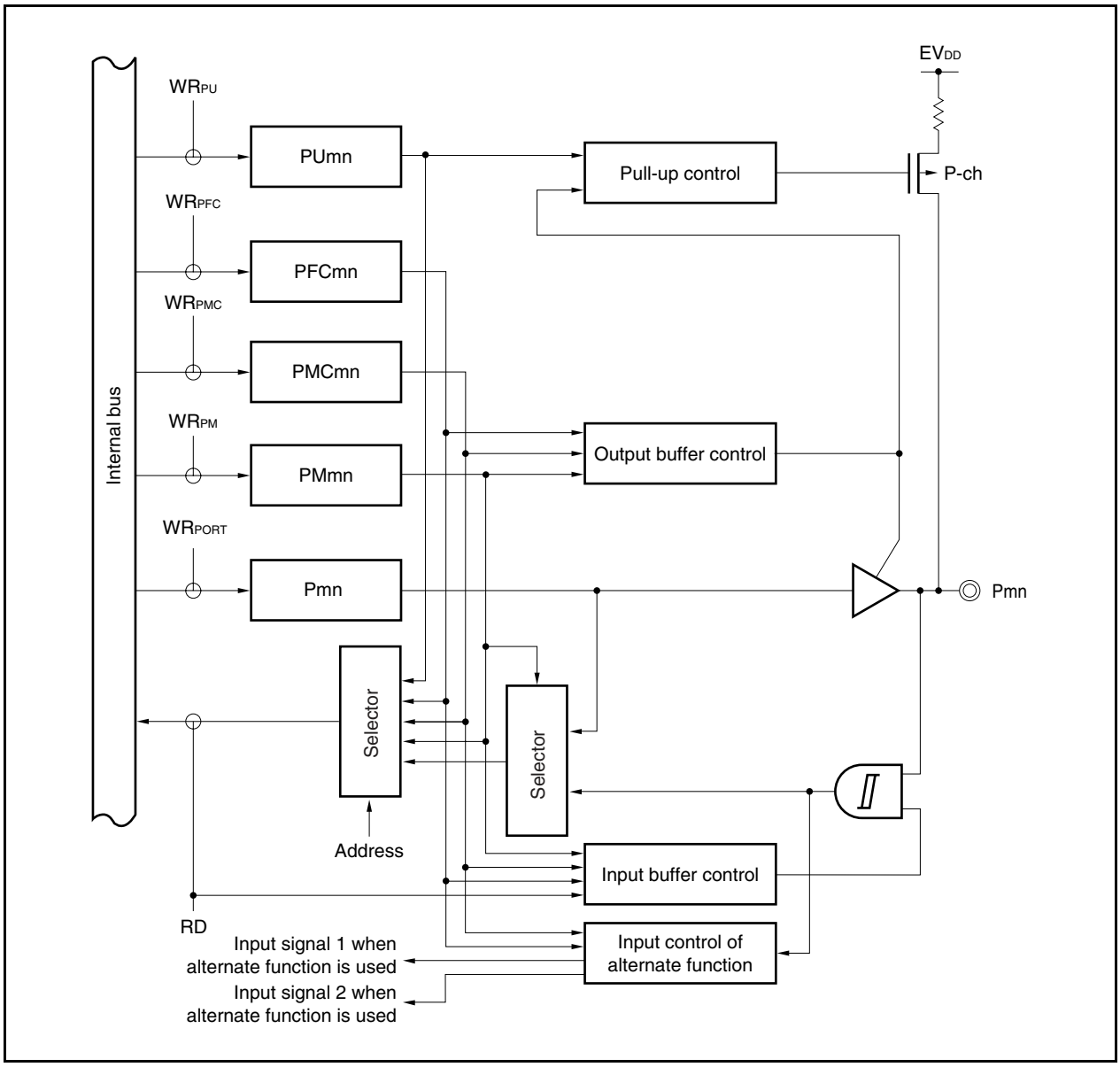


Figure 4-20. Block Diagram of Type E21-U

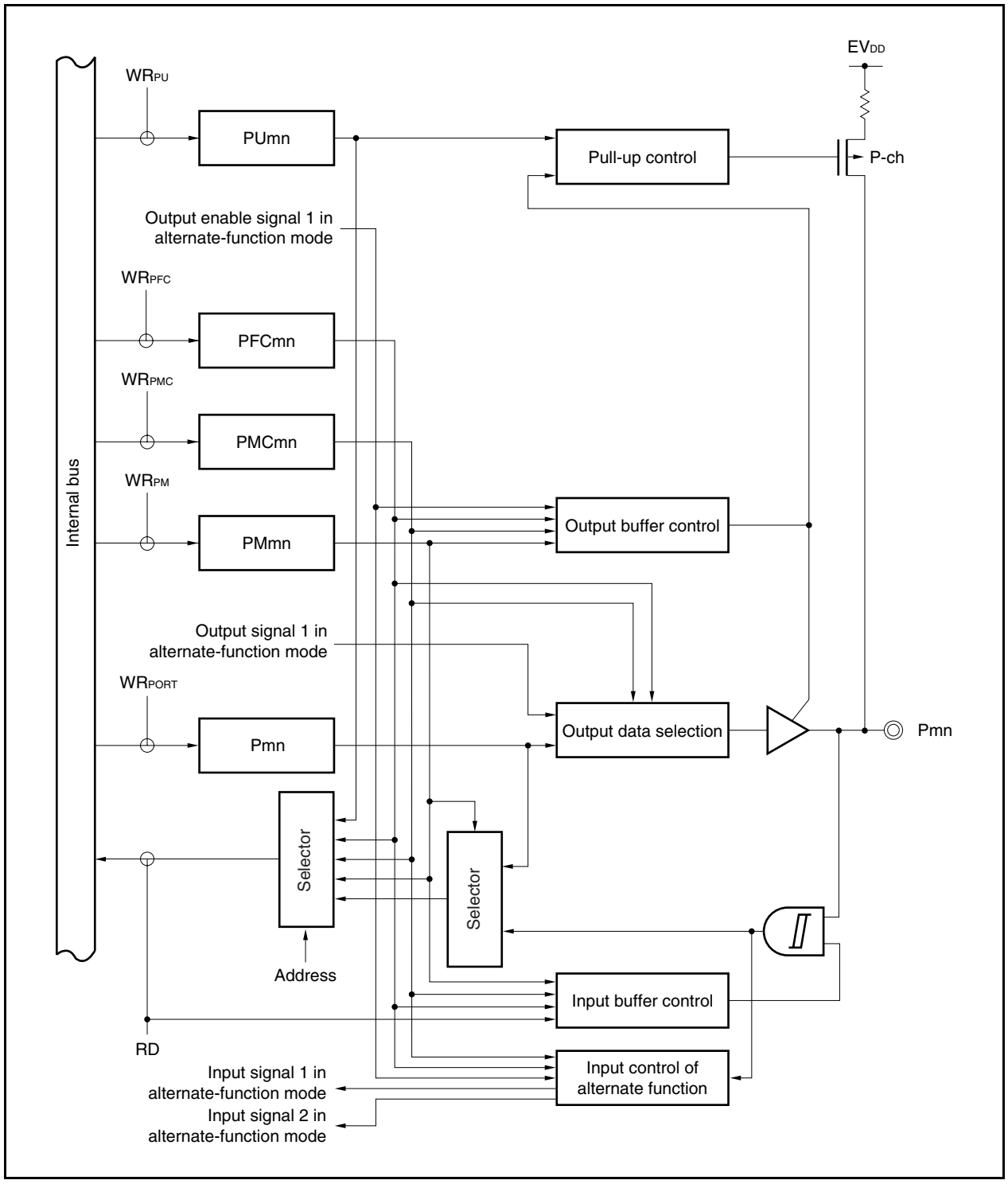


Figure 4-21. Block Diagram of Type Ex0-U

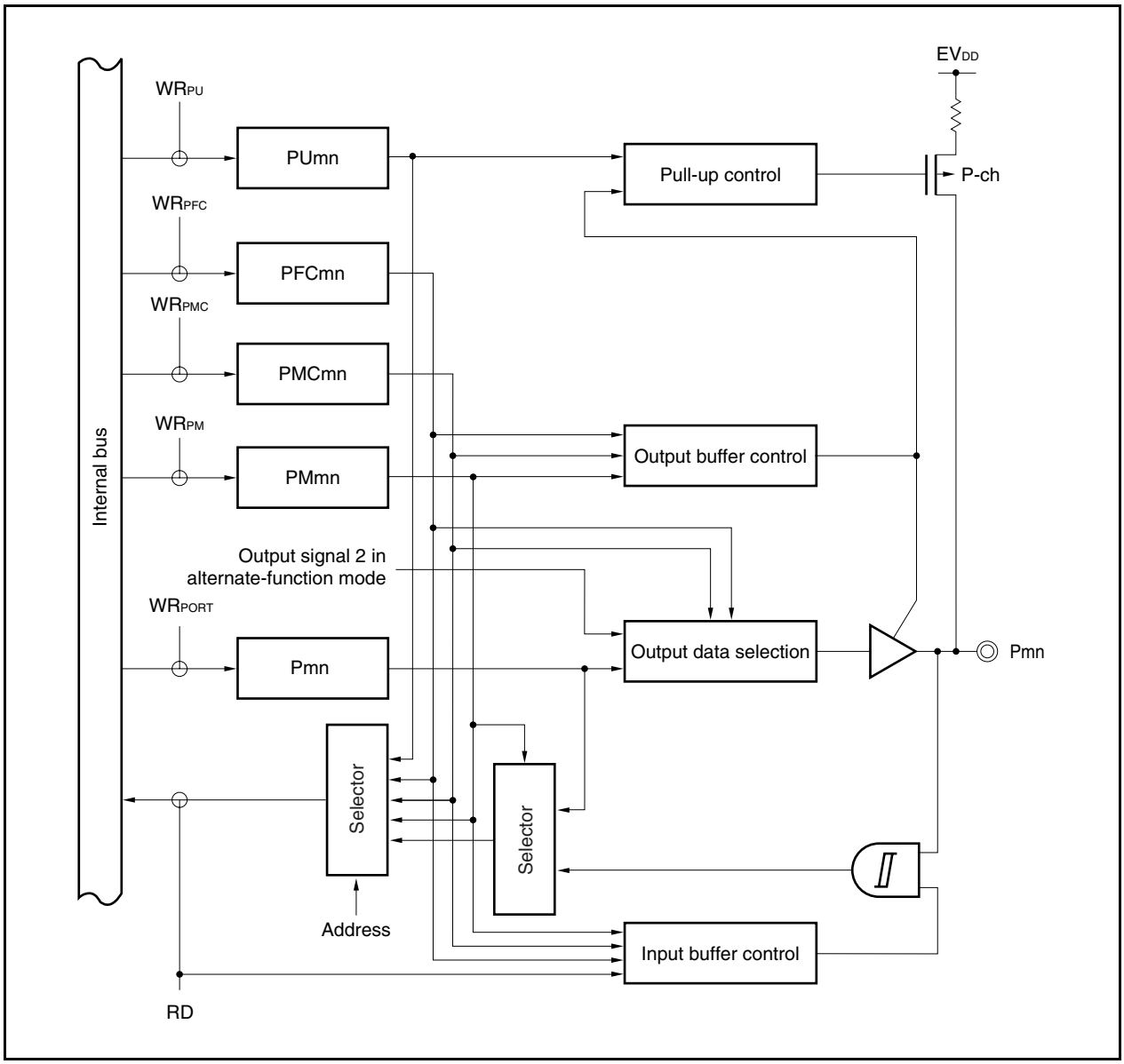


Figure 4-22. Block Diagram of Type Ex1-U

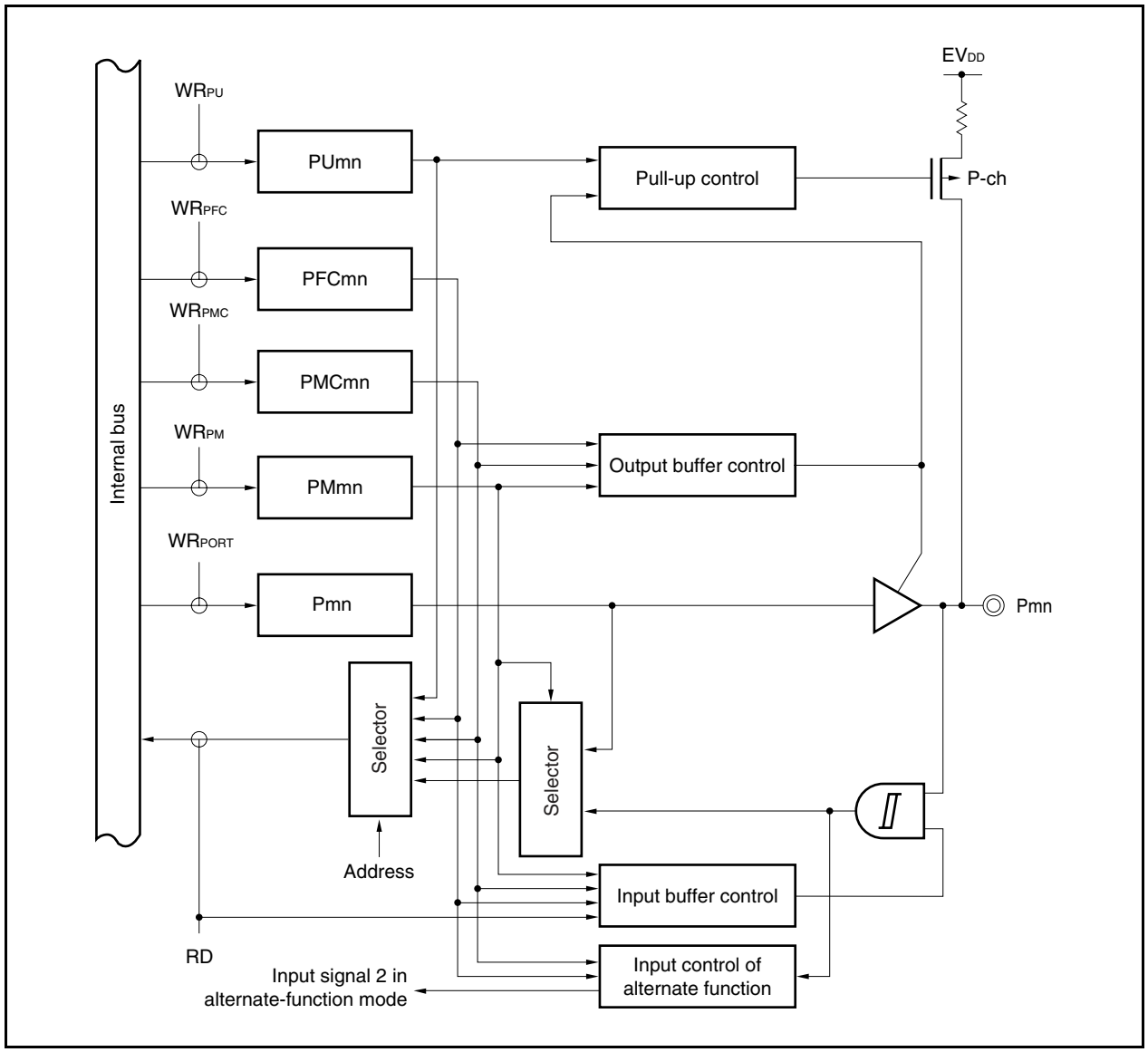


Figure 4-23. Block Diagram of Type Ex1-UI

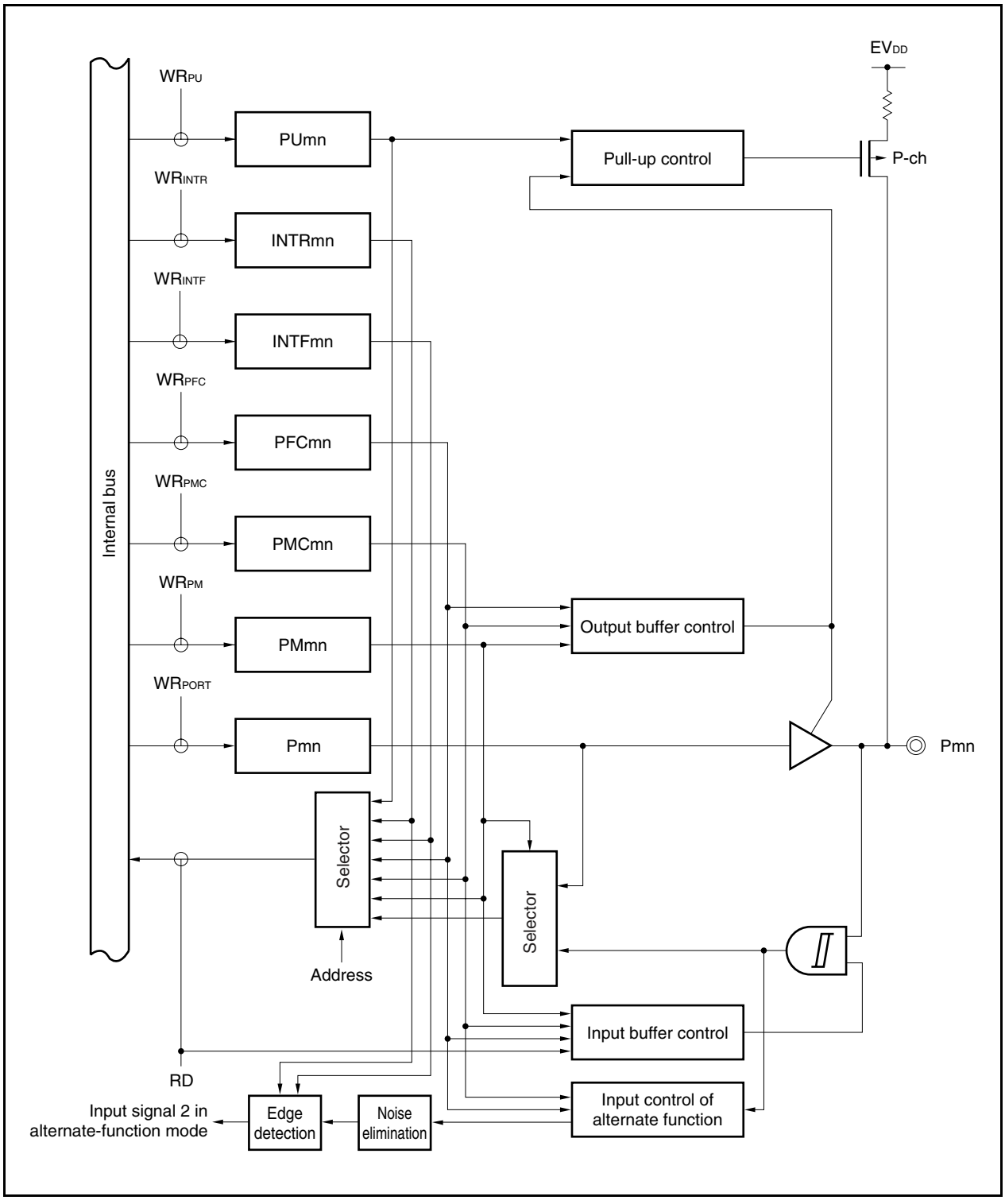


Figure 4-24. Block Diagram of Type Ex2-U

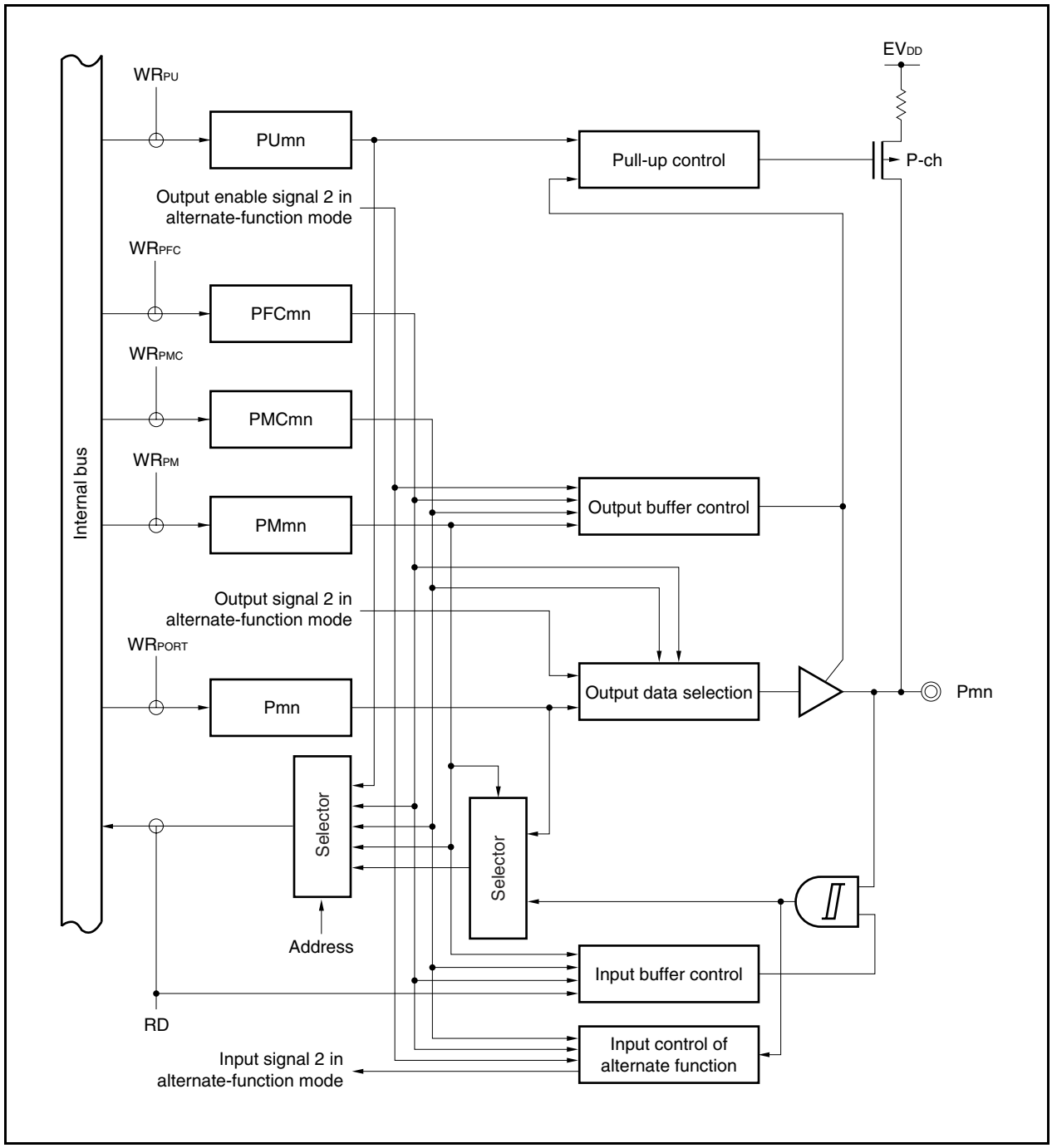


Figure 4-25. Block Diagram of Type F010x-U

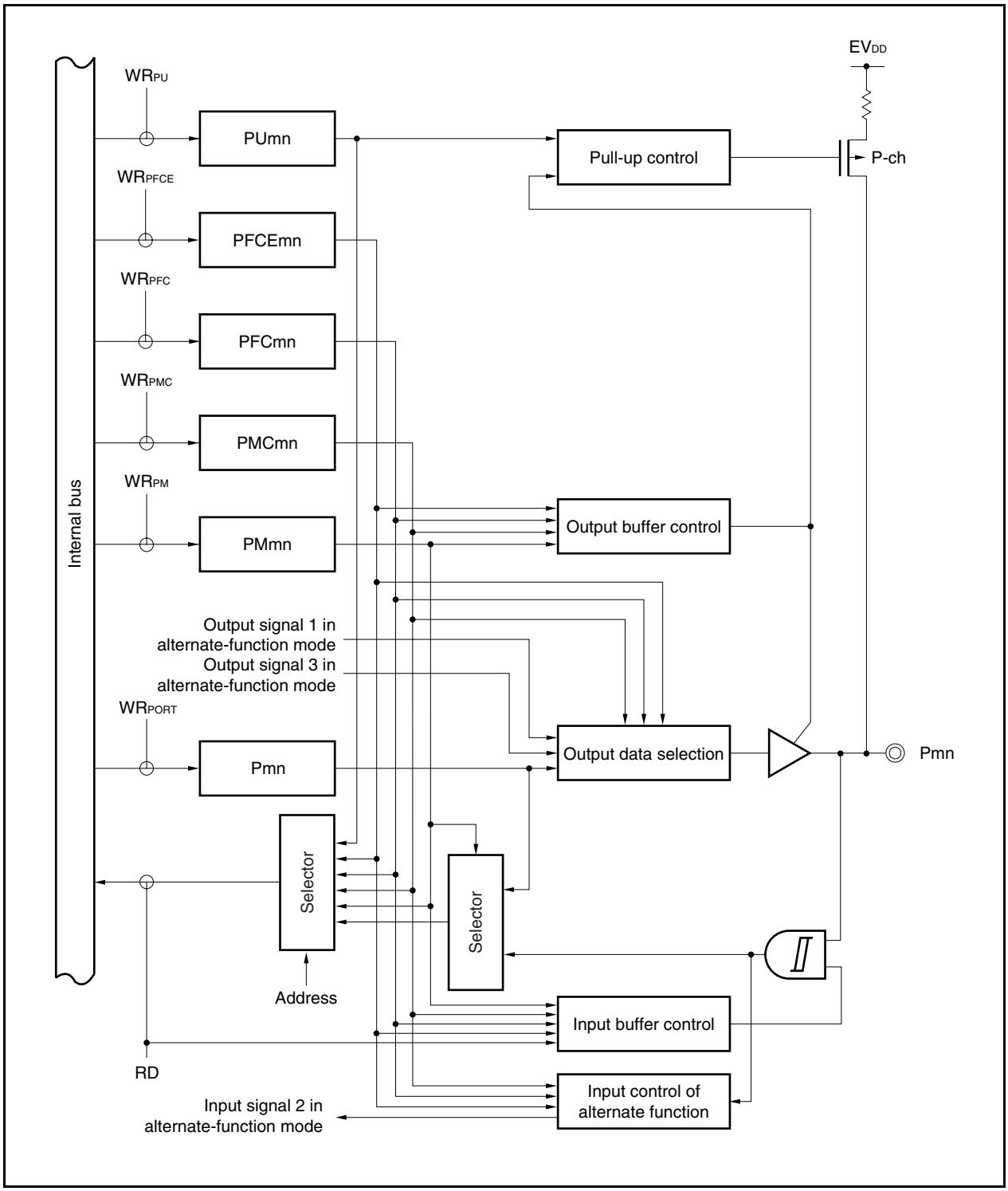


Figure 4-26. Block Diagram of Type F1xx0O1-U

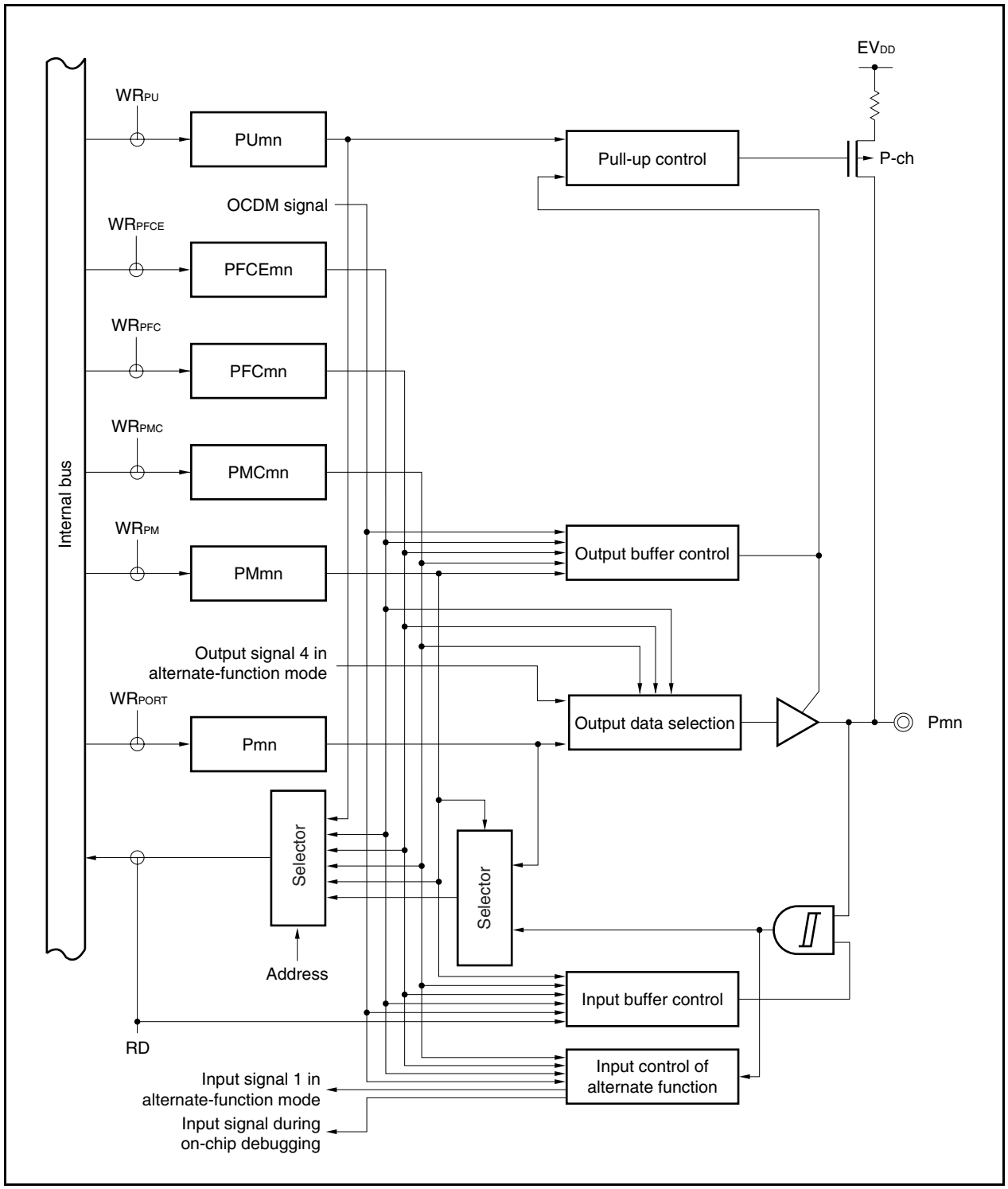


Figure 4-27. Block Diagram of Type F1x10-UI

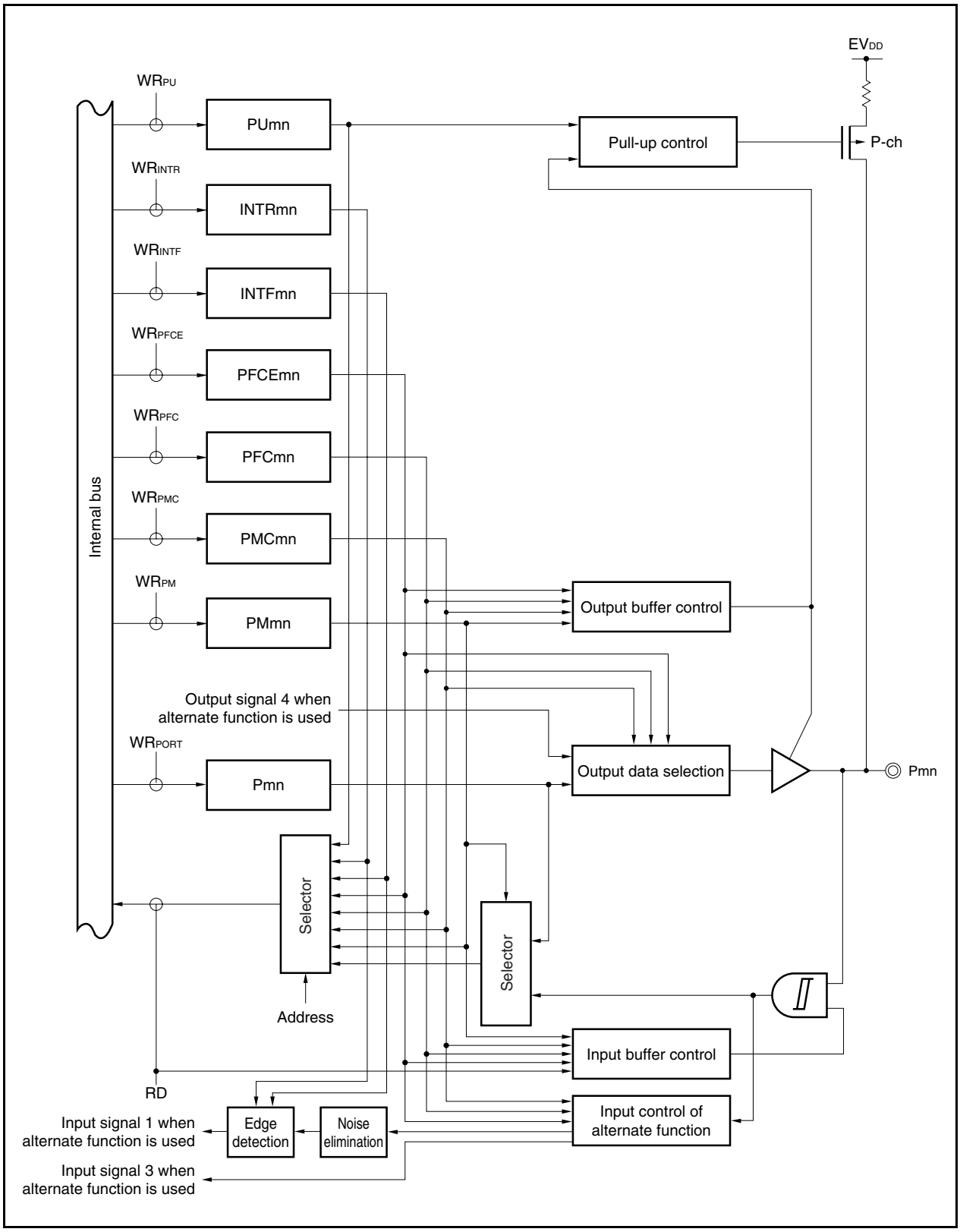


Figure 4-28. Block Diagram of Type F1010-U

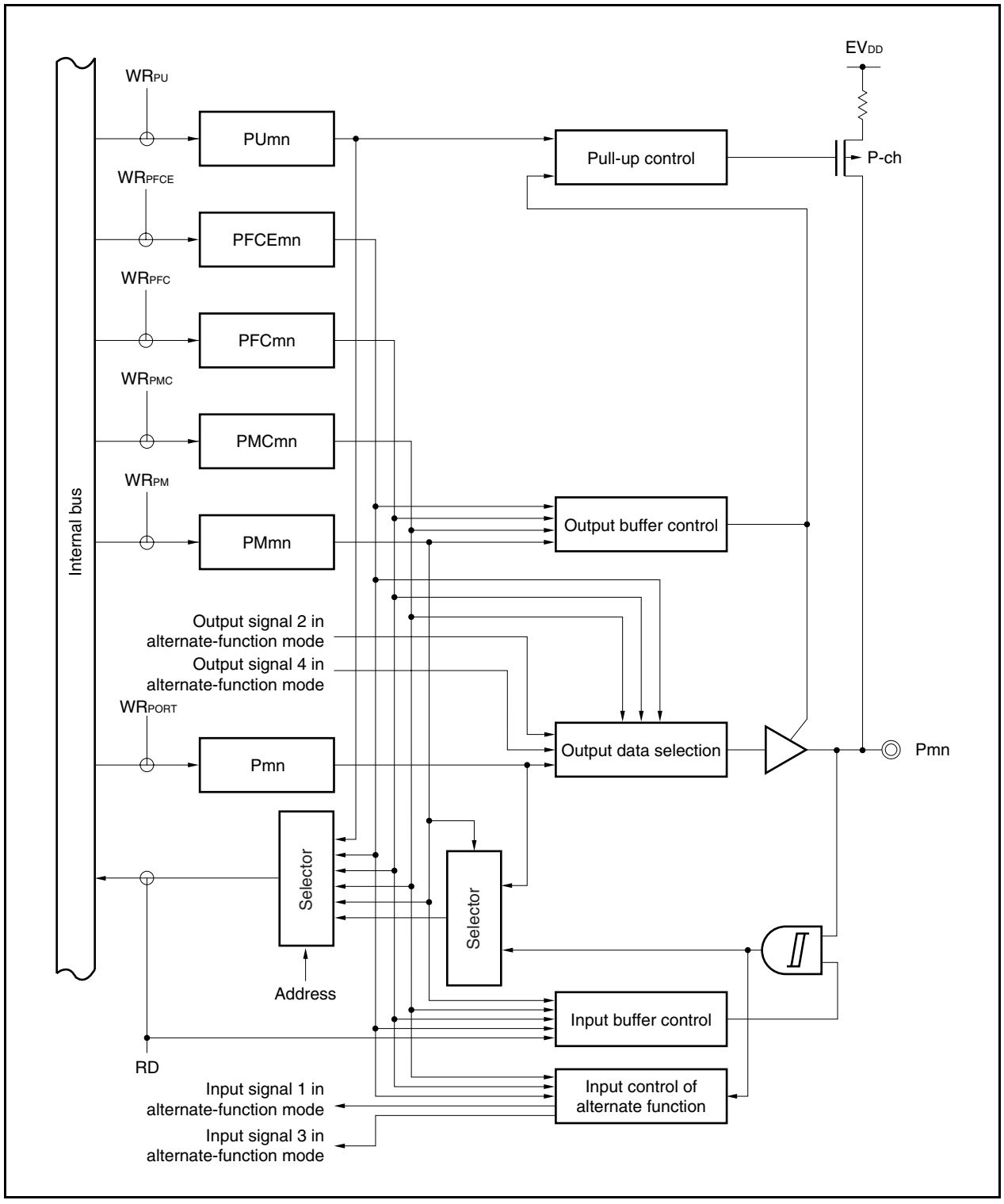


Figure 4-29. Block Diagram of Type F1100O0-U

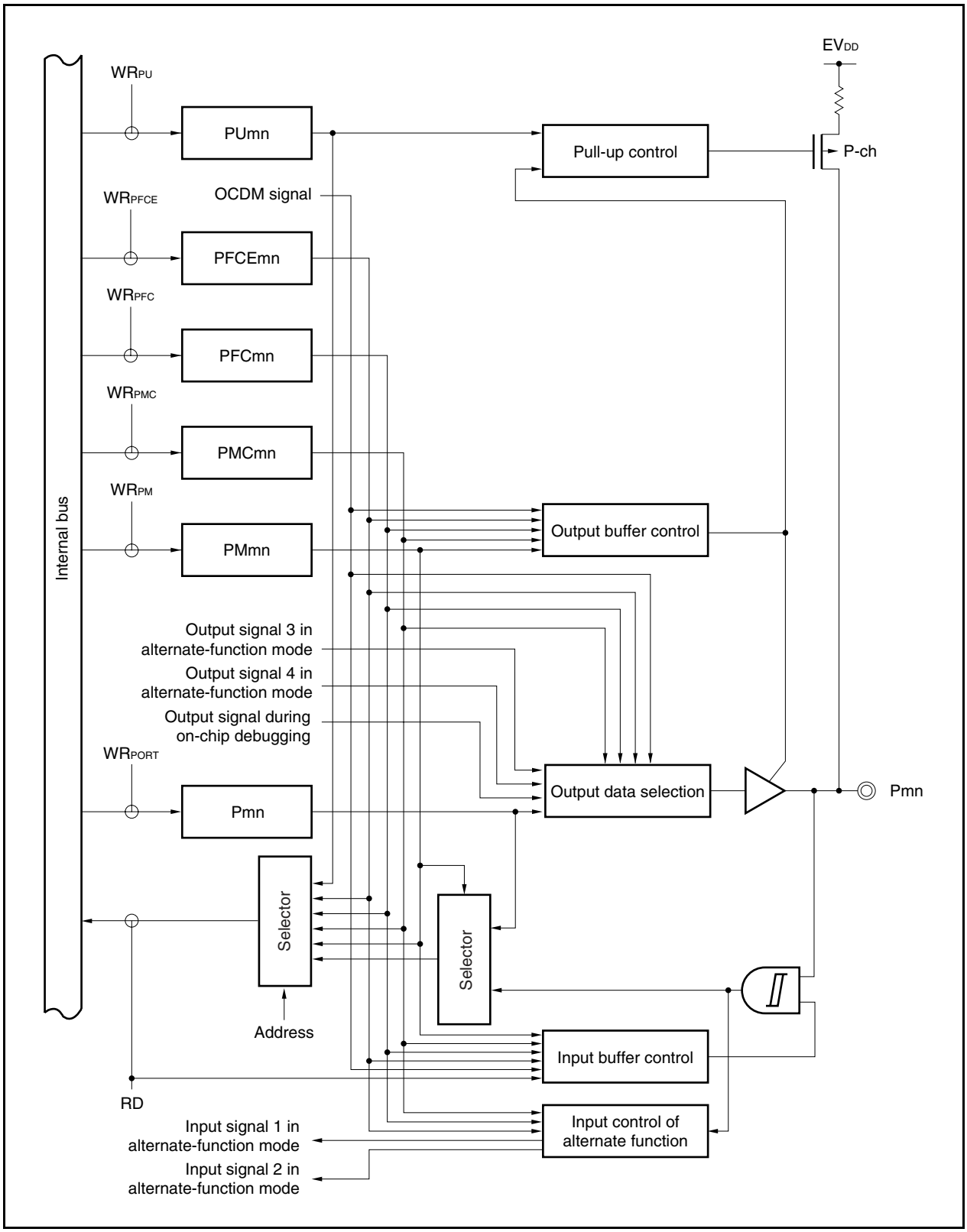


Figure 4-30. Block Diagram of Type F1100O1-U

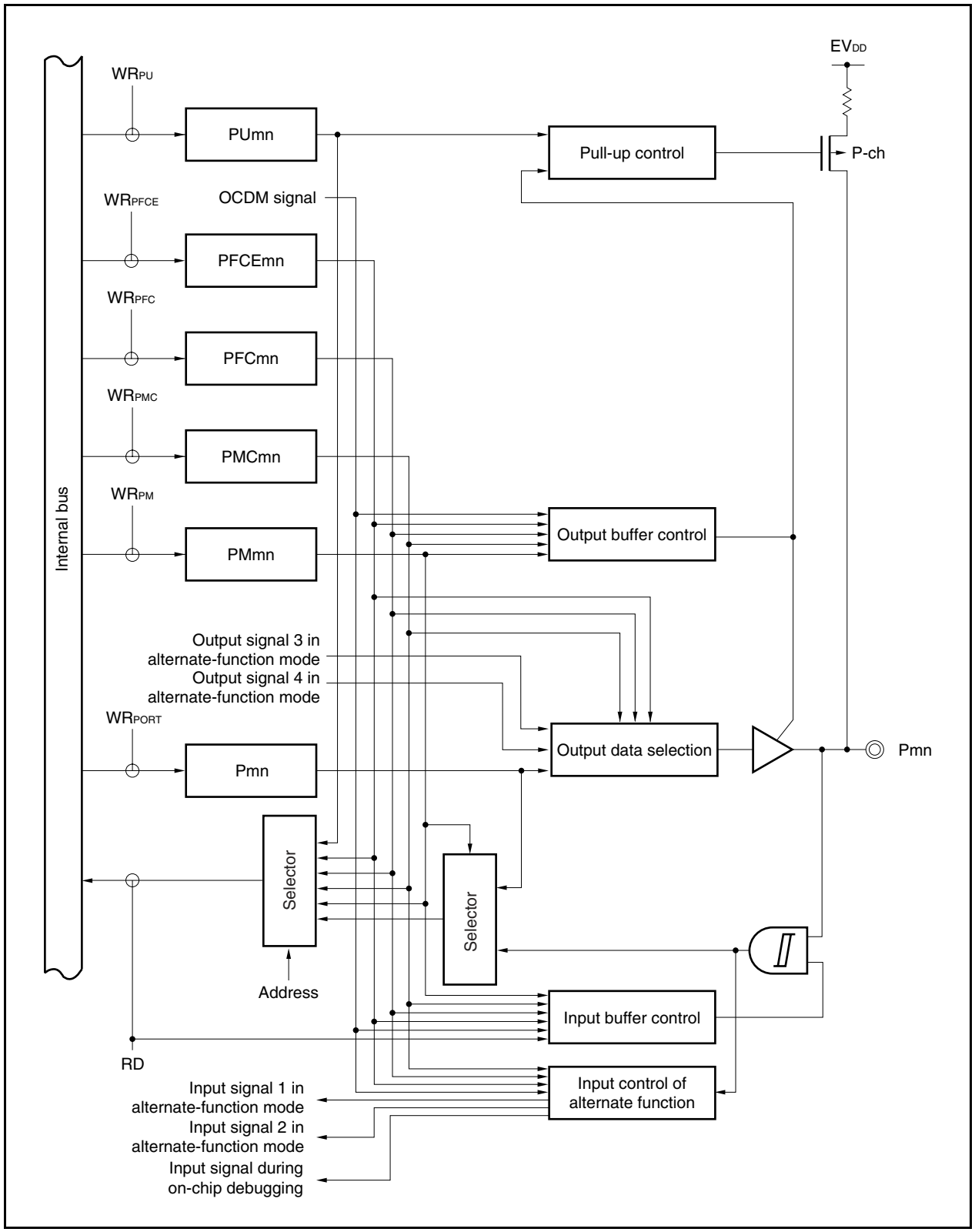


Figure 4-31. Block Diagram of Type F1100-U

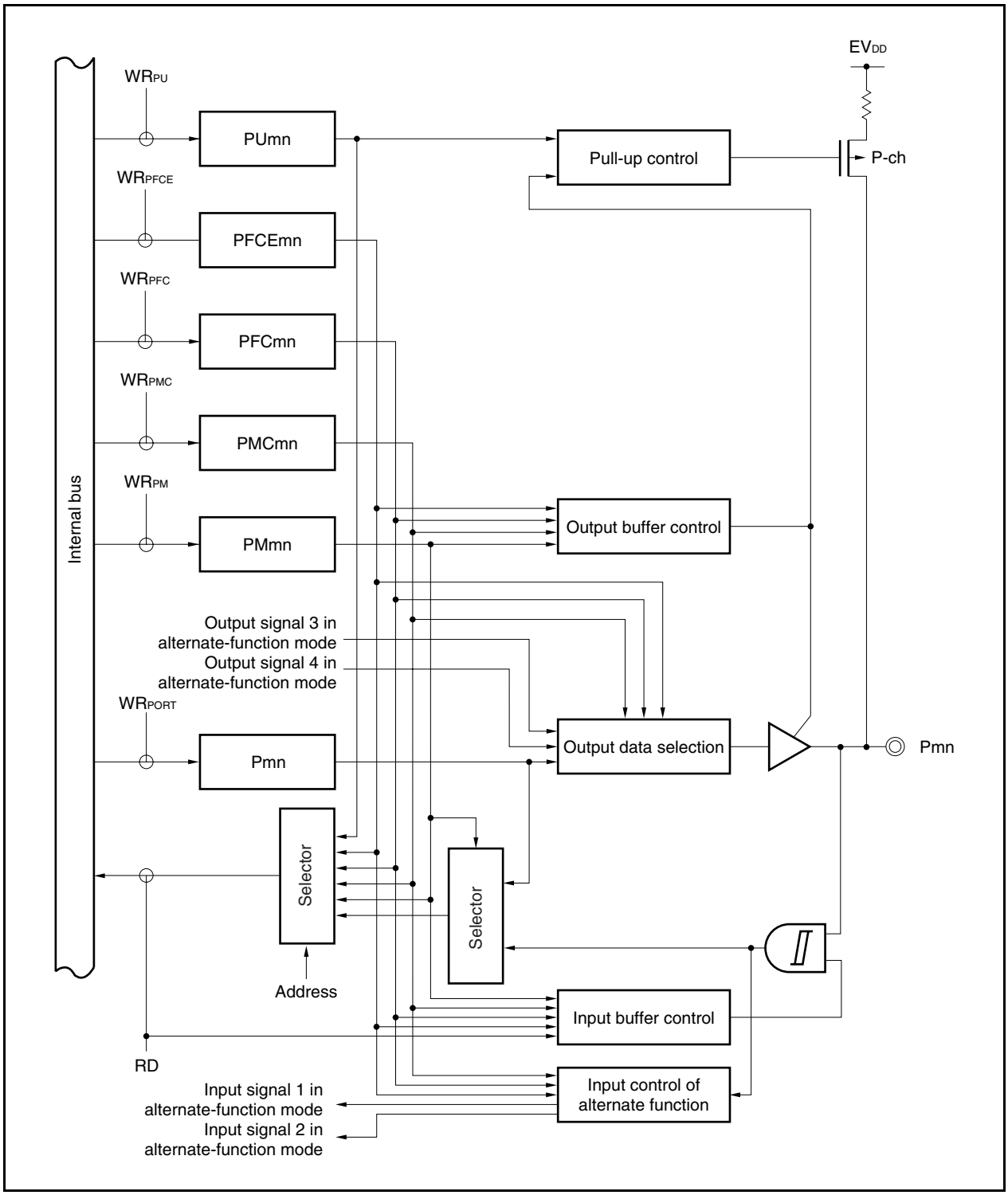


Figure 4-32. Block Diagram of Type F1110-UI

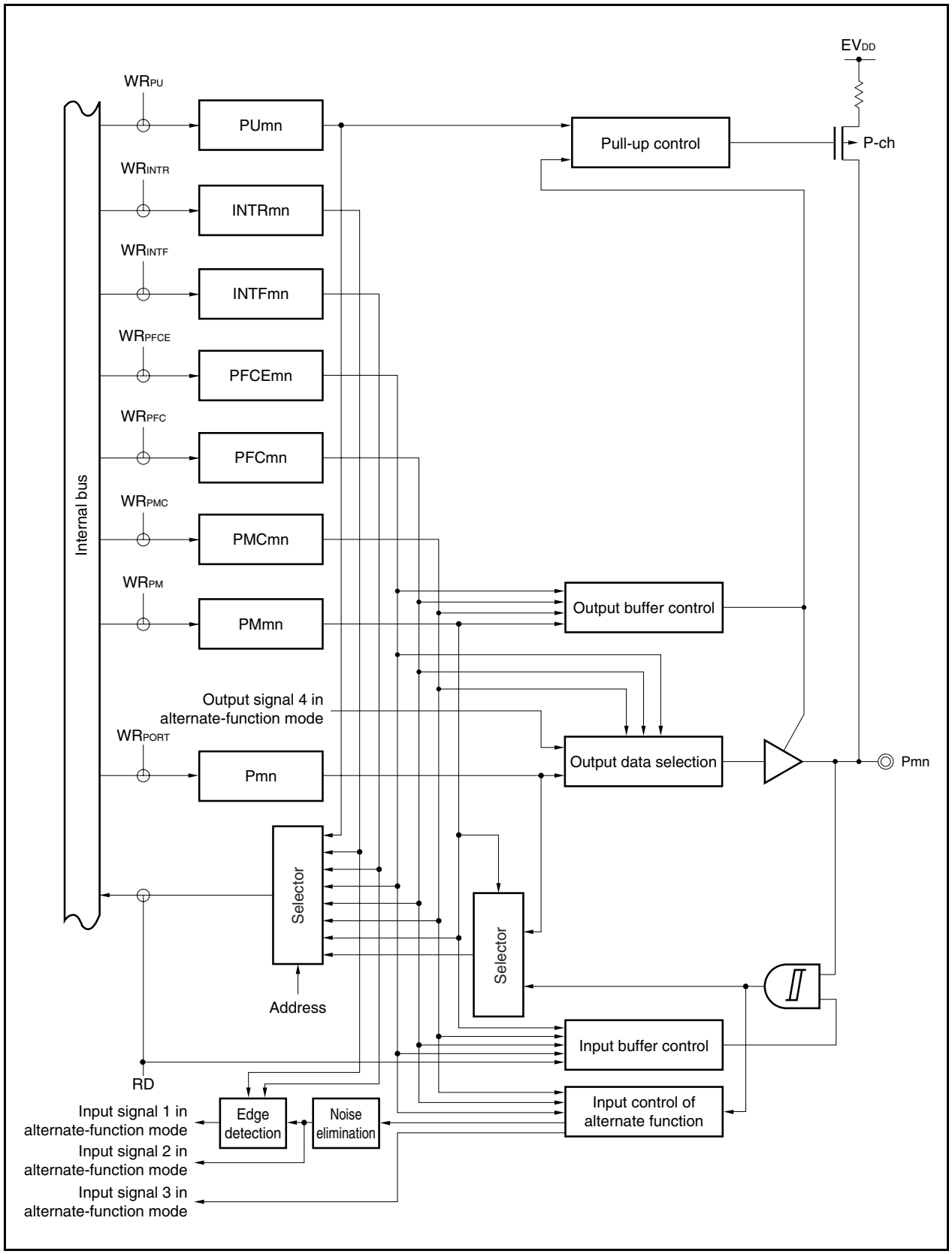


Figure 4-33. Block Diagram of Type F113x-UI

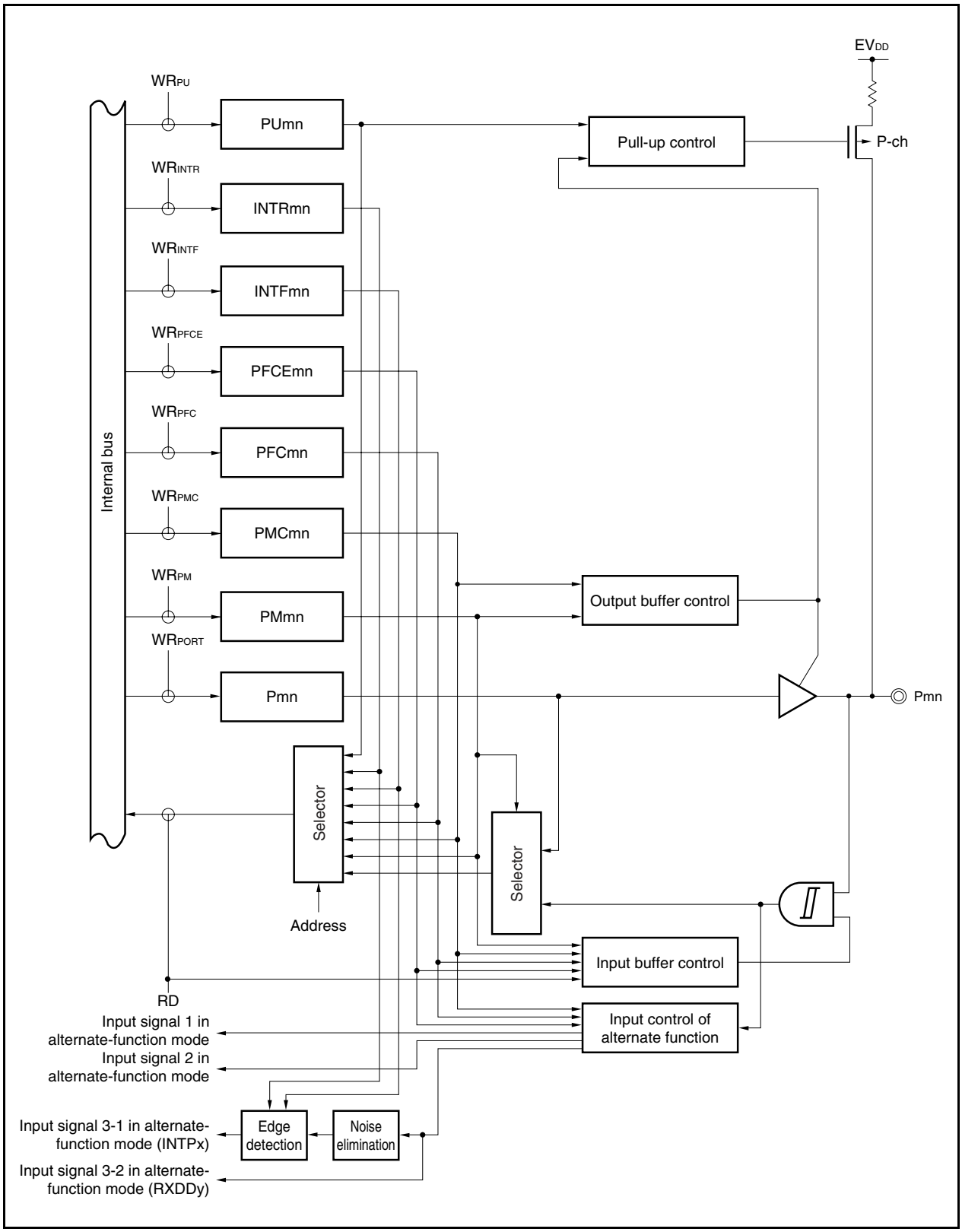


Figure 4-34. Block Diagram of Type Fx10x-U

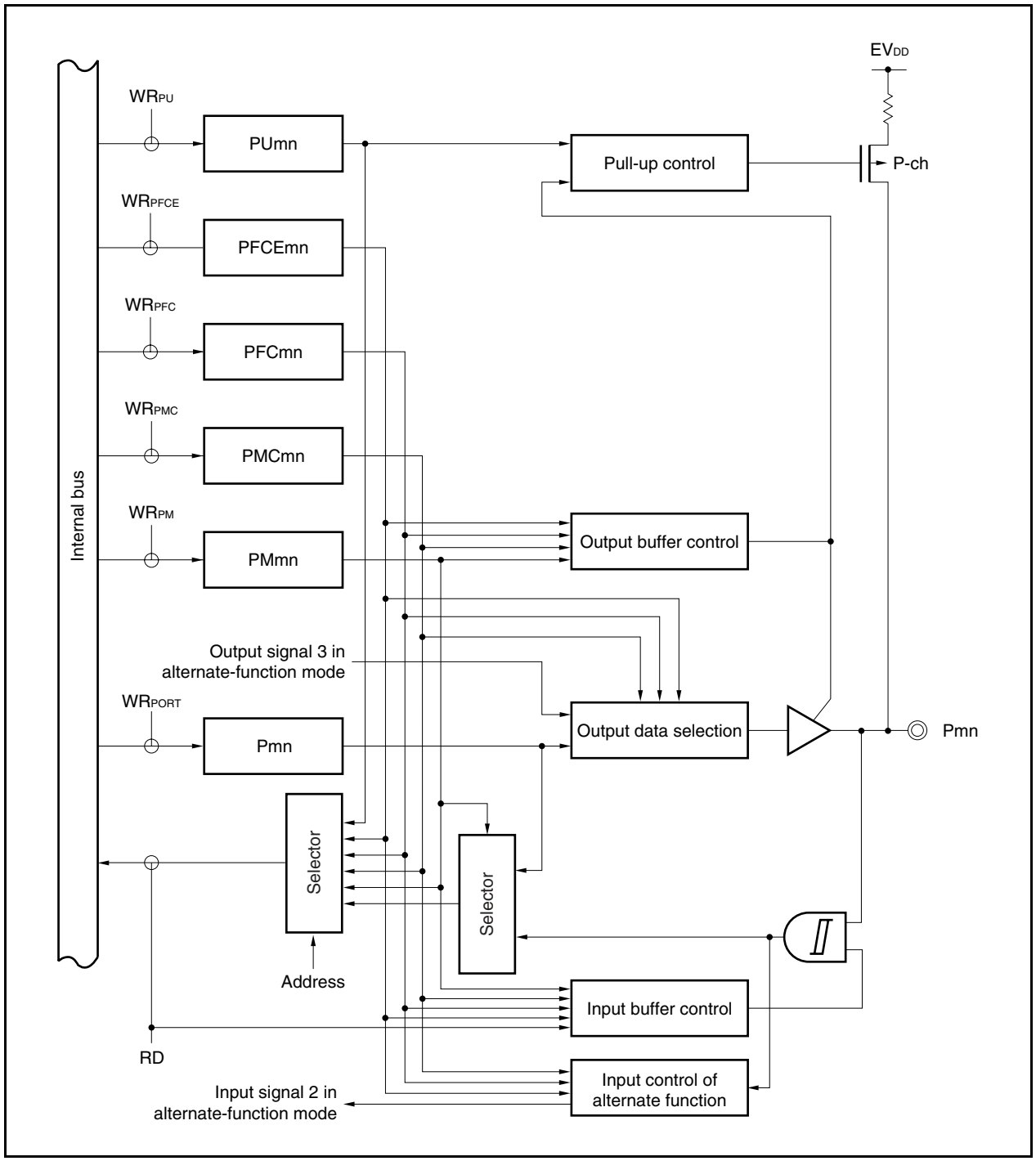


Figure 4-35. Block Diagram of Type Fx10x-UI

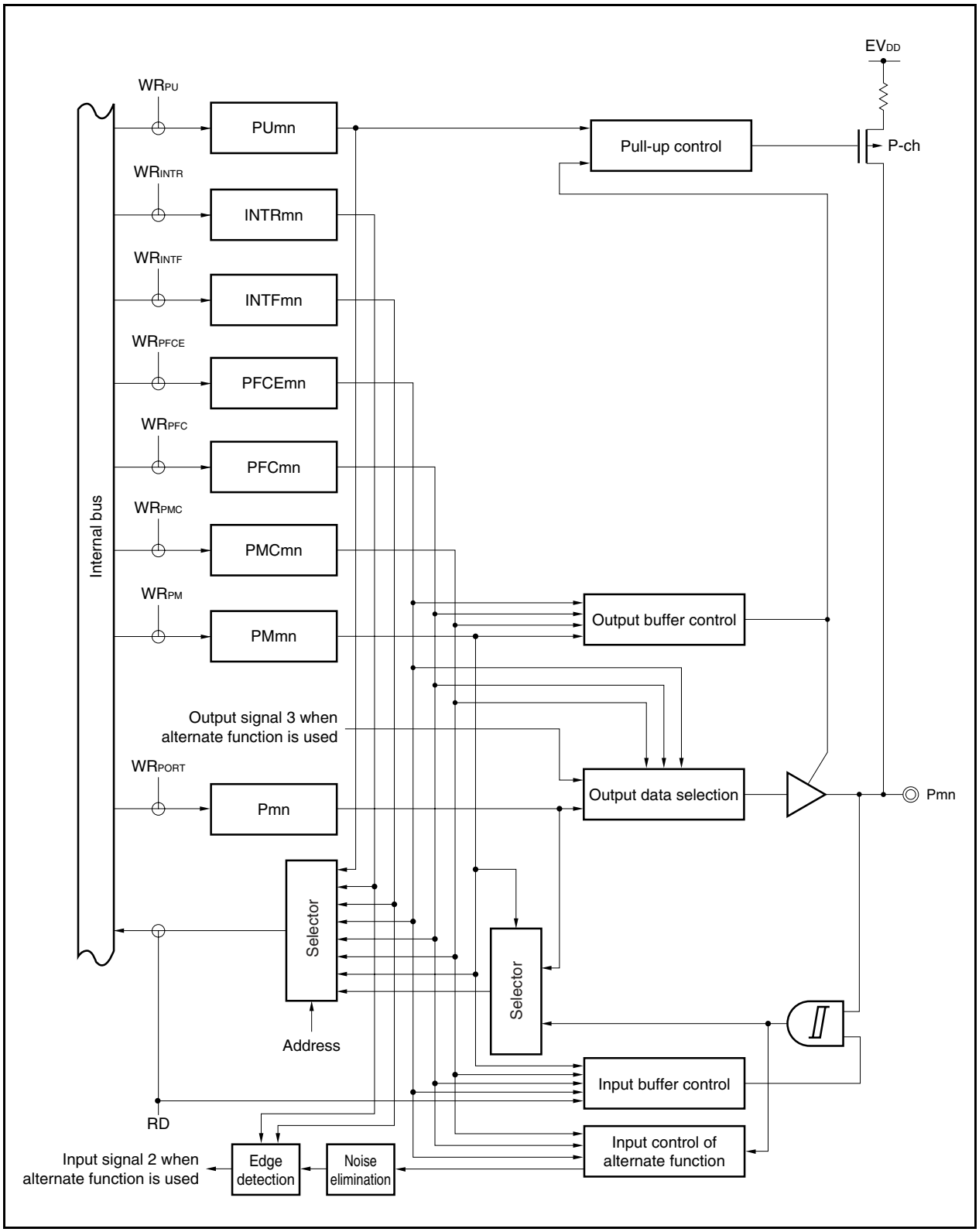


Figure 4-36. Block Diagram of Type Fx010-U

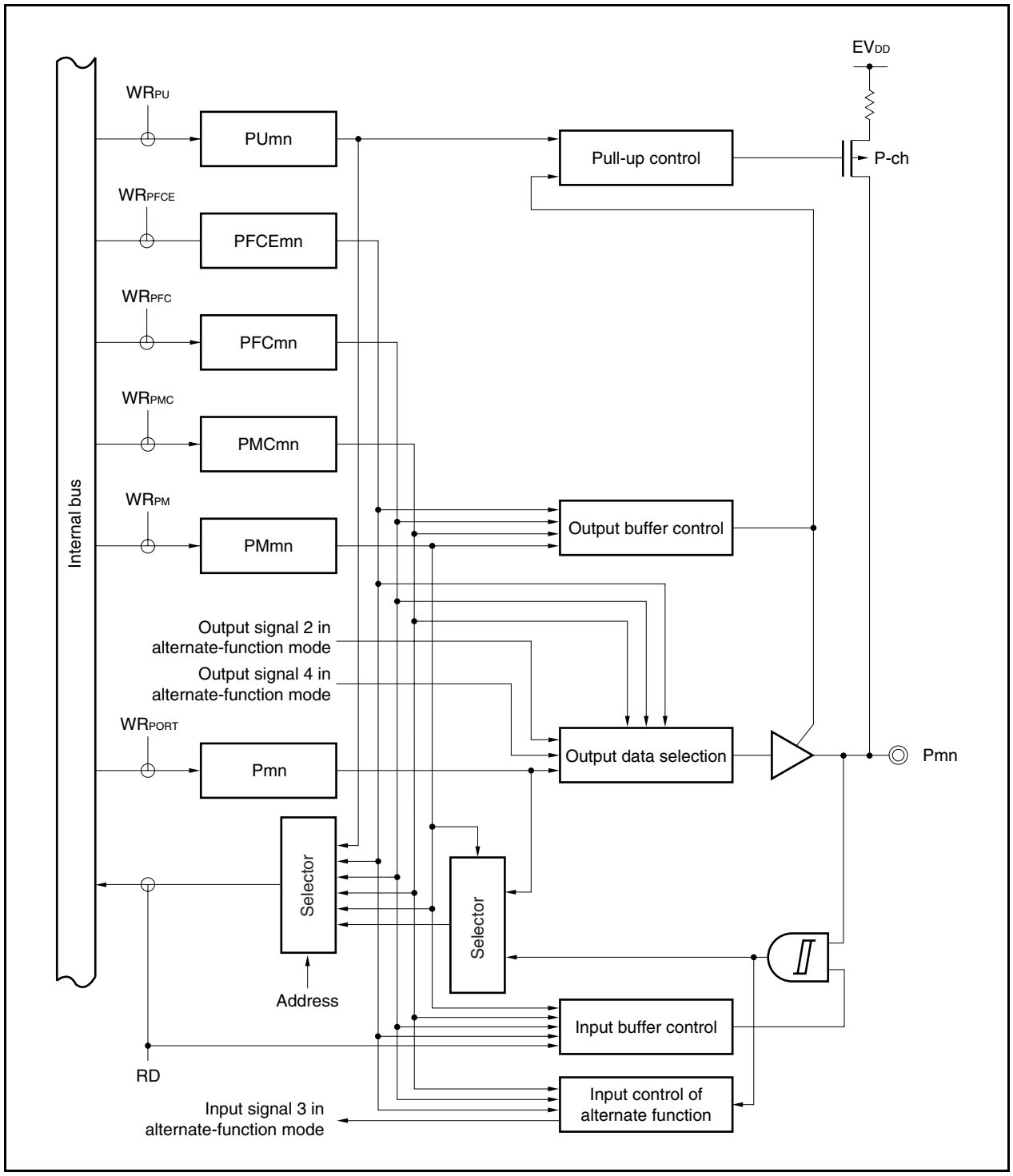


Figure 4-37. Block Diagram of Type Fx12x-UFI

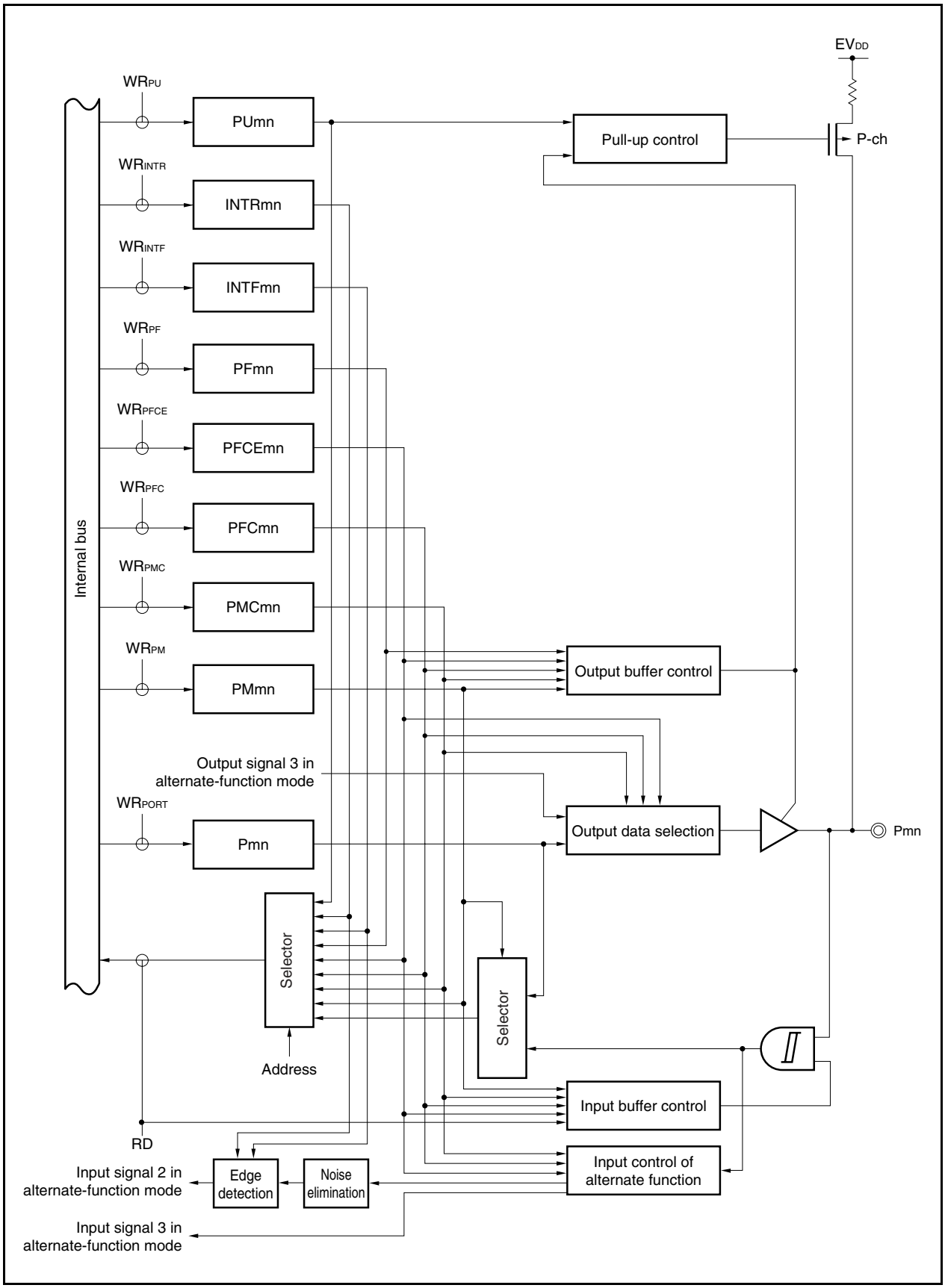


Figure 4-38. Block Diagram of Type Fx13x-U

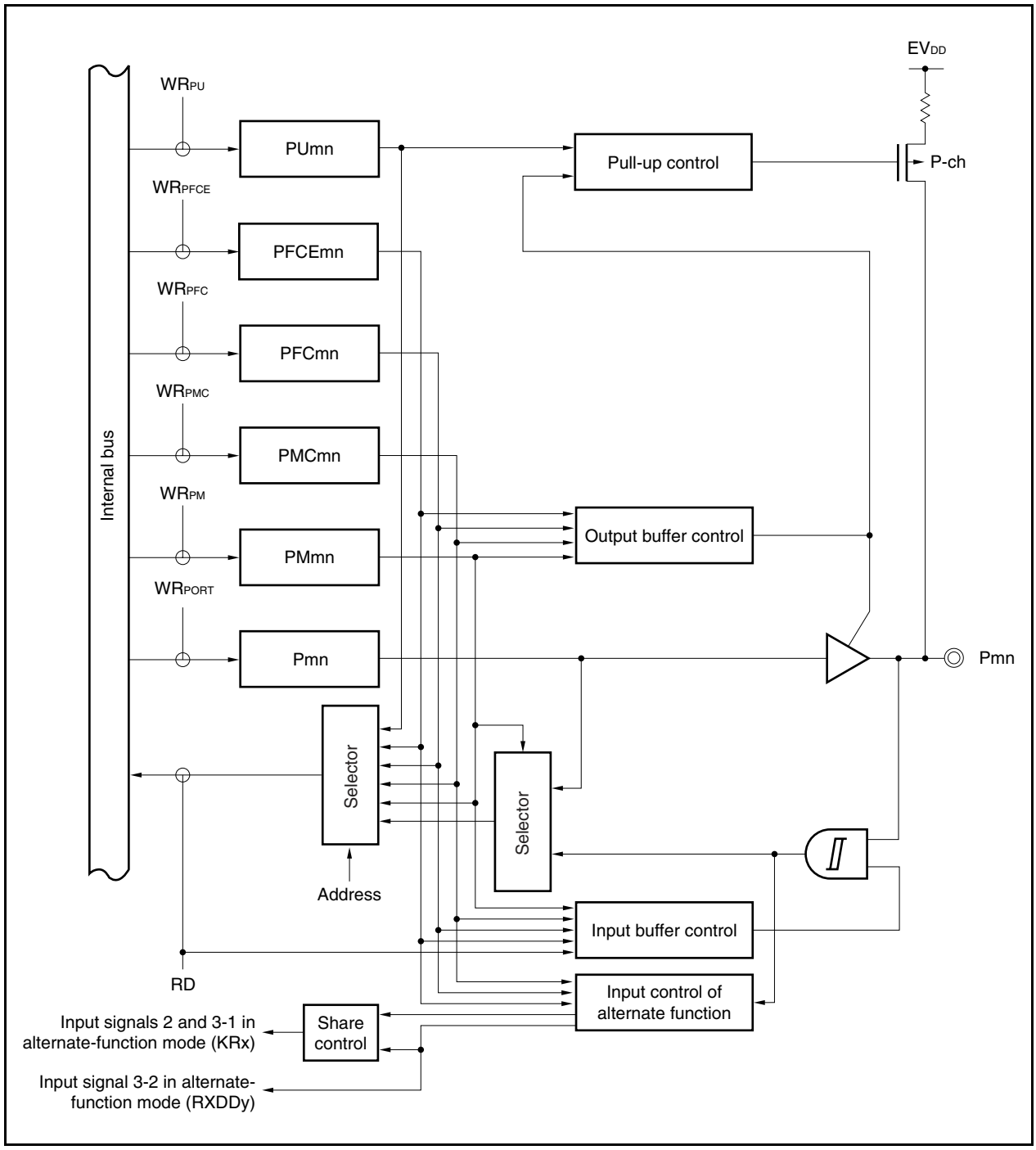


Figure 4-39. Block Diagram of Type Fx103-UI

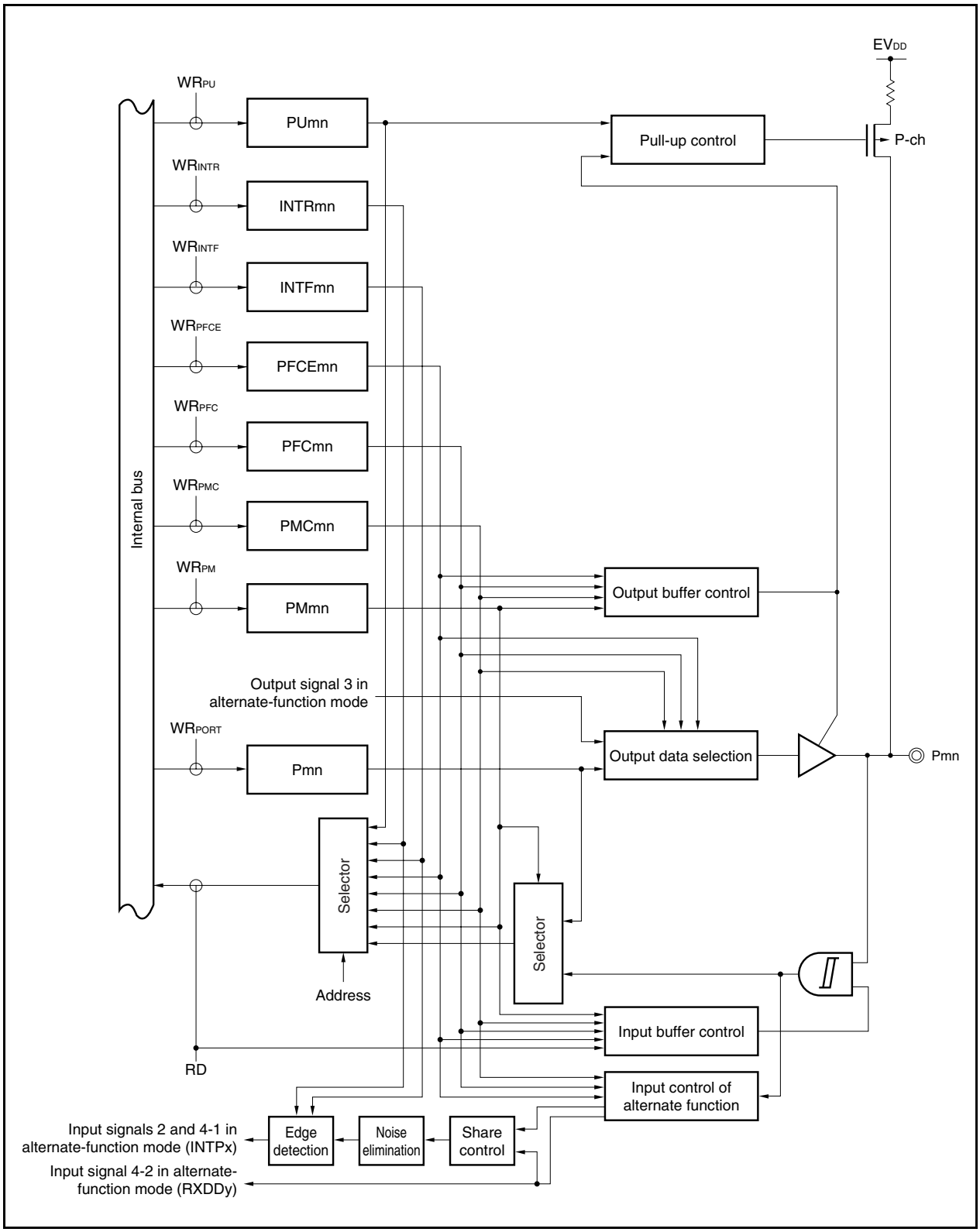


Figure 4-40. Block Diagram of Type Fx110-U

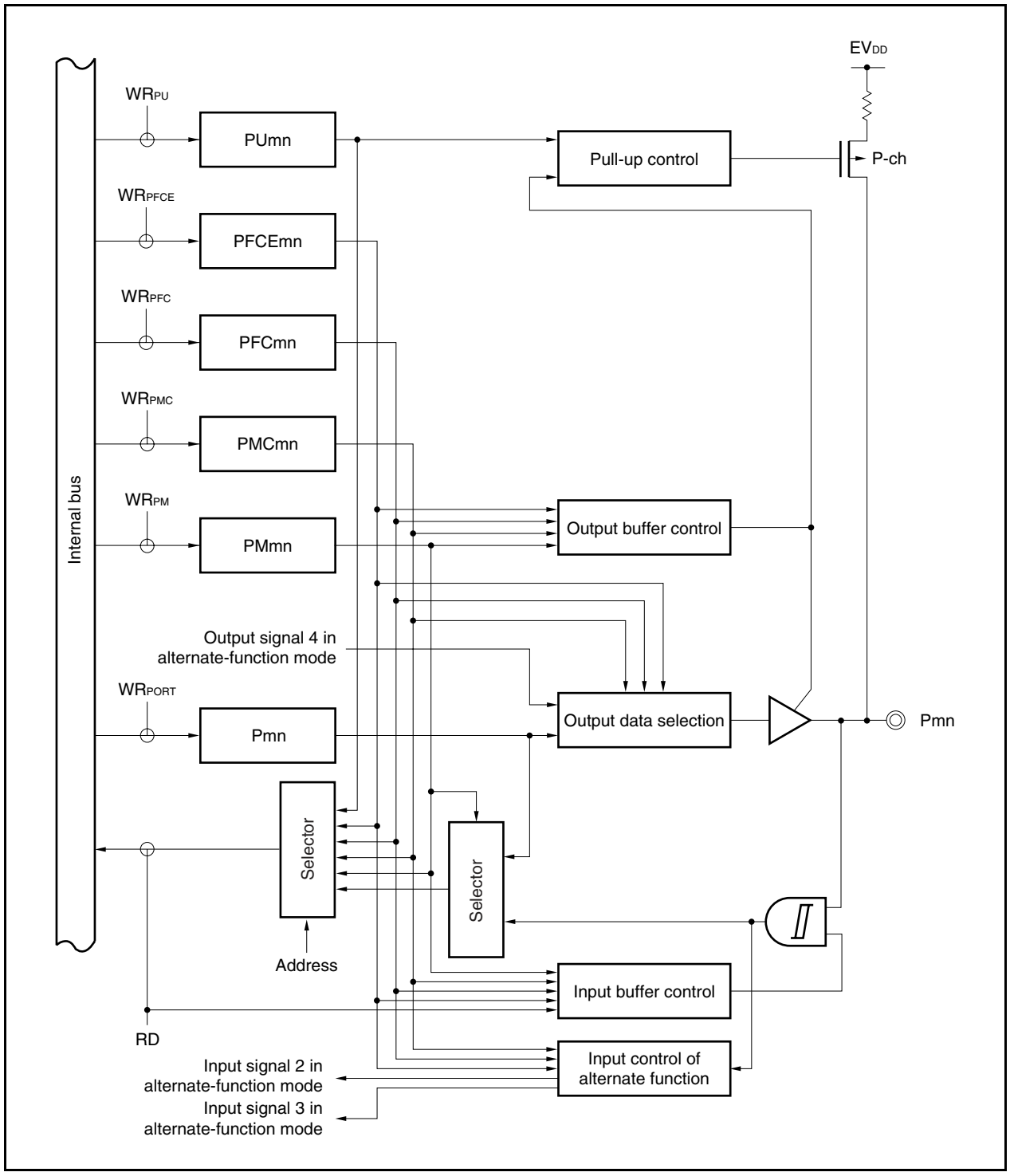


Figure 4-41. Block Diagram of Type Fx120-UFI

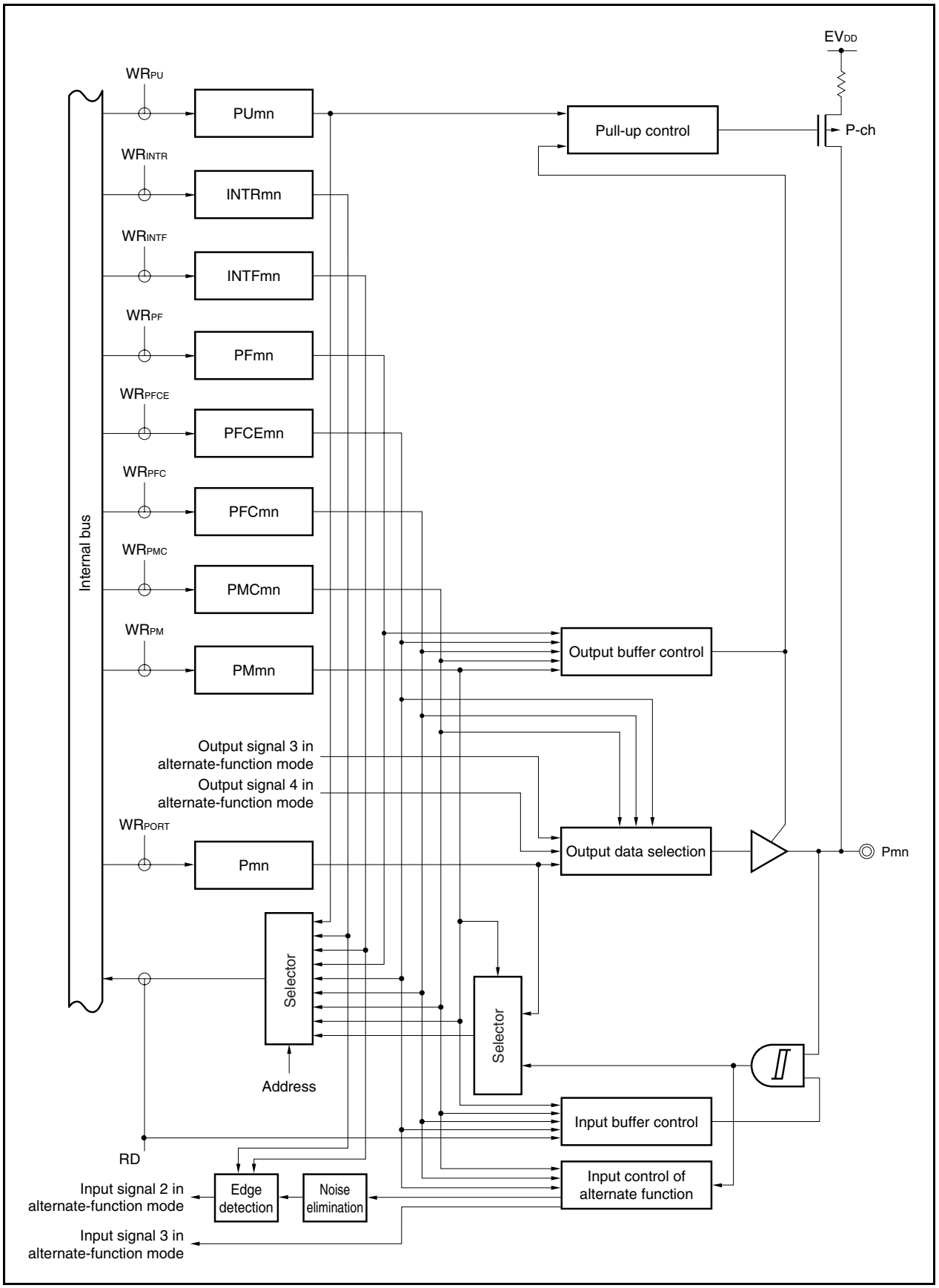


Figure 4-42. Block Diagram of Type Fx123-UFI

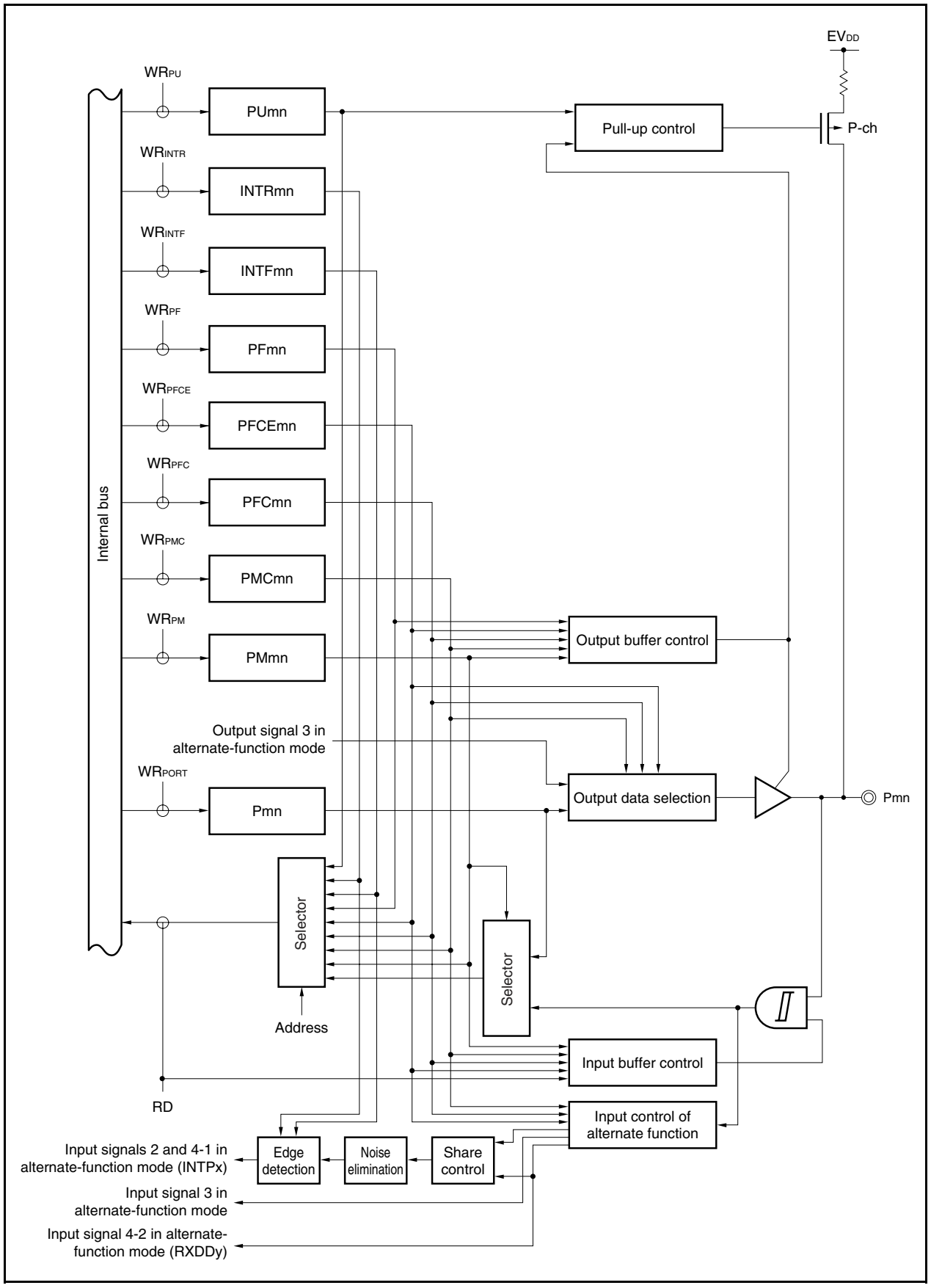


Figure 4-43. Block Diagram of Type Fx2x0-U

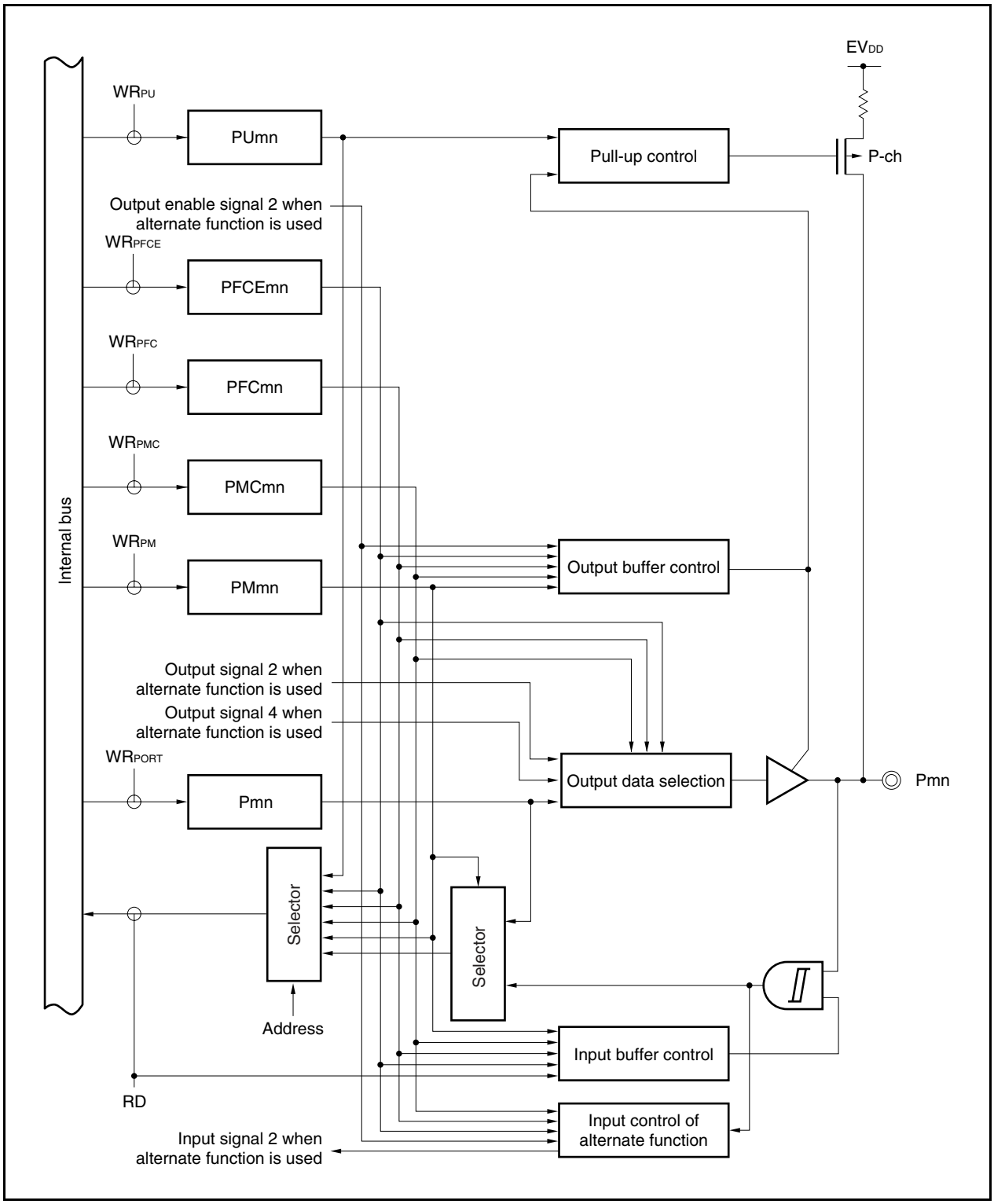


Figure 4-44. Block Diagram of Type Fx210-U

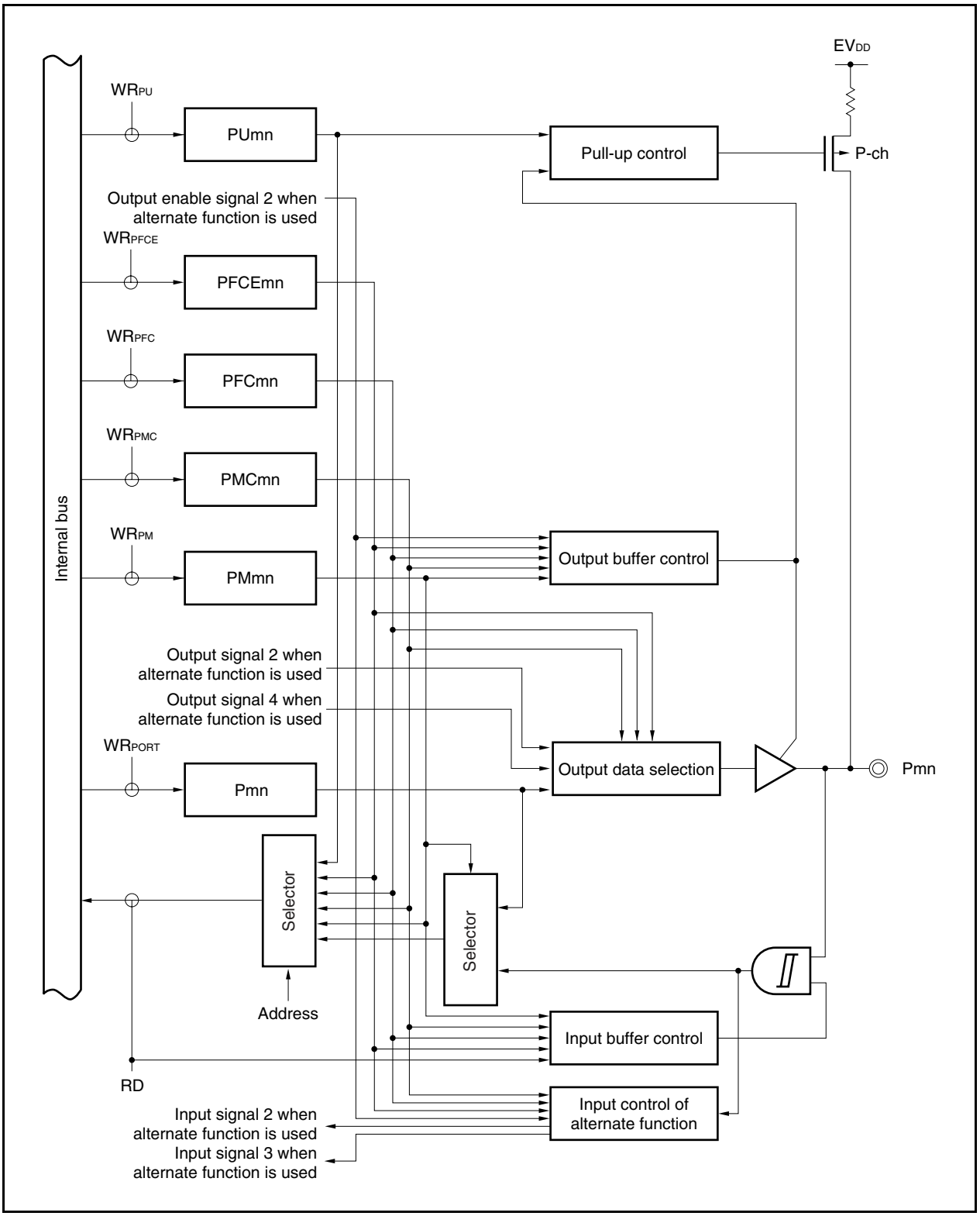
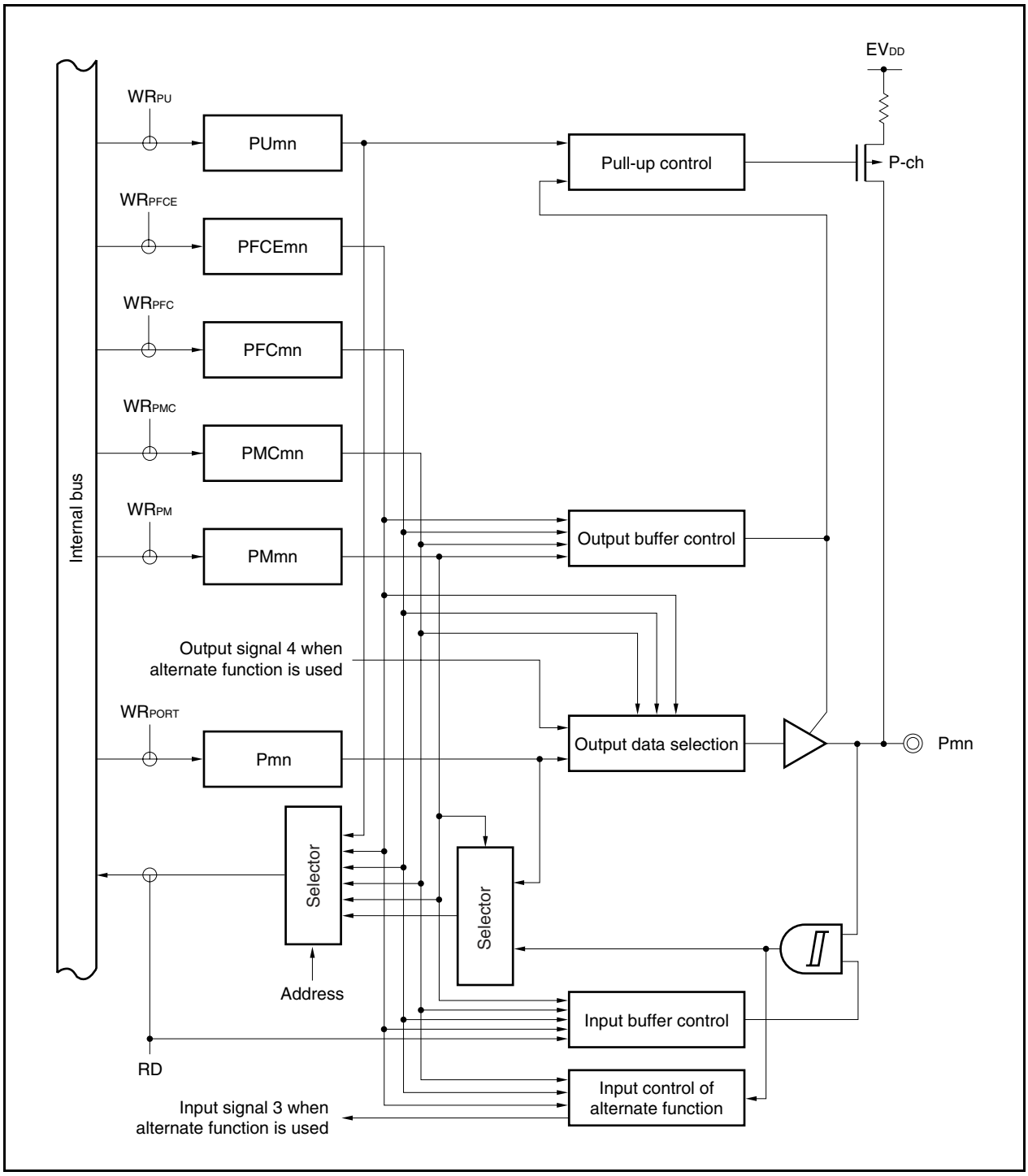


Figure 4-45. Block Diagram of Type Fxx10-U



4.5 Port Register Settings When Alternate Function Is Used

Table 4-25 shows the port register settings when each port is used for an alternate function.
When using a port pin as an alternate-function pin, refer to the description of each pin.

Table 4-25. Using Port Pin as Alternate-Function Pin (1/9)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)	HE3	HF3	HG3	HJ3
	Name	I/O										
P00	TIAA31	Input	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	–	PFC00 = 0		√	√	√	√
	TOAA31	Output	P00 = Setting not required	PM00 = Setting not required	PMC00 = 1	–	PFC00 = 1		√	√	√	√
P01	TIAA30	Input	P01 = Setting not required	PM01 = Setting not required	PMC01 = 1	–	PFC01 = 0		√	√	√	√
	TOAA30	Output	P01 = Setting not required	PM01 = Setting not required	PMC01 = 1	–	PFC01 = 1		√	√	√	√
P02	NMI	Input	P02 = Setting not required	PM02 = Setting not required	PMC02 = 1	PFCE02 = 0	PFC02 = 0		√	√	√	√
	TIAA40	Input	P02 = Setting not required	PM02 = Setting not required	PMC02 = 1	PFCE02 = 1	PFC02 = 0		√	√	√	√
	TOAA40	Output	P02 = Setting not required	PM02 = Setting not required	PMC02 = 1	PFCE02 = 0	PFC02 = 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		√	√	√	√
	TIAA41	Input	P03 = Setting not required	PM03 = Setting not required	PMC03 = 1	PFCE03 = 1	PFC03 = 0		√	√	√	√
	TOAA41	Output	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	–	–		√	√	√	√
P05	INTP2	Input	P05 = Setting not required	PM05 = Setting not required	PMC05 = 1	–	–		√	√	√	√
	$\overline{\text{DRST}}$	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	INTP9	Input	P10 = Setting not required	PM10 = Setting not required	PMC10 = 1	–	–		√	√	√	√
P11	INTP10	Input	P11 = Setting not required	PM11 = Setting not required	PMC11 = 1	–	–				√	√

Table 4-25. Using Port Pin as Alternate-Function Pin (2/9)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)	HE3	HF3	HG3	HJ3
P30	TXDD0	Output	P30 = Setting not required	PM30 = Setting not required	PMC30 = 1	–	–		√	√	√	√
P31	RXDD0	Input	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	–	**Note 1**		√	√	√	√
	INTP7	Input	P31 = Setting not required	PM31 = Setting not required	PMC31 = 1	–	**Note 1**		√	√	√	√
P32	ASCKD0	Input	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 0	PFC32 = 0		√	√	√	√
	TOAA01	Output	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 0	PFC32 = 1		√	√	√	√
	TIAA00	Input	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 1	PFC32 = 0		√	√	√	√
	TOAA00	Output	P32 = Setting not required	PM32 = Setting not required	PMC32 = 1	PFCE32 = 1	PFC32 = 1		√	√	√	√
P33	TIAA01	Input	P33 = Setting not required	PM33 = Setting not required	PMC33 = 1	–	PFC33 = 0		√	√	√	√
	TOAA01	Output	P33 = Setting not required	PM33 = Setting not required	PMC33 = 1	–	PFC33 = 1		√	√	√	√
P34	TIAA10	Input	P34 = Setting not required	PM34 = Setting not required	PMC34 = 1	–	PFC34 = 0		√	√	√	√
	TOAA10	Output	P34 = Setting not required	PM34 = Setting not required	PMC34 = 1	–	PFC34 = 1		√	√	√	√
P35	TIAA11	Input	P35 = Setting not required	PM35 = Setting not required	PMC35 = 1	–	PFC35 = 0		√	√	√	√
	TOAA11	Output	P35 = Setting not required	PM35 = Setting not required	PMC35 = 1	–	PFC35 = 1		√	√	√	√
P38	TXDD2	Output	P38 = Setting not required	PM38 = Setting not required	PMC38 = 1	–	–				√	√
P39	RXDD2	Input	P39 = Setting not required	PM39 = Setting not required	PMC39 = 1	–	**Note 2**				√	√
	INTP8	Input	P39 = Setting not required	PM39 = Setting not required	PMC39 = 1	–	**Note 2**				√	√

Notes 1. The INTP7 function and RXDD0 function are alternately used. When using as the RXDD0 function, disable edge detection for the INTP7 function (set the INTF3.INTF31 bit and the INTR3.INTR31 bit to 0). When using as the INTP7 function, stop UARTD0 reception (set the UD0CTL0.UD0RXE bit to 0).

2. The INTP8 function and RXDD2 function are alternately used. When using as the RXDD2 function, disable edge detection for the INTP8 function (set the INTF3.INTF39 bit and the INTR3.INTR39 bit to 0). When using as the INTP8 function, stop UARTD2 reception (set the UD2CTL0.UD2RXE bit to 0).

Table 4-25. Using Port Pin as Alternate-Function Pin (3/9)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)	HE3	HF3	HG3	HJ3
P40	SIB0	Input	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	PFCE40 = 0	PFC40 = 0		√	√	√	√
	KR0	Input	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	PFCE40 = 0	PFC40 = 1		√	√	√	√
	RXDD3	Input	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	PFCE40 = 1	**Note 1**, PFC40 = 0					Note 2
	INTP14	Input	P40 = Setting not required	PM40 = Setting not required	PMC40 = 1	PFCE40 = 1	**Note 1**, PFC40 = 0					Note 2
P41	SOB0	Output	P41 = Setting not required	PM41 = Setting not required	PMC41 = 1	PFCE41 = 0	PFC41 = 0		√	√	√	√
	KR1	Input	P41 = Setting not required	PM41 = Setting not required	PMC41 = 1	PFCE41 = 0	PFC41 = 1		√	√	√	√
	TXDD3	Output	P41 = Setting not required	PM41 = Setting not required	PMC41 = 1	PFCE41 = 1	PFC41 = 0					Note 2
P42	$\overline{\text{SCKB0}}$	I/O	P42 = Setting not required	PM42 = Setting not required	PMC42 = 1	–	PFC42 = 0		√	√	√	√
	KR2	Input	P42 = Setting not required	PM42 = Setting not required	PMC42 = 1	–	PFC42 = 1		√	√	√	√
P50	KR0	Input	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 0	PFC50 = 0		√	√	√	√
	TIAB01	Input	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 0	PFC50 = 1		√	√	√	√
	TOAB01	Output	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 1	PFC50 = 0		√	√	√	√
	TOAB0T1	Output	P50 = Setting not required	PM50 = Setting not required	PMC50 = 1	PFCE50 = 1	PFC50 = 1		√	√	√	√
P51	KR1	Input	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 0	PFC51 = 0		√	√	√	√
	TIAB02	Input	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 0	PFC51 = 1		√	√	√	√
	TOAB02	Output	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 1	PFC51 = 0		√	√	√	√
	TOAB0B1	Output	P51 = Setting not required	PM51 = Setting not required	PMC51 = 1	PFCE51 = 1	PFC51 = 1		√	√	√	√
P52	KR2	Input	P52 = Setting not required	PM52 = Setting not required	PMC52 = 1	PFCE52 = 0	PFC52 = 0		√	√	√	√
	TIAB03	Input	P52 = Setting not required	PM52 = Setting not required	PMC52 = 1	PFCE52 = 0	PFC52 = 1		√	√	√	√
	TOAB03	Output	P52 = Setting not required	PM52 = Setting not required	PMC52 = 1	PFCE52 = 1	PFC52 = 0		√	√	√	√
	TOAB0T2	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	PFCE52 = Setting not required	PFC52 = Setting not required	OCDM0 (OCDM) = 1	√	√	√	√

Notes 1. The INTP14 function and RXDD3 function are alternately used. When using as the RXDD3 function, disable edge detection for the INTP14 function (set the INTF4.INTF40 bit and the INTR4.INTR40 bit to 0). When using as the INTP14 function, stop UARTD3 reception (set the UD3CTL0.UD3RXE bit to 0).

2. μPD70F3757 only

Table 4-25. Using Port Pin as Alternate Function-Pin (4/9)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)	HE3	HF3	HG3	HJ3
P53	KR3	Input	P53 = Setting not required	PM53 = Setting not required	PMC53 = 1	PFCE53 = 0	PFC53 = 0		√	√	√	√
	TIAB00	Input	P53 = Setting not required	PM53 = Setting not required	PMC53 = 1	PFCE53 = 0	PFC53 = 1		√	√	√	√
	TOAB00	Output	P53 = Setting not required	PM53 = Setting not required	PMC53 = 1	PFCE53 = 1	PFC53 = 0		√	√	√	√
	TOAB0B2	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	KR4	Input	P54 = Setting not required	PM54 = Setting not required	PMC54 = 1	PFCE54 = 0	PFC54 = 0		√	√	√	√
	TOAB0T3	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	KR5	Input	P55 = Setting not required	PM55 = Setting not required	PMC55 = 1	PFCE55 = 0	PFC55 = 0		√	√	√	√
	TOAB0B3	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	√	√	√	√
P60	INTP11	Input	P60 = Setting not required	PM60 = Setting not required	PMC60 = 1	–	PFC60 = 1					√
P61	INTP12	Input	P61 = Setting not required	PM61 = Setting not required	PMC61 = 1	–	PFC61 = 1					√
P62	INTP13	Input	P62 = Setting not required	PM62 = Setting not required	PMC62 = 1	–	PFC62 = 1					√
P610	TIAB20	Input	P610 = Setting not required	PM610 = Setting not required	PMC610 = 1	–	PFC610 = 0					√
	TOAB20	Output	P610 = Setting not required	PM610 = Setting not required	PMC610 = 1	–	PFC610 = 1					√
P611	TIAB21	Input	P611 = Setting not required	PM611 = Setting not required	PMC611 = 1	–	PFC611 = 0					√
	TOAB21	Output	P611 = Setting not required	PM611 = Setting not required	PMC611 = 1	–	PFC611 = 1					√
P612	TIAB22	Input	P612 = Setting not required	PM612 = Setting not required	PMC612 = 1	–	PFC612 = 0					√
	TOAB22	Output	P612 = Setting not required	PM612 = Setting not required	PMC612 = 1	–	PFC612 = 1					√
P613	TIAB23	Input	P613 = Setting not required	PM613 = Setting not required	PMC613 = 1	–	PFC613 = 0					√
	TOAB23	Output	P613 = Setting not required	PM613 = Setting not required	PMC613 = 1	–	PFC613 = 1					√

Table 4-25. Using Port Pin as Alternate Function-Pin (5/9)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)	HE3	HF3	HG3	HJ3
P70	ANI0	Input	P70 = Setting not required	PM70 = Setting not required	PMC70 = 1	–	–		√	√	√	√
P71	ANI1	Input	P71 = Setting not required	PM71 = Setting not required	PMC71 = 1	–	–		√	√	√	√
P72	ANI2	Input	P72 = Setting not required	PM72 = Setting not required	PMC72 = 1	–	–		√	√	√	√
P73	ANI3	Input	P73 = Setting not required	PM73 = Setting not required	PMC73 = 1	–	–		√	√	√	√
P74	ANI4	Input	P74 = Setting not required	PM74 = Setting not required	PMC74 = 1	–	–		√	√	√	√
P75	ANI5	Input	P75 = Setting not required	PM75 = Setting not required	PMC75 = 1	–	–		√	√	√	√
P76	ANI6	Input	P76 = Setting not required	PM76 = Setting not required	PMC76 = 1	–	–		√	√	√	√
P77	ANI7	Input	P77 = Setting not required	PM77 = Setting not required	PMC77 = 1	–	–		√	√	√	√
P78	ANI8	Input	P78 = Setting not required	PM78 = Setting not required	PMC78 = 1	–	–		√	√	√	√
P79	ANI9	Input	P79 = Setting not required	PM79 = Setting not required	PMC79 = 1	–	–		√	√	√	√
P710	ANI10	Input	P710 = Setting not required	PM710 = Setting not required	PMC710 = 1	–	–			√	√	√
P711	ANI11	Input	P711 = Setting not required	PM711 = Setting not required	PMC711 = 1	–	–			√	√	√
P712	ANI12	Input	P712 = Setting not required	PM712 = Setting not required	PMC712 = 1	–	–			√	√	√
P713	ANI13	Input	P713 = Setting not required	PM713 = Setting not required	PMC713 = 1	–	–				√	√
P714	ANI14	Input	P714 = Setting not required	PM714 = Setting not required	PMC714 = 1	–	–				√	√
P715	ANI15	Input	P715 = Setting not required	PM715 = Setting not required	PMC715 = 1	–	–				√	√
P80	RXDD3	Input	P80 = Setting not required	PM80 = Setting not required	PMC80 = 1	–	–					Note 2
	INTP14	Input	P80 = Setting not required	PM80 = Setting not required	PMC80 = 1	–	–					√
P81	TXDD3	Output	P81 = Setting not required	PM81 = Setting not required	PMC81 = 1	–	–					Note 2

Notes 1. The INTP14 function and RXDD3 function are alternately used. When using as the RXDD3 function, disable edge detection for the INTP14 function (set the INTF8.INTF80 bit and the INTR8.INTR80 bit to 0). When using as the INTP14 function, stop UARTD3 reception (set the UD3CTL0.UD3RXE bit to 0).

2. μPD70F3757 only

Table 4-25. Using Port Pin as Alternate-Function Pin (6/9)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)	HE3	HF3	HG3	HJ3
P90	KR6	Input	P90 = Setting not required	PM90 = Setting not required	PMC90 = 1	PFCE90 = 0	PFC90 = 1		√	√	√	√
	TXDD1	Output	P90 = Setting not required	PM90 = Setting not required	PMC90 = 1	PFCE90 = 1	PFC90 = 0		√	√	√	√
P91	KR7	Input	P91 = Setting not required	PM91 = Setting not required	PMC91 = 1	PFCE91 = 0	PFC91 = 1		√	√	√	√
	RXDD1/ KR7[Note]	Input	P91 = Setting not required	PM91 = Setting not required	PMC91 = 1	PFCE91 = 1	PFC91 = 0		√	√	√	√
P92	TIAB11	Input	P92 = Setting not required	PM92 = Setting not required	PMC92 = 1	PFCE92 = 0	PFC92 = 1				√	√
	TOAB11	Output	P92 = Setting not required	PM92 = Setting not required	PMC92 = 1	PFCE92 = 1	PFC92 = 0				√	√
P93	TIAB12	Input	P93 = Setting not required	PM93 = Setting not required	PMC93 = 1	PFCE93 = 0	PFC93 = 1				√	√
	TOAB12	Output	P93 = Setting not required	PM93 = Setting not required	PMC93 = 1	PFCE93 = 1	PFC93 = 0				√	√
P94	TIAB13	Input	P94 = Setting not required	PM94 = Setting not required	PMC94 = 1	PFCE94 = 0	PFC94 = 1				√	√
	TOAB13	Output	P94 = Setting not required	PM94 = Setting not required	PMC94 = 1	PFCE94 = 1	PFC94 = 0				√	√
P95	TIAB10	Input	P95 = Setting not required	PM95 = Setting not required	PMC95 = 1	PFCE95 = 0	PFC95 = 1				√	√
	TOAB10	Output	P95 = Setting not required	PM95 = Setting not required	PMC95 = 1	PFCE95 = 1	PFC95 = 0				√	√
P96	TIAA21	Input	P96 = Setting not required	PM96 = Setting not required	PMC96 = 1	PFCE96 = 1	PFC96 = 0		√	√	√	√
	TOAA21	Output	P96 = Setting not required	PM96 = Setting not required	PMC96 = 1	PFCE96 = 1	PFC96 = 1		√	√	√	√
P97	SIB1	Input	P97 = Setting not required	PM97 = Setting not required	PMC97 = 1	PFCE97 = 0	PFC97 = 1		√	√	√	√
	TIAA20	Input	P97 = Setting not required	PM97 = Setting not required	PMC97 = 1	PFCE97 = 1	PFC97 = 0		√	√	√	√
	TOAA20	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	PFCE98 = 0	PFC98 = 1		√	√	√	√
	TIAB03	Input	P98 = Setting not required	PM98 = Setting not required	PMC98 = 1	PFCE98 = 1	PFC98 = 0		√	√	√	√
	TOAB03	Output	P98 = Setting not required	PM98 = Setting not required	PMC98 = 1	PFCE98 = 1	PFC98 = 1		√	√	√	√
P99	$\overline{\text{SCKB1}}$	I/O	P99 = Setting not required	PM99 = Setting not required	PMC99 = 1	PFCE99 = 0	PFC99 = 1		√	√	√	√
	TIAB00	Input	P99 = Setting not required	PM99 = Setting not required	PMC99 = 1	PFCE99 = 1	PFC99 = 0		√	√	√	√
	TOAB00	Output	P99 = Setting not required	PM99 = Setting not required	PMC99 = 1	PFCE99 = 1	PFC99 = 1		√	√	√	√

Note The RXDD1 and KR7 functions must not be used at the same time. When using the RXDD1 function, do not use the KR7 function. When using the KR7 function, do not use the RXDD1 function (it is recommended to set the PFC91 bit to 1 and set the PFCE91 bit to 0).

Table 4-25. Using Port Pin as Alternate-Function Pin (7/9)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)	HE3	HF3	HG3	HJ3
P910	SIB2	Input	P910 = Setting not required	PM910 = Setting not required	PMC910 = 1	–	PFC910 = 1					√
P911	SOB2	Output	P911 = Setting not required	PM911 = Setting not required	PMC911 = 1	–	PFC911 = 1					√
P912	$\overline{\text{SCKB2}}$	I/O	P912 = Setting not required	PM912 = Setting not required	PMC912 = 1	PFCE912 = 0	PFC912 = 1					√
	TXDD5	Output	P912 = Setting not required	PM912 = Setting not required	PMC912 = 1	PFCE912 = 1	PFC912 = 1					Note 3
P913	INTP4	Input	P913 = Setting not required	PM913 = Setting not required	PMC913 = 1	PFCE913 = 0	PFC913 = 1		√	√	√	√
	PCL	Output	P913 = Setting not required	PM913 = Setting not required	PMC913 = 1	PFCE913 = 1	PFC913 = 0		√	√	√	√
	RXDD5	Input	P913 = Setting not required	PM913 = Setting not required	PMC913 = 1	**Note 1**, PFCE913 = 1	**Note 1**, PFC913 = 1					Note 3
	INTP4	Input	P913 = Setting not required	PM913 = Setting not required	PMC913 = 1	**Note 1**, PFCE913 = 1	**Note 1**, PFC913 = 1					Note 3
P914	INTP5	Input	P914 = Setting not required	PM914 = Setting not required	PMC914 = 1	PFCE914 = 0	PFC914 = 1		√	√	√	√
	SDA00	I/O	P914 = Setting not required	PM914 = Setting not required	PMC914 = 1	PFCE914 = 1	PFC914 = 0	PF914 = 1	√	√	√	√
	RXDD4	Input	P914 = Setting not required	PM914 = Setting not required	PMC914 = 1	**Note 2**, PFCE914 = 1	**Note 2**, PFC914 = 1					Note 3
	INTP5	Input	P914 = Setting not required	PM914 = Setting not required	PMC914 = 1	**Note 2**, PFCE914 = 1	**Note 2**, PFC914 = 1					Note 3
P915	INTP6	Input	P915 = Setting not required	PM915 = Setting not required	PMC915 = 1	PFCE915 = 0	PFC915 = 1		√	√	√	√
	SCL00	I/O	P915 = Setting not required	PM915 = Setting not required	PMC915 = 1	PFCE915 = 1	PFC915 = 0	PF915 = 1	√	√	√	√
	TXDD4	Output	P915 = Setting not required	PM915 = Setting not required	PMC915 = 1	PFCE915 = 1	PFC915 = 1					Note 3

Notes 1. The INTP4 function and RXDD5 function are alternately used.
When using as the RXDD5 function, set the valid edge specification of the INTP4 function to “No edge detected” (set the INTF9H.INTF913 bit and the INTR9H.INTR913 bit to “00”). Also, when using as the INTP4 function, it is recommended to set the PFC913 bit to 1 and set the PFCE913 bit to 0.

2. The INTP5 function and RXDD4 function are alternately used.
When using as the RXDD4 function, set the valid edge specification of the INTP5 function to “No edge detected” (set the INTF9H.INTF914 bit and the INTR9H.INTR914 bit to “00”). Also, when using as the INTP5 function, it is recommended to set the PFC914 bit to 1 and set the PFCE914 bit to 0.

3. μPD70F3757 only

Table 4-25. Using Port Pin as Alternate-Function Pin (8/9)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)	HE3	HF3	HG3	HJ3
P120	ANI16	Input	P120 = Setting not required	PM120 = Setting not required	PMC120 = 1	–	–					√
P121	ANI17	Input	P121 = Setting not required	PM121 = Setting not required	PMC121 = 1	–	–					√
P122	ANI18	Input	P122 = Setting not required	PM122 = Setting not required	PMC122 = 1	–	–					√
P123	ANI19	Input	P123 = Setting not required	PM123 = Setting not required	PMC123 = 1	–	–					√
P124	ANI20	Input	P124 = Setting not required	PM124 = Setting not required	PMC124 = 1	–	–					√
P125	ANI21	Input	P125 = Setting not required	PM125 = Setting not required	PMC125 = 1	–	–					√
P126	ANI22	Input	P126 = Setting not required	PM126 = Setting not required	PMC126 = 1	–	–					√
P127	ANI23	Input	P127 = Setting not required	PM127 = Setting not required	PMC127 = 1	–	–					√
PCM0	$\overline{\text{WAIT}}$	Input	PCM0 = Setting not required	PMCM0 = Setting not required	PMCCM0 = 1	–	–					√
PCM1	CLKOUT	Output	PCM1 = Setting not required	PMCM1 = Setting not required	PMCCM1 = 1	–	–		√	√	√	√
PCM2	$\overline{\text{HLDAK}}$	Output	PCM2 = Setting not required	PMCM2 = Setting not required	PMCCM2 = 1	–	–					√
PCM3	$\overline{\text{HLDRQ}}$	Input	PCM3 = Setting not required	PMCM3 = Setting not required	PMCCM3 = 1	–	–					√
PCS0	$\overline{\text{CS0}}$	Output	PCS0 = Setting not required	PMCS0 = Setting not required	PMCCS0 = 1	–	–					√
PCS1	$\overline{\text{CS1}}$	Output	PCS1 = Setting not required	PMCS1 = Setting not required	PMCCS1 = 1	–	–					√
PCS2	$\overline{\text{CS2}}$	Output	PCS2 = Setting not required	PMCS2 = Setting not required	PMCCS2 = 1	–	–					√
PCS3	$\overline{\text{CS3}}$	Output	PCS3 = Setting not required	PMCS3 = Setting not required	PMCCS3 = 1	–	–					√
PCT0	$\overline{\text{WR0}}$	Output	PCT0 = Setting not required	PMCT0 = Setting not required	PMCCT0 = 1	–	–					√
PCT1	$\overline{\text{WR1}}$	Output	PCT1 = Setting not required	PMCT1 = Setting not required	PMCCT1 = 1	–	–					√
PCT4	$\overline{\text{RD}}$	Output	PCT4 = Setting not required	PMCT4 = Setting not required	PMCCT4 = 1	–	–					√
PCT6	ASTB	Output	PCT6 = Setting not required	PMCT6 = Setting not required	PMCCT6 = 1	–	–					√

Table 4-25. Using Port Pin as Alternate-Function Pin (9/9)

Pin Name	Alternate Function		Pnx Bit of Pn Register	PMnx Bit of PMn Register	PMCnx Bit of PMCn Register	PFCEnx Bit of PFCEn Register	PFCnx Bit of PFCn Register	Other Bits (Registers)	HE3	HF3	HG3	HJ3
PDL0	AD0	I/O	PDL0 = Setting not required	PMDL0 = Setting not required	PMCDL0 = 1	–	–					√
PDL1	AD1	I/O	PDL1 = Setting not required	PMDL1 = Setting not required	PMCDL1 = 1	–	–					√
PDL2	AD2	I/O	PDL2 = Setting not required	PMDL2 = Setting not required	PMCDL2 = 1	–	–					√
PDL3	AD3	I/O	PDL3 = Setting not required	PMDL3 = Setting not required	PMCDL3 = 1	–	–					√
PDL4	AD4	I/O	PDL4 = Setting not required	PMDL4 = Setting not required	PMCDL4 = 1	–	–					√
PDL5	AD5	I/O	PDL5 = Setting not required	PMDL5 = Setting not required	PMCDL5 = 1	–	–					√
	FLMD1[Note]	Input	PDL5 = Setting not required	PMDL5 = Setting not required	PMCDL5 = Setting not required	–	–					√
PDL6	AD6	I/O	PDL6 = Setting not required	PMDL6 = Setting not required	PMCDL6 = 1	–	–					√
PDL7	AD7	I/O	PDL7 = Setting not required	PMDL7 = Setting not required	PMCDL7 = 1	–	–					√
PDL8	AD8	I/O	PDL8 = Setting not required	PMDL8 = Setting not required	PMCDL8 = 1	–	–					√
PDL9	AD9	I/O	PDL9 = Setting not required	PMDL9 = Setting not required	PMCDL9 = 1	–	–					√
PDL10	AD10	I/O	PDL10 = Setting not required	PMDL10 = Setting not required	PMCDL10 = 1	–	–					√
PDL11	AD11	I/O	PDL11 = Setting not required	PMDL11 = Setting not required	PMCDL11 = 1	–	–					√
PDL12	AD12	I/O	PDL12 = Setting not required	PMDL12 = Setting not required	PMCDL12 = 1	–	–					√
PDL13	AD13	I/O	PDL13 = Setting not required	PMDL13 = Setting not required	PMCDL13 = 1	–	–					√
PDL14	AD14	I/O	PDL14 = Setting not required	PMDL14 = Setting not required	PMCDL14 = 1	–	–					√
PDL15	AD15	I/O	PDL15 = Setting not required	PMDL15 = Setting not required	PMCDL15 = 1	–	–					√

Note Since this pin is set in the flash memory programming mode, it does not need to be manipulated using the port control register. For details, see **CHAPTER 26 FLASH MEMORY**.

4.6 Cautions

4.6.1 Cautions on setting port pins

(1) In the V850ES/Hx3, the general-purpose port function and several peripheral function I/O pin share a pin. To switch between the general-purpose port (port mode) and the peripheral function I/O pin (alternate-function mode), set by the PMCn register. In regards to this register setting sequence, note with caution the following.

(a) Cautions on switching from port mode to alternate-function mode
To switch from the port mode to alternate-function mode in the following order.

<1> Set the PFCn and PFCEn registers: Alternate-function selection
<2> Set the corresponding bit of the PMCn register to 1: Switch to alternate-function mode
<3> Set the PUn register[Note]: On-chip pull-up resistor connection setting

If the PMCn register is set first, note with caution that, at that moment or depending on the change of the pin states in accordance with the setting of the PFn, PFCn, and PFCEn registers, unexpected operations may occur.

Note Corresponding input pin

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

- **Pn register read: Read the port output latch value (when PMn.PMnm bit = 0), or read the pin states (PMn.PMnm bit = 1).**
- **Pn register write: Write to the port output latch**

Set the N-ch open-drain pin in the following order.

To use the port mode register
<1> PMCnm bit = 0
<2> PFnm bit = 1

To use the alternate function (I^2C)
<1> Set the PFCn and PFCEn registers
<2> PFnm bit = 1
<3> PMCnm bit = 1

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

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

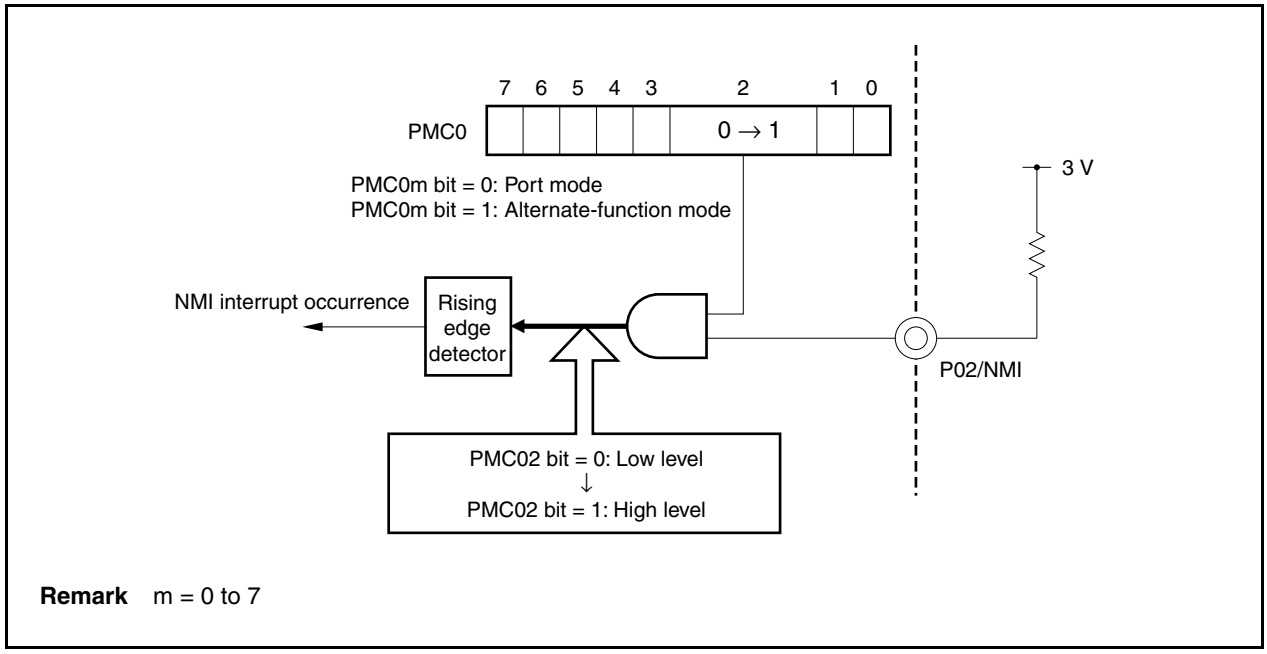
- To switch from port mode to alternate-function mode (input)
  Set the pins to the alternate-function mode using the PMCn register and then enable the alternate-function operation.
- To switch from alternate-function mode (input) to port mode
  Stop the alternate-function operation and then switch the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

The concrete examples are shown as Example 1 and Example 2.

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

When the P02/NMI pin is pulled up as shown in Figure 4-46 and the rising edge is specified in the NMI pin edge detection setting, even though high level is input continuously to the NMI pin during switching from the P02 pin to the an NMI pin (PMC02 bit = 0 → 1), this is detected as a rising edge as if the low level changed to high level, and an NMI interrupt occurs.
To avoid it, set the NMI pin's valid edge after switching from the P02 pin to the NMI pin.

Figure 4-46. Example of Switching from P02 to NMI (Incorrect)



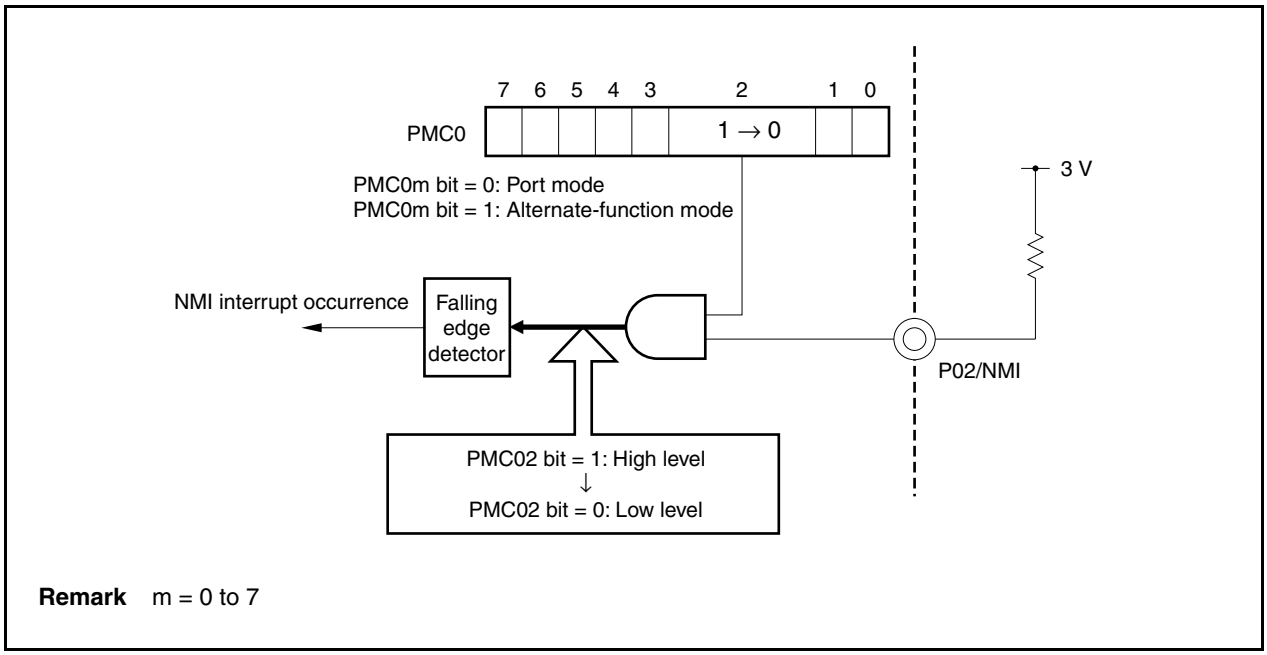
Remark m = 0 to 7

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

When the P02/NMI pin is pulled up as shown in Figure 4-47 and the falling edge is specified in the NMI pin edge detection setting, even though high level is input continuously to the NMI pin at switching from the NMI pin to the P02 pin (PMC02 bit = 1 → 0), this is detected as falling edge as if high level changed to low level, and NMI interrupt occurs.

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

Figure 4-47. Example of Switching from NMI to P02 (Incorrect)



Remark m = 0 to 7

(2) In port mode, the PFn.PFnm bit is valid only in the output mode (PMn.PMnm bit = 0). In the input mode (PMnm bit = 1), the value of the PFnm 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, 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 P90 pin is an output port, P91 to P97 pins are input ports (all pin statuses are high level), and the value of the port latch is 00H, if the output of P90 pin is changed from low level to high level via a bit manipulation instruction, the value of the port latch is FFH.

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

A bit manipulation instruction is executed in the following order in the V850ES/Hx3.

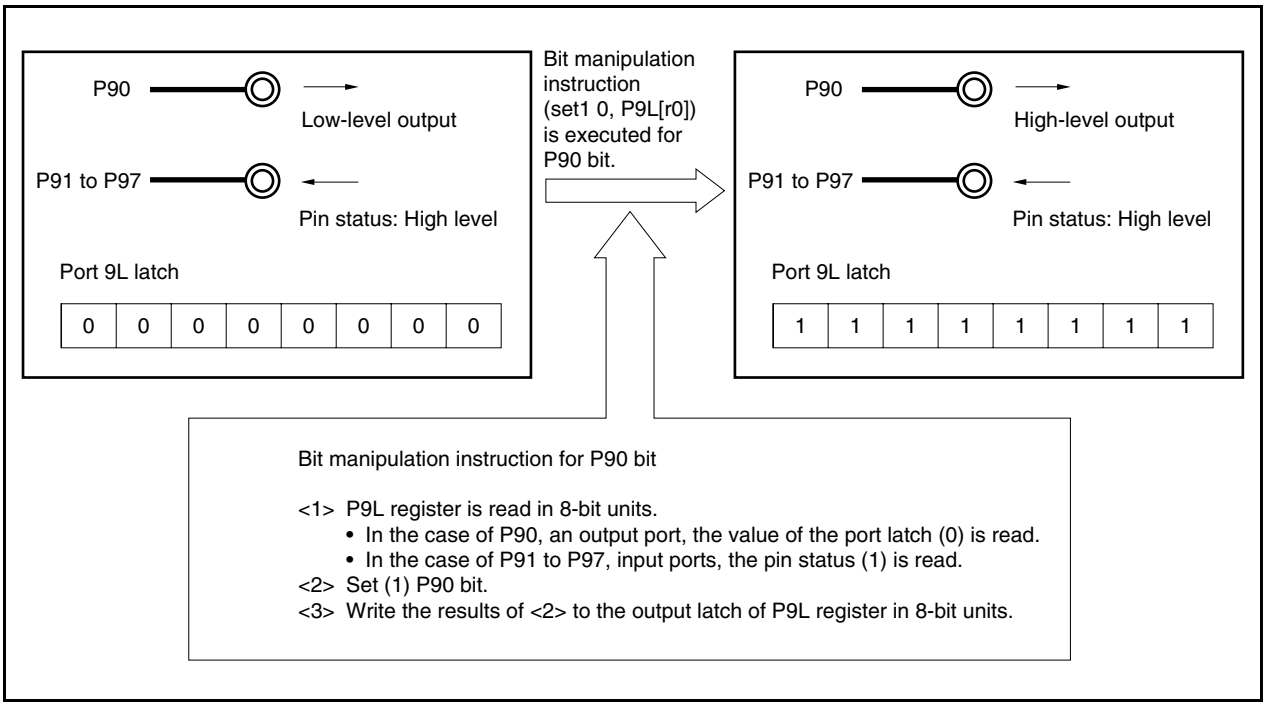
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

In step <1>, the value of the output latch (0) of P90 pin, which is an output port, is read, while the pin statuses of P91 to P97 pins, which are input ports, are read. If the pin statuses of P91 to P97 pins are high level at this time, the read value is FEH.

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

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

Figure 4-48. 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 kΩ TYP.). After a reset by the $\overline{\text{RESET}}$ pin, a pull-down resistor is connected. The pull-down resistor is disconnected when the OCDM0 bit is cleared (0).

4.6.5 Cautions on P53 pin when power is turned on

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

- P53/KR3/TIAB00/TOAB00/TOAB0B2/DDO pin

4.6.6 Hysteresis characteristics

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

P00 to P06
P10, P11
P31 to P35, P39
P40 to P42
P50 to P55
P60 to P62, P610 to P613
P80, P81
P90 to P910, P912 to P915

CHAPTER 5 BUS CONTROL FUNCTION (V850ES/HJ3 ONLY)

The V850ES/HJ3 is provided with an external bus interface function by which external memories such as ROM and RAM, and I/O can be connected.

5.1 Features

- ○ Multiplexed bus with a minimum of 3 bus cycles
- ○ 8-bit/16-bit data bus selectable
- ○ Wait function
  - Programmable wait function of up to 7 states
  - External wait function using $\overline{\text{WAIT}}$ pin
- ○ Idle state function
- ○ Bus hold function
- ○ External devices can be connected using alternate-function port pins
- ○ Misaligned access possible
- ○ Chip select function (4 spaces)

5.2 Bus Control Pins

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

Table 5-1. Bus Control Pins (Multiplexed Bus)

Bus Control Pin	Alternate-Function Pin	I/O	Function
AD0 to AD15	PDL0 to PDL15	I/O	Address/data bus
$\overline{\text{WAIT}}$	PCM0	Input	External wait control
CLKOUT	PCM1	Output	Internal system clock
$\overline{\text{WR0}}$, $\overline{\text{WR1}}$	PCT0, PCT1	Output	Write strobe signal
$\overline{\text{RD}}$	PCT4	Output	Read strobe signal
ASTB	PCT6	Output	Address strobe signal
$\overline{\text{HLDRQ}}$	PCM3	Input	Bus hold control
$\overline{\text{HLDAK}}$	PCM2	Output	
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$	PCS0 to PCS3	Output	Chip select

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

When the internal ROM, internal RAM, or on-chip peripheral I/O are accessed, the status of each pin is as follows.

Table 5-2. Pin Statuses When Internal ROM, Internal RAM, or On-Chip Peripheral I/O Is Accessed

Address/data bus (AD15 to AD0)	Undefined
Control signal	Inactive

Caution When a write access is performed to the internal ROM area, address, data, and control signals are activated in the same way as access to the external memory area.

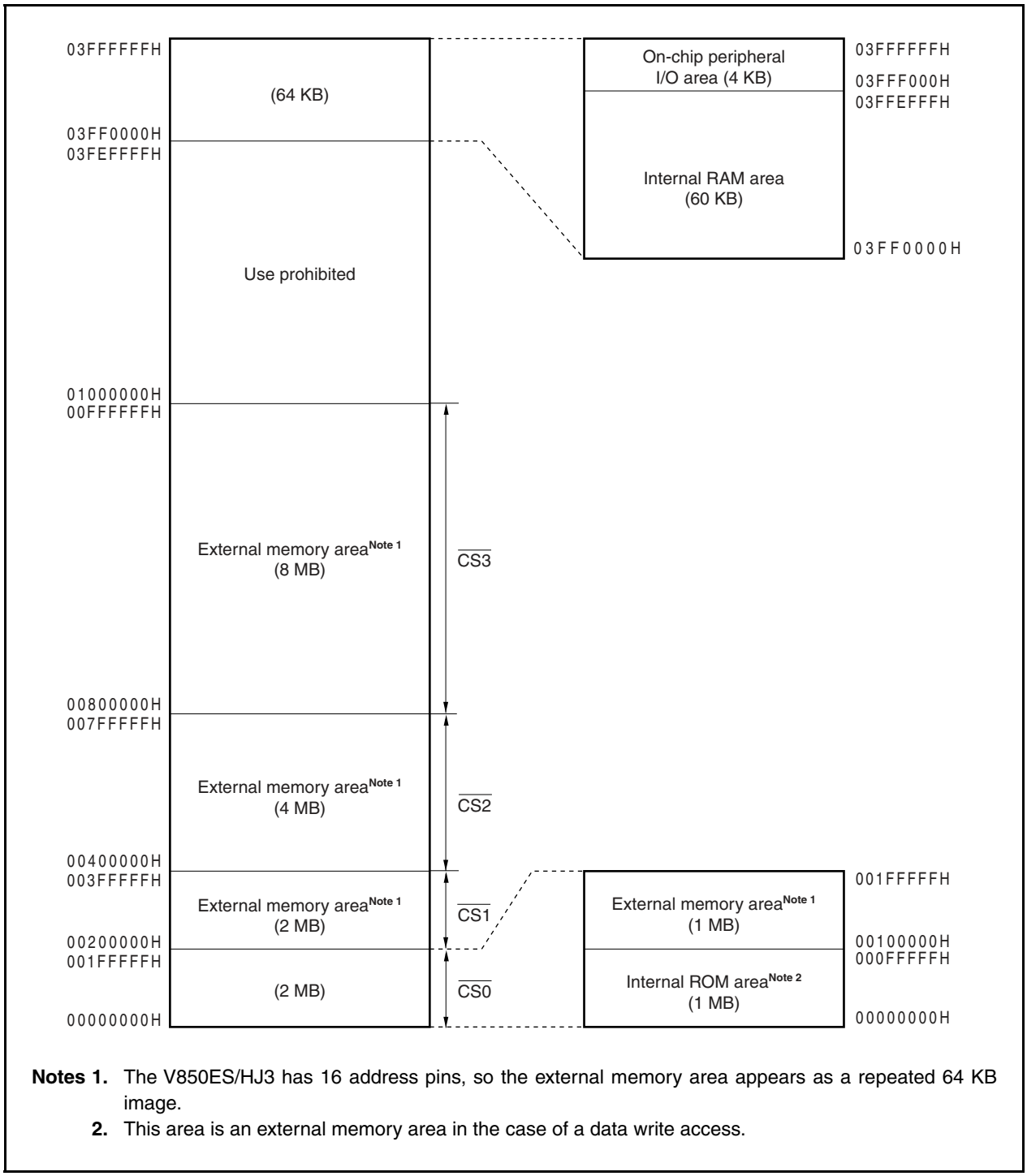
5.2.2 Pin status in each operation mode

For the pin status of the V850ES/HJ3 in each operation mode, see **2.2 Pin Status**.

5.3 Memory Block Function

The 16 MB external memory space is divided into memory blocks of (lower) 2 MB, 2 MB, 4 MB, and 8 MB. The programmable wait function and bus cycle operation mode for each of these blocks can be independently controlled in one-block units.

Figure 5-1. Data Memory Map: Physical Address



Notes 1. The V850ES/HJ3 has 16 address pins, so the external memory area appears as a repeated 64 KB image.

2. This area is an external memory area in the case of a data write access.

5.4 Bus Access

5.4.1 Number of clocks for access

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

Table 5-3. Number of Clocks for Access (μPD70F3755)

Bus Cycle Type \ Area (Bus Width)	Internal ROM (32 Bits)	Internal RAM (32 Bits)	External Memory (16 Bits)
Instruction fetch (normal access)	1	1[Note]	3 + n
Instruction fetch (branch)	2	2[Note]	3 + n
Operand data access	3	1	3 + n

Note Increases by 1 if a conflict with a data access occurs.

Remarks 1. Unit: Clocks/access

2. n = Number of waits inserted

Table 5-4. Number of Clocks for Access (μPD70F3757)

Bus Cycle Type \ Area (Bus Width)	Internal ROM (32 Bits)	Internal RAM (32 Bits)	External Memory (16 Bits)
Instruction fetch (normal access)	1	1[Note]	3 + n
<R> Instruction fetch (branch)	3	2[Note]	3 + n
Operand data access	5	1	3 + n

Note Increases by 1 if a conflict with a data access occurs.

Remarks 1. Unit: Clocks/access

2. n = Number of waits inserted

5.4.2 Bus size setting function

Each external memory area selected by $\overline{CSn}$ can be set by using the BSC register. However, the bus size can be set to 8 bits and 16 bits only.

The external memory area of the V850ES/HJ3 is selected by $\overline{CS0}$ to $\overline{CS3}$.

(1) Bus size configuration register (BSC)

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

Reset sets this register to 5555H.

Caution Write to the BSC register after reset, and then do not change the set values. Also, do not access an external memory area until the initial settings of the BSC register are complete.

After reset: 5555H R/W Address: FFFFF066H

	15	14	13	12	11	10	9	8
BSC	0	1	0	1	0	1	0	1

7	6	5	4	3	2	1	0
0	BS30	0	BS20	0	BS10	0	BS00
	$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$

BSn0	Data bus width of CSn space (n = 0 to 3)
0	8 bits
1	16 bits

Caution Be sure to set bits 14, 12, 10, and 8 to “1”, and set bits 15, 13, 11, 9, 7, 5, 3, and 1 to “0”.

5.4.3 Access by bus size

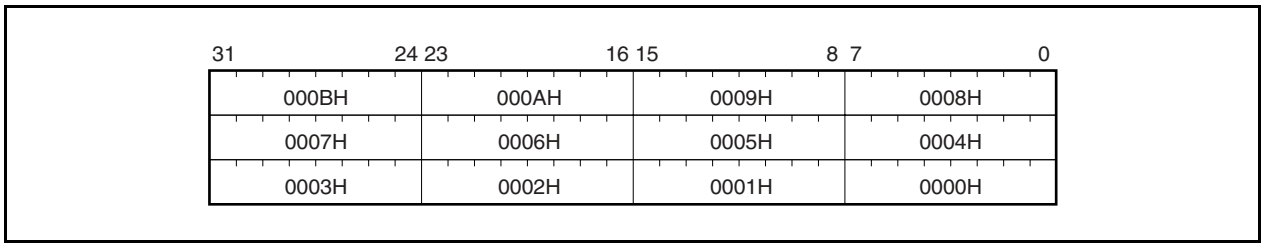
The V850ES/HJ3 accesses the on-chip peripheral I/O and external memory in 8-bit, 16-bit, or 32-bit units. The bus size is as follows.

- The bus size of the on-chip peripheral I/O is fixed to 16 bits.
- The bus size of the external memory is selectable from 8 bits or 16 bits (by using the BSC register).

The operation when each of the above is accessed is described below. All data is accessed starting from the lower side.

The V850ES/HJ3 supports only the little-endian format.

Figure 5-2. Little-Endian Address in Word



(1) Data space

The V850ES/HJ3 has an address misalign function.

With this function, data can be placed at all addresses, regardless of the format of the data (word data or halfword data). However, if the word data or halfword data is not aligned at the boundary, a bus cycle is generated at least twice, causing the bus efficiency to drop.

(a) Halfword-length data access

A byte-length bus cycle is generated twice if the least significant bit of the address is 1.

(b) Word-length data access

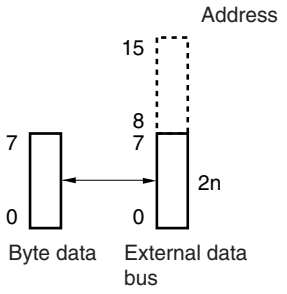
(i) A byte-length bus cycle, halfword-length bus cycle, and byte-length bus cycle are generated in that order if the least significant bit of the address is 1.

(ii) A halfword-length bus cycle is generated twice if the lower 2 bits of the address are 10.

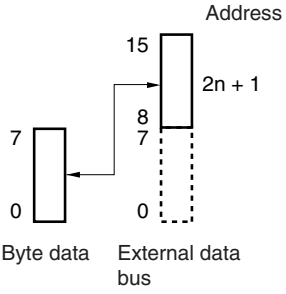
(2) Byte access (8 bits)

(a) 16-bit data bus width

<1> Access to even address (2n)

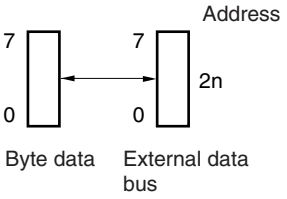


<2> Access to odd address (2n + 1)

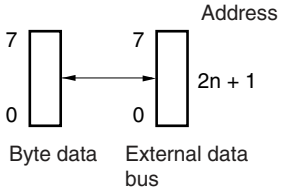


(b) 8-bit data bus width

<1> Access to even address (2n)



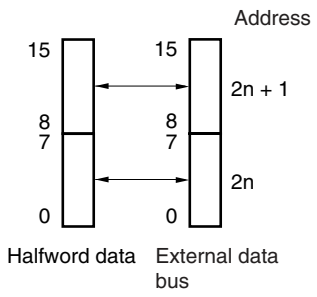
<2> Access to odd address (2n + 1)



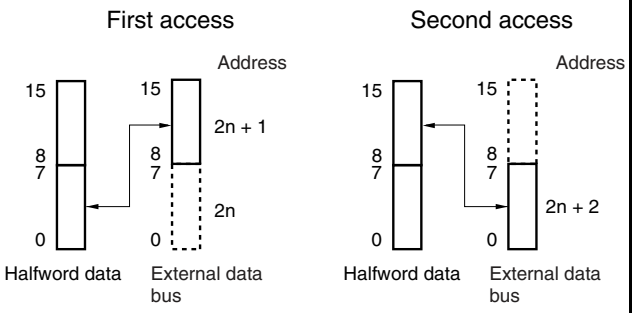
(3) Halfword access (16 bits)

(a) With 16-bit data bus width

<1> Access to even address (2n)

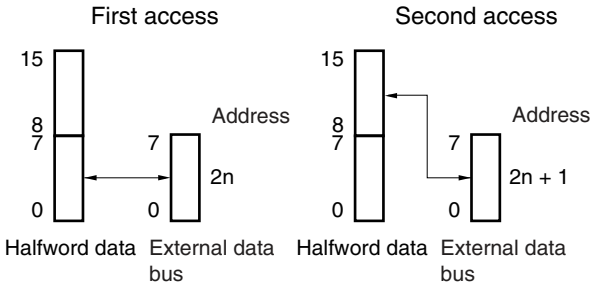


<2> Access to odd address (2n + 1)

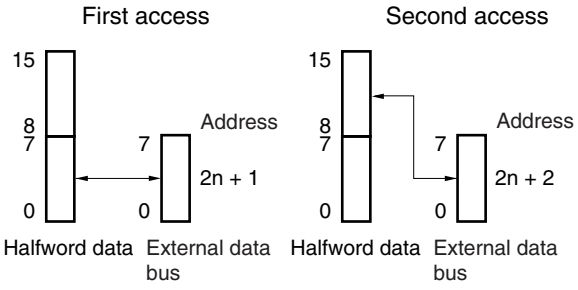


(b) 8-bit data bus width

<1> Access to even address (2n)



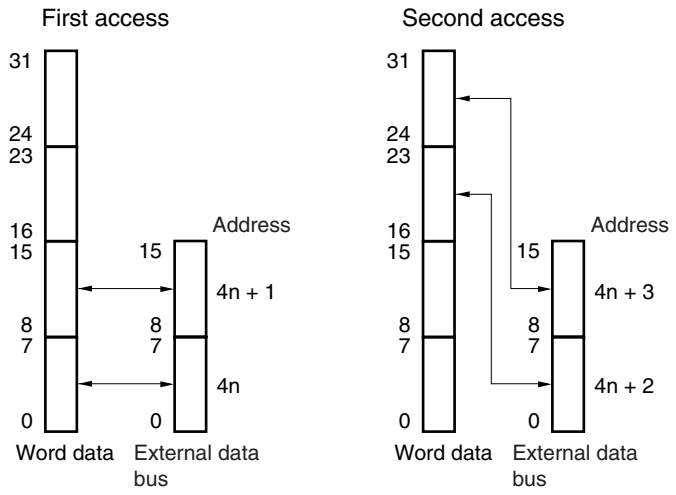
<2> Access to odd address (2n + 1)



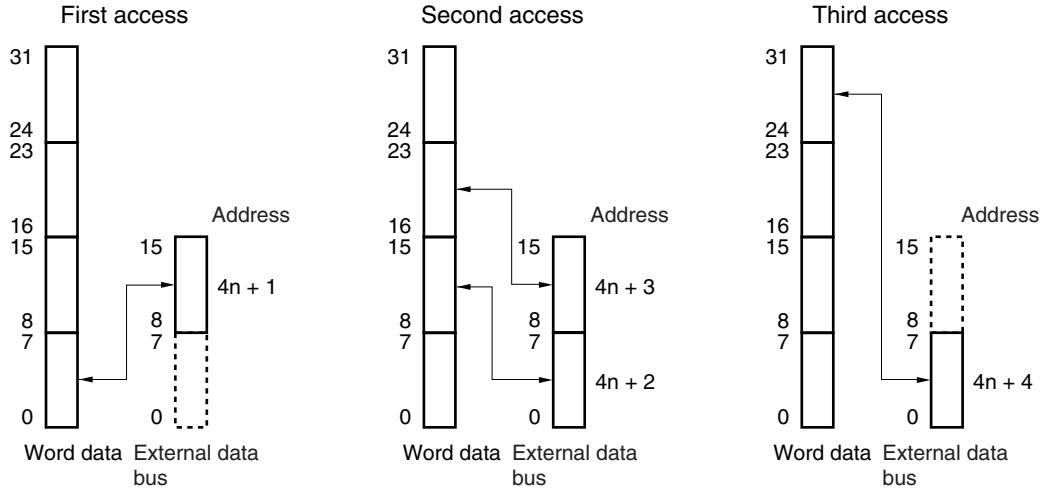
(4) Word access (32 bits)

(a) 16-bit data bus width (1/2)

<1> Access to address (4n)

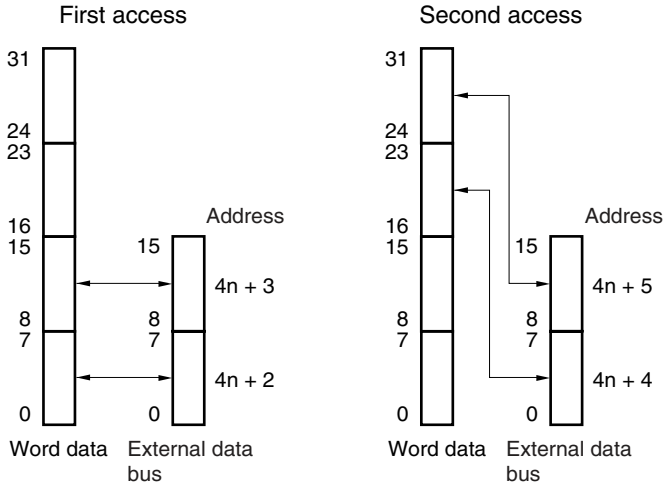


<2> Access to address (4n + 1)

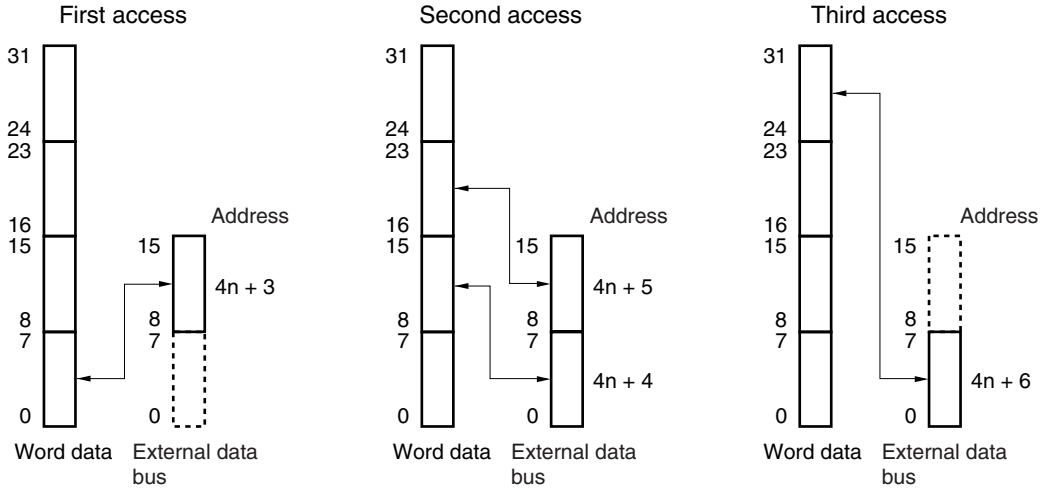


(a) 16-bit data bus width (2/2)

<3> Access to address (4n + 2)

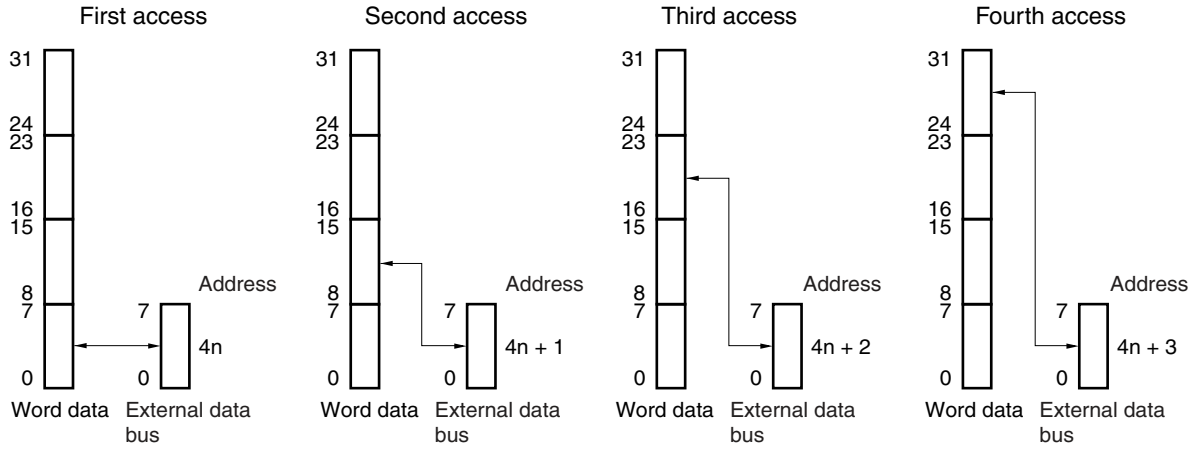


<4> Access to address (4n + 3)

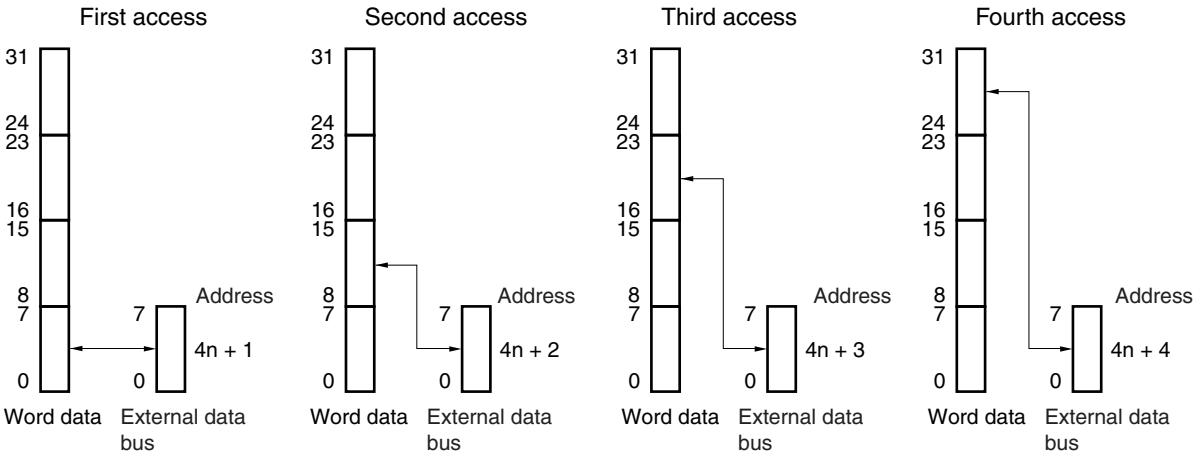


(b) 8-bit data bus width (1/2)

<1> Access to address (4n)

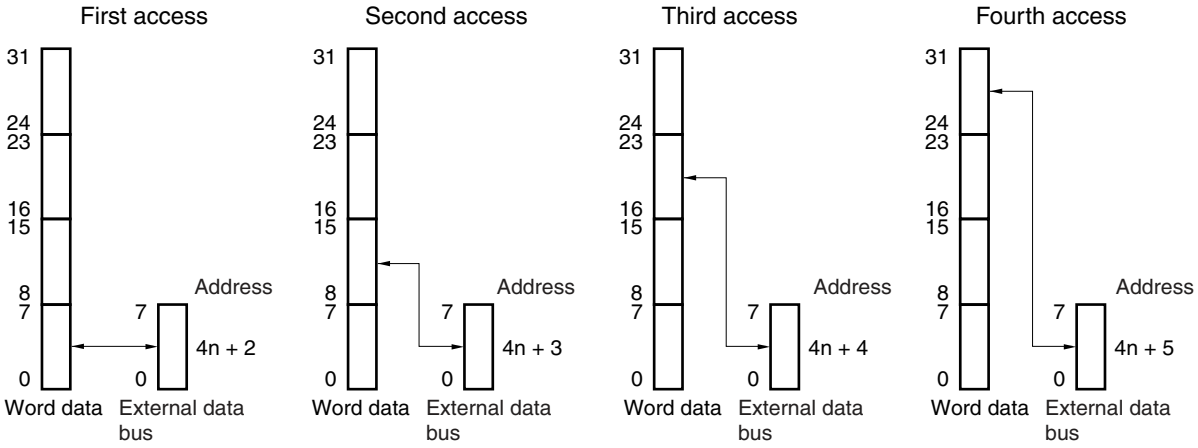


<2> Access to address (4n + 1)

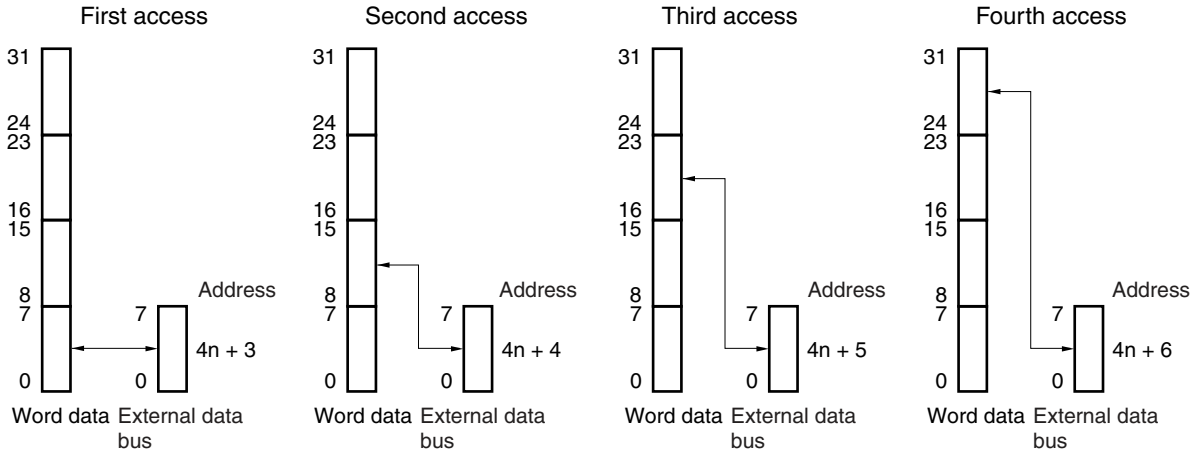


(b) 8-bit data bus width (2/2)

<3> Access to address (4n + 2)



<4> Access to address (4n + 3)



5.5 Wait Function

5.5.1 Programmable wait function

(1) Data wait control register 0 (DWC0)

To realize interfacing with a low-speed memory or I/O, up to seven data wait states can be inserted in the bus cycle that is executed for each CS space.

The number of wait states can be programmed by using the DWC0 register. Immediately after system reset, 7 data wait states are inserted for all the blocks.

The DWC0 register can be read or written in 16-bit units.

Reset sets this register to 7777H.

Cautions 1. The internal ROM and internal RAM areas are not subject to programmable wait, and are always accessed without a wait state. The on-chip peripheral I/O area is also not subject to programmable wait, and only wait control from each peripheral function is performed.

2. Write to the DWC0 register after reset, and then do not change the set values. Also, do not access an external memory area until the initial settings of the DWC0 register are complete.

After reset: 7777H R/W Address: FFFFF484H

	15	14	13	12	11	10	9	8
DWC0	0	DW32	DW31	DW30	0	DW22	DW21	DW20
			$\overline{CS3}$				$\overline{CS2}$	

	7	6	5	4	3	2	1	0
	0	DW12	DW11	DW10	0	DW02	DW01	DW00
			$\overline{CS1}$				$\overline{CS0}$	

DWn2	DWn1	DWn0	Number of wait states inserted in CSn space (n = 0 to 3)
0	0	0	Not inserted
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Caution Be sure to set bits 15, 11, 7, and 3 to "0".

5.5.2 External wait function

To synchronize an extremely slow external memory, I/O, or asynchronous system, any number of wait states can be inserted in the bus cycle by using the external wait pin ($\overline{\text{WAIT}}$).

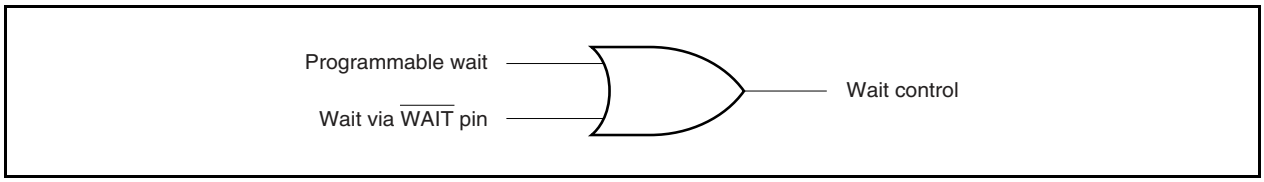
When the PCM0 pin is set to alternate function, the external wait function is enabled.

Access to each area of the internal ROM, internal RAM, and on-chip peripheral I/O is not subject to control by the external wait function, in the same manner as the programmable wait function.

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

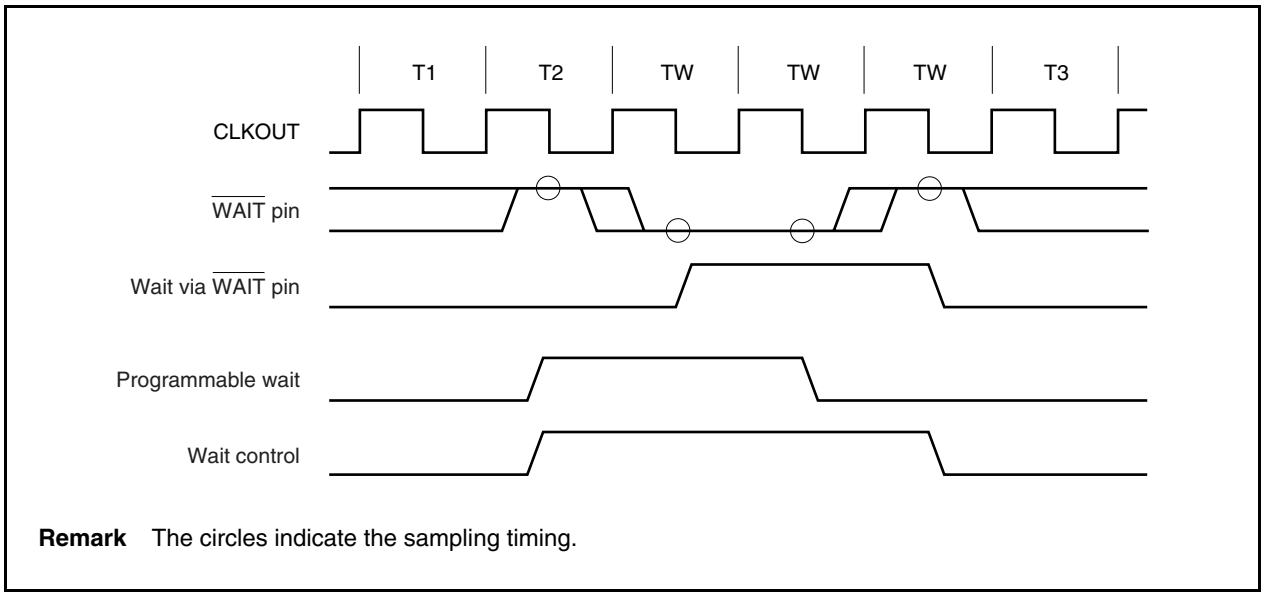
5.5.3 Relationship between programmable wait and external wait

Wait cycles are inserted as the result of an OR operation between the wait cycles specified by the set value of the programmable wait and the wait cycles controlled by the $\overline{\text{WAIT}}$ pin.



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

Figure 5-3. Inserting Wait Example



Remark The circles indicate the sampling timing.

5.5.4 Programmable address wait function

Address-setup or address-hold waits to be inserted in each bus cycle can be set by using the AWC register. Address wait insertion is set for each chip select area ($\overline{CS0}$ to $\overline{CS3}$).

If an address setup wait is inserted, it seems that the high-clock period of the T1 state is extended by 1 clock. If an address hold wait is inserted, it seems that the low-clock period of the T1 state is extended by 1 clock.

(1) Address wait control register (AWC)

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

Reset sets this register to FFFFH.

Cautions 1. Address setup wait and address hold wait cycles are not inserted when the internal ROM area, internal RAM area, and on-chip peripheral I/O areas are accessed.

2. Write to the AWC register after reset, and then do not change the set values. Also, do not access an external memory area until the initial settings of the AWC register are complete.

After reset: FFFFH R/W Address: FFFFF488H

	15	14	13	12	11	10	9	8
AWC	1	1	1	1	1	1	1	1

7	6	5	4	3	2	1	0
AHW3	ASW3	AHW2	ASW2	AHW1	ASW1	AHW0	ASW0
$\overline{CS3}$		$\overline{CS2}$		$\overline{CS1}$		$\overline{CS0}$	

AHWn	Specifies insertion of address hold wait (n = 0 to 3)
0	Not inserted
1	Inserted

ASWn	Specifies insertion of address setup wait (n = 0 to 3)
0	Not inserted
1	Inserted

Caution Be sure to set bits 15 to 8 to “1”.

5.6 Idle State Insertion Function

To facilitate interfacing with low-speed memories, one idle state (TI) can be inserted after the T3 state in the bus cycle that is executed for each space selected by the chip select function. By inserting an idle state, the data output float delay time of the memory can be secured during read access (an idle state cannot be inserted during write access).

Whether the idle state is to be inserted can be programmed by using the BCC register.

An idle state is inserted for all the areas immediately after system reset.

(1) Bus cycle control register (BCC)

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

Reset sets this register to AAAAH.

Cautions 1. The internal ROM, internal RAM, and on-chip peripheral I/O areas are not subject to idle state insertion.

2. Write to the BCC register after reset, and then do not change the set values. Also, do not access an external memory area until the initial settings of the BCC register are complete.

After reset: AAAAH R/W Address: FFFFF48AH

	15	14	13	12	11	10	9	8
BCC	1	0	1	0	1	0	1	0

7	6	5	4	3	2	1	0
BC31	0	BC21	0	BC11	0	BC01	0

$\overline{CS3}$ $\overline{CS2}$ $\overline{CS1}$ $\overline{CS0}$

BCn1	Specifies insertion of idle state (n = 0 to 3)
0	Not inserted
1	Inserted

Caution Be sure to set bits 15, 13, 11, and 9 to “1”, and set bits 14, 12, 10, 8, 6, 4, 2, and 0 to “0”.

5.7 Bus Hold Function

5.7.1 Functional outline

The $\overline{\text{HLDRQ}}$ and $\overline{\text{HLDAK}}$ functions are valid if the PCM2 and PCM3 pins are set in the control mode.

When the $\overline{\text{HLDRQ}}$ pin is asserted (low level), indicating that another bus master has requested bus mastership, the external address/data bus goes into a high-impedance state and is released (bus hold status). If the request for the bus mastership is cleared and the $\overline{\text{HLDRQ}}$ pin is deasserted (high level), driving these pins is started again.

During the bus hold period, execution of the program in the internal ROM and internal RAM is continued until an on-chip peripheral I/O register or the external memory is accessed.

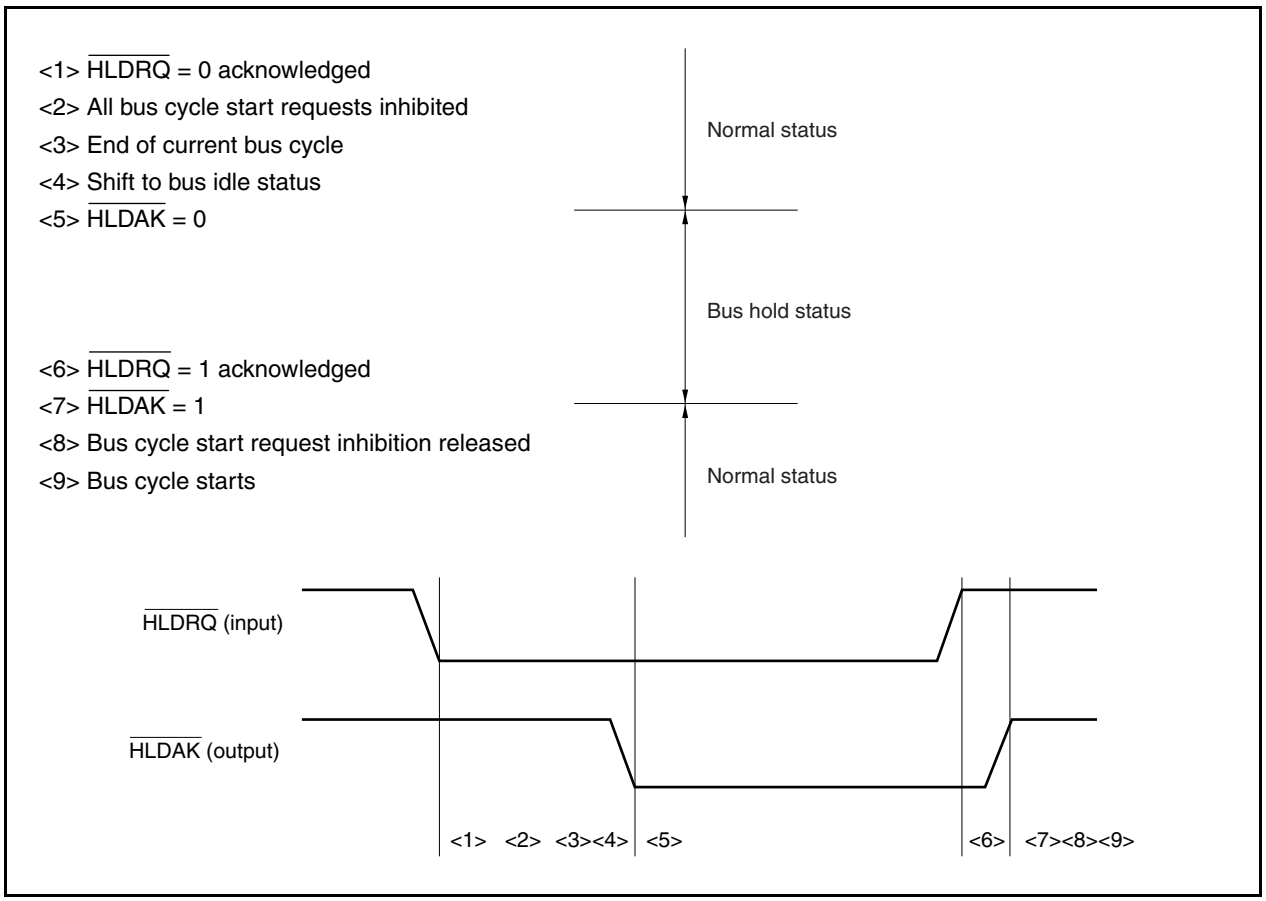
The bus hold status is indicated by assertion of the $\overline{\text{HLDAK}}$ pin (low level). The bus hold function enables the configuration of multi-processor type systems in which two or more bus masters exist.

Note that the bus hold request is not acknowledged during a multiple-access cycle initiated by the bus sizing function or a bit manipulation instruction.

Status	Data Bus Width	Access Type	Timing at Which Bus Hold Request Is Not Acknowledged
CPU bus lock	16 bits	Word access to even address	Between first and second access
		Word access to odd address	Between first and second access
			Between second and third access
		Halfword access to odd address	Between first and second access
	8 bits	Word access	Between first and second access
			Between second and third access
			Between third and fourth access
		Halfword access	Between first and second access
Read-modify-write access of bit manipulation instruction	–	–	Between read access and write access

5.7.2 Bus hold procedure

The bus hold status transition procedure is shown below.



5.7.3 Operation in power save mode

Because the internal system clock is stopped in the STOP, IDLE1, IDLE2, and sub-IDLE modes, the bus hold status is not entered even if the $\overline{\text{HLDRQ}}$ pin is asserted.

In the HALT mode, the $\overline{\text{HLDAK}}$ pin is asserted as soon as the $\overline{\text{HLDRQ}}$ pin has been asserted, and the bus hold status is entered. When the $\overline{\text{HLDRQ}}$ pin is later deasserted, the $\overline{\text{HLDAK}}$ pin is also deasserted, and the bus hold status is cleared.

5.8 Bus Priority

Bus hold, DMA transfer, operand data accesses, instruction fetch (branch), and instruction fetch (successive) are executed in the external bus cycle.

Bus hold has the highest priority, followed by DMA transfer, operand data access, instruction fetch (branch), and instruction fetch (successive).

An instruction fetch may be inserted between the read access and write access in a read-modify-write access.

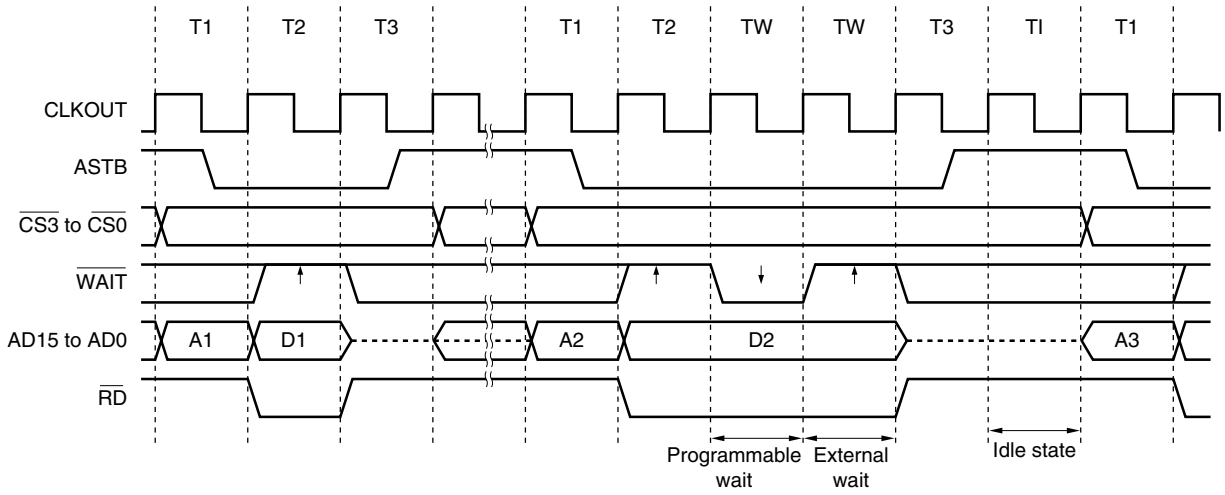
If an instruction is executed for two or more accesses, an instruction fetch and bus hold are not inserted between accesses due to bus size limitations.

Table 5-5. Bus Priority

Priority	External Bus Cycle	Bus Master
High ↕ Low	Bus hold	External device
	DMA transfer	DMAC
	Operand data access	CPU
	Instruction fetch (branch)	CPU
	Instruction fetch (successive)	CPU

5.9 Bus Timing

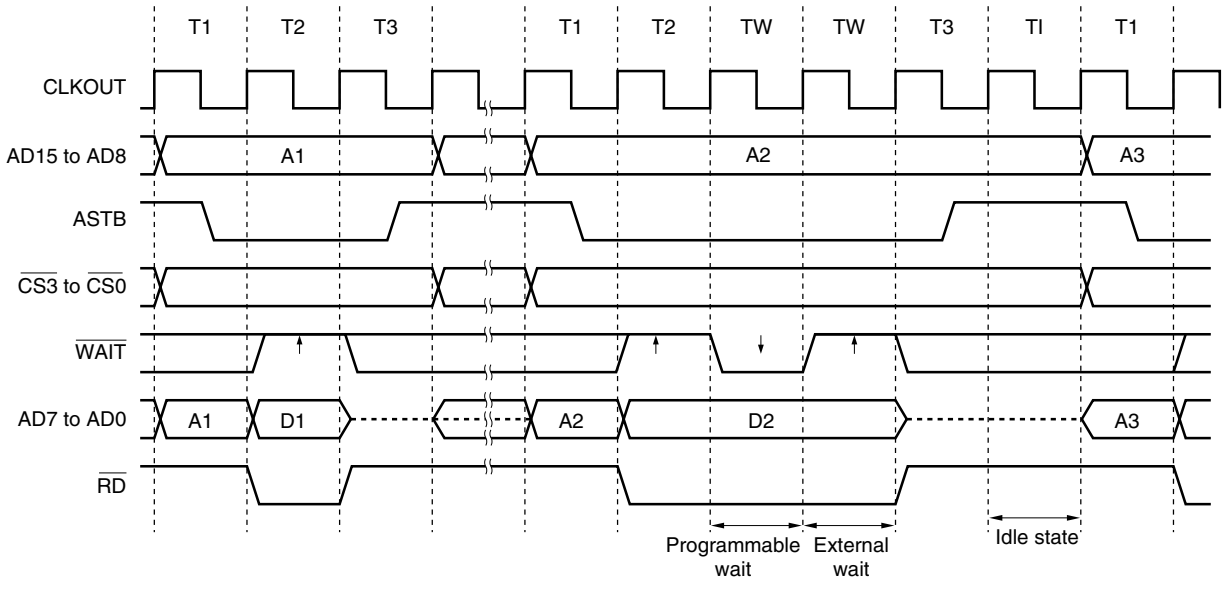
Figure 5-4. Bus Read Timing (Bus Size: 16 Bits, 16-Bit Access)



8-bit access	Odd address	Even address
AD15 to AD8	Active	Hi-Z
AD7 to AD0	Hi-Z	Active

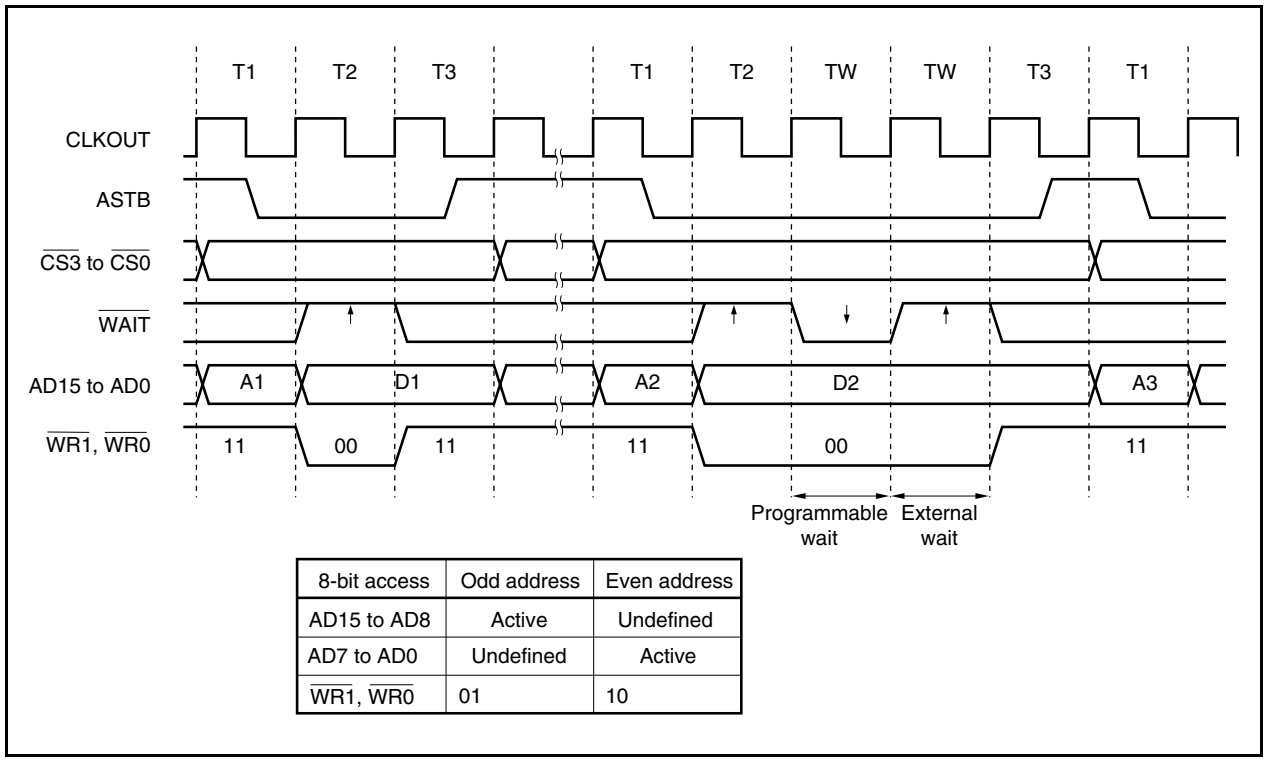
Remark The broken lines indicate high impedance.

Figure 5-5. Bus Read Timing (Bus Size: 8 Bits)



Remark The broken lines indicate high impedance.

Figure 5-6. Bus Write Timing (Bus Size: 16 Bits, 16-Bit Access)



8-bit access	Odd address	Even address
AD15 to AD8	Active	Undefined
AD7 to AD0	Undefined	Active
WR1, WR0	01	10

Figure 5-7. Bus Write Timing (Bus Size: 8 Bits)

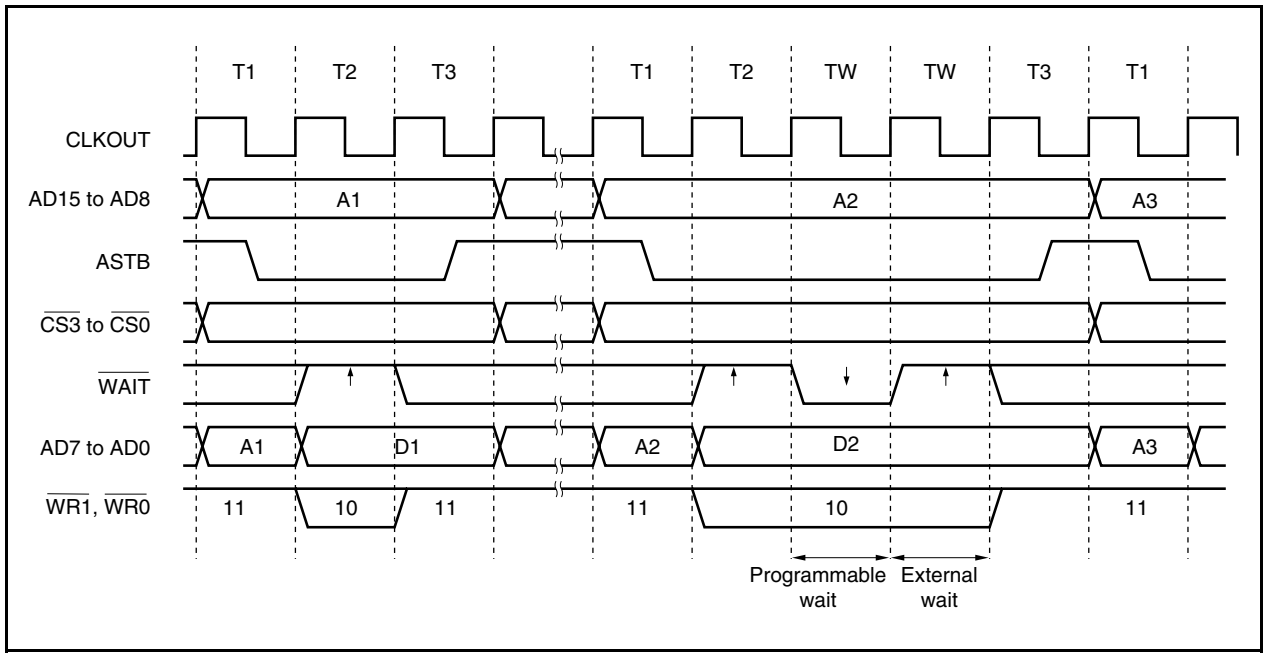
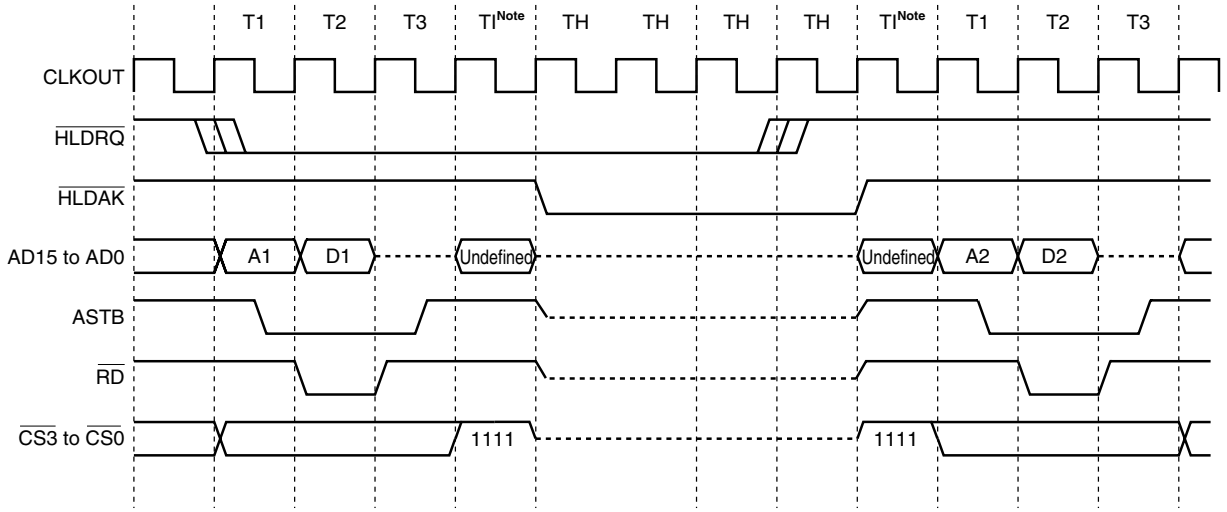


Figure 5-8. Bus Hold Timing (Bus Size: 16 Bits, 16-Bit Access)

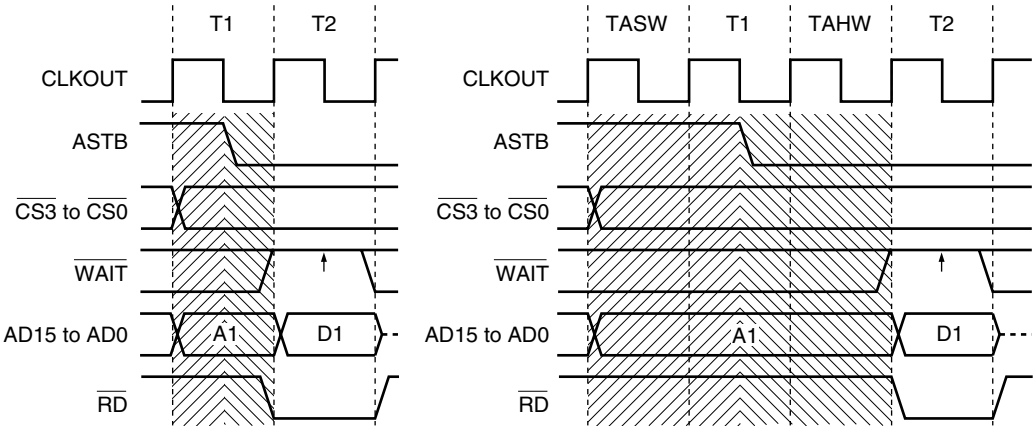


Note This idle state (TI) does not depend on the BCC register settings.

Remarks 1. See **Table 2-5** for the pin statuses in the bus hold mode.

2. The broken lines indicate high impedance.

Figure 5-9. Address Wait Timing (Bus Size: 16 Bits, 16-Bit Access)



Remarks 1. TASW (address setup wait): Image of high-level width of T1 state expanded.

2. TAHW (address hold wait): Image of low-level width of T1 state expanded.

3. The broken lines indicate high impedance.

CHAPTER 6 CLOCK GENERATION FUNCTION

6.1 Overview

The following clock generation functions are available.

○ Main clock oscillator
- In clock-through mode
  f_X = 4 to 16 MHz (f_{XX} = 4 to 16 MHz)
- In PLL mode

<R> f_X = 6 to 12 MHz (f_{XX} = 12 to 32 MHz)
- In SSCG mode

<R> f_X = 4 to 16 MHz (f_{XX} = 10.5 to 32 MHz[Note])

○ Subclock oscillator
- f_{XT} = 32.768 kHz

○ Multiply function by PLL (Phase Locked Loop)
- Clock-through mode/PLL mode selectable

○ Multiply function by SSCG (Spread Spectrum Clock Generator)
- Frequency modulation rate and modulation cycle controllable
- Effect of reducing peak value of EMI noise

○ Low-speed internal oscillator
- f_{RL} = 240 kHz (TYP.)

○ High-speed internal oscillator
- f_{RH} = 8 MHz (TYP.)

○ 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

○ Clock output (CLKOUT) function

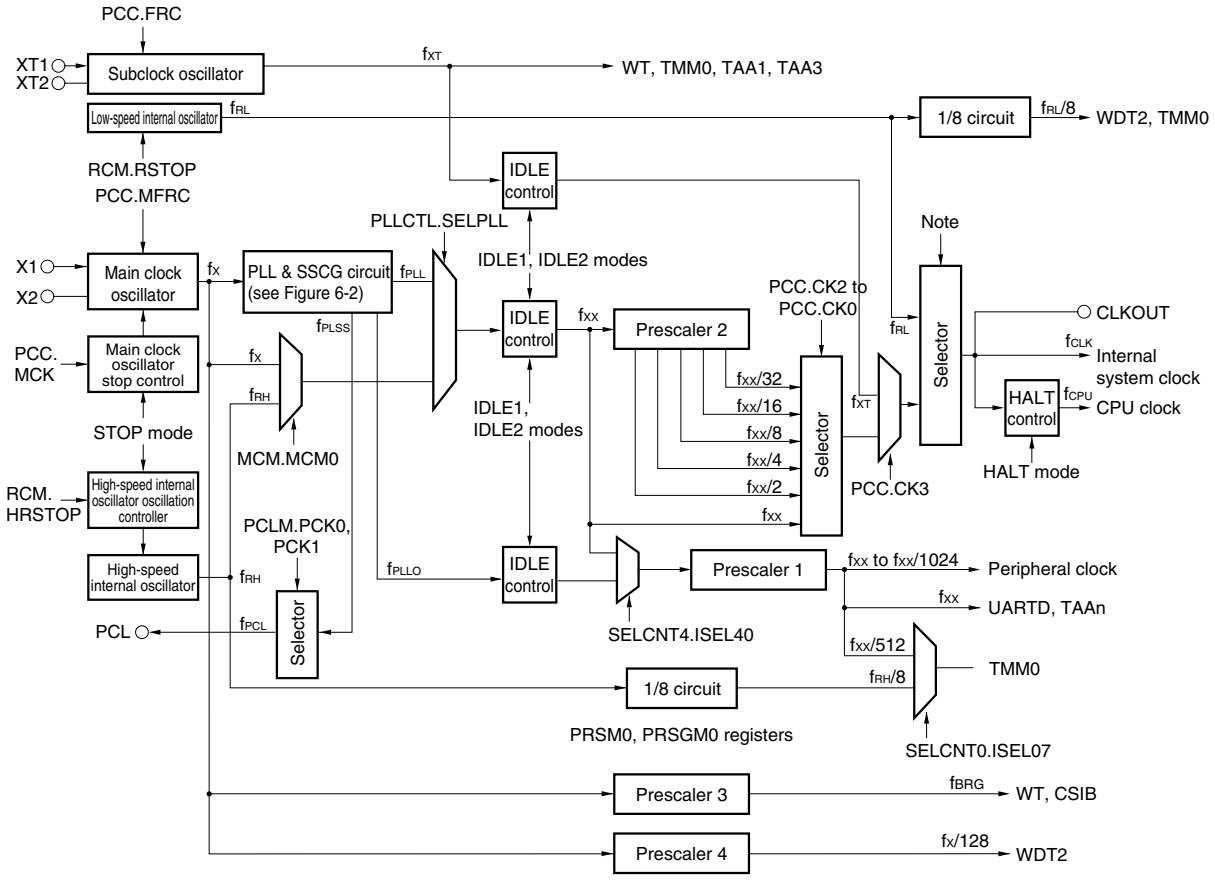
○ Programmable clock output (PCL) function

Note Do not set a frequency exceeding 32 MHz, taking an increment of modulation rate into consideration.

Remark f_X: Main clock oscillation frequency
f_{XX}: Main clock frequency
f_{XT}: Subclock frequency
f_{RL}: Low-speed internal oscillation clock frequency
f_{RH}: High-speed internal oscillation clock frequency

6.2 Configuration

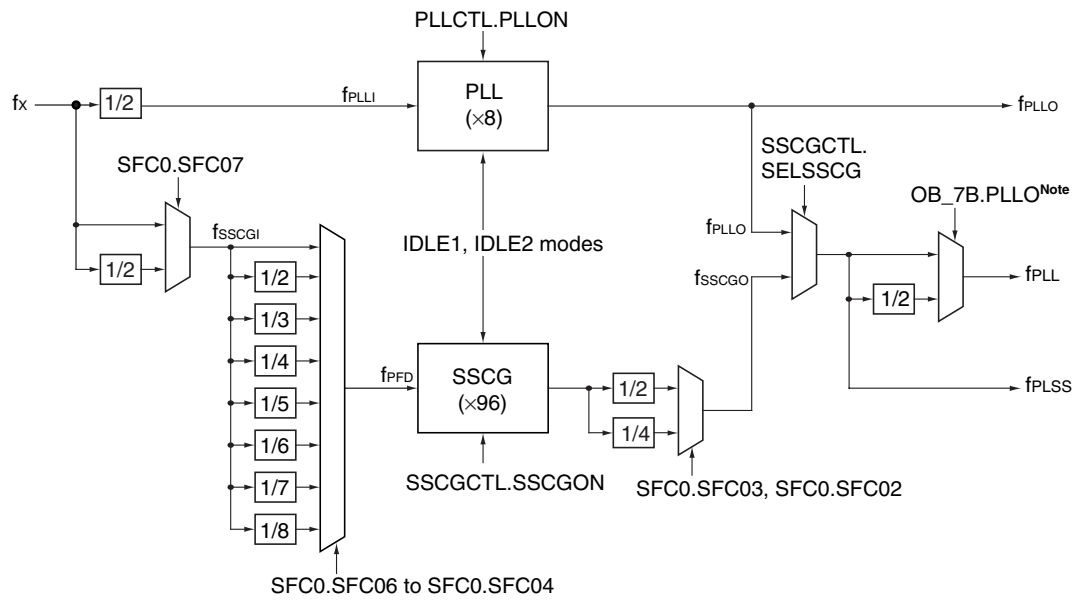
Figure 6-1. Clock Generator



Note The low-speed internal oscillation clock is selected when watchdog timer 2 overflows during the oscillation stabilization time.

Remark f_X: Main clock oscillation frequency
f_{XX}: Main clock frequency
f_{CLK}: Internal system clock frequency
f_{XT}: Subclock frequency
f_{CPU}: CPU clock frequency
f_{BRG}: Watch timer clock frequency
f_{RL}: Low-speed internal oscillation clock frequency
f_{RH}: High-speed internal oscillation clock frequency
f_{PLL}: Multiplication block output frequency
f_{PLLO}: PLL output clock frequency
f_{PLSS}: Multiplication block output frequency for PCL output
f_{PCL}: PCL output clock frequency

Figure 6-2. PLL/SSCG Circuit



Note See **CHAPTER 27 OPTION BYTE FUNCTION**.

Remark f_X: Main clock oscillation frequency
f_{PLLI}: PLL input clock frequency
f_{PLLO}: PLL output clock frequency
f_{PLL}: Multiplication block output frequency
f_{PLSS}: Multiplication block output frequency for PCL output
f_{SSCGI}: SSCG input clock frequency
f_{PFD}: Intermediate SSCG clock frequency
f_{SSCGO}: SSCG output clock frequency

(1) Main clock oscillator

This circuit oscillates the following frequencies (f_X).

- In clock-through mode
  f_X = 4 to 16 MHz
- In PLL mode
  f_X = 6 to 12 MHz <R>
- In SSCG mode
  f_X = 4 to 16 MHz <R>

(2) Subclock oscillator

This circuit oscillates a frequency (f_{XT}) of 32.768 kHz.

(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 = 1 (valid only when the PCC.CLS bit = 1).

(4) Low-speed internal oscillator

This circuit oscillates a frequency (f_{RL}) of 240 kHz (TYP.).

(5) High-speed internal oscillator

This circuit oscillates a frequency (f_{RH}) of 8 MHz (TYP.).

(6) Prescaler 1

This circuit generates the clock (f_{XX} to f_{XX}/1,024) to be supplied to the on-chip peripheral functions.

The block to be supplied is shown below.

TAAn, TABn, TMM0, CSIBn, UARTDn, I^2C00, ADC, and WDT2

(7) 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, and can be output from the CLKOUT pin.

(8) Prescaler 3

This circuit divides the clock generated by the main clock oscillator (f_X) to a specific frequency (32.768 kHz) and supplies that clock to the watch timer block.

For details, see **CHAPTER 11 WATCH TIMER FUNCTIONS**.

(9) PLL

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

It operates and stops 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, and operates and stops by using the PLLCTL.PLLON bit.

(10) SSCG (Spread Spectrum Clock Generator)

SSCG can control the frequency modulation rate and modulation cycle, and can also reduce the peak value of EMI noise.

This circuit selects a division ratio (f_{SSCGI}) of the main clock oscillation (f_X) and multiplies it by 96 (f_{SSCGO}).

SSCG is controlled by the SELSSCTL register.

After reset release, SSCG is stopped.

6.3 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 **3.4.8 Special registers**).

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

Reset sets this register to 40H.

After reset: 40H R/W Address: FFFFF828H

	7	<6>	5	<4>	<3>	2	1	0
PCC	FRC	MCK	MFRC	CLS[Note]	CK3	CK2	CK1	CK0

FRC	Use of subclock on-chip feedback resistor
0	Used
1	Not used

MCK	Main clock oscillator control
0	Oscillation enabled
1	Oscillation stopped

- Even if the MCK bit is set to 1 while the system is operating with the main clock as the CPU clock, the operation of the main clock does not stop. Do not set the MCK bit to 1 while the CPU is operating with the main clock.
- Before setting the MCK bit from 0 to 1, stop the on-chip peripheral functions operating with the main clock.
- When the main clock is stopped and the device is operating with the high-speed internal oscillator or subclock, clear the MCK bit to 0 and secure the oscillation stabilization time by software before switching the CPU clock to the main clock or operating the on-chip peripheral functions.

MFRC	Use of main clock on-chip feedback resistor
0	Used
1	Not used

CLS[Note]	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	$f_{XX}/8$
0	1	0	0	$f_{XX}/16$
0	1	0	1	$f_{XX}/32$
0	1	1	×	Setting prohibited
1	×	×	×	f_{XT}

Note The CLS bit is a read-only bit.

Cautions 1. Do not change the CPU clock (by using the CK3 to CK0 bits) while CLKOUT is being output.

2. 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.

Be sure to manipulate the CK3 bit while the CPU is operating with the main clock (MCM.MCS = 1). The CK3 bit cannot be set to 1 while the CPU is operating with the high-speed internal oscillator (MCM.MCS = 0).

Do not change the values of the MCM.MCM0 and CK2 to CK0 bits while the CPU is operating with the subclock.

Remark ×: don't care

(a) Example of setting high-speed internal oscillation (f_{RH}) → main clock (f_X) after reset release

After reset release, the high-speed internal oscillation (f_{RH}) is supplied as the main clock frequency (f_{XX}) while the main clock oscillation (f_X) is stopped.

An example of changing the oscillation clock of the main clock frequency (f_{XX}) from high-speed internal oscillation (f_{RH}) to main clock oscillation (f_X) is given below.

<1> PCC.MCK = 0: Enables oscillation by the main clock oscillator.
When the MCK bit is changed from 1 to 0, the oscillation stabilization time selection register (OSTS) automatically counts the oscillation stabilization time.

<2> OSTC.MSTS = 1?: The OSTS register[Note 2] is monitored by the OSTC register[Note 1]. When the oscillation stabilization time has ended, the OSTC.MSTS bit is set to 1.

<3> MCM.MCM0 = 1: The main clock oscillation (f_X) is supplied as the main clock frequency (f_{XX}) when the MCM0 bit of the main system clock mode register (MCM) is set to 1.

Notes 1. See **20.2 (4) Oscillation stabilization time count status register (OSTC)**.

2. See **20.2 (3) Oscillation stabilization time selection register (OSTS)**.

(b) Example of setting main clock operation → subclock operation

How to change the internal system clock (f_{CLK}) from the main clock (f_{XX}) to the subclock (f_{XT}) is described below.

Before making the following setting, select main clock oscillation (f_X) as the source clock of the main clock frequency (f_{XX}) by referring to **6.3 (1) (a) Example of setting high-speed internal oscillation (f_{RH}) → main clock (f_X) after reset release**. Confirm that the MCM.MCS bit is 1 after setting.

<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 with the main clock.

2. If the following conditions are not satisfied, change the CK2 to CK0 bits so that the conditions are satisfied, and then change to the subclock operation mode.

Internal system clock (f_{CLK}) > Subclock (f_{XT}: 32.768 KHz) × 4

Remark Internal system clock (f_{CLK}): Clock generated from the main clock (f_{XX}) by setting the CK2 to CK0 bits

[Description example]

```
     _DMA_DISABLE:
     clrl        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:
     setl        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 read in a closed loop.

(c) Example of setting subclock operation → main clock operation

How to change the internal system clock (f_{CLK}) from the subclock (f_{XT}) to the main clock (f_{XX}) is described below.

<1> MCK bit ← 0: Main clock starts oscillating

<2> Insert waits by the program and wait until the oscillation stabilization time of the main clock elapses. The OSTS register is monitored by the OSTC register. When the oscillation stabilization time has ended, the OSTC.MSTS bit is set to 1.

<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 or read the CLS bit to check if main clock operation has started.

Caution Enable operation of the on-chip peripheral functions operating with 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:
     clrl       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
     cmp        r0, r11
     bne                _WAIT_OST
<3>  st.b       r0, PRCMD[r0]
     clr1       3, PCC[r0]                -- CK3 ← 0
<4>  _CHECK_CLS :
     tst1       4, PCC[r0]                -- Wait until main clock operation starts.
     bnz        _CHECK_CLS
     _DMA_ENABLE:
     setl       0, DCHCn[r0]              -- DMA operation enabled.  n = 0 to 3
```

Remark The description above is simply an example. Note that in <4> above, the CLS bit is read in a closed loop.

(2) Internal oscillation 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 80H.

After reset: 80H R/W Address: FFFFF80CH

	7	6	5	4	3	2	<1>	<0>
RCM	RSTS[Note]	0	0	0	0	0	HRSTOP	RSTOP

RSTS[Note]	Status of high-speed internal oscillator
0	High-speed internal oscillator (f_{RH}: 8 MHz) stops or waits for oscillation stabilization
1	High-speed internal oscillator (f_{RH}: 8 MHz) oscillates

HRSTOP	Oscillation/stop of high-speed internal oscillator
0	High-speed internal oscillator (f_{RH}: 8 MHz) oscillates
1	High-speed internal oscillator (f_{RH}: 8 MHz) stopped

RSTOP	Oscillation/stop of low-speed internal oscillator
0	Low-speed internal oscillator (f_{RL}: 240 kHz) oscillates
1	Low-speed internal oscillator (f_{RL}: 240 kHz) stops

Note The RSTS bit is a read-only bit.

Cautions 1. The high-speed internal oscillator (f_{RH}) cannot be stopped while the CPU is operating with the high-speed internal oscillation clock (f_{RH}) (MCM.MCM0 bit = 1). Do not set the HRSTOP bit to 1.

2. The low-speed internal oscillator (f_{RL}) cannot be stopped while the CPU is operating with the low-speed internal oscillation clock (f_{RL}) (CCLS.CCLSF bit = 1). Do not set the RSTOP bit to 1.

3. The low-speed internal oscillator (f_{RL}) oscillates if the CCLS.CCLSF bit is set to 1 (when WDT overflow occurs during oscillation stabilization time) even when the RSTOP bit is set to 1. At this time, the RSTOP bit remains being set to 1.

(3) Main clock mode register (MCM)

The MCM register is an 8-bit register that selects the source clock of the main clock (f_{XX}) in the clock-through mode and indicates its status.

The MCM register is a special register and can be written only by a specific combination of sequences (see **3.4.8 Special registers**).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF860H

	7	6	5	4	3	2	<1>	<0>
MCM	0	0	0	0	0	0	MCS[Note]	MCM0

MCS[Note]	Status of main clock frequency (f_{XX})
0	Operating with high-speed internal oscillator (f_{RH}: 8 MHz)
1	Operating at main clock oscillation frequency (f_X)

MCM0	Selection of main clock frequency (f_{XX})
0	High-speed internal oscillator (f_{RH}: 8 MHz)
1	Main clock oscillation frequency (f_X)

Note The MCS bit is a read-only bit.

Cautions 1. Be sure to change the status of the MCM0 bit in the clock-through mode (PLLCTL.SELPLL = 0). Rewriting the MCM0 bit is prohibited if the clock output by PLL or SSCG (PLLTCL.SELPLL = 1) or subclock (PCC.CK3 = 1) is used as the CPU clock.

2. Rewriting this register is prohibited if oscillation of the selected clock is not stabilized.

3. If the low-speed internal oscillation clock (f_{RL}) (CCLS.CCLSF = 1) is used as the CPU clock, do not set the MCM0 bit to 1.

(4) 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	CCLSF

CCLSF	CPU operation clock status
0	Operating with main clock (f_X) or subclock (f_{XT}).
1	Operating with low-speed internal oscillation clock (f_{RL}: 240 kHz).

Note If WDT overflow occurs during oscillation stabilization time after a reset is released, the CCLSF bit is set to 1 and the reset value is 01H.

(5) Programmable clock mode register (PCLM)

The PCLM register is an 8-bit register that controls the PCL output.

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

After reset: 00H R/W Address: FFFFF82FH

	7	6	5	<4>	3	2	1	0
PCLM	0	0	0	PCLE	0	0	PCK1	PCK0

PCLE	Selection of PCL pin output operation
0	PCL pin output disabled (PCL pin is fixed to low level)
1	PCL pin output enabled

PCK1	PCK0	Selection of PLL output clock
0	0	$f_{PCL} = f_{PLSS}/4$
0	1	$f_{PCL} = f_{PLSS}/8$
1	0	$f_{PCL} = f_{PLSS}/16$
1	1	$f_{PCL} = f_{PLSS}/32$

Cautions 1. After setting the PCL function by using port-related control registers (PMC9, PFC9, and PFCE9 registers), set the PCLE bit to 1.

2. Set the PCLE bit to 1 only during PLL operation. Before stopping PLL, set the PCLE bit to 0.

Remark f_{PCL}: Programmable frequency

6.4 Operation

6.4.1 Setting CPU clock

Table 6-1 shows the registers and their bits necessary for setting a clock that can operate as the CPU clock (f_{CPU}).

Table 6-1. Setting/Status of CPU Clock (f_{CPU})

Operating Clock (f_{CPU})	MCS Bit of MCM Register	SELPLL Bit of PLLCTL Register	SELSSCG Bit of SSCGCTL Register	CK3 Bit of PCC Register	CCLSF Bit[Note 2] of CCLS Register
High-speed internal oscillation clock (f_{RH}) operation	0 (operates with f_{RH})	0 (clock-through mode)	X	0 (main clock operation)	0 (operates with f_{XX})
Main clock (f_{XX}) operation	1 (operates with f_X)				
PLL clock operation		1 (PLL/SSCG mode)	0[Note 1] (PLL mode)		
SSCG clock operation			1[Note 1] (SSCG mode)		
Low-speed internal oscillation clock (f_{RL}) operation		X	X		1 (operates with f_{RL})
Subclock (f_{XT}) operation		X	X	1 (subclock operation)	0 (operates with f_{XT})

Notes 1. In addition to setting the above registers, the option byte must be set to select a frequency to be used. For details, see **CHAPTER 27 OPTION BYTE FUNCTION**.

2. The CCLS.CCLSF bit indicates which clock operates as the CPU clock (f_{CPU}) and does not have to be set.

6.4.2 Operation of each clock

The following table shows the operation status of each clock.

Table 6-2. Operation Status of Each Clock

Operation Status		f_X Note 1, f_{PLLI}	f_{XT} Note 2	f_{RL} Note 3	f_{RH} Note 4	f_{PLLO}, f_{SSCGO}, f_{PCL}	f_{XX}	f_{CLK}	f_{CPU}
During reset		×	√	×	×	×	×	×	×
While high-speed internal oscillator (f_{RH}) is being set up after reset release		×	√	√	√	×	√	×	×
High-speed internal oscillator (f_{RH})	Operation	Operable	√	Operable	√	Operable	√	√	√
	HALT mode	Operable	√	Operable	√	Operable	√	√	×
	IDLE1 mode	Operable	√	Operable	√	Operable	×	×	×
	STOP mode	×	√	Operable	×	×	×	×	×
	Oscillation stabilization time after release of STOP	Operable	√	Operable	√	×	√	×	×
Main clock oscillation (f_{XX})	Operation	√	√	Operable	Operable	Operable	√	√	√
	HALT mode	√	√	Operable	Operable	Operable	√	√	×
	IDLE1 mode	√	√	Operable	Operable	Operable	×	×	×
	IDLE2 mode	√	√	Operable	Operable	×	×	×	×
	Setting up after release of IDLE2	√	√	Operable	Operable	×	×	×	×
	STOP mode	×	√	Operable	×	×	×	×	×
	Oscillation stabilization time after release of STOP	√	√	Operable	Operable	×	×	×	×
PLL, SSCG	Operation	√	√	Operable	Operable	√	√	√	√
	HALT mode	√	√	Operable	Operable	√	√	√	×
	IDLE1 mode	√	√	Operable	Operable	√	×	×	×
	IDLE2 mode	√	√	Operable	Operable	×	×	×	×
	Setting up after release of IDLE2	√	√	Operable	Operable	×	×	×	×
	STOP mode	×	√	Operable	×	×	×	×	×
	Oscillation stabilization time after release of STOP	√	√	Operable	Operable	×	×	×	×
Subclock oscillation (f_{XT})	Operation	Operable	√	Operable	Operable	Operable	Operable	√	√
	IDLE mode	Operable	√	Operable	Operable	Operable	×	×	×
Low-speed internal oscillator (f_{RL})	Operation	–	√	√	Operable	–	Operable	√	√
	HALT mode	–	√	√	Operable	–	Operable	√	×

Notes 1. Including the clock for watch timer, CSIB0, and watchdog timer

2. Including the clock for watch timer

3. Including the clock for watchdog timer and TMM

4. Including the clock for TMM

Remarks 1. √: Operable

×: Stopped

2. f_X: Main clock oscillation frequency

f_{PLLI}: PLL input clock frequency

f_{XT}: Subclock frequency

f_{RL}: Low-speed internal oscillation clock frequency

f_{RH}: High-speed internal oscillation clock frequency

f_{PLLO}: PLL output clock frequency

f_{SSCGO}: SSCG output clock frequency

f_{PCL}: PCL output clock frequency

f_{XX}: Main clock frequency

f_{CLK}: Internal system clock frequency

f_{CPU}: CPU clock frequency

6.4.3 Clock output function

The clock output function is used to output the internal system clock (f_{CLK}) from the CLKOUT pin.

The internal system clock (f_{CLK}) is selected by using the PCC.CK3 to PCC.CK0 bits.

The CLKOUT pin functions alternately as the PCM1 pin and functions as a clock output pin if so specified by the control register of port CM.

The status of the CLKOUT pin is the same as the internal system clock (f_{CLK}) in Table 6-2 and the pin can output the clock when it is in the operable status. It outputs a low level in the stopped status. However, the CLKOUT pin is in the port mode (PCM1 pin: input mode) after reset and until it is set in the output mode. Therefore, the status of the pin is Hi-Z.

6.5 PLL/SSCG Function

6.5.1 Overview

In the V850ES/Hx3, the following modes can be selected for the operating clock of the CPU and on-chip peripheral functions.

- Outputting oscillation frequency multiplied by 8 with PLL function
- Outputting oscillation frequency multiplied by 96 with SSCG function
- Clock-through mode

<R> • When PLL function is used (multiplied by 8): Input clock = 6 to 12 MHz (output = 12 to 32 MHz)
<R> • When SSCG function is used (multiplied by 96): Input clock = 4 to 16 MHz (output = 10.5 to 32 MHz[Note])
• Clock-through mode: Input clock = 4 to 16 MHz (output = 4 to 16 MHz)

Note Do not set a frequency exceeding 32 MHz, taking an increment of modulation rate into consideration.

6.5.2 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 00H.

After reset: 00H R/W Address: FFFFF82CH

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

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

PLLON	PLL operation stop register
0	PLL stopped
1	PLL operating (After PLL operation starts, a lockup time is required for frequency stabilization)

Cautions 1. When the PLLON bit is set to 0, the SELPLL bit is automatically set to 0 (clock-through mode).
2. The SELPLL bit can be set to 1 only when the PLL clock frequency is stabilized. If not (unlocked), "0" is written to the SELPLL bit if data is written to it.

(2) Lock register (LOCKR)

Phase lock occurs at a given frequency after the PLLCTL.PLLON bit has been changed from 0 to 1 or immediately after the IDLE2 or STOP mode has been released, and the time required for stabilization is the lockup time (frequency stabilization time). This state until stabilization is called the lockup status, and the stabilized state 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 01H.

After reset: 01H 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
- In IDLE2 or STOP mode
- Upon setting of PLL stop (setting of PLLCTL.PLLON bit to 0)
- Upon stopping main clock and using CPU with high-speed internal oscillation clock or subclock (setting of PCC.CK3 bit to 1 and setting of PCC.MCK bit to 1)

[Clear conditions]

- 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 is released (time set by the OSTS register) when the IDLE2 mode is set during PLL operation.

(3) 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	PLLS2	PLLS1	PLLS0

PLLS2	PLLS1	PLLS0	Selection of PLL lockup time
0	1	0	$2^{12}/f_X$
0	1	1	$2^{13}/f_X$
1	0	0	$2^{14}/f_X$
Other than above			Setting prohibited

Cautions 1. Set so that the lockup time is 800 μs or longer. The lockup time must be 1 ms or more when SSCG is used.

2. Do not change the PLLS register setting during the lockup period.

(4) SSCG control register (SSCGCTL)

The SSCGCTL register is an 8-bit register that controls the spread spectrum clock generator (SSCG).

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

After reset: 00H R/W Address: FFFFF3F0H

	7	6	5	4	3	2	<1>	<0>
SSCGCTL	0	0	0	0	0	0	SELSSCG	SSCGON

SELSSCG	Selection of clock to be output from multiplication block
0	PLL output ($f_{PLL} = f_{PLLO}$)
1	SSCG output ($f_{PLL} = f_{SSCGO}$)

SSCGON	SSCG function operation enable/disable
0	SSCG stopped
1	SSCG operating

Cautions 1. Write the SSCGCTL register bits when both PLL and SSCG stop (PLLCTL.PLLON bit = 0) or are in the lock status.

2. The SELSSCG bit can be set to 1 only when the SSCGON bit = 1. When the SSCGON bit is set to 0, the SELSSCG bit is automatically set to 0 (PLL output).

3. If the PLLCTL.PLLON bit is set to 0 or the main clock is stopped while the SSCGON bit = 1, SSCG stops operating.

4. When SSCG starts operating, time until SSCG is locked is required.

- **If the SSCGON bit is changed from 0 to 1 when the PLLCTL.PLLON bit = 1, make sure by software that the lockup time of SSCG (1 ms or more) elapses.**
- **If the PLLCTL.PLLON bit is changed from 0 to 1 after the SSCGON bit has been set to 1, set SSCG lockup time (1 ms or more) to the PLLS register (this is because the lockup time of SSCG is longer than that of PLL).**
- **Set a value two times the SSCG lockup time (1 ms or more) to the OSTS register.**

The relationship between the PLL/SSCG mode and PLLCTL.PLLON/SSCGCTL.SSCGON bit is shown below.

Table 6-3. Operation Condition of PLL/SSCG Mode

PLLCTL.PLLON Bit	SSCGCTL.SSCGON Bit	PLL Mode	SSCG Mode
0	0	Stops	Stops
0	1		
1	0	Operates	Stops
1	1		Operates

(5) SSCG frequency control register 0 (SFC0)

The SFC0 register is an 8-bit register that controls the SSCG frequency.

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: FFFFF3F1H

	7	6	5	4	3	2	1	0
SFC0	SFC07	SFC06	SFC05	SFC04	SFC03	SFC02	SFC01	SFC00

SFC07	Division ratio setting of clock (f_{SSCGI}) to be input to SSCG
0	$f_{SSCGI} = f_X$ (not divided)
1	$f_{SSCGI} = f_X/2$ (divided by two)

SFC06	SFC05	SFC04	SSCG input clock (f_{PFD})
0	0	0	f_{SSCGI}
0	0	1	$f_{SSCGI}/2$
0	1	0	$f_{SSCGI}/3$
0	1	1	$f_{SSCGI}/4$
1	0	0	$f_{SSCGI}/5$
1	0	1	$f_{SSCGI}/6$
1	1	0	$f_{SSCGI}/7$
1	1	1	$f_{SSCGI}/8$

SFC03	SFC02	SSCG output clock division ratio
0	0	Setting prohibited
0	1	Divided by two
1	0	Divided by four
1	1	Setting prohibited

SFC01	SFC00	Frequency range specification
0	0	0.87 MHz ≤ f_{PFD} < 1.00 MHz
0	1	1.00 MHz ≤ f_{PFD} < 1.22 MHz
1	0	1.22 MHz ≤ f_{PFD} < 1.45 MHz
1	1	1.45 MHz ≤ f_{PFD} ≤ 1.74 MH

Note SSCG stops when PLLCTL.PLLON = 0.

Caution Change the status of the SFC0 register only when SSCG is stopped (SSCGCTL.SSCGON = 0).
Writing the SFC0 register is invalid when SSCGCTL.SSCGON = 0.

Remark For an example of setting a clock in the SSCG mode, see **Table 6-4**.

(6) SSCG frequency control register 1 (SFC1)

The SFC1 register is an 8-bit register that controls the frequency modulation rate and modulation cycle of SSCG.

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: FFFFF3F2H

	7	6	5	4	3	2	1	0
SFC1	SFC17	SFC16	SFC15	SFC14	0	0	SFC11	SFC10

SFC17	Frequency modulation control
0	Frequency modulation disabled (not modulated)
1	Frequency modulation enabled (modulated)

SFC16	SFC15	SFC14	SSCG input clock (f_{PFD})
0	0	0	±0.5% (TYP.)
0	0	1	±1.0% (TYP.)
0	1	0	±2.0% (TYP.)
0	1	1	±3.0% (TYP.)
1	0	0	±4.0% (TYP.)
1	0	1	±5.0% (TYP.)
Other than above			Setting prohibited

SFC11	SFC10	Modulation cycle specification
0	0	About 40 kHz
0	1	About 50 kHz
1	0	About 60 kHz
1	1	Setting prohibited

Cautions 1. Change the status of the SFC1 register only when SSCG is stopped (SSCGCTL.SSCGON = 0).

Writing the SFC1 register is invalid when SSCGCTL.SSCGON = 1.

2. The SSCG output frequency (f_{SSCG}) changes as follows depending on the central value of the frequency modulation range specified by the SFC16 to SFC14 bits.

Make sure that the maximum frequency, including the frequency modulation rate, is not exceeded (see CHAPTER 29 to CHAPTER 32).

$f_{SSCG} = f_{SSCGO} \times (1 \pm$ (Frequency modulation rate))

3. Be sure to set bits 3 and 2 to "0".

(7) Selector operation control register 4 (SELCNT4)

The SELCNT4 register is an 8-bit register that allows the CPU to operate with SSCG and a peripheral function connected to prescaler 1 to operate without SSCG.

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: FFFFF3F8H

	7	6	5	4	3	2	1	0
SELCNT4	0	0	0	0	0	0	0	ISEL40

ISEL40	Selection of source clock to prescaler 1
0	f_{XX} (main clock)
1	f_{PLLO} (PLL output clock)

Cautions 1. Setting the ISEL40 bit to 1 is prohibited when PLL is stopped (PLLCTL.PLLON = 0). At this time, data cannot be written to the SELCNT4 register (a write access to the register is invalid).

2. Be sure to set the ISEL40 bit to 0 before changing the PLLON bit from 1 to 0 while the CPU is operating with SSCG.

Even if the PLLON bit is changed from 1 to 0 while the ISEL40 bit = 1, clock supply to prescaler 1 is stopped because f_{PLLO} (clock output by PLL) is continuously supplied to prescaler 1. At this time, the ISEL40 bit is "0" when it is read.

3. The ISEL40 bit can be set (1) only when the CPU operates with SSCG. If SSCG is not used (SSCGCTL.SSCGON bit = 0), set the ISEL40 bit to 0.

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

6.5.3 Using PLL

(1) When PLL is used

- Determines a desired frequency by referring Table 6-4 and sets the option byte (OB_7B.PLLO). For details, see **CHAPTER 27 OPTION BYTE FUNCTION**.
- The PLL is stopped (PLLCTL.PLLON bit = 0) after the reset has been released. Operate the main clock (see **6.3 (1) (a) Example of setting high-speed internal oscillation (f_{RH}) → main clock (f_X) after reset release**) and then operate PLL.

  To enable PLL operation, first set the PLLON bit to 1, and then set the SELPLL bit to 1 after the LOCKR.LOCK bit = 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 IDLE2 or STOP mode regardless of the setting and is restored from IDLE2 or STOP mode to the status before transition. The time required for restoration is as follows.

  (a) When transiting to IDLE2 or STOP mode from the clock through mode
  - Set the OSTS register so that the oscillation stabilization time is 54 μs or longer.

  (b) When shifting to the IDLE2 or STOP mode while remaining in the PLL operation mode
  - STOP mode: Set the OSTS register so that the oscillation stabilization time is 1600 μs or longer.
  - IDLE2 mode: Set the OSTS register so that the setup time is 800 μs or longer.

  (c) When shifting to the IDLE2 or STOP mode while remaining in the SSCG operation mode
  - STOP mode: Set the OSTS register so that the oscillation stabilization time is 2 ms or longer.
  - IDLE2 mode: Set the OSTS register so that the setup time is 1 ms or longer.

  When shifting 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 set after the reset has been released. The PLL is stopped (PLLON bit = 0).

The relationships of the main clock oscillation frequency (fx), PLLCTL.SELPLL bit, option byte, and main clock frequency in the PLL mode are shown below.

Table 6-4. Example of Setting Clock in PLL Mode

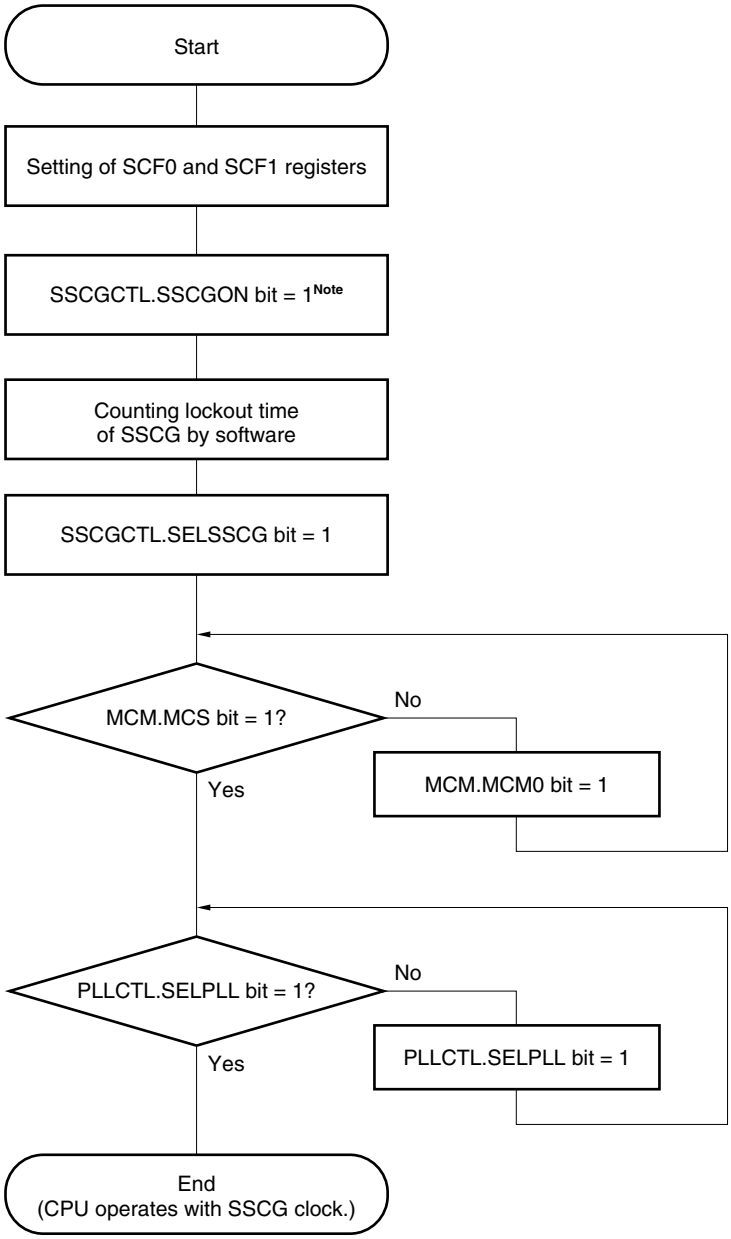
fx [MHz]	PLLCTL.SELPLL	OB_7B.PLLO[Note]	fxx [MHz]	Operating Clock
6	0	×	6	Clock-through mode
	1	0	24	PLL mode
	1	1	12	
8	0	×	8	Clock-through mode
	1	0	32	PLL mode
	1	1	16	
10	0	×	10	Clock-through mode
	1	0	Setting prohibited	PLL mode
	1	1	20	
12	0	×	12	Clock-through mode
	1	0	Setting prohibited	PLL mode
	1	1	24	

Note Set this bit in advance by using the option byte (see **CHAPTER 27 OPTION BYTE FUNCTION**).

6.5.4 Using SSCG

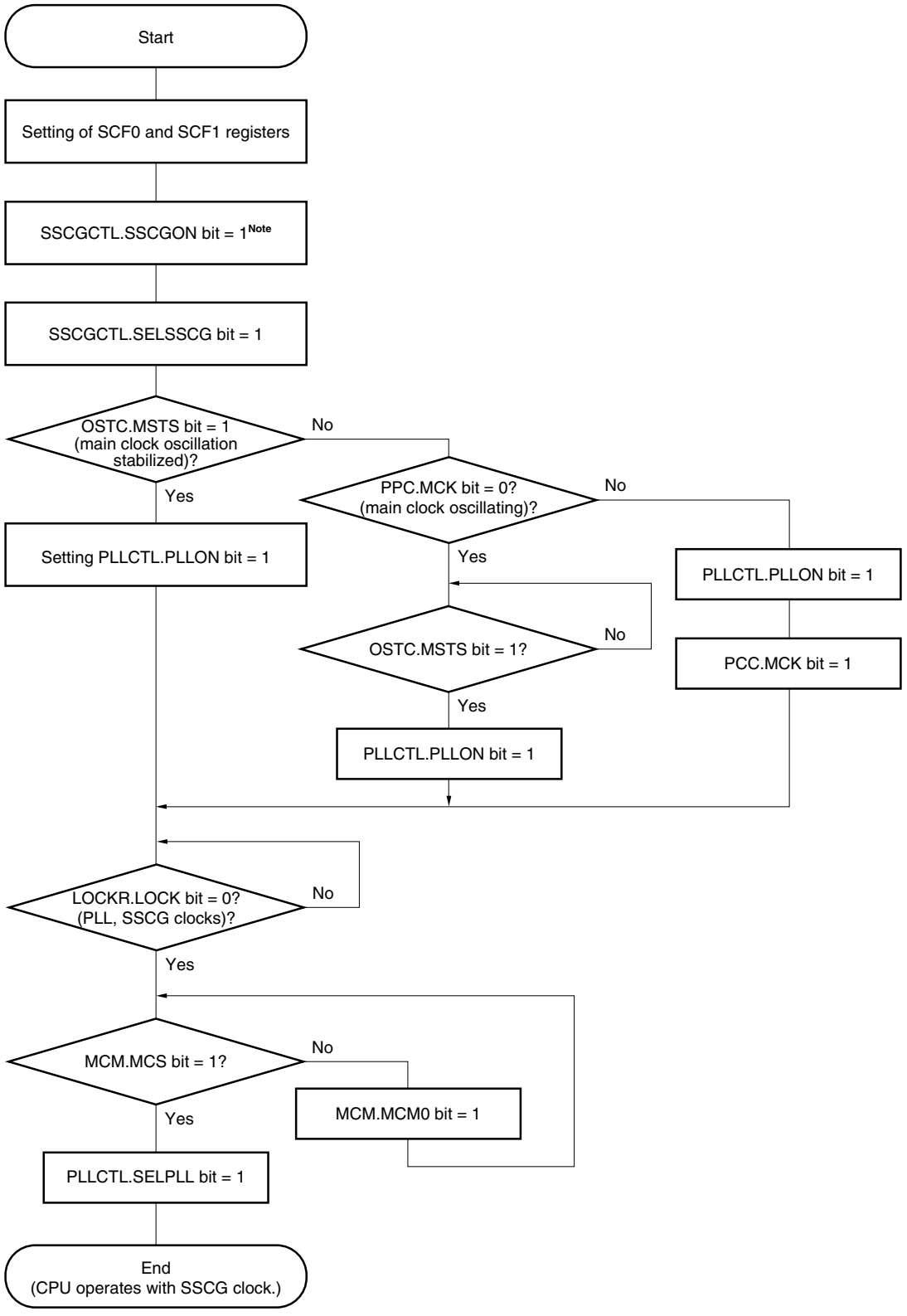
The flowcharts in Figures 6-3 and 6-4 illustrate how to use SSCG.

Figure 6-3. Procedure for Setting SSCG (When PLL Is Operating (Locked) (LOCKR.LOCK = 0)



Note After setting the SFC0 and SFC1 registers, check by software if a setup time of at least 1 μs elapses before the SSCGON bit is set to 1.

Figure 6-4. Procedure for Setting SSCG (When PLL Is Stopped (LOCKR.LOCK = 1))



Note After setting the SFC0 and SFC1 registers, check by software if a setup time of at least 1 μs elapses before the SSCGON bit is set to 1.

The relationships of the main clock oscillation frequency (fx), SFC0 register, option byte, and main clock frequency in the SSCG mode are shown below.

Table 6-5. Example of Setting Clock in SSCG Mode

fx (MHz)	SFC0 Register				f_{SSCGO} (MHz)	Main Clock Frequency (f_{XX})	
	SFC07 to SFC04 Bits	f_{PFD} (MHz)	SFC01, SFC00 Bits	SFC03, SFC02 Bits		PLLO = 0[Note]	PLLO = 1[Note]
4	0010B	1.333	10B	10B	32	Setting prohibited	16
	0011B	1	01B	01B	48	Setting prohibited	24
				10B	24	24	12
5	0010B	1.667	11B	10B	40	Setting prohibited	20
	0011B	1.25	10B	10B	30	30	15
	0100B	1	01B	01B	48	Setting prohibited	24
				10B	24	24	12
6	0011B	1.5	11B	10B	36	Setting prohibited	18
	0100B	1.2	01B	10B	28.8	28.8	14.4
	0101B	1	01B	01B	48	Setting prohibited	24
				10B	24	24	12
8	0100B	1.6	11B	10B	38.4	Setting prohibited	19.2
	0101B	1.333	10B	10B	32	Setting prohibited	16
	0110B	1.143	01B	10B	27.4	27.4	13.7
	0111B	1	01B	01B	48	Setting prohibited	24
				10B	24	24	12
10	0101B	1.667	11B	10B	40	Setting prohibited	20
	0110B	1.429	10B	10B	34.2	Setting prohibited	17.1
	0111B	1.25	01B	10B	30	30	15
	1100B	1	01B	01B	48	Setting prohibited	24
				10B	24	24	12
12	0110B	1.714	11B	10B	41.1	Setting prohibited	20.5
	0111B	1.5	11B	10B	36	Setting prohibited	18
	1100B	1.2	01B	10B	28.8	28.8	14.4
	1101B	1	01B	01B	48	Setting prohibited	24
				10B	24	24	12
14	1100B	1.4	10B	10B	33.6	Setting prohibited	16.8
	1101B	1.167	01B	10B	28	28	14
	1110B	1	01B	01B	48	Setting prohibited	24
				10B	24	24	12
	1111B	0.875	00B	01B	42	Setting prohibited	21
				10B	21	21	10.5
16	1100B	1.6	11B	10B	38.4	Setting prohibited	19.2
	1101B	1.333	10B	10B	32	Setting prohibited	16
	1110B	1.143	01B	10B	27.4	27.4	13.7
	1111B	1	01B	01B	48	Setting prohibited	24
				10B	24	24	12

Note Set this bit in advance by using the option byte (see **CHAPTER 27 OPTION BYTE FUNCTION**).

CHAPTER 7 16-BIT TIMER/EVENT COUNTER AA (TAA)

The number of TAAs in the V850ES/Hx3 is shown below.

Product Name	V850ES/HE3	V850ES/HF3	V850ES/HG3	V850ES/HJ3
Number of channels	5 channels (TAA0 to TAA4)	5 channels (TAA0 to TAA4)	5 channels (TAA0 to TAA4)	5 channels (TAA0 to TAA4)

7.1 Overview

An outline of TAAn is shown below.

- Clock selection: 8 ways
- Capture/trigger input pins: 2
- External event count input pins: 1
- External trigger input pins: 1
- Timer/counters: 1
- Capture/compare registers: 2
  (32-bit capture function available by using a cascade connection with TAA0 and TAA1, TAA2 and TAA3.)
- Capture/compare match interrupt request signals: 2
- Timer output pins: 2

Remark n = 0 to 4

7.2 Functions

TAAn has the following functions.

- Interval timer
- External event counter
- External trigger pulse output
- One-shot pulse output
- PWM output
- Free-running timer
- Pulse width measurement
- Timer tuned function

Remark n = 0 to 4

7.3 Configuration

TAAn includes the following hardware.

Table 7-1. Configuration of TAAn

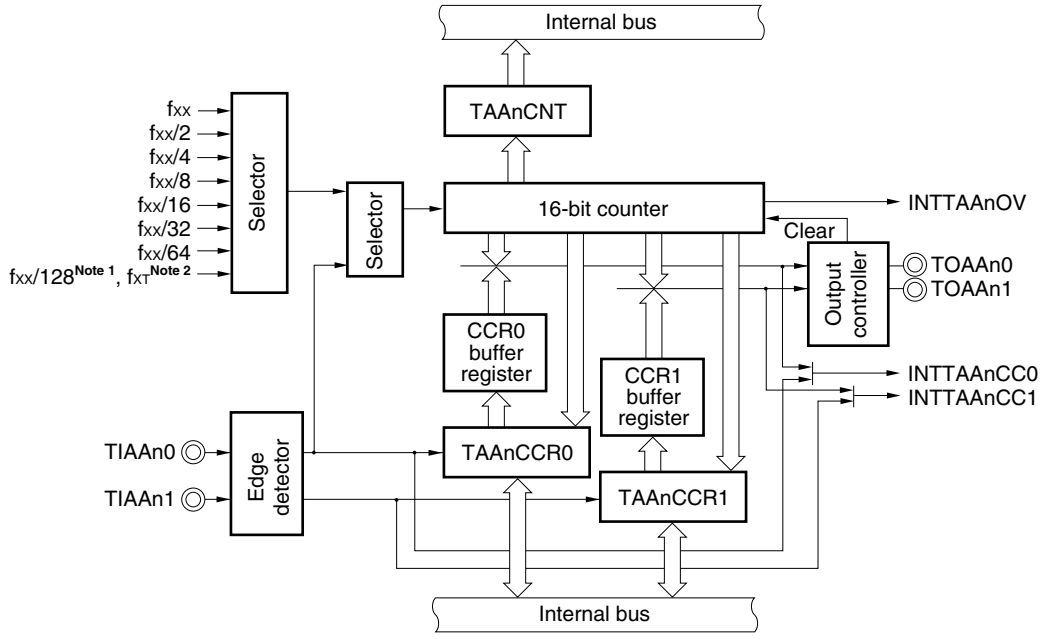
Item	Configuration
Timer register	16-bit counter
Registers	TAAn capture/compare registers 0, 1 (TAAnCCR0, TAAnCCR1) TAAn counter read buffer register (TAAnCNT) CCR0, CCR1 buffer registers
Timer inputs	2 (TIAAn0[Note 1], TIAAn1 pins)
Timer outputs	2 (TOAAn0, TOAAn1 pins)
Control registers[Note 2]	TAAn control registers 0, 1 (TAAnCTL0, TAAnCTL1) TAAn I/O control registers 0 to 2 (TAAnIOC0 to TAAnIOC2) TAAn option registers 0, 1 (TAAnOPT0, TAAnOPT1)

Notes 1. The TIAAn0 pin functions alternately as a capture trigger input signal, external event count input signal, and external trigger input signal.

2. When using the functions of the TIAAn0, TIAAn1, TOAAn0, and TOAAn1 pins, see **Table 4-25 Using Port Pin as Alternate-Function Pin**.

Remark n = 0 to 4

Figure 7-1. Block Diagram of TAAn



Notes 1. TAA0, TAA2, TAA4

2. TAA1, TAA3 (Counting operation cannot be performed with the subclock when the main clock is stopped.)

Remark f_{XX}: Main clock frequency
f_{XT}: Subclock frequency

(1) 16-bit counter

This 16-bit counter can count internal clocks or external events.

The count value of this counter can be read by using the TAAnCNT register.

When the TAAnCTL0.TAAnCE bit = 0, the value of the 16-bit counter is FFFFH. If the TAAnCNT register is read at this time, 0000H is read.

Reset sets the TAAnCE bit to 0. Therefore, the 16-bit counter is set to FFFFH.

(2) CCR0 buffer register

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

When the TAAnCCR0 register is used as a compare register, the value written to the TAAnCCR0 register is transferred to the CCR0 buffer register. When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTAAnCC0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is cleared to 0000H after reset, as the TAAnCCR0 register is cleared to 0000H.

(3) CCR1 buffer register

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

When the TAAnCCR1 register is used as a compare register, the value written to the TAAnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTAAnCC1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is cleared to 0000H after reset, as the TAAnCCR1 register is cleared to 0000H.

(4) Edge detector

This circuit detects the valid edges input to the TIAAn0 and TIAAn1 pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TAAnIOC1 and TAAnIOC2 registers.

(5) Output controller

This circuit controls the output of the TOAAn0 and TOAAn1 pins. The output controller is controlled by the TAAnIOC0 register.

(6) Selector

This selector selects the count clock for the 16-bit counter. Eight types of internal clocks or an external event can be selected as the count clock.

7.4 Registers

The registers that control TAAn are as follows.

- TAAn control register 0 (TAAnCTL0)
- TAAn control register 1 (TAAnCTL1)
- TAAn I/O control register 0 (TAAnIOC0)
- TAAn I/O control register 1 (TAAnIOC1)
- TAAn I/O control register 2 (TAAnIOC2)
- TAAn option register 0 (TAAnOPT0)
- TAAn option register 1 (TAAnOPT1)
- TAAn capture/compare register 0 (TAAnCCR0)
- TAAn capture/compare register 1 (TAAnCCR1)
- TAAn counter read buffer register (TAAnCNT)

Remarks 1. When using the functions of the TIAAn0, TIAAn1, TOAAn0, and TOAAn1 pins, see **Table 4-25 Using Port Pin as Alternate-Function Pin**.

2. n = 0 to 4

(1) TAAn control register 0 (TAAnCTL0)

The TAAnCTL0 register is an 8-bit register that controls the operation of TAAn.

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

Reset sets this register to 00H.

The same value can always be written to the TAAnCTL0 register by software.

After reset: 00H R/W Address: TAA0CTL0 FFFFF590H, TAA1CTL0 FFFFF5A0H, TAA2CTL0 FFFFF5B0H, TAA3CTL0 FFFFF5C0H, TAA4CTL0 FFFFF5D0H

	7	6	5	4	3	2	1	0
TAAnCTL0 (n = 0 to 4)	TAAnCE	0	0	0	0	TAAnCKS2	TAAnCKS1	TAAnCKS0

TAAnCE	TAAn operation control
0	TAAn operation disabled (TAAn reset asynchronously[Note 1]).
1	TAAn operation enabled. TAAn operation started.

TAAnCKS2	TAAnCKS1	TAAnCKS0	Internal count clock selection	
			n = 0, 2, 4	n = 1, 3
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	$f_{XX}/128$	f_{XT}[Note 2]

Notes 1. TAAn0PT0.TAAnOVF bit, 16-bit counter, timer output (TOAAn0, TOAAn1 pins)

2. Counting operation cannot be performed with the subclock when the main clock is stopped.

Cautions 1. Set the TAAnCKS2 to TAAnCKS0 bits when the TAAnCE bit = 0. When the value of the TAAnCE bit is changed from 0 to 1, the TAAnCKS2 to TAAnCKS0 bits can be set simultaneously.

2. Be sure to set bits 3 to 6 to "0".

Remark f_{XX}: Main clock frequency

f_{XT}: Subclock frequency

(2) TAAn control register 1 (TAAnCTL1)

The TAAnCTL1 register is an 8-bit register that controls the operation of TAAn.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: TAA0CTL1 FFFFF591H, TAA1CTL1 FFFFF5A1H, TAA2CTL1 FFFFF5B1H, TAA3CTL1 FFFFF5C1H, TAA4CTL1 FFFFF5D1H

TAAnCTL1 (n = 0 to 4)

7	6	5	4	3	2	1	0
TAAnSYE	TAAnEST	TAAnEEE	0	0	TAAnMD2	TAAnMD1	TAAnMD0

TAAnSYE	Tuned operation mode enable control
0	Independent operation mode (asynchronous operation mode)
1	Tuned operation mode (specification of slave operation) In this mode, timer AA can operate in synchronization with a master timer. Master timer / Slave timer: TAA0 / TAA1; TAA2 / TAA3; TAB0 / TAA4 For the tuned operation mode, see **7.6 Timer Tuned Operation Function**. **Caution Be sure to clear the TAA0SYE and TAA2SYE bits to 0.**

TAAnEST	Software trigger control
0	–
1	Generate a valid signal for external trigger input. • In one-shot pulse output mode: A one-shot pulse is output with writing 1 to the TAAnEST bit as the trigger. • In external trigger pulse output mode: A PWM waveform is output with writing 1 to the TAAnEST bit as the trigger.

Cautions 1. The TAAnEST 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. Be sure to set bits 3 and 4 to "0".

(2/2)

TAAnEEE	Count clock selection
0	Disable operation with external event count input. (Perform counting with the count clock selected by the TAAnCTL0.TAAnCK0 to TAAnCK2 bits.)
1	Enable operation with external event count input. (Perform counting at the valid edge of the external event count input signal.)
The TAAnEEE bit selects whether counting is performed with the internal count clock or the valid edge of the external event count input.	

TAAnMD2	TAAnMD1	TAAnMD0	Timer mode selection
0	0	0	Interval timer mode
0	0	1	External event count mode
0	1	0	External trigger pulse 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. External event count input is selected in the external event count mode regardless of the value of the TAAnEEE bit.

2. Set the TAAnEEE and TAAnMD2 to TAAnMD0 bits when the TAAnCTL0.TAAnCE bit = 0. (The same value can be written when the TAAnCE bit = 1.) The operation is not guaranteed when rewriting is performed with the TAAnCE bit = 1. If rewriting was mistakenly performed, clear the TAAnCE bit to 0 and then set the bits again.

(3) TAAn I/O control register 0 (TAAnIOC0)

The TAAnIOC0 register is an 8-bit register that controls the timer output (TOAAn0, TOAAn1 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: TAA0IOC0 FFFFF592H, TAA1IOC0 FFFFF5A2H, TAA2IOC0 FFFFF5B2H, TAA3IOC0 FFFFF5C2H, TAA4IOC0 FFFFF5D2H

TAAnIOC0 (n = 0 to 4)

7	6	5	4	3	2	1	0
0	0	0	0	TAAnOL1	TAAnOE1	TAAnOL0	TAAnOE0

TAAnOL1	TOAAn1 pin output level setting[Note]
0	TOAAn1 pin output starts at high level
1	TOAAn1 pin output starts at low level

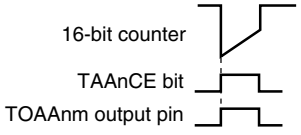
TAAnOE1	TOAAn1 pin output setting
0	Timer output disabled • When TAAnOL1 bit = 0: Low level is output from the TOAAn1 pin • When TAAnOL1 bit = 1: High level is output from the TOAAn1 pin
1	Timer output enabled (a square wave is output from the TOAAn1 pin).

TAAnOL0	TOAAn0 pin output level setting[Note]
0	TOAAn0 pin output starts at high level
1	TOAAn0 pin output starts at low level

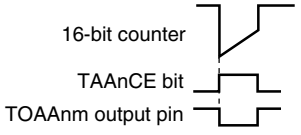
TAAnOE0	TOAAn0 pin output setting
0	Timer output disabled • When TAAnOL0 bit = 0: Low level is output from the TOAAn0 pin • When TAAnOL0 bit = 1: High level is output from the TOAAn0 pin
1	Timer output enabled (a square wave is output from the TOAAn0 pin).

Note The output level of the timer output pin (TOAAnm) specified by the TAAnOLm bit is shown below.

• When TAAnOLm bit = 0



• When TAAnOLm bit = 1



Cautions 1. Rewrite the TAAnOL1, TAAnOE1, TAAnOL0, and TAAnOE0 bits when the TAAnCTL0.TAAnCE bit = 0. (The same value can be written when the TAAnCE bit = 1.) If rewriting was mistakenly performed, clear the TAAnCE bit to 0 and then set the bits again.

2. Even if the TAAnOLm bit is manipulated when the TAAnCE and TAAnOEm bits are 0, the TOAAnm pin output level varies.

Remark n = 0 to 4, m = 0, 1

(4) TAAn I/O control register 1 (TAAnIOC1)

The TAAnIOC1 register is an 8-bit register that controls the valid edge of the capture trigger input signals (TIAAn0, TIAAn1 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: TAA0IOC1 FFFFF593H, TAA1IOC1 FFFFF5A3H, TAA2IOC1 FFFFF5B3H, TAA3IOC1 FFFFF5C3H. TAA4IOC1 FFFFF5D3H

	7	6	5	4	3	2	1	0
TAAnIOC1 (n = 0 to 4)	0	0	0	0	TAAnIS3	TAAnIS2	TAAnIS1	TAAnIS0

TAAnIS3	TAAnIS2	Capture trigger input signal (TIAAn1 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

TAAnIS1	TAAnIS0	Capture trigger input signal (TIAAn0 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 TAAnIS3 to TAAnIS0 bits when the TAAnCTL0.TAAnCE bit = 0. (The same value can be written when the TAAnCE bit = 1.) If rewriting was mistakenly performed, clear the TAAnCE bit to 0 and then set the bits again.

2. The TAAnIS3 to TAAnIS0 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) TAAn I/O control register 2 (TAAnIOC2)

The TAAnIOC2 register is an 8-bit register that controls the valid edge of the external event count input signal (TIAAn0 pin) and external trigger input signal (TIAAn0 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: TAA0IOC2 FFFFF594H, TAA1IOC2 FFFFF5A4H, TAA2IOC2 FFFFF5B4H, TAA3IOC2 FFFFF5C4H. TAA4IOC2 FFFFF5D4H

TAAnIOC2 (n = 0 to 4)

7	6	5	4	3	2	1	0
0	0	0	0	TAAnEES1	TAAnEES0	TAAnETS1	TAAnETS0

TAAnEES1	TAAnEES0	External event count input signal (TIAAn0 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

TAAnETS1	TAAnETS0	External trigger input signal (TIAAn0 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 TAAnEES1, TAAnEES0, TAAnETS1, and TAAnETS0 bits when the TAAnCTL0.TAAnCE bit = 0. (The same value can be written when the TAAnCE bit = 1.) If rewriting was mistakenly performed, clear the TAAnCE bit to 0 and then set the bits again.

2. The TAAnEES1 and TAAnEES0 bits are valid only when the TAAnCTL1.TAAnEEE bit = 1 or when the external event count mode (TAAnCTL1.TAAnMD2 to TAAnCTL1.TAAnMD0 bits = 001) has been set.

3. The TAAnETS1 and TAAnETS0 bits are valid only when the external trigger pulse output mode (TAAnCTL1.TAAnMD2 to TAAnCTL1.TAAnMD0 bits = 010) or the one-shot pulse output mode (TAAnCTL1.TAAnMD2 to TAAnCTL1.TAAnMD0 = 011) is set.

(6) TAAn option register 0 (TAAnOPT0)

The TAAnOPT0 register is an 8-bit register used to set the capture/compare operation and detect an overflow.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TAA0OPT0 FFFFF595H, TAA1OPT0 FFFFF5A5H, TAA2OPT0 FFFFF5B5H, TAA3OPT0 FFFFF5C5H, TAA4OPT0 FFFFF5D5H

	7	6	5	4	3	2	1	0
TAAnOPT0 (n = 0 to 4)	0	0	TAAnCCS1	TAAnCCS0	0	0	0	TAAnOVF

TAAnCCS1	TAAnCCR1 register capture/compare selection
0	Compare register selected
1	Capture register selected
The TAAnCCS1 bit setting is valid only in the free-running timer mode.	

TAAnCCS0	TAAnCCR0 register capture/compare selection
0	Compare register selected
1	Capture register selected
The TAAnCCS0 bit setting is valid only in the free-running timer mode.	

TAAnOVF	TAAn overflow detection flag
Set (1)	Overflow occurred
Reset (0)	TAAnOVF bit 0 written or TAAnCTL0.TAAnCE bit = 0
• The TAAnOVF bit is set to 1 when the 16-bit counter count value overflows from FFFFH to 0000H in the free-running timer mode or the pulse width measurement mode. • An interrupt request signal (INTTAAnOV) is generated at the same time that the TAAnOVF bit is set to 1. The INTTAAnOV signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode. • The TAAnOVF bit is not cleared even when the TAAnOVF bit or the TAAnOPT0 register are read when the TAAnOVF bit = 1. • The TAAnOVF bit can be both read and written, but the TAAnOVF bit cannot be set to 1 by software. Writing 1 has no influence on the operation of TAAn.	

Cautions 1. Rewrite the TAAnCCS1 and TAAnCCS0 bits when the TAAnCE bit = 0. (The same value can be written when the TAAnCE bit = 1.) If rewriting was mistakenly performed, clear the TAAnCE bit to 0 and then set the bits again.

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

(7) TAAn option register 1 (TAAnOPT1)

The TAAnOPT1 register is an 8-bit register that controls a 32-bit capture function by cascade connection.

Rewriting this register is prohibited while the timer is operating (TAAnCTL0.TAAnCE = 1).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TAA1OPT1 FFFFF5ADH, TAA3OPT1 FFFFF5CDH

	<7>	6	5	4	3	2	1	0
TAAnOPT1 (n = 1, 3)	TAAnCSE	0	0	0	0	0	0	0

TAAnCSE	Cascade control
0	Unit operation or operation as lower side of cascade function
1	Operation as higher side of cascade function

Cautions 1. Cascade connection and timer tuned operation cannot be used together. Be sure to set TAAnCTL1.TAAnSYE to 0 for cascade connection.

2. For cascade connection, set the free-running timer mode and use the TAAnCCR0 and TAAnCCR1 registers as capture registers. For details of cascade connection, see 7.7 Cascade Connection.

(8) TAAn capture/compare register 0 (TAAnCCR0)

The TAAnCCR0 register can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TAAnOPT0.TAAnCCS0 bit. In the pulse width measurement mode, the TAAnCCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TAAnCCR0 register can be read or written during operation.

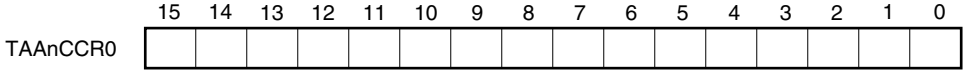
This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

Caution Accessing the TAAnCCR0 register is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- **When the CPU operates with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the internal oscillation clock**

After reset: 0000H R/W Address: TAA0CCR0 FFFFF596H, TAA1CCR0 FFFFF5A6H, TAA2CCR0 FFFFF5B6H, TAA3CCR0 FFFFF5C6H, TAA4CCR0 FFFFF5D6H



(a) Function as compare register

The TAAnCCR0 register can be rewritten even when the TAAnCTL0.TAAnCE bit = 1.

The set value of the TAAnCCR0 register is transferred to the CCR0 buffer register. When the value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTAAnCC0) is generated. If TOAAn0 pin output is enabled at this time, the output of the TOAAn0 pin is inverted.

When the TAAnCCR0 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 TAAnCCR0 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 TAAnCCR0 register if the valid edge of the capture trigger input pin (TIAAn0 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TAAnCCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIAAn0) is detected.

Even if the capture operation and reading the TAAnCCR0 register conflict, the correct value of the TAAnCCR0 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 7-2. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write
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	–

(9) TAAn capture/compare register 1 (TAAnCCR1)

The TAAnCCR1 register can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TAAnOPT0.TAAnCCS1 bit. In the pulse width measurement mode, the TAAnCCR1 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TAAnCCR1 register can be read or written during operation.

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

Reset sets this register to 0000H.

Caution Accessing the TAAnCCR1 register is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- **When the CPU operates with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the internal oscillation clock**

After reset: 0000H R/W Address: TAA0CCR1 FFFFF598H, TAA1CCR1 FFFFF5A8H, TAA2CCR1 FFFFF5B8H, TAA3CCR1 FFFFF5C8H, TAA4CCR1 FFFFF5D8H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAAnCCR1																

(n = 0 to 4)

(a) Function as compare register

The TAAnCCR1 register can be rewritten even when the TAAnCTL0.TAAnCE bit = 1.

The set value of the TAAnCCR1 register is transferred to the CCR1 buffer register. When the value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTAAnCC1) is generated. If TOAAn1 pin output is enabled at this time, the output of the TOAAn1 pin is inverted.

(b) Function as capture register

When the TAAnCCR1 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 TAAnCCR1 register if the valid edge of the capture trigger input pin (TIAAn1 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TAAnCCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIAAn1) is detected.

Even if the capture operation and reading the TAAnCCR1 register conflict, the correct value of the TAAnCCR1 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 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	–

(10) TAAn counter read buffer register (TAAnCNT)

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

If this register is read when the TAAnCTL0.TAAnCE bit = 1, the count value of the 16-bit timer can be read.

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

The value of the TAAnCNT register is cleared to 0000H when the TAAnCE bit = 0. If the TAAnCNT register is read at this time, the value of the 16-bit counter (FFFFH) is not read, but 0000H is read.

The value of the TAAnCNT register is cleared to 0000H after reset, as the TAAnCE bit is cleared to 0.

Caution Accessing the TAAnCNT register is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- **When the CPU operates with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the internal oscillation clock**

After reset: 0000H R Address: TAA0CNT FFFFF59AH, TAA1CNT FFFFF5AAH, TAA2CNT FFFFF5BAH, TAA3CNT FFFFF5CAH, TAA4CNT FFFFF5DAH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAAnCNT (n = 0 to 4)																

7.5 Operation

TAAn can perform the following operations.

Operation	TAAnCTL1.TAAnEST Bit (Software Trigger Bit)	TIAAn0 Pin (External Trigger Input)	Capture/Compare Register Setting	Compare Register Write
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode[Note 1]	Invalid	Invalid	Compare only	Anytime write
External trigger pulse output mode[Note 2]	Valid	Valid	Compare only	Batch write
One-shot pulse output mode[Note 2]	Valid	Valid	Compare only	Anytime write
PWM output mode	Invalid	Invalid	Compare only	Batch write
Free-running timer mode	Invalid	Invalid	Switching enabled	Anytime write
Pulse width measurement mode[Note 2]	Invalid	Invalid	Capture only	Not applicable

Notes 1. To use the external event count mode, specify that the valid edge of the TIAAn0 pin capture trigger input is not detected (by clearing the TAAnIOC1.TAAnIS1 and TAAnIOC1.TAAnIS0 bits to “00”).

2. When using the external trigger pulse output mode, one-shot pulse output mode, and pulse width measurement mode, select the internal clock as the count clock (by clearing the TAAnCTL1.TAAnEEE bit to 0).

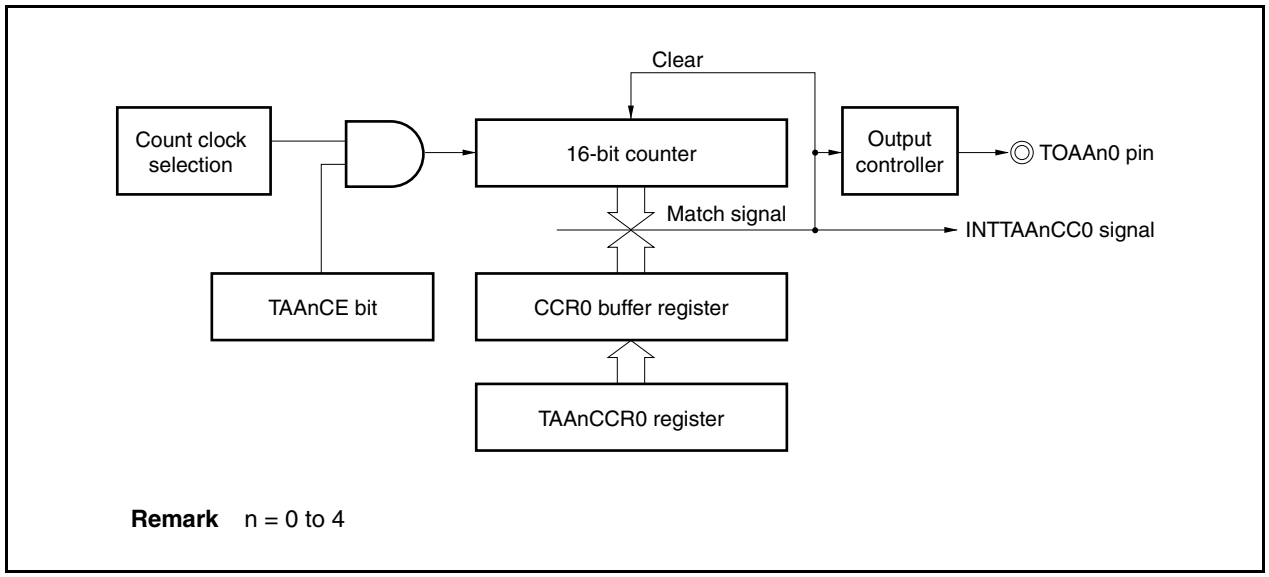
Remark n = 0 to 4

7.5.1 Interval timer mode (TAAnMD2 to TAAnMD0 bits = 000)

In the interval timer mode, an interrupt request signal (INTTAAnCC0) is generated at the specified interval if the TAAnCTL0.TAAnCE bit is set to 1. A square wave whose half cycle is equal to the interval can be output from the TOAAn0 pin.

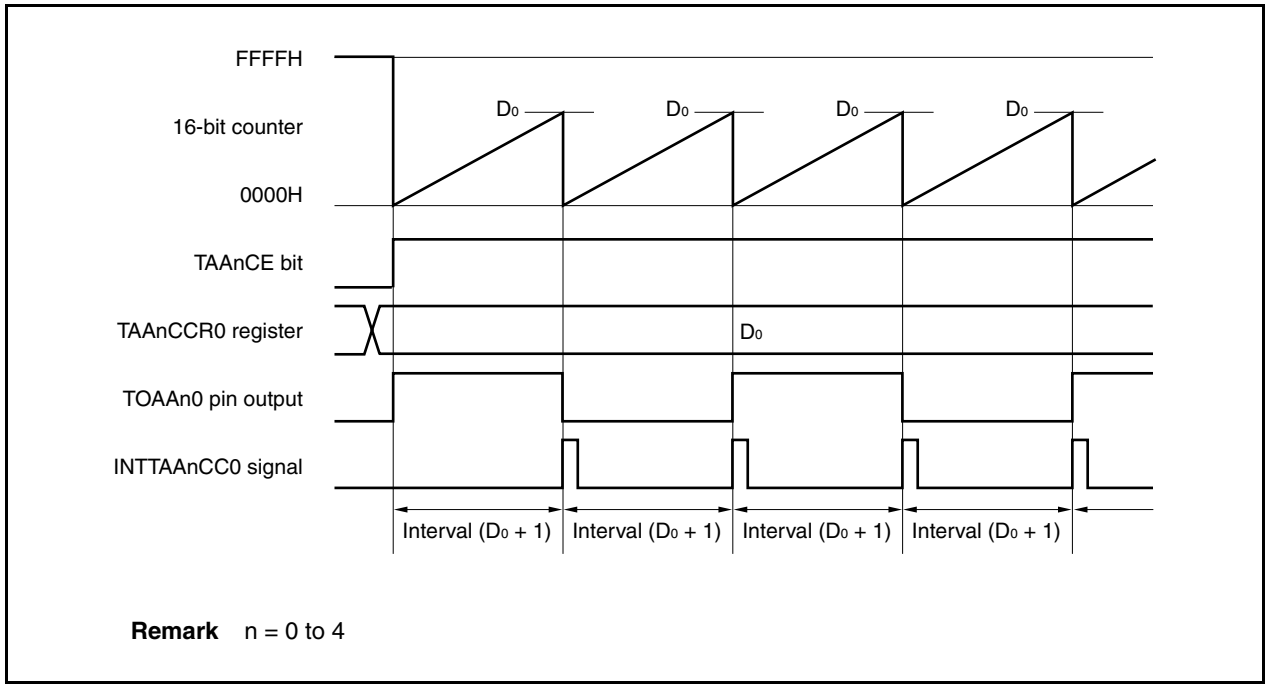
Usually, the TAAnCCR1 register is not used in the interval timer mode.

Figure 7-2. Configuration of Interval Timer



Remark n = 0 to 4

Figure 7-3. Basic Timing of Operation in Interval Timer Mode



Remark n = 0 to 4

When the TAAnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H in synchronization with the count clock, and the counter starts counting. At this time, the output of the TOAAn0 pin is inverted. Additionally, the set value of the TAAnCCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, the output of the TOAAn0 pin is inverted, and a compare match interrupt request signal (INTTAAnCC0) is generated.

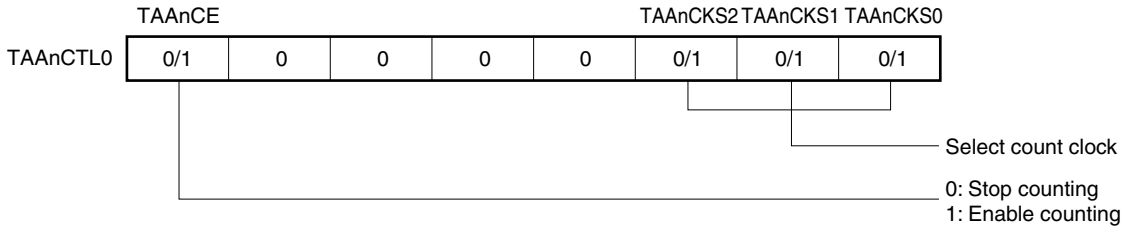
The interval can be calculated by the following expression.

Interval = (Set value of TAAnCCR0 register + 1) × Count clock cycle

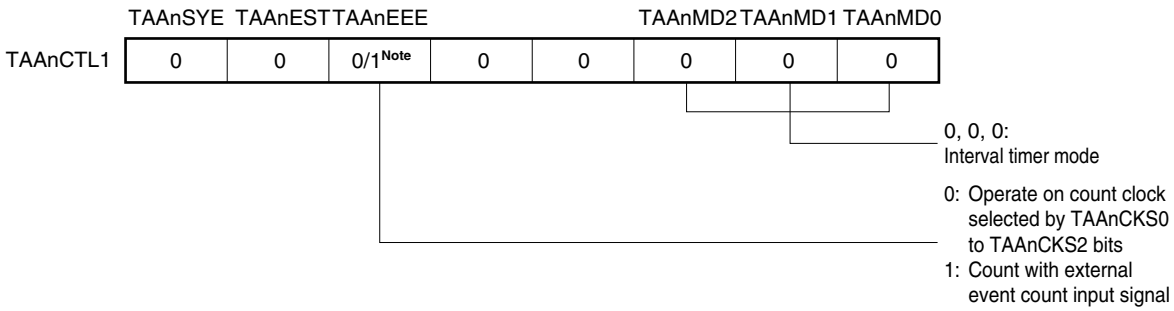
Remark n = 0 to 4

Figure 7-4. Register Setting for Interval Timer Mode Operation (1/2)

(a) TAAn control register 0 (TAAnCTL0)



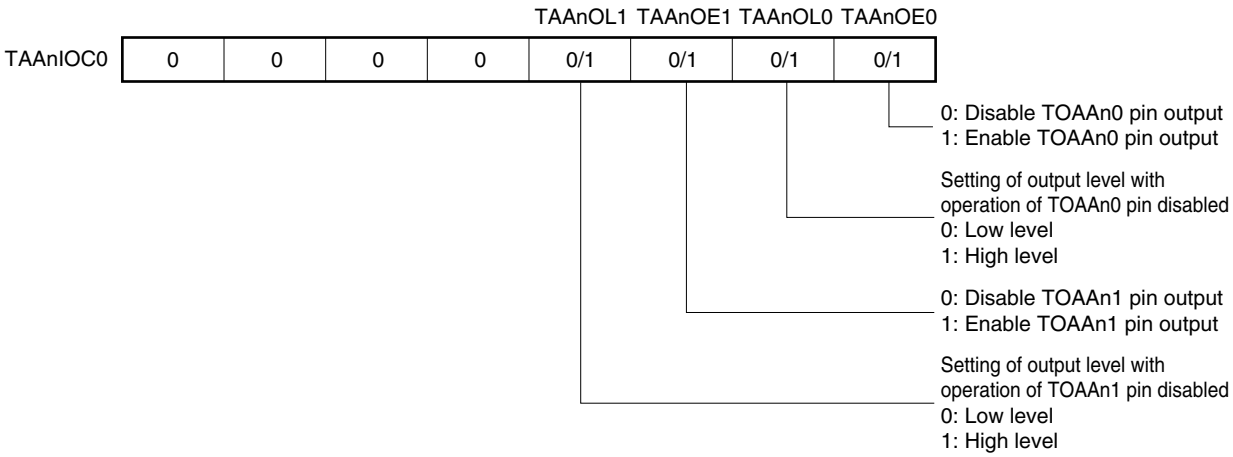
(b) TAAn control register 1 (TAAnCTL1)



Note This bit can be set to 1 only when the interrupt request signals (INTTAAnCC0 and INTTAAnCC1) are masked by the interrupt mask flags (TAAnCCMK0 and TAAnCCMK1) and timer output (TOAAn1) is performed at the same time. However, set the TAAnCCR0 and TAAnCCR1 registers to the same value (see **7.5.1 (2) (d) Operation of TAAnCCR1 register**).

Figure 7-4. Register Setting for Interval Timer Mode Operation (2/2)

(c) TAAn I/O control register 0 (TAAnIOC0)



(d) TAAn counter read buffer register (TAAnCNT)

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

(e) TAAn capture/compare register 0 (TAAnCCR0)

If the TAAnCCR0 register is set to D_0, the interval is as follows.

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

(f) TAAn capture/compare register 1 (TAAnCCR1)

Usually, the TAAnCCR1 register is not used in the interval timer mode. However, the set value of the TAAnCCR1 register is transferred to the CCR1 buffer register. A compare match interrupt request signal (INTTAAnCC1) is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

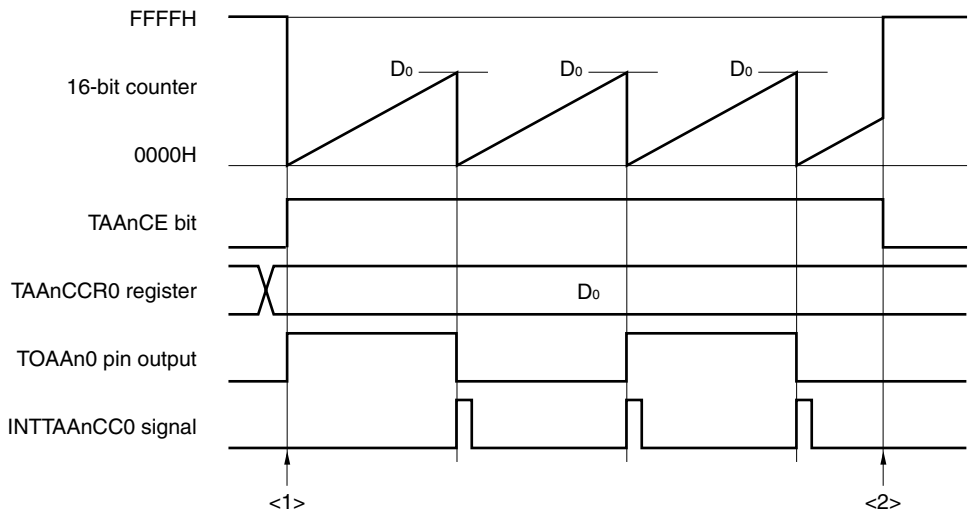
Therefore, mask the interrupt request by using the corresponding interrupt mask flag (TAAnCCMK1).

Remarks 1. TAAn I/O control register 1 (TAAnIOC1), TAAn I/O control register 2 (TAAnIOC2), and TAAn option register 0 (TAAnOPT0) are not used in the interval timer mode.

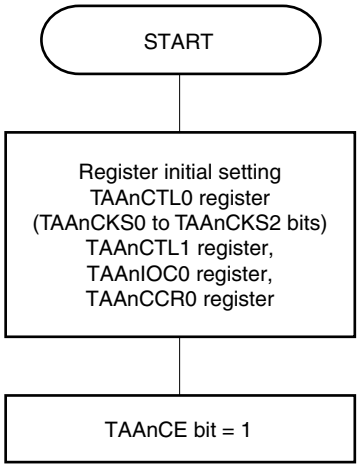
2. n = 0 to 4

(1) Interval timer mode operation flow

Figure 7-5. Software Processing Flow in Interval Timer Mode



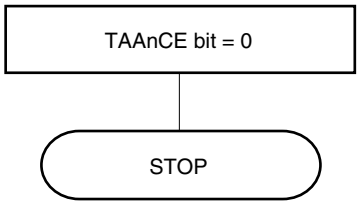
<1> Count operation start flow



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

The TAAnCKS0 to TAAnCKS2 bits can be set at the same time when counting has been started (TAAnCE bit = 1).

<2> Count operation stop flow



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

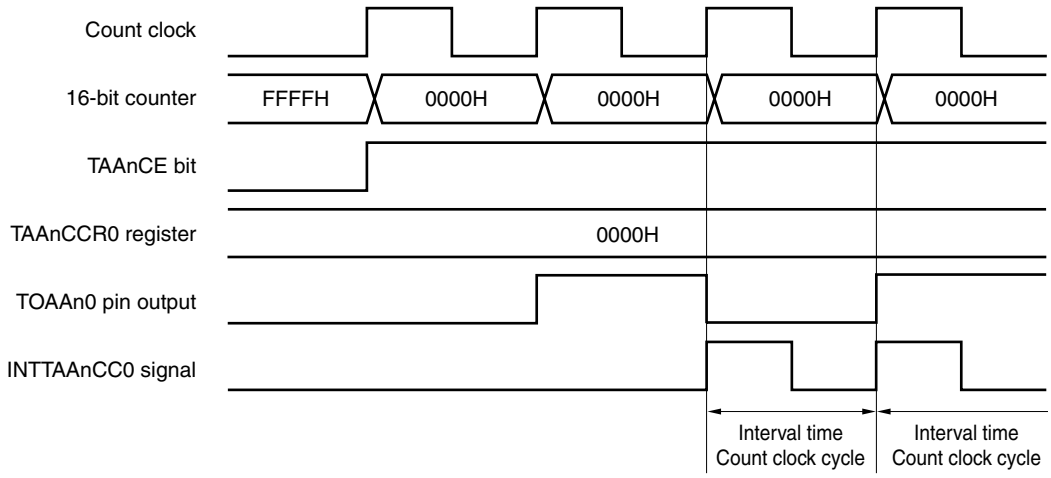
Remark n = 0 to 4

(2) Interval timer mode operation timing

(a) Operation if TAAnCCR0 register is set to 0000H

If the TAAnCCR0 register is set to 0000H, the INTTAAnCC0 signal is generated at each count clock subsequent to the first count clock, and the output of the TOAAn0 pin is inverted.

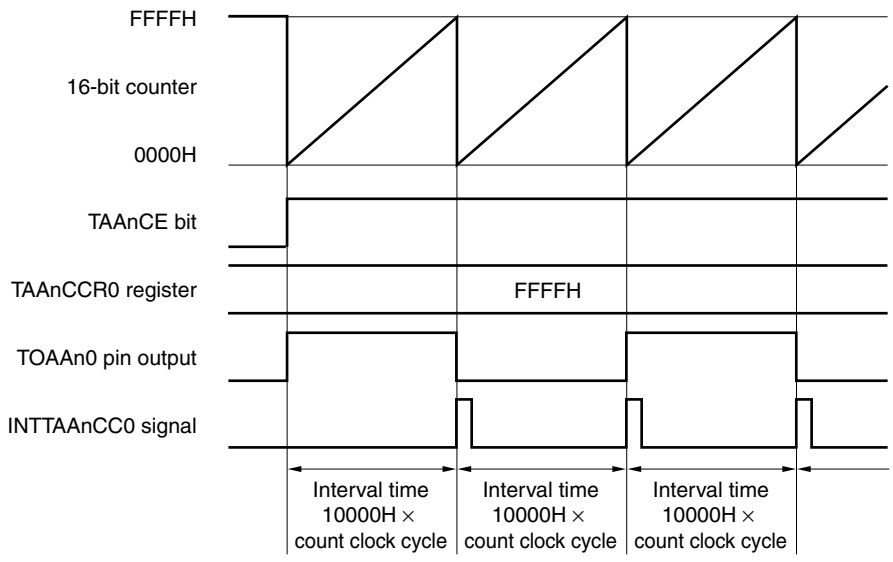
The value of the 16-bit counter is always 0000H.



Remark n = 0 to 4

(b) Operation if TAAnCCR0 register is set to FFFFH

If the TAAnCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTAAnCC0 signal is generated and the output of the TOAAn0 pin is inverted. At this time, an overflow interrupt request signal (INTTAAnOV) is not generated, nor is the overflow flag (TAAnOPT0.TAAnOVF bit) set to 1.

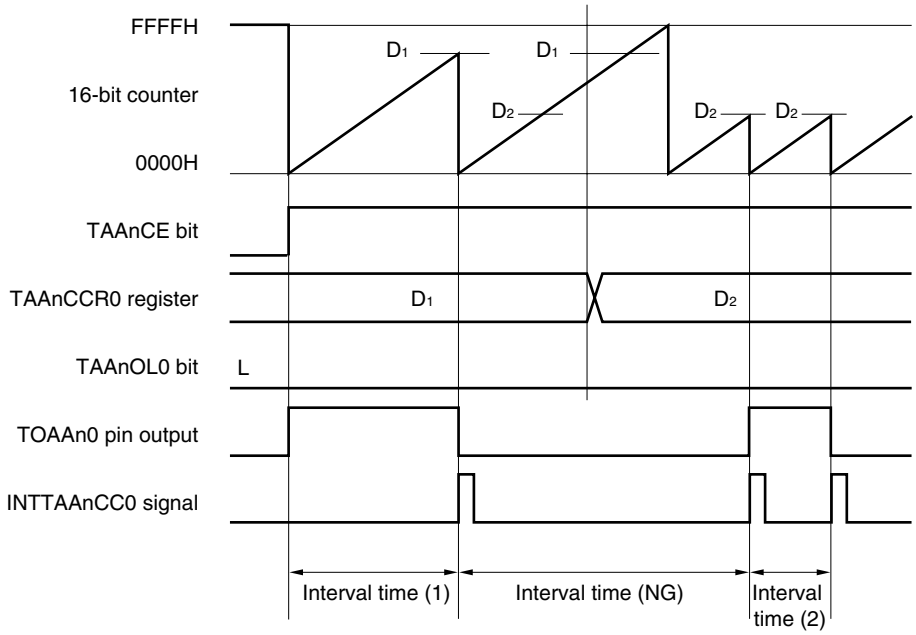


Remark n = 0 to 4

(c) Notes on rewriting TAAnCCR0 register

To change the value of the TAAnCCR0 register to a smaller value, stop counting once and then change the set value.

If the value of the TAAnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.



Remarks 1. Interval time (1): $(D_1 + 1) \times$ Count clock cycle
Interval time (NG): $(10000H + D_2 + 1) \times$ Count clock cycle
Interval time (2): $(D_2 + 1) \times$ Count clock cycle

2. n = 0 to 4

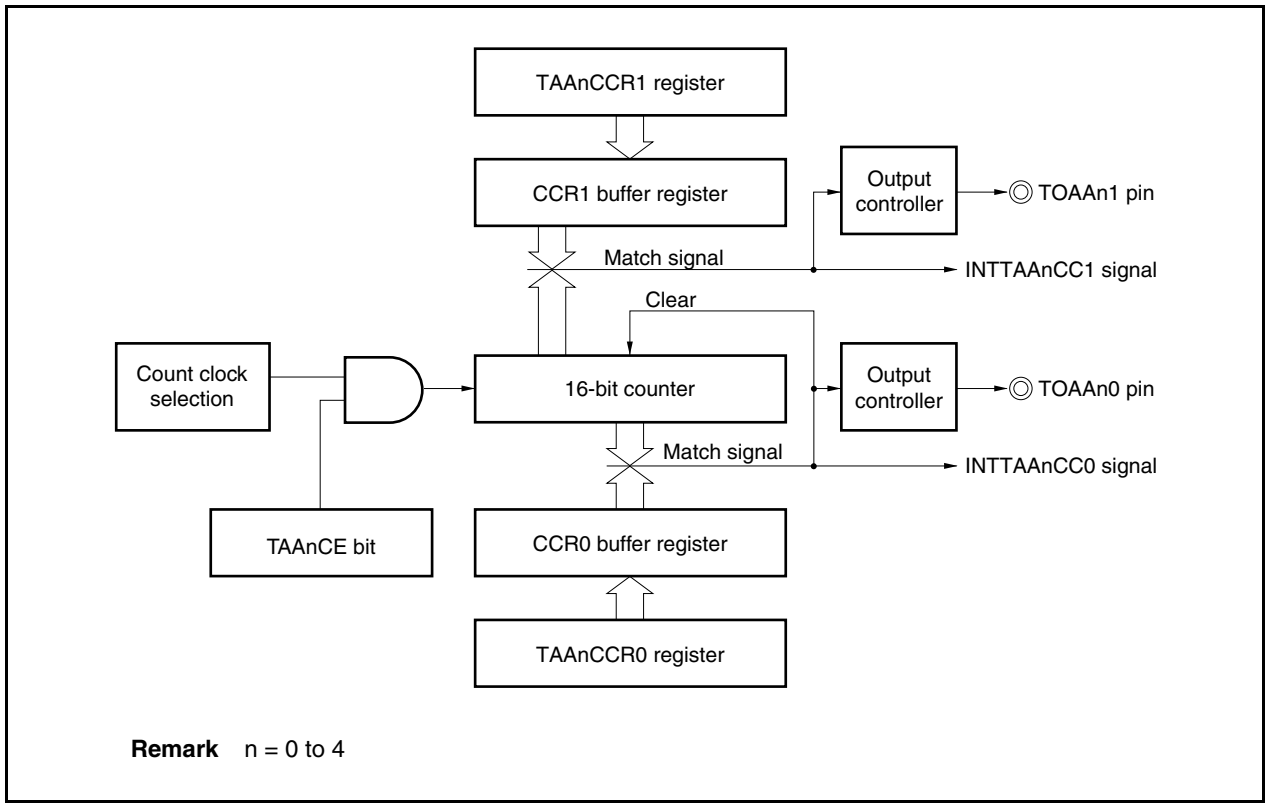
If the value of the TAAnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TAAnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTAAnCC0 signal is generated and the output of the TOAAn0 pin is inverted.

Therefore, the INTTAAnCC0 signal may not be generated at the interval time "$(D_1 + 1) \times$ Count clock cycle" or "$(D_2 + 1) \times$ Count clock cycle" originally expected, but may be generated at an interval of "$(10000H + D_2 + 1) \times$ Count clock period".

(d) Operation of TAAnCCR1 register

Figure 7-6. Configuration of TAAnCCR1 Register

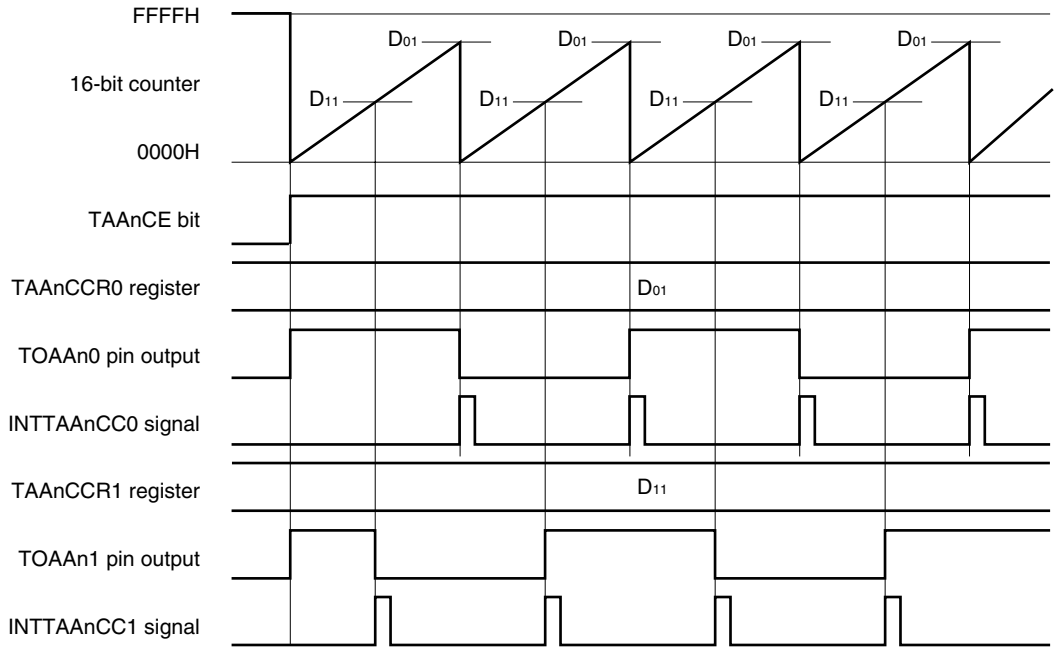


Remark n = 0 to 4

If the set value of the TAAnCCR1 register is less than the set value of the TAAnCCR0 register, the INTTAAnCC1 signal is generated once per cycle. At the same time, the output of the TOAAn1 pin is inverted.

The TOAAn1 pin outputs a square wave with the same cycle as that output by the TOAAn0 pin.

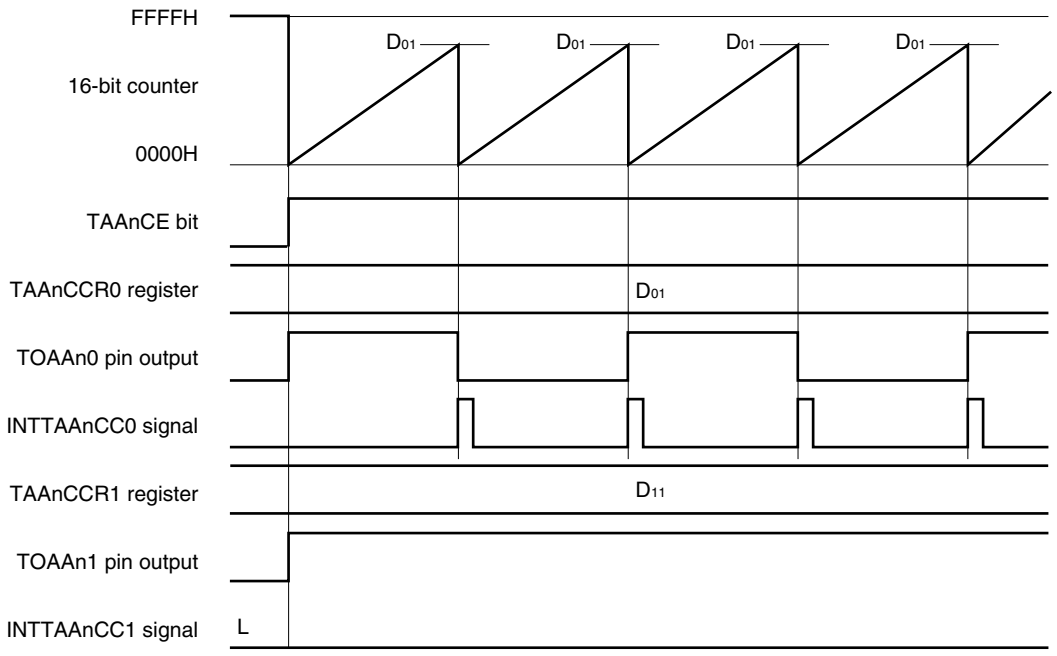
Figure 7-7. Timing Chart When $D_{01} \geq D_{11}$



Remark n = 0 to 4

If the set value of the TAAnCCR1 register is greater than the set value of the TAAnCCR0 register, the count value of the 16-bit counter does not match the value of the TAAnCCR1 register. Consequently, the INTTAAnCC1 signal is not generated, nor is the output of the TOAAn1 pin changed.

Figure 7-8. Timing Chart When $D_{01} < D_{11}$



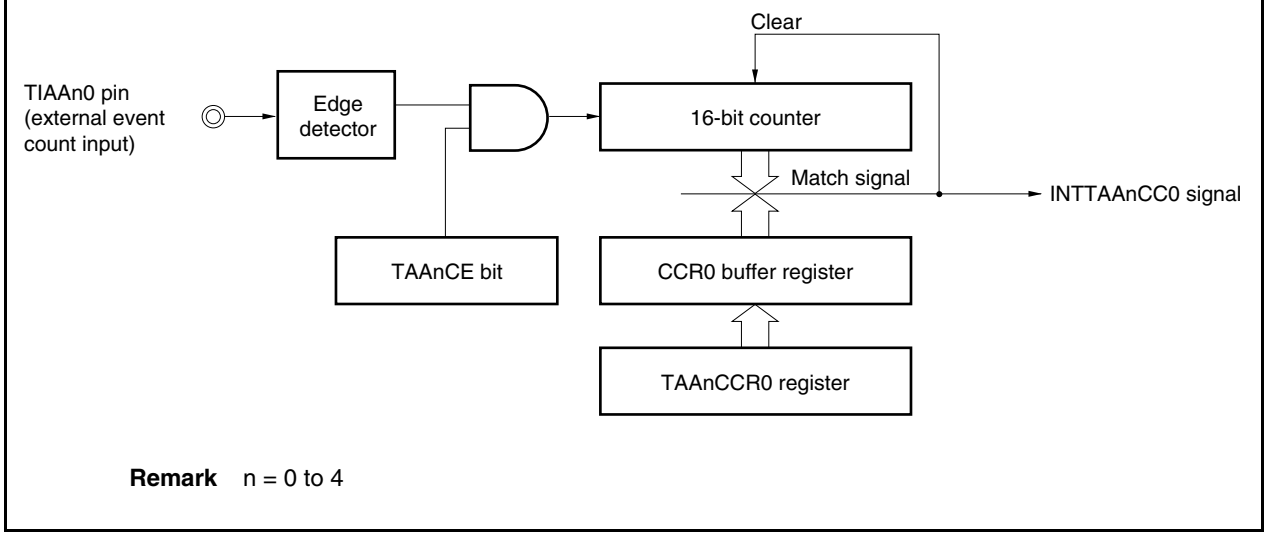
Remark n = 0 to 4

7.5.2 External event count mode (TAAnMD2 to TAAnMD0 bits = 001)

In the external event count mode, the valid edge of the external event count input is counted when the TAAnCTL0.TAAnCE bit is set to 1, and an interrupt request signal (INTTAAnCC0) is generated each time the specified number of edges have been counted. The TOAAn0 pin cannot be used.

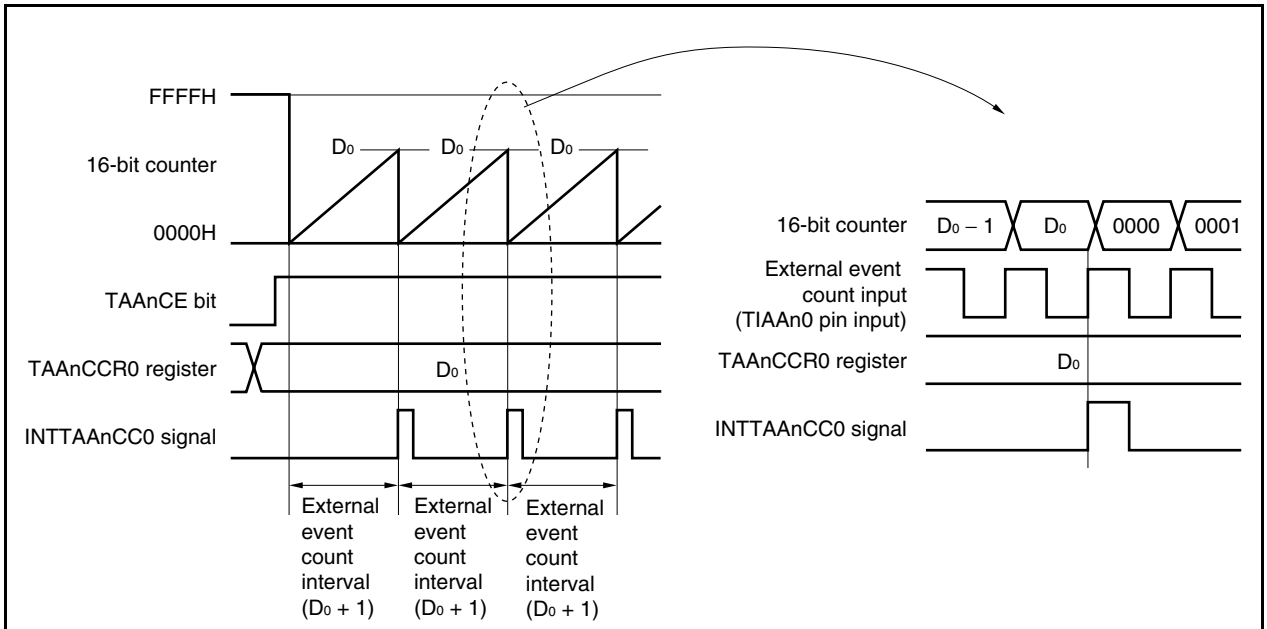
Usually, the TAAnCCR1 register is not used in the external event count mode.

Figure 7-9. Configuration in External Event Count Mode



Remark n = 0 to 4

Figure 7-10. Basic Timing in External Event Count Mode



Remarks 1. This figure shows the basic timing when the rising edge is specified as the valid edge of the external event count input.

2. n = 0 to 4

When the TAAnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H. The counter counts each time the valid edge of external event count input is detected. Additionally, the set value of the TAAnCCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTAAnCC0) is generated.

The INTTAAnCC0 signal is generated each time the valid edge of the external event count input has been detected (set value of TAAnCCR0 register + 1) times.

Figure 7-11. Register Setting for Operation in External Event Count Mode (1/2)

(a) TAAn control register 0 (TAAnCTL0)

	TAAnCE					TAAnCKS2	TAAnCKS1	TAAnCKS0
TAAnCTL0	0/1	0	0	0	0	0	0	0

0: Stop counting
1: Enable counting

(b) TAAn control register 1 (TAAnCTL1)

	TAAnSYE	TAAnEST	TAAnEEE			TAAnMD2	TAAnMD1	TAAnMD0
TAAnCTL1	0	0	0	0	0	0	0	1

0, 0, 1:
External event count mode

(c) TAAn I/O control register 0 (TAAnIOC0)

					TAAnOL1	TAAnOE1	TAAnOL0	TAAnOE0
TAAnIOC0	0	0	0	0	0	0	0	0

0: Disable TOAAn0 pin output
0: Disable TOAAn1 pin output

(d) TAAn I/O control register 2 (TAAnIOC2)

					TAAnEES1	TAAnEES0	TAAnETS1	TAAnETS0
TAAnIOC2	0	0	0	0	0/1	0/1	0	0

Select valid edge of
external event count input

(e) TAAn counter read buffer register (TAAnCNT)

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

Figure 7-11. Register Setting for Operation in External Event Count Mode (2/2)

(f) TAAn capture/compare register 0 (TAAnCCR0)

If D_0 is set to the TAAnCCR0 register, the counter is cleared and a compare match interrupt request signal (INTTAAnCC0) is generated when the number of external event counts reaches (D_0 + 1).

(g) TAAn capture/compare register 1 (TAAnCCR1)

Usually, the TAAnCCR1 register is not used in the external event count mode. However, the set value of the TAAnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTAAnCC1) is generated.

Therefore, mask the interrupt signal by using the interrupt mask flag (TAAnCCMK1).

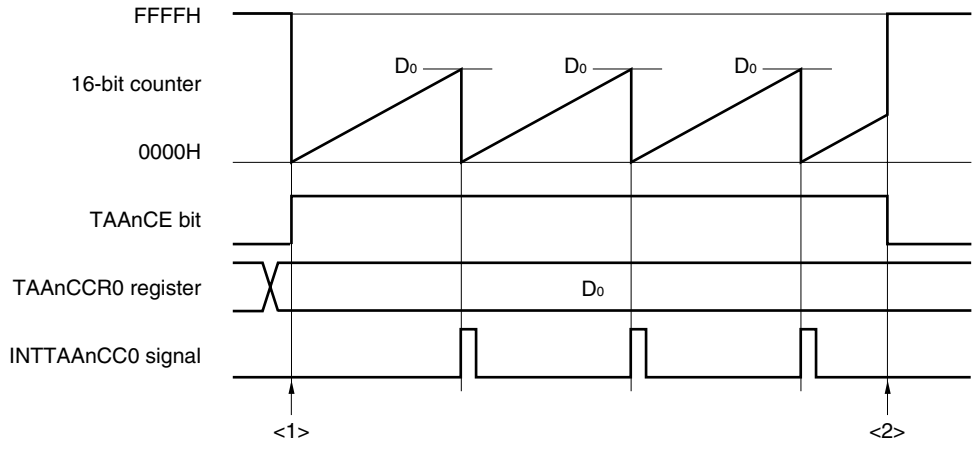
Caution When an external clock is used as the count clock, the external clock can be input only from the TIAAn0 pin. At this time, set the TAAnIOC1.TAAnIS1 and TAAnIOC1.TAAnIS0 bits to 00 (capture trigger input (TIAAn0 pin): no edge detection).

Remarks 1. TAAn I/O control register 1 (TAAnIOC1) and TAAn option register 0 (TAAnOPT0) are not used in the external event count mode.

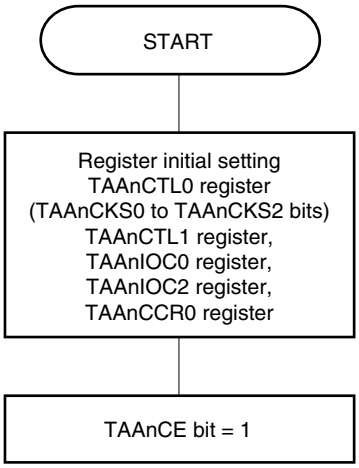
2. n = 0 to 4

(1) External event count mode operation flow

Figure 7-12. Flow of Software Processing in External Event Count Mode



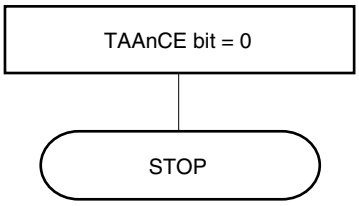
<1> Count operation start flow



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

The TAAnCKS0 to TAAnCKS2 bits can be set at the same time when counting has been started (TAAnCE bit = 1).

<2> Count operation stop flow



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

Remark n = 0 to 4

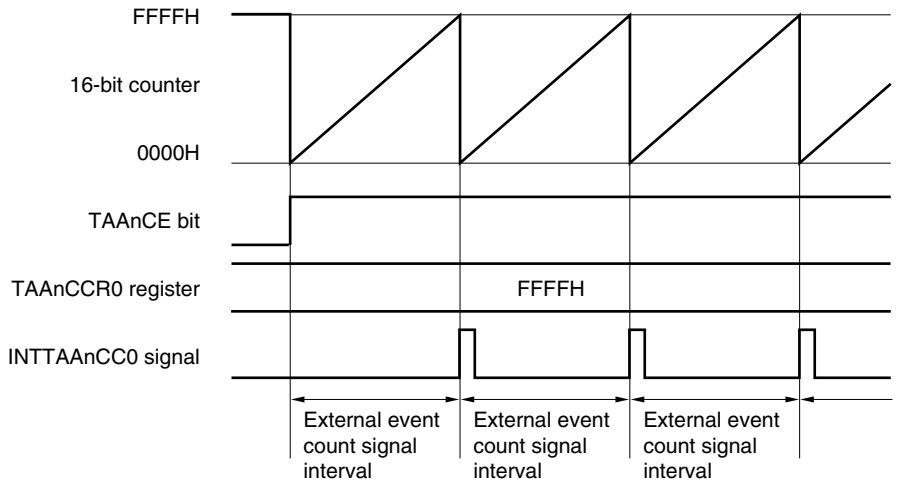
(2) Operation timing in external event count mode

Cautions 1. In the external event count mode, do not set the TAAnCCR0 register to 0000H.

2. In the external event count mode, use of the timer output is disabled. If performing timer output using external event count input, set the interval timer mode, and select the operation enabled by the external event count input for the count clock (TAAnCTL1.TAAnMD2 to TAAnCTL1.TAAnMD0 bits = 000, TAAnCTL1.TAAnEEE bit = 1).

(a) Operation if TAAnCCR0 register is set to FFFFH

If the TAAnCCR0 register is set to FFFFH, the 16-bit counter counts to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTTAAnCC0 signal is generated. At this time, the TAAnOPT0.TAAnOVF bit is not set.

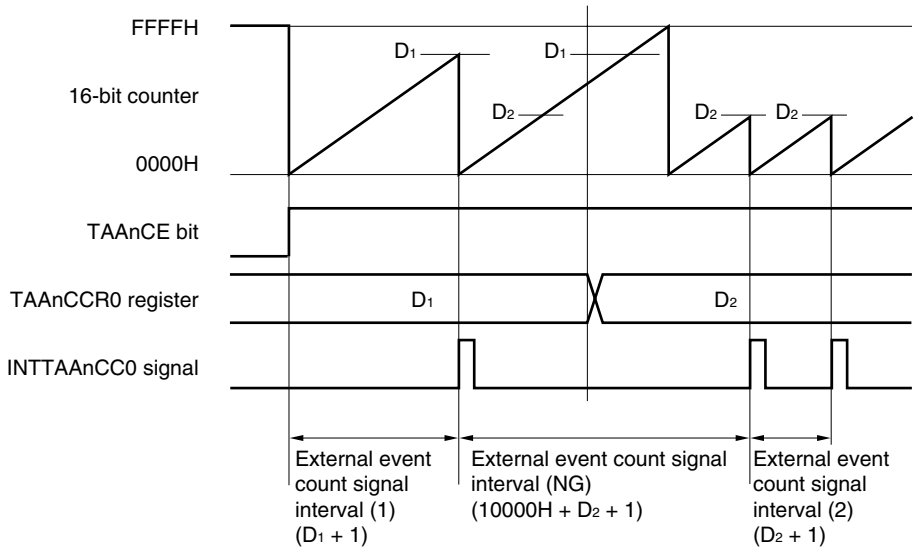


Remark n = 0 to 4

(b) Notes on rewriting the TAAnCCR0 register

To change the value of the TAAnCCR0 register to a smaller value, stop counting once and then change the set value.

If the value of the TAAnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.



Remark n = 0 to 4

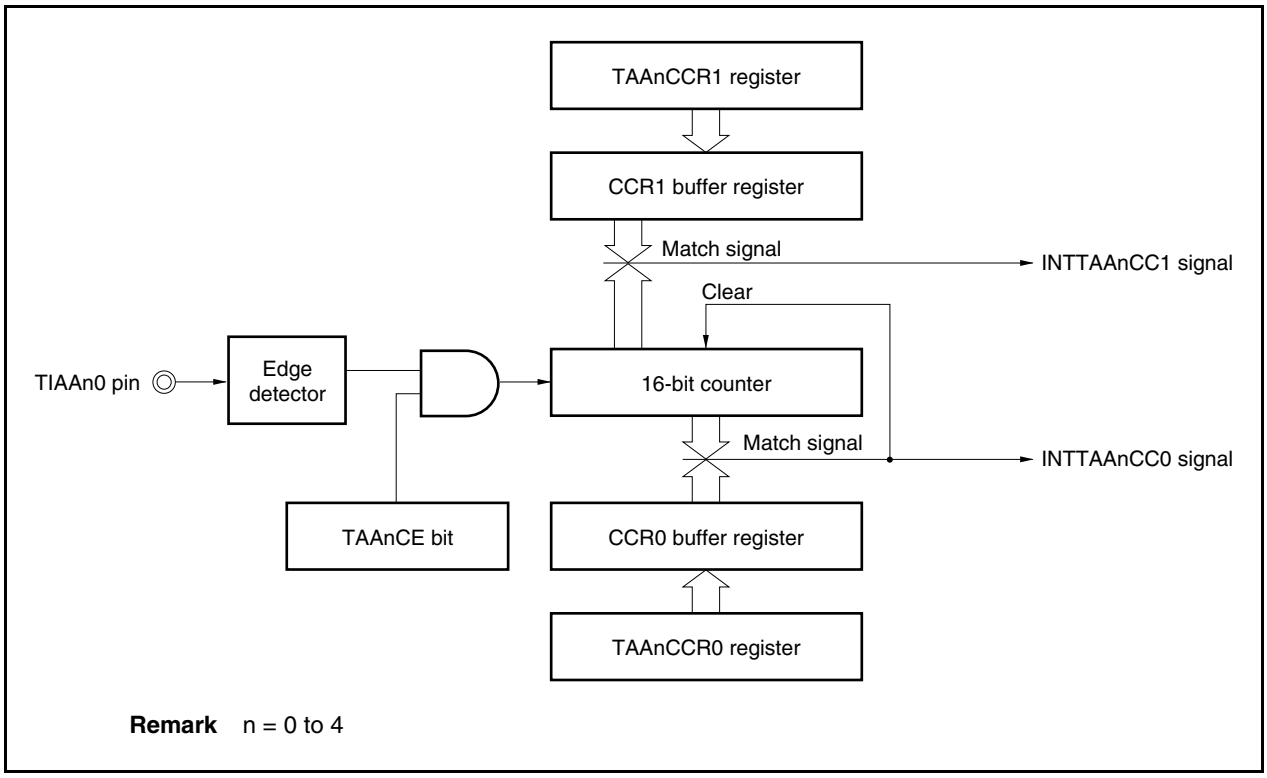
If the value of the TAAnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TAAnCCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTAAnCC0 signal is generated.

Therefore, the INTTAAnCC0 signal may not be generated at the valid edge count of "(D_1 + 1) times" or "(D_2 + 1) times" originally expected, but may be generated at the valid edge count of "(10000H + D_2 + 1) times".

(c) Operation of TAAnCCR1 register

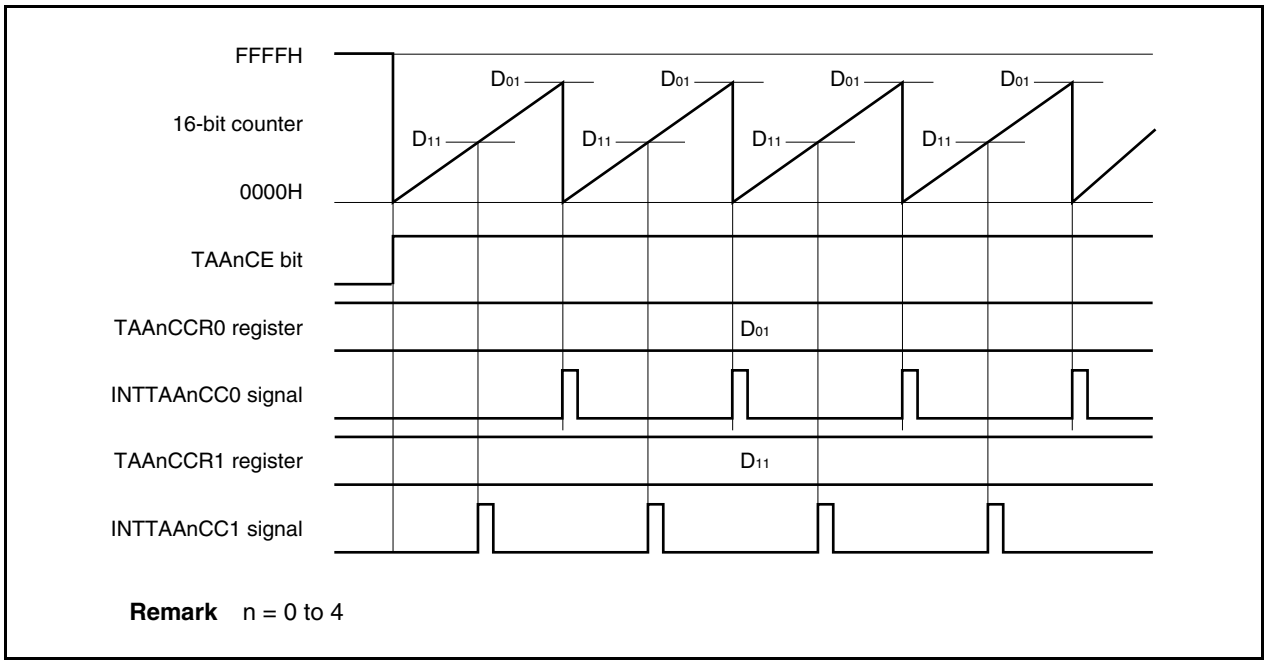
Figure 7-13. Configuration of TAAnCCR1 Register



Remark n = 0 to 4

If the set value of the TAAnCCR1 register is smaller than the set value of the TAAnCCR0 register, the INTTAAnCC1 signal is generated once per cycle.

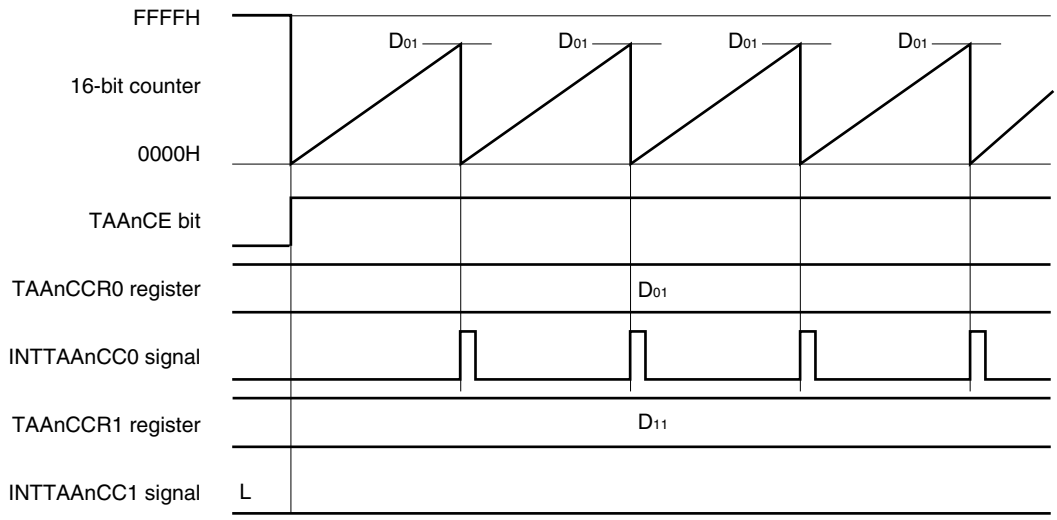
Figure 7-14. Timing Chart When $D_{01} \geq D_{11}$



Remark n = 0 to 4

If the set value of the TAAnCCR1 register is greater than the set value of the TAAnCCR0 register, the INTTAAnCC1 signal is not generated because the count value of the 16-bit counter and the value of the TAAnCCR1 register do not match.

Figure 7-15. Timing Chart When $D_{01} < D_{11}$



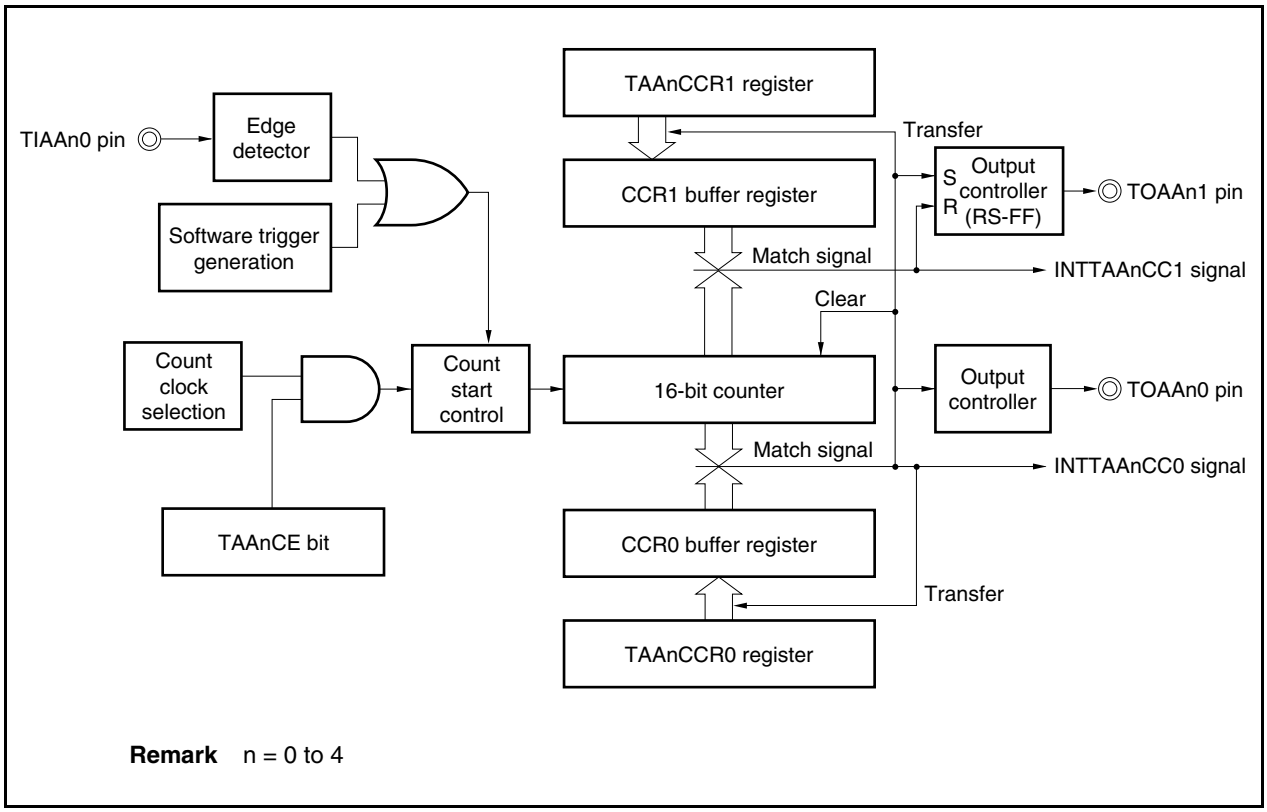
Remark n = 0 to 4

7.5.3 External trigger pulse output mode (TAAnMD2 to TAAnMD0 bits = 010)

In the external trigger pulse output mode, 16-bit timer/event counter AA waits for a trigger when the TAAnCTL0.TAAnCE bit is set to 1. When the valid edge of an external trigger input signal is detected, 16-bit timer/event counter AA starts counting, and outputs a PWM waveform from the TOAAn1 pin.

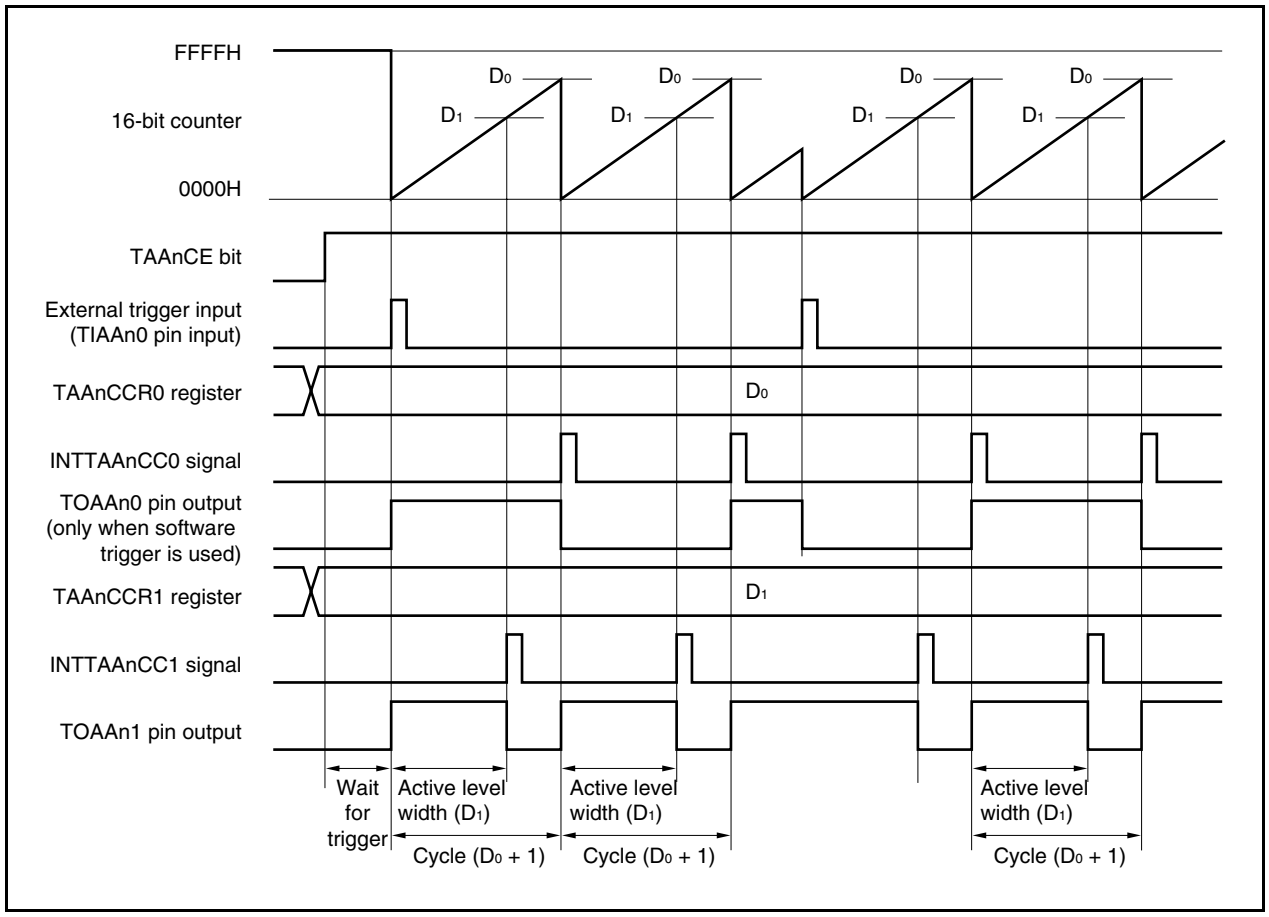
Pulses can also be output by generating a software trigger instead of using the external trigger. When using a software trigger, a square wave that has one cycle of the PWM waveform as half its cycle can also be output from the TOAAn0 pin.

Figure 7-16. Configuration in External Trigger Pulse Output Mode



Remark n = 0 to 4

Figure 7-17. Basic Timing in External Trigger Pulse Output Mode



16-bit timer/event counter AA waits for a trigger when the TAAnCE bit is set to 1. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting at the same time, and outputs a PWM waveform from the TOAAn1 pin. If the trigger is generated again while the counter is operating, the counter is cleared to 0000H and restarted. (The output of the TOAAn0 pin is inverted. The TOAAn1 pin outputs a high level regardless of the status (high/low) when a trigger occurs.)

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

Active level width = (Set value of TAAnCCR1 register) × Count clock cycle
Cycle = (Set value of TAAnCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TAAnCCR1 register)/(Set value of TAAnCCR0 register + 1)

The compare match request signal INTTAAnCC0 is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal INTTAAnCC1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TAAnCCRm register is transferred to the CCRm buffer register when the count value of the 16-bit counter matches the value of the CCRm buffer register and the 16-bit counter is cleared to 0000H.

The valid edge of an external trigger input signal, or setting the software trigger (TAAnCTL1.TAAnEST bit) to 1 is used as the trigger.

Remark n = 0 to 4, m = 0, 1

Figure 7-18. Setting of Registers in External Trigger Pulse Output Mode (1/2)

(a) TAAn control register 0 (TAAnCTL0)

	TAAnCE					TAAnCKS2	TAAnCKS1	TAAnCKS0
TAAnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock

0: Stop counting
1: Enable counting

(b) TAAn control register 1 (TAAnCTL1)

	TAAnSYE	TAAnEST	TAAnEEE			TAAnMD2	TAAnMD1	TAAnMD0
TAAnCTL1	0	0/1	0	0	0	0	1	0

0, 1, 0:
External trigger pulse output mode

Generate software trigger when 1 is written

(c) TAAn I/O control register 0 (TAAnIOC0)

					TAAnOL1	TAAnOE1	TAAnOL0	TAAnOE0
TAAnIOC0	0	0	0	0	0/1	0/1	0/1[Note]	0/1[Note]

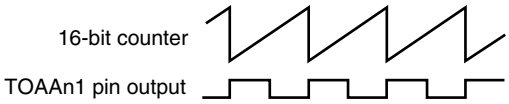
0: Disable TOAAn0 pin output
1: Enable TOAAn0 pin output

Settings of output level while operation of TOAAn0 pin is disabled
0: Low level
1: High level

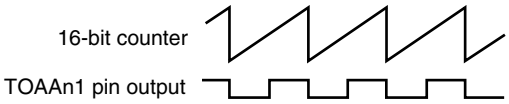
0: Disable TOAAn1 pin output
1: Enable TOAAn1 pin output

Specifies active level of TOAAn1 pin output
0: Active-high
1: Active-low

- When TAAnOL1 bit = 0



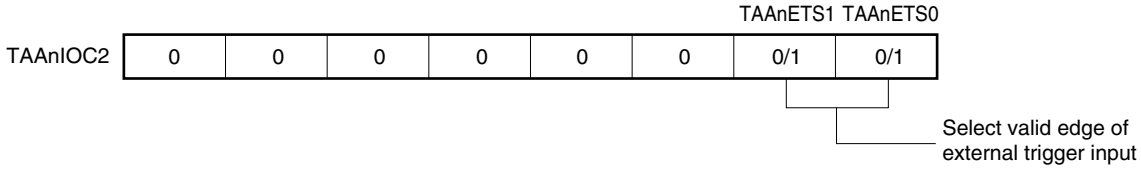
- When TAAnOL1 bit = 1



Note Clear this bit to 0 when the TOAAn0 pin is not used in the external trigger pulse output mode.

Figure 7-18. Setting of Registers in External Trigger Pulse Output Mode (2/2)

(d) TAAn I/O control register 2 (TAAnIOC2)



(e) TAAn counter read buffer register (TAAnCNT)

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

(f) TAAn capture/compare registers 0 and 1 (TAAnCCR0 and TAAnCCR1)

If D_0 is set to the TAAnCCR0 register and D_1 to the TAAnCCR1 register, the cycle and active level of the PWM waveform are as follows.

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

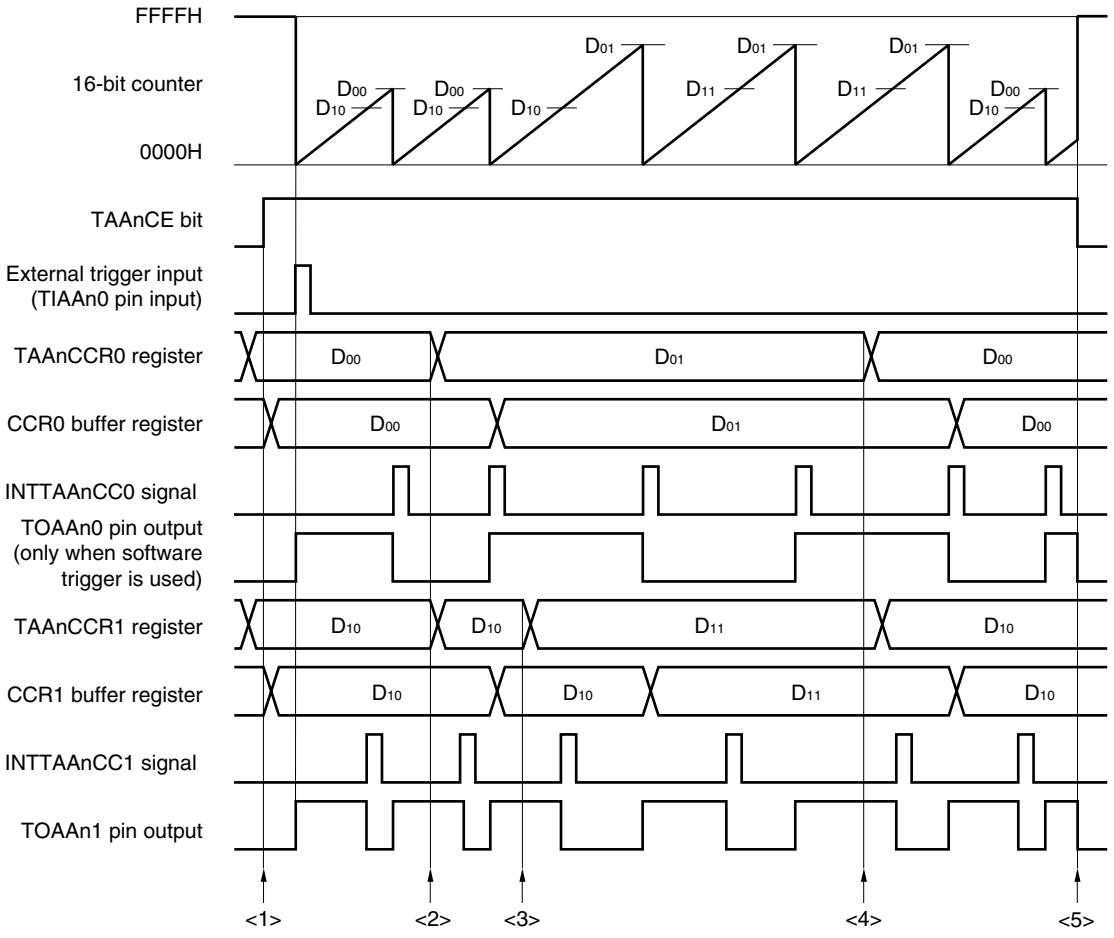
Active level width = $D_1 \times$ Count clock cycle

Remarks 1. TAAn I/O control register 1 (TAAnIOC1) and TAAn option register 0 (TAAnOPT0) are not used in the external trigger pulse output mode.

2. n = 0 to 4

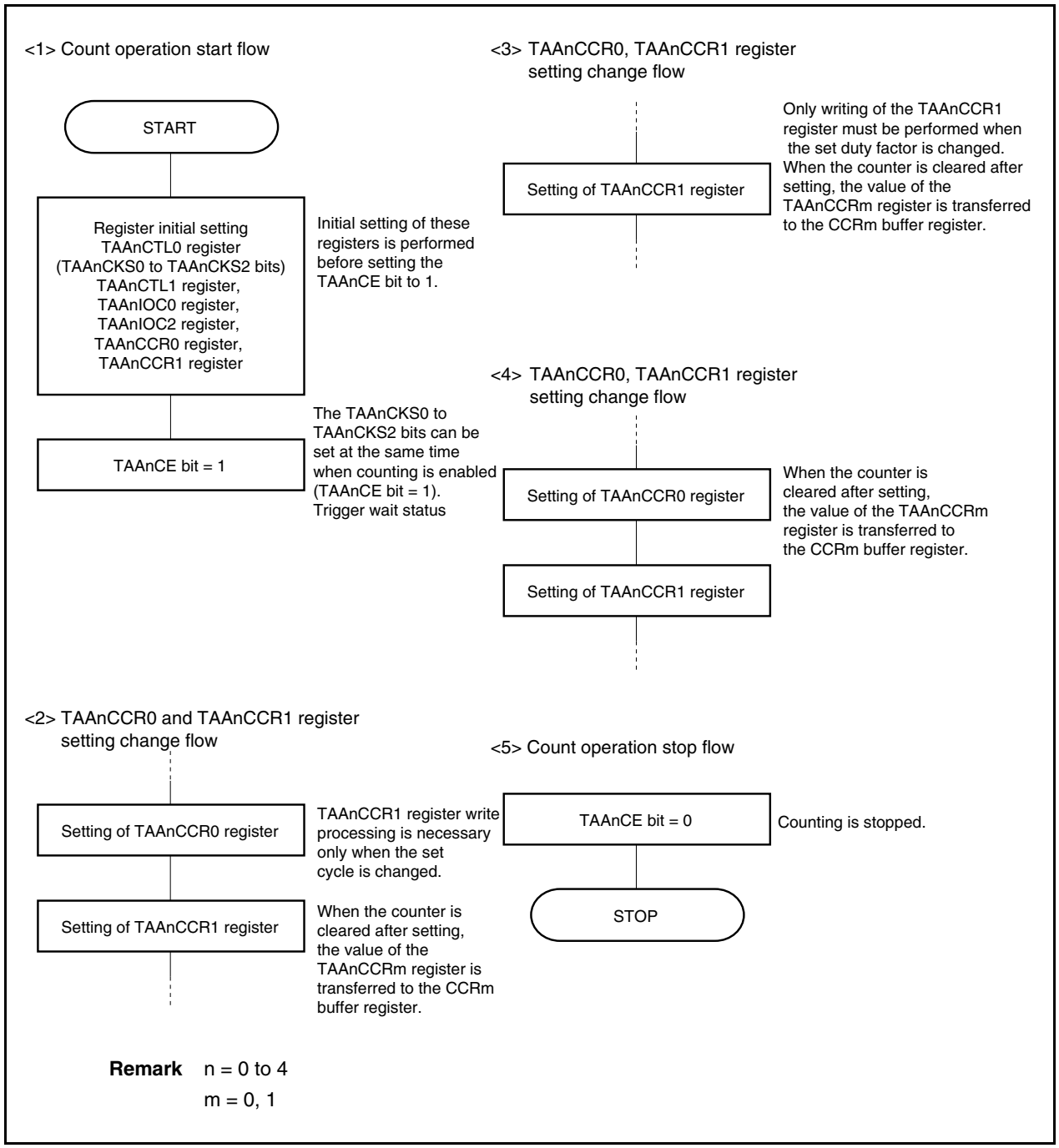
(1) Operation flow in external trigger pulse output mode

Figure 7-19. Software Processing Flow in External Trigger Pulse Output Mode (1/2)



Remark n = 0 to 4

Figure 7-19. Software Processing Flow in External Trigger Pulse Output Mode (2/2)



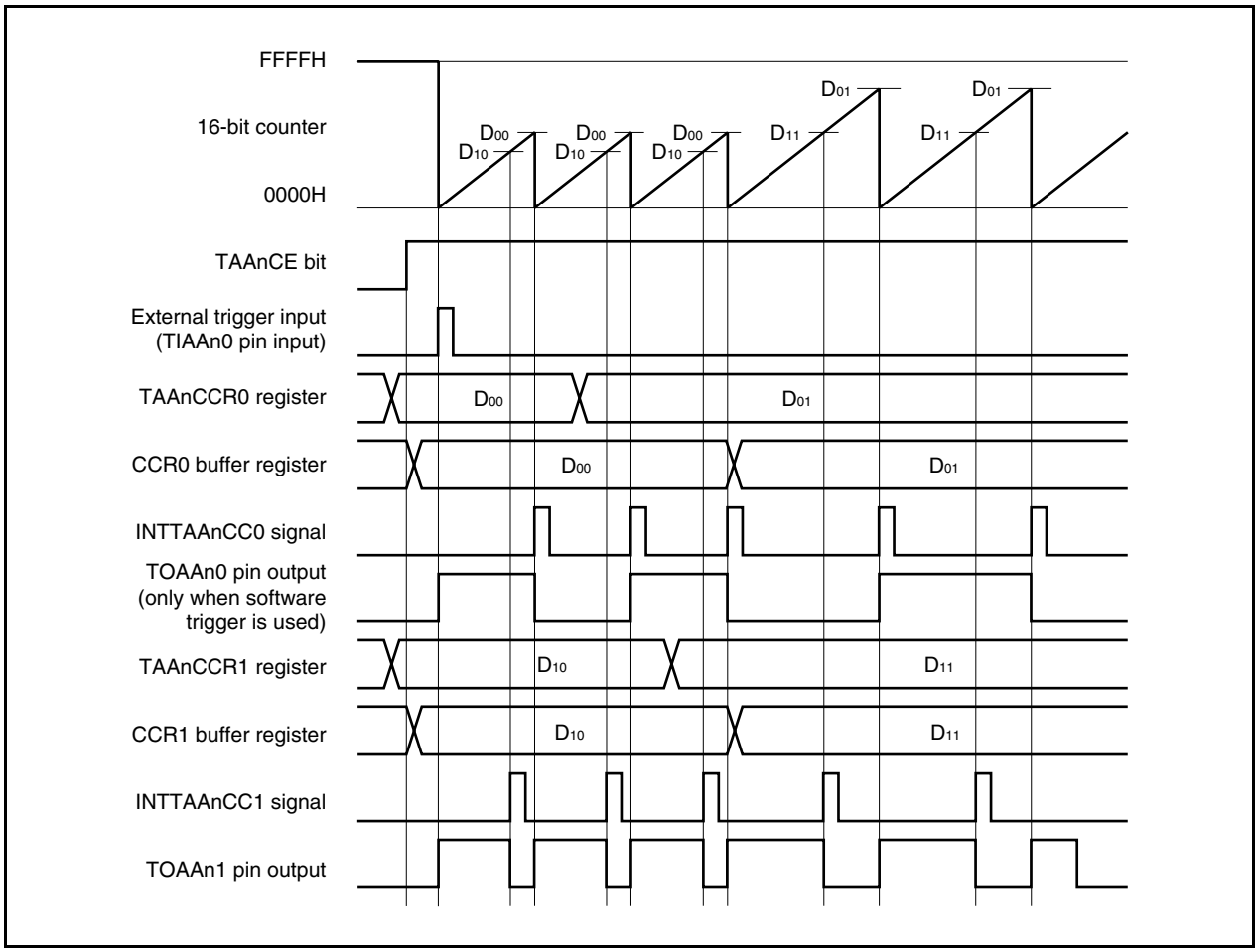
Remark n = 0 to 4
m = 0, 1

(2) External trigger pulse output mode operation timing

(a) Note on changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TAAnCCR1 register last.

Rewrite the TAAnCCRm register after writing the TAAnCCR1 register after the INTTAAnCC0 signal is detected.



In order to transfer data from the TAAnCCRm register to the CCRm buffer register, the TAAnCCR1 register must be written.

To change both the cycle and active level width of the PWM waveform at this time, first set the cycle to the TAAnCCR0 register and then set the active level width to the TAAnCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TAAnCCR0 register, and then write the same value to the TAAnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TAAnCCR1 register has to be set.

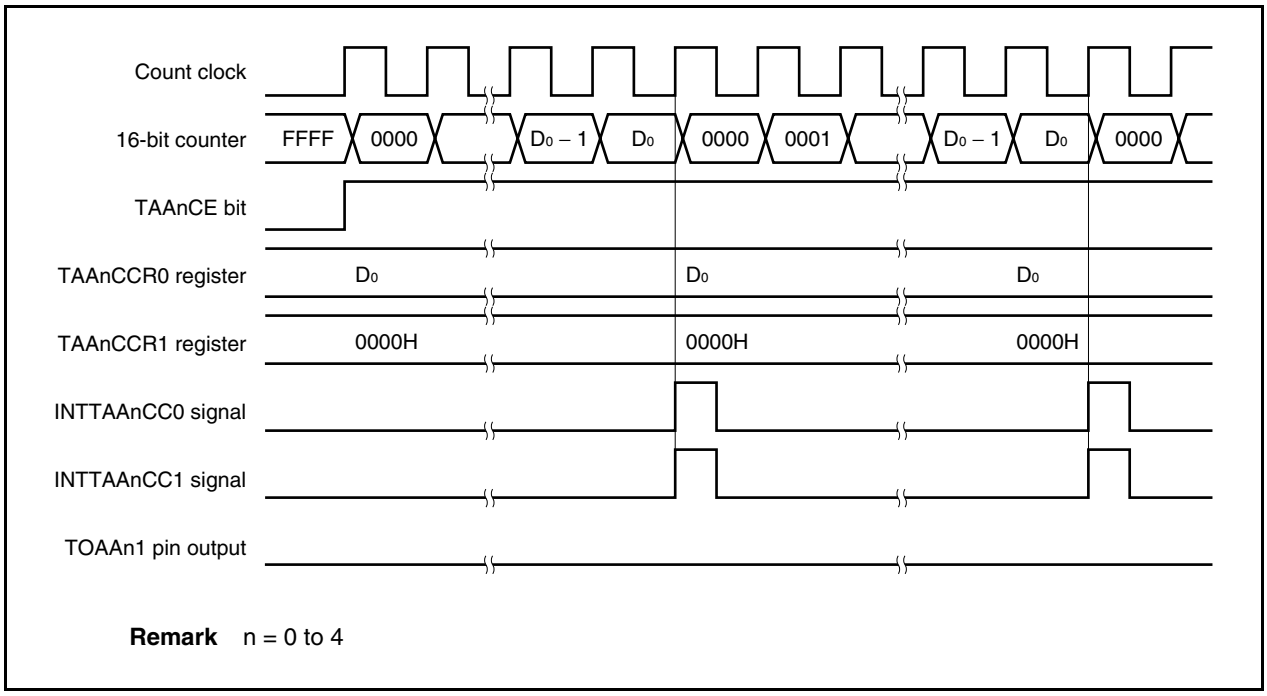
After data is written to the TAAnCCR1 register, the value written to the TAAnCCRm register is transferred to the CCRm buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TAAnCCR0 or TAAnCCR1 register again after writing the TAAnCCR1 register once, do so after the INTTAAnCC0 signal is generated. Otherwise, the value of the CCRm buffer register may become undefined because the timing of transferring data from the TAAnCCRm register to the CCRm buffer register conflicts with writing the TAAnCCRm register.

Remark n = 0 to 4
m = 0, 1

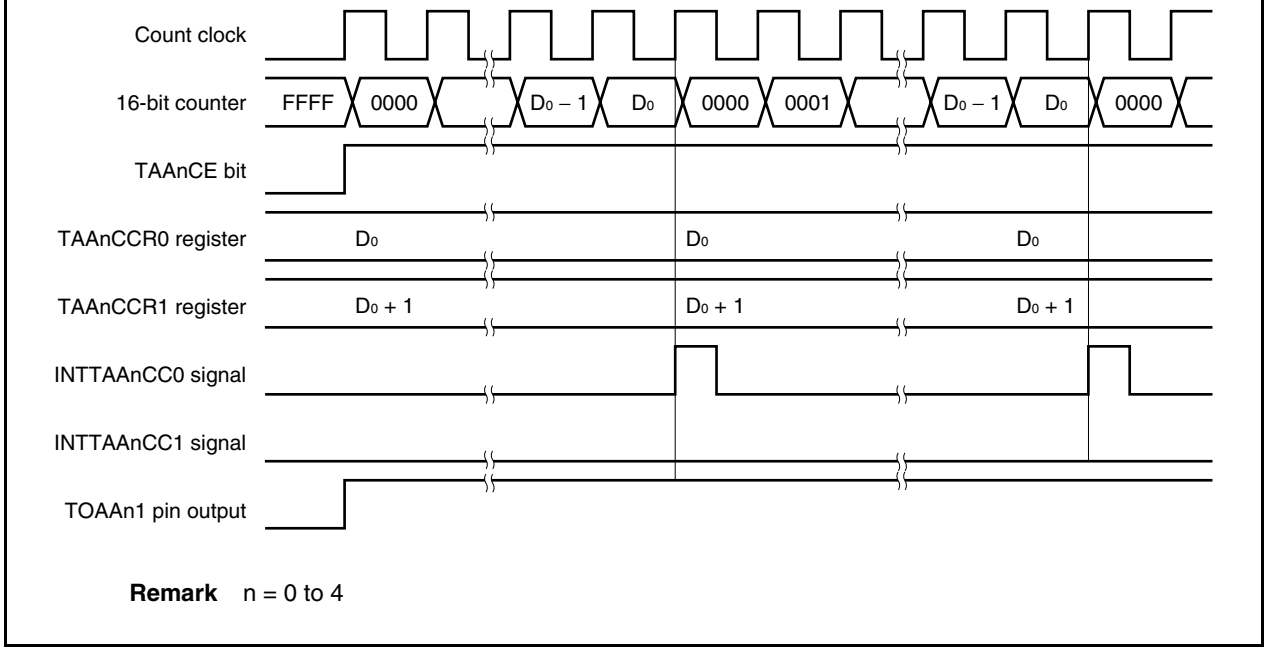
(b) 0%/100% output of PWM waveform

To output a 0% waveform, set the TAAnCCR1 register to 0000H. If the set value of the TAAnCCR0 register is FFFFH, the INTTAAnCC1 signal is generated periodically.



Remark n = 0 to 4

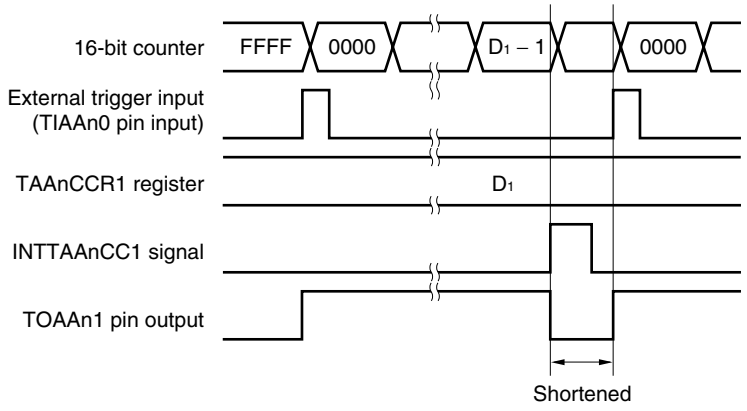
To output a 100% waveform, set a value of (set value of TAAnCCR0 register + 1) to the TAAnCCR1 register. If the set value of the TAAnCCR0 register is FFFFH, 100% output cannot be produced.



Remark n = 0 to 4

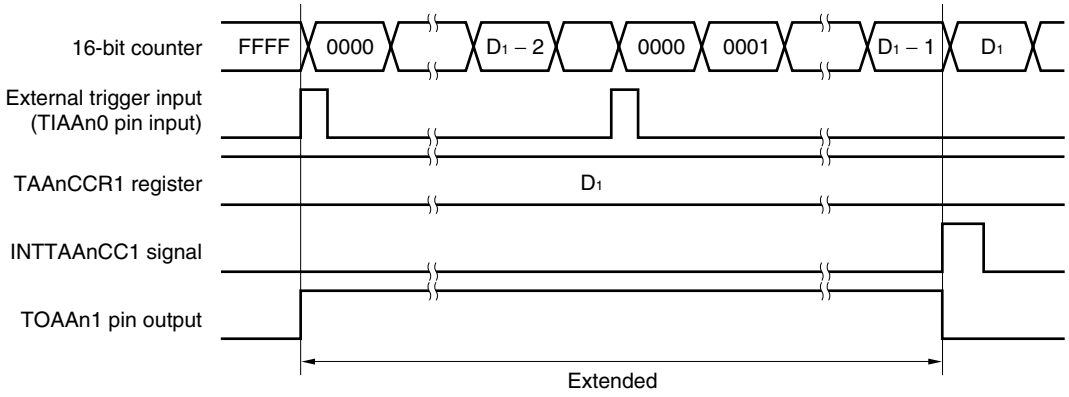
(c) Conflict between trigger detection and match with TAAnCCR1 register

If the trigger is detected immediately after the INTTAAnCC1 signal is generated, the 16-bit counter is immediately cleared to 0000H, the output signal of the TOAAn1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark n = 0 to 4

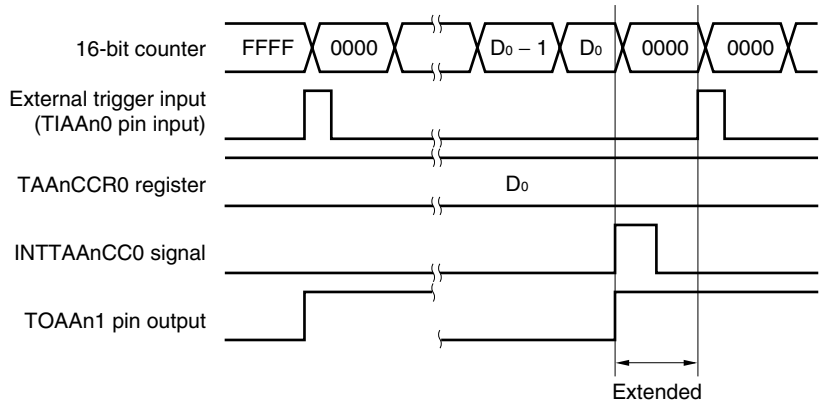
If the trigger is detected immediately before the INTTAAnCC1 signal is generated, the INTTAAnCC1 signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOAAn1 pin remains active. Consequently, the active period of the PWM waveform is extended.



Remark n = 0 to 4

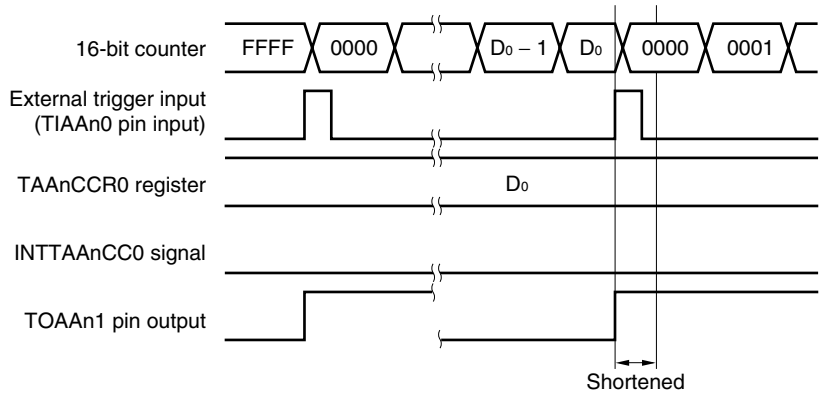
(d) Conflict between trigger detection and match with TAAnCCR0 register

If the trigger is detected immediately after the INTTAAnCC0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOAAn1 pin is extended by time from generation of the INTTAAnCC0 signal to trigger detection.



Remark n = 0 to 4

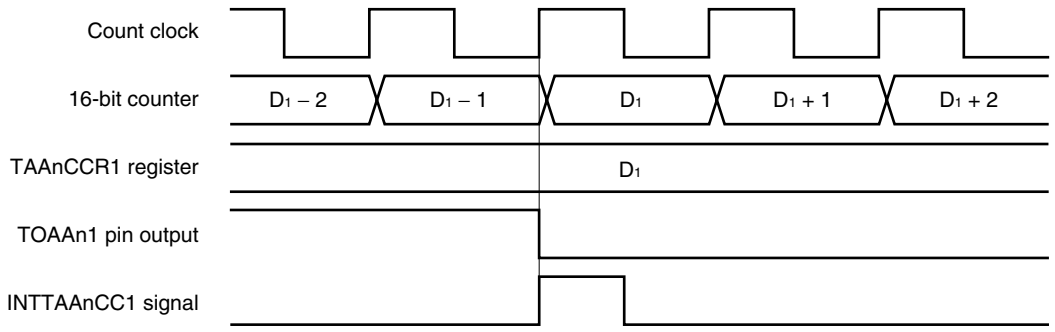
If the trigger is detected immediately before the INTTAAnCC0 signal is generated, the INTTAAnCC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOAAn1 pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark n = 0 to 4

(e) Generation timing of compare match interrupt request signal (INTTAAnCC1)

The timing of generation of the INTTAAnCC1 signal in the external trigger pulse output mode differs from the timing of other INTTAAnCC1 signals; the INTTAAnCC1 signal is generated when the count value of the 16-bit counter matches the value of the TAAnCCR1 register.



Remark n = 0 to 4

Usually, the INTTAAnCC1 signal is generated in synchronization with the next count up, after the count value of the 16-bit counter matches the value of the TAAnCCR1 register.

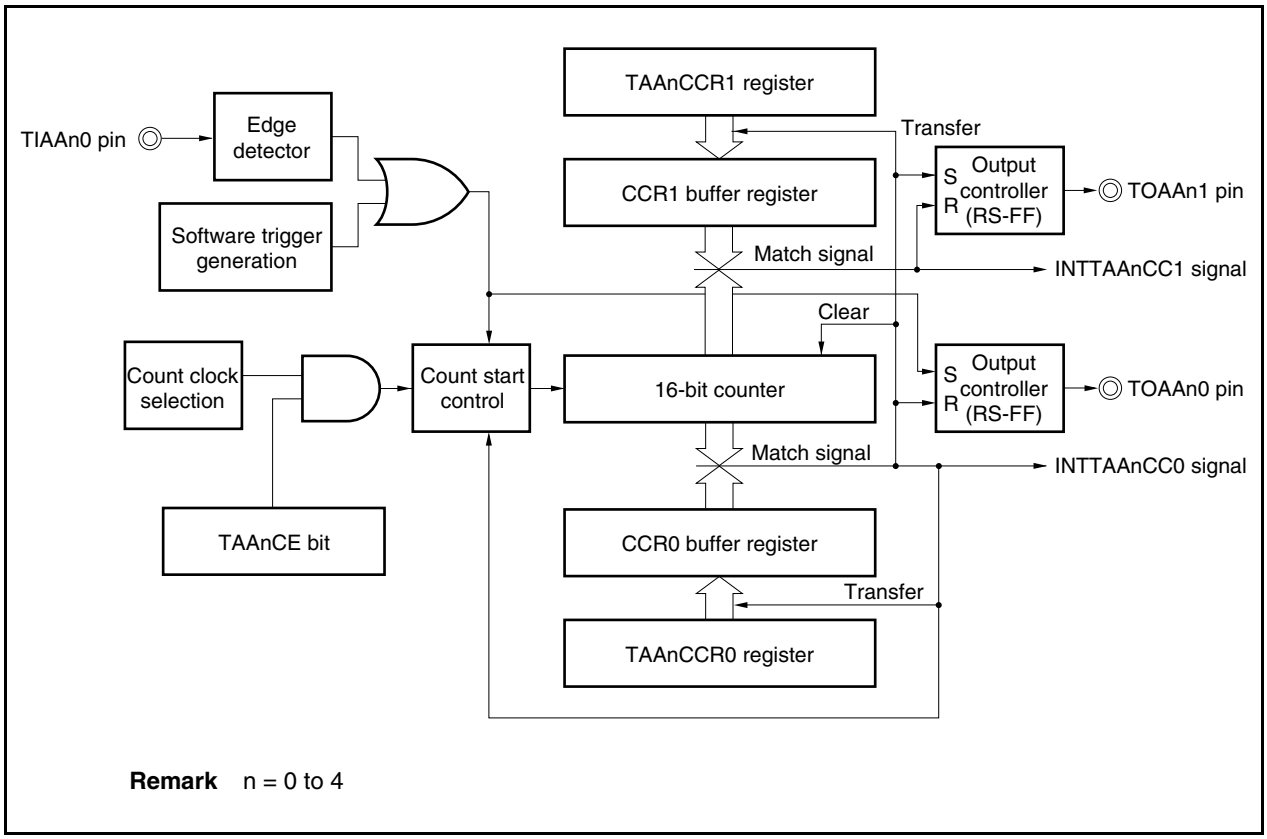
In the external trigger pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the timing of changing the output signal of the TOAAn1 pin.

7.5.4 One-shot pulse output mode (TAAnMD2 to TAAnMD0 bits = 011)

In the one-shot pulse output mode, 16-bit timer/event counter AA waits for a trigger when the TAAnCTL0.TAAnCE bit is set to 1. When the valid edge of an external trigger input is detected, 16-bit timer/event counter AA starts counting, and outputs a one-shot pulse from the TOAAn1 pin.

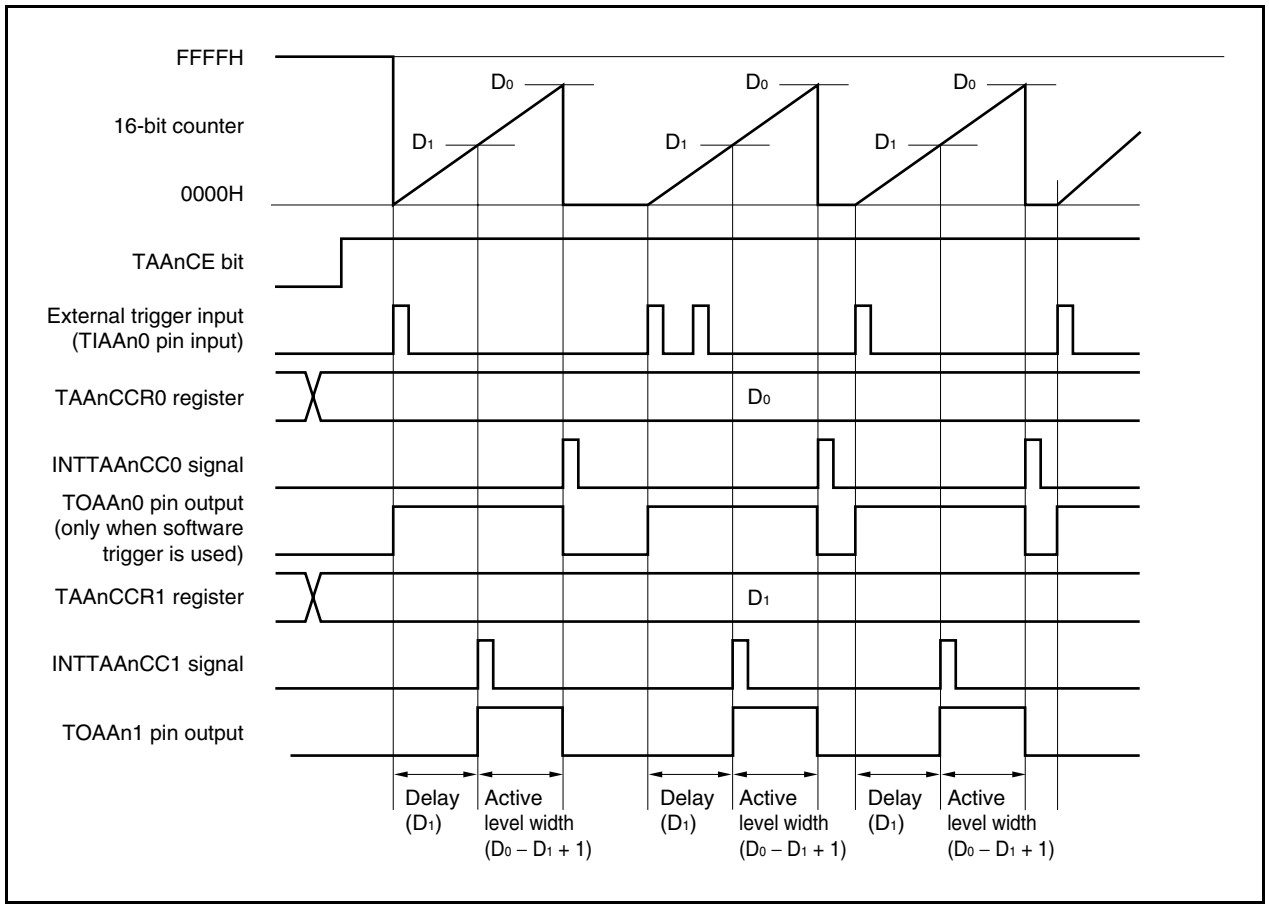
Instead of the external trigger, a software trigger can also be generated to output the pulse. When the software trigger is used, the TOAAn0 pin outputs the active level while the 16-bit counter is counting, and the inactive level when the counter is stopped (waiting for a trigger).

Figure 7-20. Configuration in One-Shot Pulse Output Mode



Remark n = 0 to 4

Figure 7-21. Basic Timing in One-Shot Pulse Output Mode



When the TAAnCE bit is set to 1, 16-bit timer/event counter AA waits for a trigger. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a one-shot pulse from the TOAAn1 pin. After the one-shot pulse is output, the 16-bit counter is set to FFFFH, stops counting, 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 TAAnCCR1 register) × Count clock cycle

Active level width = (Set value of TAAnCCR0 register − Set value of TAAnCCR1 register + 1) × Count clock cycle

The compare match interrupt request signal INTTAAnCC0 is generated when the 16-bit counter counts after its count value matches the value of the CCR0 buffer register. The compare match interrupt request signal INTTAAnCC1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The valid edge of an external trigger input or setting the software trigger (TAAnCTL1.TAAnEST bit) to 1 is used as the trigger.

Remark n = 0 to 4

Figure 7-22. Register Setting for Operation in One-Shot Pulse Output Mode (1/2)

(a) TAAn control register 0 (TAAnCTL0)

	TAAnCE					TAAnCKS2	TAAnCKS1	TAAnCKS0
TAAnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock

0: Stop counting
1: Enable counting

(b) TAAn control register 1 (TAAnCTL1)

	TAAnSYE	TAAnEST	TAAnEEE			TAAnMD2	TAAnMD1	TAAnMD0
TAAnCTL1	0	0/1	0	0	0	0	1	1

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

Generate software trigger
when 1 is written

(c) TAAn I/O control register 0 (TAAnIOC0)

					TAAnOL1	TAAnOE1	TAAnOL0	TAAnOE0
TAAnIOC0	0	0	0	0	0/1	0/1	0/1Note	0/1Note

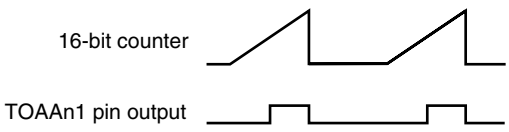
0: Disable TOAAn0 pin output
1: Enable TOAAn0 pin output

Setting of output level while
operation of TOAAn0 pin is disabled
0: Low level
1: High level

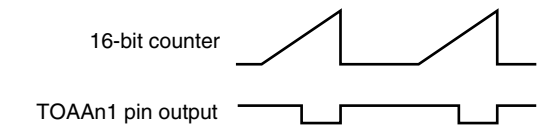
0: Disable TOAAn1 pin output
1: Enable TOAAn1 pin output

Specifies active level of
TOAAn1 pin output
0: Active-high
1: Active-low

• When TAAnOL1 bit = 0



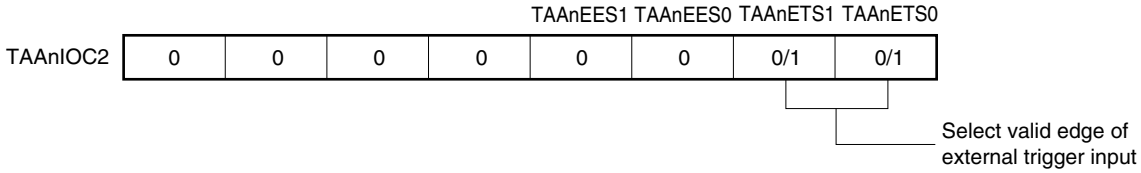
• When TAAnOL1 bit = 1



Note Clear this bit to 0 when the TOAAn0 pin is not used in the one-shot pulse output mode.

Figure 7-22. Register Setting for Operation in One-Shot Pulse Output Mode (2/2)

(d) TAAn I/O control register 2 (TAAnIOC2)



(e) TAAn counter read buffer register (TAAnCNT)

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

(f) TAAn capture/compare registers 0 and 1 (TAAnCCR0 and TAAnCCR1)

If D_0 is set to the TAAnCCR0 register and D_1 to the TAAnCCR1 register, the active level width and output delay period of the one-shot pulse are as follows.

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

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

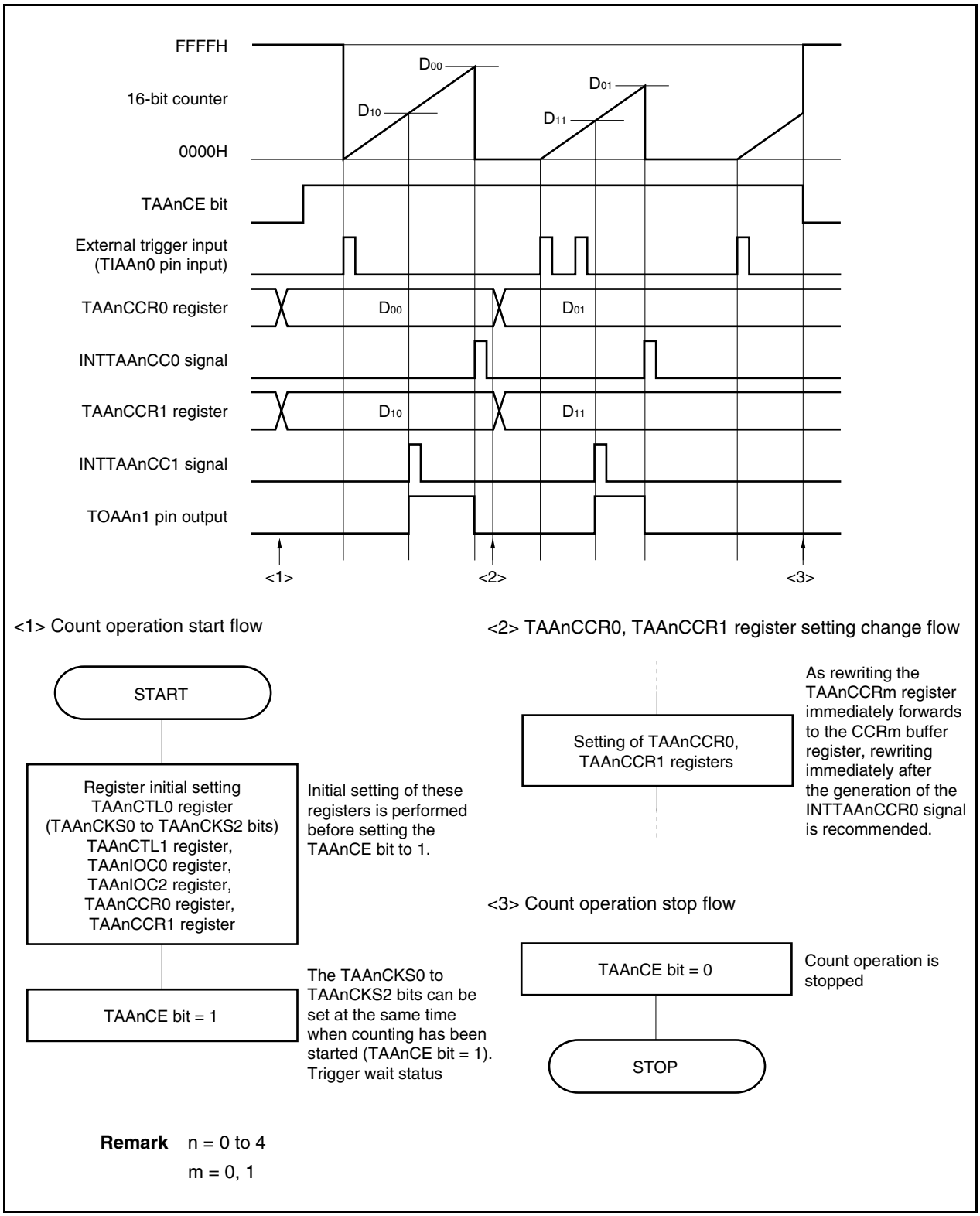
Caution One-shot pulses are not output even in the one-shot pulse output mode, if the set value in the TAAnCCR1 register is greater than that value in the TAAnCCR0 register.

Remarks 1. TAAn I/O control register 1 (TAAnIOC1) and TAAn option register 0 (TAAnOPT0) are not used in the one-shot pulse output mode.

2. n = 0 to 4

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

Figure 7-23. Software Processing Flow in One-Shot Pulse Output Mode



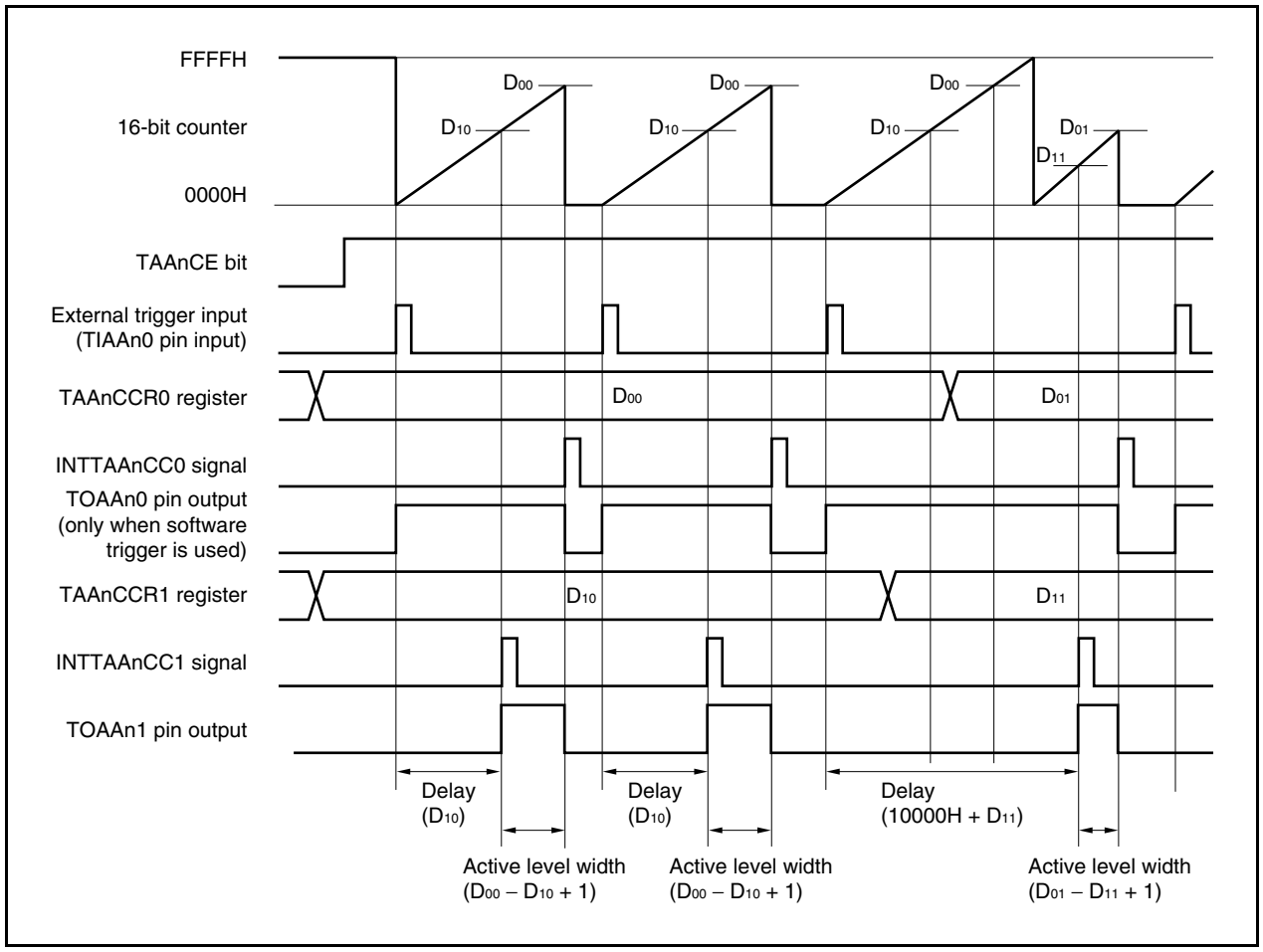
Remark n = 0 to 4
m = 0, 1

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

(a) Note on rewriting TAAnCCRm register

To change the set value of the TAAnCCRm register to a smaller value, stop counting once, and then change the set value.

If the value of the TAAnCCRm register is rewritten to a smaller value during counting, the 16-bit counter may overflow.



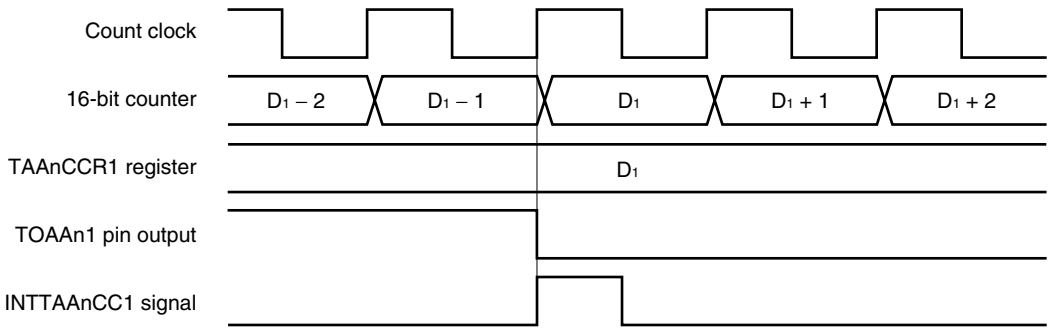
When the TAAnCCR0 register is rewritten from D_{00} to D_{01} and the TAAnCCR1 register from D_{10} to D_{11} where $D_{00} > D_{01}$ and $D_{10} > D_{11}$, if the TAAnCCR1 register is rewritten when the count value of the 16-bit counter is greater than D_{11} and less than D_{10} and if the TAAnCCR0 register is rewritten when the count value is greater than D_{01} and less than D_{00}, each set value is reflected as soon as the register has been rewritten and compared with the count value. The counter counts up to FFFFH and then counts up again from 0000H. When the count value matches D_{11}, the counter generates the INTTAAnCC1 signal and asserts the TOAAn1 pin. When the count value matches D_{01}, the counter generates the INTTAAnCC0 signal, deasserts the TOAAn1 pin, and stops counting.

Therefore, the counter may output a pulse with a delay period or active period different from that of the one-shot pulse that is originally expected.

Remark n = 0 to 4
m = 0, 1

(b) Generation timing of compare match interrupt request signal (INTTAAnCC1)

The generation timing of the INTTAAnCC1 signal in the one-shot pulse output mode is different from other INTTAAnCC1 signals; the INTTAAnCC1 signal is generated when the count value of the 16-bit counter matches the value of the TAAnCCR1 register.



Remark n = 0 to 4

Usually, the INTTAAnCC1 signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TAAnCCR1 register.

In the one-shot pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the TOAAn1 pin.

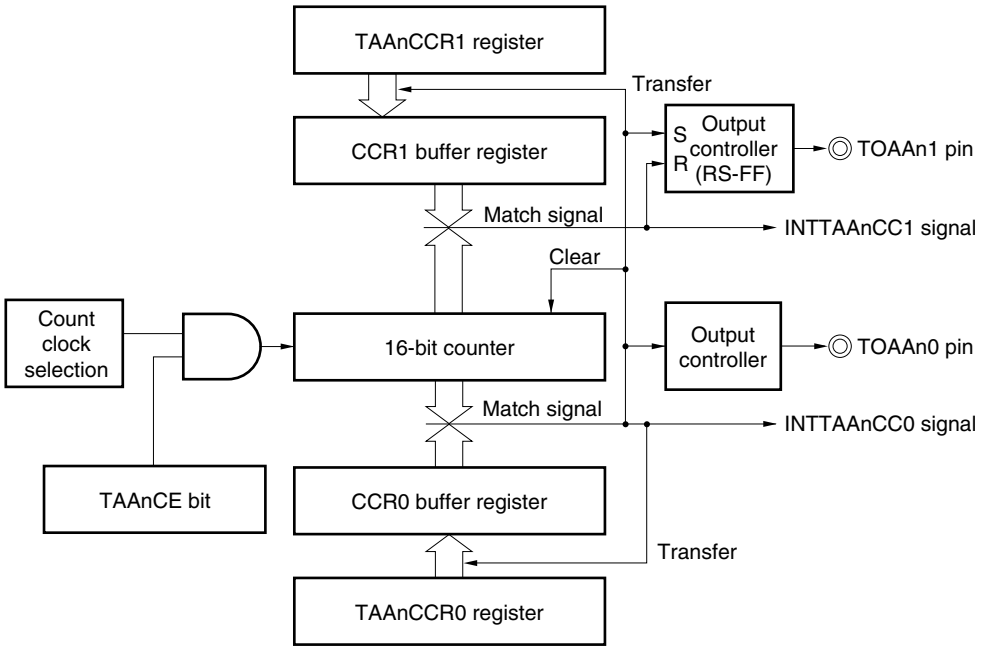
Remark n = 0 to 4

7.5.5 PWM output mode (TAAnMD2 to TAAnMD0 bits = 100)

In the PWM output mode, a PWM waveform is output from the TOAAn1 pin when the TAAnCTL0.TAAnCE bit is set to 1.

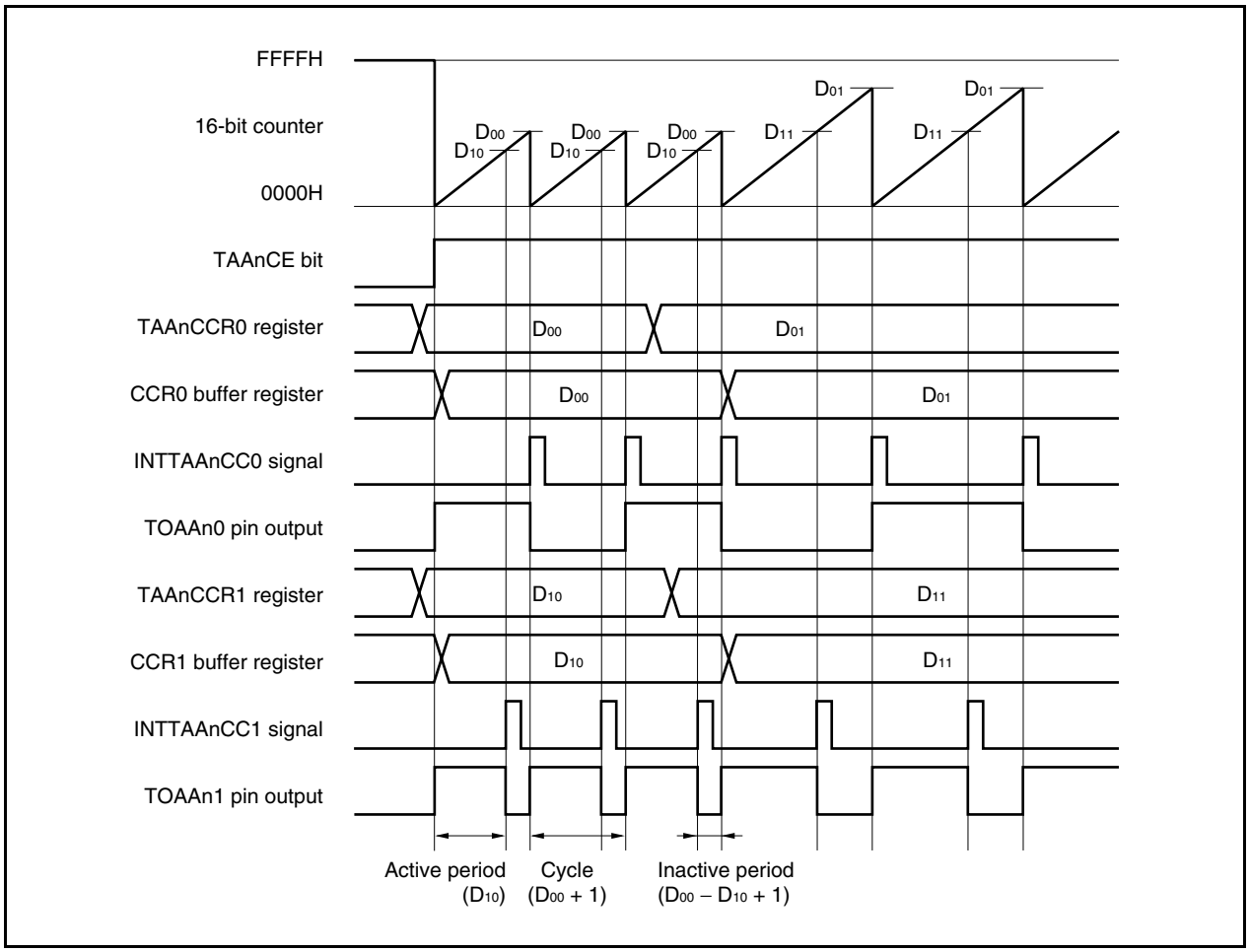
In addition, a pulse with one cycle of the PWM waveform as half its cycle is output from the TOAAn0 pin.

Figure 7-24. Configuration in PWM Output Mode



Remark n = 0 to 4

Figure 7-25. Basic Timing in PWM Output Mode



When the TAAnCE bit is set to 1, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a PWM waveform from the TOAAn1 pin.

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

Active level width = (Set value of TAAnCCR1 register) × Count clock cycle
Cycle = (Set value of TAAnCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TAAnCCR1 register)/(Set value of TAAnCCR0 register + 1)

The PWM waveform can be changed by rewriting the TAAnCCRm register while the counter is operating. The newly written value is reflected when the count value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

The compare match interrupt request signal INTTAAnCC0 is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal INTTAAnCC1 is generated when the count value of the 16-bit counter matches the value of the CCR1 buffer register.

The value set to the TAAnCCRm register is transferred to the CCRm buffer register when the count value of the 16-bit counter matches the value of the CCRm buffer register and the 16-bit counter is cleared to 0000H.

Remark n = 0 to 4, m = 0, 1

Figure 7-26. Setting of Registers in PWM Output Mode (1/2)

(a) TAAn control register 0 (TAAnCTL0)

	TAAnCE					TAAnCKS2	TAAnCKS1	TAAnCKS0
TAAnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock[Note 1]

0: Stop counting
1: Enable counting

(b) TAAn control register 1 (TAAnCTL1)

	TAAnSYE	TAAnEST	TAAnEEE			TAAnMD2	TAAnMD1	TAAnMD0
TAAnCTL1	0	0	0/1	0	0	1	0	0

1, 0, 0:
PWM output mode

0: Operate on count clock selected by TAAnCKS0 to TAAnCKS2 bits
1: Count external event input signal

(c) TAAn I/O control register 0 (TAAnIOC0)

					TAAnOL1	TAAnOE1	TAAnOL0	TAAnOE0
TAAnIOC0	0	0	0	0	0/1	0/1	0/1[Note 2]	0/1[Note 2]

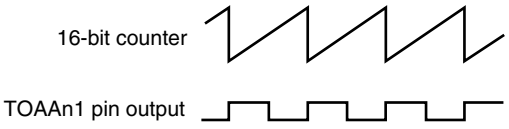
0: Disable TOAAn0 pin output
1: Enable TOAAn0 pin output

Setting of output level while operation of TOAAn0 pin is disabled
0: Low level
1: High level

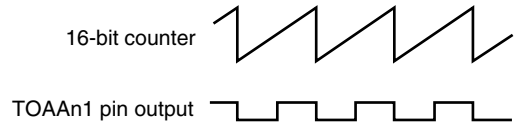
0: Disable TOAAn1 pin output
1: Enable TOAAn1 pin output

Specifies active level of TOAAn1 pin output
0: Active-high
1: Active-low

- When TAAnOL1 bit = 0



- When TAAnOL1 bit = 1

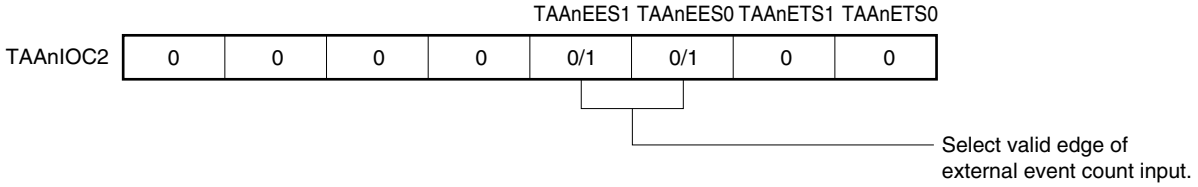


Notes 1. The setting is invalid when the TAAnCTL1.TAAnEEE bit = 1.

2. Clear this bit to 0 when the TOAAn0 pin is not used in the PWM output mode.

Figure 7-26. Register Setting in PWM Output Mode (2/2)

(d) TAAn I/O control register 2 (TAAnIOC2)



(e) TAAn counter read buffer register (TAAnCNT)

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

(f) TAAn capture/compare registers 0 and 1 (TAAnCCR0 and TAAnCCR1)

If D_0 is set to the TAAnCCR0 register and D_1 to the TAAnCCR1 register, the cycle and active level of the PWM waveform are as follows.

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

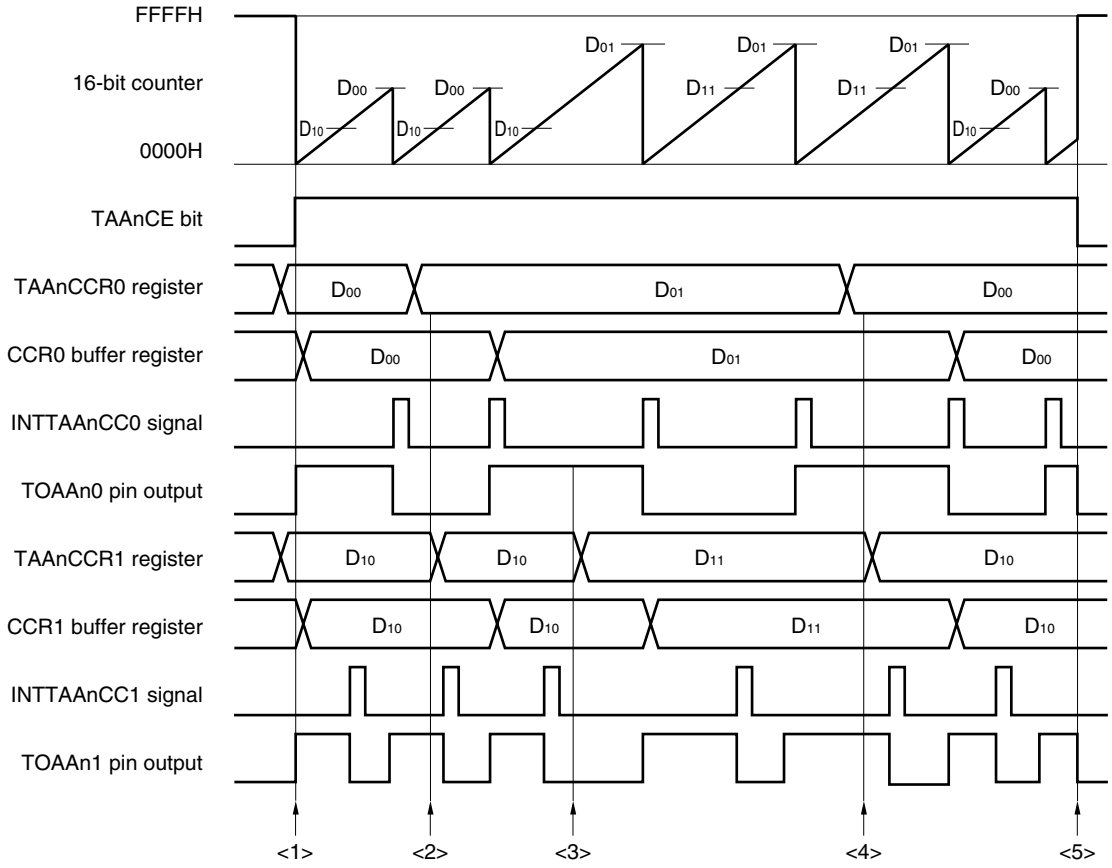
Active level width = $D_1 \times$ Count clock cycle

Remarks 1. TAAn I/O control register 1 (TAAnIOC1) and TAAn option register 0 (TAAnOPT0) are not used in the PWM output mode.

2. n = 0 to 4

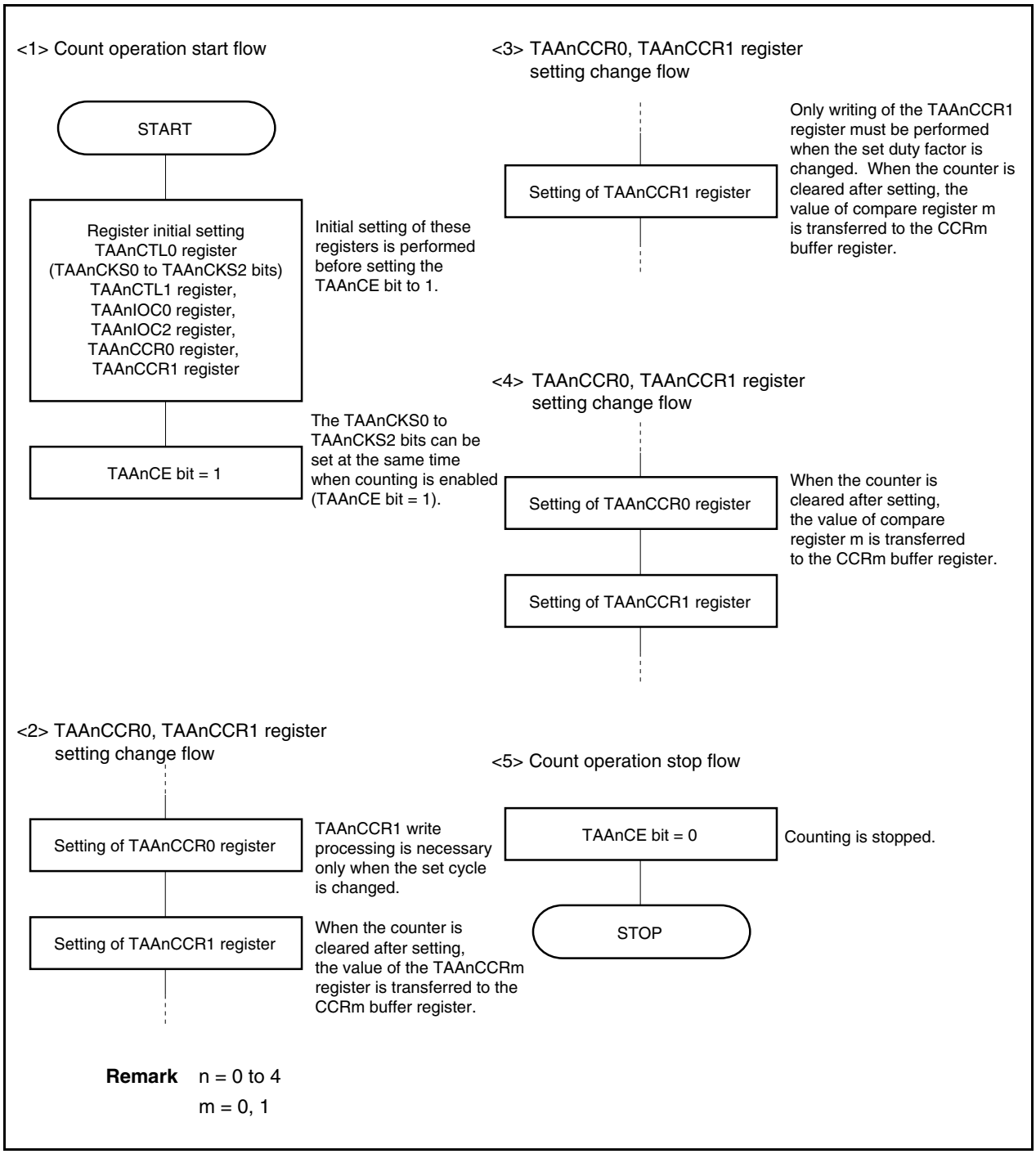
(1) Operation flow in PWM output mode

Figure 7-27. Software Processing Flow in PWM Output Mode (1/2)



Remark n = 0 to 4
m = 0, 1

Figure 7-27. Software Processing Flow in PWM Output Mode (2/2)



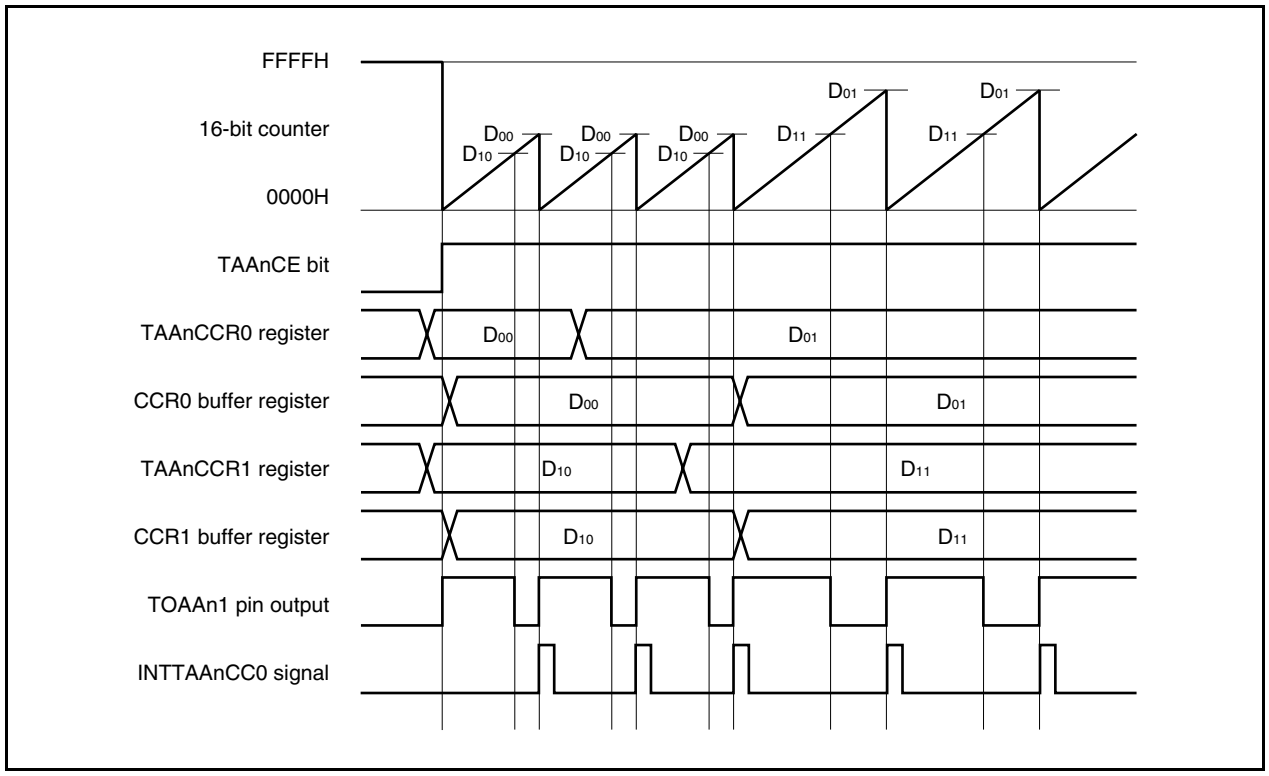
Remark n = 0 to 4
m = 0, 1

(2) PWM output mode operation timing

(a) Changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TAAnCCR1 register last.

Rewrite the TAAnCCRm register after writing the TAAnCCR1 register after the INTTAAnCC1 signal is detected.



To transfer data from the TAAnCCRm register to the CCRm buffer register, the TAAnCCR1 register must be written.

To change both the cycle and active level of the PWM waveform at this time, first set the cycle to the TAAnCCR0 register and then set the active level to the TAAnCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TAAnCCR0 register, and then write the same value to the TAAnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, only the TAAnCCR1 register has to be set.

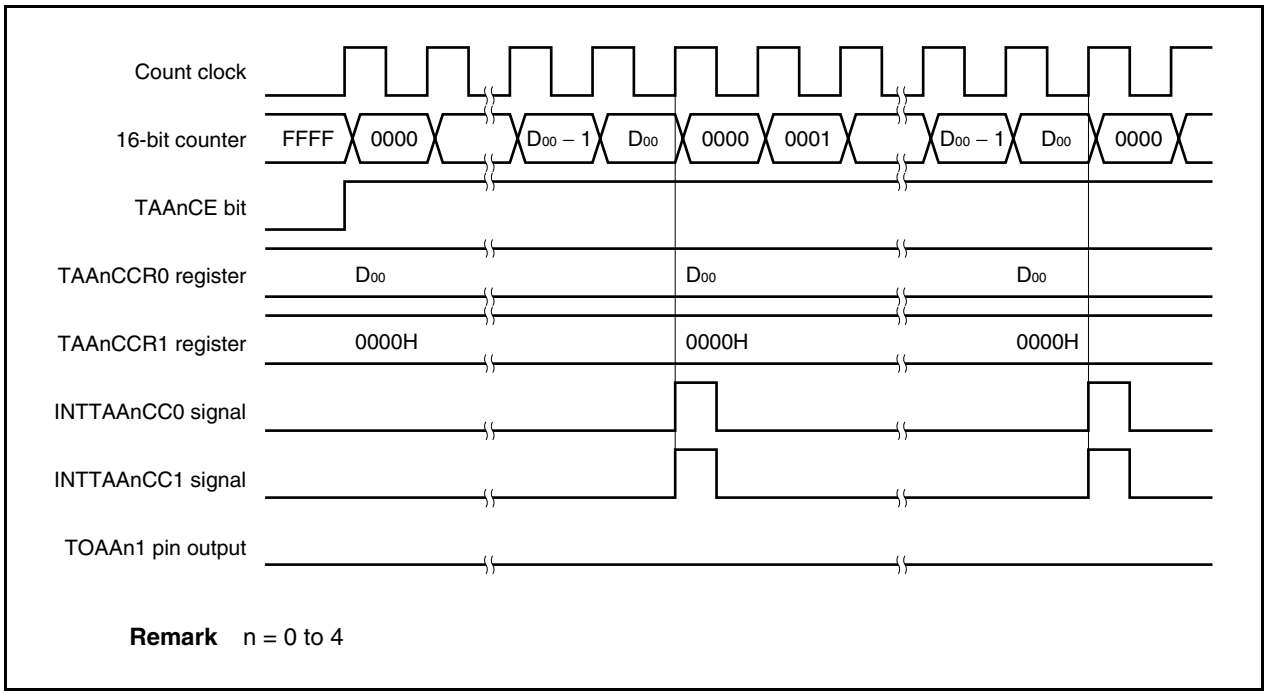
After data is written to the TAAnCCR1 register, the value written to the TAAnCCRm register is transferred to the CCRm buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TAAnCCR0 or TAAnCCR1 register again after writing the TAAnCCR1 register once, do so after the INTTAAnCC0 signal is generated. Otherwise, the value of the CCRm buffer register may become undefined because the timing of transferring data from the TAAnCCRm register to the CCRm buffer register conflicts with writing the TAAnCCRm register.

Remark n = 0 to 4, m = 0, 1

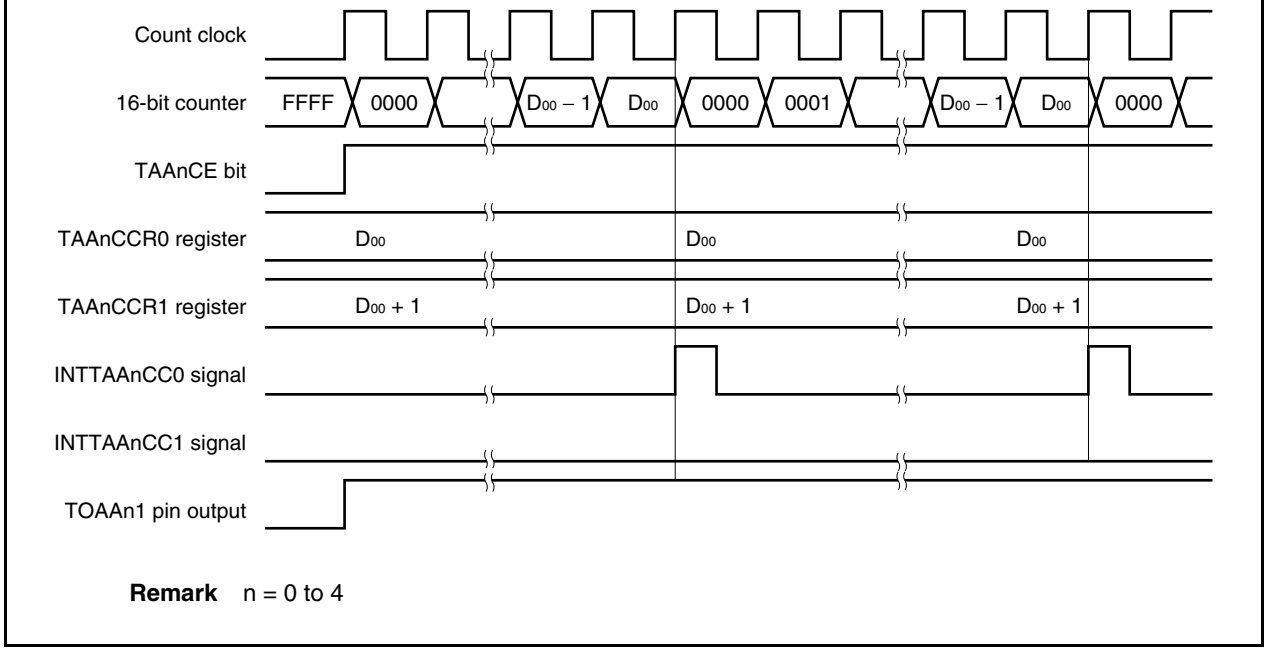
(b) 0%/100% output of PWM waveform

To output a 0% waveform, set the TAAnCCR1 register to 0000H. If the set value of the TAAnCCR0 register is FFFFH, the INTTAAnCC1 signal is generated periodically.



Remark n = 0 to 4

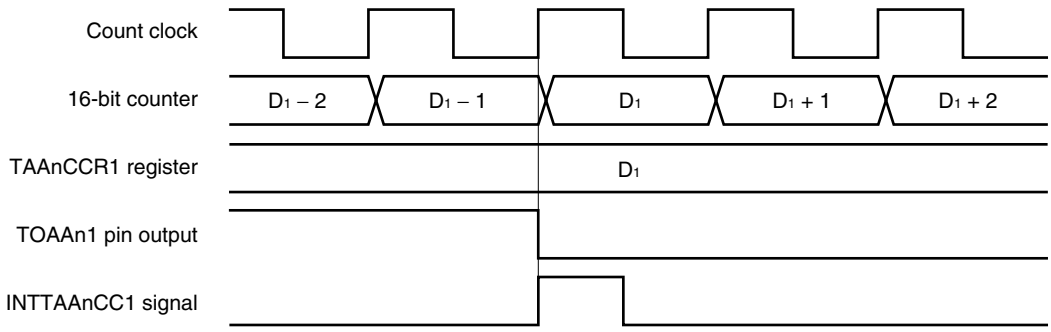
To output a 100% waveform, set a value of (set value of TAAnCCR0 register + 1) to the TAAnCCR1 register. If the set value of the TAAnCCR0 register is FFFFH, 100% output cannot be produced.



Remark n = 0 to 4

(c) Generation timing of compare match interrupt request signal (INTTAAnCC1)

The timing of generation of the INTTAAnCC1 signal in the PWM output mode differs from the timing of other INTTAAnCC1 signals; the INTTAAnCC1 signal is generated when the count value of the 16-bit counter matches the value of the TAAnCCR1 register.



Remark n = 0 to 4

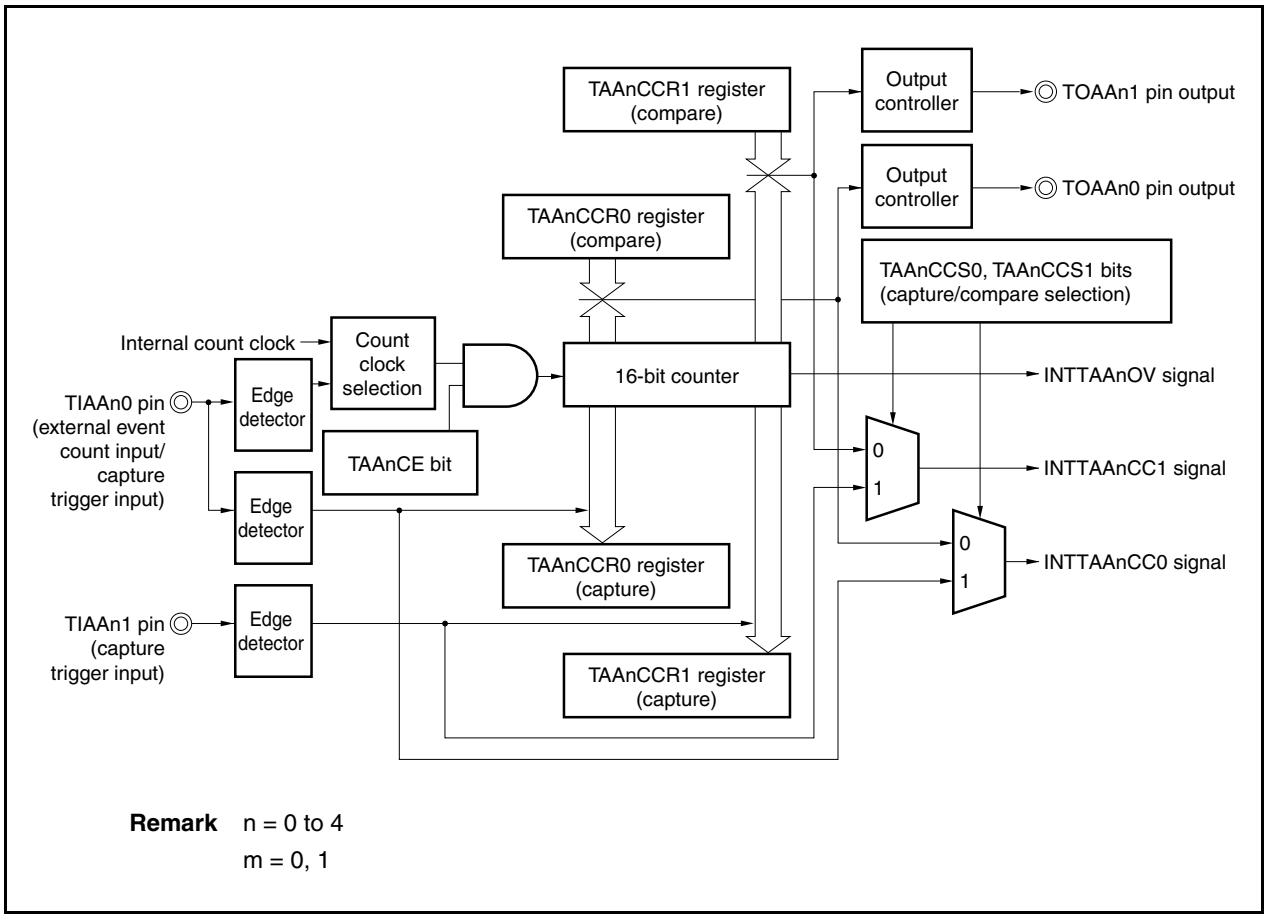
Usually, the INTTAAnCC1 signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TAAnCCR1 register.

In the PWM output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the output signal of the TOAAn1 pin.

7.5.6 Free-running timer mode (TAAnMD2 to TAAnMD0 bits = 101)

In the free-running timer mode, 16-bit timer/event counter AA starts counting when the TAAnCTL0.TAAnCE bit is set to 1. At this time, the TAAnCCRm register can be used as a compare register or a capture register, depending on the setting of the TAAnOPT0.TAAnCCS0 and TAAnOPT0.TAAnCCS1 bits.

Figure 7-28. Configuration in Free-Running Timer Mode



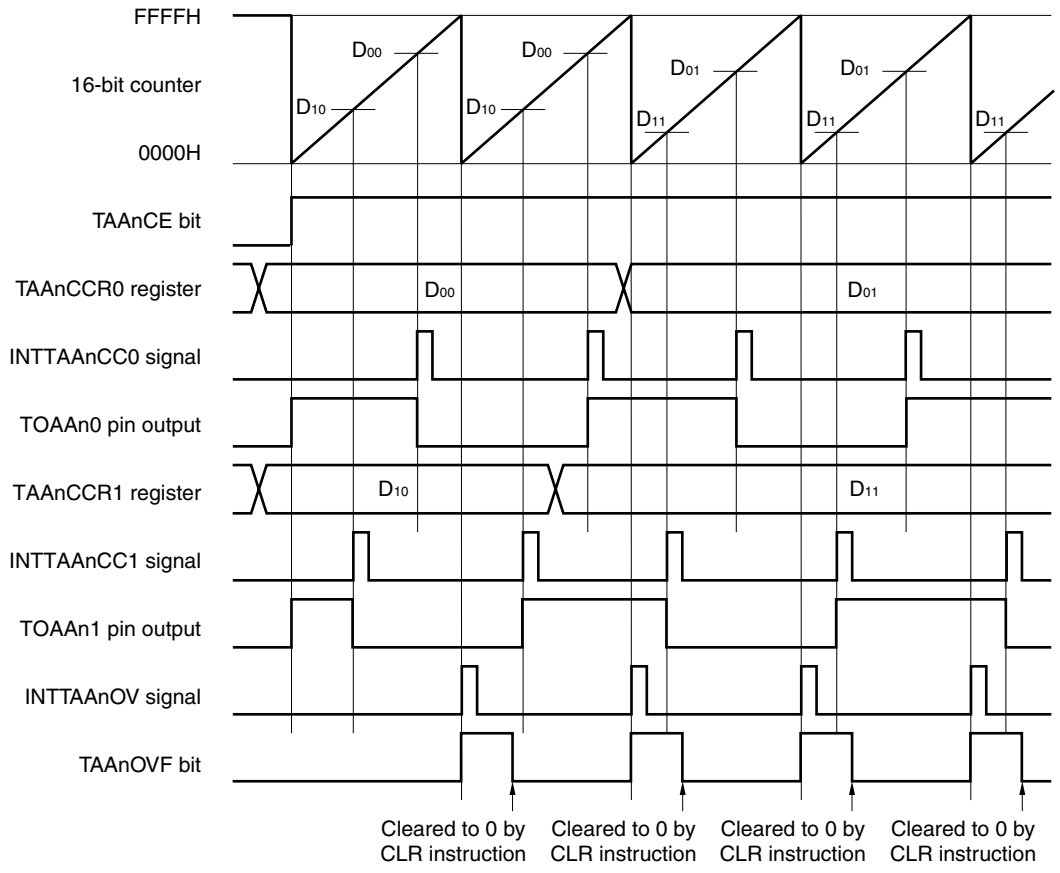
Remark n = 0 to 4
m = 0, 1

When the TAAnCE bit is set to 1, 16-bit timer/event counter AA starts counting, and the output signals of the TOAAn0 and TOAAn1 pins are inverted. When the count value of the 16-bit counter later matches the set value of the TAAnCCRm register, a compare match interrupt request signal (INTTAAnCCm) is generated, and the output signal of the TOAAnm pin is inverted.

The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTAAnOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TAAnOPT0.TAAnOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction by software.

The TAAnCCRm register can be rewritten while the counter is operating. If it is rewritten, the new value is reflected at that time, and compared with the count value.

Figure 7-29. Basic Timing in Free-Running Timer Mode (Compare Function)

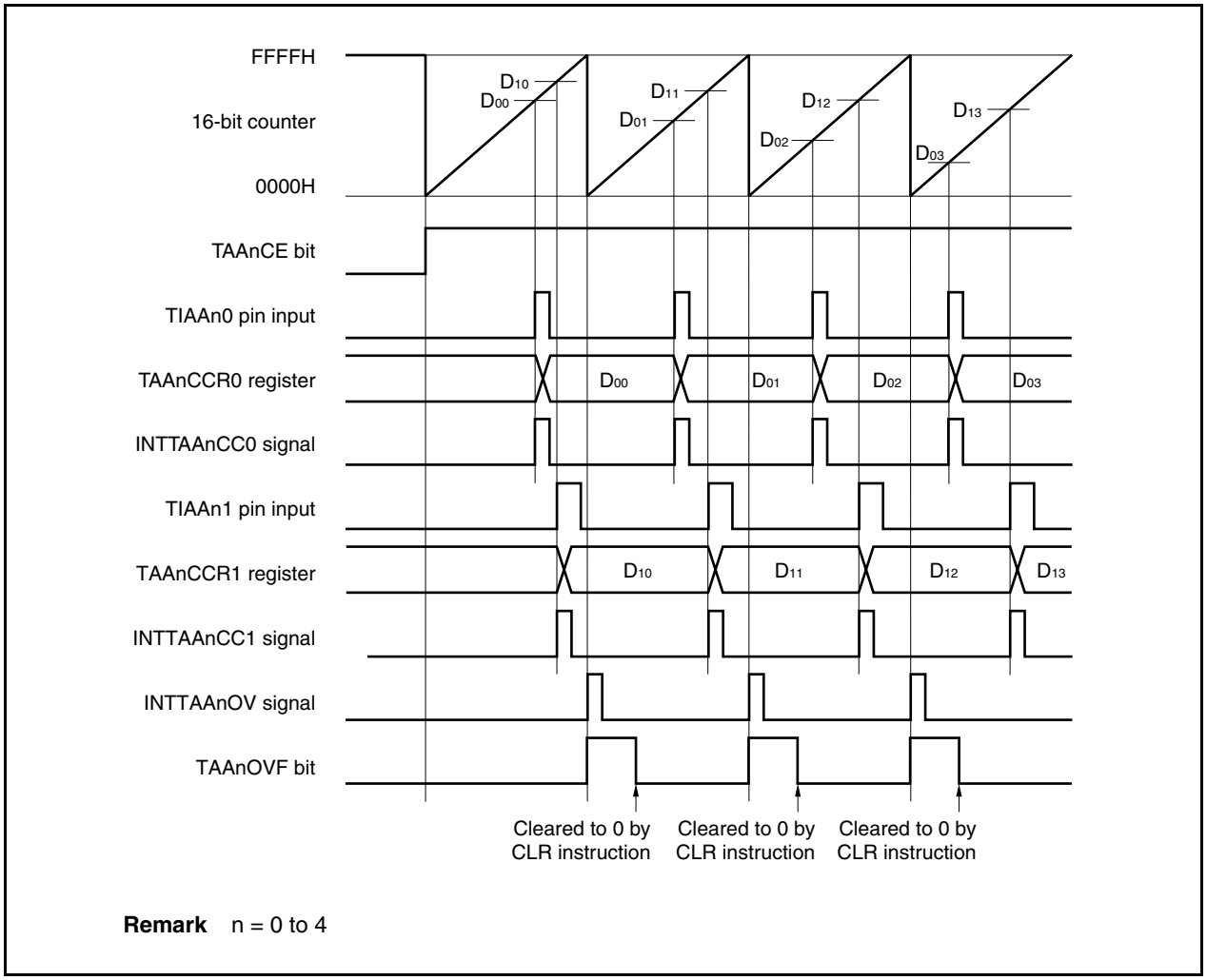


Remark n = 0 to 4
m = 0, 1

When the TAAnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIAAnm pin is detected, the count value of the 16-bit counter is stored in the TAAnCCRm register, and a capture interrupt request signal (INTTAAnCCm) is generated.

The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTAAnOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TAAnOPT0.TAAnOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction by software.

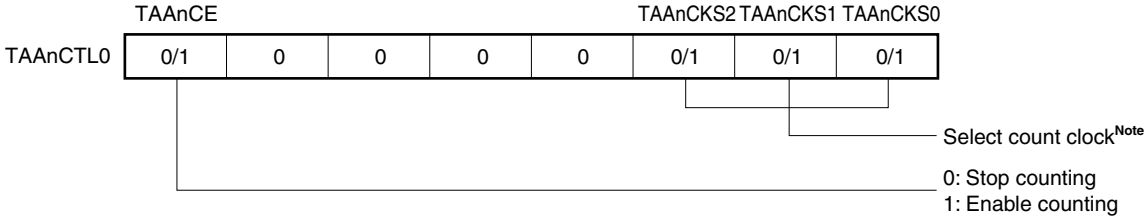
Figure 7-30. Basic Timing in Free-Running Timer Mode (Capture Function)



Remark n = 0 to 4

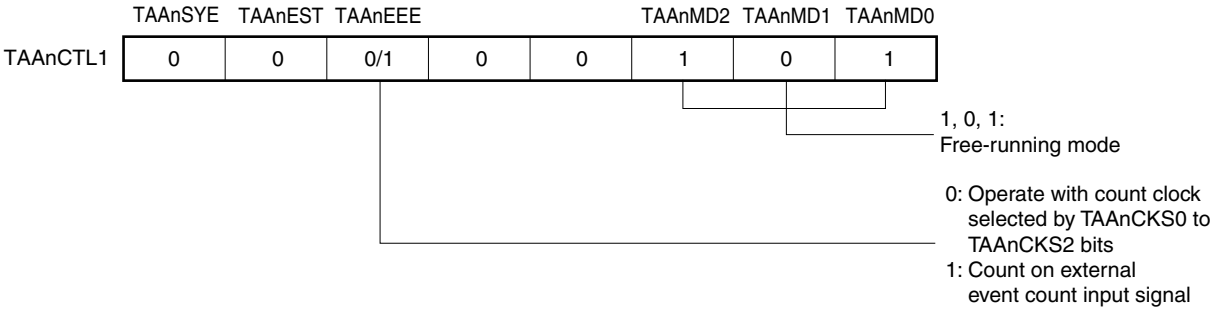
Figure 7-31. Register Setting in Free-Running Timer Mode (1/2)

(a) TAAn control register 0 (TAAnCTL0)



Note The setting is invalid when the TAAnCTL1.TAAnEEE bit = 1

(b) TAAn control register 1 (TAAnCTL1)



(c) TAAn I/O control register 0 (TAAnIOC0)

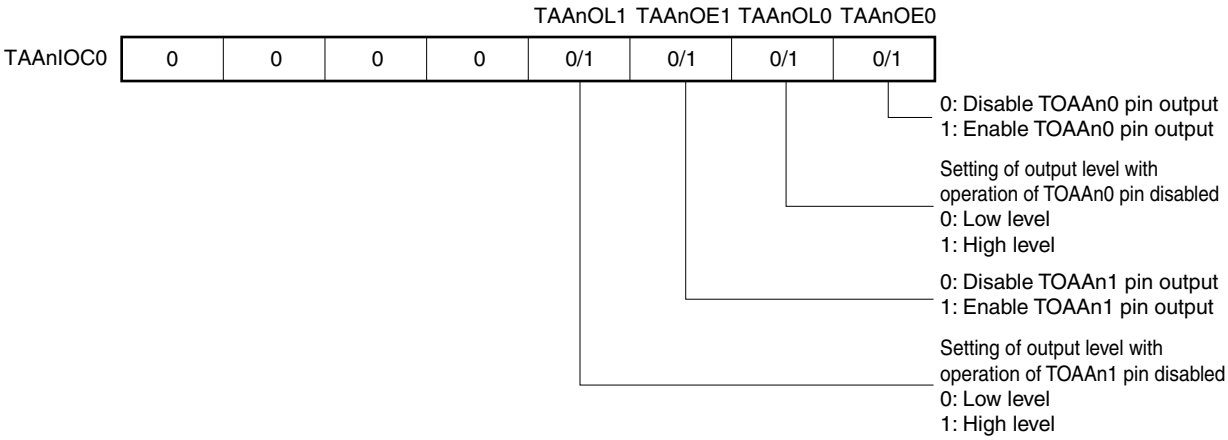
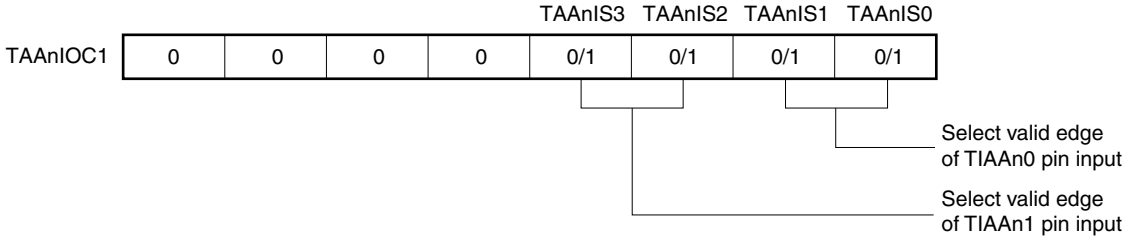
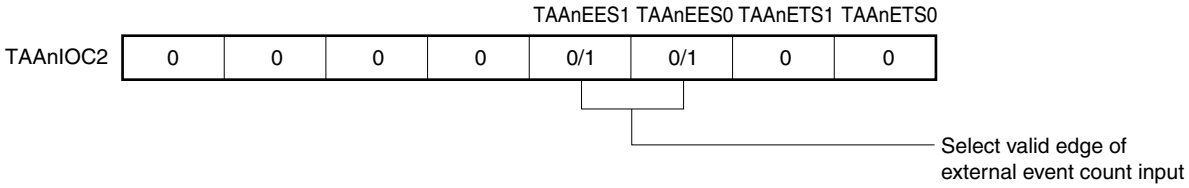


Figure 7-31. Register Setting in Free-Running Timer Mode (2/2)

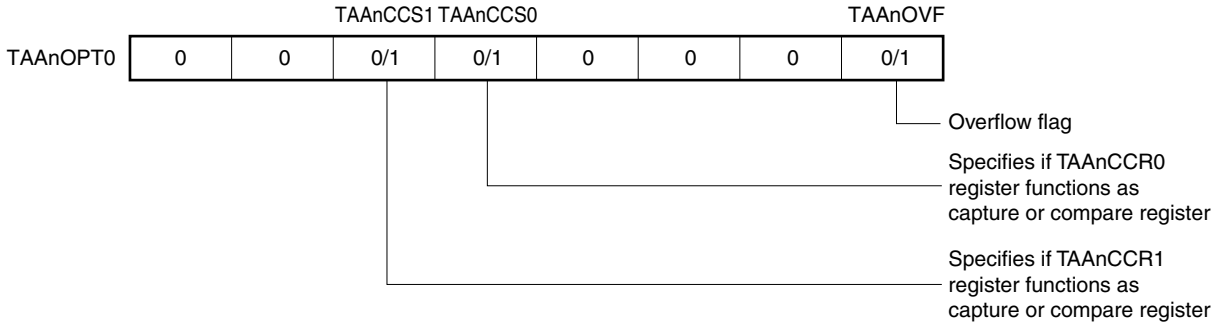
(d) TAAn I/O control register 1 (TAAnIOC1)



(e) TAAn I/O control register 2 (TAAnIOC2)



(f) TAAn option register 0 (TAAnOPT0)



(g) TAAn counter read buffer register (TAAnCNT)

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

(h) TAAn capture/compare registers 0 and 1 (TAAnCCR0 and TAAnCCR1)

These registers function as capture registers or compare registers depending on the setting of the TAAnOPT0.TAAnCCSm bit.

When the registers function as capture registers, they store the count value of the 16-bit counter when the valid edge input to the TIAAnm pin is detected.

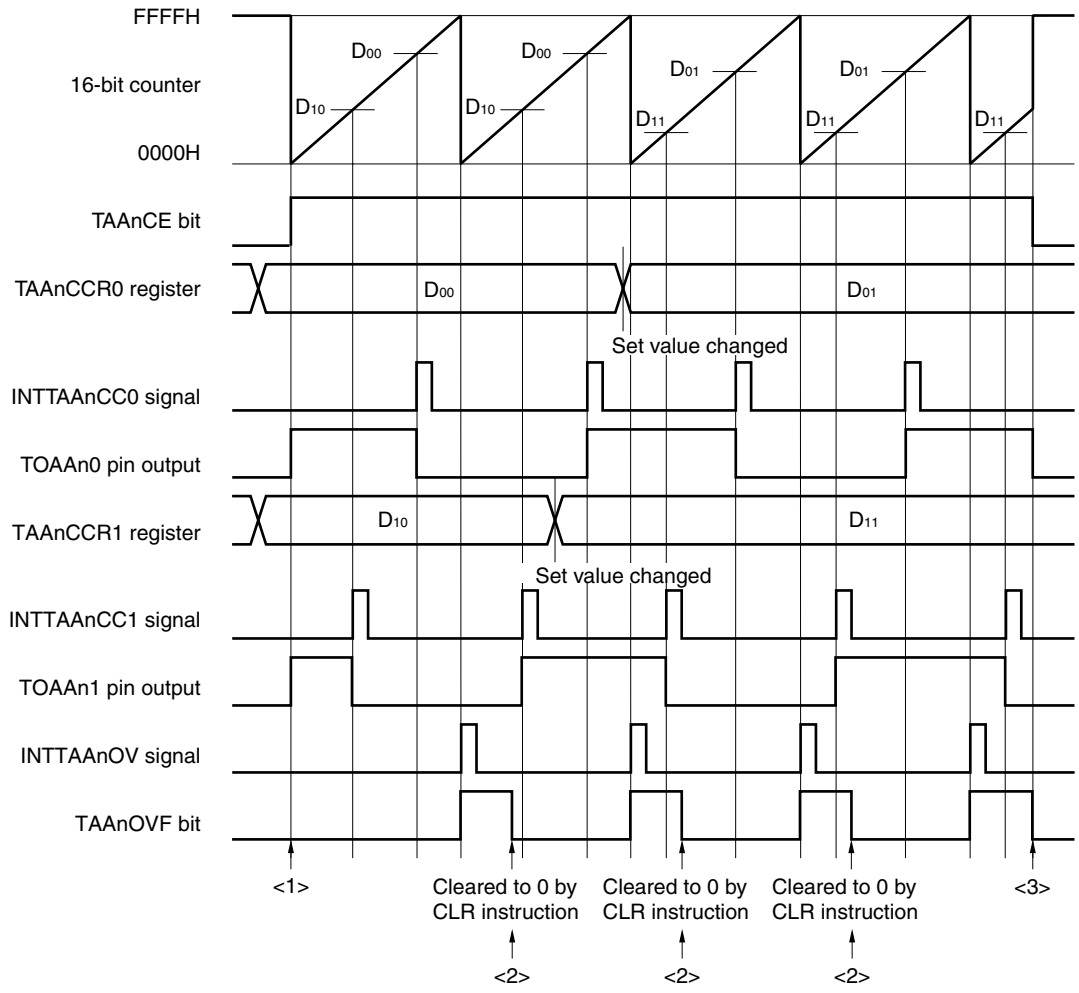
When the registers function as compare registers and when D_m is set to the TAAnCCRm register, the INTTAAnCCm signal is generated when the counter reaches (D_m + 1), and the output signal of the TOAAnm pin is inverted.

Remark n = 0 to 4
m = 0, 1

(1) Operation flow in free-running timer mode

(a) When using capture/compare register as compare register

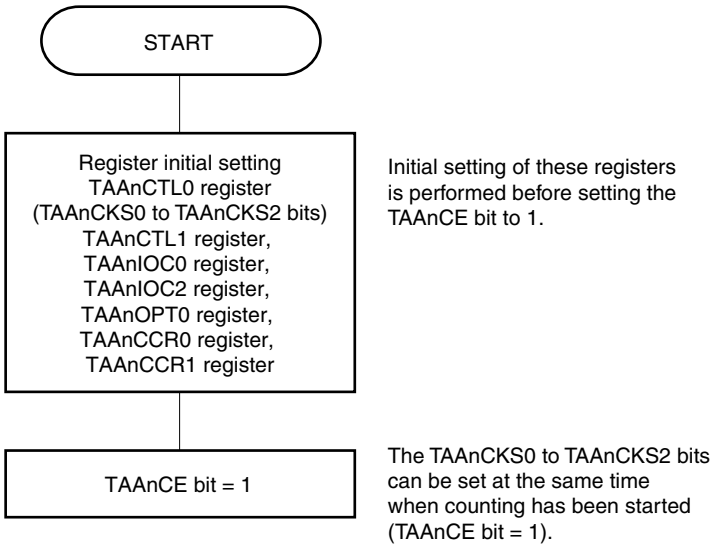
Figure 7-32. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)



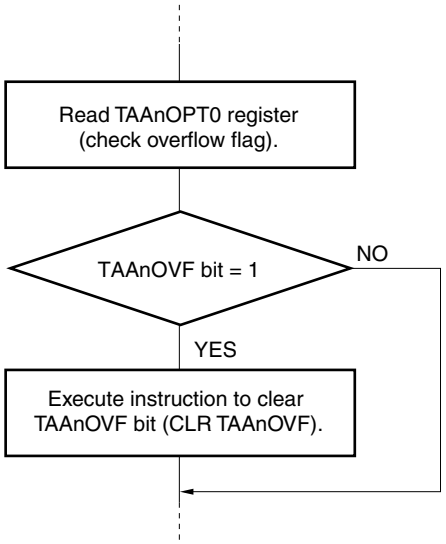
Remark n = 0 to 4

Figure 7-32. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)

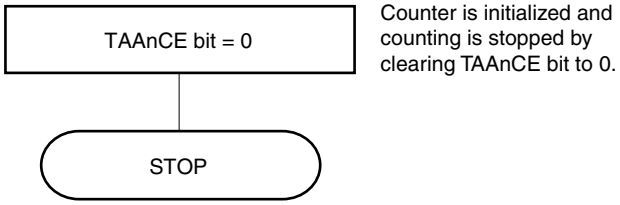
<1> Count operation start flow



<2> Overflow flag clear flow



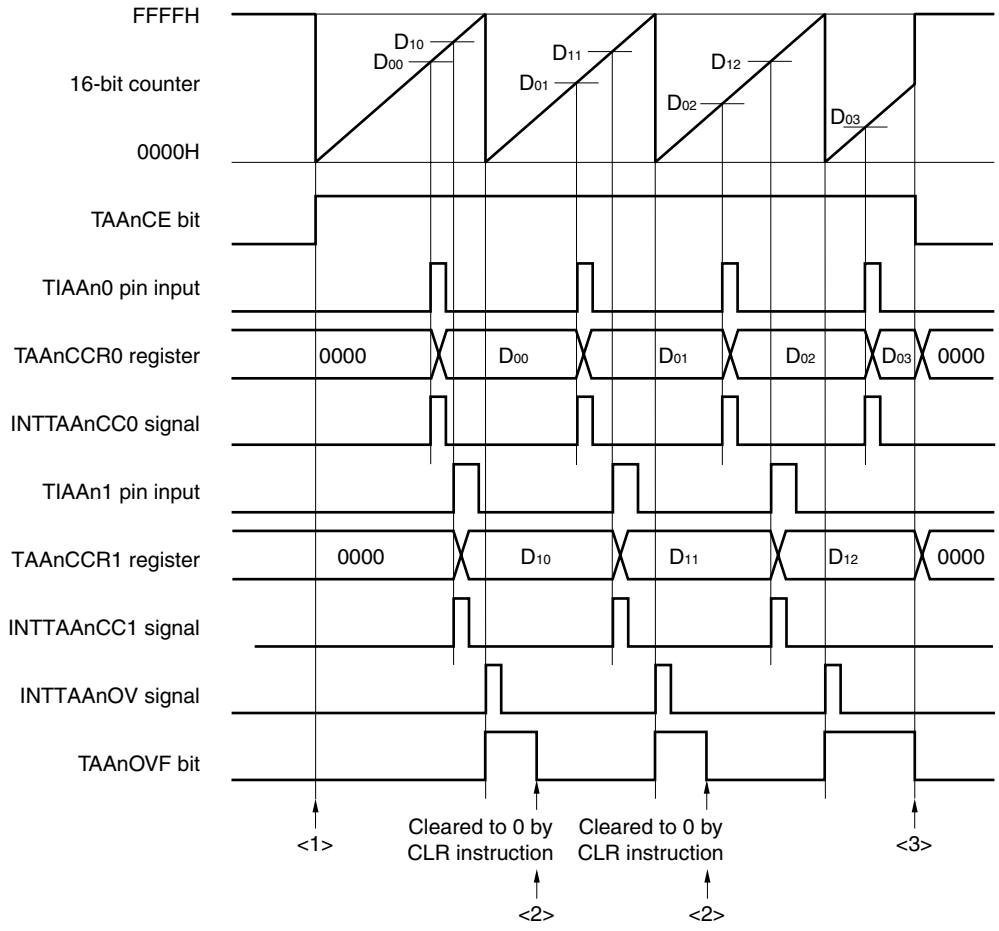
<3> Count operation stop flow



Remark n = 0 to 4

(b) When using capture/compare register as capture register

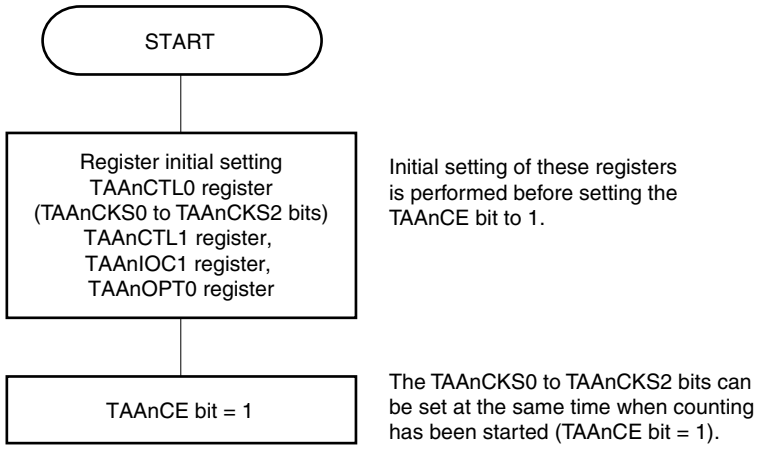
Figure 7-33. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)



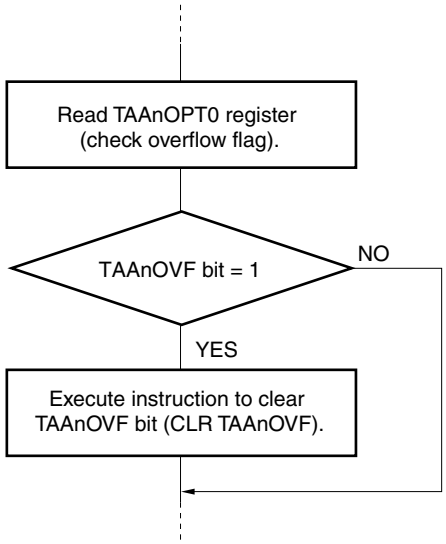
Remark n = 0 to 4

Figure 7-33. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)

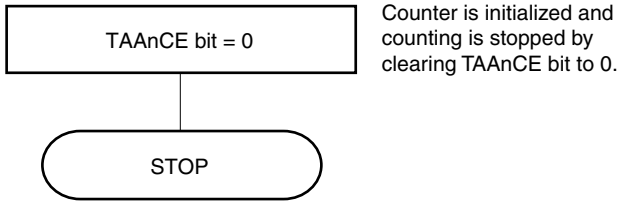
<1> Count operation start flow



<2> Overflow flag clear flow



<3> Count operation stop flow

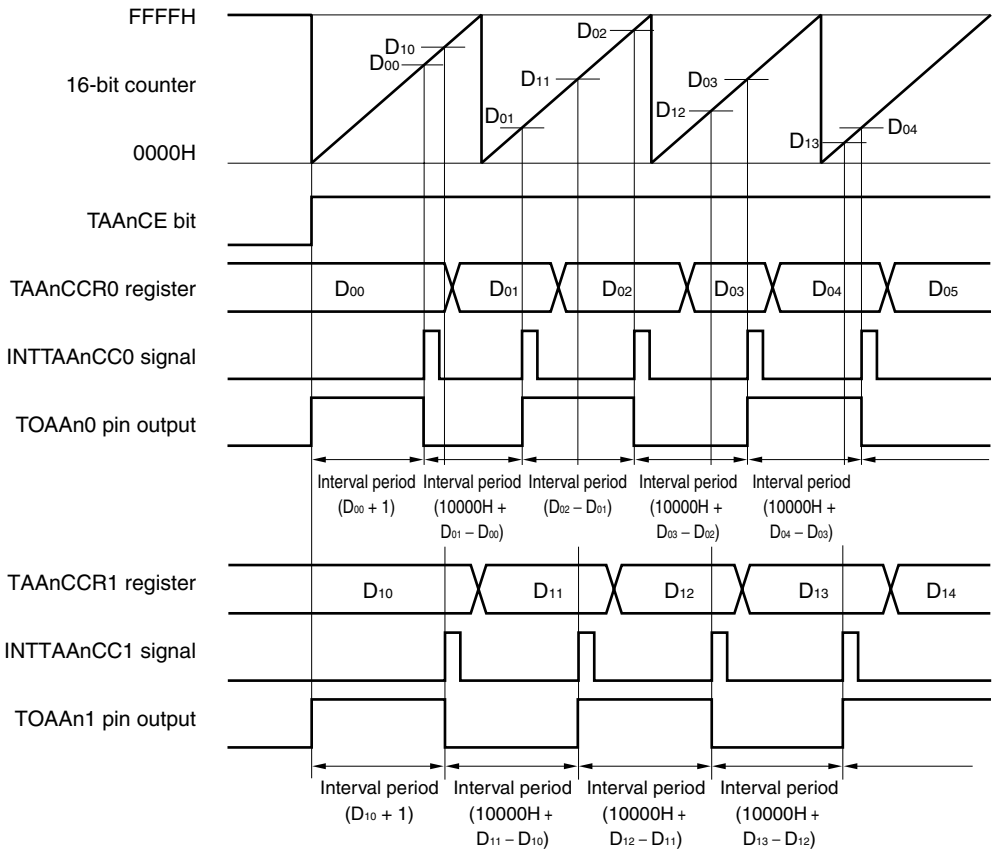


Remark n = 0 to 4

(2) Operation timing in free-running timer mode

(a) Interval operation with compare register

When 16-bit timer/event counter AA is used as an interval timer with the TAAnCCRm register used as a compare register, software processing is necessary for setting a comparison value to generate the next interrupt request signal each time the INTTAAnCCm signal has been detected.



When performing an interval operation in the free-running timer mode, two intervals can be set with one channel.

To perform the interval operation, the value of the corresponding TAAnCCRm register must be re-set in the interrupt servicing that is executed when the INTTAAnCCm signal is detected.

The set value for re-setting the TAAnCCRm register can be calculated by the following expression, where "D_m" is the interval period.

Compare register default value: $D_m - 1$

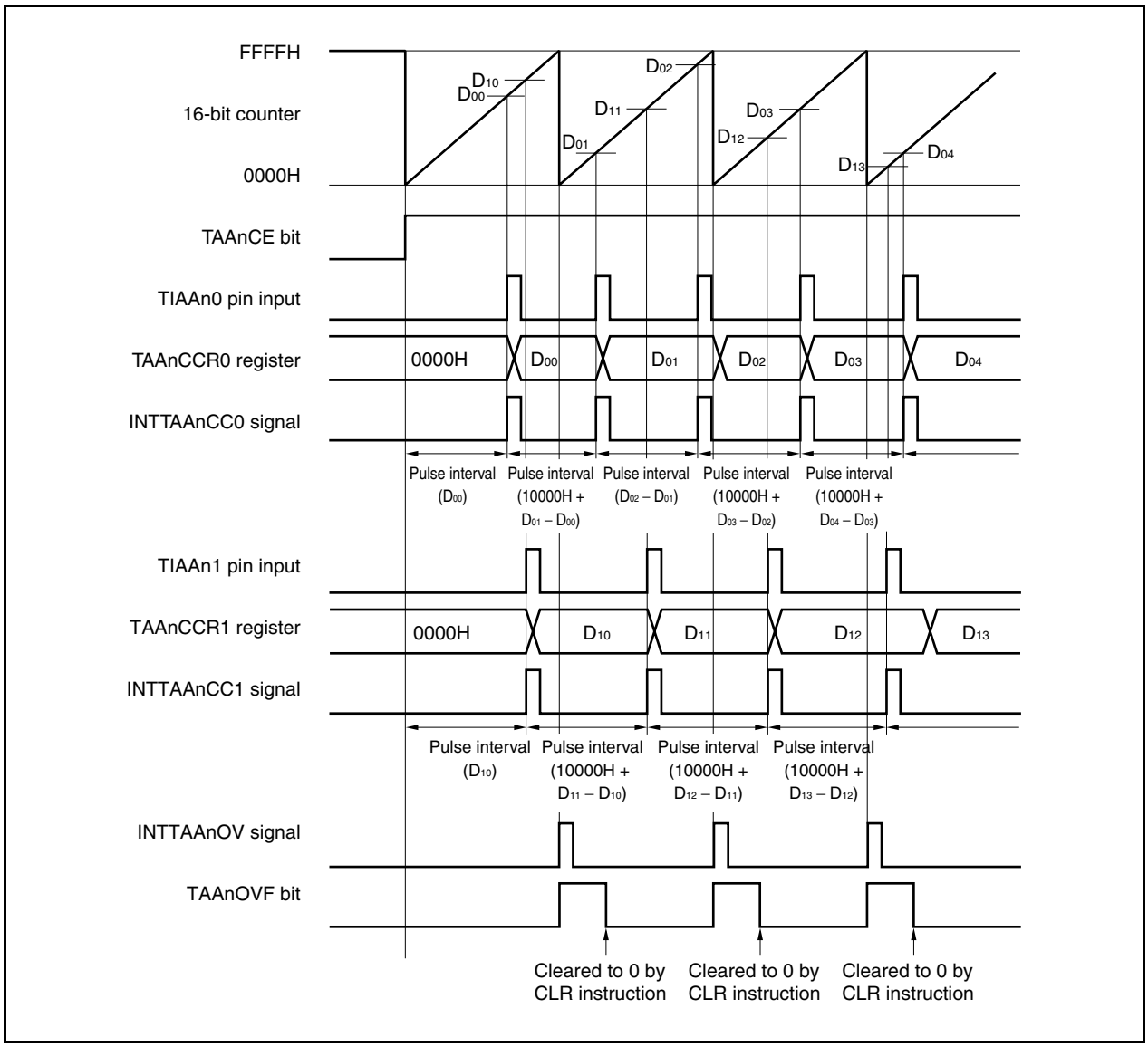
Value set to compare register second and subsequent time: Previous set value + D_m

(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

Remark n = 0 to 4
m = 0, 1

(b) Pulse width measurement with capture register

When pulse width measurement is performed with the TAAnCCRm register used as a capture register, software processing is necessary for reading the capture register each time the INTTAAnCCm signal has been detected and for calculating an interval.



When executing pulse width measurement in the free-running timer mode, two pulse widths can be measured with one channel.

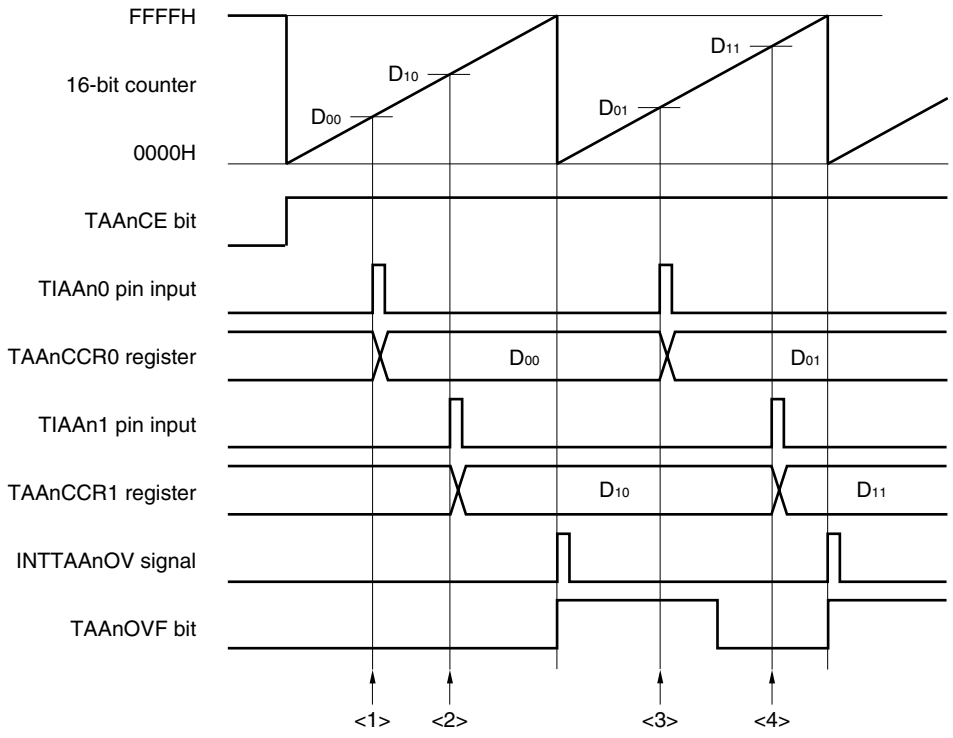
To measure a pulse width, the pulse width can be calculated by reading the value of the TAAnCCRm register in synchronization with the INTTAAnCCm signal, and calculating the difference between the read value and the previously read value.

Remark n = 0 to 4
m = 0, 1

(c) Processing of overflow when two capture registers are used

Care must be exercised in processing the overflow flag when two capture registers are used. First, an example of incorrect processing is shown below.

Example of incorrect processing when two capture registers are used

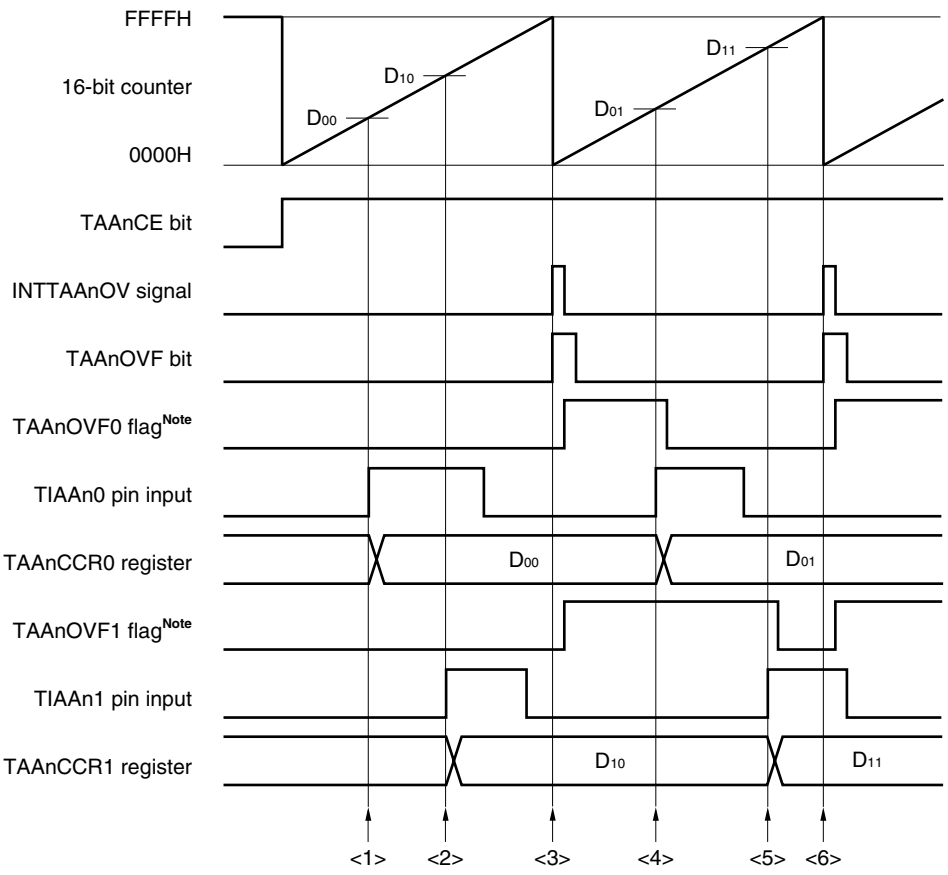


The following problem may occur when two pulse widths are measured in the free-running timer mode.

<1> Read the TAAnCCR0 register (setting of the default value of the TIAAn0 pin input).
<2> Read the TAAnCCR1 register (setting of the default value of the TIAAn1 pin input).
<3> Read the TAAnCCR0 register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).
<4> Read the TAAnCCR1 register.
Read the overflow flag. Because the flag is cleared in <3>, 0 is read.
Because the overflow flag is 0, the pulse width can be calculated by (D_{11} – D_{10}) (incorrect).

When two capture registers are used, and if the overflow flag is cleared to 0 by one capture register, the other capture register may not obtain the correct pulse width.
Use software when using two capture registers. An example of how to use software is shown below.

Example when two capture registers are used (using overflow interrupt)



Note The TAAnOVF0 and TAAnOVF1 flags are set on the internal RAM by software.

<1> Read the TAAnCCR0 register (setting of the default value of the TIAAn0 pin input).

<2> Read the TAAnCCR1 register (setting of the default value of the TIAAn1 pin input).

<3> An overflow occurs. Set the TAAnOVF0 and TAAnOVF1 flags to 1 in the overflow interrupt servicing, and clear the overflow flag to 0.

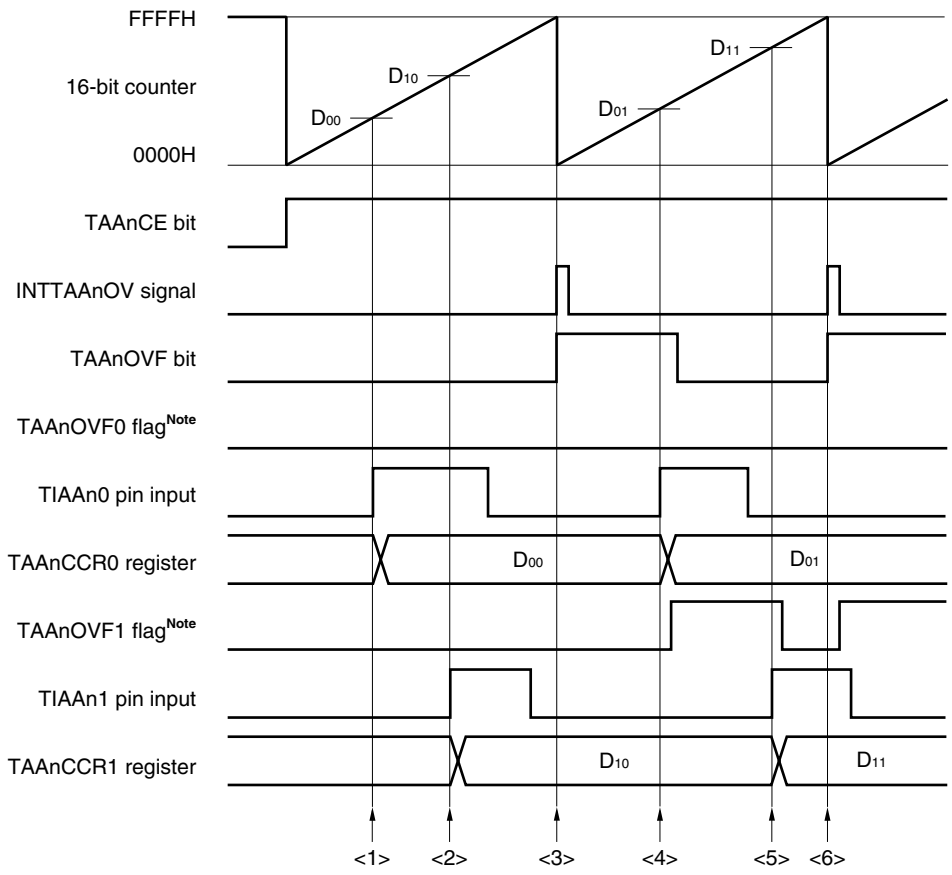
<4> Read the TAAnCCR0 register.
Read the TAAnOVF0 flag. If the TAAnOVF0 flag is 1, clear it to 0.
Because the TAAnOVF0 flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).

<5> Read the TAAnCCR1 register.
Read the TAAnOVF1 flag. If the TAAnOVF1 flag is 1, clear it to 0 (the TAAnOVF0 flag is cleared in <4>, and the TAAnOVF1 flag remains 1).
Because the TAAnOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} – D_{10}) (correct).

<6> Same as <3>

(2/2)

Example when two capture registers are used (without using overflow interrupt)



Note The TAAnOVF0 and TAAnOVF1 flags are set on the internal RAM by software.

<1> Read the TAAnCCR0 register (setting of the default value of the TIAAn0 pin input).

<2> Read the TAAnCCR1 register (setting of the default value of the TIAAn1 pin input).

<3> An overflow occurs. Nothing is done by software.

<4> Read the TAAnCCR0 register.

Read the overflow flag. If the overflow flag is 1, set only the TAAnOVF1 flag to 1, and clear the overflow flag to 0.

Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).

<5> Read the TAAnCCR1 register.

Read the overflow flag. Because the overflow flag is cleared in <4>, 0 is read.

Read the TAAnOVF1 flag. If the TAAnOVF1 flag is 1, clear it to 0.

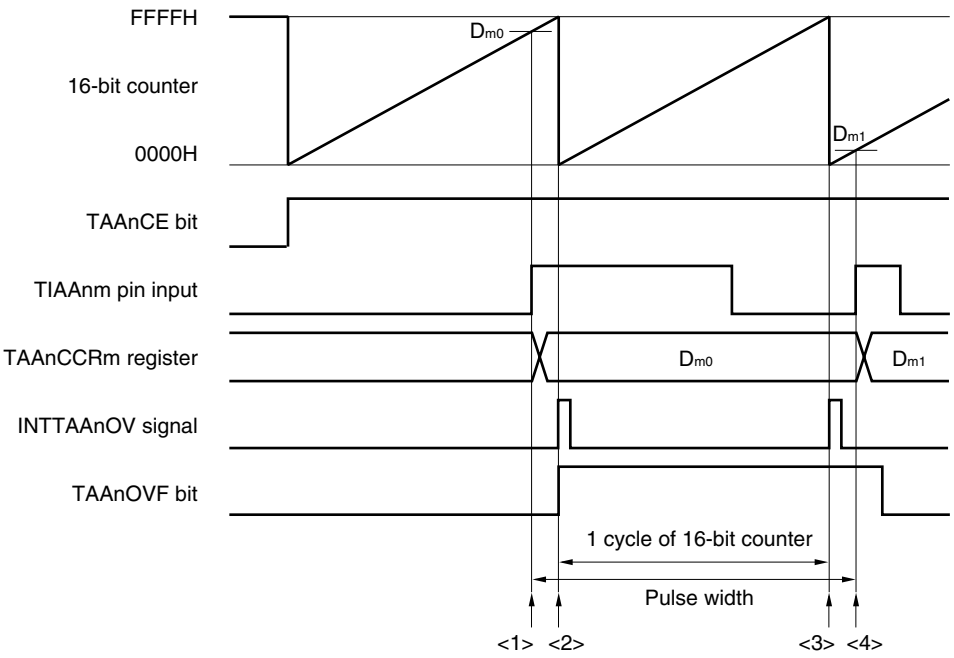
Because the TAAnOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} – D_{10}) (correct).

<6> Same as <3>

(d) Processing of overflow if capture trigger interval is long

If the pulse width is greater than one cycle of the 16-bit counter, care must be exercised because an overflow may occur more than once from the first capture trigger to the next. First, an example of incorrect processing is shown below.

Example of incorrect processing when capture trigger interval is long



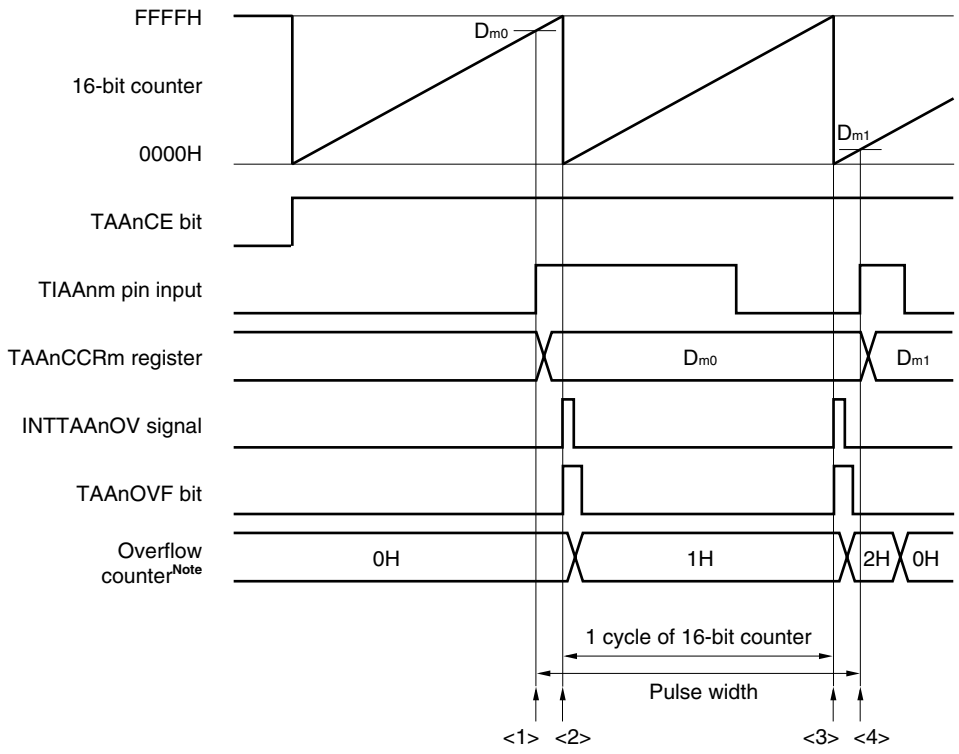
The following problem may occur when long pulse width is measured in the free-running timer mode.

<1> Read the TAAnCCRm register (setting of the default value of the TIAAnm pin input).
<2> An overflow occurs. Nothing is done by software.
<3> An overflow occurs a second time. Nothing is done by software.
<4> Read the TAAnCCRm register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{m1} – D_{m0}) (incorrect).
Actually, the pulse width must be (20000H + D_{m1} – D_{m0}) because an overflow occurs twice.

If an overflow occurs twice or more when the capture trigger interval is long, the correct pulse width may not be obtained.

If the capture trigger interval is long, slow the count clock to lengthen one cycle of the 16-bit counter, or use software. An example of how to use software is shown next.

Example when capture trigger interval is long



Note The overflow counter is set arbitrarily by software on the internal RAM.

<1> Read the TAAnCCRm register (setting of the default value of the TIAAnm pin input).

<2> An overflow occurs. Increment the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.

<3> An overflow occurs a second time. Increment (+1) the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.

<4> Read the TAAnCCRm register.

Read the overflow counter.

→ When the overflow counter is “N”, the pulse width can be calculated by ($N \times 10000H + D_{m1} - D_{m0}$).

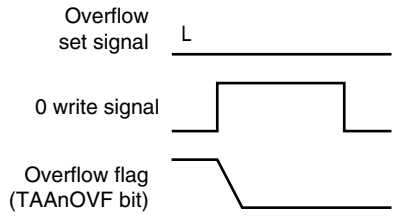
In this example, the pulse width is ($20000H + D_{m1} - D_{m0}$) because an overflow occurs twice.

Clear the overflow counter (0H).

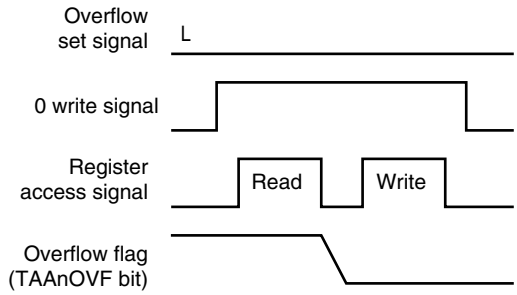
(e) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TAAnOVF bit to 0 with the CLR instruction and by writing 8-bit data (bit 0 is 0) to the TAAnOPT0 register. To accurately detect an overflow, read the TAAnOVF bit when it is 1, and then clear the overflow flag by using a bit manipulation instruction.

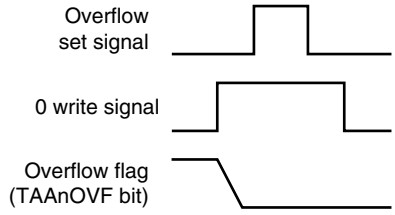
(i) Operation to write 0 (without conflict with setting)



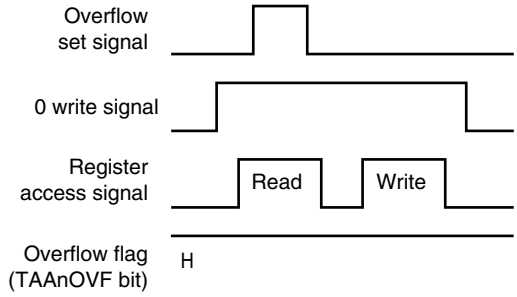
(iii) Operation to clear to 0 (without conflict with setting)



(ii) Operation to write 0 (conflict with setting)



(iv) Operation to clear to 0 (conflict with setting)



Remark n = 0 to 4

To clear the overflow flag to 0, read the overflow flag to check if it is set to 1, and clear it with the CLR instruction. If 0 is written to the overflow flag without checking if the flag is 1, the set information of overflow may be erased by writing 0 ((ii) in the above chart). Therefore, software may judge that no overflow has occurred even when an overflow actually has occurred.

If execution of the CLR instruction conflicts with occurrence of an overflow when the overflow flag is cleared to 0 with the CLR instruction, the overflow flag remains set even after execution of the clear instruction.

7.5.7 Pulse width measurement mode (TAAnMD2 to TAAnMD0 bits = 110)

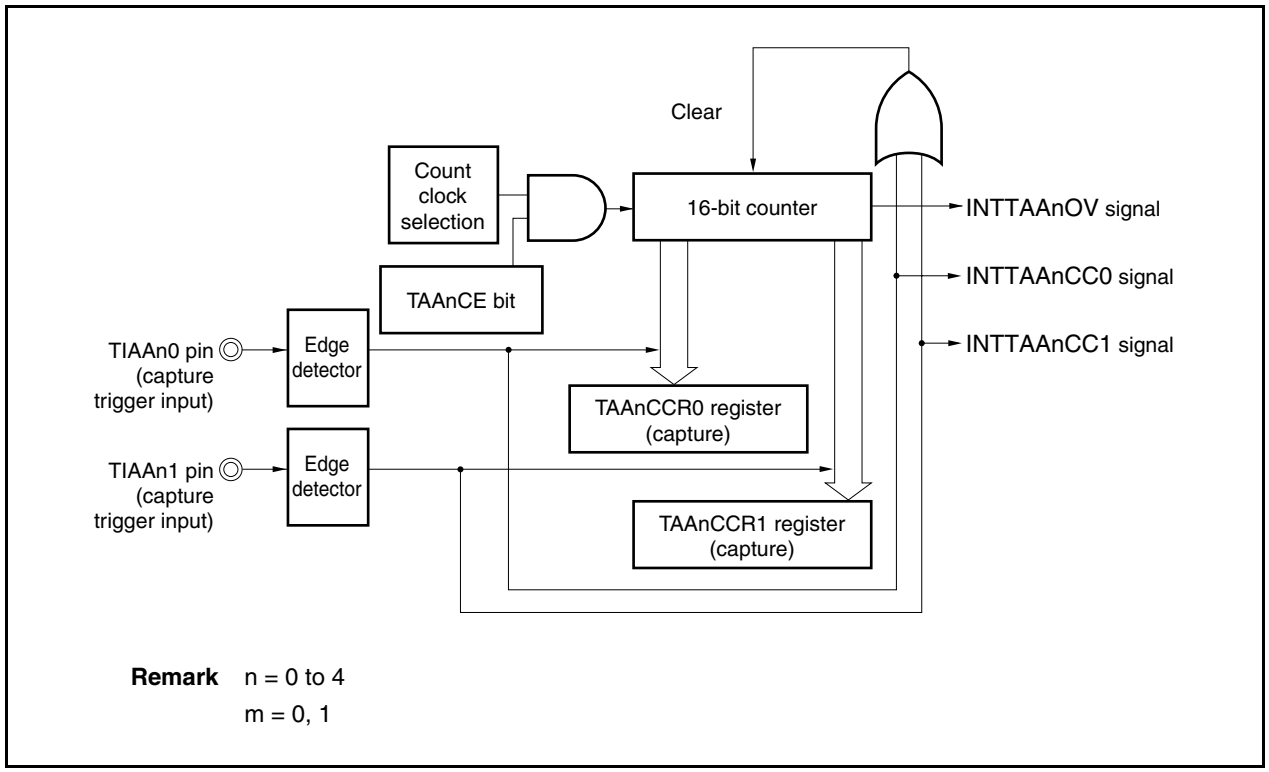
In the pulse width measurement mode, 16-bit timer/event counter AA starts counting when the TAAnCTL0.TAAnCE bit is set to 1. Each time the valid edge input to the TIAAnm pin has been detected, the count value of the 16-bit counter is stored in the TAAnCCRm register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TAAnCCRm register after a capture interrupt request signal (INTTAAnCCm) occurs.

Select either the TIAAn0 or TIAAn1 pin as the capture trigger input pin. Specify “No edge detected” by using the TAAnIOC1 register for the unused pins.

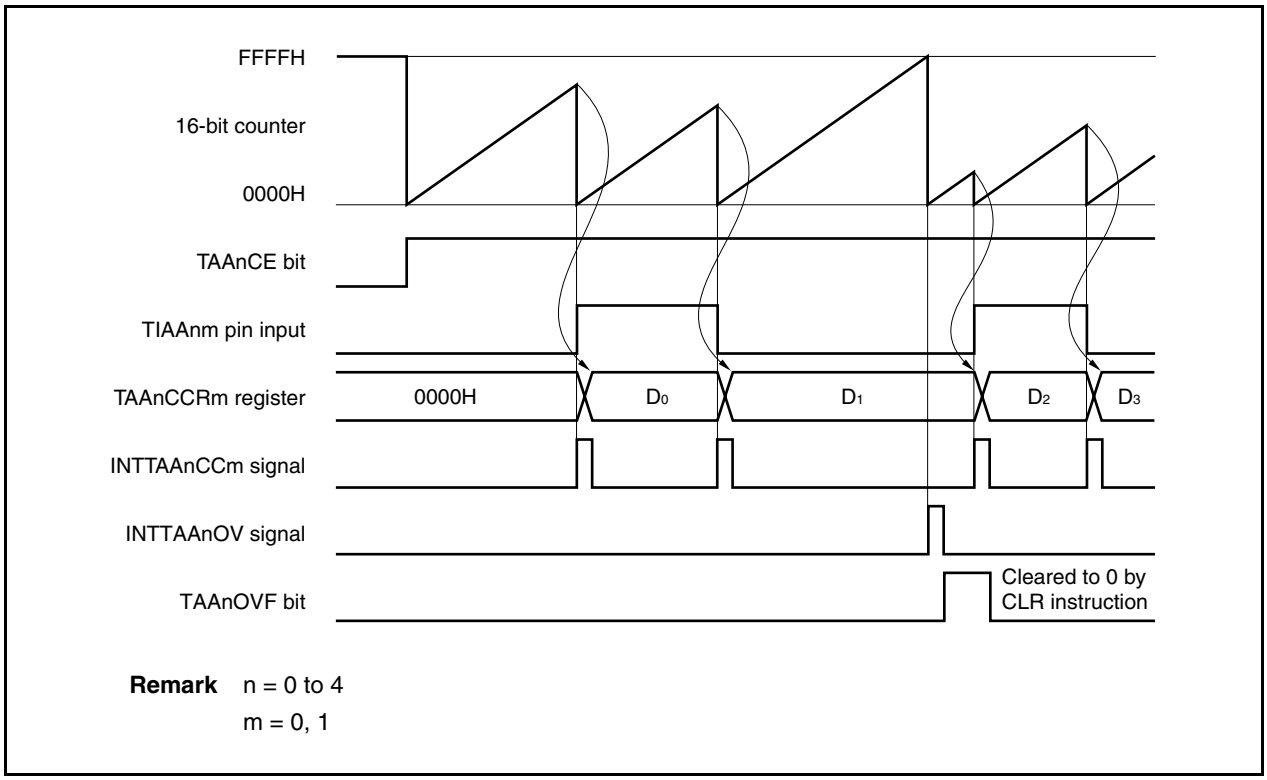
When an external clock is used as the count clock, measure the pulse width of the TIAAn1 pin because the external clock is fixed to the TIAAn0 pin. At this time, clear the TAAnIOC1.TAAnIS1 and TAAnIOC1.TAAnIS0 bits to 00 (capture trigger input (TIAAn0 pin): No edge detected).

Figure 7-34. Configuration in Pulse Width Measurement Mode



Remark n = 0 to 4
m = 0, 1

Figure 7-35. Basic Timing in Pulse Width Measurement Mode



Remark n = 0 to 4
m = 0, 1

When the TAAnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIAAnm pin is later detected, the count value of the 16-bit counter is stored in the TAAnCCRm register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTTAAnCCm) is generated.

The pulse width is calculated as follows.

Pulse width = Captured value × Count clock cycle

If the valid edge is not input to the TIAAnm pin even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTAAnOV) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TAAnOPT0.TAAnOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction via software.

If the overflow flag is set to 1, the pulse width can be calculated as follows.

Pulse width = (10000H × TAAnOVF bit set (1) count + Captured value) × Count clock cycle

Remark n = 0 to 4
m = 0, 1

Figure 7-36. Register Setting in Pulse Width Measurement Mode (1/2)

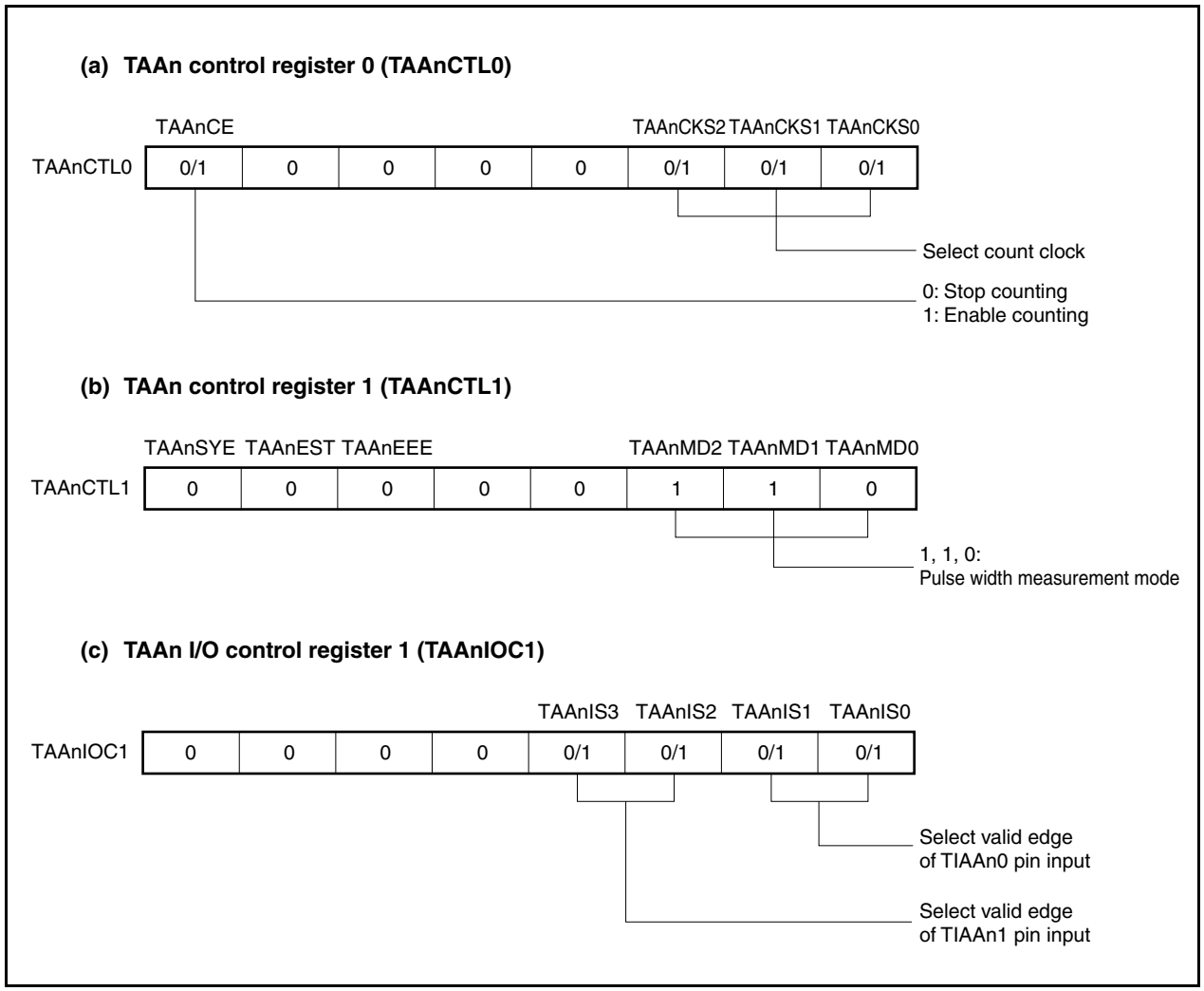


Figure 7-36. Register Setting in Pulse Width Measurement Mode (2/2)

(d) TAAn option register 0 (TAAnOPT0)

	TAAnCCS1	TAAnCCS0					TAAnOVF
TAAnOPT0	0	0	0	0	0	0	0/1

Overflow flag

(e) TAAn counter read buffer register (TAAnCNT)

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

(f) TAAn capture/compare registers 0 and 1 (TAAnCCR0 and TAAnCCR1)

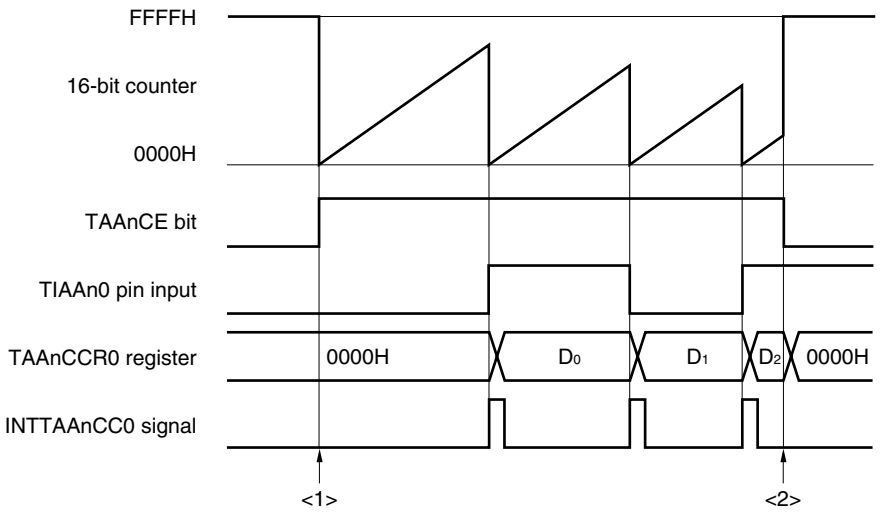
These registers store the count value of the 16-bit counter when the valid edge input to the TIAAnm pin is detected.

<R> **Remarks 1.** TAAn I/O control register 0 (TAAnIOC0) and TAAn I/O control register 2 (TAAnIOC2) are not used in the pulse width measurement mode.

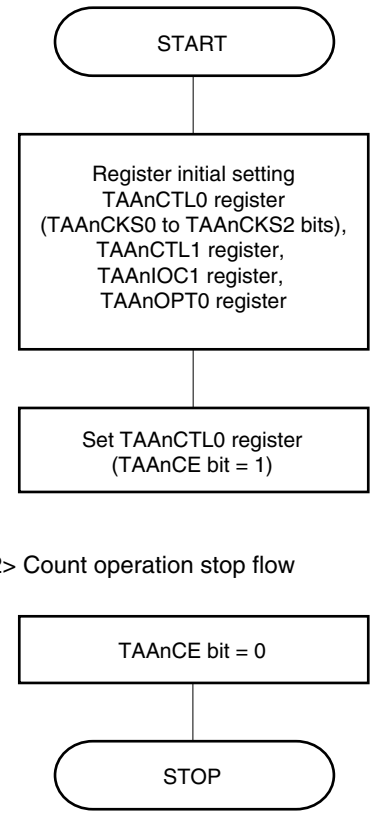
2. n = 0 to 4
m = 0, 1

(1) Operation flow in pulse width measurement mode

Figure 7-37. Software Processing Flow in Pulse Width Measurement Mode



<1> Count operation start flow



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

The TAAnCKS0 to TAAnCKS2 bits can be set at the same time when counting has been started (TAAnCE bit = 1).

<2> Count operation stop flow

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

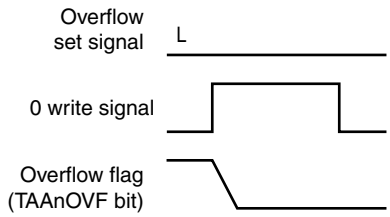
Remark n = 0 to 4

(2) Operation timing in pulse width measurement mode

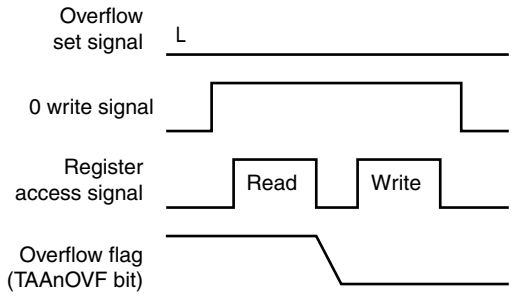
(a) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TAAnOVF bit to 0 with the CLR instruction and by writing 8-bit data (bit 0 is 0) to the TAAnOPT0 register. To accurately detect an overflow, read the TAAnOVF bit when it is 1, and then clear the overflow flag by using a bit manipulation instruction.

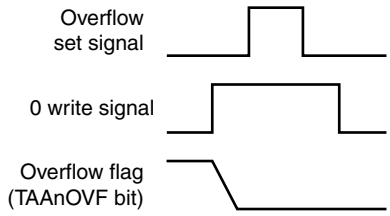
(i) Operation to write 0 (without conflict with setting)



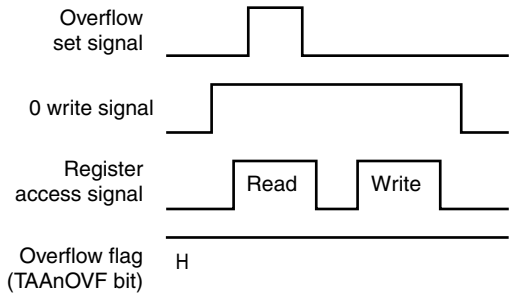
(iii) Operation to clear to 0 (without conflict with setting)



(ii) Operation to write 0 (conflict with setting)



(iv) Operation to clear to 0 (conflict with setting)



Remark n = 0 to 4

To clear the overflow flag to 0, read the overflow flag to check if it is set to 1, and clear it with the CLR instruction. If 0 is written to the overflow flag without checking if the flag is 1, the set information of overflow may be erased by writing 0 ((ii) in the above chart). Therefore, software may judge that no overflow has occurred even when an overflow actually has occurred.

If execution of the CLR instruction conflicts with occurrence of an overflow when the overflow flag is cleared to 0 with the CLR instruction, the overflow flag remains set even after execution of the clear instruction.

7.5.8 Timer output operations

The following table shows the operations and output levels of the TOAAn0 and TOAAn1 pins.

Table 7-4. Timer Output Control in Each Mode

Operation Mode	TOAAn1 Pin	TOAAn0 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	–	

Remark n = 0 to 4

Table 7-5. Truth Table of TOAAn0 and TOAAn1 Pins Under Control of Timer Output Control Bits

TAAnIOC0.TAAnOLm Bit	TAAnIOC0.TAAnOEm Bit	TAAnCTL0.TAAnCE Bit	Level of TOAAnm Pin
0	0	×	Low-level output
	1	0	Low-level output
		1	Low level immediately before counting, high level after counting is started
1	0	×	High-level output
	1	0	High-level output
		1	High level immediately before counting, low level after counting is started

Remark n = 0 to 4
m = 0, 1

7.6 Timer-Tuned Operation Function

Timer AA and timer AB have a timer-tuned operation function.

The timer-tuned operation function is used to tune the internal timers of the V850ES/Hx3, so that the number of capture or compare registers of the slave timer (the number of timer outputs and the number of compare match interrupts of the slave timer) can be added to the master timer. The timers that can be tuned are listed in Table 7-6.

Table 7-6. Tuned Operation Mode of Timers

Master Timer	Slave Timer
TAA0	TAA1
TAA2	TAA3
TAB0	TAA4
TAB1	TAB2

The tuned operation function has the following modes.

- PWM output mode
- Free-running timer mode
- Triangular wave PWM mode (when TAB1 is used as the master timer)

Figure 7-38 shows an example where the unit operation and tuned operation are performed in the PWM output mode with TAA0 as the master timer and TAA1 as the slave timer.

Figure 7-38. Differences Between Unit Operation and Tuned Operation Using TAA0 and TAA1

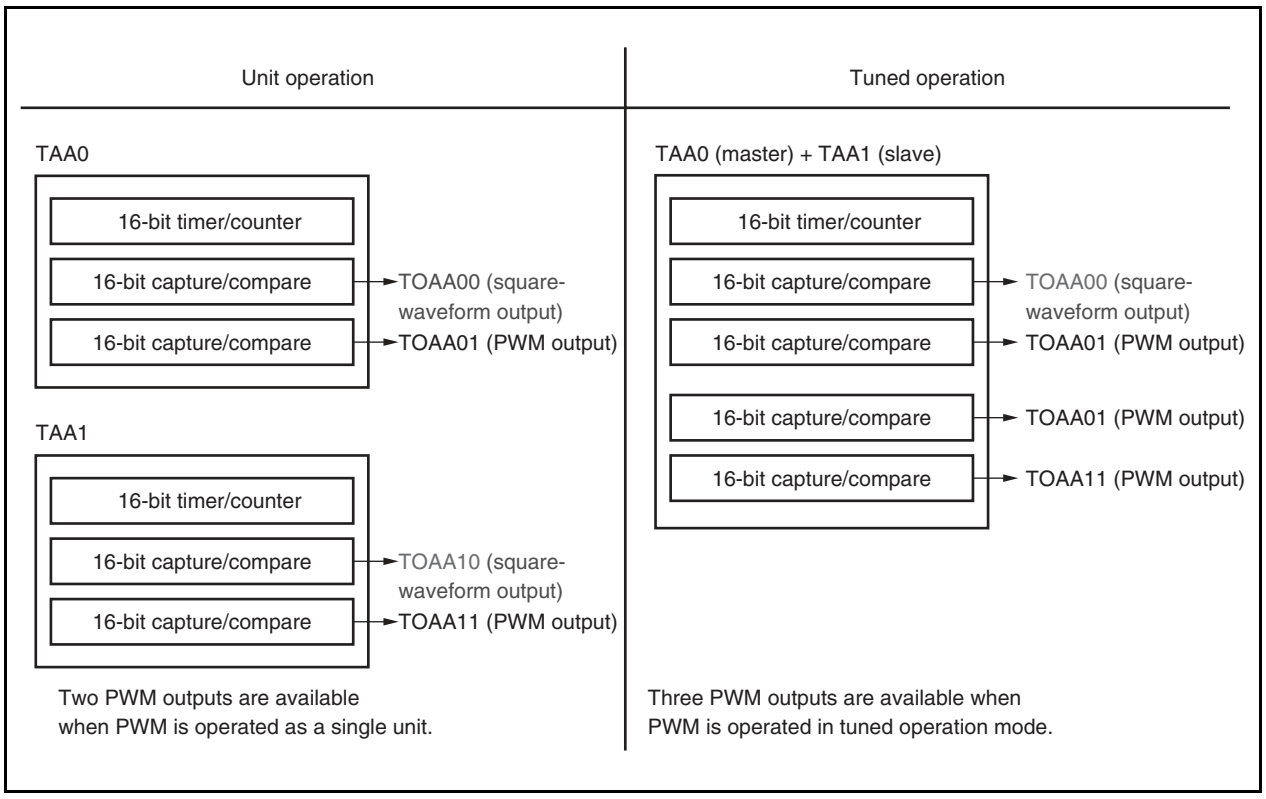


Table 7-7 show the timer modes that can be used in the tuned operation mode and Table 7-8 shows the differences of timer output function between unit operation and tuned operation (√: Settable, ×: Not settable).

Table 7-7. Timer Modes Usable in Tuned Operation Mode

Master Timer	Slave Timer	Free-Running Timer Mode	PWM Mode	Triangular Wave PWM Mode
TAA0	TAA1	√	√	×
TAA2	TAA3	√	√	×
TAB0	TAA4	√	√	×
TAB1	TAB2	√	√	√

Table 7-8. Timer Output Functions

Tuned Channel	Timer	Pin	Free-Running Timer Mode		PWM Mode		Triangular Wave PWM Mode	
			Tuning OFF	Tuning ON	Tuning OFF	Tuning ON	Tuning OFF	Tuning ON
Ch0	TAA0 (master)	TOAA00	PPG	←	Toggle	←	N/A	←
		TOAA01	PPG	←	PWM	←	N/A	←
	TAA1 (slave)	TOAA10	PGP	←	Toggle	PWM	N/A	←
		TOAA11	PPG	←	PWM	←	N/A	←
Ch1	TAA2 (master)	TOAA20	PPG	←	Toggle	←	N/A	←
		TOAA21	PPG	←	PWM	←	N/A	←
	TAA3 (slave)	TOAA30	PPG	←	Toggle	PWM	N/A	←
		TOAA31	PPG	←	PWM	←	N/A	←
Ch2	TAB0 (master)	TOAB00	PPG	←	Toggle	←	Toggle	N/A
		TOAB01 to TOAB03	PPG	←	PWM	←	Triangular wave PWM	N/A
	TAA4 (slave)	TOAA40	PPG	←	Toggle	PWM	N/A	←
		TOAA41	PPG	←	PWM	←	N/A	←
Ch3	TAB1 (master)	TOAB10	PPG	←	Toggle	←	Toggle	←
		TOAB11 to TOAB13	PPG	←	PWM	←	Triangular wave PWM	←
	TAB2 (slave)	TOAB20	PPG	←	Toggle	PWM	Toggle	Triangular wave PWM
		TOAB21 to TOAB23	PPG	←	PWM	←	Triangular wave PWM	←

Remark The timing of transmitting data from the compare register of the master timer to the compare register of the slave timer is as follows.

- PPG: CPU write timing
- Toggle, PWM, triangular wave PWM: Timing at which timer counter and compare register match TOAAn0 and TOABm0

7.6.1 Free-running timer mode (during timer-tuned operation)

This section explains the free-running timer mode of the timer-tuned operation. For the combination of timer-tuned operations, see Table 7-6. In this section, an example of timer-tuned operation using TAA0 and TAA1 is shown.

(i) Selecting capture/compare registers

When the free-running timer mode of the timer-tuned operation is used with TAA0 and TAA1 connected to each other, the two capture/compare registers of TAA0 and two capture/compare registers of TAA1 can be used in combination.

How the capture and compare registers are combined is not restricted and can be selected by using the TAAnCCSn bit of the master or slave timer. When the compare register is selected, the set value of the compare register can be rewritten during operation (n = 0, 1) and the rewriting method is the anytime write.

(ii) Overflow

If the counter overflows, an overflow interrupt (INTTAA0OV) of the master timer is generated and the overflow flag (TAA0OVF) is set to “1”.

The overflow interrupt (INTTAA1OV) and overflow flag (TAA1OVF) of the slave timer do not operate and are always at the low level.

(1) Setting in free-running timer mode (compare function)

[Initial setting]

Master timer: TAA0CTL0.TAA0CE = 0 (operation disabled)
Slave timer: TAA1CTL0.TAA1CE = 0 (operation disabled)

[Initial setting of master timer (TAA0)]

- TAA0CTL1.TAA0MD2 to TAA0CTL1.TAA0MD0 = 101 (setting of free-running timer mode)
- TAA0OPT0.TAA0CCS1 and TAA0OPT0.TAA0CCS0 = 00 (setting of capture/compare select bit to "compare".)
- TAA0CTL1.TAA0CKS2 to TAA0CTL1.TAA0CKS0 (setting of count clock (any))
- TAA0CCR1 and TAA0CCR0 registers are set.

[Initial setting of slave timer (TAA1)]

- TAA1CTL1.TAA1SYE = 1 (setting of timer-tuned operation)
- TAA1CTL1.TAA1MD2 to TAA1CTL1.TAA1MD0 = 101 (setting of free-running timer mode)
- TAA1OPT0.TAA1CCS1 and TAA1OPT0.TAA1CCS0 = 00 (setting of capture/compare select bit to "compare".)
- TAA1CCR0 and TAA1CCR1 registers are set.

Remark Initial setting of the master timer and slave timer may be performed in any order.

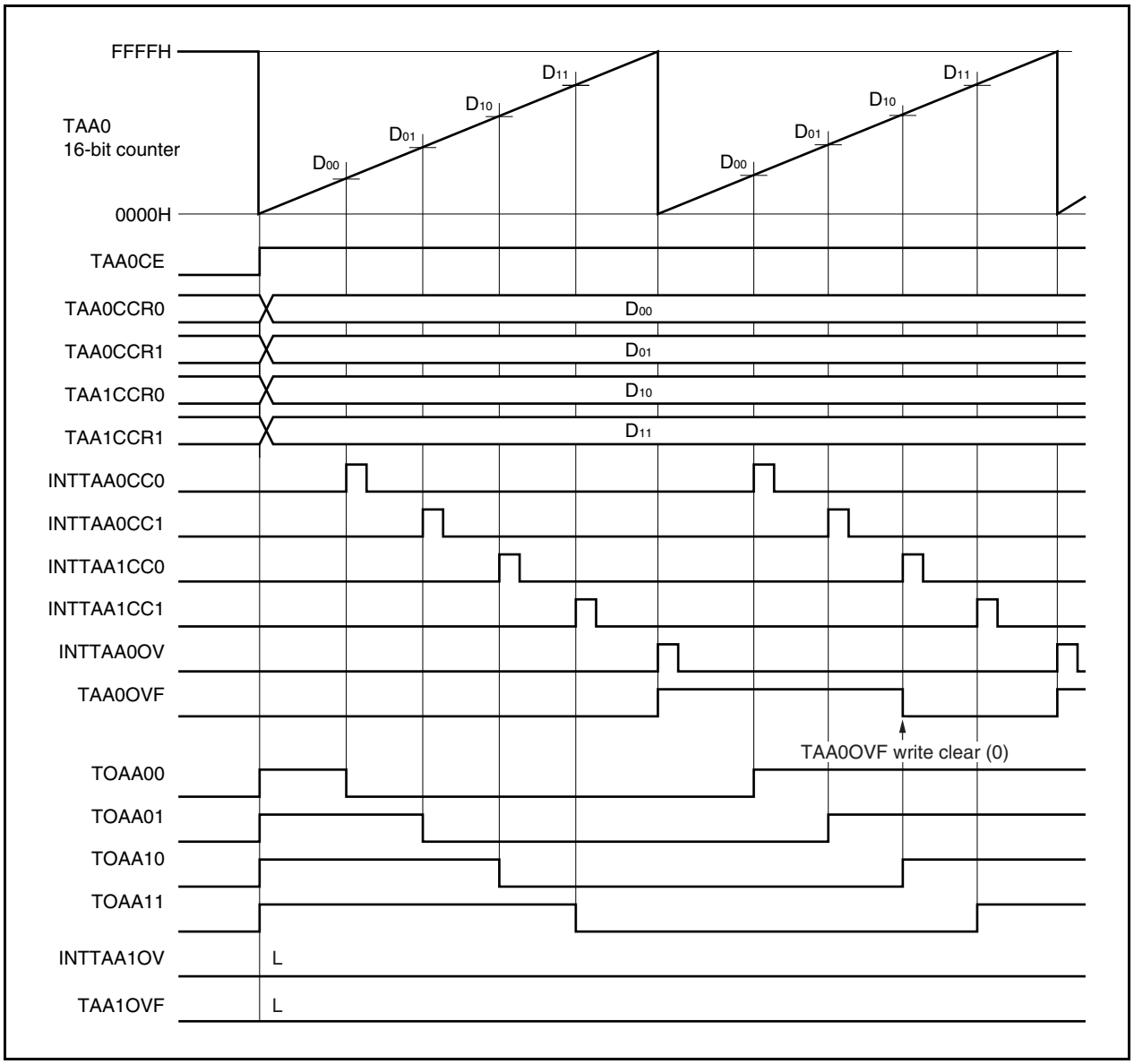
[Starting counting]

<1> Set TAA0CTL0.TAA0CE of the master timer to 1.
<2> Start counting.
<3> Changing the setting of the register during operation
  - The compare register can be rewritten (anytime write).

[End condition]

- Set TAA0CTL0.TAA0CE of the master timer to 0.

Figure 7-39. Example of Timing in Free-Running Mode (Compare Function)



(2) Setting in free-running timer mode (capture function)

[Initial setting]

Master timer: TAA0CTL0.TAA0CE = 0 (operation disabled)
Slave timer: TAA1CTL0.TAA1CE = 0 (operation disabled)

[Initial setting of master timer (TAA0)]

- TAA0CTL1.TAA0MD2 to TAA0CTL1.TAA0MD0 = 101 (setting of free-running timer mode)
- TAA0OPT0.TAA0CCS1 and TAA0OPT0.TAA0CCS0 = 11 (setting of capture/compare select bit to "capture".)
- TAA0CTL1.TAA0CKS2 to TAA0CTL1.TAA0CKS0 (setting of count clock (any))
- TAA0.TAA0IS3 to TAA0.TAA0IS0 (specification of valid edge of capture trigger)

[Initial setting of slave timer (TAA1)]

- TAA1CTL1.TAA1SYE = 1 (setting of timer-tuned operation)
- TAA1CTL1.TAA1MD2 to TAA1CTL1.TAA1MD0 = 101 (setting of free-running timer mode)
- TAA1OPT0.TAA1CCS1 and TAA1OPT0.TAA1CCS0 = 11 (setting of capture/compare select bit to "capture".)
- TAA1.TAA1IS3 to TAA1.TAA1IS0 (specification of valid edge of capture trigger)

Remark Initial setting of the master timer and slave timer may be performed in any order.

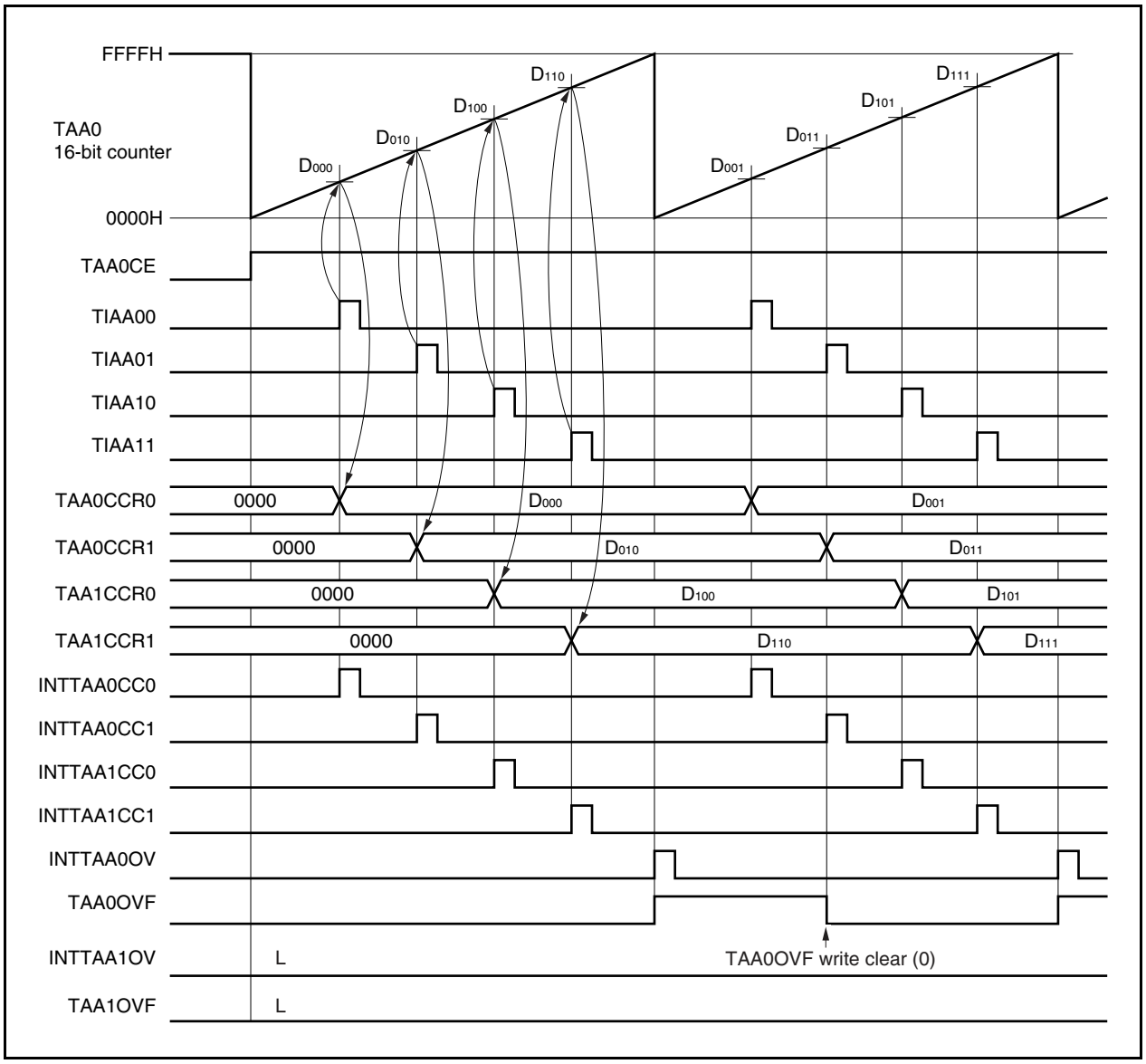
[Starting counting]

<1> Set TAA0CTL0.TAA0CE of the master timer to 1.
<2> Start counting.

[End condition]

- Set TAA0CTL0.TAA0CE of the master timer to 0.

Figure 7-40. Example of Timing in Free-Running Mode (Capture Function)



(3) Setting in free-running timer mode (capture/compare used together)

An example of using TAA0 as a capture register and TAA1 as a compare register is shown below.

[Initial setting]

Master timer: TAA0CTL0.TAA0CE = 0 (operation disabled)

Slave timer: TAA1CTL0.TAA1CE = 0 (operation disabled)

[Initial setting of master timer (TAA0)]

- TAA0CTL1.TAA0MD2 to TAA0CTL1.TAA0MD0 = 101 (setting of free-running timer mode)
- TAA0OPT0.TAA0CCS1 and TAA0OPT0.TAA0CCS0 = 11 (setting of capture/compare select bit to “capture”.)
- TAA0CTL1.TAA0CKS2 to TAA0CTL1.TAA0CKS0 (setting of count clock (any))
- TAA0.TAA0IS3 to TAA0.TAA0IS0 (specification of valid edge of capture trigger)

[Initial setting of slave timer (TAA1)]

- TAA1CTL1.TAA1SYE = 1 (setting of timer-tuned operation)
- TAA1CTL1.TAA1MD2 to TAA1CTL1.TAA1MD0 = 101 (setting of free-running timer mode)
- TAA1OPT0.TAA1CCS1 and TAA1OPT0.TAA1CCS0 = 00 (setting of capture/compare select bit to “compare”.)
- TAA1CCR0 and TAA1CCR1 registers are set.

Remark Initial setting of the master timer and slave timer may be performed in any order.

[Starting counting]

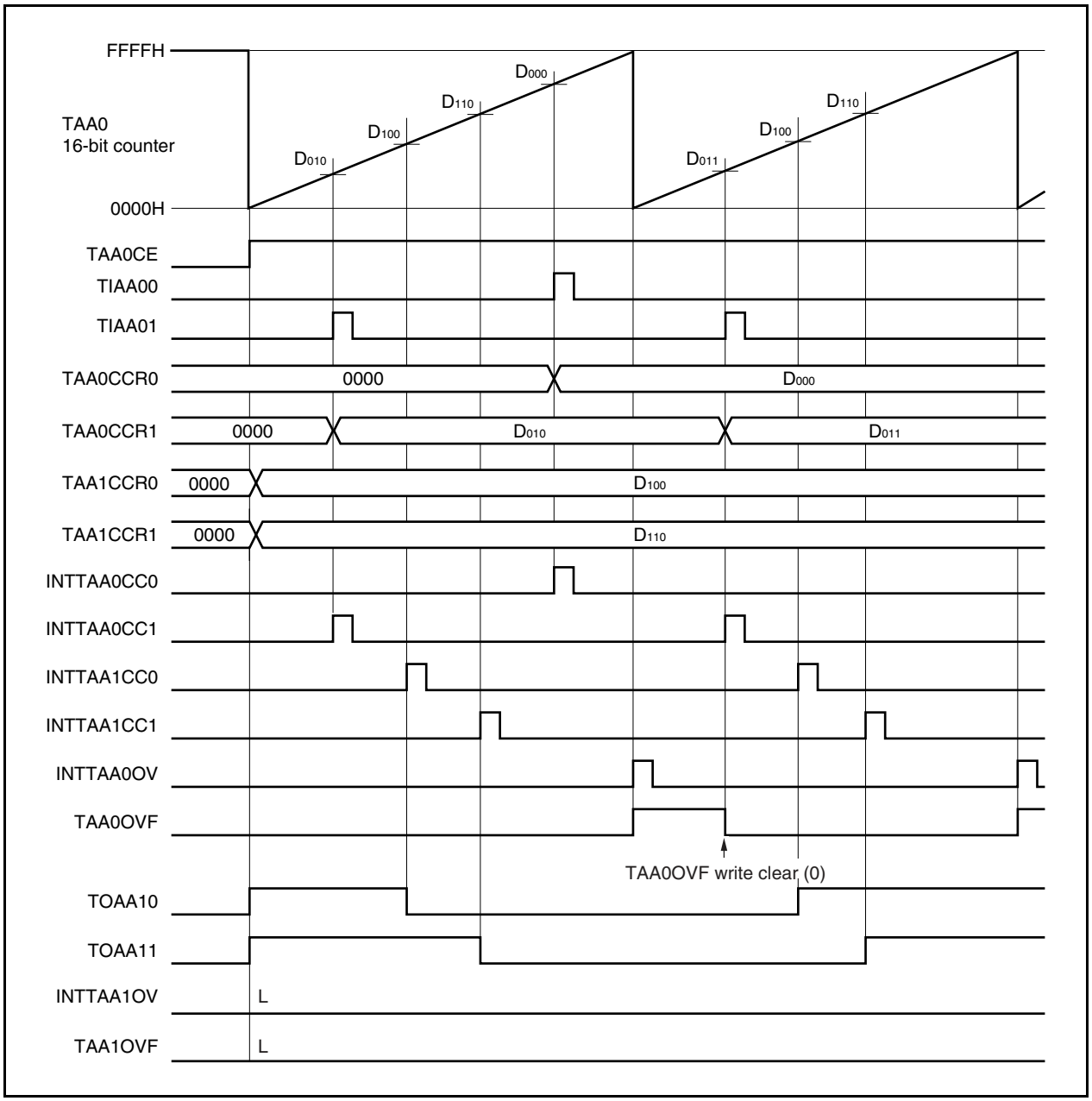
<1> Set TAA0CTL0.TAA0CE of the master timer to 1.

<2> Start counting.

[End condition]

- Set TAA0CTL0.TAA0CE of the master timer to 0.

Figure 7-41. Example of Timing in Free-Running Mode (Capture/Compare Used Together)



7.6.2 PWM output mode (during timer-tuned operation)

This section explains the PWM output mode of timer-tuned operation. For combinations of timer-tuned operations, see Table 7-6. This section presents an example of a timer-tuned operation with TAB0 and TAA4.

The TAB0CCR0 register of the master timer (TAB0) is used as a compare register for cycle, and the TAB0CCR1, TAB0CCR2, and TAB0CCR3 registers of the master timer (TAB0) and the TAA4CCR0 and TAA4CCR1 registers of the slave timer (TAA4) are used as compare registers for duty.

The compare registers can be rewritten during operation and the rewriting method is batch writing.

Batch writing is enabled when the TAB0CCR1 register of the master timer (TAB0) is written, and all the compare registers of the master and slave timers are rewritten or the same value is written to them when an interrupt, which is generated if the value of the TAB0CCR0 register of the master timer (TAB0) matches the value of the timer counter, is generated.

(1) Setting in PWM output mode

[Initial setting]

Master timer: TAB0CTL0.TAB0CE = 0 (operation disabled)
Slave timer: TAA4CTL0.TAA4CE = 0 (operation disabled)

[Initial setting of master timer (TAB0)]

- TAB0CTL1.TAB0MD2 to TAB0CTL1.TAB0MD0 = 100 (setting of PWM output mode)
- TAB0OPT0.TAB0CCS3 to TAB0OPT0.TAB0CCS0 = 0000 (setting of capture/compare select bit to “compare”.)
- TAB0CCR0, TAB0CCR1, TAB0CCR2, and TAB0CCR3 registers are set.

[Initial setting of slave timer (TAA4)]

- TAA4CTL1.TAA4SYE = 1 (setting of timer-tuned operation)
- TAA4CTL1.TAA4MD2 to TAA4CTL1.TAA4MD0 = 101 (setting of free-running timer mode)
- TAA4OPT0.TAA4CCS1 and TAA4OPT0.TAA4CCS0 = 00 (setting of capture/compare select bit to “compare”.)
- TAA4CCR0 and TAA4CCR1 registers are set.

Remark Initial setting of the master timer and slave timer may be performed in any order.

[Starting counting]

<1> Set TAB0CTL0.TAB0CE of the master timer to 1.
<2> Start counting.
<3> Changing the setting of the register during operation
    - The compare register can be rewritten (batch write).

[End condition]

- Set TAB0CTL0.TAB0CE of the master timer to 0.

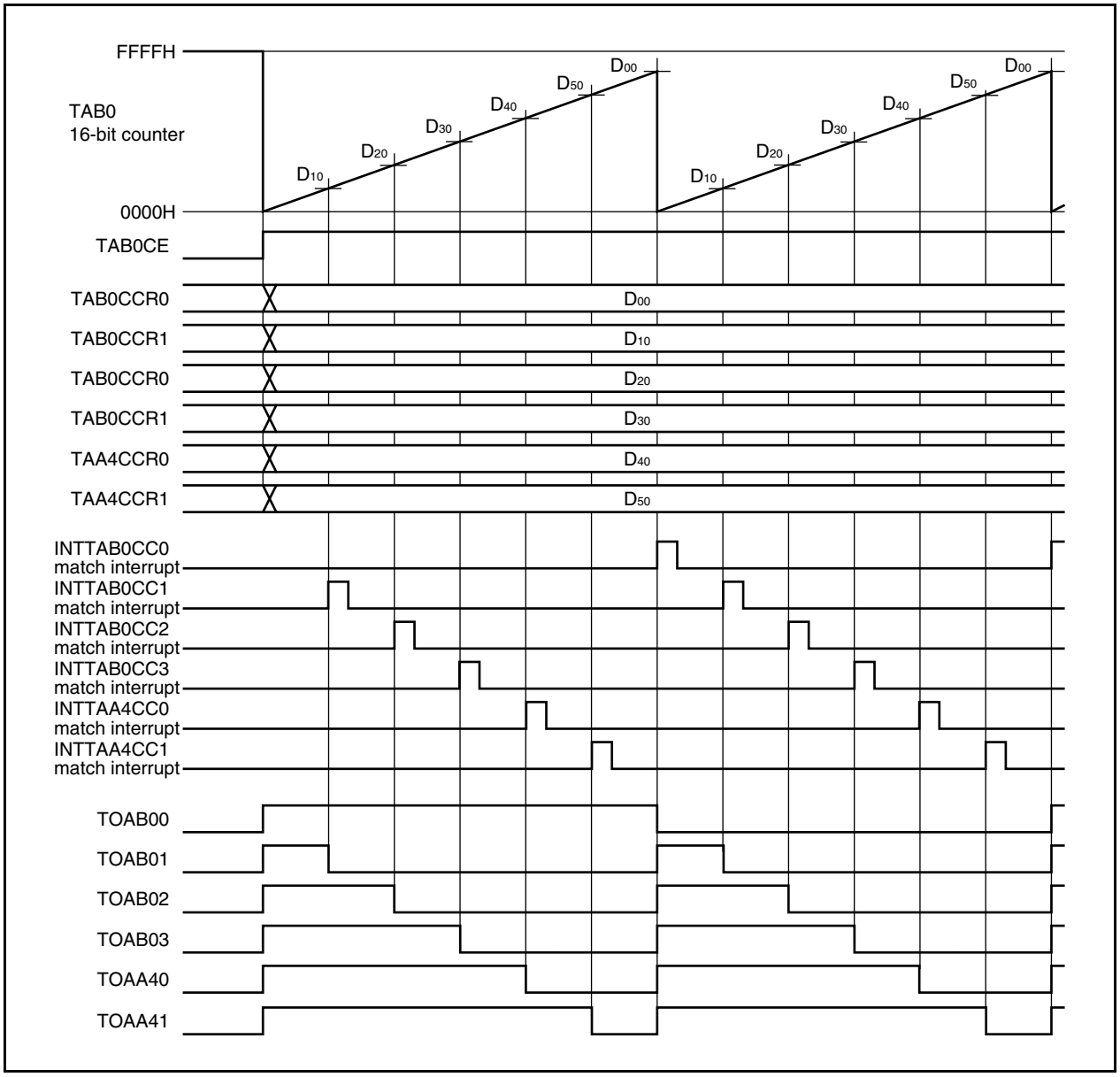
[Batch write]

In the PWM output mode, the next batch write is enabled by writing the TAB0CCR1 register of the master timer (TAB0). After all the compare registers that must be rewritten have been rewritten, therefore, the TAB0CCR1 register of the master timer (TAB0) must be written.

Batch writing is executed when the value of the timer counter matches the value of the compare register for cycle (TAB0CCR0).

If the TAB0CCR1 register of the master timer (TAB0) is not written, batch writing is not enabled even if any other compare register is rewritten. Consequently, the value of the compare registers is not rewritten even when the value of the timer counter matches the value of the compare register for cycle (TAB0CCR0).

Figure 7-42. Timing Example of Tuned PWM Function (TAB0, TAA4)



7.6.3 Triangular PWM output mode (during timer-tuned operation) (V850ES/HJ3 only)

This section explains the triangular wave PWM mode of timer-tuned operation, taking an example where TAB1 is used as the master timer and TAB2 is used as the slave timer.

The triangular wave PWM output mode can be used during timer-tuned operation only when TAB1 is used as the master timer and TAB2 is used as the slave timer.

The TAB1CCR0 register of the master timer serves as a compare register for cycle in the triangular wave PWM output mode.

The TAB1CCR1, TAB1CCR2, and TAB1CCR3 registers of the master timer and the TAB2CCR0, TAB2CCR1, TAB2CCR2, and TAB2CCR3 registers of the slave timer (TAB2) are used as compare registers for duty.

All the compare registers can be rewritten during operation and the rewriting method is batch writing.

Batch writing is enabled by writing the TAB0CCR1 register of the master timer (TAB1). All the compare registers of the master and slave timers are rewritten or the same value is written to them when the timer counter underflow interrupt (valley interrupt) of the master timer (TAB1) is generated.

(a) Valley interrupt (underflow)

At the valley of the counter (0001H → 0000H), the overflow interrupt (INTTAB1OV) of the master timer is generated as a valley interrupt (underflow).

At this time, the overflow flag (TAB1CTL1.TAB1OVF) of the master timer (TAB1) and overflow interrupt (INTTAB2OV) and overflow flag (TAB2CTL1.TAB2OVF) of the slave timer do not operate and are always at the low level.

[Initial setting]

Master timer: TAB1CTL0.TAB1CE = 0 (operation disabled)

Slave timer: TAB2CTL0.TAB2CE = 0 (operation disabled)

[Initial setting of master timer (TAB1)]

- TAB1CTL1.TAB1MD2 to TAB1CTL1.TAB1MD0 = 111 (setting of triangular wave PWM output mode)
- TAB1OPT0.TAB1CCS3 to TAB1OPT0.TAB1CCS0 = 0000 (setting of capture/compare select bit to “compare”.)
- TAB1CCR0, TAB1CCR1, TAB1CCR2, and TAB1CCR3 registers are set.

[Initial setting of slave timer (TAB2)]

- TAB2CTL1.TAB2SYE = 1 (setting of timer-tuned operation)
- TAB2CTL1.TAB2MD2 to TAB2CTL1.TAB2MD0 = 101 (setting of free-running timer mode)
- TAB2OPT0.TAB2CCS3 to TAB2OPT0.TAB2CCS0 = 0000 (setting of capture/compare select bit to “compare”.)
- TAB2CCR0, TAB2CCR1, TAB2CCR2, and TAB2CCR3 registers are set.

Remark Initial setting of the master timer and slave timer may be performed in any order.

[Starting counting]

<1> Set TAB1CTL0.TAB1CE of the master timer to 1.

<2> Start counting.

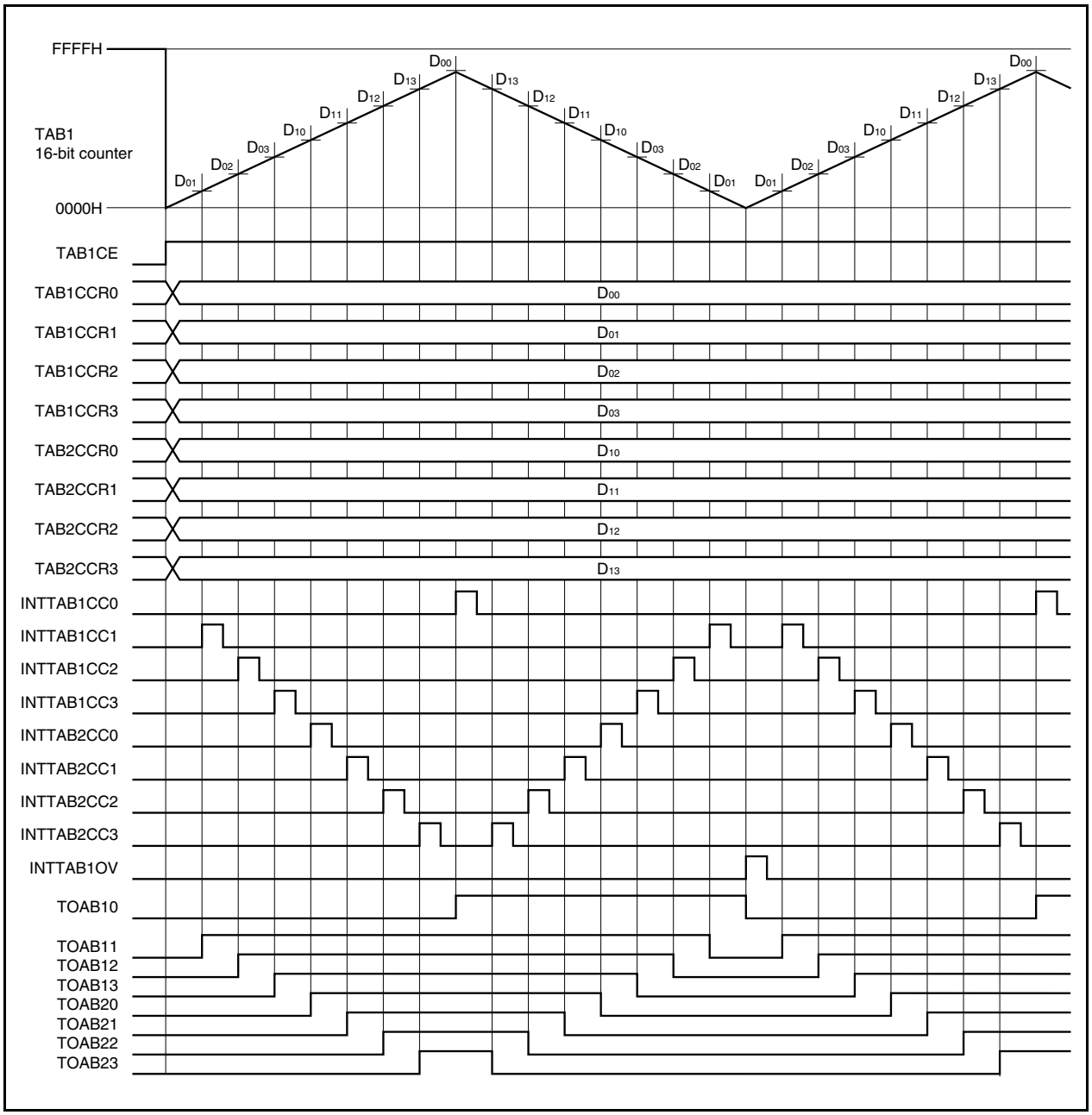
<3> Changing the setting of the register during operation

- The compare register can be rewritten (batch write).

[End condition]

- Set TAB1CTL0.TAB1CE of the master timer to 0.

Figure 7-43. Example of Timing in Triangular Wave PWM Mode (TAB1, TAB2)



7.7 Cascade Connection

This section explains an operation of connecting two channels of TAA in cascade to form a 32-bit capture timer.

For cascade connection, the “free-running timer mode” must be set and all the capture/compare registers must be set as “capture registers (TAA0CCSn = 1)”.

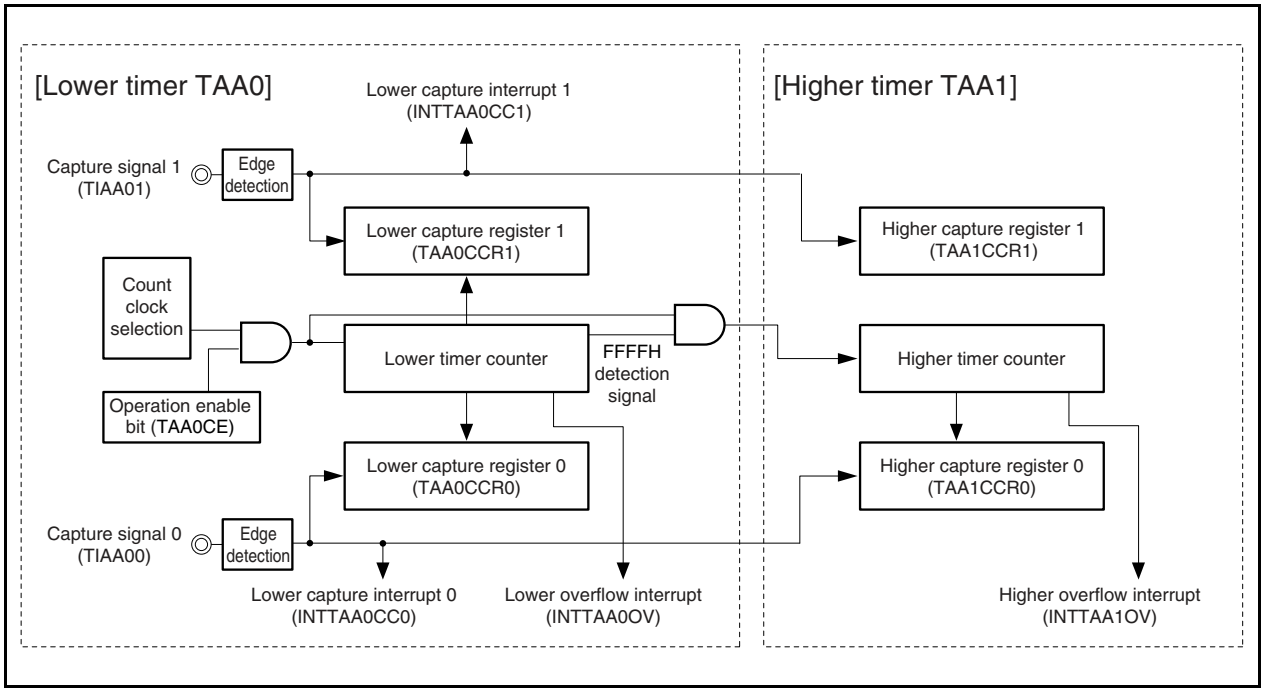
Combinations of TAA channels that can be connected in cascade are shown in the following table.

Table 7-9. Cascade Connection of TAA

Lower Timer (Master Timer)	Higher Timer (Slave Timer)
TAA0	TAA1
TAA2	TAA3

In the following example, TAA0 is used as the lower timer (master timer) and TAA1 is used as the higher timer (slave timer) to create a 32-bit capture timer by cascade connection.

Figure 7-44. Cascade Connection Example



The operation of each pin and signal in cascade connection is shown below.

Table 7-10. Status in Cascade Connection

Name	Higher/Lower	Function	Operation
TIAA00 pin input	Lower	Capture input 0	The value of the lower timer counter is stored in the TAA0CCR0 register and the value of the higher timer counter is stored in the TAA1CCR0 register when the valid edge of this input is detected.
TIAA01 pin input	Lower	Capture input 1	The value of the lower timer counter is stored in the TAA0CCR0 register and the value of the higher timer counter is stored in the TAA1CCR0 register when the valid edge of this input is detected.
INTTAA0CCR0 interrupt signal	Lower	Capture interrupt 0	This interrupt is generated when the valid edge of the TIAA00 pin is detected.
INTTAA0CCR1 interrupt signal	Lower	Capture interrupt 1	This interrupt is generated when the valid edge of the TIAA01 pin is detected.
INTTAA0OV interrupt signal	Lower	Overflow interrupt	This interrupt is generated when an overflow of the lower timer counter is detected.
TIAA10 pin input	Higher	Capture input 0	Does not operate.
TIAA11 pin input	Higher	Capture input 1	Does not operate.
INTTAA1CCR0 interrupt signal	Higher	Capture interrupt 0	Does not operate.
INTTAA1CCR1 interrupt signal	Higher	Capture interrupt 1	Does not operate.
INTTAA1OV interrupt signal	Higher	Overflow interrupt	This interrupt is generated when an overflow of the higher timer counter is detected.

Figure 7-45. Operation Flow in Cascade Connection (1/2)

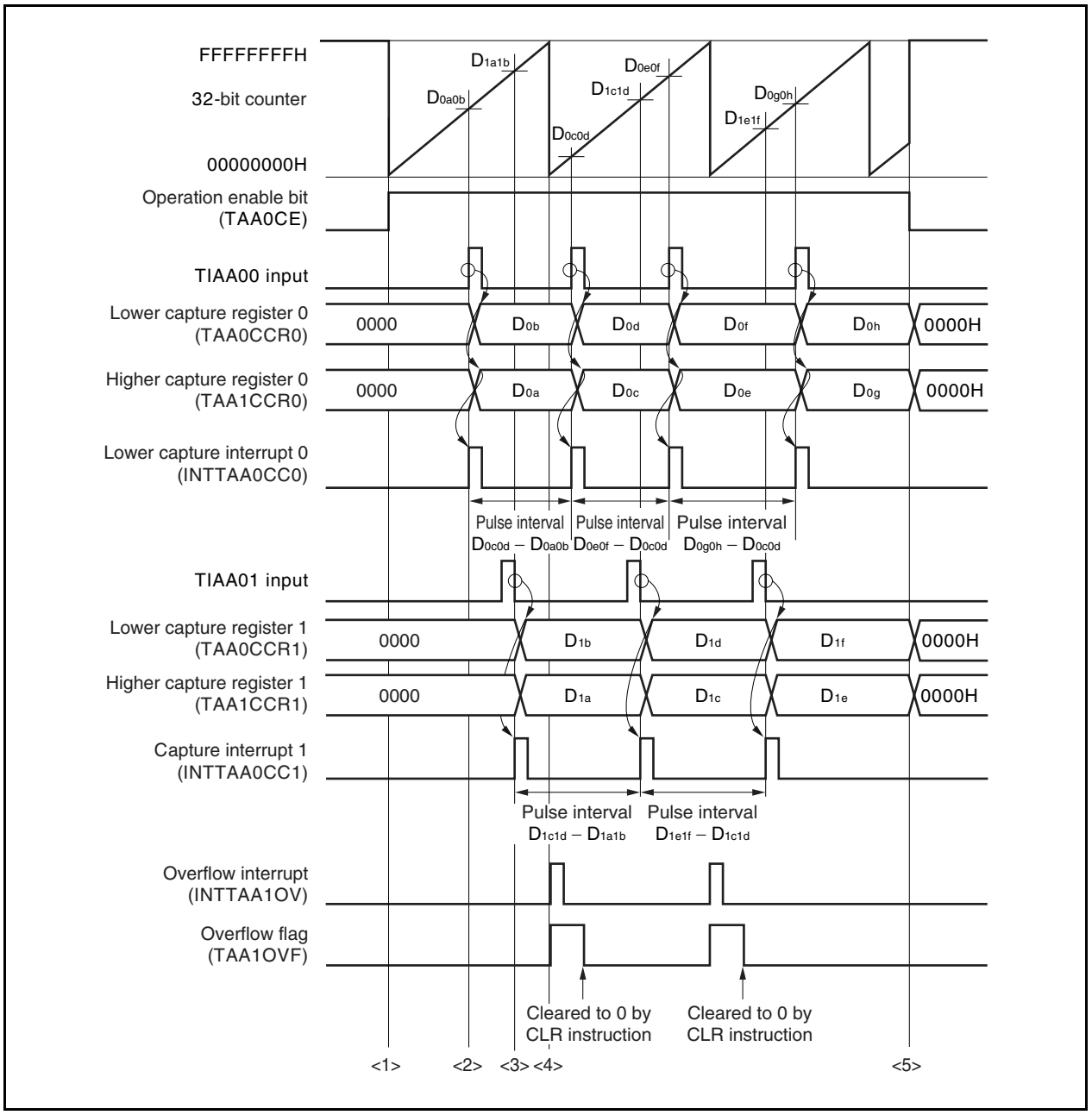


Figure 7-45. Operation Flow in Cascade Connection (2/2)

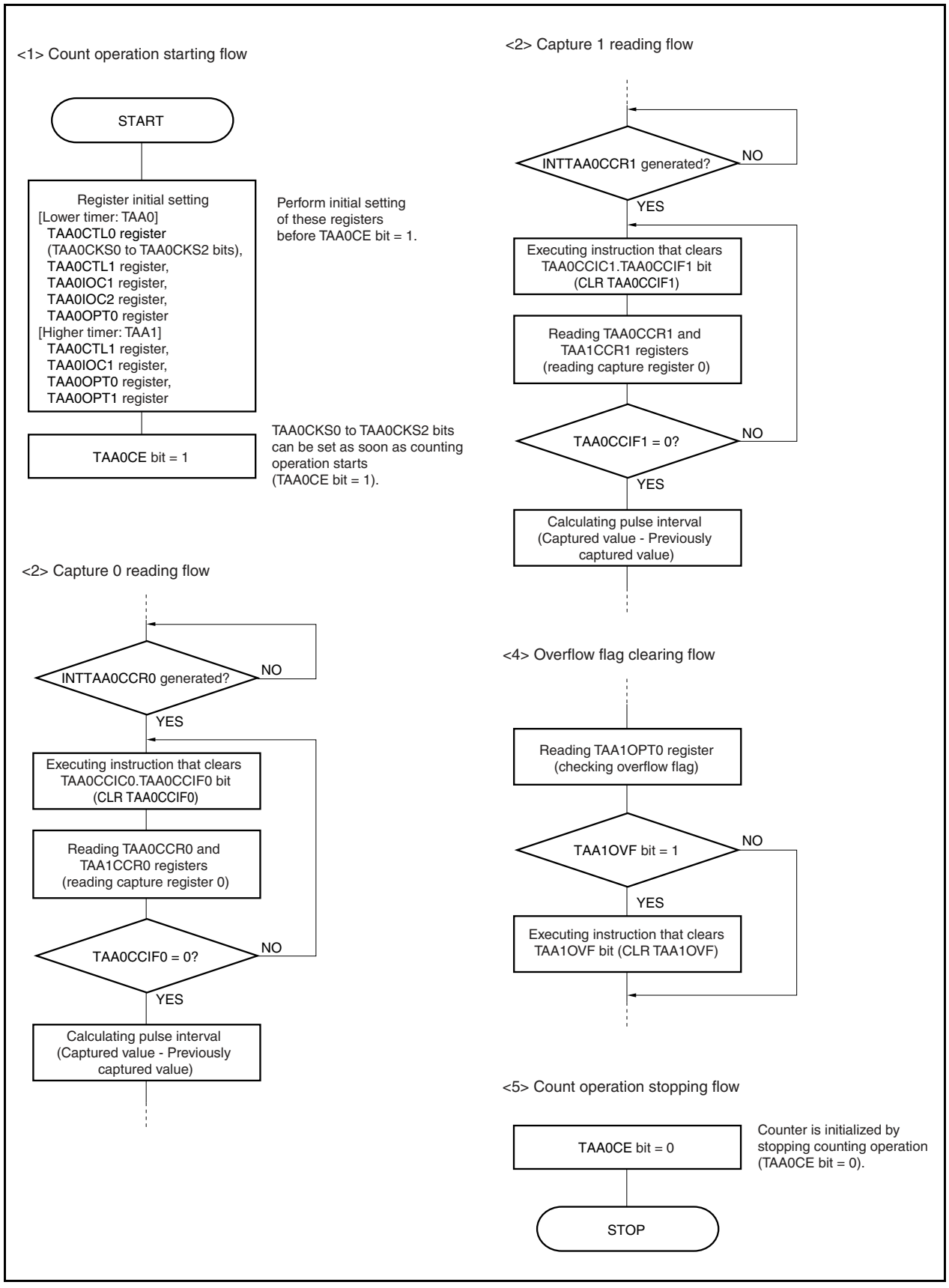
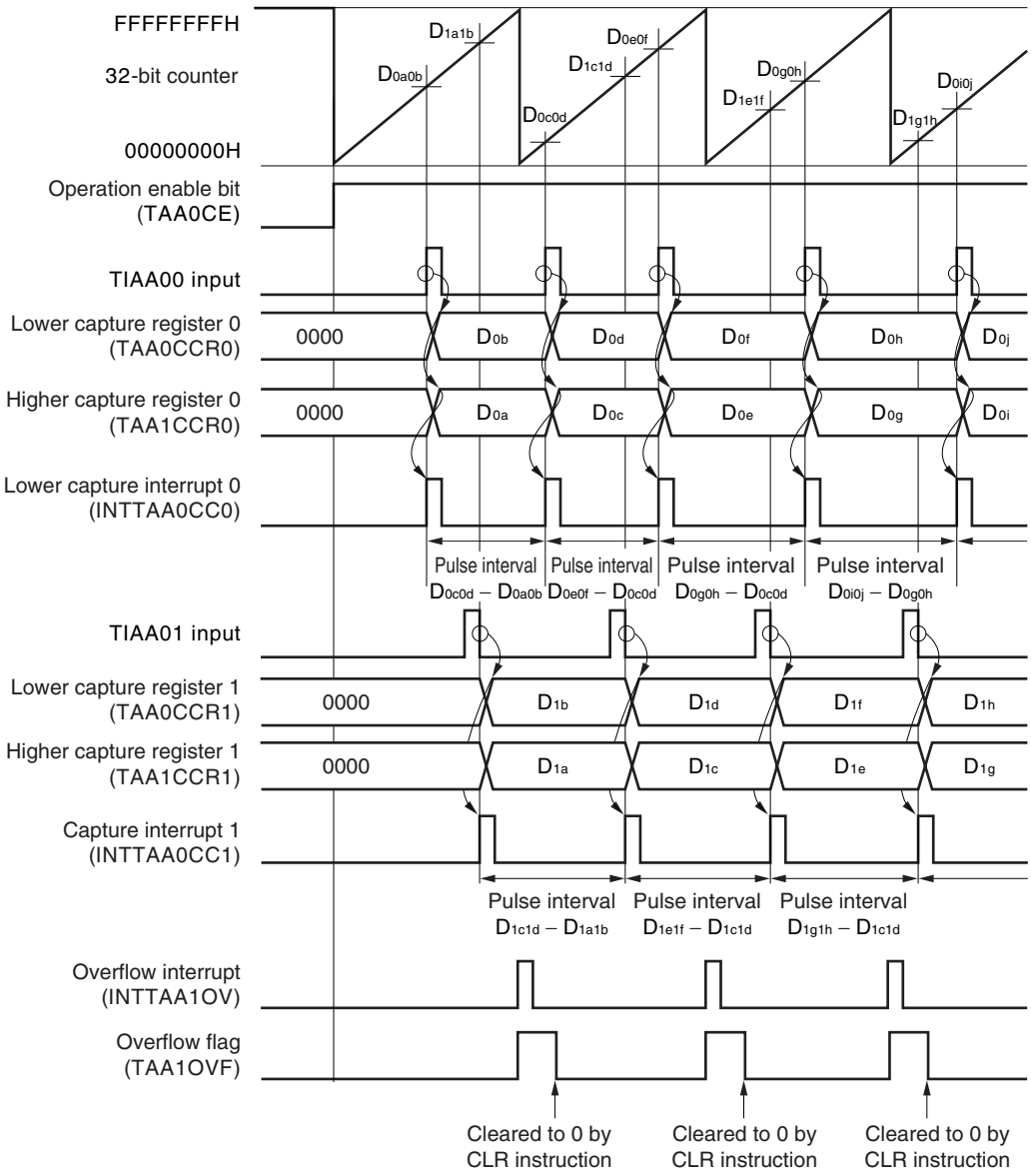


Figure 7-46. Example of Basic Timing When TAA0 and TAA1 Are Connected in Cascade



The counting operation is started when the TAA0CTL.TAA0CE bit is set to 1 and the count clock is supplied. When the valid edge input to the TIAA00 pin is detected, the count value is stored in the capture register 0 (TAA0CCR0 and TAA1CCR0), and capture interrupt 0 signal (INTTAA0CC0) is issued.

The timer counter continues the counting operation in synchronization with the count clock. When it counts up to FFFFFFFFH, the overflow interrupt (INTTAA1OV) is generated at the next clock and the overflow flag (TAA1OVF) is set to 1. The timer counter is cleared to 00000000H and continues counting up.

The overflow flag (TAA1OVF) is cleared by an instruction issued from the CPU that writes “0” to it.

Because the free-running timer mode is set, the timer counter cannot be cleared by detection of the valid edge input to the TIAA00 pin.

Using TOAA00 output is prohibited because it alternately functions as the TIAA00 input.

Capture register 1 (TAA0CCR1 and TAA1CCR1) also operates in the same manner.

If the lower timer counter (TAA0) overflows, an overflow interrupt (TAA0OVF) is generated. However, it is recommended to mask this interrupt because it cannot be used as an overflow interrupt of the 32-bit counter.

7.8 Selector Function

In the V850ES/Hx3, the alternate-function pins of port or peripheral I/O (TAA, TMM, or UARTD) signal can be selected as the capture trigger input of TAA.

If the signal input from the RXDDn pin is selected by the selector function when UARTD is used, baud rate error of the LIN reception transfer rate of UARTD can be calculated.

(1) Selector operation control register 0 (SELCNT0)

The SELCNT0 register is an 8-bit register that selects the capture trigger for TAA0 to TAA3.

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

	7	6	5	4	3	2	1	0
SELCNT0	0	ISEL06Note 1	ISEL05Note 2	ISEL04	ISEL03	ISEL02	0	0

ISEL06Note 1	Selection of TIAA31 capture trigger input signal
0	TIAA31 (alternately functions as P00) pin
1	RXDD3 (alternately functions as P40 or P80) pin

ISEL05Note 2	Selection of TIAA30 capture trigger input signal
0	TIAA30 (alternately functions as P01) pin
1	RXDD2 (alternately functions as P39) pin

ISEL04	Selection of TIAA11 capture trigger input signal
0	TIAA11 (alternately functions as P35) pin
1	RXDD1 (alternately functions as P91) pin

ISEL03	Selection of TIAA10 capture trigger input signal
0	TIAA10 (alternately functions as P34) pin
1	RXDD0 (alternately functions as P31) pin

ISEL02	Selection of TIAA01 capture trigger input signal
0	TIAA01 (alternately functions as P33) pin
1	INTTM0EQ0 signal of TMMNote 3

Notes 1. μPD70F3757 only

2. V850ES/HG3, V850ES/HJ3 only

3. Use the INTTM0EQ0 interrupt signal as the TIAA01 input signal under the following condition.
TMM0 operation clock ≥ TMAA1 operation clock × 4

Cautions 1. To set the ISEL02 to ISEL06 bits to 1, set the corresponding function pin in the capture input mode.

2. Set the ISEL02 to ISEL06 bits when operation of TAA0 to TAA3, UARTD3 to UARTD0, and TMM0 is stopped.

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

(2) Selector operation control register 1 (SELCNT1) (μPD70F3757 only)

The SELCNT1 register is an 8-bit register that selects the capture trigger for TAA4.

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: FFFFF30AH

	7	6	5	4	3	2	1	0
SELCNT1	0	0	0	0	0	ISEL12	0	0

ISEL12	Selection of TIAA41 capture trigger input signal
0	TIAA41 (alternately functions as P03) pin
1	RXDD5 (alternately functions as P913) pin

Cautions 1. To set the ISEL12 bit to 1, set the corresponding TIAA41 function pin in the capture input mode.

2. Set the ISEL12 bit when operation of TAA4 and UARTD5 is stopped.

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

(3) Selector operation control register 3 (SELCNT3) (μPD70F3757 only)

The SELCNT3 register is an 8-bit register that selects the capture trigger for TAA4.

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: FFFFF30EH

	7	6	5	4	3	2	1	0
SELCNT3	0	0	0	0	0	0	0	ISEL30

ISEL30	Selection of TIAA40 capture trigger input signal
0	TIAA40 (alternately functions as P02) pin
1	RXDD4 (alternately functions as P914) pin

Cautions 1. To set the ISEL30 bit to 1, set the corresponding TIAA40 function pin in the capture input mode.

2. Set the ISEL30 bit when operation of TAA4 and UARTD4 is stopped.

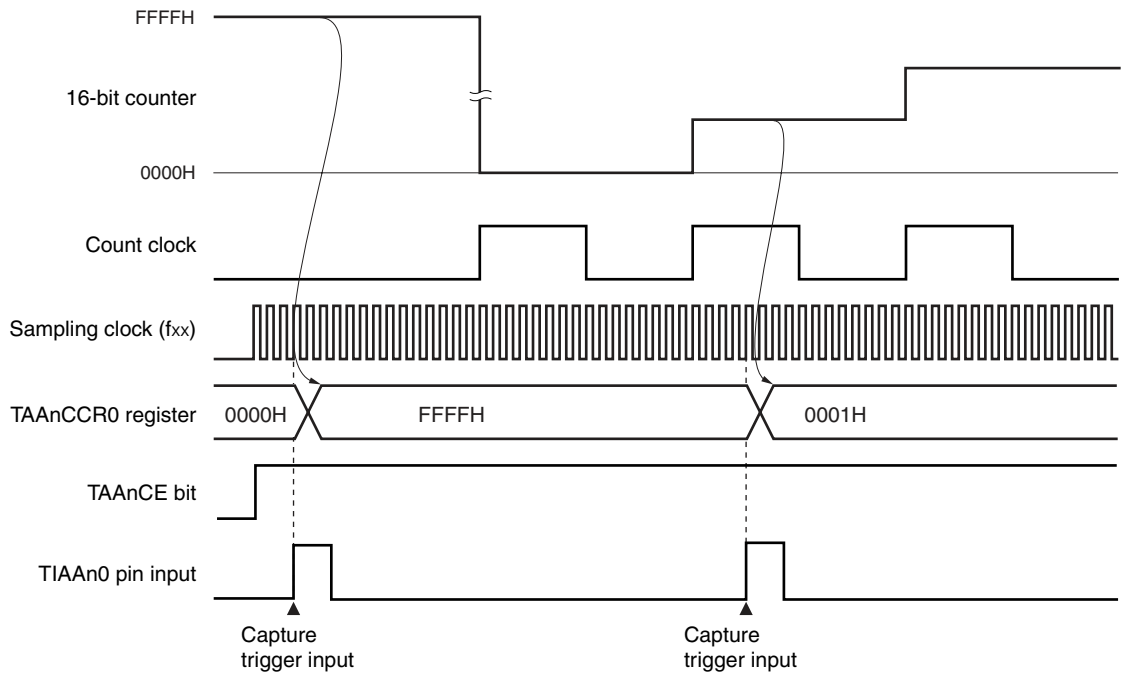
3. Be sure to set bits 7 to 1 to "0".

7.9 Cautions

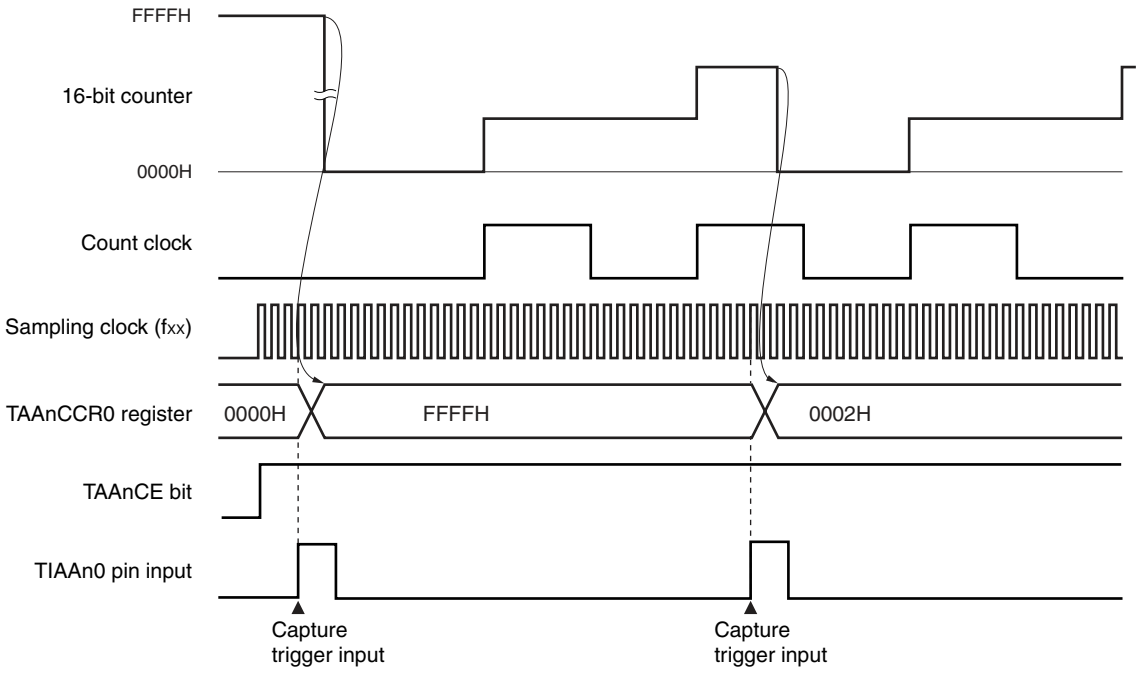
(1) Capture operation

When the capture operation is used and a slow clock is selected as the count clock, FFFFH, not 0000H, may be captured in the TAAnCCR0 and TAAnCCR1 registers if the capture trigger is input immediately after the TAAnCE bit is set to 1.

(a) Free-running timer mode



(b) Pulse width measurement mode



CHAPTER 8 16-BIT TIMER/EVENT COUNTER AB (TAB)

Timer AB (TAB) is a 16-bit timer/event counter.

The number of channels of each product of the V850ES/Hx3 is different from the others. Table 8-1 shows the number of channels of each product.

Table 8-1. Number of Channels of 16-bit Timer/Event Counter AB (TAB)

Product Name	V850ES/HE3	V850ES/HF3	V850ES/HG3	V850ES/HJ3
Number of channels	1 channel (TAB0)	1 channel (TAB0)	2 channels (TAB0, TAB1)	3 channels (TAB0 to TAB2)

In this chapter, the number of channels is expressed as n.

8.1 Overview

An outline of TABn is shown below.

- Clock selection: 8 ways
- Capture/trigger input pins: 4
- External event count input pins: 1
- External trigger input pins: 1
- Timer counters: 1
- Capture/compare registers: 4
- Capture/compare match interrupt request signals: 4
- Timer output pins: 4

8.2 Functions

TABn has the following functions.

- Interval timer
- External event counter
- External trigger pulse output
- One-shot pulse output
- PWM output
- Free-running timer
- Pulse width measurement
- Triangular wave PWM output
- Timer-tuned operation function

8.3 Configuration

TABn includes the following hardware.

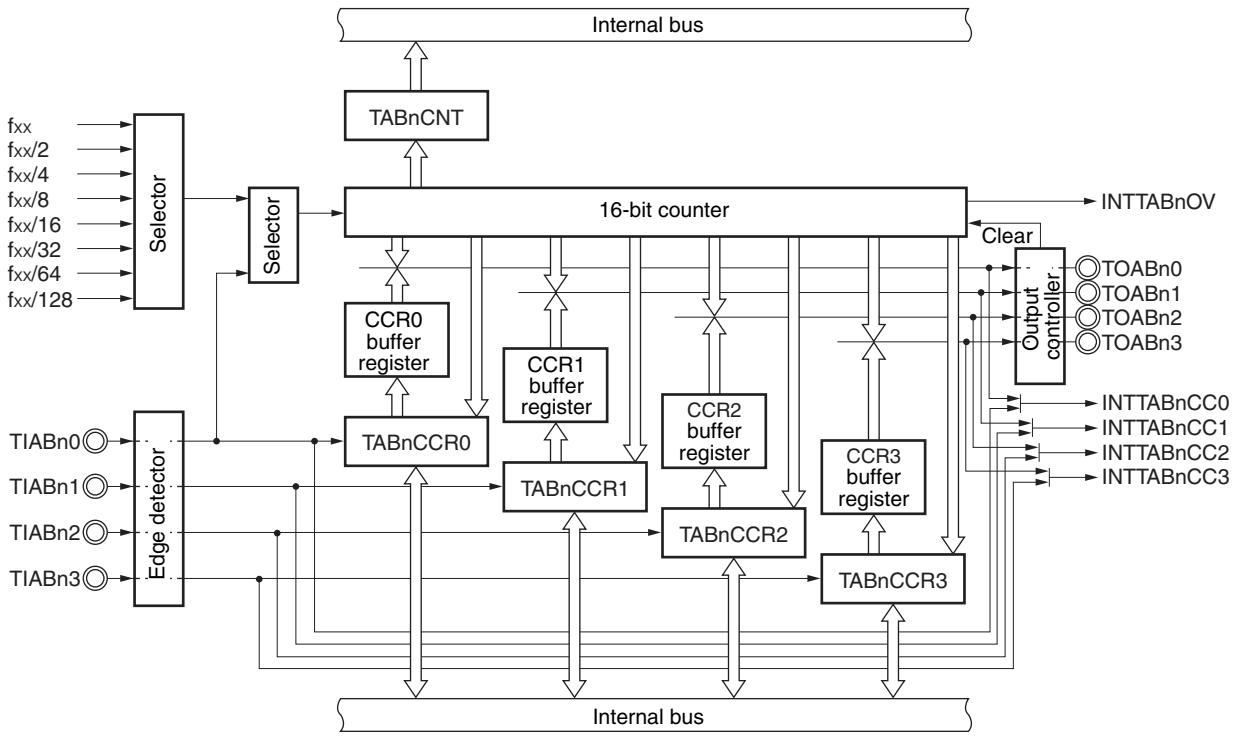
Table 8-2. Configuration of TABn

Item	Configuration
Timer register	16-bit counter
Registers	TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3) TABn counter read buffer register (TABnCNT) CCR0 to CCR3 buffer registers
Timer inputs	4 (TIABn0[Note 1] to TIABn3 pins)
Timer outputs	4 (TOABn0 to TOABn3 pins)
Control registers[Note 2]	TABn control registers 0, 1 (TABnCTL0, TABnCTL1) TABn I/O control registers 0 to 2 (TABnIOC0 to TABnIOC2) TABn option register 0 (TABnOPT0)

Notes 1. The TIABn0 pin functions alternately as a capture trigger input signal, external event count input signal, and external trigger input signal.

2. When using the functions of the TIABn0 to TIABn3 and TOABn0 to TOABn3 pins, see **Table 4-25 Using Port Pin as Alternate-Function Pin**.

Figure 8-1. Block Diagram of TABn



Remark fxx: Main clock frequency

(1) 16-bit counter

This 16-bit counter can count internal clocks or external events.

The count value of this counter can be read by using the TABnCNT register.

When the TABnCTL0.TABnCE bit = 0, the value of the 16-bit counter is FFFFH. If the TABnCNT register is read at this time, 0000H is read.

Reset sets the TABnCE bit to 0. Therefore, the 16-bit counter is set to FFFFH.

(2) CCR0 buffer register

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

When the TABnCCR0 register is used as a compare register, the value written to the TABnCCR0 register is transferred to the CCR0 buffer register. When the count value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTABnCC0) is generated.

The CCR0 buffer register cannot be read or written directly.

The CCR0 buffer register is cleared to 0000H after reset, as the TABnCCR0 register is cleared to 0000H.

(3) CCR1 buffer register

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

When the TABnCCR1 register is used as a compare register, the value written to the TABnCCR1 register is transferred to the CCR1 buffer register. When the count value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTABnCC1) is generated.

The CCR1 buffer register cannot be read or written directly.

The CCR1 buffer register is cleared to 0000H after reset, as the TABnCCR1 register is cleared to 0000H.

(4) CCR2 buffer register

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

When the TABnCCR2 register is used as a compare register, the value written to the TABnCCR2 register is transferred to the CCR2 buffer register. When the count value of the 16-bit counter matches the value of the CCR2 buffer register, a compare match interrupt request signal (INTTABnCC2) is generated.

The CCR2 buffer register cannot be read or written directly.

The CCR2 buffer register is cleared to 0000H after reset, as the TABnCCR2 register is cleared to 0000H.

(5) CCR3 buffer register

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

When the TABnCCR3 register is used as a compare register, the value written to the TABnCCR3 register is transferred to the CCR3 buffer register. When the count value of the 16-bit counter matches the value of the CCR3 buffer register, a compare match interrupt request signal (INTTABnCC3) is generated.

The CCR3 buffer register cannot be read or written directly.

The CCR3 buffer register is cleared to 0000H after reset, as the TABnCCR3 register is cleared to 0000H.

(6) Edge detector

This circuit detects the valid edges input to the TIABn0 and TIABn3 pins. No edge, rising edge, falling edge, or both the rising and falling edges can be selected as the valid edge by using the TABnIOC1 and TABnIOC2 registers.

(7) Output controller

This circuit controls the output of the TOABn0 to TOABn3 pins. The output controller is controlled by the TABnIOC0 register.

(8) Selector

This selector selects the count clock for the 16-bit counter. Eight types of internal clocks or an external event can be selected as the count clock.

8.4 Registers

The registers that control TABn are as follows.

- TABn control register 0 (TABnCTL0)
- TABn control register 1 (TABnCTL1)
- TABn I/O control register 0 (TABnIOC0)
- TABn I/O control register 1 (TABnIOC1)
- TABn I/O control register 2 (TABnIOC2)
- TABn option register 0 (TABnOPT0)
- TABn capture/compare register 0 (TABnCCR0)
- TABn capture/compare register 1 (TABnCCR1)
- TABn capture/compare register 2 (TABnCCR2)
- TABn capture/compare register 3 (TABnCCR3)
- TABn counter read buffer register (TABnCNT)

Remark When using the functions of the TIABn0 to TIABn3 and TOABn0 to TOABn3 pins, see **Table 4-25 Using Port Pin as Alternate-Function Pin**.

(1) TABn control register 0 (TABnCTL0)

The TABnCTL0 register is an 8-bit register that controls the operation of TABn.

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

Reset sets this register to 00H.

The same value can always be written to the TABnCTL0 register by software.

After reset: 00H R/W Address: TAB0CTL0 FFFFF540H, TAB1CTL0 FFFFF610H, TAB2CTL0 FFFFF620H

	7	6	5	4	3	2	1	0
TABnCTL0	TABnCE	0	0	0	0	TABnCKS2	TABnCKS1	TABnCKS0

TABnCE	TABn operation control
0	TABn operation disabled (TABn reset asynchronously[Note]).
1	TABn operation enabled. TABn operation started.

TABnCKS2	TABnCKS1	TABnCKS0	Internal count clock selection
0	0	0	f_{XX}
0	0	1	$f_{XX}/2$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/8$
1	0	0	$f_{XX}/16$
1	0	1	$f_{XX}/32$
1	1	0	$f_{XX}/64$
1	1	1	$f_{XX}/128$

Note TABnOPT0.TABnOVF bit, 16-bit counter, timer output (TOABn0 to TOABn3 pins)

Cautions 1. Set the TABnCKS2 to TABnCKS0 bits when the TABnCE bit = 0. When the value of the TABnCE bit is changed from 0 to 1, the TABnCKS2 to TABnCKS0 bits can be set simultaneously.

2. Be sure to set bits 3 to 6 to "0".

Remark f_{XX}: Main clock frequency

(2) TABn control register 1 (TABnCTL1)

The TABnCTL1 register is an 8-bit register that controls the operation of TABn.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: TAB0CTL1 FFFFF541H, TAB1CTL1 FFFFF611H, TAB2CTL1 FFFFF621H

	7	6	5	4	3	2	1	0
TABnCTL1	TABnSYE	TABnEST	TABnEEE	0	0	TABnMD2	TABnMD1	TABnMD0

TABnSYE	Tuned operation mode enable control
0	Independent operation mode (asynchronous operation mode)
1	Tuned operation mode (specification of slave operation) In this mode, timer P can operate in synchronization with a master timer. Master timer: TAB1, Slave timer: TAB2 For the tuned operation mode, see **7.6 Timer-Tuned Operation Function**. **Caution Be sure to set the TAB0SYE and TAB1SYE bits to 0.**

TABnEST	Software trigger control
0	–
1	Generate a valid signal for external trigger input. • In one-shot pulse output mode: A one-shot pulse is output with writing 1 to the TABnEST bit as the trigger. • In external trigger pulse output mode: A PWM waveform is output with writing 1 to the TABnEST bit as the trigger.

Cautions 1. The TABnEST bit is valid only in the external trigger pulse output mode or one-shot pulse output mode. In any other mode, writing 1 to this bit is ignored.

2. Be sure to set bits 3 and 4 to "0".

(2/2)

TABnEEE	Count clock selection
0	Disable operation with external event count input. (Perform counting with the count clock selected by the TABnCTL0.TABnCK0 to TABnCK2 bits.)
1	Enable operation with external event count input. (Perform counting at the valid edge of the external event count input signal.)
The TABnEEE bit selects whether counting is performed with the internal count clock or the valid edge of the external event count input.	

TABnMD2	TABnMD1	TABnMD0	Timer mode selection
0	0	0	Interval timer mode
0	0	1	External event count mode
0	1	0	External trigger pulse output mode
0	1	1	One-shot pulse output mode
1	0	0	PWM output mode
1	0	1	Free-running timer mode
1	1	0	Pulse width measurement mode
1	1	1	Triangular wave PWM mode

Cautions 1. External event count input is selected in the external event count mode regardless of the value of the TABnEEE bit.

2. Set the TABnEEE and TABnMD2 to TABnMD0 bits when the TABnCTL0.TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) The operation is not guaranteed when rewriting is performed with the TABnCE bit = 1. If rewriting was mistakenly performed, clear the TABnCE bit to 0 and then set the bits again.

(3) TABn I/O control register 0 (TABnIOC0)

The TABnIOC0 register is an 8-bit register that controls the timer output (TOABn0 to TOABn3 pins).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0IOC0 FFFFF542H, TAB1IOC0 FFFFF612H, TAB2IOC0 FFFFF622H

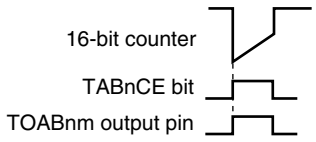
	7	6	5	4	3	2	1	0
TABnIOC0	TABnOL3	TABnOE3	TABnOL2	TABnOE2	TABnOL1	TABnOE1	TABnOL0	TABnOE0

TABnOLm	TOABnm pin output level setting (m = 0 to 3)[Note]
0	TOABnm pin high level start
1	TOABnm pin low level start

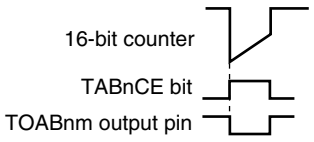
TABnOEm	TOABnm pin output setting (m = 0 to 3)
0	Timer output disabled • When TABnOLm bit = 0: Low level is output from the TOABnm pin • When TABnOLm bit = 1: High level is output from the TOABnm pin
1	Timer output enabled (A square wave is output from the TOABnm pin).

Note The output level of the timer output pin (TOABnm) specified by the TABnOLm bit is shown below.

• When TABnOLm bit = 0



• When TABnOLm bit = 1



Cautions 1. Rewrite the TABnOLm and TABnOEm bits when the TABnCTL0.TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) If rewriting was mistakenly performed, clear the TABnCE bit to 0 and then set the bits again.

2. Even if the TABnOLm bit is manipulated when the TABnCE and TABnOEm bits are 0, the TOABnm pin output level varies.

Remark m = 0 to 3

(4) TABn I/O control register 1 (TABnIOC1)

The TABnIOC1 register is an 8-bit register that controls the valid edge of the capture trigger input signals (TIABn0 to TIABn3 pins).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0IOC1 FFFFF543H, TAB1IOC1 FFFFF613H, TAB2IOC1 FFFFF623H

	7	6	5	4	3	2	1	0
TABnIOC1	TABnIS7	TABnIS6	TABnIS5	TABnIS4	TABnIS3	TABnIS2	TABnIS1	TABnIS0

TABnIS7	TABnIS6	Capture trigger input signal (TIABn3 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TABnIS5	TABnIS4	Capture trigger input signal (TIABn2 pin) valid edge detection
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TABnIS3	TABnIS2	Capture trigger input signal (TIABn1 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TABnIS1	TABnIS0	Capture trigger input signal (TIABn0 pin) valid edge setting
0	0	No edge detection (capture operation invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Cautions 1. Rewrite the TABnIS7 to TABnIS0 bits when the TABnCTL0.TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) If rewriting was mistakenly performed, clear the TABnCE bit to 0 and then set the bits again.

2. The TABnIS7 to TABnIS0 bits are valid only in the free-running timer mode and the pulse width measurement mode. In all other modes, a capture operation is not possible.

(5) TABn I/O control register 2 (TABnIOC2)

The TABnIOC2 register is an 8-bit register that controls the valid edge of the external event count input signal (TIABn0 pin) and external trigger input signal (TIABn0 pin).

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0IOC2 FFFFF544H, TAB1IOC2 FFFFF614H, TAB2IOC2 FFFFF624H

	7	6	5	4	3	2	1	0
TABnIOC2	0	0	0	0	TABnEES1	TABnEES0	TABnETS1	TABnETS0

TABnEES1	TABnEES0	External event count input signal (TIABn0 pin) valid edge setting
0	0	No edge detection (external event count invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

TABnETS1	TABnETS0	External trigger input signal (TIABn0 pin) valid edge setting
0	0	No edge detection (external trigger invalid)
0	1	Detection of rising edge
1	0	Detection of falling edge
1	1	Detection of both edges

Cautions 1. Rewrite the TABnEES1, TABnEES0, TABnETS1, and TABnETS0 bits when the TABnCTL0.TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) If rewriting was mistakenly performed, clear the TABnCE bit to 0 and then set the bits again.

2. The TABnEES1 and TABnEES0 bits are valid only when the TABnCTL1.TABnEEE bit = 1 or when the external event count mode (TABnCTL1.TABnMD2 to TABnCTL1.TABnMD0 bits = 001) has been set.

3. The TABnETS1 and TABnETS0 bits are valid only when the external trigger pulse output mode (TABnCTL1.TABnMD2 to TABnCTL1.TABnMD0 bits = 010) or the one-shot pulse output mode (TABnCTL1.TABnMD2 to TABnCTL1.TABnMD0 = 011) is set.

(6) TABn option register 0 (TABnOPT0)

The TABnOPT0 register is an 8-bit register used to set the capture/compare operation and detect an overflow.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: TAB0OPT0 FFFFF545H, TAB1OPT0 FFFFF615H, TAB2OPT0 FFFFF625H

TABnOPT0 (n = 0 to 2)

7	6	5	4	3	2	1	0
TABnCCS3	TABnCCS2	TABnCCS1	TABnCCS0	0	TAB0CMS Note	TAB0CUF Note	TABnOVF

TABnCCSm	TABnCCRm register capture/compare selection
0	Compare register selected
1	Capture register selected
The TABnCCSm bit setting is valid only in the free-running timer mode.	

TABnOVF	TABn overflow detection
Set (1)	Overflow occurred
Reset (0)	TABnOVF bit 0 written or TABnCTL0.TABnCE bit = 0
• The TABnOVF bit is set to 1 when the 16-bit counter count value overflows from FFFFH to 0000H in the free-running timer mode or the pulse width measurement mode. • An interrupt reABuest signal (INTTABnOV) is generated at the same time that the TABnOVF bit is set to 1. The INTTABnOV signal is not generated in modes other than the free-running timer mode and the pulse width measurement mode. • The TABnOVF bit is not cleared even when the TABnOVF bit or the TABnOPT0 register are read when the TABnOVF bit = 1. • The TABnOVF bit can be both read and written, but the TABnOVF bit cannot be set to 1 by software. Writing 1 has no influence on the operation of TABn.	

Note The TAB0CMS and TAB0CUF bits are used for the motor control function. For details, see **CHAPTER 10 MOTOR CONTROL FUNCTION**.

Cautions 1. Rewrite the TABnCCS3 to TABnCCS0 bits when the TABnCTL0.TABnCE bit = 0. (The same value can be written when the TABnCE bit = 1.) If rewriting was mistakenly performed, clear the TABnCE bit to 0 and then set the bits again.

2. Be sure to set bit 3 to "0". When the motor control function is not used, be sure to set bits 1 and 2 to "0".

Remark m = 0 to 3

(7) TABn capture/compare register 0 (TABnCCR0)

The TABnCCR0 register can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TABnOPT0.TABnCCS0 bit. In the pulse width measurement mode, the TABnCCR0 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TABnCCR0 register can be read or written during operation.

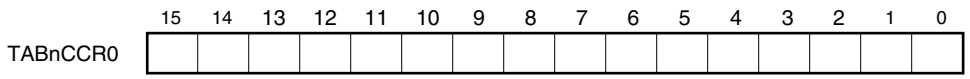
This register can be read or written in 16-bit units.

Reset sets this register to 0000H.

Caution Accessing the TABnCCR0 register is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- **When the CPU operates with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the internal oscillation clock**

After reset: 0000H R/W Address: TAB0CCR0 FFFFF546H, TAB1CCR0 FFFFF616H, TAB2CCR0 FFFFF626H



(a) Function as compare register

The TABnCCR0 register can be rewritten even when the TABnCTL0.TABnCE bit = 1.

The set value of the TABnCCR0 register is transferred to the CCR0 buffer register. When the value of the 16-bit counter matches the value of the CCR0 buffer register, a compare match interrupt request signal (INTTABnCC0) is generated. If TOABn0 pin output is enabled at this time, the output of the TOABn0 pin is inverted.

When the TABnCCR0 register is used as a cycle register in the interval timer mode, external event count mode, external trigger pulse output mode, one-shot pulse output mode, PWM output mode, or triangular wave PWM 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 TABnCCR0 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TABnCCR0 register if the valid edge of the capture trigger input pin (TIABn0 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TABnCCR0 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIABn0 pin) is detected.

Even if the capture operation and reading the TABnCCR0 register conflict, the correct value of the TABnCCR0 register can be read.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 8-3. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	Capture/Compare Register	How to Write Compare Register
Interval timer	Compare register	Anytime write
External event counter	Compare register	Anytime write
External trigger pulse output	Compare register	Batch write
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	–
Triangular wave PWM mode	Compare register	Batch write

(8) TABn capture/compare register 1 (TABnCCR1)

The TABnCCR1 register can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TABnOPT0.TABnCCS1 bit. In the pulse width measurement mode, the TABnCCR1 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TABnCCR1 register can be read or written during operation.

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

Reset sets this register to 0000H.

Caution Accessing the TABnCCR1 register is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- **When the CPU operates with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the internal oscillation clock**

After reset: 0000H R/W Address: TAB0CCR1 FFFFF548H, TAB1CCR1 FFFFF618H, TAB2CCR1 FFFFF628H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TABnCCR1																

(a) Function as compare register

The TABnCCR1 register can be rewritten even when the TABnCTL0.TABnCE bit = 1.

The set value of the TABnCCR1 register is transferred to the CCR1 buffer register. When the value of the 16-bit counter matches the value of the CCR1 buffer register, a compare match interrupt request signal (INTTABnCC1) is generated. If TOABn1 pin output is enabled at this time, the output of the TOABn1 pin is inverted.

(b) Function as capture register

When the TABnCCR1 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TABnCCR1 register if the valid edge of the capture trigger input pin (TIABn1 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TABnCCR1 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIABn1 pin) is detected.

Even if the capture operation and reading the TABnCCR1 register conflict, the correct value of the TABnCCR1 register can be read.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 8-4. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	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	–
Triangular wave PWM mode	Compare register	Batch write

(9) TABn capture/compare register 2 (TABnCCR2)

The TABnCCR2 register can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TABnOPT0.TABnCCS2 bit. In the pulse width measurement mode, the TABnCCR2 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TABnCCR2 register can be read or written during operation.

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

Reset sets this register to 0000H.

Caution Accessing the TABnCCR2 register is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- **When the CPU operates with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the internal oscillation clock**

After reset: 0000H R/W Address: TAB0CCR2 FFFFF54AH, TAB1CCR2 FFFFF61AH, TAB2CCR2 FFFFF62AH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TABnCCR2																

(a) Function as compare register

The TABnCCR2 register can be rewritten even when the TABnCTL0.TABnCE bit = 1.

The set value of the TABnCCR2 register is transferred to the CCR2 buffer register. When the value of the 16-bit counter matches the value of the CCR2 buffer register, a compare match interrupt request signal (INTTABnCC2) is generated. If TOABn2 pin output is enabled at this time, the output of the TOABn2 pin is inverted.

(b) Function as capture register

When the TABnCCR2 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TABnCCR2 register if the valid edge of the capture trigger input pin (TIABn2 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TABnCCR2 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIABn2 pin) is detected.

Even if the capture operation and reading the TABnCCR2 register conflict, the correct value of the TABnCCR2 register can be read.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 8-5. Function of Capture/Compare Register in Each Mode and How to Write Compare Register

Operation Mode	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	–
Triangular wave PWM mode	Compare register	Batch write

(10) TABn capture/compare register 3 (TABnCCR3)

The TABnCCR3 register can be used as a capture register or a compare register depending on the mode.

This register can be used as a capture register or a compare register only in the free-running timer mode, depending on the setting of the TABnOPT0.TABnCCS3 bit. In the pulse width measurement mode, the TABnCCR3 register can be used only as a capture register. In any other mode, this register can be used only as a compare register.

The TABnCCR3 register can be read or written during operation.

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

Reset sets this register to 0000H.

Caution Accessing the TABnCCR3 register is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- **When the CPU operates with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the internal oscillation clock**

After reset: 0000H R/W Address: TAB0CCR3 FFFFF54CH, TAB1CCR3 FFFFF61CH, TAB2CCR3 FFFFF62CH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TABnCCR3																

(a) Function as compare register

The TABnCCR3 register can be rewritten even when the TABnCTL0.TABnCE bit = 1.

The set value of the TABnCCR3 register is transferred to the CCR3 buffer register. When the value of the 16-bit counter matches the value of the CCR3 buffer register, a compare match interrupt request signal (INTTABnCC3) is generated. If TOABn3 pin output is enabled at this time, the output of the TOABn3 pin is inverted.

(b) Function as capture register

When the TABnCCR3 register is used as a capture register in the free-running timer mode, the count value of the 16-bit counter is stored in the TABnCCR3 register if the valid edge of the capture trigger input pin (TIABn3 pin) is detected. In the pulse-width measurement mode, the count value of the 16-bit counter is stored in the TABnCCR3 register and the 16-bit counter is cleared (0000H) if the valid edge of the capture trigger input pin (TIABn3 pin) is detected.

Even if the capture operation and reading the TABnCCR3 register conflict, the correct value of the TABnCCR3 register can be read.

The following table shows the functions of the capture/compare register in each mode, and how to write data to the compare register.

Table 8-6. 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	–
Triangular wave PWM mode	Compare register	Batch write

(11) TABn counter read buffer register (TABnCNT)

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

If this register is read when the TABnCTL0.TABnCE bit = 1, the count value of the 16-bit timer can be read.

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

The value of the TABnCNT register is cleared to 0000H when the TABnCE bit = 0. If the TABnCNT register is read at this time, the value of the 16-bit counter (FFFFH) is not read, but 0000H is read.

The value of the TABnCNT register is cleared to 0000H after reset, as the TABnCE bit is cleared to 0.

Caution Accessing the TABnCNT register is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- **When the CPU operates with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the internal oscillation clock**

After reset: 0000H R Address: TAB0CNT FFFFF54EH, TAB1CNT FFFFF61EH, TAB2CNT FFFFF62EH

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

8.5 Operation

TABn can perform the following operations.

Operation	TABnCTL1.TABnEST Bit (Software Trigger Bit)	TIABn0 Pin (External Trigger Input)	Capture/Compare Register Setting	Compare Register Write
Interval timer mode	Invalid	Invalid	Compare only	Anytime write
External event count mode[Note 1]	Invalid	Invalid	Compare only	Anytime write
External trigger pulse output mode[Note 2]	Valid	Valid	Compare only	Batch write
One-shot pulse output mode[Note 2]	Valid	Valid	Compare only	Anytime write
PWM output mode	Invalid	Invalid	Compare only	Batch write
Free-running timer mode	Invalid	Invalid	Switching enabled	Anytime write
Pulse width measurement mode[Note 2]	Invalid	Invalid	Capture only	Not applicable
Triangular wave PWM mode	Invalid	Invalid	Compare only	Batch write

Notes 1. To use the external event count mode, specify that the valid edge of the TIABn0 pin capture trigger input is not detected (by clearing the TABnIOC1.TABnIS1 and TABnIOC1.TABnIS0 bits to “00”).

2. When using the external trigger pulse output mode, one-shot pulse output mode, and pulse width measurement mode, select the internal clock as the count clock (by clearing the TABnCTL1.TABnEEE bit to 0).

8.5.1 Interval timer mode (TABnMD2 to TABnMD0 bits = 000)

In the interval timer mode, an interrupt request signal (INTTABnCC0) is generated at the specified interval if the TABnCTL0.TABnCE bit is set to 1. A square wave whose half cycle is equal to the interval can be output from the TOABn0 pin.

Usually, the TABnCCR1 to TABnCCR3 registers are not used in the interval timer mode.

Figure 8-2. Configuration of Interval Timer

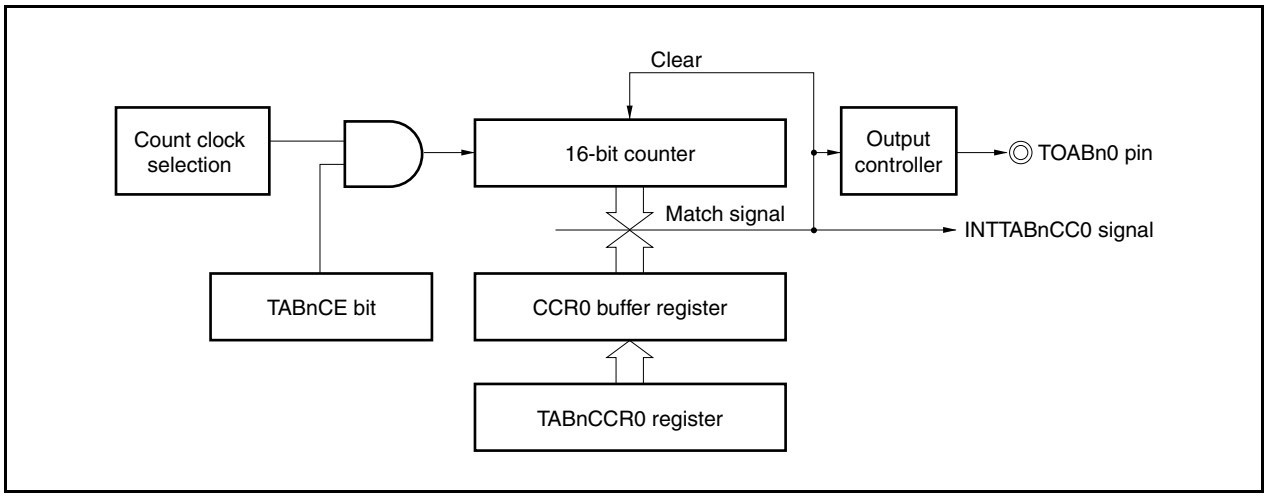
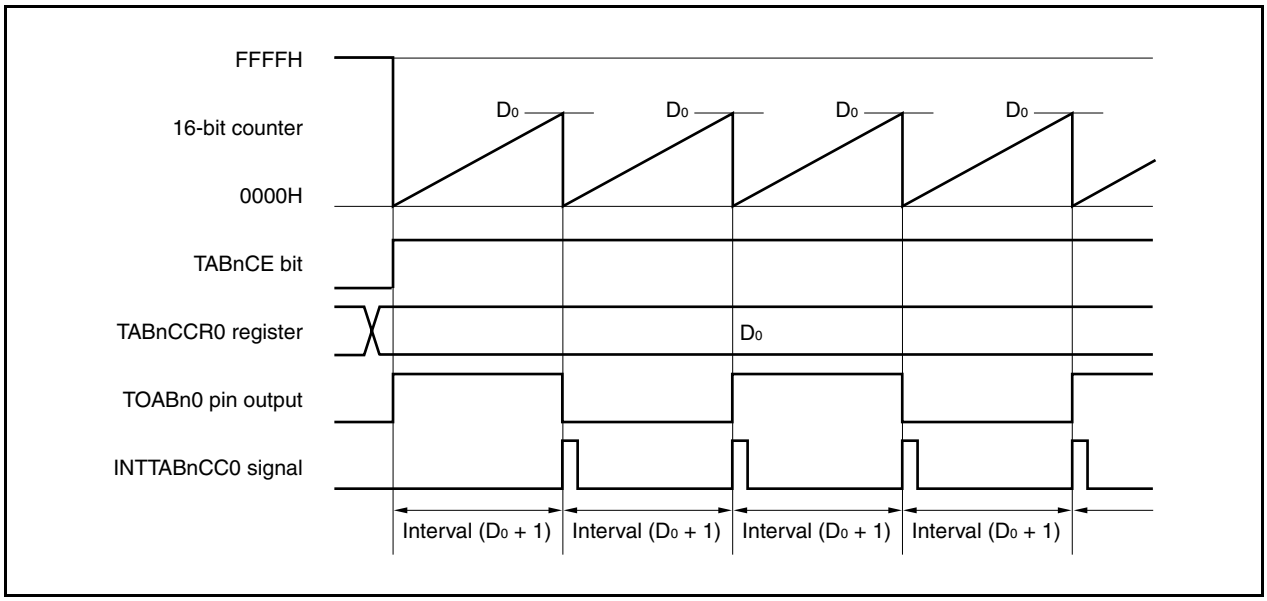


Figure 8-3. Basic Timing of Operation in Interval Timer Mode



When the TABnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H in synchronization with the count clock, and the counter starts counting. At this time, the output of the TOABn0 pin is inverted. Additionally, the set value of the TABnCCR0 register is transferred to the CCR0 buffer register.

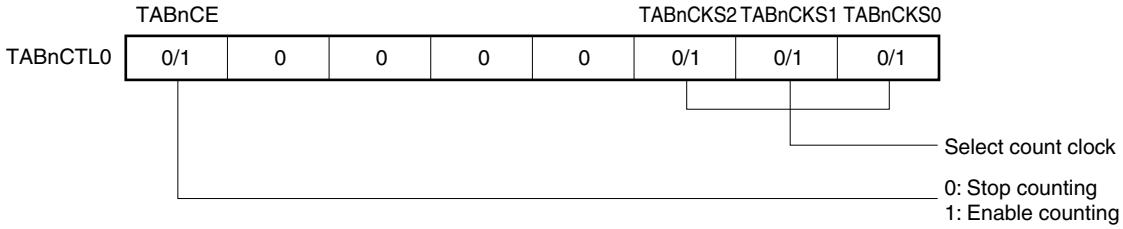
When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, the output of the TOABn0 pin is inverted, and a compare match interrupt request signal (INTTABnCC0) is generated.

The interval can be calculated by the following expression.

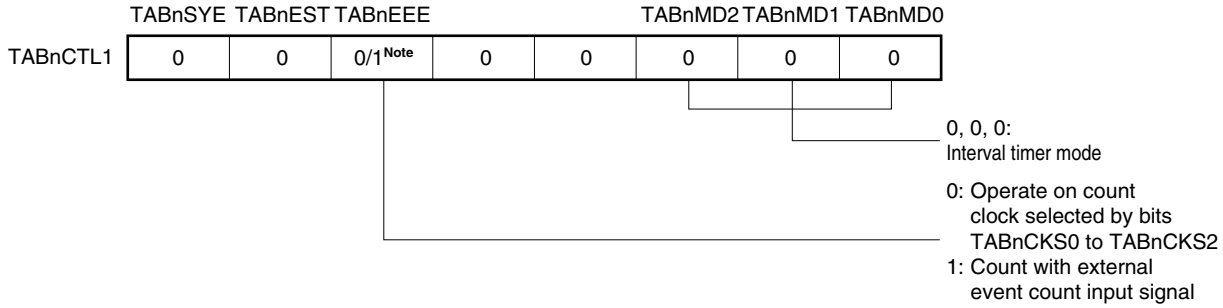
Interval = (Set value of TABnCCR0 register + 1) × Count clock cycle

Figure 8-4. Register Setting for Interval Timer Mode Operation (1/2)

(a) TABn control register 0 (TABnCTL0)



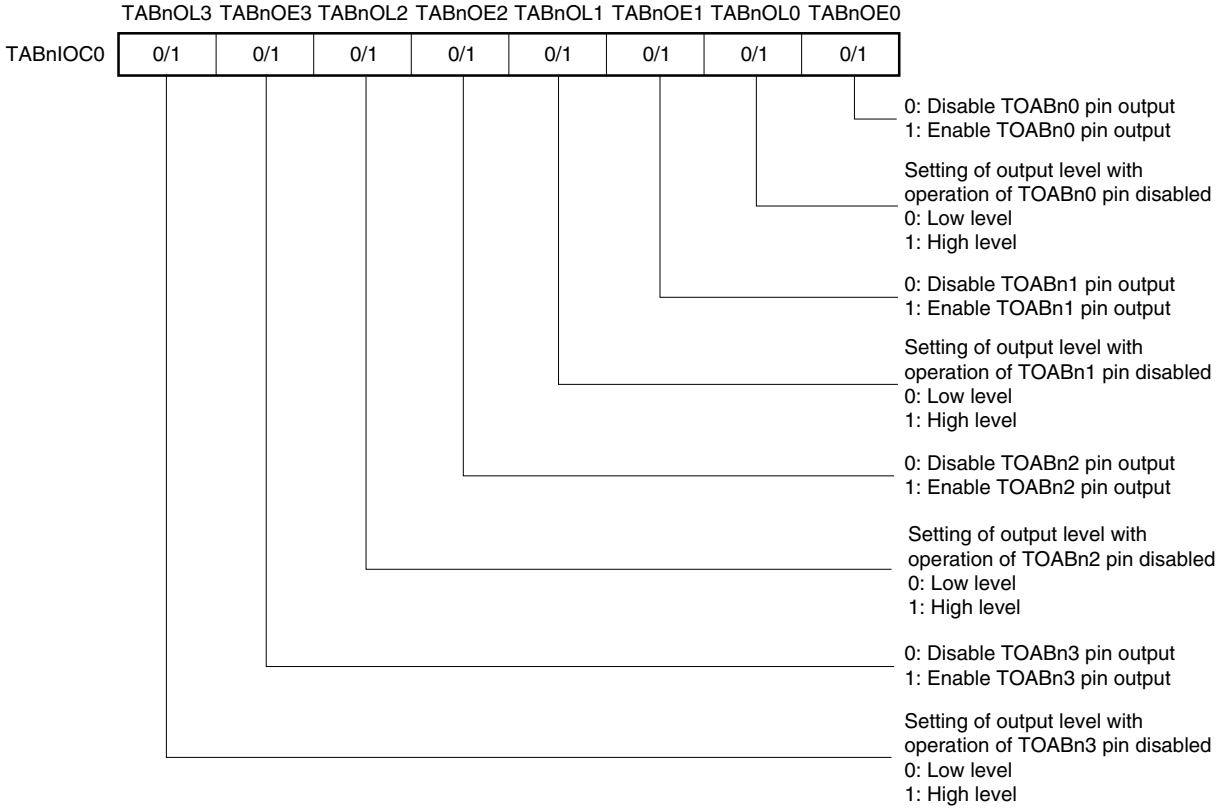
(b) TABn control register 1 (TABnCTL1)



Note This bit can be set to 1 only when the interrupt request signals (INTTABnCC0 and INTTABnCCk) are masked by the interrupt mask flags (TABnCCMK0 to TABnCCMKk) and the timer output (TOABnk) is performed at the same time. However, the TABnCCR0 and TABnCCRk registers must be set to the same value (see **8.5.1 (2) (d) Operation of TABnCCR1 to TABnCCR3 registers**) (k = 1 to 3).

Figure 8-4. Register Setting for Interval Timer Mode Operation (2/2)

(c) TABn I/O control register 0 (TABnIOC0)



(d) TABn counter read buffer register (TABnCNT)

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

(e) TABn capture/compare register 0 (TABnCCR0)

If the TABnCCR0 register is set to D_0, the interval is as follows.

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

(f) TABn capture/compare registers 1 to 3 (TABnCCR1 to TABnCCR3)

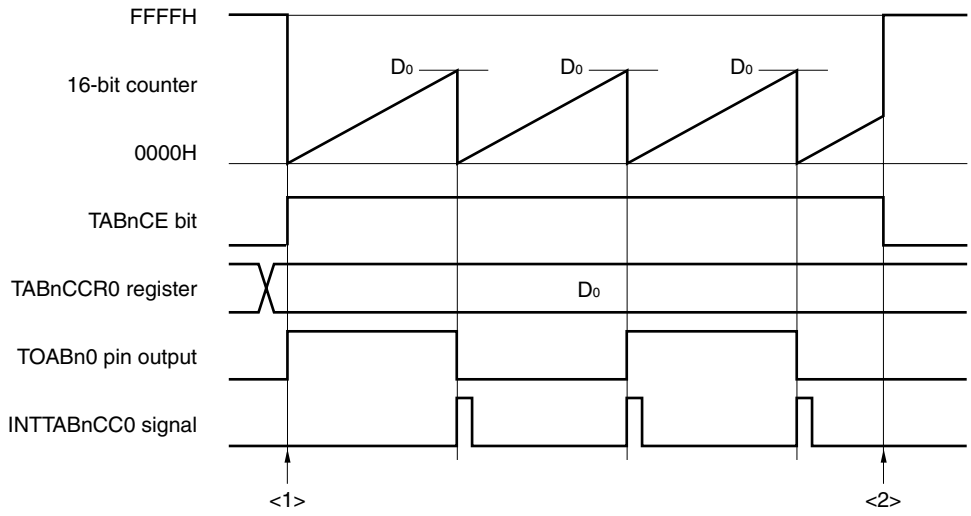
Usually, the TABnCCR1 to TABnCCR3 registers are not used in the interval timer mode. However, the set value of the TABnCCR1 to TABnCCR3 registers are transferred to the CCR1 to CCR3 buffer registers. The compare match interrupt request signals (INTTABnCCR1 to INTTABnCCR3) is generated when the count value of the 16-bit counter matches the value of the CCR1 to CCR3 buffer registers.

Therefore, mask the interrupt request by using the corresponding interrupt mask flags (TABnCCMK1 to TABnCCMK3).

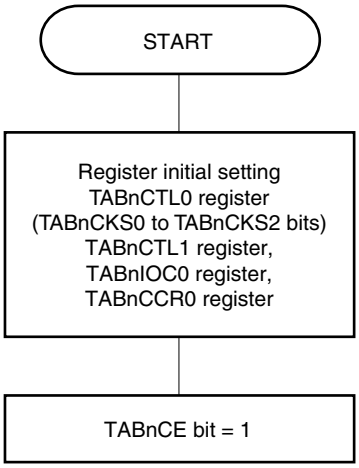
Remark TABn I/O control register 1 (TABnIOC1), TABn I/O control register 2 (TABnIOC2), and TABn option register 0 (TABnOPT0) are not used in the interval timer mode.

(1) Interval timer mode operation flow

Figure 8-5. Software Processing Flow in Interval Timer Mode



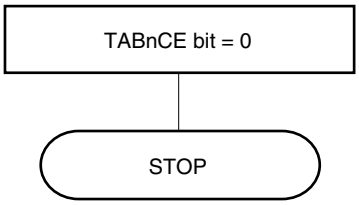
<1> Count operation start flow



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

The TABnCKS0 to TABnCKS2 bits can be set at the same time when counting has been started (TABnCE bit = 1).

<2> Count operation stop flow



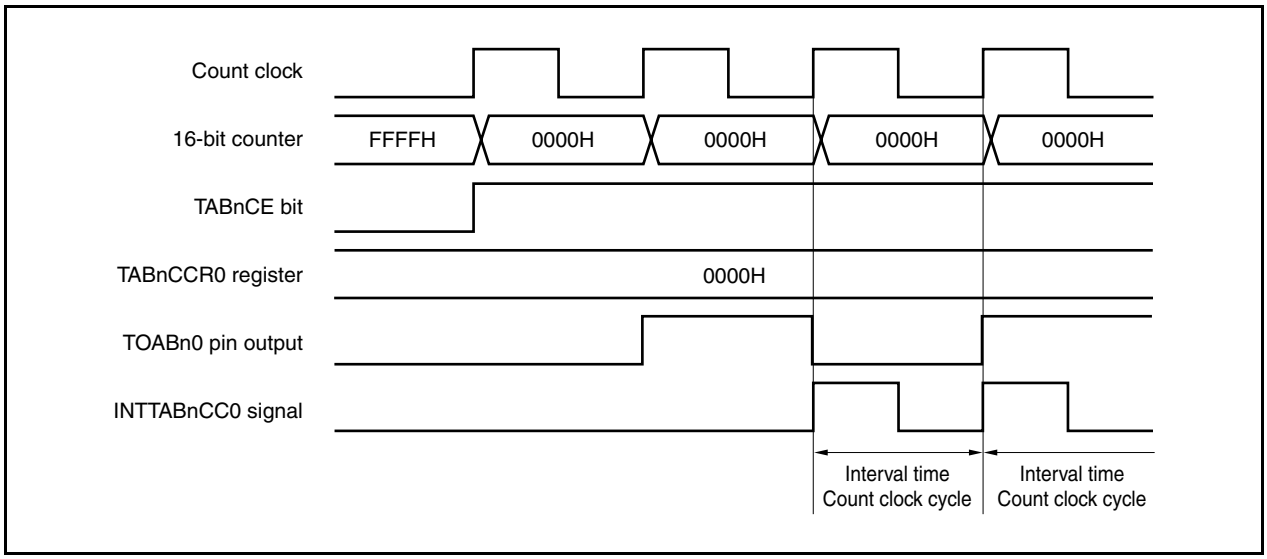
The counter is initialized and counting is stopped by clearing the TABnCE bit to 0.

(2) Interval timer mode operation timing

(a) Operation if TABnCCR0 register is set to 0000H

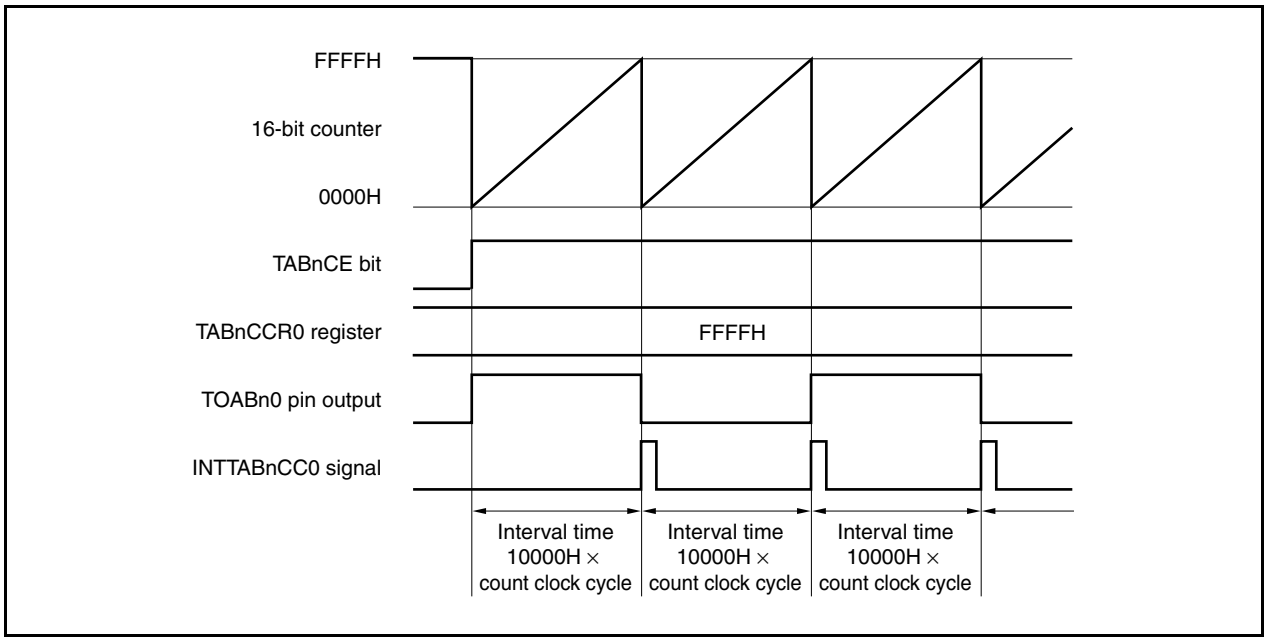
If the TABnCCR0 register is set to 0000H, the INTTABnCC0 signal is generated at each count clock subsequent to the first count clock, and the output of the TOABn0 pin is inverted.

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



(b) Operation if TABnCCR0 register is set to FFFFH

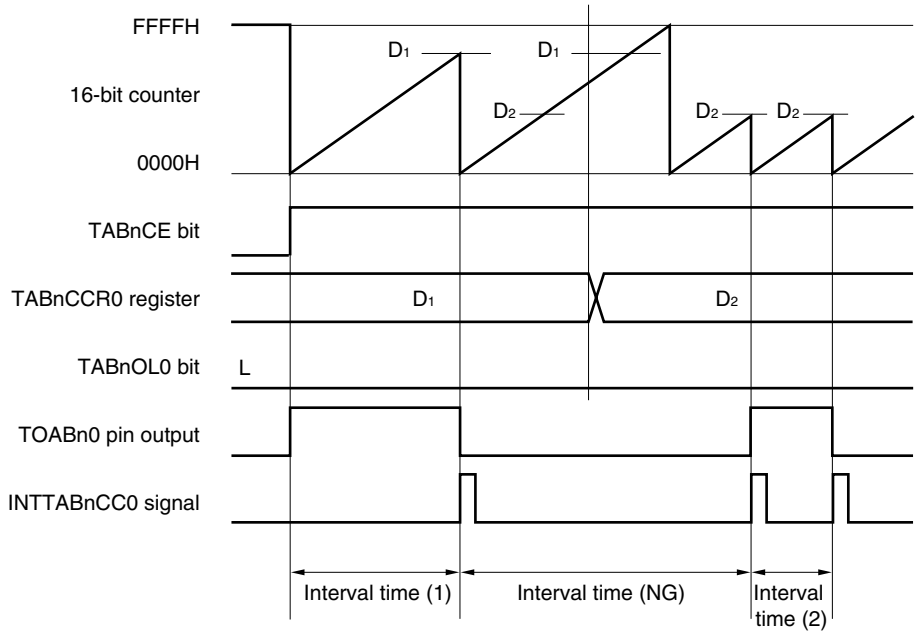
If the TABnCCR0 register is set to FFFFH, the 16-bit counter counts up to FFFFH. The counter is cleared to 0000H in synchronization with the next count-up timing. The INTTABnCC0 signal is generated and the output of the TOABn0 pin is inverted. At this time, an overflow interrupt request signal (INTTABnOV) is not generated, nor is the overflow flag (TABnOPT0.TABnOVF bit) set to 1.



(c) Notes on rewriting TABnCCR0 register

To change the value of the TABnCCR0 register to a smaller value, stop counting once and then change the set value.

If the value of the TABnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.



Remark Interval time (1): $(D_1 + 1) \times$ Count clock cycle
Interval time (NG): $(10000H + D_2 + 1) \times$ Count clock cycle
Interval time (2): $(D_2 + 1) \times$ Count clock cycle

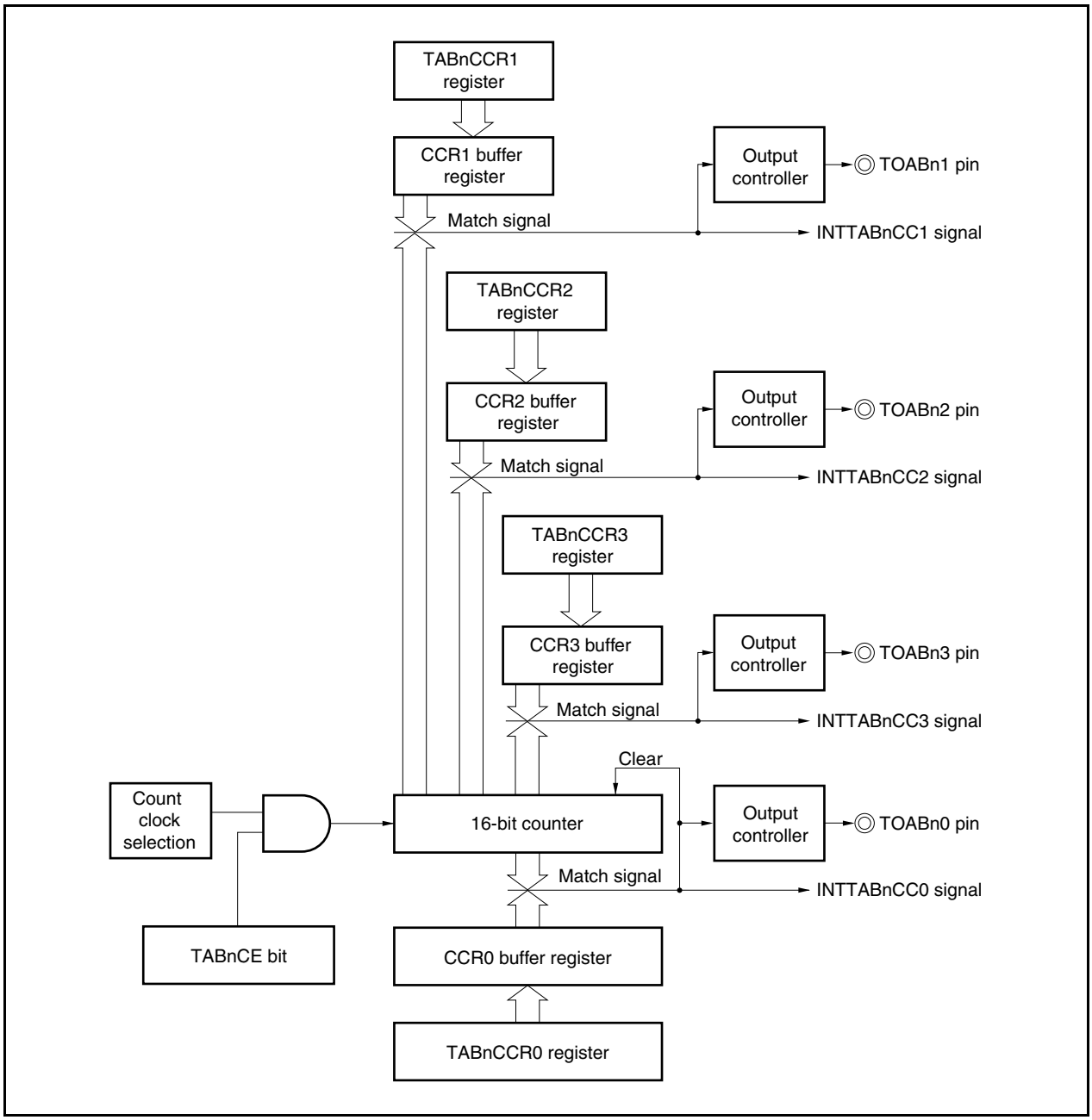
If the value of the TABnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TABnCCR0 register has been rewritten. Consequently, the value of the 16-bit counter that is compared is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTABnCC0 signal is generated and the output of the TOABn0 pin is inverted.

Therefore, the INTTABnCC0 signal may not be generated at the interval time "$(D_1 + 1) \times$ Count clock cycle" or "$(D_2 + 1) \times$ Count clock cycle" originally expected, but may be generated at an interval of "$(10000H + D_2 + 1) \times$ Count clock period".

(d) Operation of TABnCCR1 to TABnCCR3 registers

Figure 8-6. Configuration of TABnCCR1 to TABnCCR3 Registers

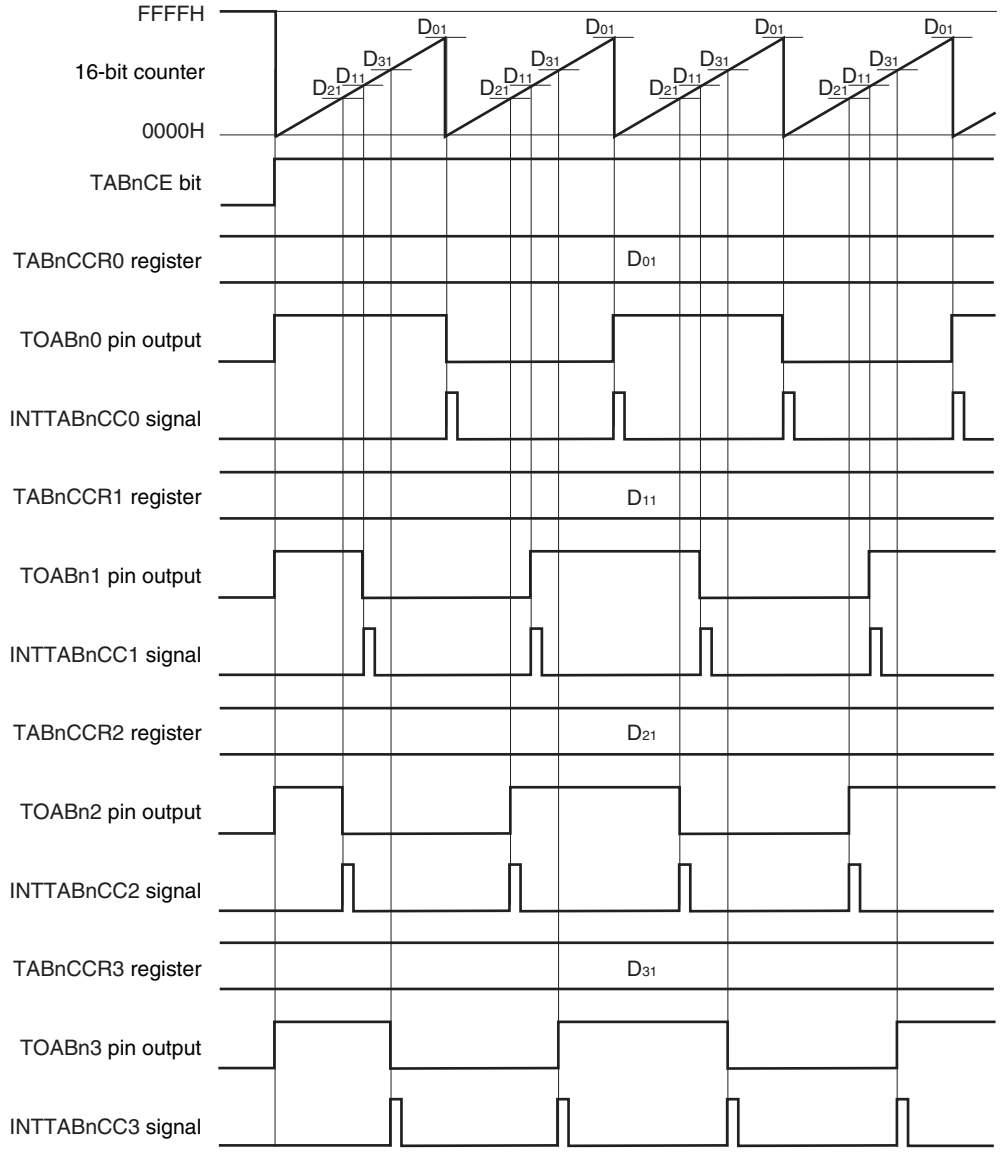


If the set value of the TABnCCRk register is less than the set value of the TABnCCR0 register, the INTTABnCCk signal is generated once per cycle. At the same time, the output of the TOABnk pin is inverted.

The TOABnk pin outputs a square wave with the same cycle as that output by the TOABn0 pin.

Remark k = 1 to 3

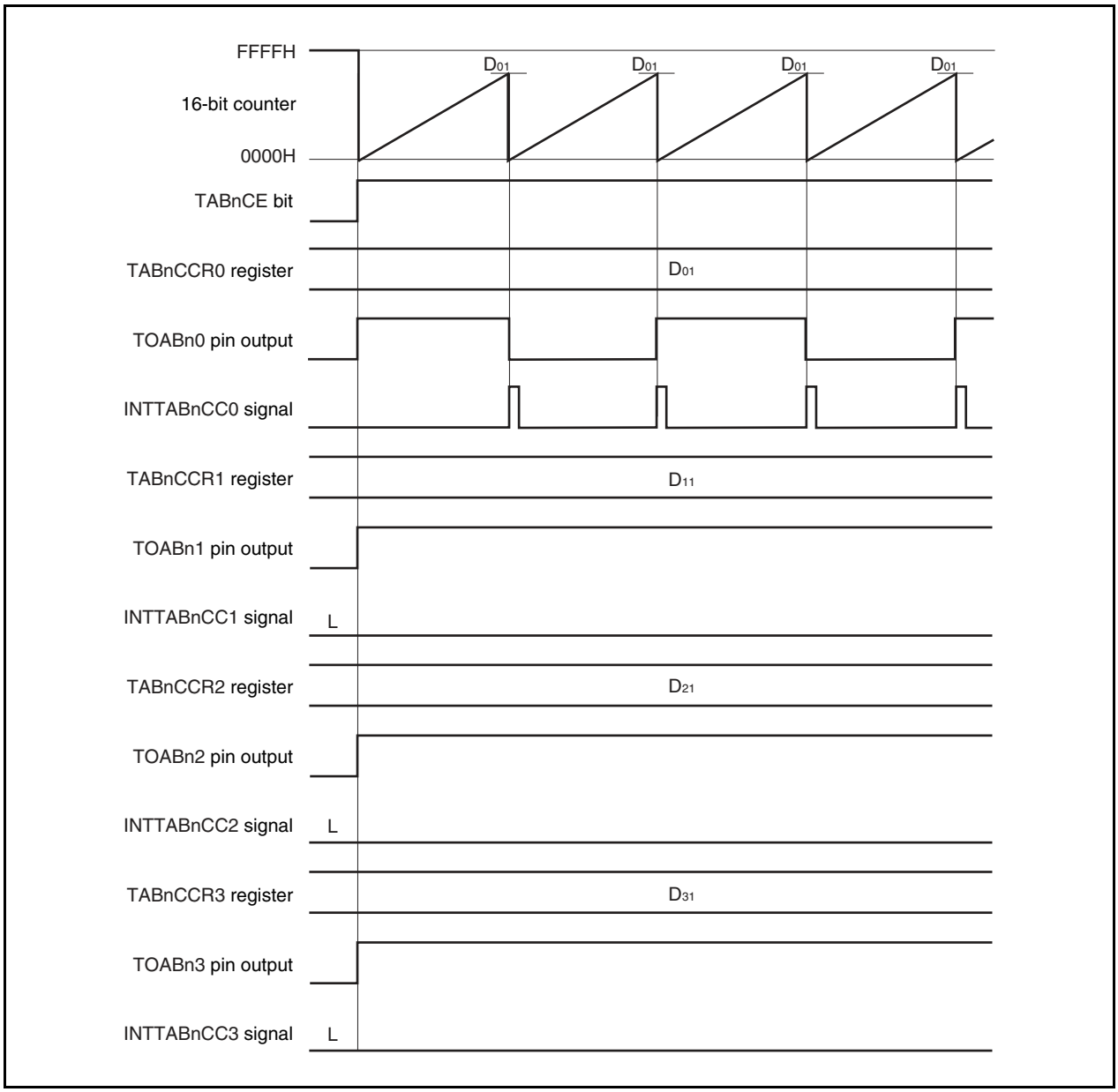
Figure 8-7. Timing Chart When $D_{01} \geq D_{k1}$



If the set value of the TABnCCRk register is greater than the set value of the TABnCCR0 register, the count value of the 16-bit counter does not match the value of the TABnCCRk register. Consequently, the INTTABnCCk signal is not generated, nor is the output of the TOABnk pin changed.

Remark k = 1 to 3

Figure 8-8. Timing Chart When $D_{01} < D_{k1}$



8.5.2 External event count mode (TABnMD2 to TABnMD0 bits = 001)

In the external event count mode, the valid edge of the external event count input is counted when the TABnCTL0.TABnCE bit is set to 1, and an interrupt request signal (INTTABnCC0) is generated each time the specified number of edges have been counted. The TOABn0 pin cannot be used.

Usually, the TABnCCR1 to TABnCCR3 registers are not used in the external event count mode.

Figure 8-9. Configuration in External Event Count Mode

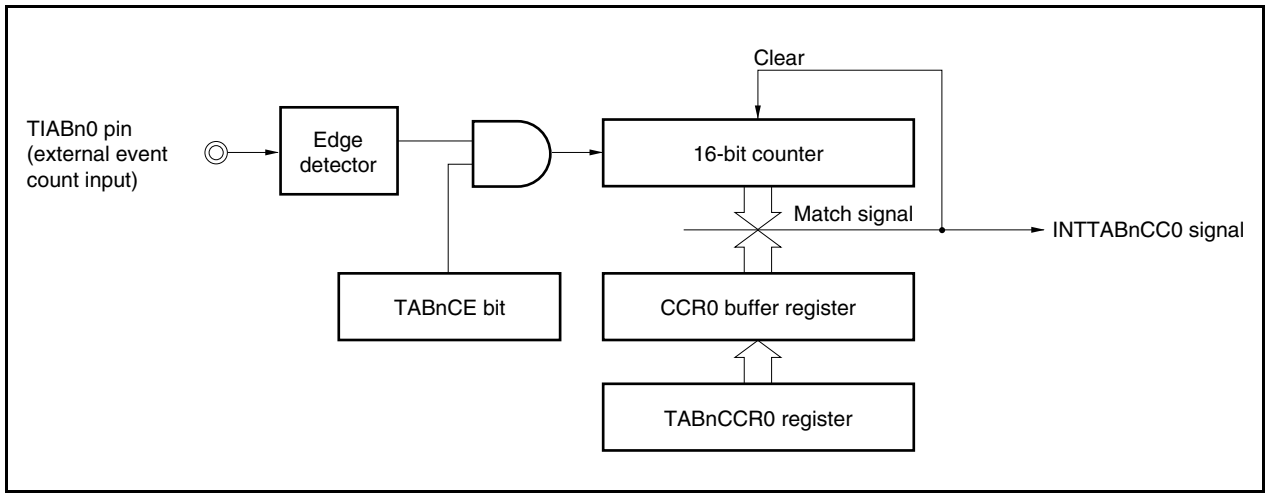
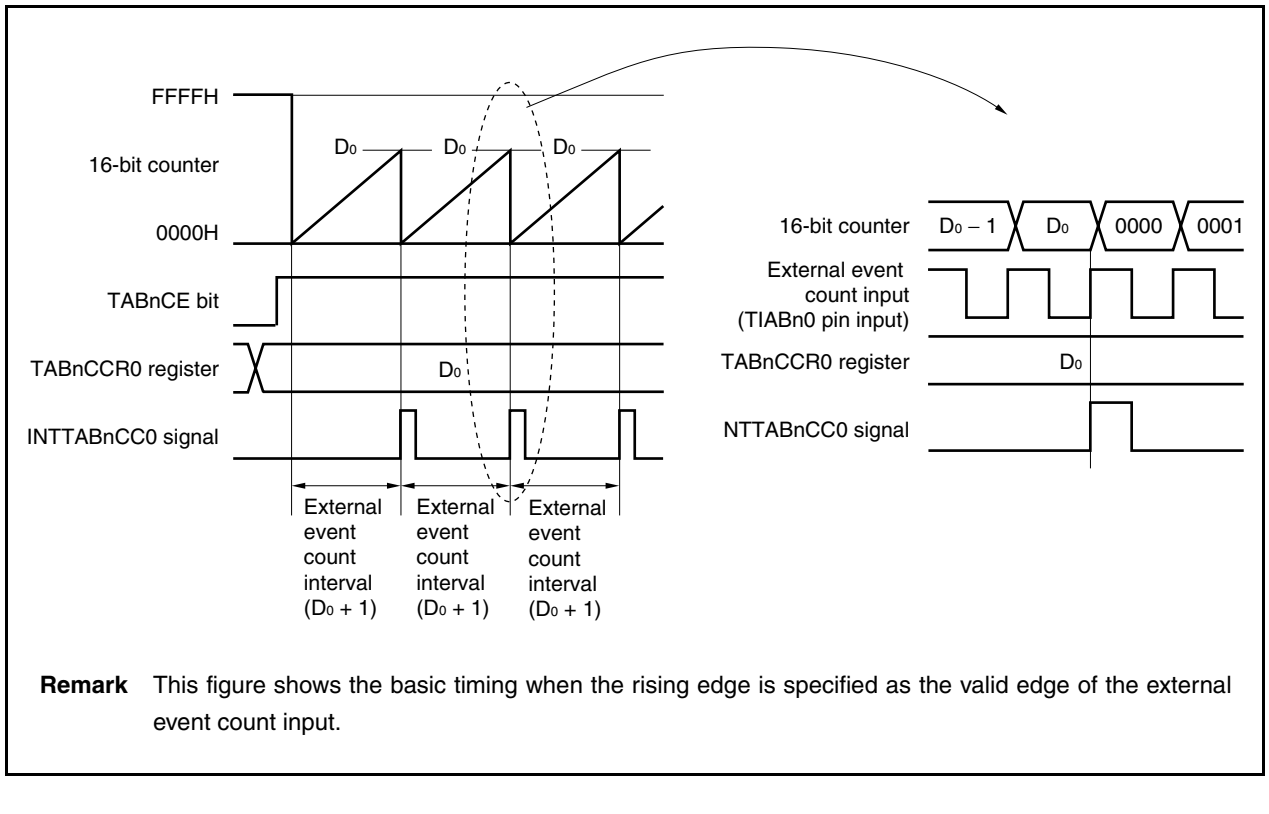


Figure 8-10. Basic Timing in External Event Count Mode



Remark This figure shows the basic timing when the rising edge is specified as the valid edge of the external event count input.

When the TABnCE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H. The counter counts each time the valid edge of external event count input is detected. Additionally, the set value of the TABnCCR0 register is transferred to the CCR0 buffer register.

When the count value of the 16-bit counter matches the value of the CCR0 buffer register, the 16-bit counter is cleared to 0000H, and a compare match interrupt request signal (INTTABnCC0) is generated.

The INTTABnCC0 signal is generated each time the valid edge of the external event count input has been detected (set value of TABnCCR0 register + 1) times.

Figure 8-11. Register Setting for Operation in External Event Count Mode (1/2)

(a) TABn control register 0 (TABnCTL0)

	TABnCE					TABnCKS2	TABnCKS1	TABnCKS0
TABnCTL0	0/1	0	0	0	0	0	0	0

0: Stop counting
1: Enable counting

(b) TABn control register 1 (TABnCTL1)

	TABnSYE	TABnEST	TABnEEE			TABnMD2	TABnMD1	TABnMD0
TABnCTL1	0	0	0	0	0	0	0	1

0, 0, 1:
External event count mode

(c) TABn I/O control register 0 (TABnIOC0)

	TABnOL3	TABnOE3	TABnOL2	TABnOE2	TABnOL1	TABnOE1	TABnOL0	TABnOE0
TABnIOC0	0	0	0	0	0	0	0	0

0: Disable TOABn0 pin output
0: Disable TOABn1 pin output
0: Disable TOABn2 pin output
0: Disable TOABn3 pin output

(d) TABn I/O control register 2 (TABnIOC2)

					TABnEES1	TABnEES0	TABnETS1	TABnETS0
TABnIOC2	0	0	0	0	0/1	0/1	0	0

Select valid edge
of external event
count input

(e) TABn counter read buffer register (TABnCNT)

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

Figure 8-11. Register Setting for Operation in External Event Count Mode (2/2)

(f) TABn capture/compare register 0 (TABnCCR0)

If D_0 is set to the TABnCCR0 register, the counter is cleared and a compare match interrupt request signal (INTTABnCC0) is generated when the number of external event counts reaches (D_0 + 1).

(g) TABn capture/compare registers 1 to 3 (TABnCCR1 to TABnCCR3)

Usually, the TABnCCR1 to TABnCCR3 registers are not used in the external event count mode. However, the set value of the TABnCCR1 to TABnCCR3 registers are transferred to the CCR1 to CCR3 buffer registers. When the count value of the 16-bit counter matches the value of the CCR1 to CCR3 buffer registers, compare match interrupt request signals (INTTABnCC1 to INTTABnCC3) are generated.

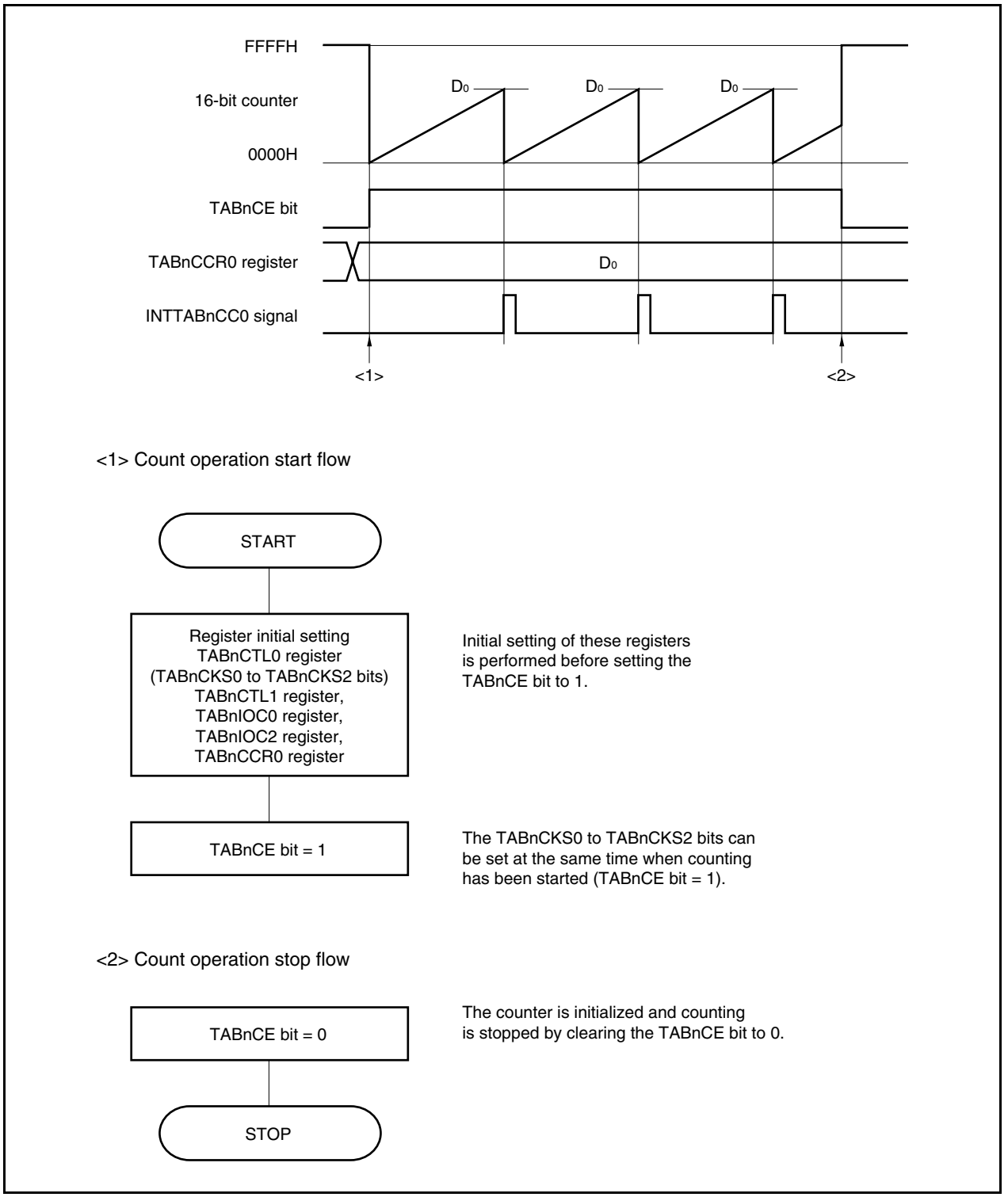
Therefore, mask the interrupt signal by using the interrupt mask flags (TABnCCMK1 to TABnCCMK3).

Caution When an external clock is used as the count clock, the external clock can be input only from the TIABn0 pin. At this time, set the TABnIOC1.TABnIS1 and TABnIOC1.TABnIS0 bits to 00 (capture trigger input (TIABn0 pin): no edge detection).

Remark The TABn I/O control register 1 (TABnIOC1) and TABn option register 0 (TABnOPT0) are not used in the external event count mode.

(1) External event count mode operation flow

Figure 8-12. Flow of Software Processing in External Event Count Mode



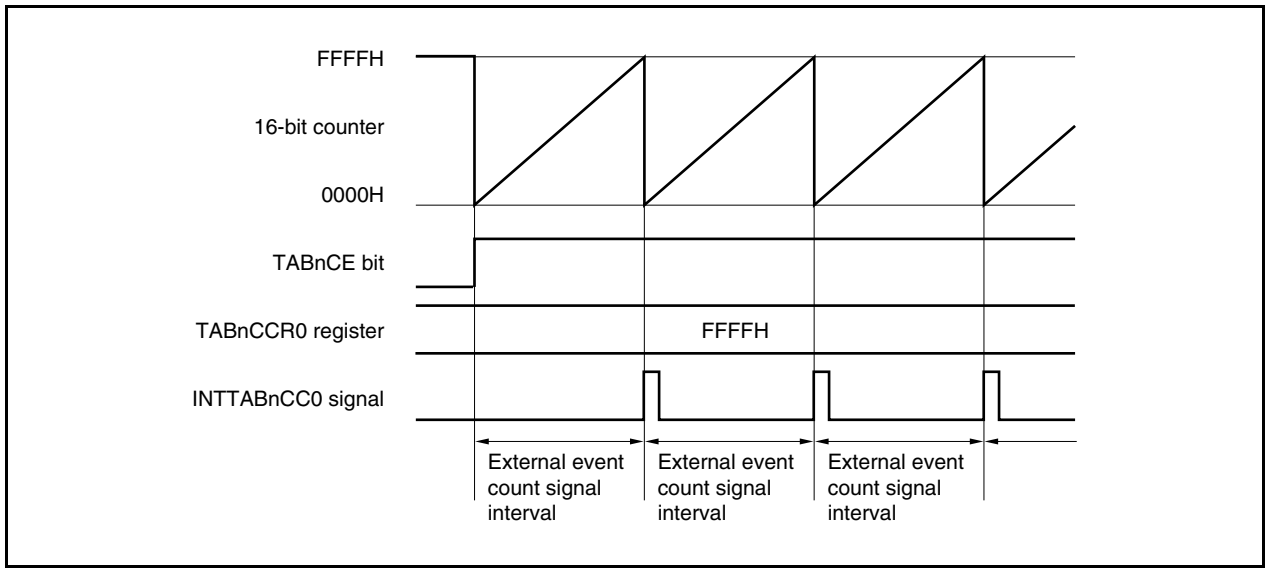
(2) Operation timing in external event count mode

Cautions 1. In the external event count mode, do not set the TABnCCR0 register to 0000H.

2. In the external event count mode, use of the timer output is disabled. If performing timer output using external event count input, set the interval timer mode, and select the operation enabled by the external event count input for the count clock (TABnCTL1.TABnMD2 to TABnCTL1.TABnMD0 bits = 000, TABnCTL1.TABnEEE bit = 1).

(a) Operation if TABnCCR0 register is set to FFFFH

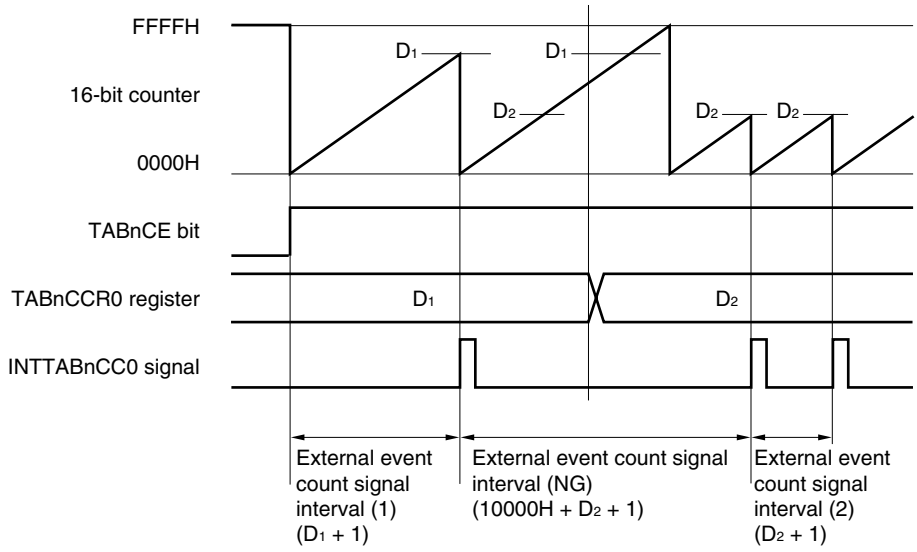
If the TABnCCR0 register is set to FFFFH, the 16-bit counter counts to FFFFH each time the valid edge of the external event count signal has been detected. The 16-bit counter is cleared to 0000H in synchronization with the next count-up timing, and the INTTABnCC0 signal is generated. At this time, the TABnOPT0.TABnOVF bit is not set.



(b) Notes on rewriting the TABnCCR0 register

To change the value of the TABnCCR0 register to a smaller value, stop counting once and then change the set value.

If the value of the TABnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.



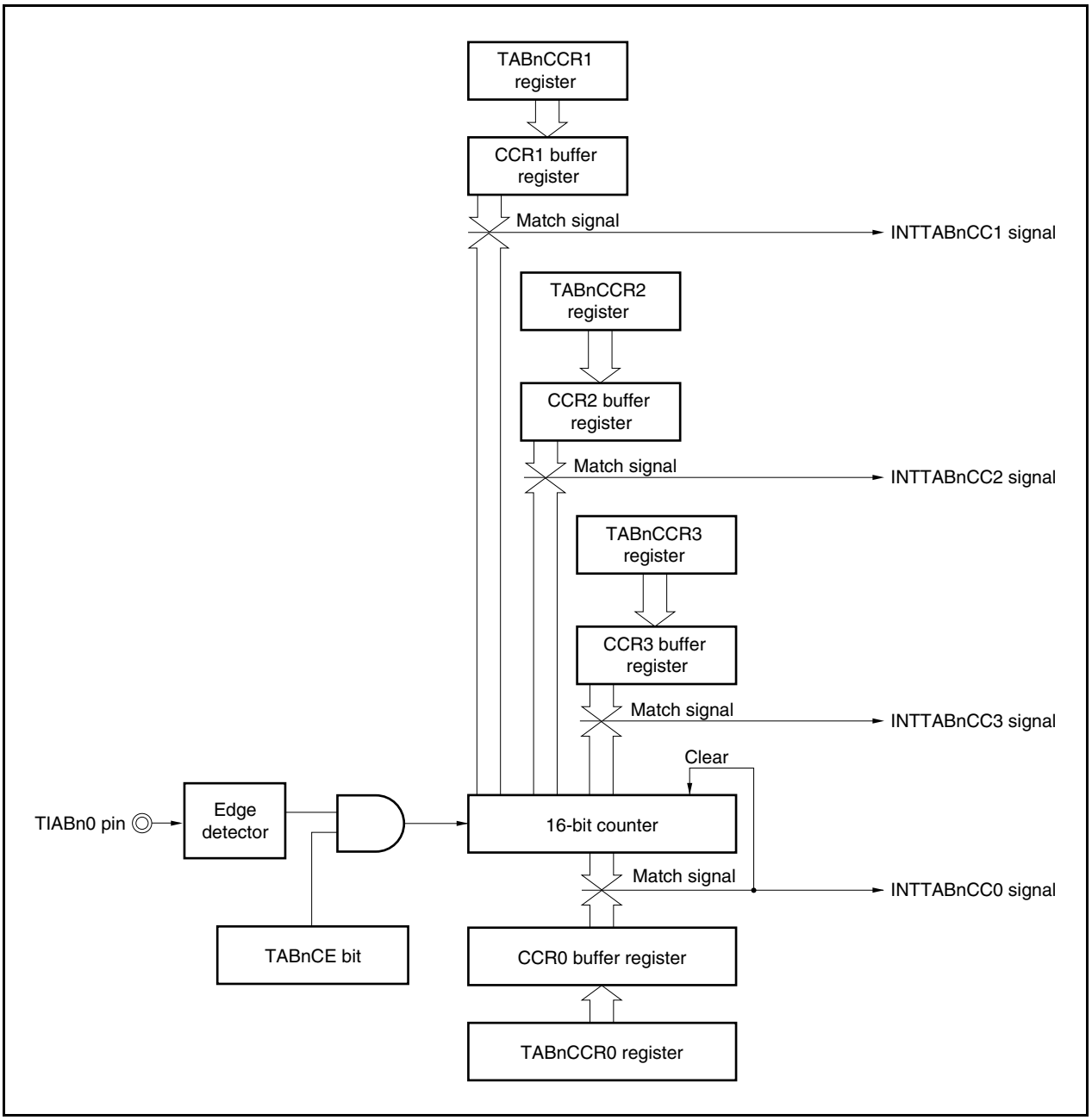
If the value of the TABnCCR0 register is changed from D_1 to D_2 while the count value is greater than D_2 but less than D_1, the count value is transferred to the CCR0 buffer register as soon as the TABnCCR0 register has been rewritten. Consequently, the value that is compared with the 16-bit counter is D_2.

Because the count value has already exceeded D_2, however, the 16-bit counter counts up to FFFFH, overflows, and then counts up again from 0000H. When the count value matches D_2, the INTTABnCC0 signal is generated.

Therefore, the INTTABnCC0 signal may not be generated at the valid edge count of "(D_1 + 1) times" or "(D_2 + 1) times" originally expected, but may be generated at the valid edge count of "(10000H + D_2 + 1) times".

(c) Operation of TABnCCR1 to TABnCCR3 registers

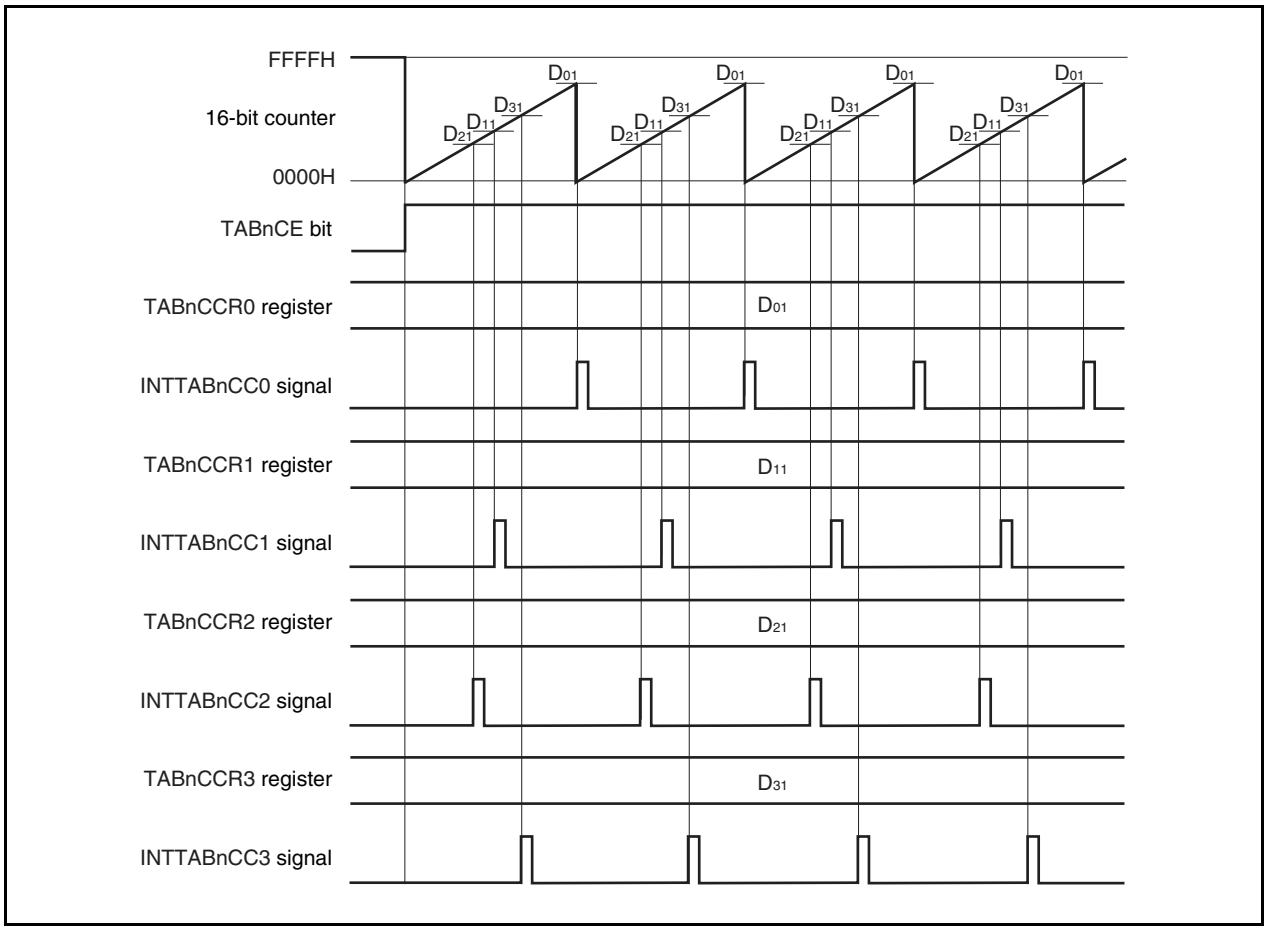
Figure 8-13. Configuration of TABnCCR1 to TABnCCR3 Registers



If the set value of the TABnCCRk register is smaller than the set value of the TABnCCR0 register, the INTTABnCCk signal is generated once per cycle.

Remark k = 1 to 3

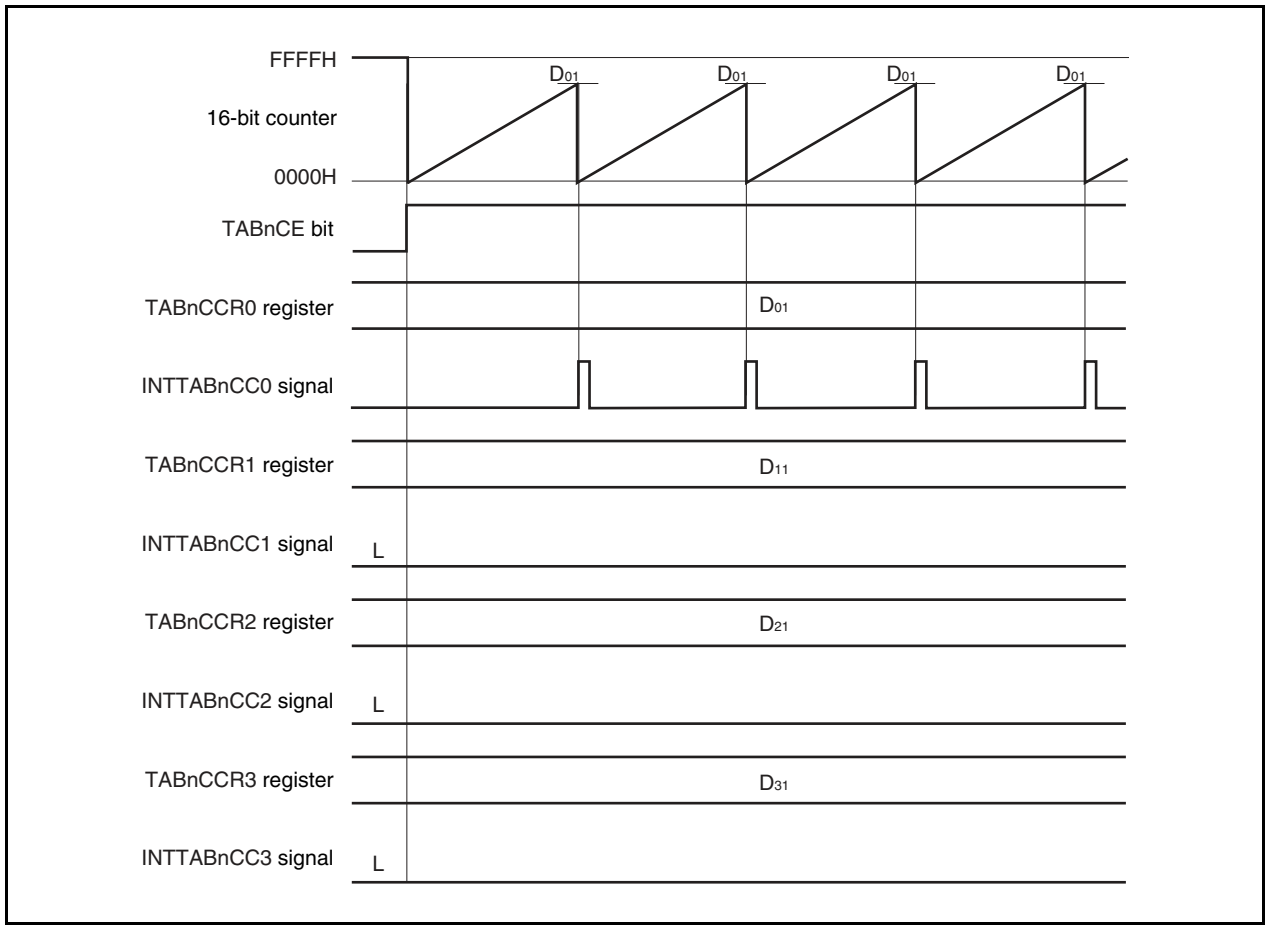
Figure 8-14. Timing Chart When $D_{01} \geq D_{k1}$



If the set value of the TABnCCRk register is greater than the set value of the TABnCCR0 register, the INTTABnCCk signal is not generated because the count value of the 16-bit counter and the value of the TABnCCRk register do not match.

Remark k = 1 to 3

Figure 8-15. Timing Chart When $D_{01} < D_{k1}$



8.5.3 External trigger pulse output mode (TABnMD2 to TABnMD0 bits = 010)

In the external trigger pulse output mode, 16-bit timer/event counter AB waits for a trigger when the TABnCTL0.TABnCE bit is set to 1. When the valid edge of an external trigger input signal is detected, 16-bit timer/event counter AB starts counting, and outputs a PWM waveform from the TOABn1 to TOABn3 pins.

Pulses can also be output by generating a software trigger instead of using the external trigger. When using a software trigger, a square wave that has one cycle of the PWM waveform as half its cycle can also be output from the TOABn0 pin.

Figure 8-16. Configuration in External Trigger Pulse Output Mode

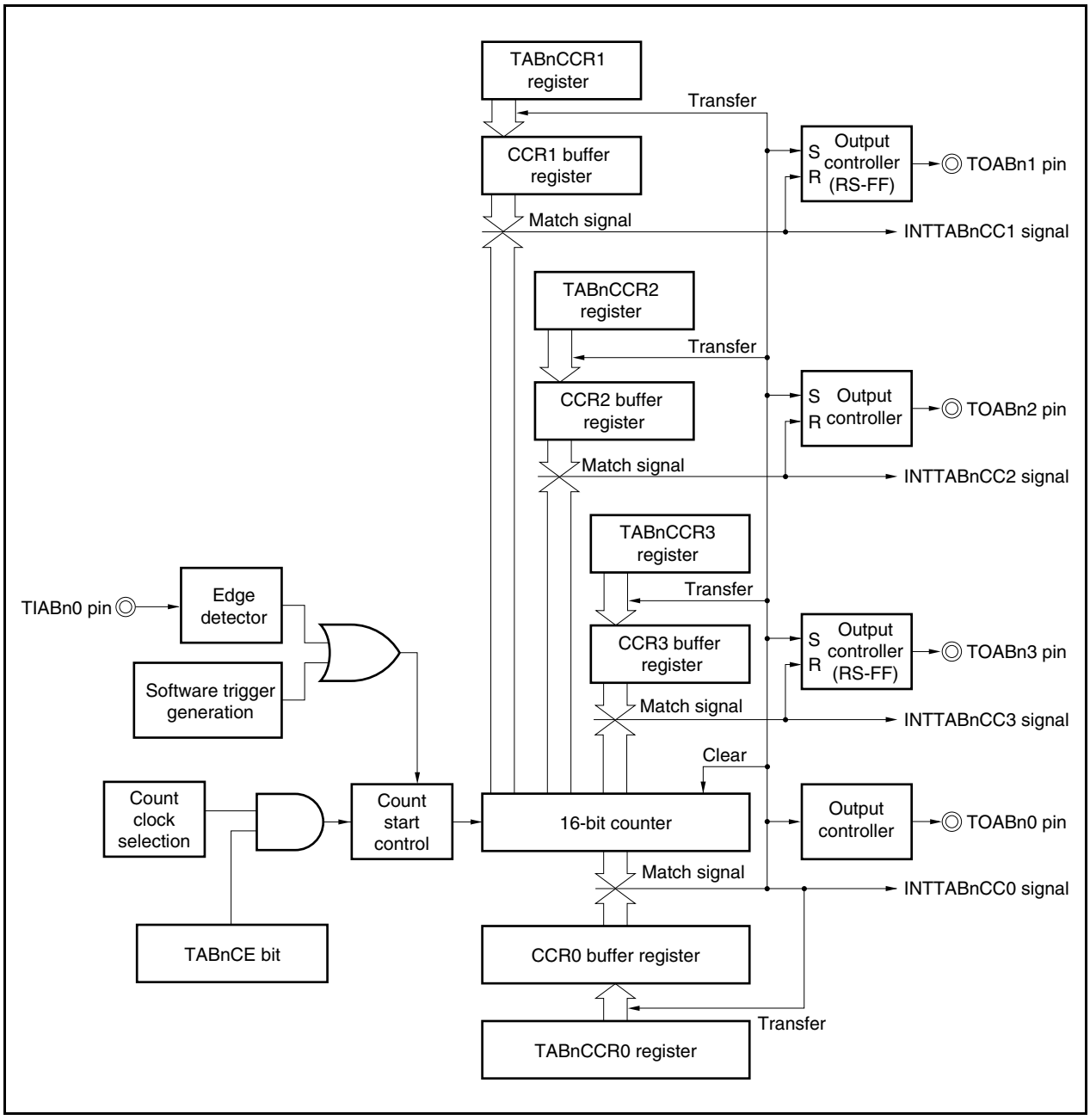
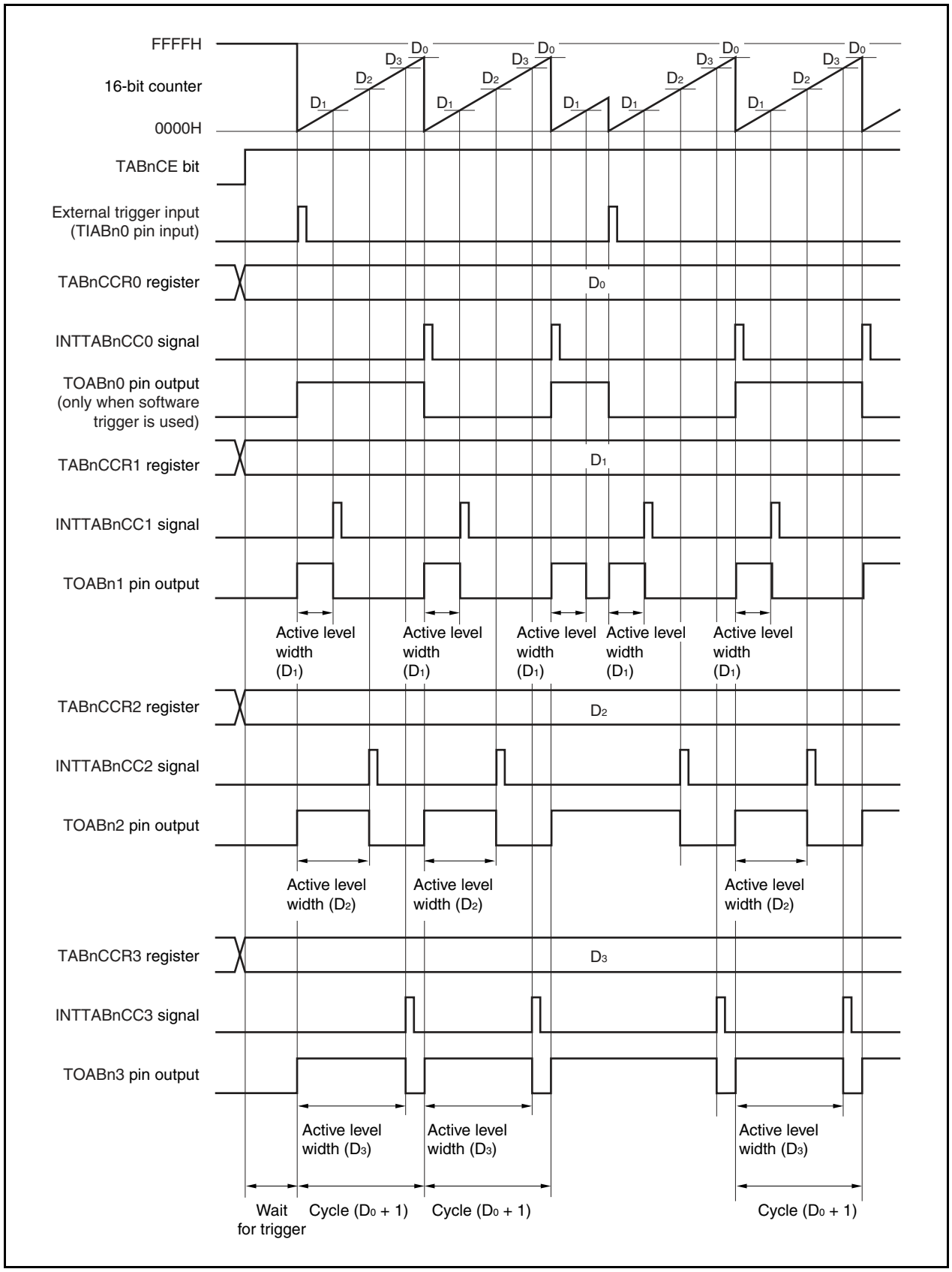


Figure 8-17. Basic Timing in External Trigger Pulse Output Mode



16-bit timer/event counter AB waits for a trigger when the TABnCE bit is set to 1. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting at the same time, and outputs a PWM waveform from the TOABnk pin. If the trigger is generated again while the counter is operating, the counter is cleared to 0000H and restarted. (The output of the TOABn0 pin is inverted. The TOABnk pin outputs a high-level regardless of the status (high/low) when a trigger is generated.)

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

Active level width = (Set value of TABnCCRk register) × Count clock cycle
Cycle = (Set value of TABnCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TABnCCRk register)/(Set value of TABnCCR0 register + 1)

The compare match request signal (INTTABnCC0) is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal (INTTABnCCk) is generated when the count value of the 16-bit counter matches the value of the CCRk buffer register.

The value set to the TABnCCRm register is transferred to the CCRm buffer register when the count value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

The valid edge of an external trigger input signal, or setting the software trigger (TABnCTL1.TABnEST bit) to 1 is used as the trigger.

Remark k = 1 to 3,
m = 0 to 3

Figure 8-18. Setting of Registers in External Trigger Pulse Output Mode (1/3)

(a) TABn control register 0 (TABnCTL0)

	TABnCE					TABnCKS2	TABnCKS1	TABnCKS0
TABnCTL0	0/1	0	0	0	0	0/1	0/1	0/1

Select count clock

0: Stop counting
1: Enable counting

(b) TABn control register 1 (TABnCTL1)

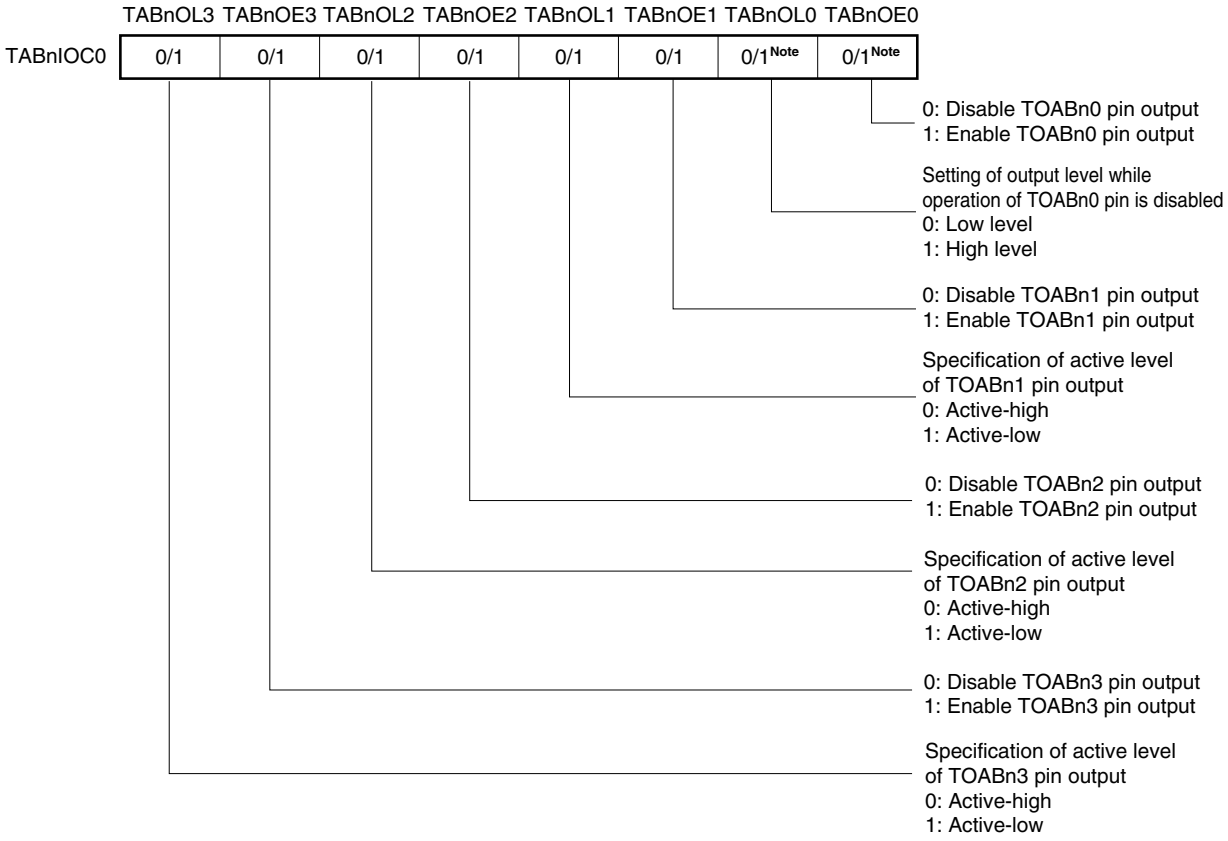
	TABnSYE	TABnEST	TABnEEE			TABnMD2	TABnMD1	TABnMD0
TABnCTL1	0	0/1	0	0	0	0	1	0

0, 1, 0:
External trigger pulse output mode

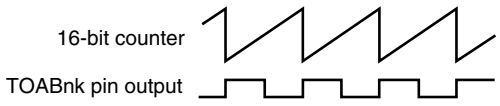
Generate software trigger when 1 is written

Figure 8-18. Setting of Registers in External Trigger Pulse Output Mode (2/3)

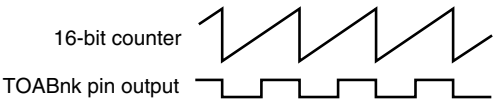
(c) TABn I/O control register 0 (TABnIOC0)



- When TABnOLk bit = 0



- When TABnOLk bit = 1



(d) TABn I/O control register 2 (TABnIOC2)

(e) TABn counter read buffer register (TABnCNT)

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

Note Clear this bit to 0 when the TOABn0 pin is not used in the external trigger pulse output mode.

Figure 8-18. Setting of Registers in External Trigger Pulse Output Mode (3/3)

(f) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)

If D_0 is set to the TABnCCR0 register, D_1 to the TABnCCR1 register, D_2 to the TABnCCR2 register, and D_3, to the TABnCCR3 register, the cycle and active level of the PWM waveform are as follows.

Cycle = $(D_0 + 1) \times$ Count clock cycle
TOABn1 pin PWM waveform active level width = $D_1 \times$ Count clock cycle
TOABn2 pin PWM waveform active level width = $D_2 \times$ Count clock cycle
TOABn3 pin PWM waveform active level width = $D_3 \times$ Count clock cycle

Remarks 1. TABn I/O control register 1 (TABnIOC1) and TABn option register 0 (TABnOPT0) are not used in the external trigger pulse output mode.

2. Updating TABn capture/compare register 2 (TABnCCR2) and TABn capture/compare register 3 (TABnCCR3) is validated by writing TABn capture/compare register 1 (TABnCCR1).

(1) Operation flow in external trigger pulse output mode

Figure 8-19. Software Processing Flow in External Trigger Pulse Output Mode (1/2)

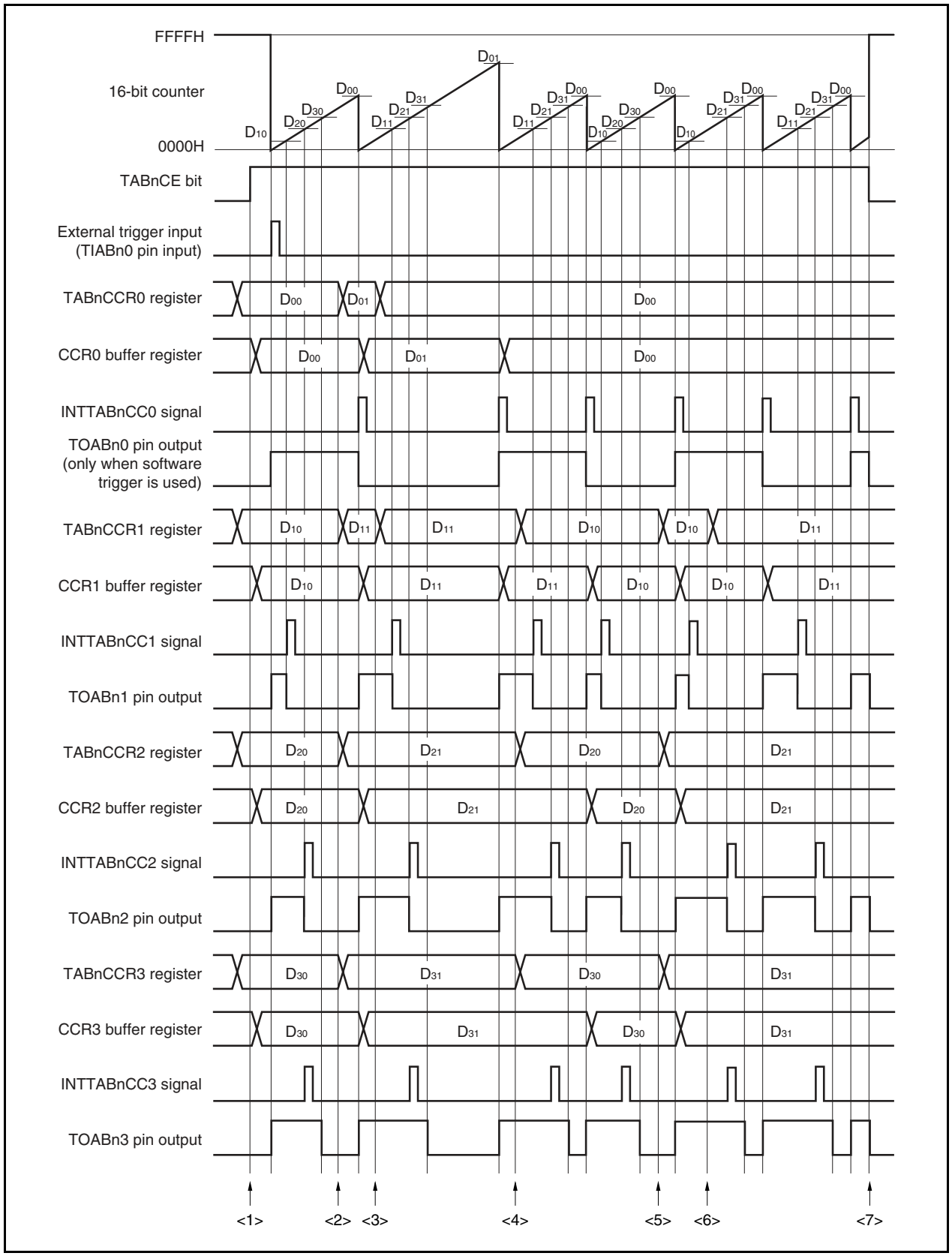
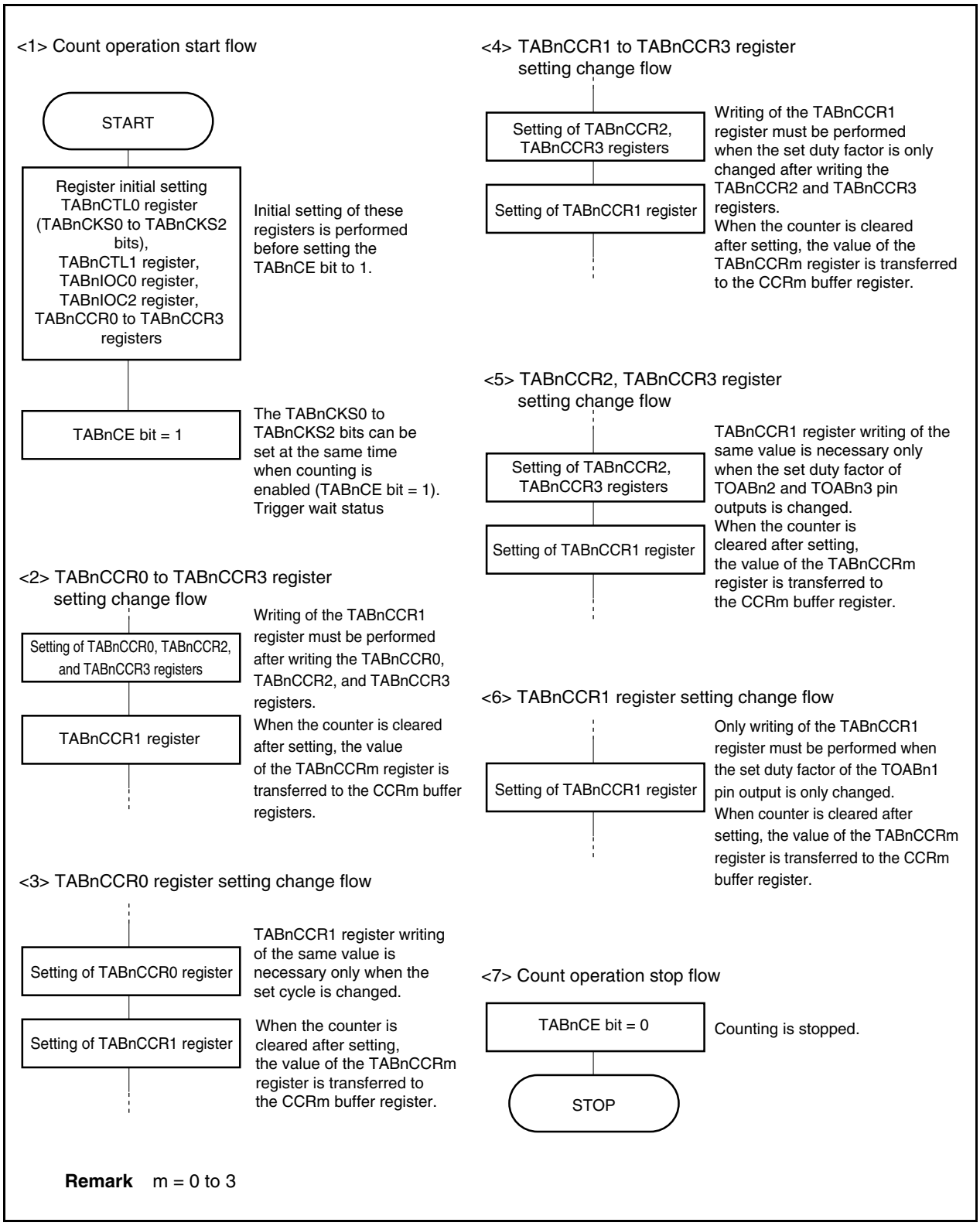


Figure 8-19. Software Processing Flow in External Trigger Pulse Output Mode (2/2)



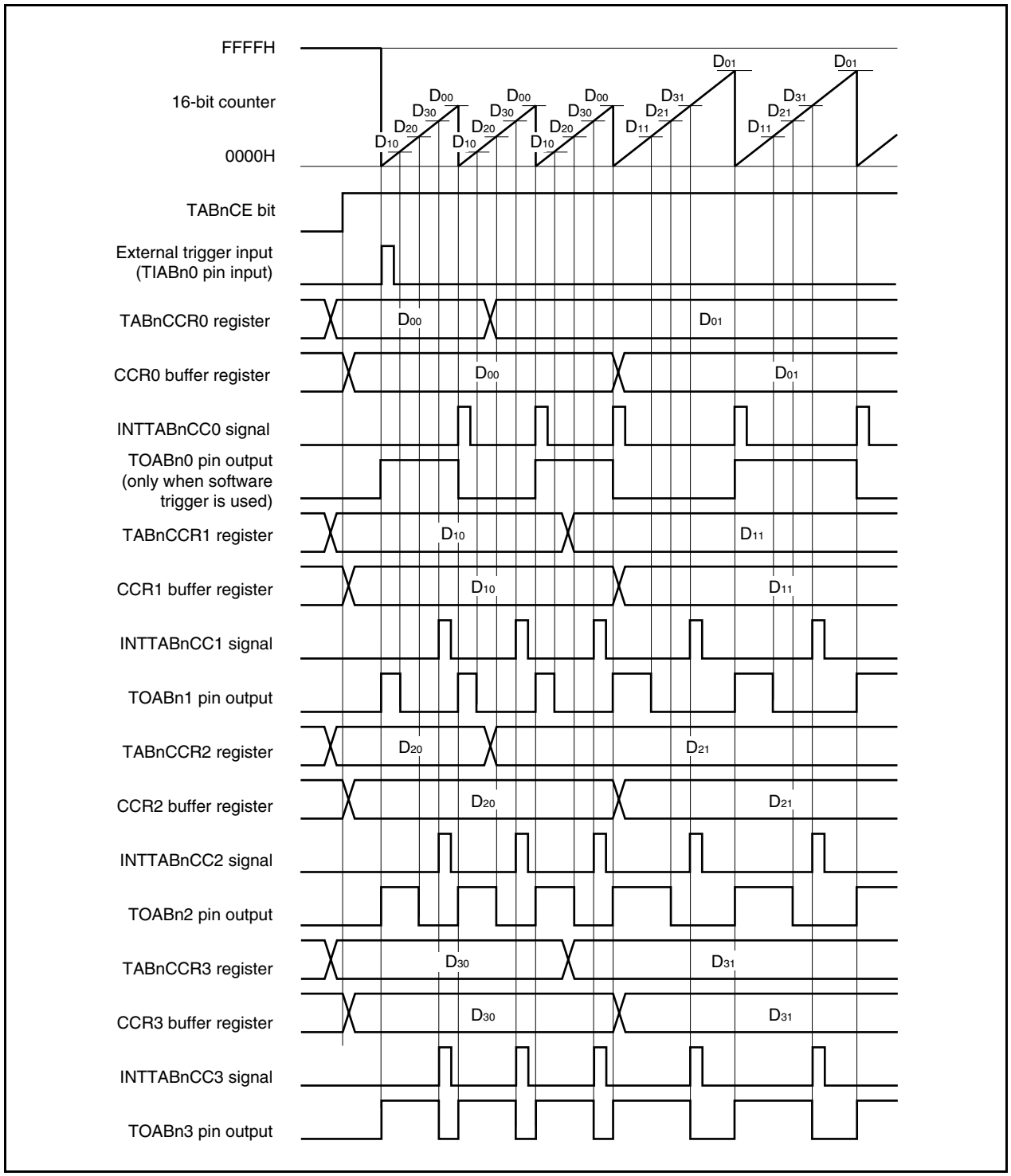
Remark m = 0 to 3

(2) External trigger pulse output mode operation timing

(a) Note on changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TABnCCR1 register last.

Rewrite the TABnCCRk register after writing the TABnCCR1 register after the INTTABnCC0 signal is detected.



In order to transfer data from the TABnCCRm register to the CCRm buffer register, the TABnCCR1 register must be written.

To change both the cycle and active level width of the PWM waveform at this time, first set the cycle to the TABnCCR0 register, set the active level width to the TABnCCR2 and TABnCCR3 registers, and then set an active level to the TABnCCR1 register.

To change only the cycle of the PWM waveform, first set the cycle to the TABnCCR0 register, and then write the same value to the TABnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform, first set an active level to the TABnCCR2 and TABnCCR3 registers and then set an active level to the TABnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform output by the TOABn1 pin, only the TABnCCR1 register has to be set.

To change only the active level width (duty factor) of the PWM waveform output by the TOABn2 and TOABn3 pins, first set an active level width to the TABnCCR2 and TABnCCR3 registers, and then write the same value to the TABnCCR1 register.

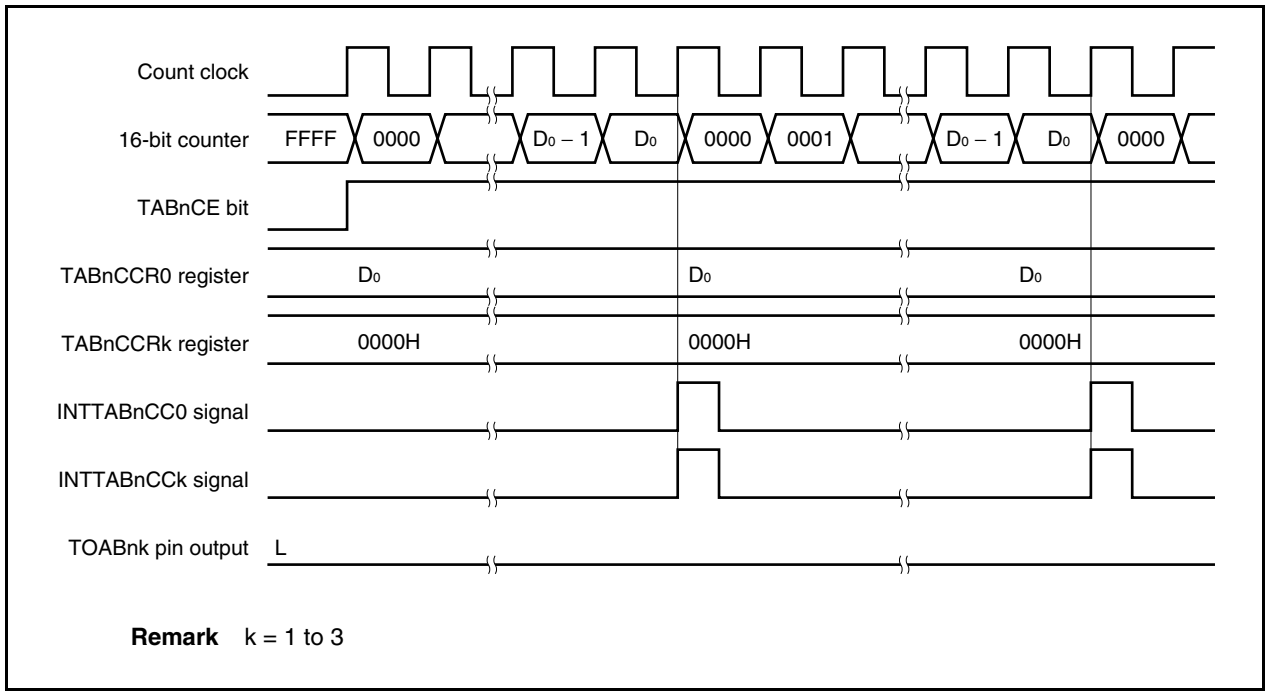
After data is written to the TABnCCR1 register, the value written to the TABnCCRm register is transferred to the CCRm buffer register in synchronization with clearing of the 16-bit counter, and is used as the value compared with the 16-bit counter.

To write the TABnCCR0 to TABnCCR3 registers again after writing the TABnCCR1 register once, do so after the INTTABnCC0 signal is generated. Otherwise, the value of the CCRm buffer register may become undefined because timing of transferring data from the TABnCCRm register to the CCRm buffer register conflicts with writing the TABnCCRm register.

Remark m = 0 to 3

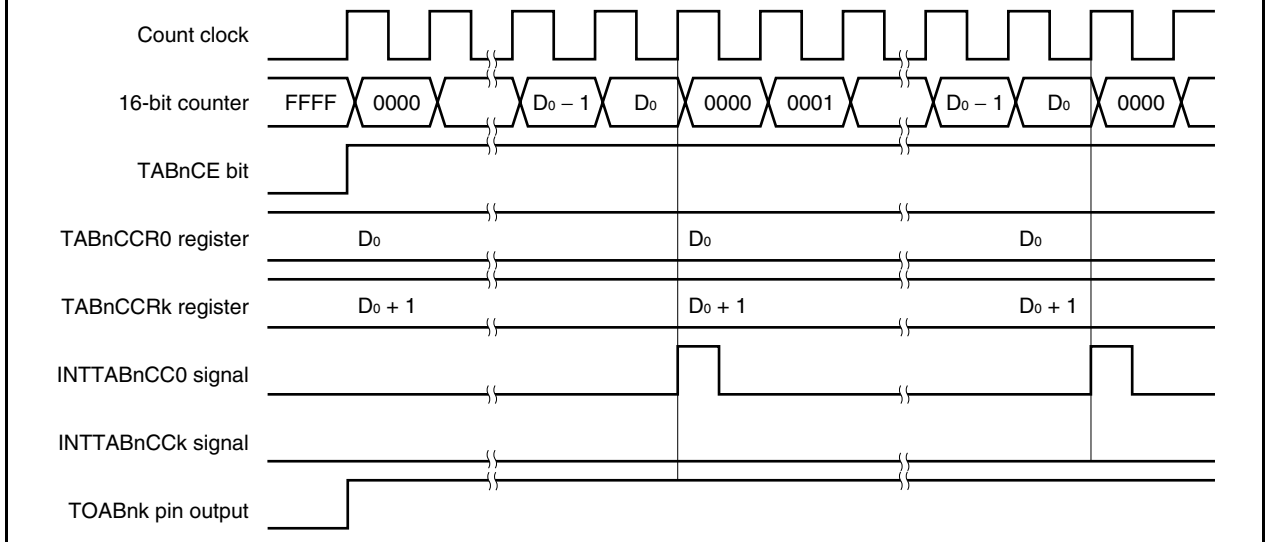
(b) 0%/100% output of PWM waveform

To output a 0% waveform, set the TABnCCRk register to 0000H. If the set value of the TABnCCR0 register is FFFFH, the INTTABnCCk signal is generated periodically.



Remark k = 1 to 3

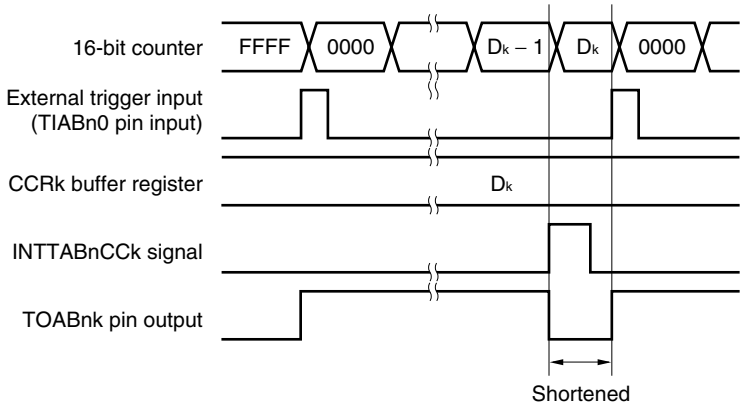
To output a 100% waveform, set a value of (set value of TABnCCR0 register + 1) to the TABnCCRk register. If the set value of the TABnCCR0 register is FFFFH, 100% output cannot be produced.



Remark k = 1 to 3

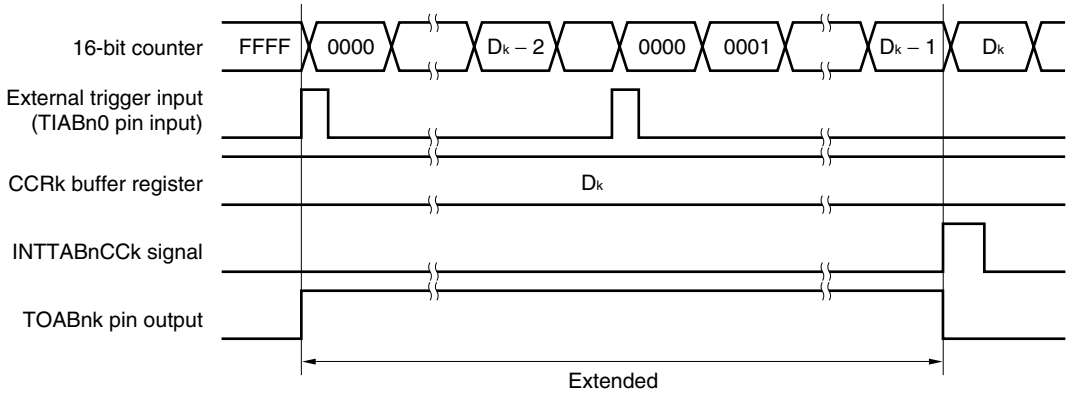
(c) Conflict between trigger detection and match with CCRk buffer register

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



Remark k = 1 to 3

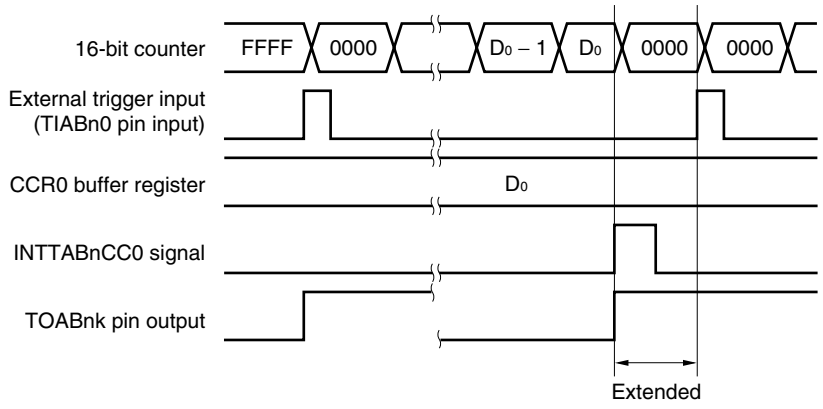
If the trigger is detected immediately before the INTTABnCCk signal is generated, the INTTABnCCk signal is not generated, and the 16-bit counter is cleared to 0000H and continues counting. The output signal of the TOABnk pin remains active. Consequently, the active period of the PWM waveform is extended.



Remark k = 1 to 3

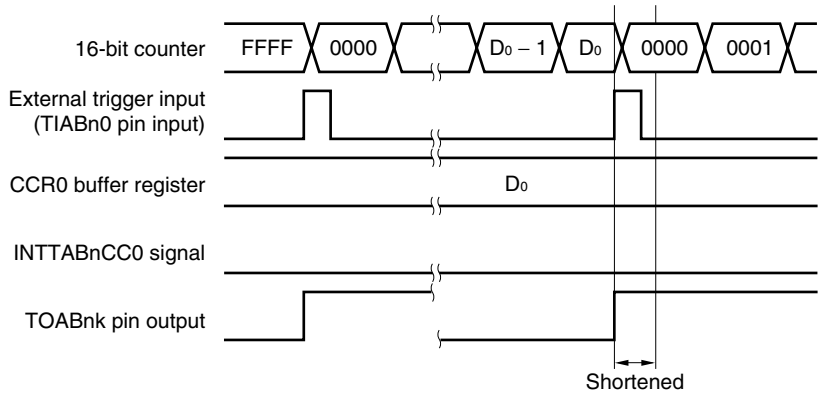
(d) Conflict between trigger detection and match with CCR0 buffer register

If the trigger is detected immediately after the INTTABnCC0 signal is generated, the 16-bit counter is cleared to 0000H and continues counting up. Therefore, the active period of the TOABnk pin is extended by time from generation of the INTTABnCC0 signal to trigger detection.



Remark k = 1 to 3

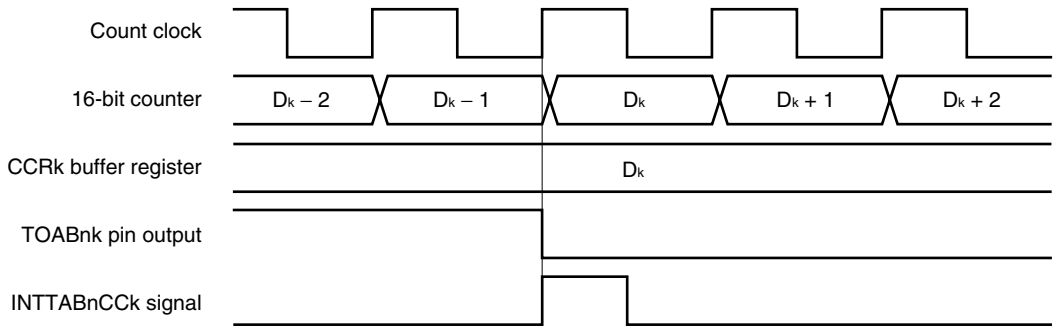
If the trigger is detected immediately before the INTTABnCC0 signal is generated, the INTTABnCC0 signal is not generated. The 16-bit counter is cleared to 0000H, the TOABnk pin is asserted, and the counter continues counting. Consequently, the inactive period of the PWM waveform is shortened.



Remark k = 1 to 3

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

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



Remark k = 1 to 3

Usually, the INTTABnCCk signal is generated in synchronization with the next count up after the count value of the 16-bit counter matches the value of the CCRk buffer register.

In the external trigger pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the timing of changing the output signal of the TOABnk pin.

8.5.4 One-shot pulse output mode (TABnMD2 to TABnMD0 bits = 011)

In the one-shot pulse output mode, 16-bit timer/event counter AB waits for a trigger when the TABnCTL0.TABnCE bit is set to 1. When the valid edge of an external trigger input is detected, 16-bit timer/event counter AB starts counting, and outputs a one-shot pulse from the TOABn1 to TOABn3 pins.

Instead of the external trigger, a software trigger can also be generated to output the pulse. When the software trigger is used, the TOABn0 pin outputs the active level while the 16-bit counter is counting, and the inactive level when the counter is stopped (waiting for a trigger).

Figure 8-20. Configuration in One-Shot Pulse Output Mode

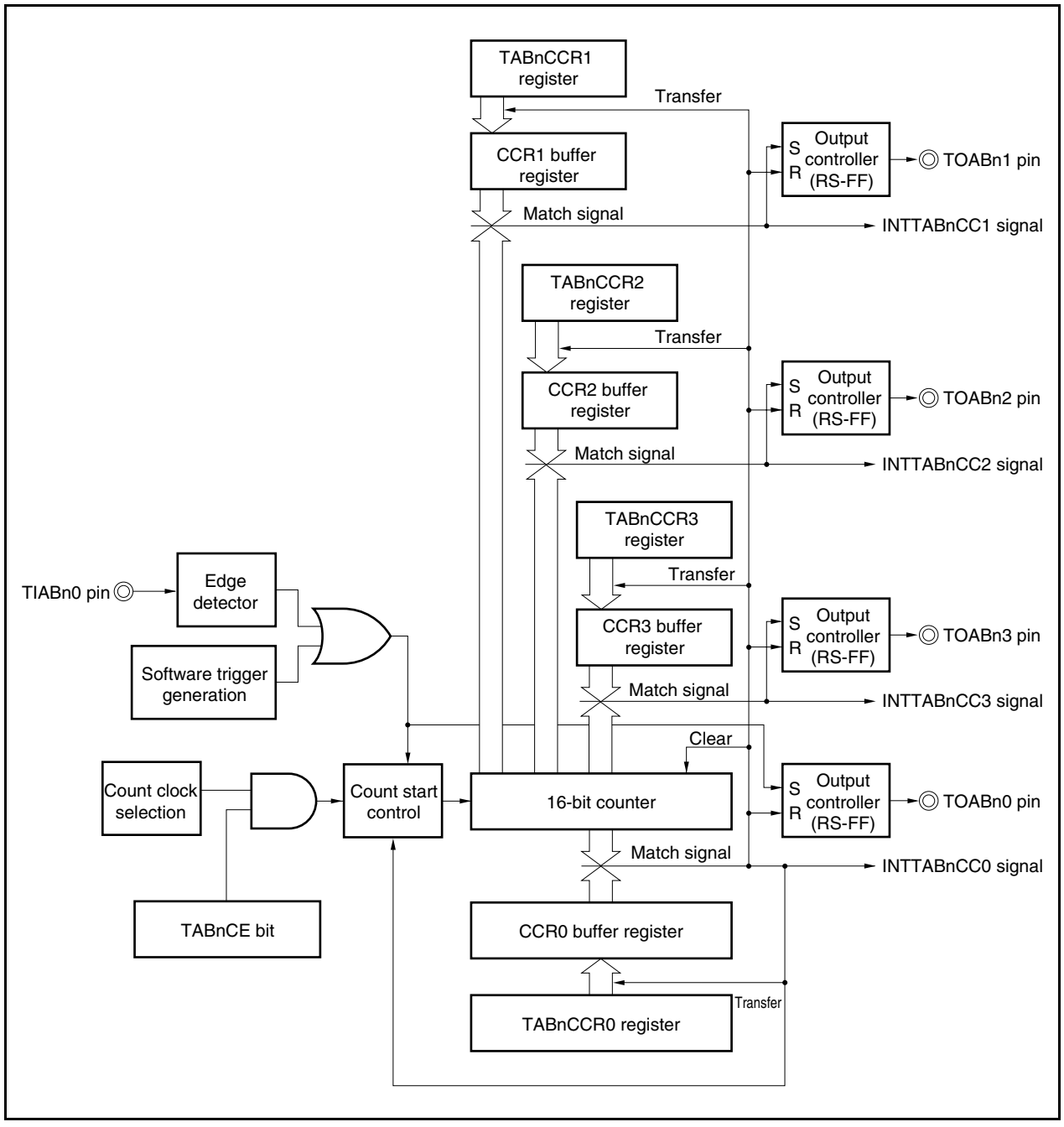
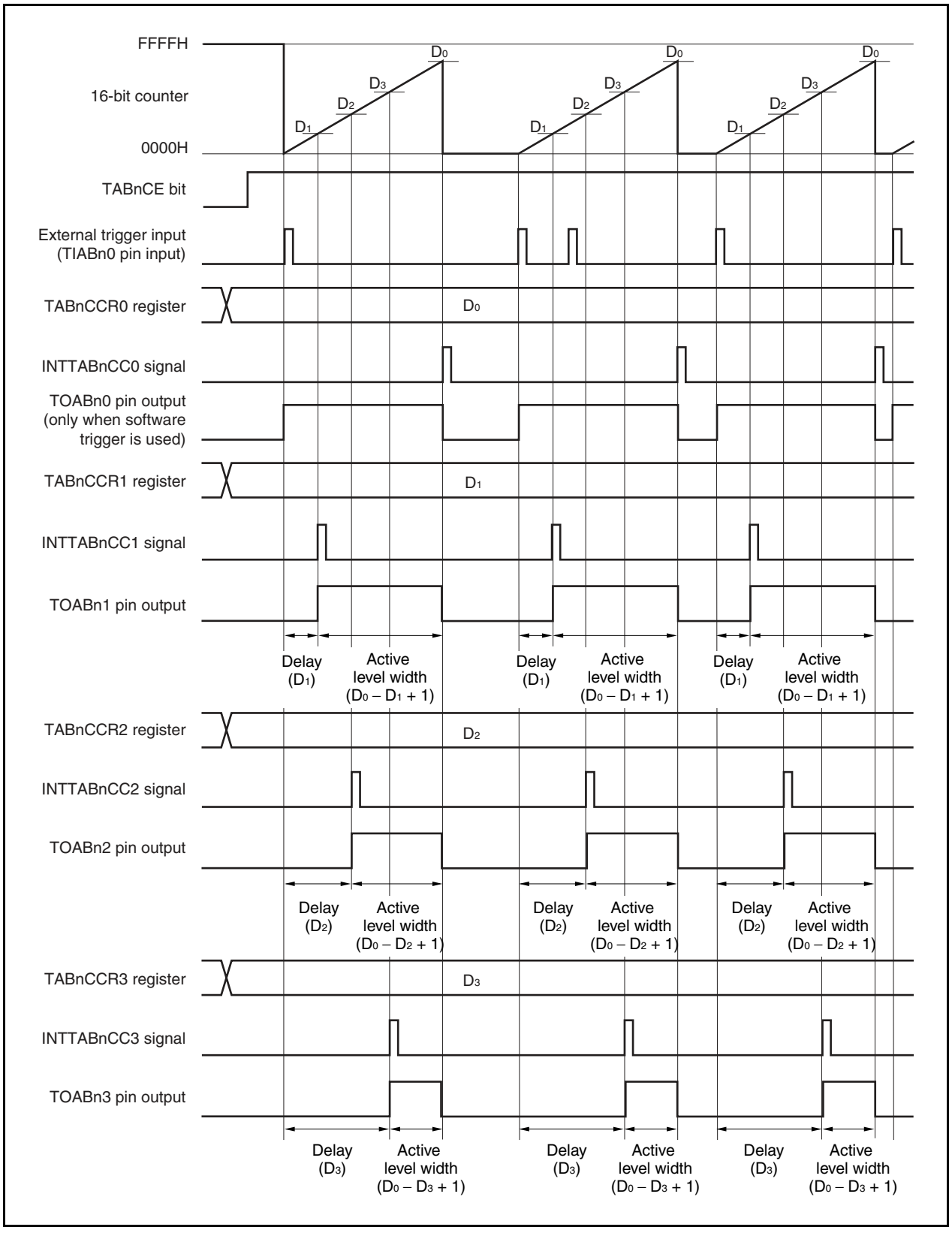


Figure 8-21. Basic Timing in One-Shot Pulse Output Mode



When the TABnCE bit is set to 1, 16-bit timer/event counter AB waits for a trigger. When the trigger is generated, the 16-bit counter is cleared from FFFFH to 0000H, starts counting, and outputs a one-shot pulse from the TOABnk pin. After the one-shot pulse is output, the 16-bit counter is set to FFFFH, stops counting, 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 TABnCCRk register) × Count clock cycle

Active level width = (Set value of TABnCCR0 register − Set value of TABnCCRk register + 1) × Count clock cycle

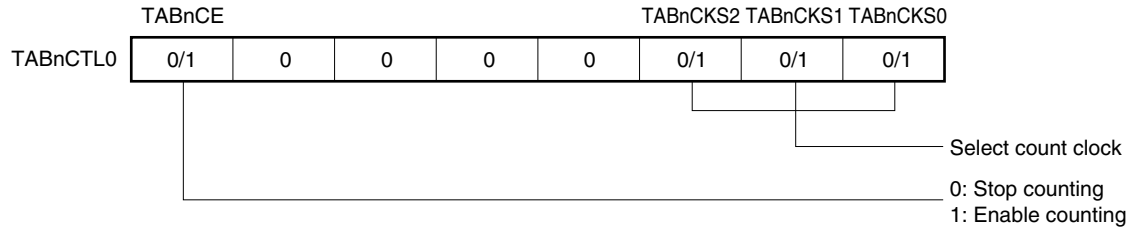
The compare match interrupt request signal INTTABnCC0 is generated when the 16-bit counter counts after its count value matches the value of the CCR0 buffer register. The compare match interrupt request signal (INTTABnCCk) is generated when the count value of the 16-bit counter matches the value of the CCRk buffer register.

The valid edge of an external trigger input or setting the software trigger (TABnCTL1.TABnEST bit) to 1 is used as the trigger.

Remark k = 1 to 3

Figure 8-22. Register Setting for Operation in One-Shot Pulse Output Mode (1/3)

(a) TABn control register 0 (TABnCTL0)



(b) TABn control register 1 (TABnCTL1)

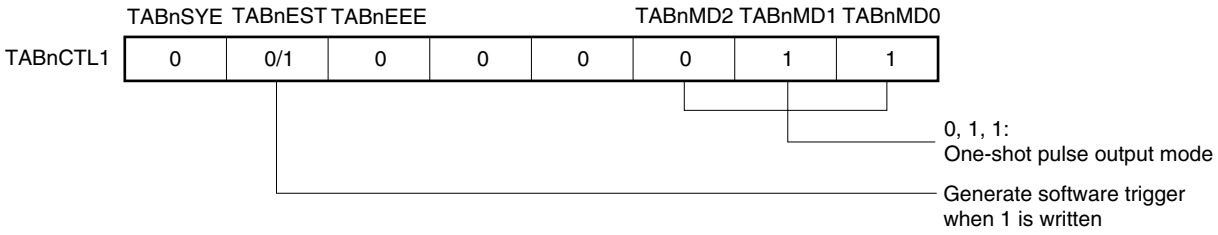
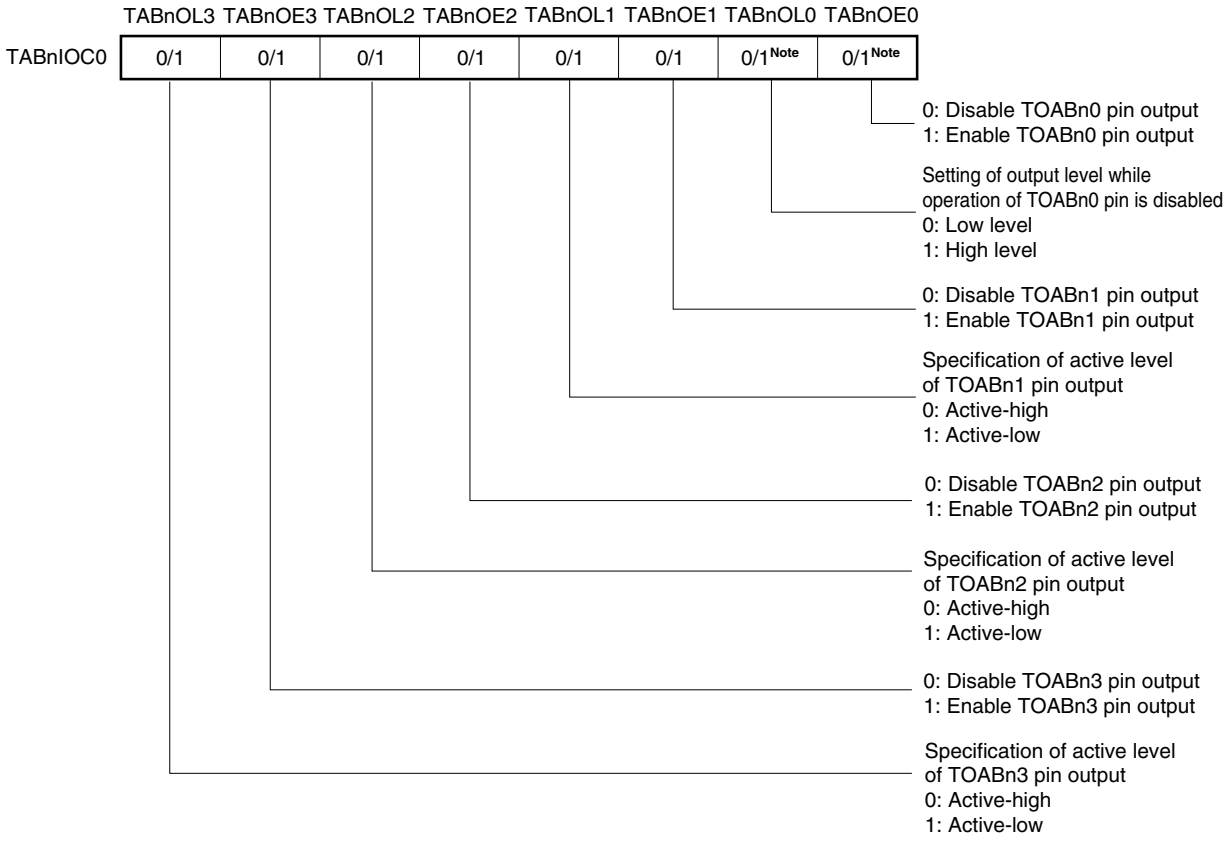
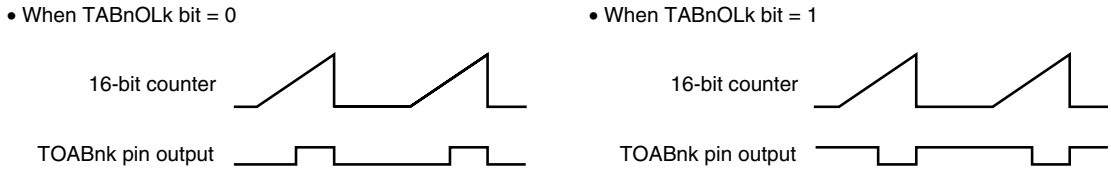


Figure 8-22. Register Setting for Operation in One-Shot Pulse Output Mode (2/3)

(c) TABn I/O control register 0 (TABnIOC0)





(d) TABn I/O control register 2 (TABnIOC2)

(e) TABn counter read buffer register (TABnCNT)

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

Note Clear this bit to 0 when the TOABn0 pin is not used in the one-shot pulse output mode.

Figure 8-22. Register Setting for Operation in One-Shot Pulse Output Mode (3/3)

(f) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)

If D_0 is set to the TABnCCR0 register and D_k to the TABnCCRk register, the active level width and output delay period of the one-shot pulse are as follows.

Active level width = $(D_0 - D_k + 1) \times$ Count clock cycle
Output delay period = $(D_k) \times$ Count clock cycle

Caution One-shot pulses are not output even in the one-shot pulse output mode, if the set value in the TABnCCRk register is greater than that value in the TABnCCR0 register.

Remarks 1. TABn I/O control register 1 (TABnIOC1) and TABn option register 0 (TABnOPT0) are not used in the one-shot pulse output mode.

2. k = 1 to 3

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

Figure 8-23. Software Processing Flow in One-Shot Pulse Output Mode (1/2)

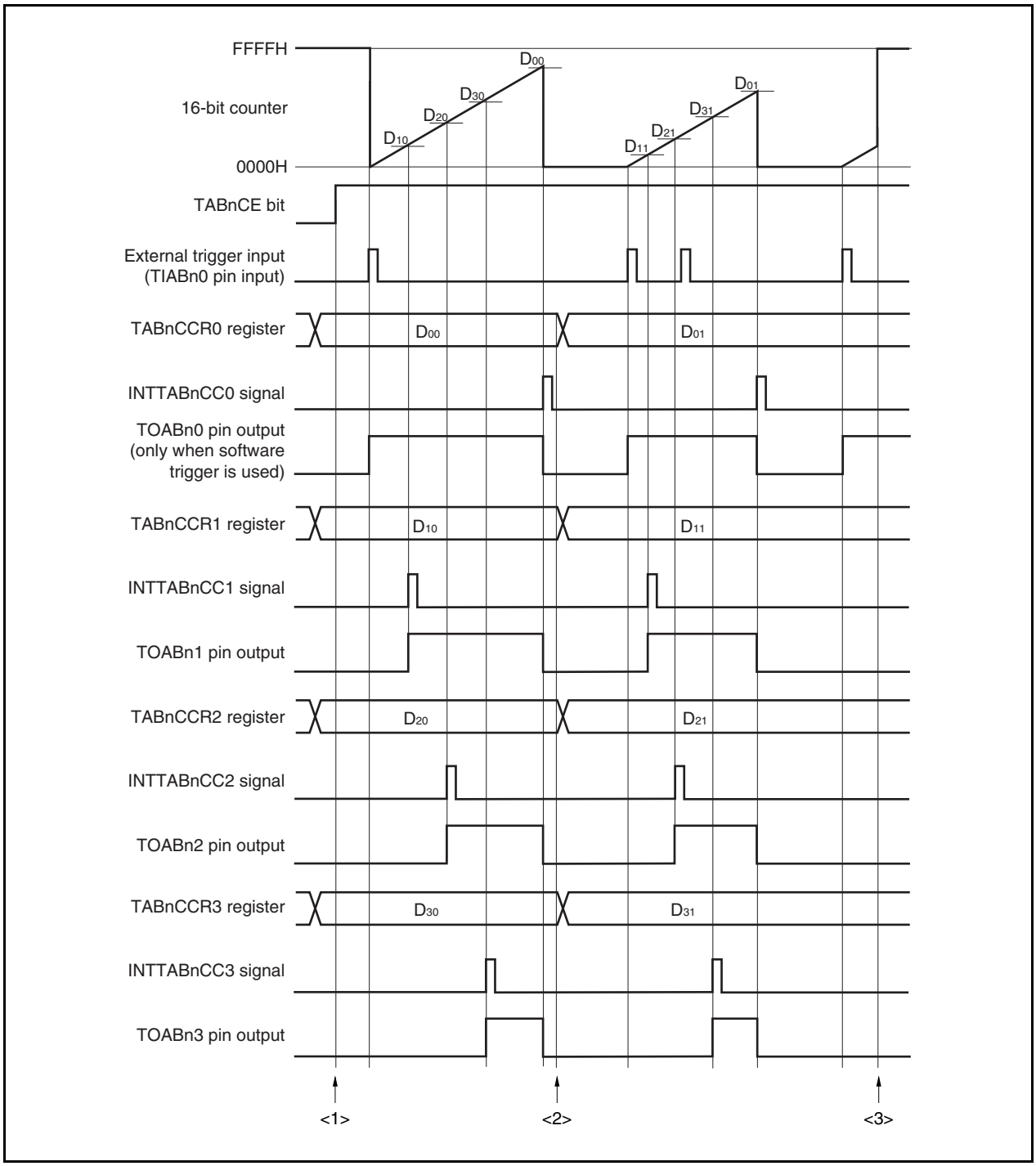
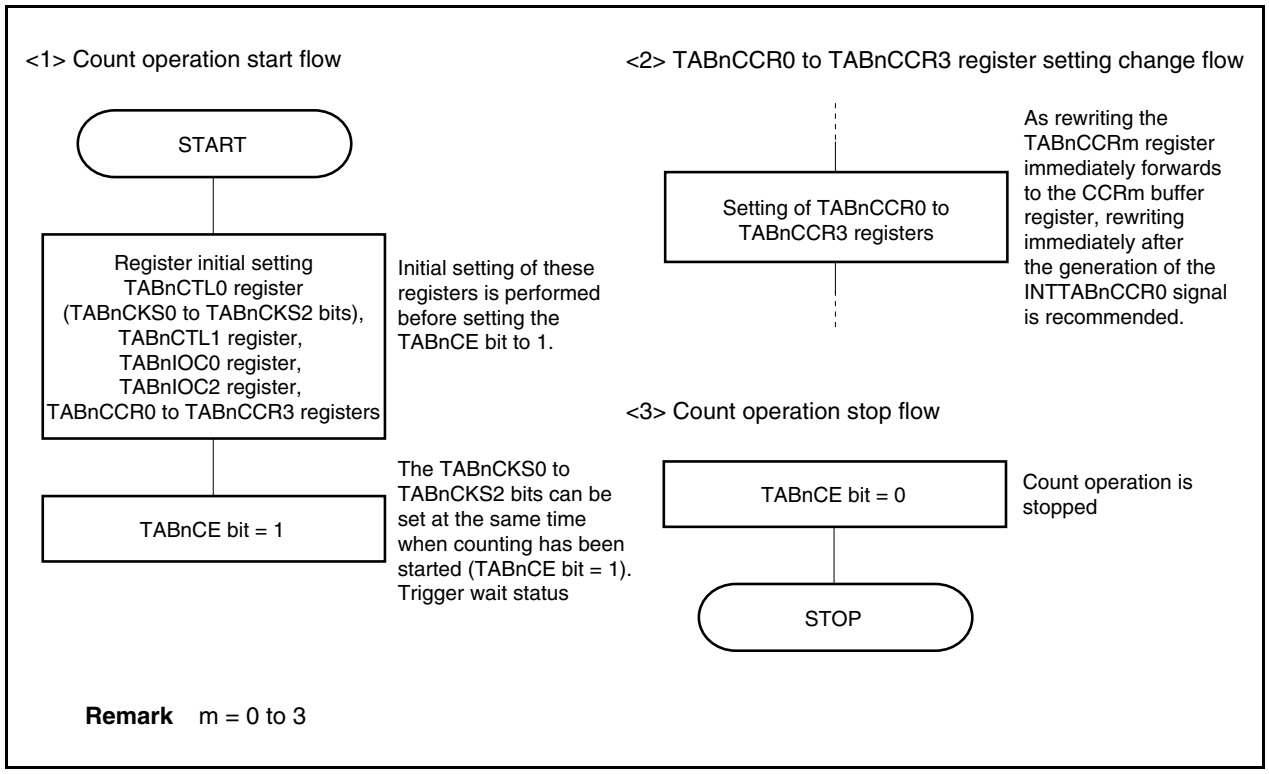


Figure 8-23. Software Processing Flow in One-Shot Pulse Output Mode (2/2)



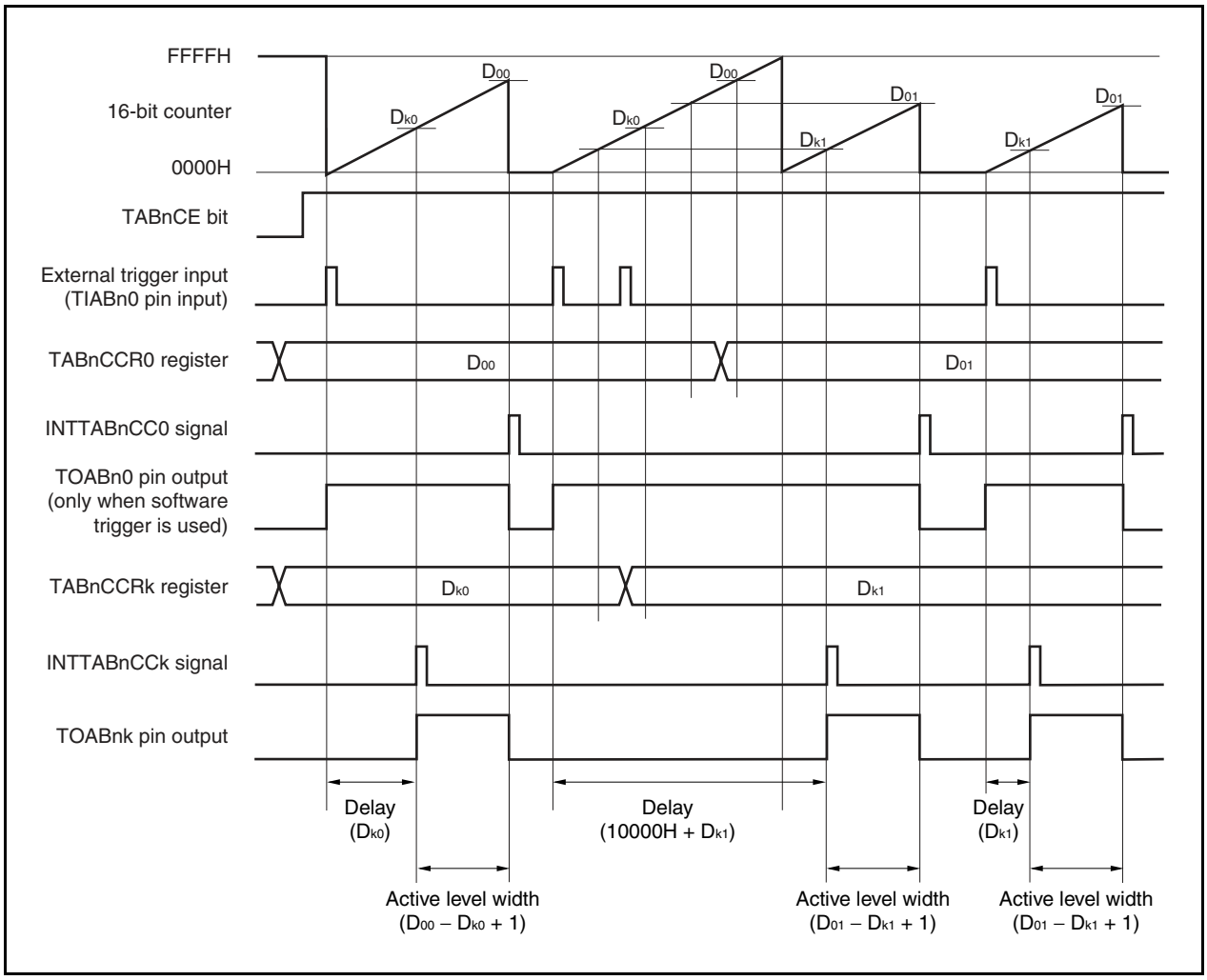
Remark m = 0 to 3

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

(a) Note on rewriting TABnCCRm register

To change the set value of the TABnCCRm register to a smaller value, stop counting once, and then change the set value.

If the value of the TABnCCR0 register is rewritten to a smaller value during counting, the 16-bit counter may overflow.



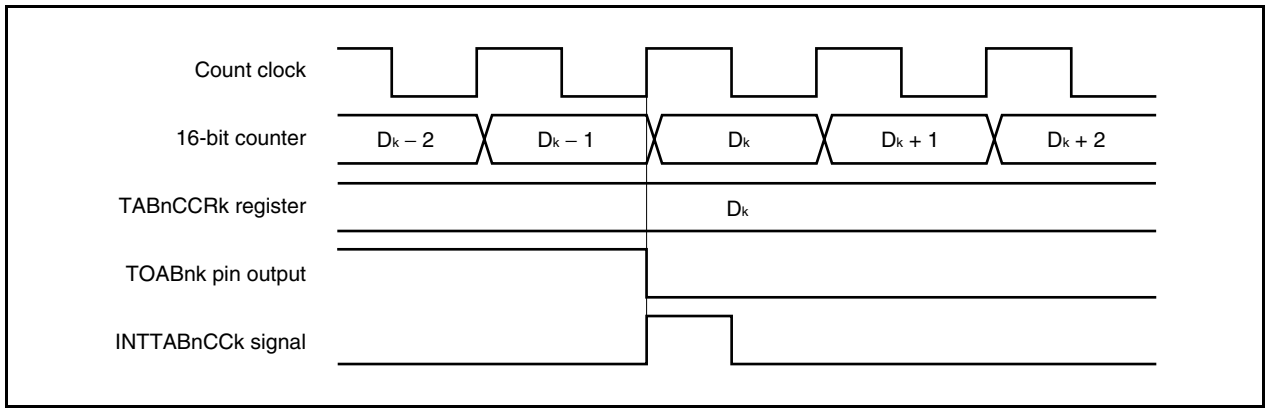
When the TABnCCR0 register is rewritten from D_{00} to D_{01} and the TABnCCRk register from D_{k0} to D_{k1} where $D_{00} > D_{01}$ and $D_{k0} > D_{k1}$, if the TABnCCRk register is rewritten when the count value of the 16-bit counter is greater than D_{k1} and less than D_{k0} and if the TABnCCR0 register is rewritten when the count value is greater than D_{01} and less than D_{00}, each set value is reflected as soon as the register has been rewritten and compared with the count value. The counter counts up to FFFFH and then counts up again from 0000H. When the count value matches D_{k1}, the counter generates the INTTABnCCk signal and asserts the TOABnk pin. When the count value matches D_{01}, the counter generates the INTTABnCC0 signal, deasserts the TOABnk pin, and stops counting.

Therefore, the counter may output a pulse with a delay period or active period different from that of the one-shot pulse that is originally expected.

Remark k = 1 to 3

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

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



Usually, the INTTABnCCk signal is generated when the 16-bit counter counts up next time after its count value matches the value of the TABnCCRk register.

In the one-shot pulse output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the TOABnk pin.

Remark k = 1 to 3

8.5.5 PWM output mode (TABnMD2 to TABnMD0 bits = 100)

In the PWM output mode, a PWM waveform is output from the TOABn1 to TOABn3 pins when the TABnCTL0.TABnCE bit is set to 1.

In addition, a pulse with one cycle of the PWM waveform as half its cycle is output from the TOABn0 pin.

Figure 8-24. Configuration in PWM Output Mode

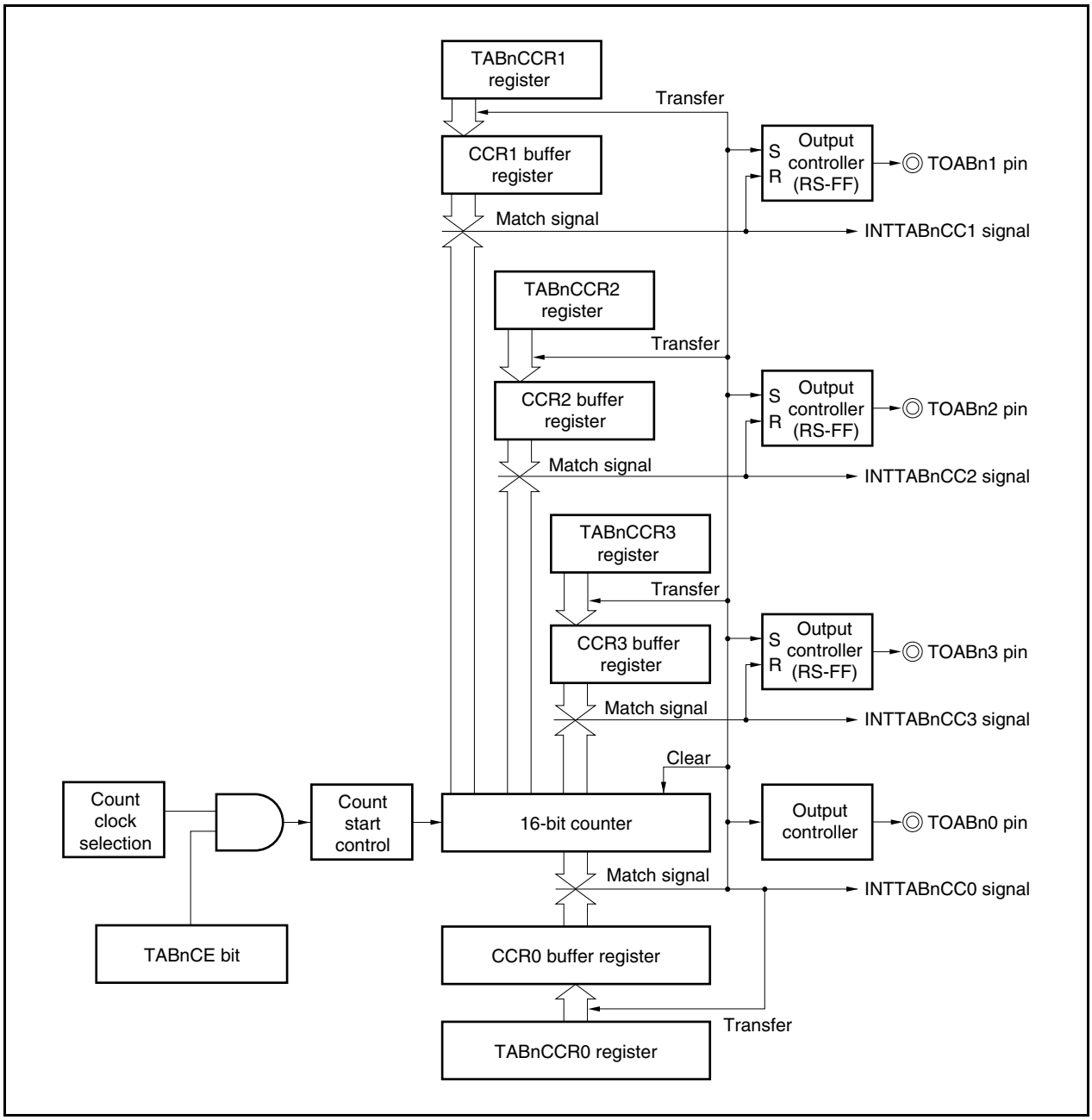
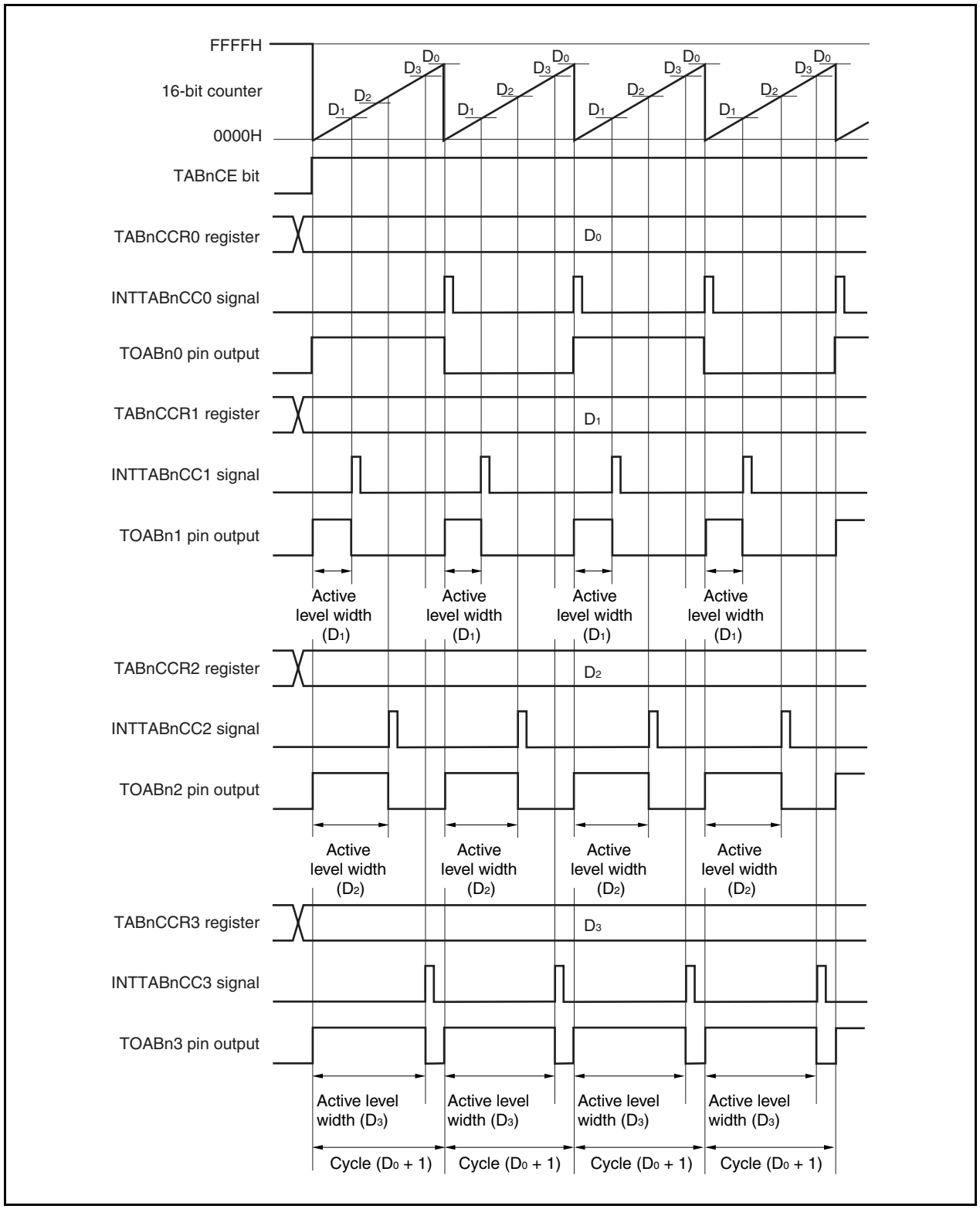


Figure 8-25. Basic Timing in PWM Output Mode



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

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

Active level width = (Set value of TABnCCRk register) × Count clock cycle
Cycle = (Set value of TABnCCR0 register + 1) × Count clock cycle
Duty factor = (Set value of TABnCCRk register)/(Set value of TABnCCR0 register + 1)

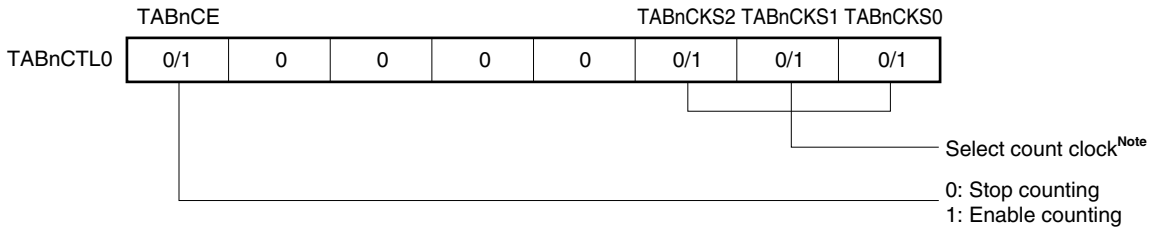
The PWM waveform can be changed by rewriting the TABnCCRm register while the counter is operating. The newly written value is reflected when the count value of the 16-bit counter matches the value of the CCR0 buffer register and the 16-bit counter is cleared to 0000H.

The compare match interrupt request signal (INTTABnCC0) is generated when the 16-bit counter counts next time after its count value matches the value of the CCR0 buffer register, and the 16-bit counter is cleared to 0000H. The compare match interrupt request signal (INTTABnCCk) is generated when the count value of the 16-bit counter matches the value of the CCRk buffer register.

Remark k = 1 to 3,
m = 0 to 3

Figure 8-26. Setting of Registers in PWM Output Mode (1/3)

(a) TABn control register 0 (TABnCTL0)

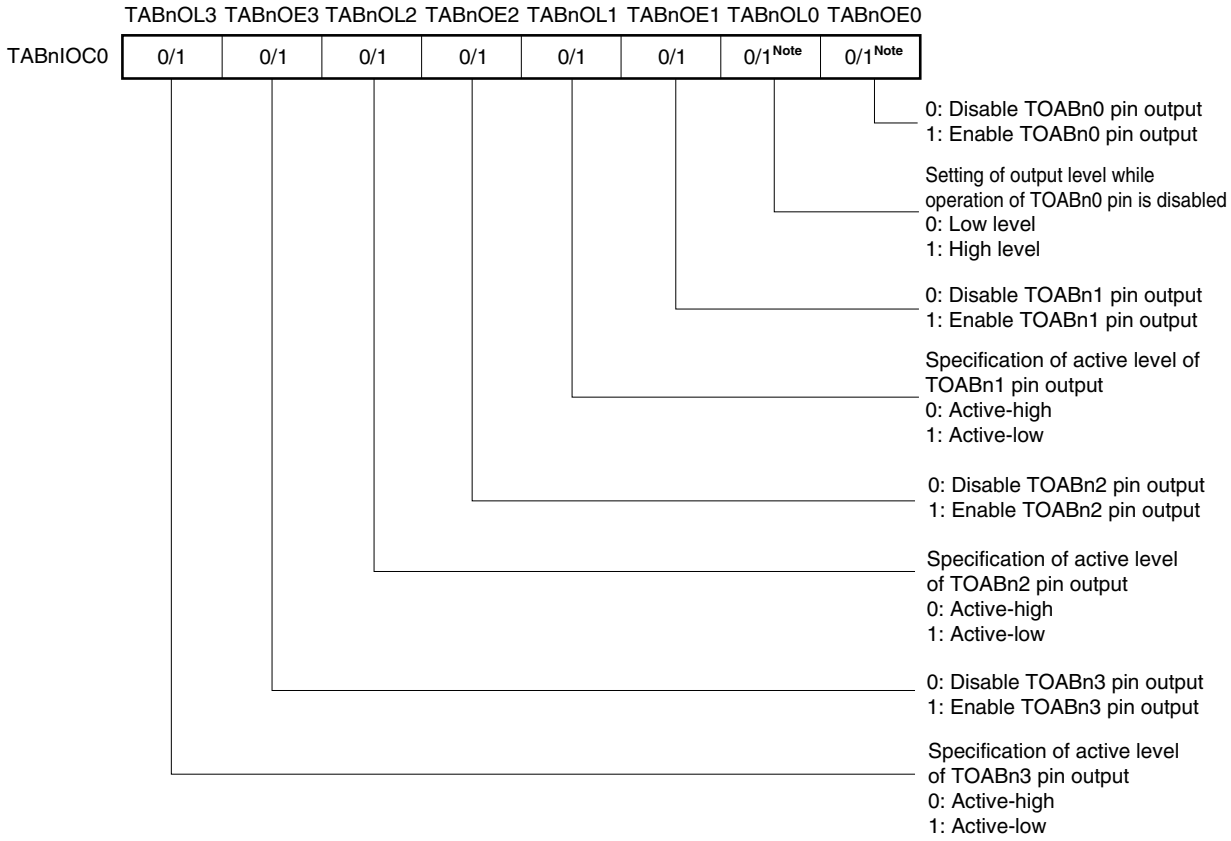


(b) TABn control register 1 (TABnCTL1)

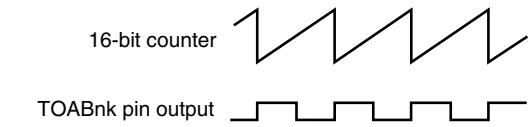
Note The setting is invalid when the TABnCTL1.TABnEEE bit = 1.

Figure 8-26. Setting of Registers in PWM Output Mode (2/3)

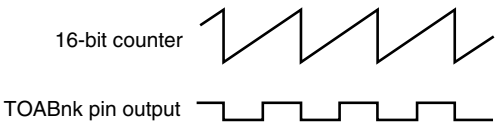
(c) TABn I/O control register 0 (TABnIOC0)



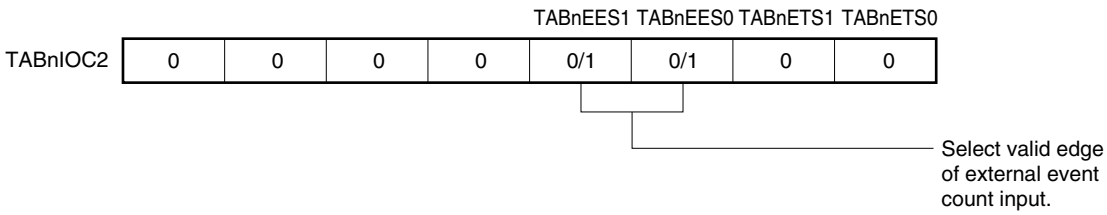
- When TABnOLk bit = 0



- When TABnOLk bit = 1



(d) TABn I/O control register 2 (TABnIOC2)



(e) TABn counter read buffer register (TABnCNT)

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

Note Clear this bit to 0 when the TOABn0 pin is not used in the PWM output mode.

Figure 8-26. Register Setting in PWM Output Mode (3/3)

(f) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)

If D_0 is set to the TABnCCR0 register and D_k to the TABnCCRk register, the cycle and active level of the PWM waveform are as follows.

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

Active level width = $D_k \times$ Count clock cycle

Remarks 1. TABn I/O control register 1 (TABnIOC1) and TABn option register 0 (TABnOPT0) are not used in the PWM output mode.

2. Updating the TABn capture/compare register 2 (TABnCCR2) and TABn capture/compare register 3 (TABnCCR3) is validated by writing the TABn capture/compare register 1 (TABnCCR1).

(1) Operation flow in PWM output mode

Figure 8-27. Software Processing Flow in PWM Output Mode (1/2)

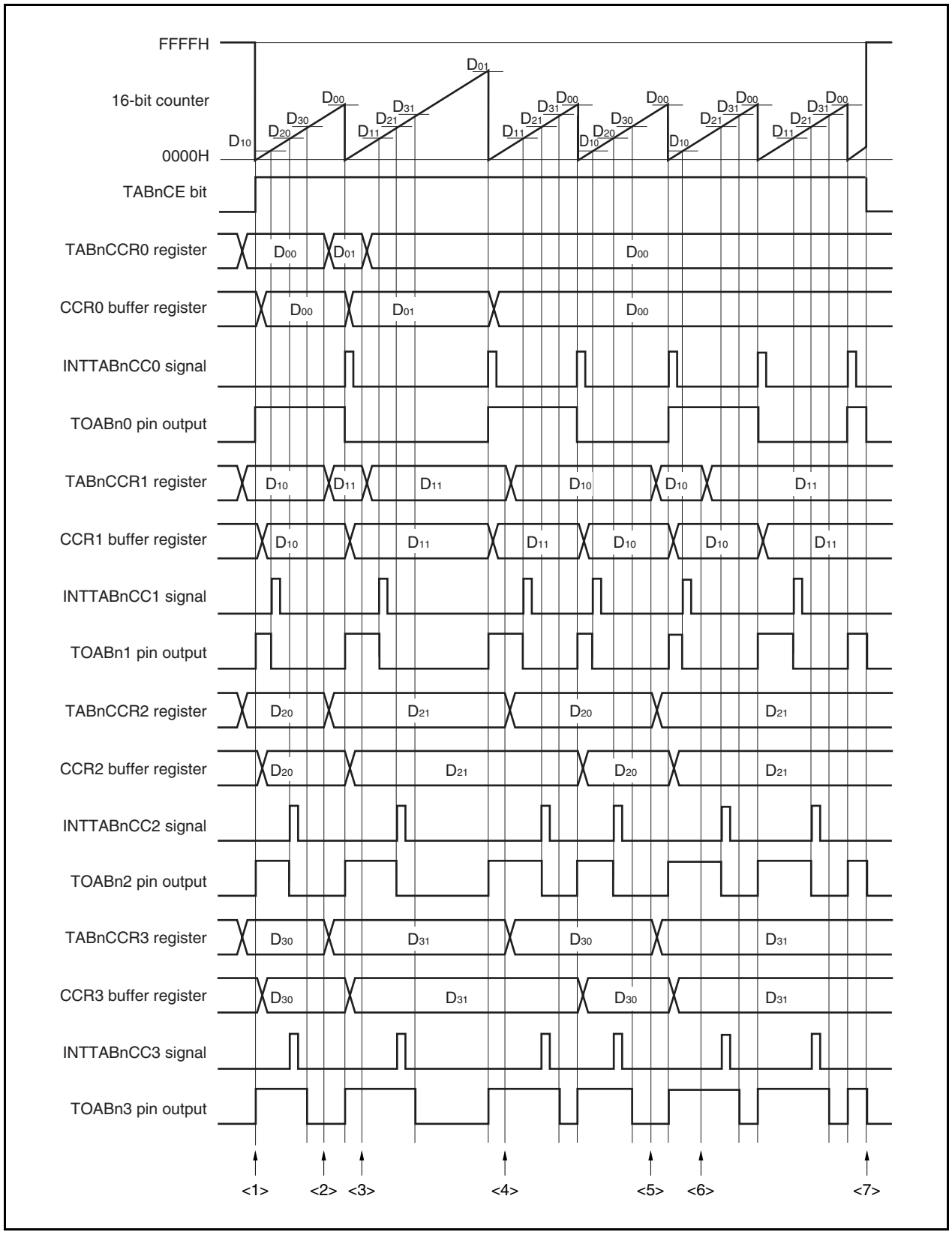
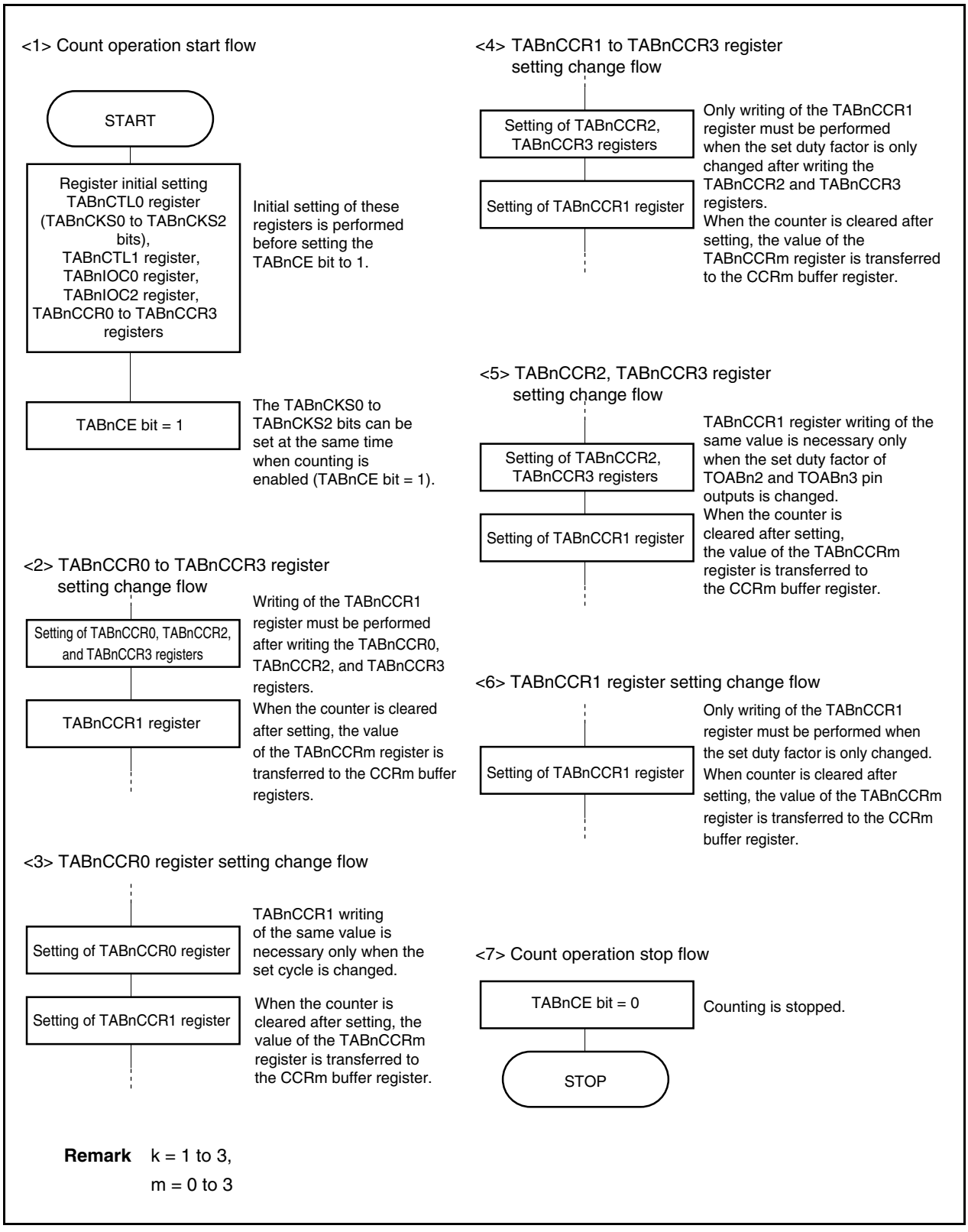


Figure 8-27. Software Processing Flow in PWM Output Mode (2/2)



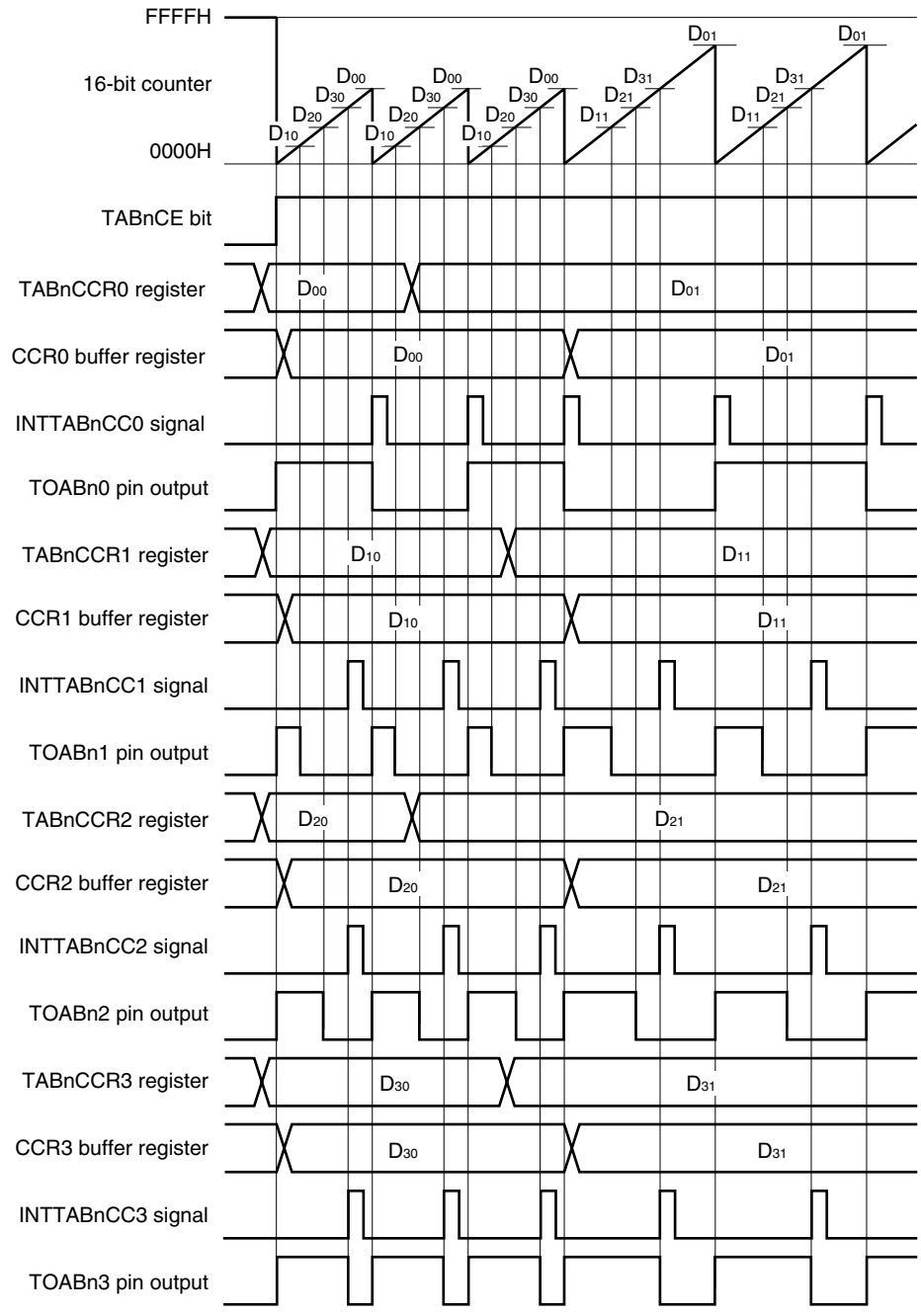
Remark k = 1 to 3,
m = 0 to 3

(2) PWM output mode operation timing

(a) Changing pulse width during operation

To change the PWM waveform while the counter is operating, write the TABnCCR1 register last.

Rewrite the TABnCCRk register after writing the TABnCCR1 register after the INTTABnCC1 signal is detected.



To transfer data from the TABnCCRm register to the CCRm buffer register, the TABnCCR1 register must be written.

To change both the cycle and active level of the PWM waveform at this time, first set the cycle to the TABnCCR0 register, set the active level width to the TABnCCR2 and TABnCCR3 registers, and then set an active level width to the TABnCCR1 register.

To change only the active level width (duty factor) of PWM wave, first set the active level to the TABnCCR2 and TABnCCR3 registers, and then set an active level to the TABnCCR1 register.

To change only the active level width (duty factor) of the PWM waveform output by the TOABn1 pin, only the TABnCCR1 register has to be set.

To change only the active level width (duty factor) of the PWM waveform output by the TOABn2 and TOABn3 pins, first set an active level width to the TABnCCR2 and TABnCCR3 registers, and then write the same value to the TABnCCR1 register.

After the TABnCCR1 register is written, the value written to the TABnCCRm register is transferred to the CCRm buffer register in synchronization with the timing of clearing the 16-bit counter, and is used as a value to be compared with the value of the 16-bit counter.

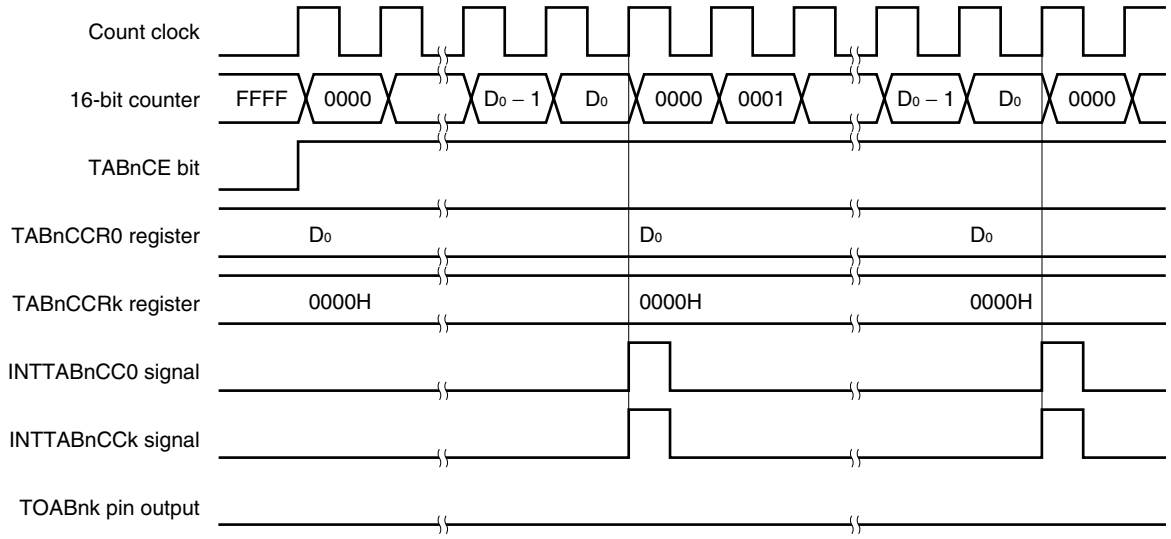
To change only the cycle of the PWM waveform, first set a cycle to the TABnCCR0 register, and then write the same value to the TABnCCR1 register.

To write the TABnCCR0 to TABnCCR3 registers again after writing the TABnCCR1 register once, do so after the INTTABnCC0 signal is generated. Otherwise, the value of the CCRm buffer register may become undefined because the timing of transferring data from the TABnCCRm register to the CCRm buffer register conflicts with writing the TABnCCRm register.

Remark m = 0 to 3

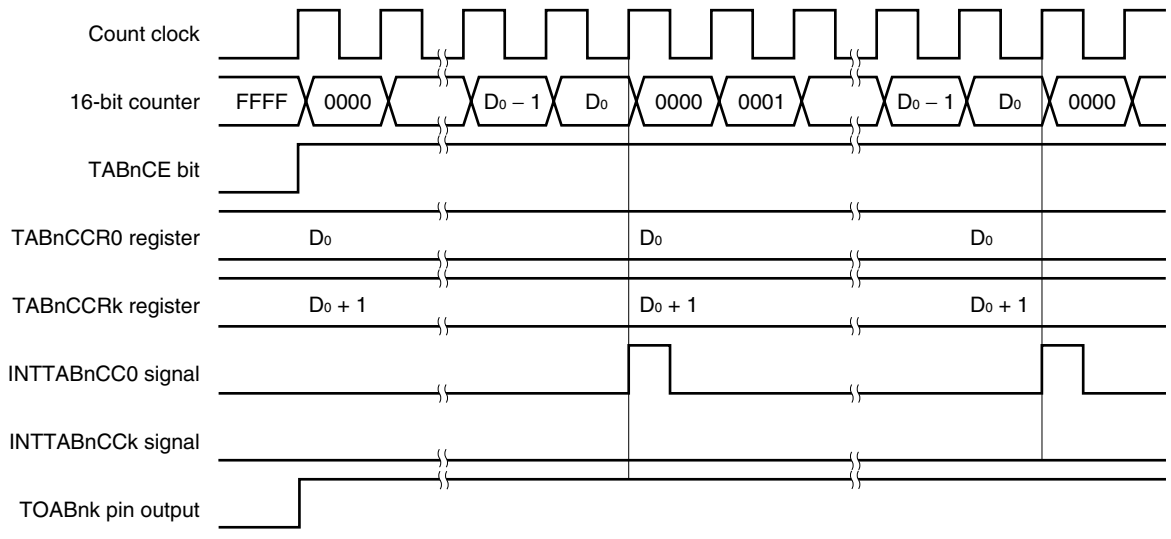
(b) 0%/100% output of PWM waveform

To output a 0% waveform, set the TABnCCRk register to 0000H. If the set value of the TABnCCR0 register is FFFFH, the INTTABnCCk signal is generated periodically.



Remark k = 1 to 3

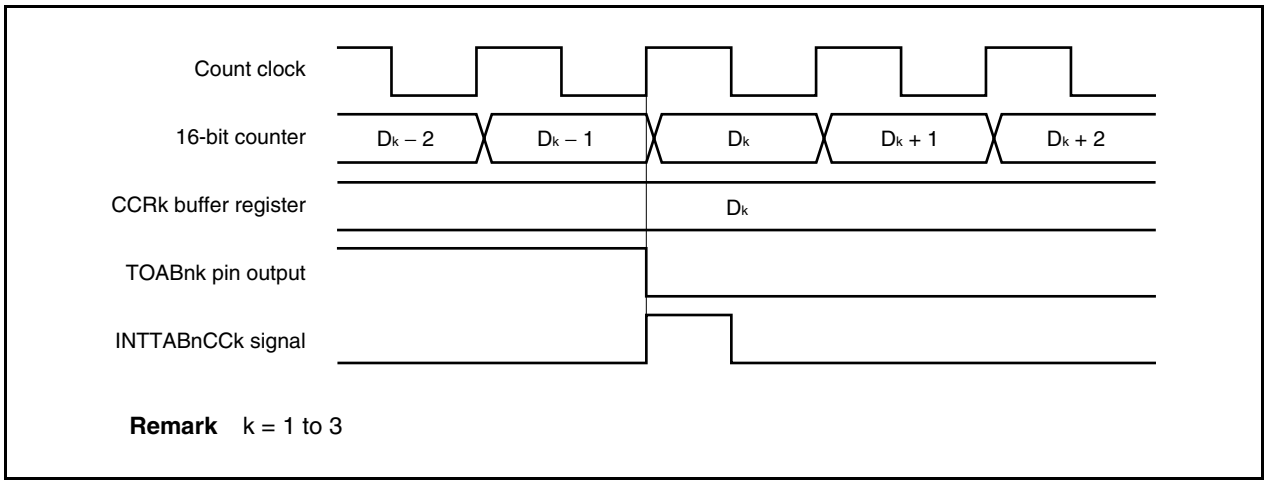
To output a 100% waveform, set a value of (set value of TABnCCR0 register + 1) to the TABnCCRk register. If the set value of the TABnCCR0 register is FFFFH, 100% output cannot be produced.



Remark k = 1 to 3

(c) Generation timing of compare match interrupt request signal (INTTABnCCk)

The timing of generation of the INTTABnCCk signal in the PWM output mode differs from the timing of other INTTABnCCk signals; the INTTABnCCk signal is generated when the count value of the 16-bit counter matches the value of the TABnCCRk register.



Remark k = 1 to 3

Usually, the INTTABnCCk signal is generated in synchronization with the next counting up after the count value of the 16-bit counter matches the value of the TABnCCRk register.

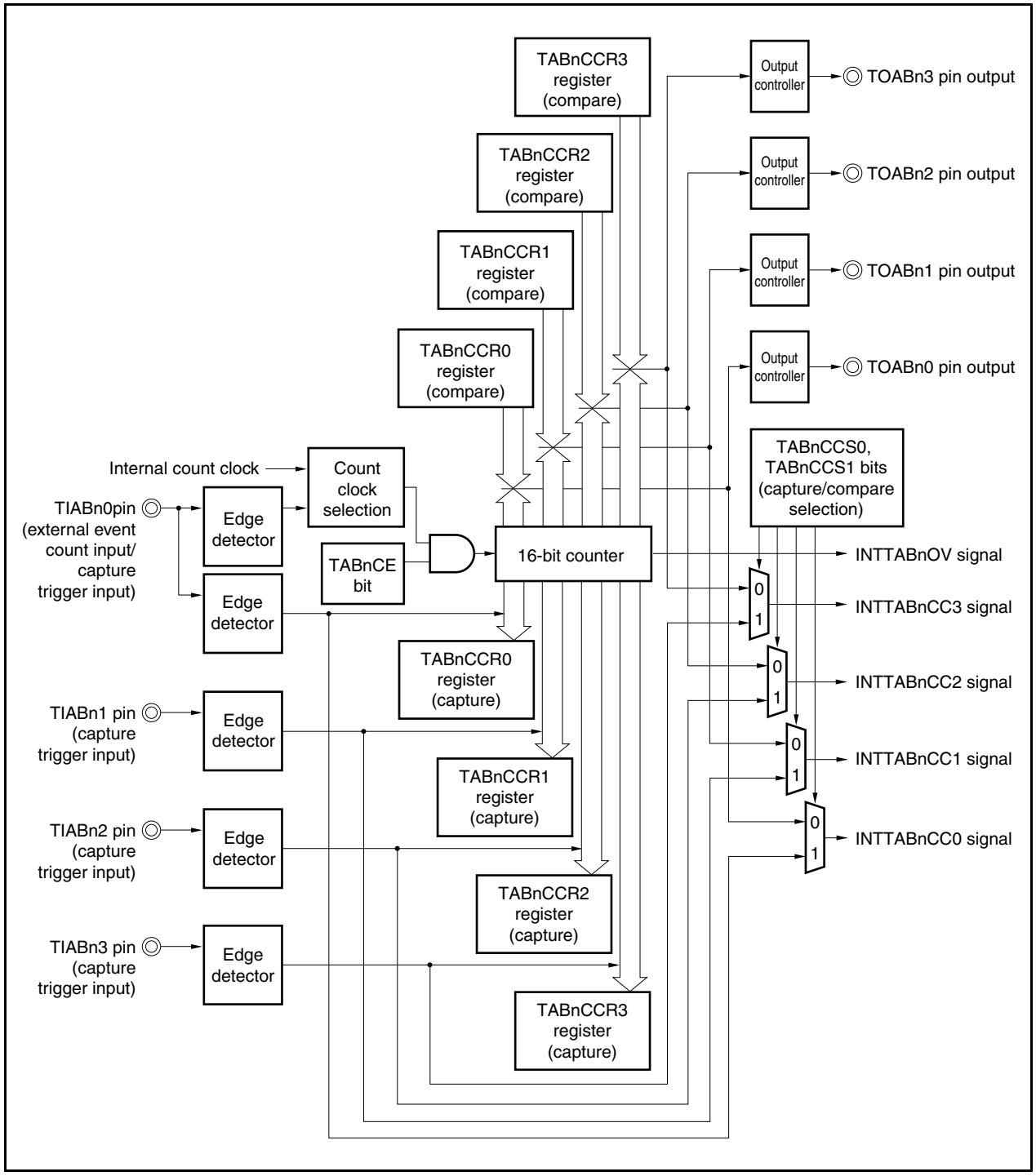
In the PWM output mode, however, it is generated one clock earlier. This is because the timing is changed to match the change timing of the output signal of the TOABnk pin.

8.5.6 Free-running timer mode (TABnMD2 to TABnMD0 bits = 101)

In the free-running timer mode, 16-bit timer/event counter AB starts counting when the TABnCTL0.TABnCE bit is set to 1. At this time, the TABnCCRm register can be used as a compare register or a capture register, depending on the setting of the TABnOPT0.TABnCCS0 and TABnOPT0.TABnCCS1 bits.

Remark m = 0 to 3

Figure 8-28. Configuration in Free-Running Timer Mode

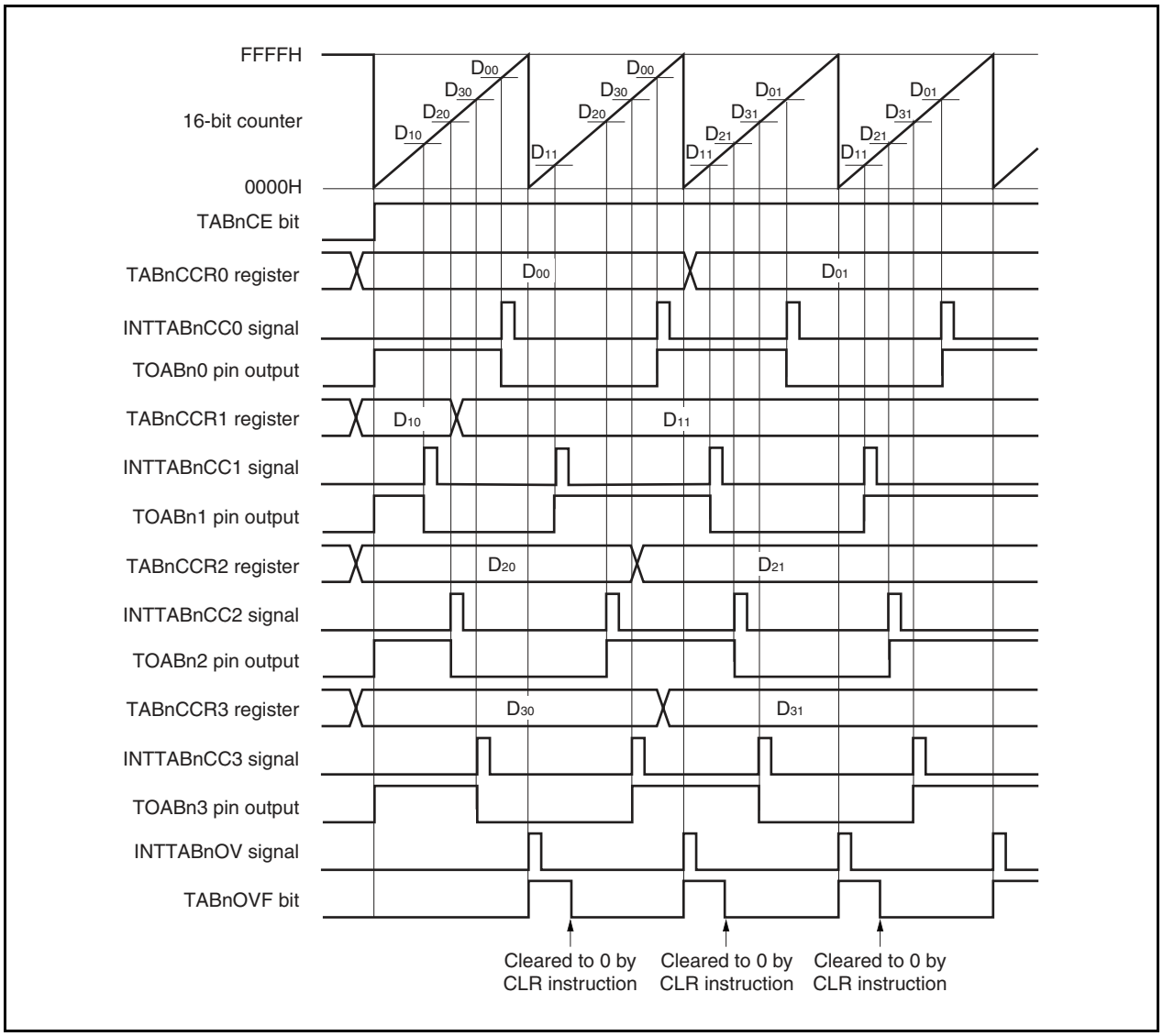


When the TABnCE bit is set to 1, 16-bit timer/event counter AB starts counting, and the output signals of the TOABn0 to TOABn3 pins are inverted. When the count value of the 16-bit counter later matches the set value of the TABnCCRm register, a compare match interrupt request signal (INTTABnCCm) is generated, and the output signal of the TOABnm pin is inverted.

The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTABnOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TABnOPT0.TABnOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction by software.

The TABnCCRm register can be rewritten while the counter is operating. If it is rewritten, the new value is reflected at that time, and compared with the count value (m = 0 to 3).

Figure 8-29. Basic Timing in Free-Running Timer Mode (Compare Function)



When the TABnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIABnm pin is detected, the count value of the 16-bit counter is stored in the TABnCCRm register, and a capture interrupt request signal (INTTABnCCm) is generated.

The 16-bit counter continues counting in synchronization with the count clock. When it counts up to FFFFH, it generates an overflow interrupt request signal (INTTABnOV) at the next clock, is cleared to 0000H, and continues counting. At this time, the overflow flag (TABnOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction by software (m = 0 to 3).

Figure 8-30. Basic Timing in Free-Running Timer Mode (Capture Function)

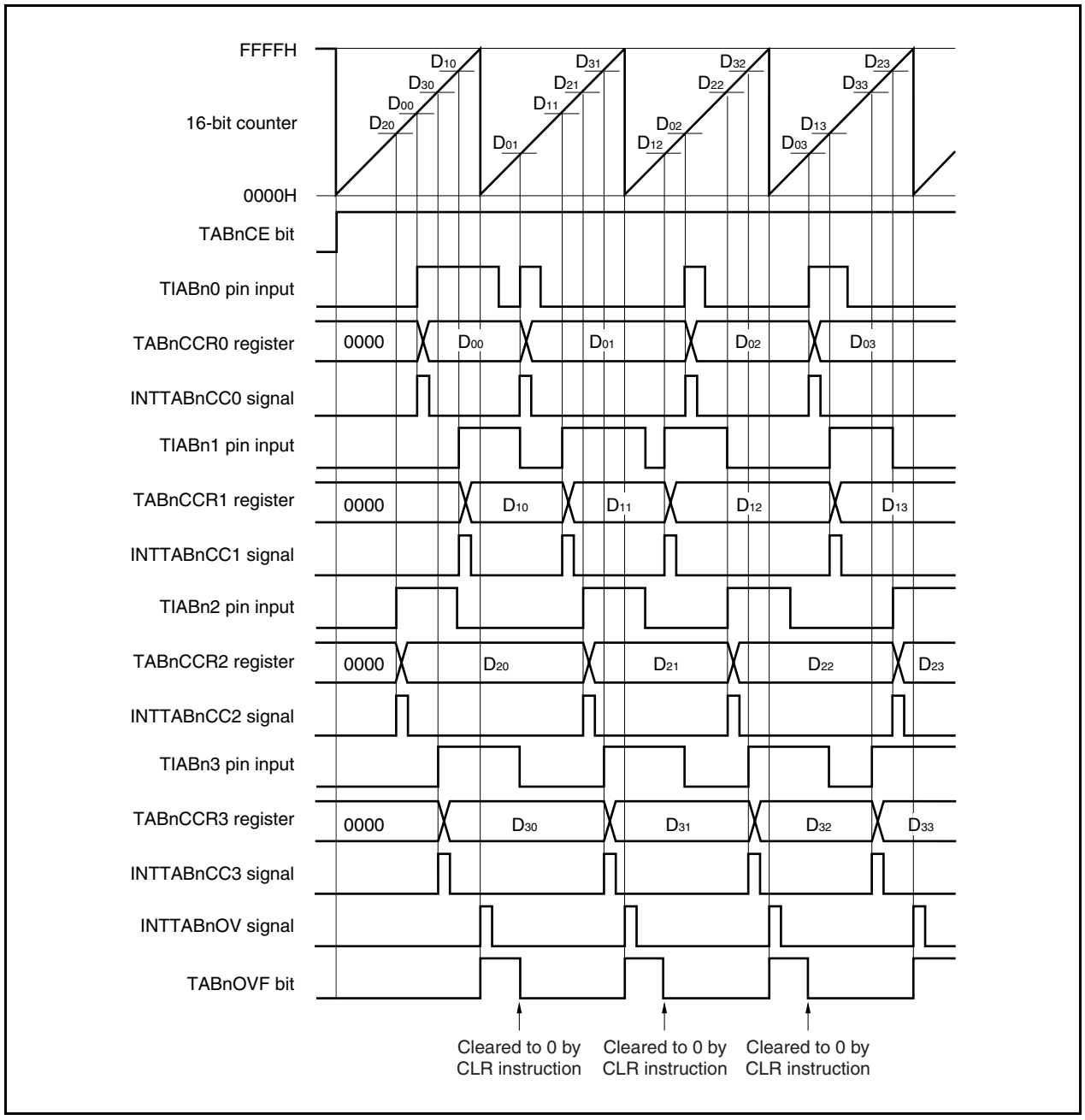
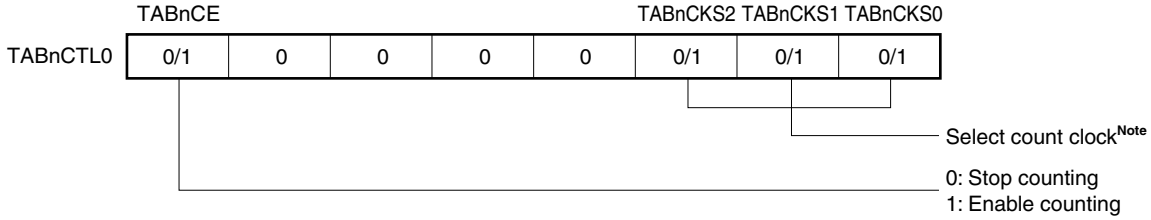


Figure 8-31. Register Setting in Free-Running Timer Mode (1/3)

(a) TABn control register 0 (TABnCTL0)



Note The setting is invalid when the TABnCTL1.TABnEEE bit = 1

(b) TABn control register 1 (TABnCTL1)

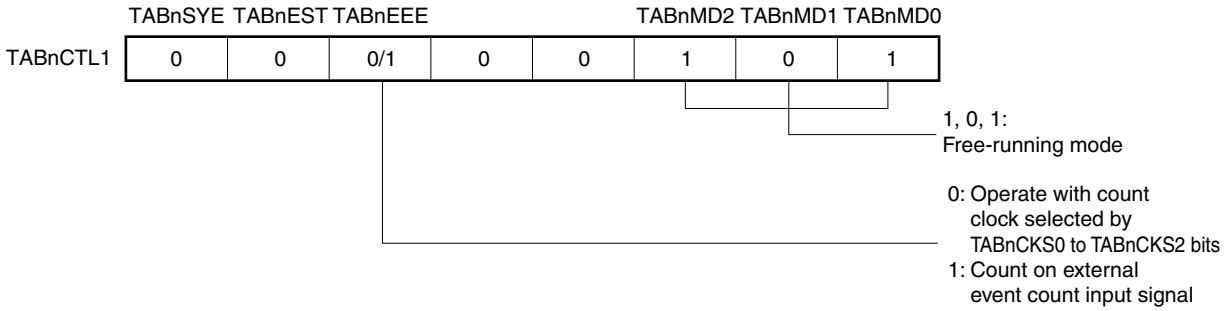
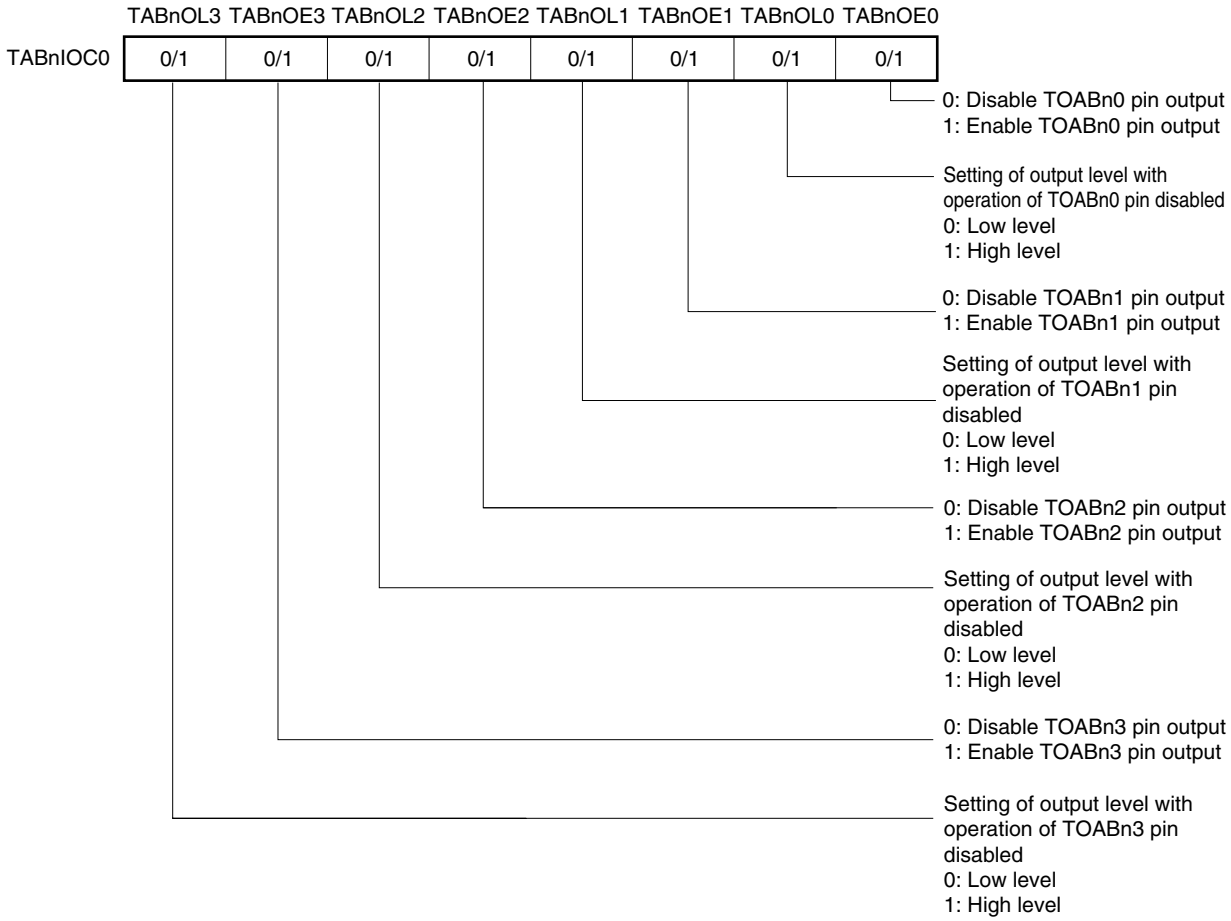


Figure 8-31. Register Setting in Free-Running Timer Mode (2/3)

(c) TABn I/O control register 0 (TABnIOC0)



(d) TABn I/O control register 1 (TABnIOC1)

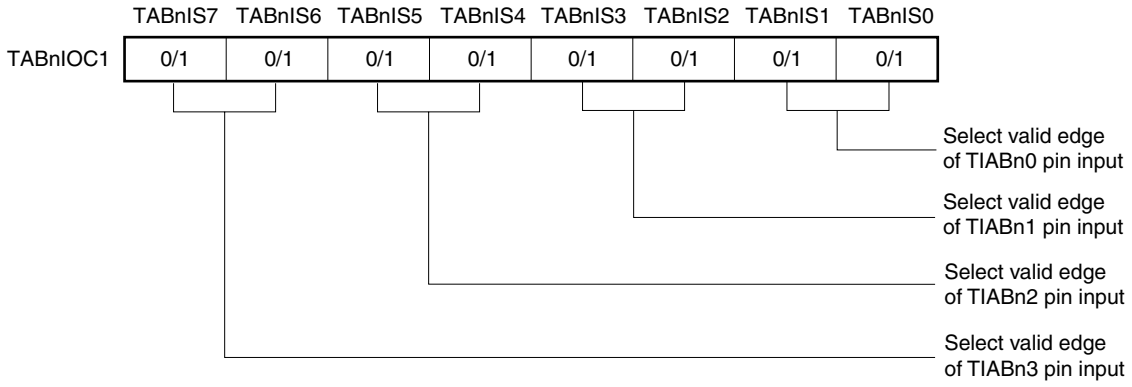
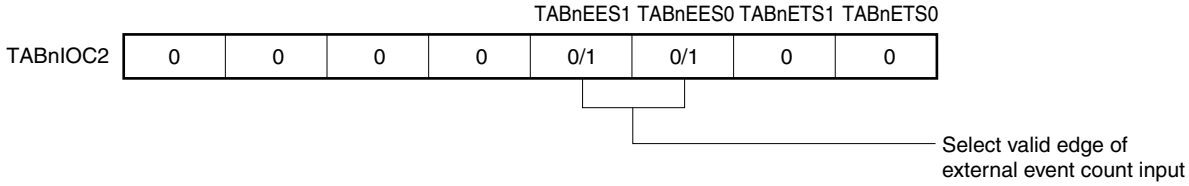
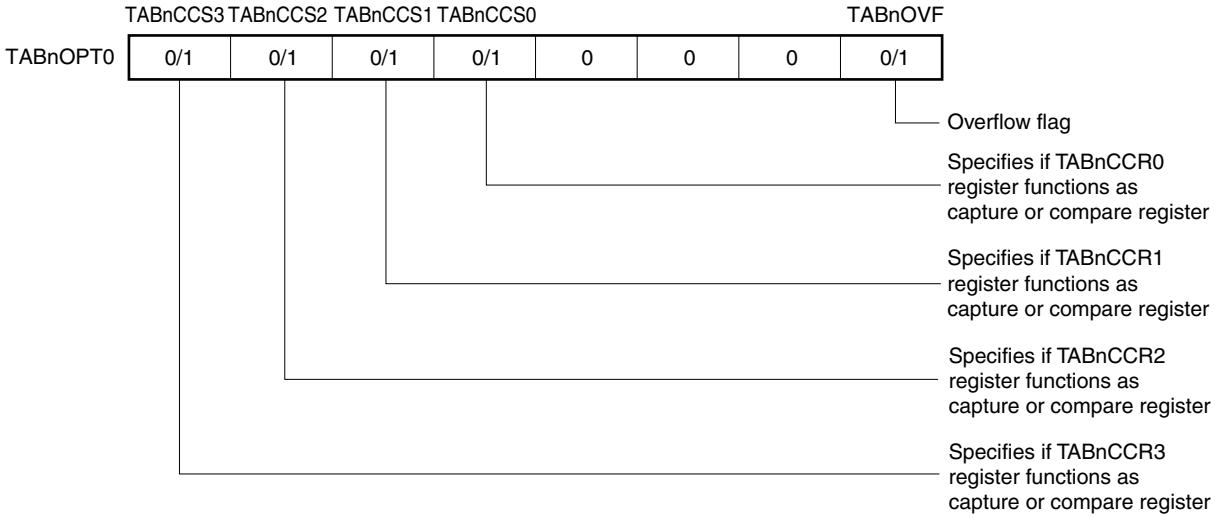


Figure 8-31. Register Setting in Free-Running Timer Mode (3/3)

(e) TABn I/O control register 2 (TABnIOC2)



(f) TABn option register 0 (TABnOPT0)



(g) TABn counter read buffer register (TABnCNT)

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

(h) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)

These registers function as capture registers or compare registers depending on the setting of the TABnOPT0.TABnCCSm bit.

When the registers function as capture registers, they store the count value of the 16-bit counter when the valid edge input to the TIABnm pin is detected.

When the registers function as compare registers and when D_m is set to the TABnCCRm register, the INTTABnCCm signal is generated when the counter reaches (D_m + 1), and the output signal of the TOABnm pin is inverted.

Remark m = 0 to 3

(1) Operation flow in free-running timer mode

(a) When using capture/compare register as compare register

Figure 8-32. Software Processing Flow in Free-Running Timer Mode (Compare Function) (1/2)

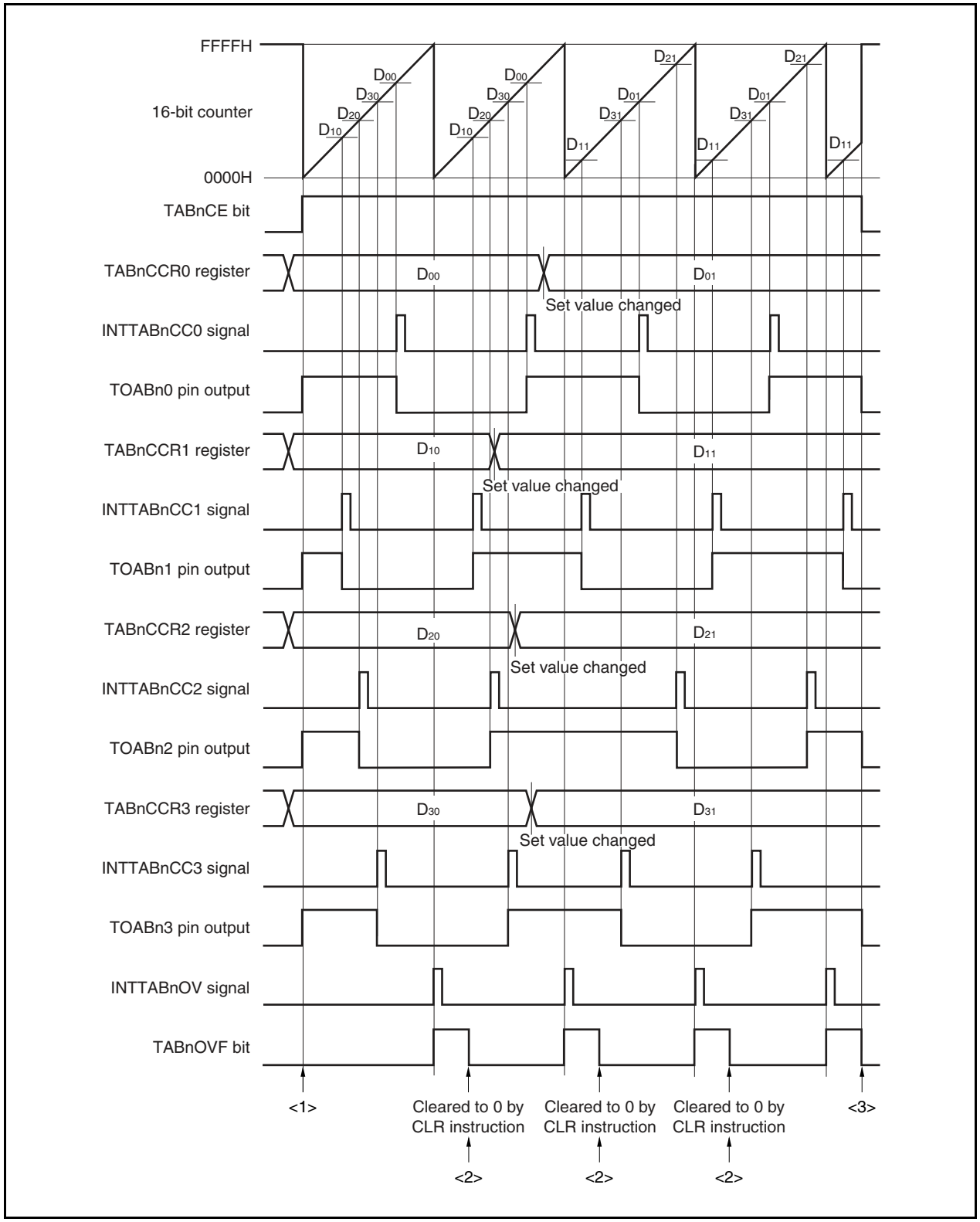
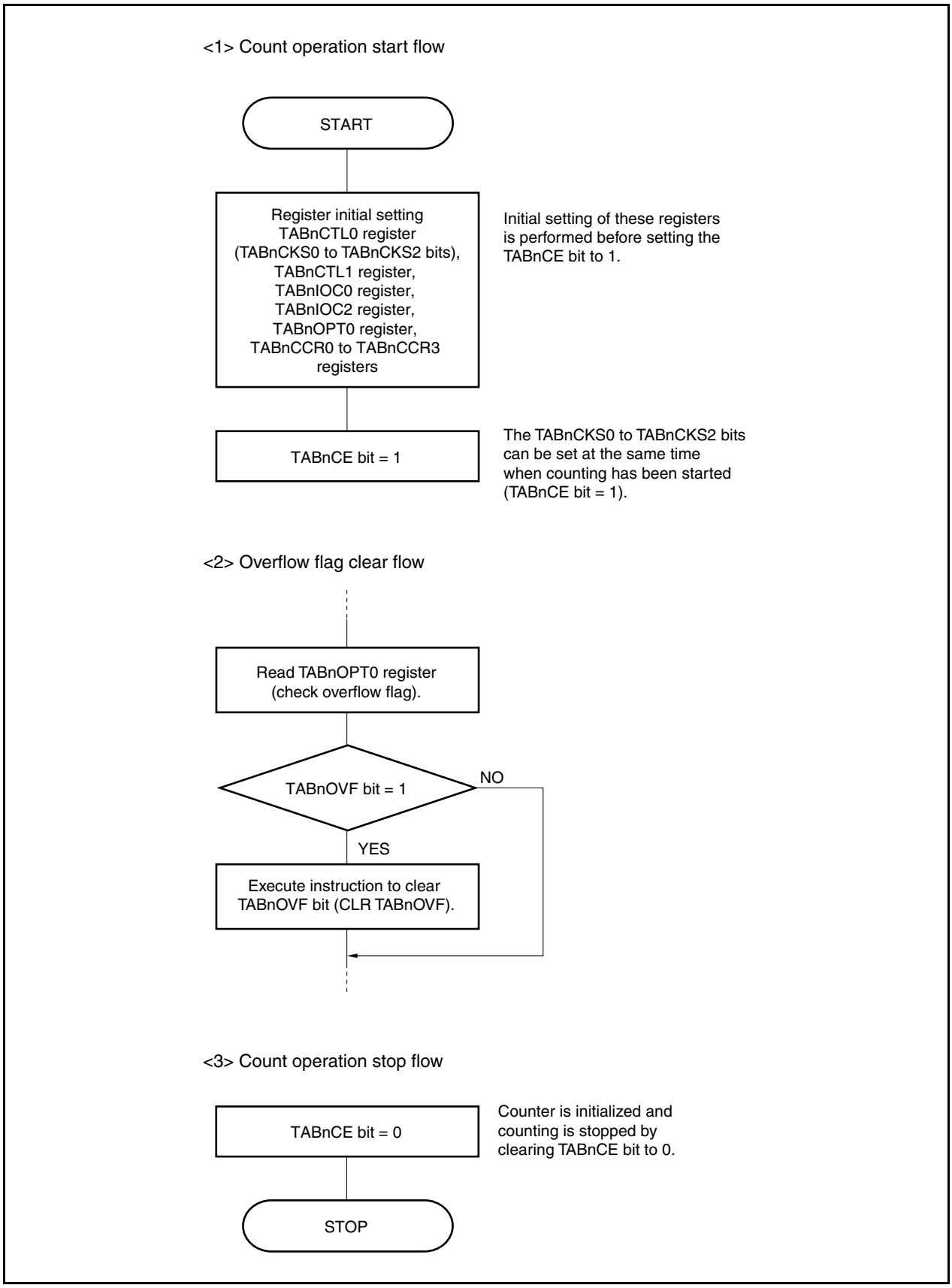


Figure 8-32. Software Processing Flow in Free-Running Timer Mode (Compare Function) (2/2)



(b) When using capture/compare register as capture register

Figure 8-33. Software Processing Flow in Free-Running Timer Mode (Capture Function) (1/2)

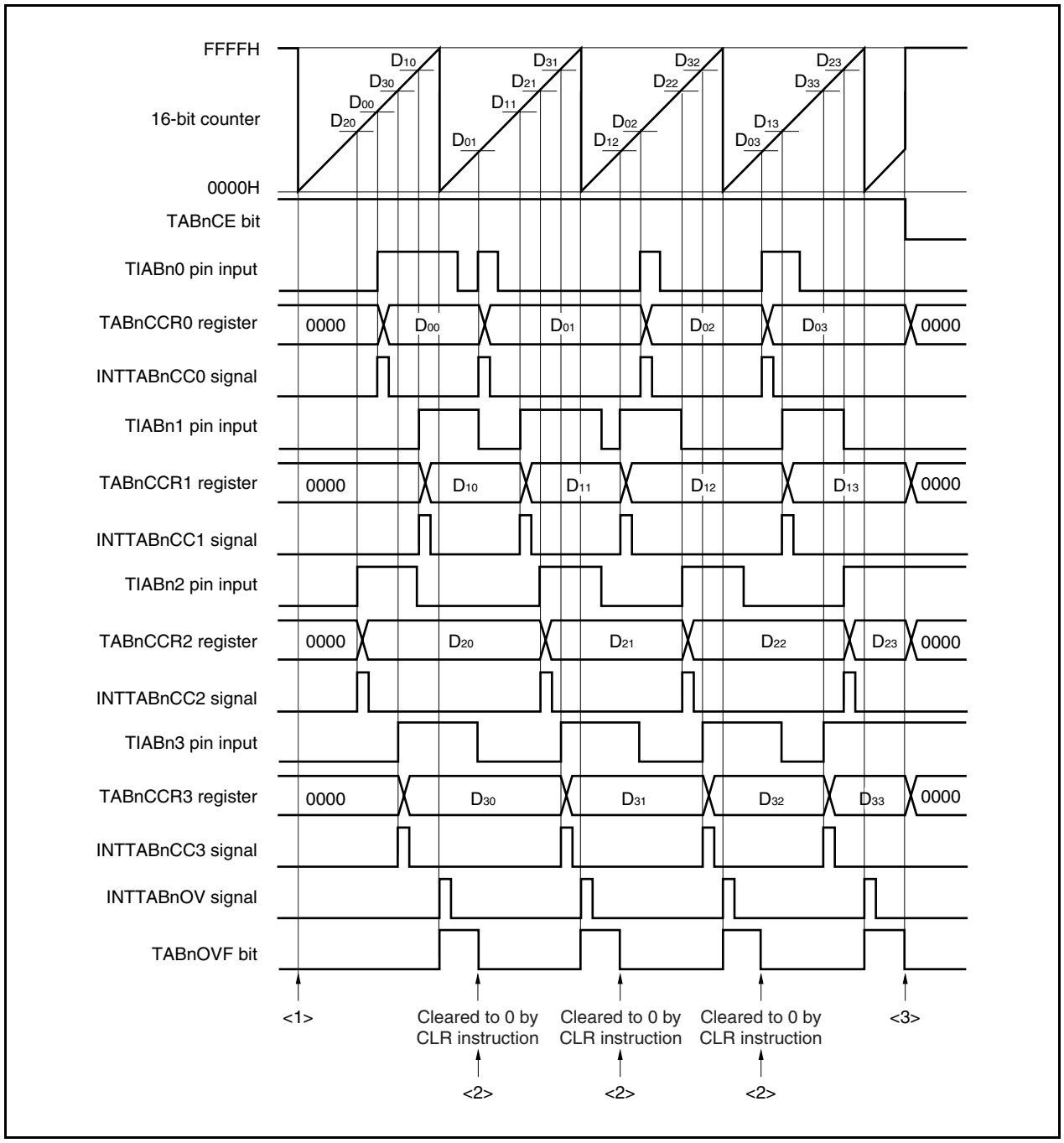
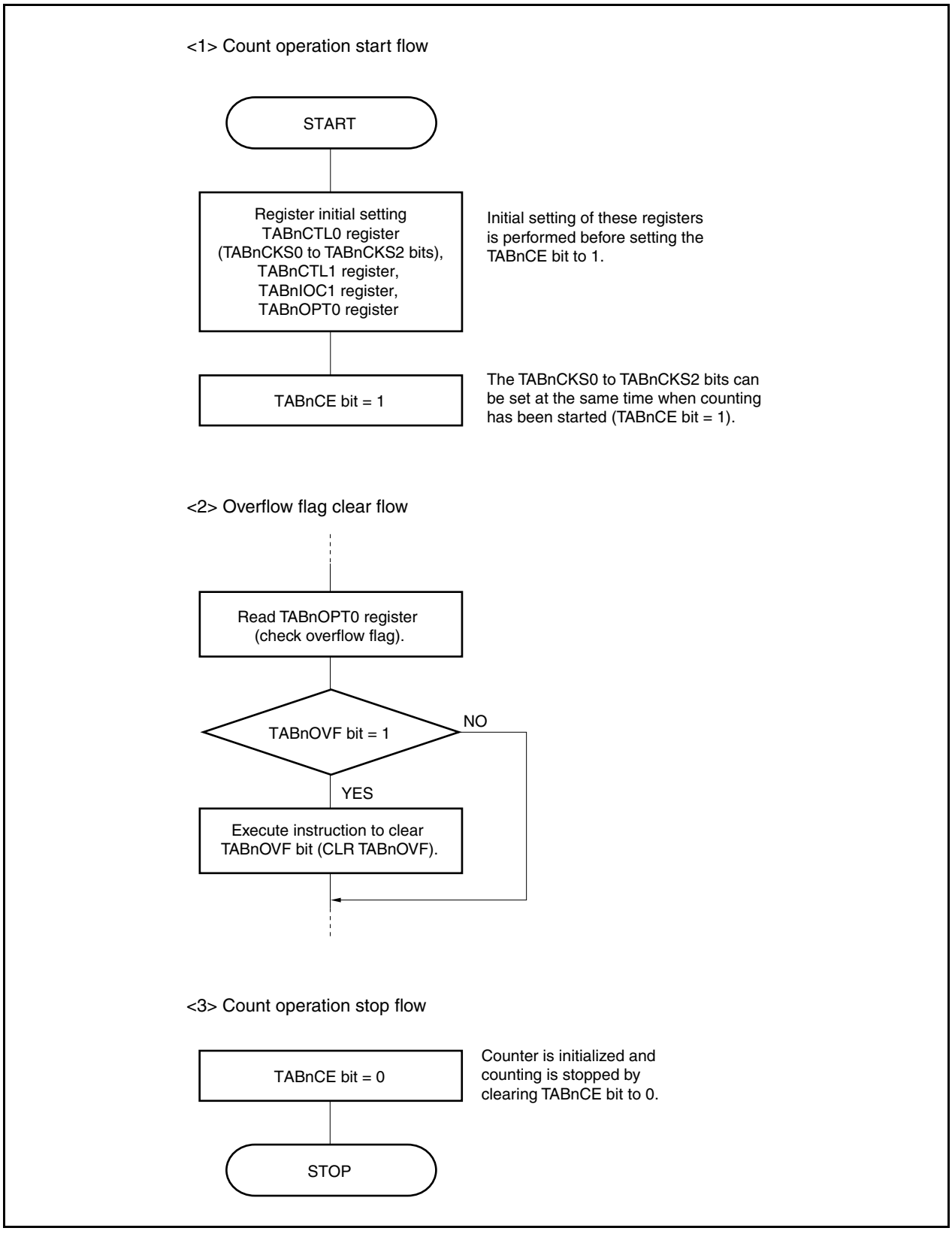


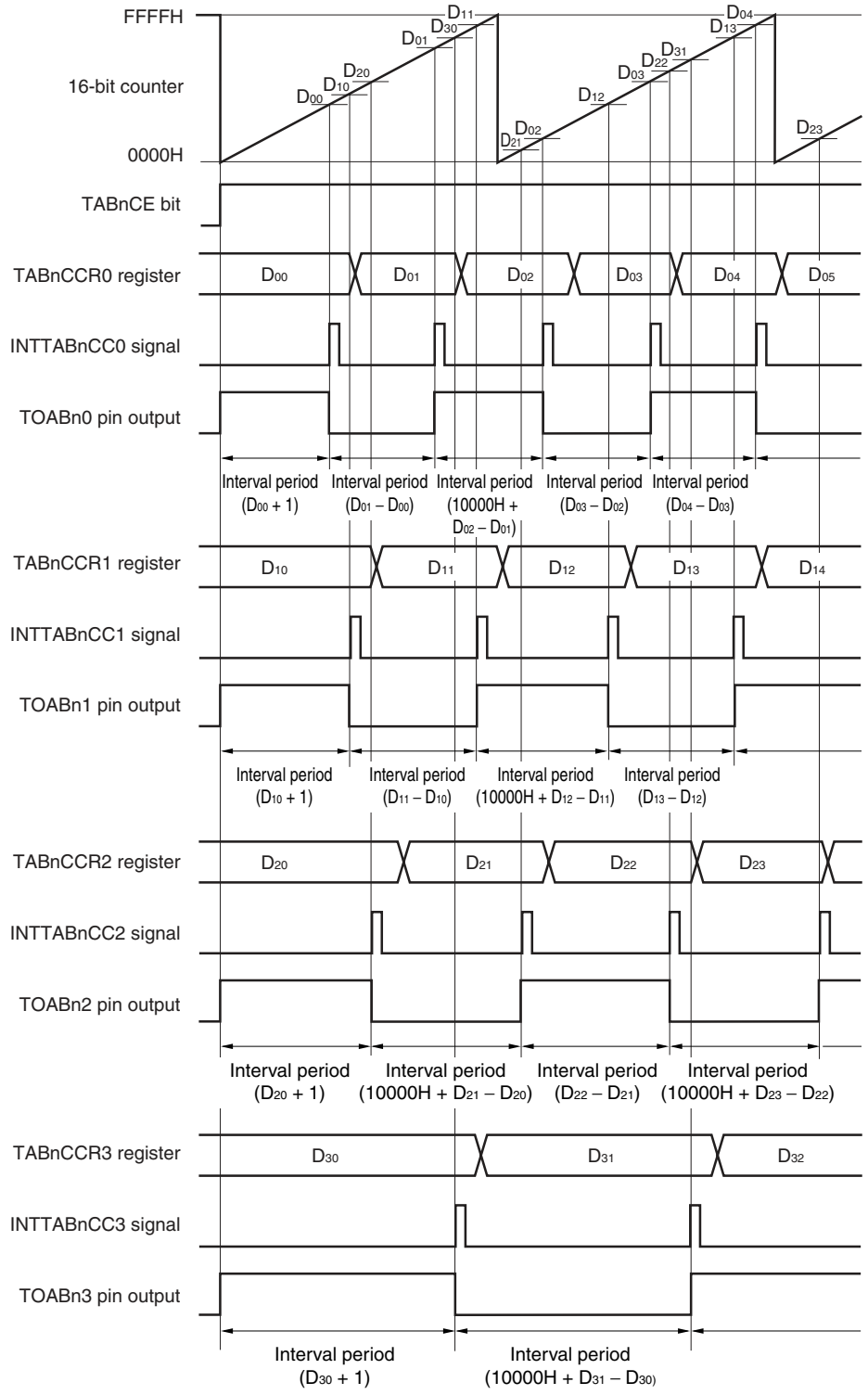
Figure 8-33. Software Processing Flow in Free-Running Timer Mode (Capture Function) (2/2)



(2) Operation timing in free-running timer mode

(a) Interval operation with compare register

When 16-bit timer/event counter AB is used as an interval timer with the TABnCCRm register used as a compare register, software processing is necessary for setting a comparison value to generate the next interrupt request signal each time the INTTABnCCm signal has been detected.



When performing an interval operation in the free-running timer mode, two intervals can be set with one channel.

To perform the interval operation, the value of the corresponding TABnCCRm register must be re-set in the interrupt servicing that is executed when the INTTABnCCm signal is detected.

The set value for re-setting the TABnCCRm register can be calculated by the following expression, where "D_m" is the interval period.

Compare register default value: $D_m - 1$

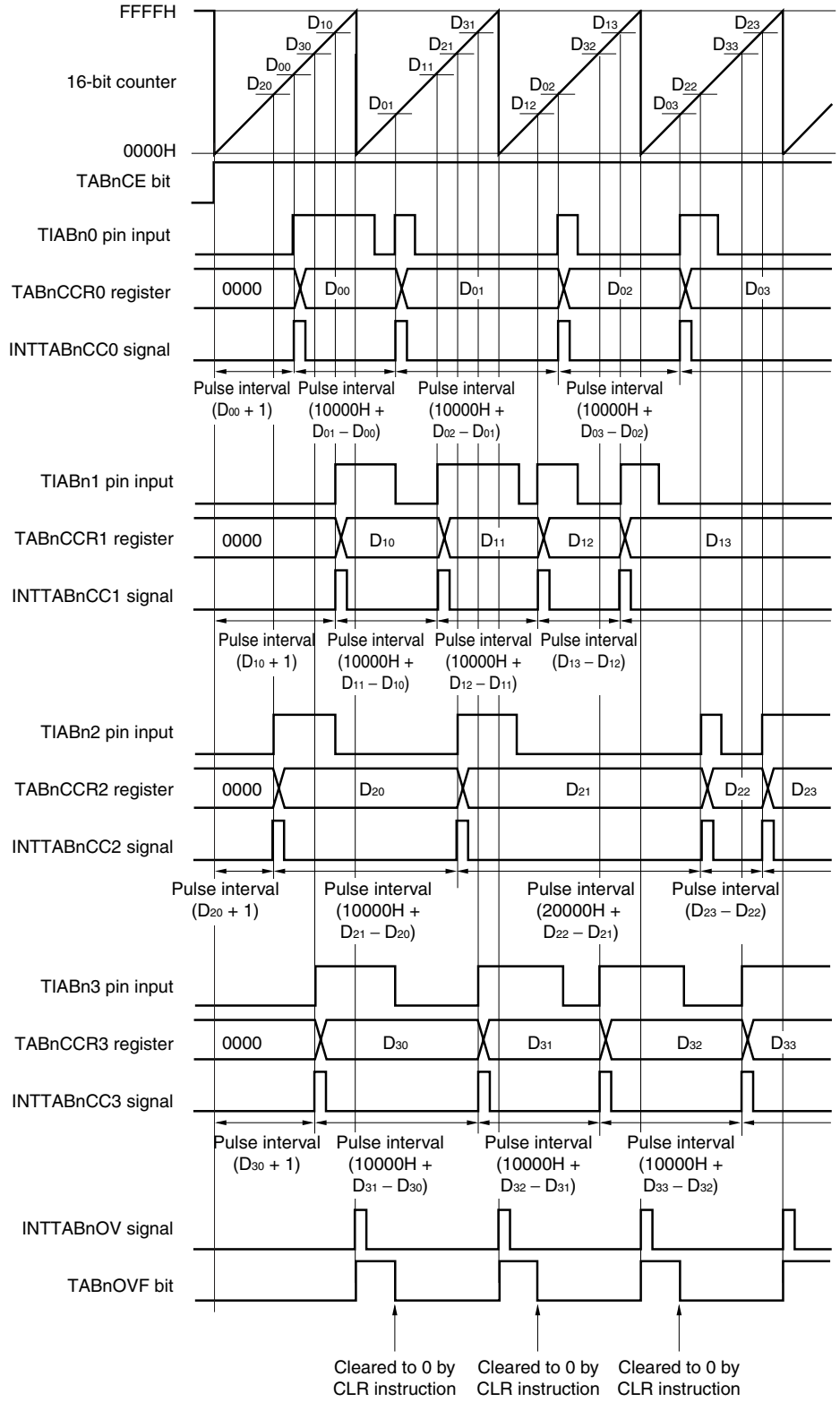
Value set to compare register second and subsequent time: Previous set value + D_m

(If the calculation result is greater than FFFFH, subtract 10000H from the result and set this value to the register.)

Remark m = 0 to 3

(b) Pulse width measurement with capture register

When pulse width measurement is performed with the TABnCCRm register used as a capture register, software processing is necessary for reading the capture register each time the INTTABnCCm signal has been detected and for calculating an interval.



When executing pulse width measurement in the free-running timer mode, four pulse widths can be measured with one channel.

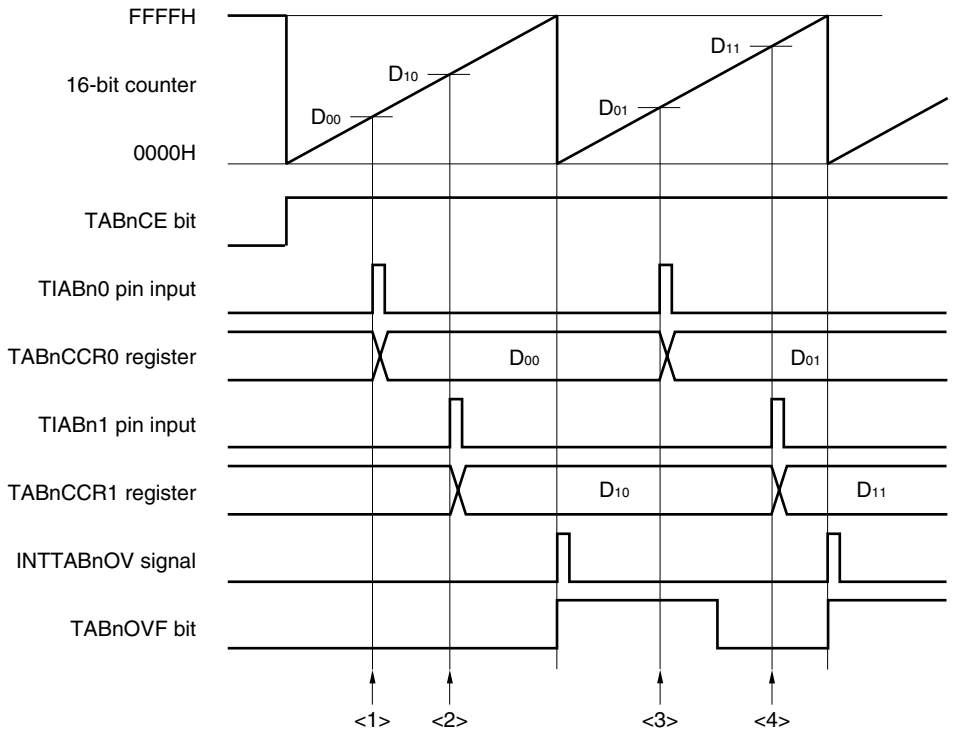
To measure a pulse width, the pulse width can be calculated by reading the value of the TABnCCRm register in synchronization with the INTTABnCCm signal, and calculating the difference between the read value and the previously read value.

Remark m = 0 to 3

(c) Processing of overflow when two or more capture registers are used

Care must be exercised in processing the overflow flag when two capture registers are used. First, an example of incorrect processing is shown below.

Example of incorrect processing when two or more capture registers are used

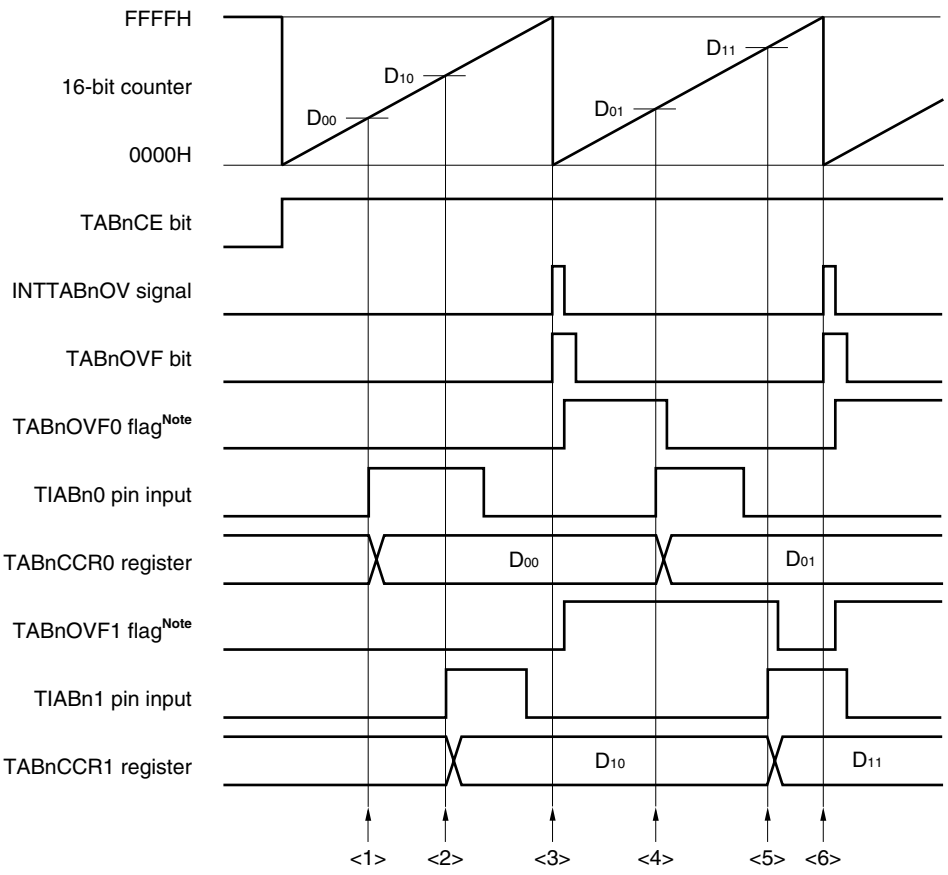


The following problem may occur when two pulse widths are measured in the free-running timer mode.

<1> Read the TABnCCR0 register (setting of the default value of the TIABn0 pin input).
<2> Read the TABnCCR1 register (setting of the default value of the TIABn1 pin input).
<3> Read the TABnCCR0 register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).
<4> Read the TABnCCR1 register.
Read the overflow flag. Because the flag is cleared in <3>, 0 is read.
Because the overflow flag is 0, the pulse width can be calculated by (D_{11} – D_{10}) (incorrect).

When two capture registers are used, and if the overflow flag is cleared to 0 by one capture register, the other capture register may not obtain the correct pulse width.
Use software when using two capture registers. An example of how to use software is shown below.

Example when two capture registers are used (using overflow interrupt)



Note The TABnOVF0 and TABnOVF1 flags are set on the internal RAM by software.

<1> Read the TABnCCR0 register (setting of the default value of the TIABn0 pin input).

<2> Read the TABnCCR1 register (setting of the default value of the TIABn1 pin input).

<3> An overflow occurs. Set the TABnOVF0 and TABnOVF1 flags to 1 in the overflow interrupt servicing, and clear the overflow flag to 0.

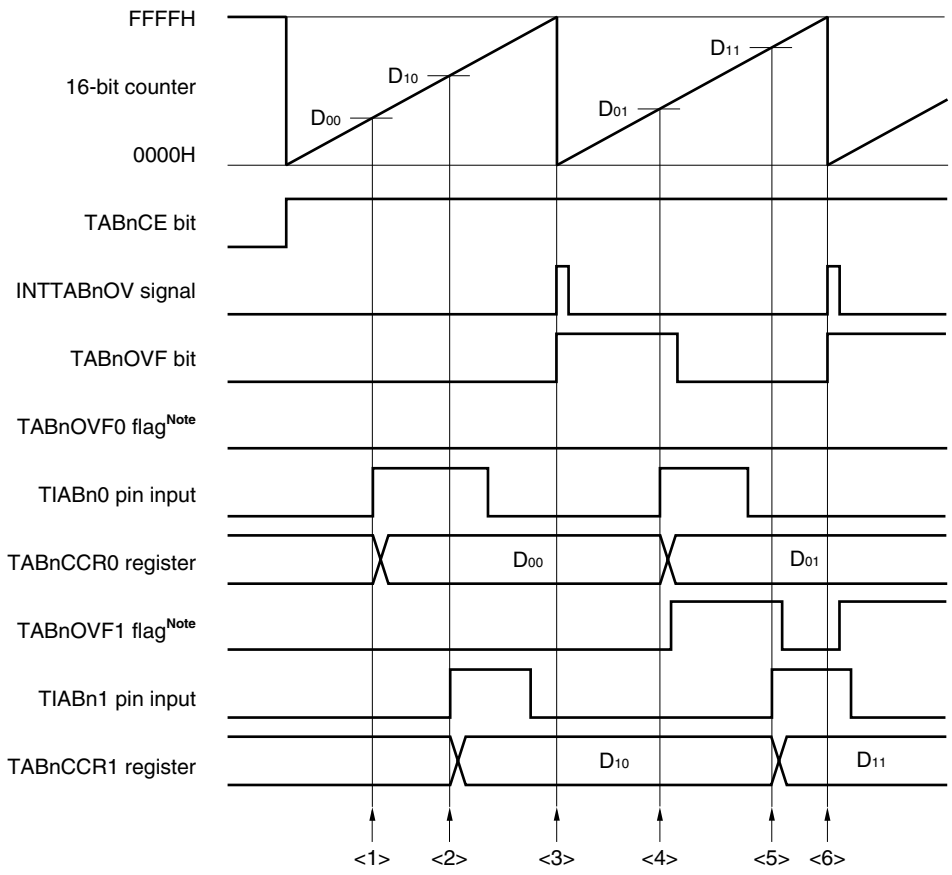
<4> Read the TABnCCR0 register.
Read the TABnOVF0 flag. If the TABnOVF0 flag is 1, clear it to 0.
Because the TABnOVF0 flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).

<5> Read the TABnCCR1 register.
Read the TABnOVF1 flag. If the TABnOVF1 flag is 1, clear it to 0 (the TABnOVF0 flag is cleared in <4>, and the TABnOVF1 flag remains 1).
Because the TABnOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} – D_{10}) (correct).

<6> Same as <3>

(2/2)

Example when two capture registers are used (without using overflow interrupt)



Note The TABnOVF0 and TABnOVF1 flags are set on the internal RAM by software.

<1> Read the TABnCCR0 register (setting of the default value of the TIABn0 pin input).

<2> Read the TABnCCR1 register (setting of the default value of the TIABn1 pin input).

<3> An overflow occurs. Nothing is done by software.

<4> Read the TABnCCR0 register.

Read the overflow flag. If the overflow flag is 1, set only the TABnOVF1 flag to 1, and clear the overflow flag to 0.

Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{01} – D_{00}).

<5> Read the TABnCCR1 register.

Read the overflow flag. Because the overflow flag is cleared in <4>, 0 is read.

Read the TABnOVF1 flag. If the TABnOVF1 flag is 1, clear it to 0.

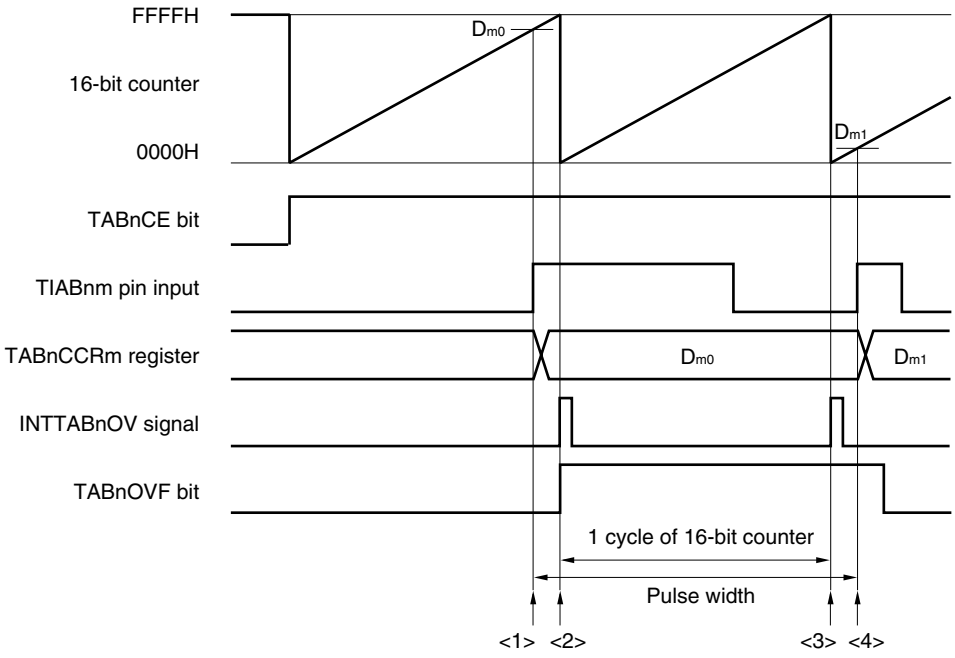
Because the TABnOVF1 flag is 1, the pulse width can be calculated by (10000H + D_{11} – D_{10}) (correct).

<6> Same as <3>

(d) Processing of overflow if capture trigger interval is long

If the pulse width is greater than one cycle of the 16-bit counter, care must be exercised because an overflow may occur more than once from the first capture trigger to the next. First, an example of incorrect processing is shown below.

Example of incorrect processing when capture trigger interval is long



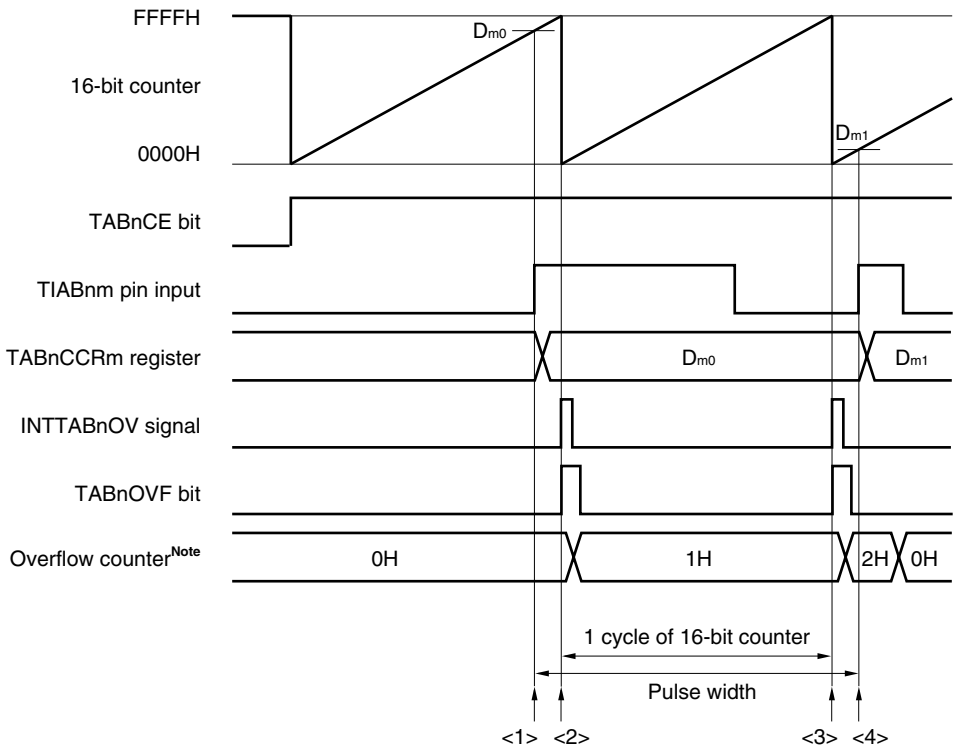
The following problem may occur when a long pulse width in the free-running timer mode.

<1> Read the TABnCCRm register (setting of the default value of the TIABnm pin input).
<2> An overflow occurs. Nothing is done by software.
<3> An overflow occurs a second time. Nothing is done by software.
<4> Read the TABnCCRm register.
Read the overflow flag. If the overflow flag is 1, clear it to 0.
Because the overflow flag is 1, the pulse width can be calculated by (10000H + D_{m1} – D_{m0}) (incorrect).
Actually, the pulse width must be (20000H + D_{m1} – D_{m0}) because an overflow occurs twice.

If an overflow occurs twice or more when the capture trigger interval is long, the correct pulse width may not be obtained.

If the capture trigger interval is long, slow the count clock to lengthen one cycle of the 16-bit counter, or use software. An example of how to use software is shown next.

Example when capture trigger interval is long



Note The overflow counter is set arbitrarily by software on the internal RAM.

<1> Read the TABnCCRm register (setting of the default value of the TIABnm pin input).

<2> An overflow occurs. Increment the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.

<3> An overflow occurs a second time. Increment (+1) the overflow counter and clear the overflow flag to 0 in the overflow interrupt servicing.

<4> Read the TABnCCRm register.

Read the overflow counter.

→ When the overflow counter is “N”, the pulse width can be calculated by ($N \times 10000H + D_{m1} - D_{m0}$).

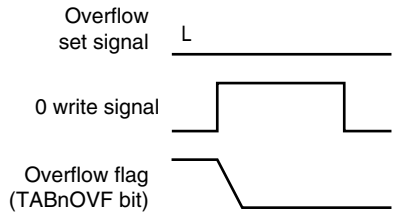
In this example, the pulse width is ($20000H + D_{m1} - D_{m0}$) because an overflow occurs twice.

Clear the overflow counter (0H).

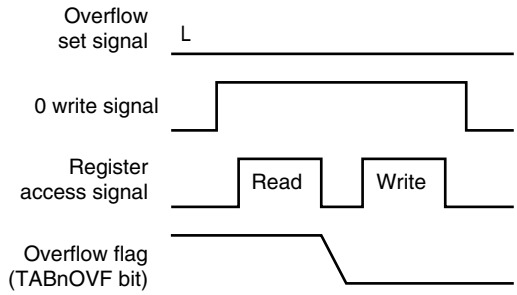
(e) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TABnOVF bit to 0 with the CLR instruction and by writing 8-bit data (bit 0 is 0) to the TABnOPT0 register. To accurately detect an overflow, read the TABnOVF bit when it is 1, and then clear the overflow flag by using a bit manipulation instruction.

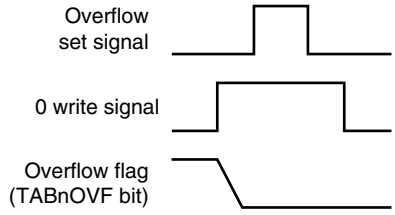
(i) Operation to write 0 (without conflict with setting)



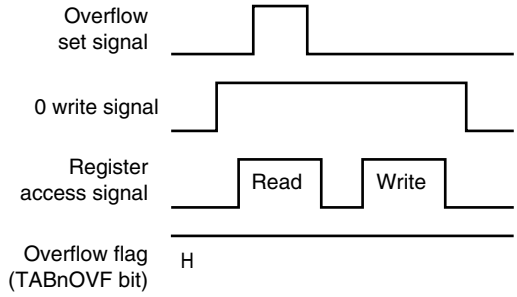
(iii) Operation to clear to 0 (without conflict with setting)



(ii) Operation to write 0 (conflict with setting)



(iv) Operation to clear to 0 (conflict with setting)



To clear the overflow flag to 0, read the overflow flag to check if it is set to 1, and clear it with the CLR instruction. If 0 is written to the overflow flag without checking if the flag is 1, the set information of overflow may be erased by writing 0 ((ii) in the above chart). Therefore, software may judge that no overflow has occurred even when an overflow actually has occurred.

If execution of the CLR instruction conflicts with occurrence of an overflow when the overflow flag is cleared to 0 with the CLR instruction, the overflow flag remains set even after execution of the clear instruction.

8.5.7 Pulse width measurement mode (TABnMD2 to TABnMD0 bits = 110)

In the pulse width measurement mode, 16-bit timer/event counter AB starts counting when the TABnCTL0.TABnCE bit is set to 1. Each time the valid edge input to the TIABnm pin has been detected, the count value of the 16-bit counter is stored in the TABnCCRm register, and the 16-bit counter is cleared to 0000H.

The interval of the valid edge can be measured by reading the TABnCCRm register after a capture interrupt request signal (INTTABnCCm) occurs.

Select either of the TIABn0 to TIABn3 pins as the capture trigger input pin. Specify “No edge detected” by using the TABnIOC1 register for the unused pins.

When an external clock is used as the count clock, measure the pulse width of the TIABnk pin because the external clock is fixed to the TIABn0 pin. At this time, clear the TABnIOC1.TABnIS1 and TABnIOC1.TABnIS0 bits to 00 (capture trigger input (TIABn0 pin): No edge detected).

Remark m = 0 to 3,
k = 1 to 3

Figure 8-34. Configuration in Pulse Width Measurement Mode

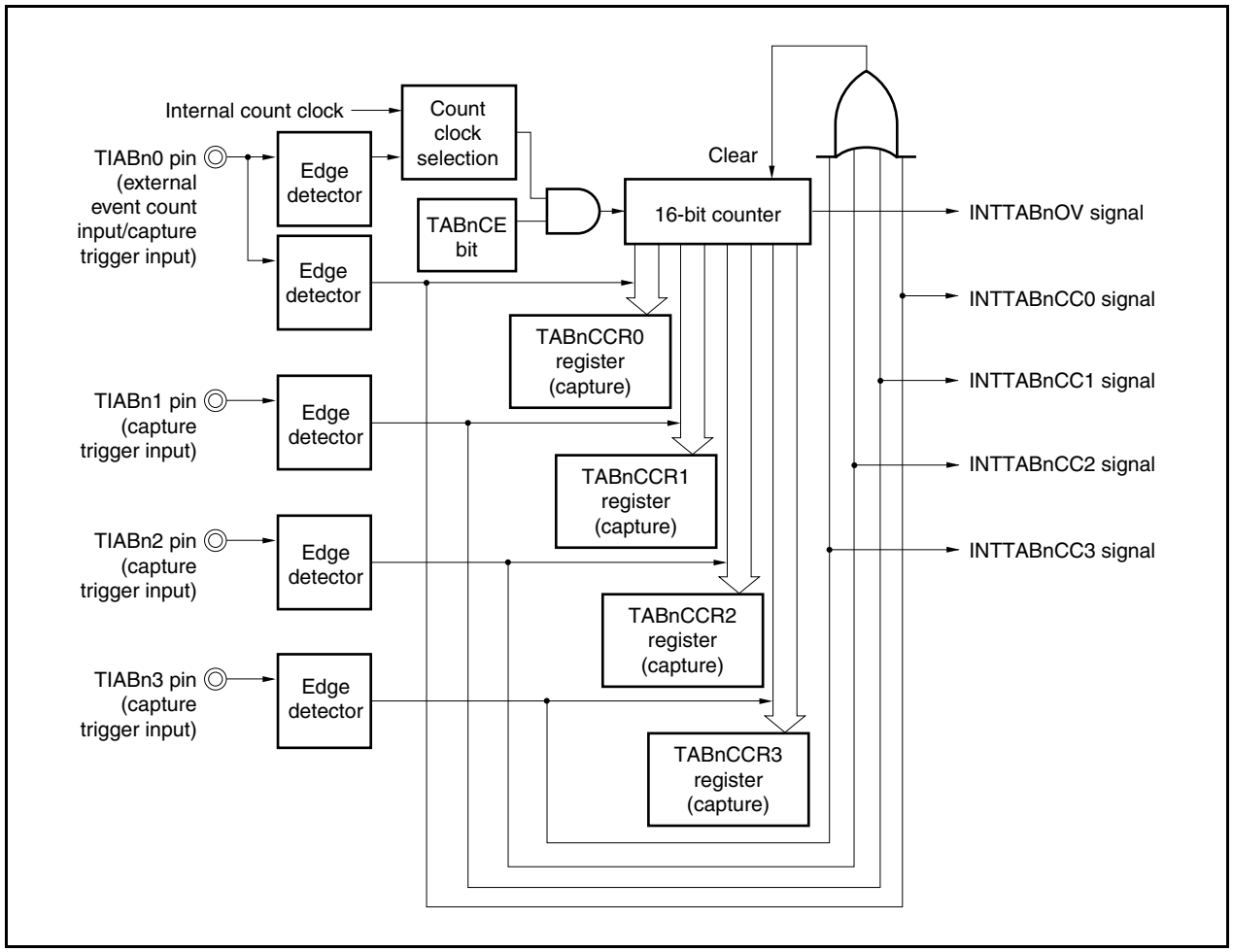
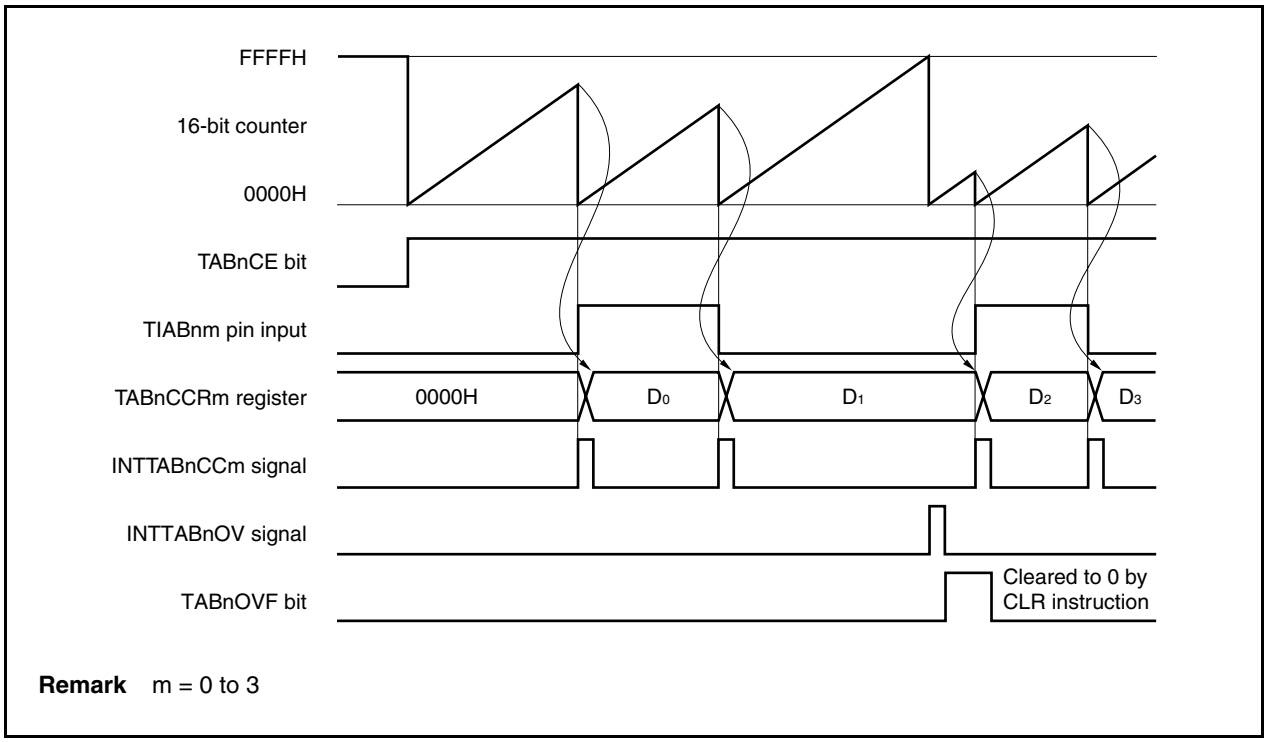


Figure 8-35. Basic Timing in Pulse Width Measurement Mode



Remark m = 0 to 3

When the TABnCE bit is set to 1, the 16-bit counter starts counting. When the valid edge input to the TIABnm pin is later detected, the count value of the 16-bit counter is stored in the TABnCCRm register, the 16-bit counter is cleared to 0000H, and a capture interrupt request signal (INTTABnCCm) is generated.

The pulse width is calculated as follows.

Pulse width = Captured value × Count clock cycle

If the valid edge is not input to the TIABnm pin even when the 16-bit counter counted up to FFFFH, an overflow interrupt request signal (INTTABnOV) is generated at the next count clock, and the counter is cleared to 0000H and continues counting. At this time, the overflow flag (TABnOPT0.TABnOVF bit) is also set to 1. Clear the overflow flag to 0 by executing the CLR instruction via software.

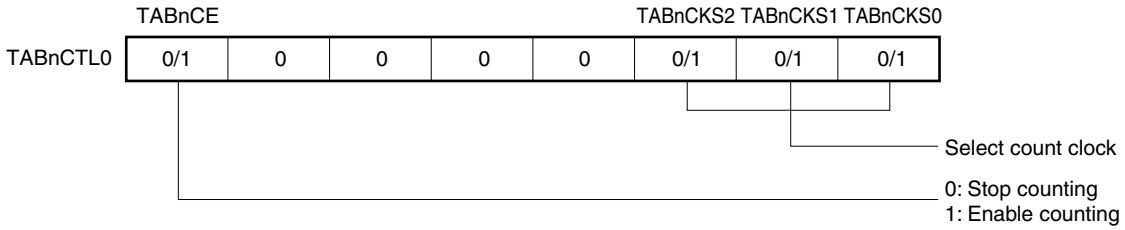
If the overflow flag is set to 1, the pulse width can be calculated as follows.

Pulse width = (10000H × TABnOVF bit set (1) count + Captured value) × Count clock cycle

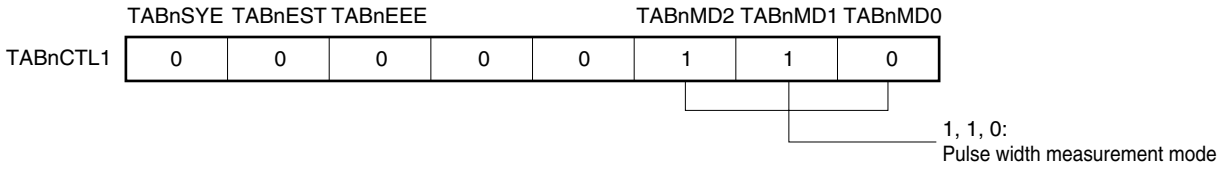
Remark m = 0 to 3

Figure 8-36. Register Setting in Pulse Width Measurement Mode (1/2)

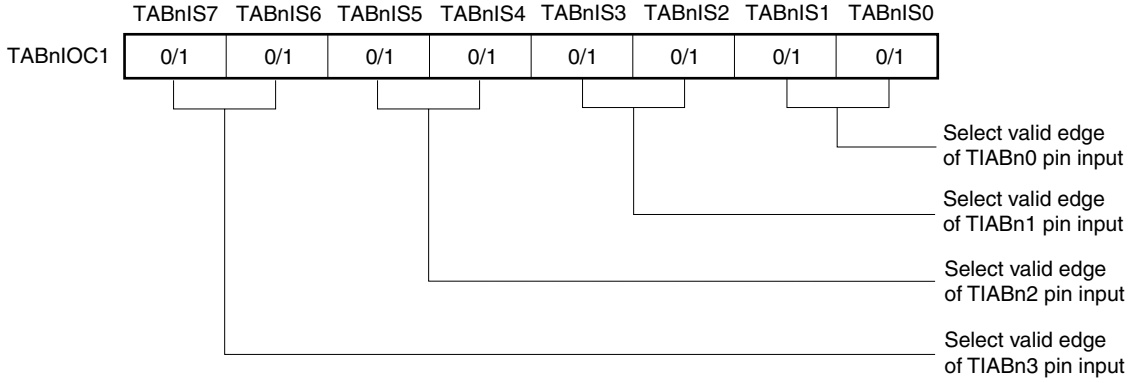
(a) TABn control register 0 (TABnCTL0)



(b) TABn control register 1 (TABnCTL1)



(c) TABn I/O control register 1 (TABnIOC1)



(d) TABn option register 0 (TABnOPT0)

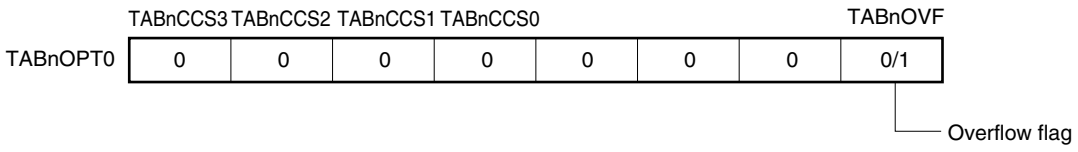


Figure 8-36. Register Setting in Pulse Width Measurement Mode (2/2)

(e) TABn counter read buffer register (TABnCNT)

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

(f) TABn capture/compare registers 0 to 3 (TABnCCR0 to TABnCCR3)

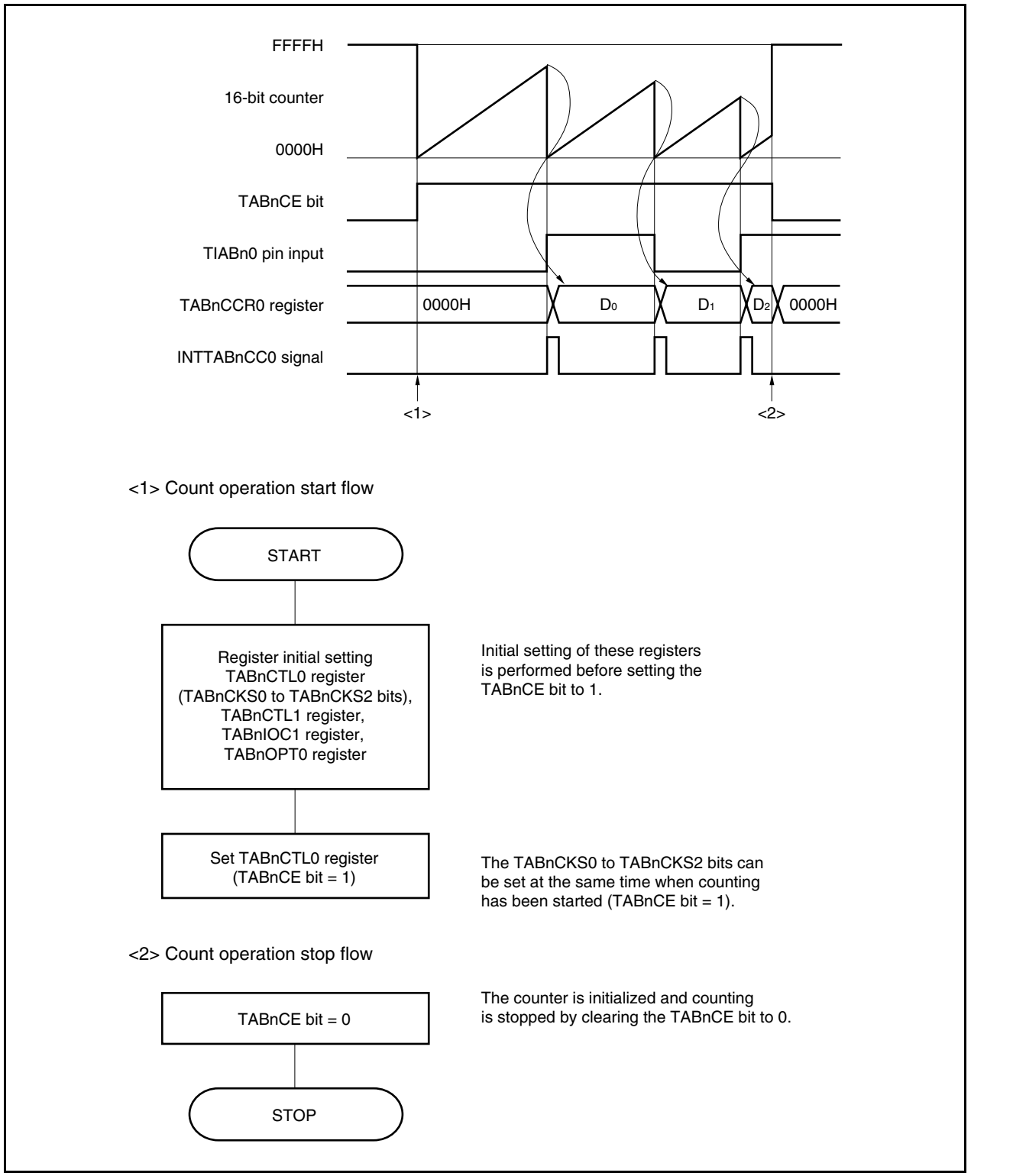
These registers store the count value of the 16-bit counter when the valid edge input to the TIABnm pin is detected.

Remarks 1. TABn I/O control register 0 (TABnIOC0) and TABn I/O control register 2 (TABnIOC2) are not used in the pulse width measurement mode.

2. m = 0 to 3

(1) Operation flow in pulse width measurement mode

Figure 8-37. Software Processing Flow in Pulse Width Measurement Mode

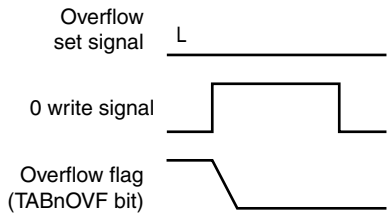


(2) Operation timing in pulse width measurement mode

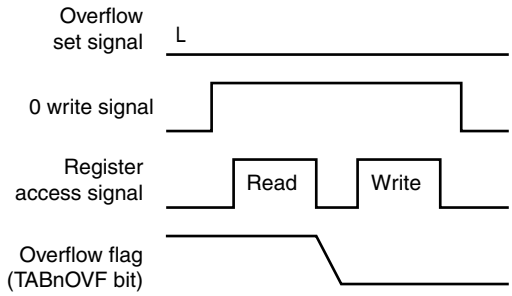
(a) Clearing overflow flag

The overflow flag can be cleared to 0 by clearing the TABnOVF bit to 0 with the CLR instruction and by writing 8-bit data (bit 0 is 0) to the TABnOPT0 register. To accurately detect an overflow, read the TABnOVF bit when it is 1, and then clear the overflow flag by using a bit manipulation instruction.

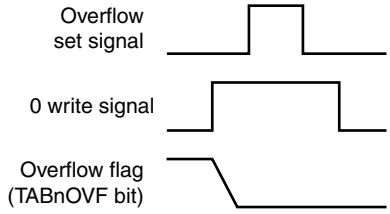
(i) Operation to write 0 (without conflict with setting)



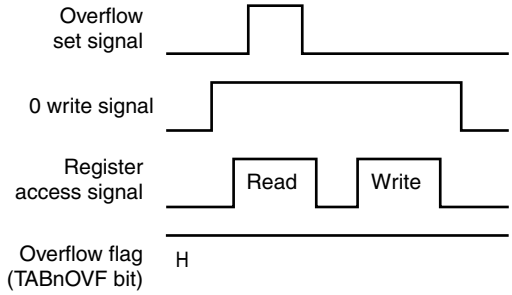
(iii) Operation to clear to 0 (without conflict with setting)



(ii) Operation to write 0 (conflict with setting)



(iv) Operation to clear to 0 (conflict with setting)



To clear the overflow flag to 0, read the overflow flag to check if it is set to 1, and clear it with the CLR instruction. If 0 is written to the overflow flag without checking if the flag is 1, the set information of overflow may be erased by writing 0 ((ii) in the above chart). Therefore, software may judge that no overflow has occurred even when an overflow actually has occurred.

If execution of the CLR instruction conflicts with occurrence of an overflow when the overflow flag is cleared to 0 with the CLR instruction, the overflow flag remains set even after execution of the clear instruction.

8.5.8 Triangular wave PWM mode (TABnMD2 to TABnMD0 = 111)

In the triangular wave PWM mode, TABn capture/compare register k (TABnCCRk) is used to set the duty factor, and TABn capture/compare register 0 (TABnCCR0) is used to set the cycle.

By using these four registers and operating the timer, triangular wave PWM with a variable cycle is output.

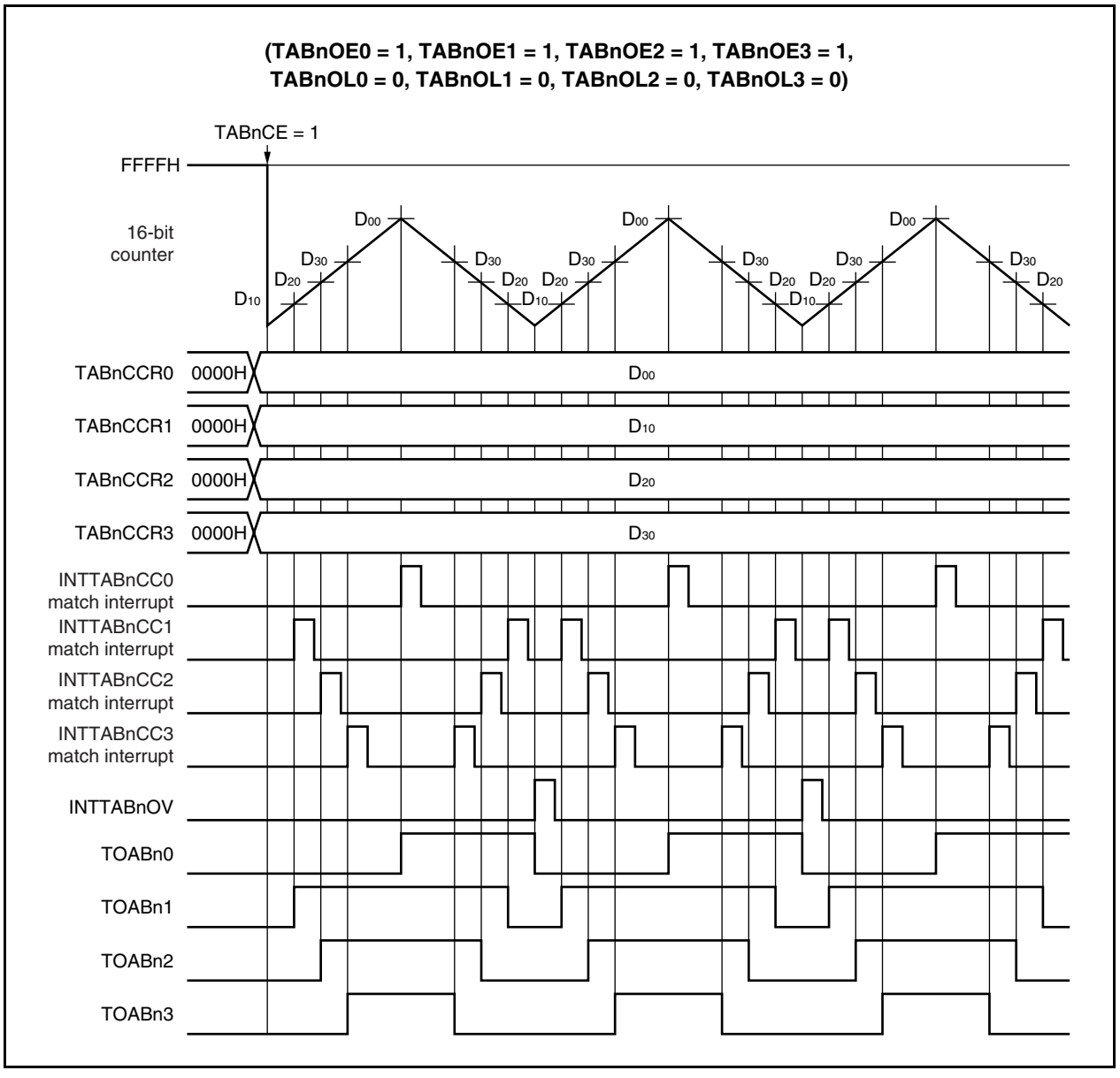
The value of the TABnCCRm register can be rewritten when TABnCE = 1.

To stop timer AB, clear TABnCE to 0. The waveform of PWM is output from the TOABnk pin. The TOABn0 pin produces a toggle output when the value of the 16-bit counter matches the value of the TABnCCR0 register and when the counter underflows.

Caution In the PWM mode, the capture function of the TABnCCRm register cannot be used because this register can be used only as a compare register.

Remark m = 0 to 3, k = 1 to 3

Figure 8-38. Timing of Basic Operation in Triangular Wave PWM Mode



8.5.9 Timer output operations

The following table shows the operations and output levels of the TOABn0 to TOABn3 pins.

Table 8-7. Timer Output Control in Each Mode

Operation Mode	TOABn0 Pin	TOABn1 Pin	TOABn2 Pin	TOABn3 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	–			
Triangular wave PWM output mode	Square wave output	Triangular PWM output	Triangular PWM output	Triangular PWM output

Table 8-8. Truth Table of TOABn0 to TOABn3 Pins Under Control of Timer Output Control Bits

TABnIOC0.TABnOLm Bit	TABnIOC0.TABnOEm Bit	TABnCTL0.TABnCE Bit	Level of TOABnm 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

8.6 Timer-Tuned Operation Function

Timer AA and timer AB have a timer-tuned operation function.
The timers that can be synchronized are listed in Table 8-9.

Table 8-9. Tuned Operation Function of Timers

Master Timer	Slave Timer
TAA0	TAA1
TAA2	TAA3
TAB0	TAA4
TAB1	TAB2

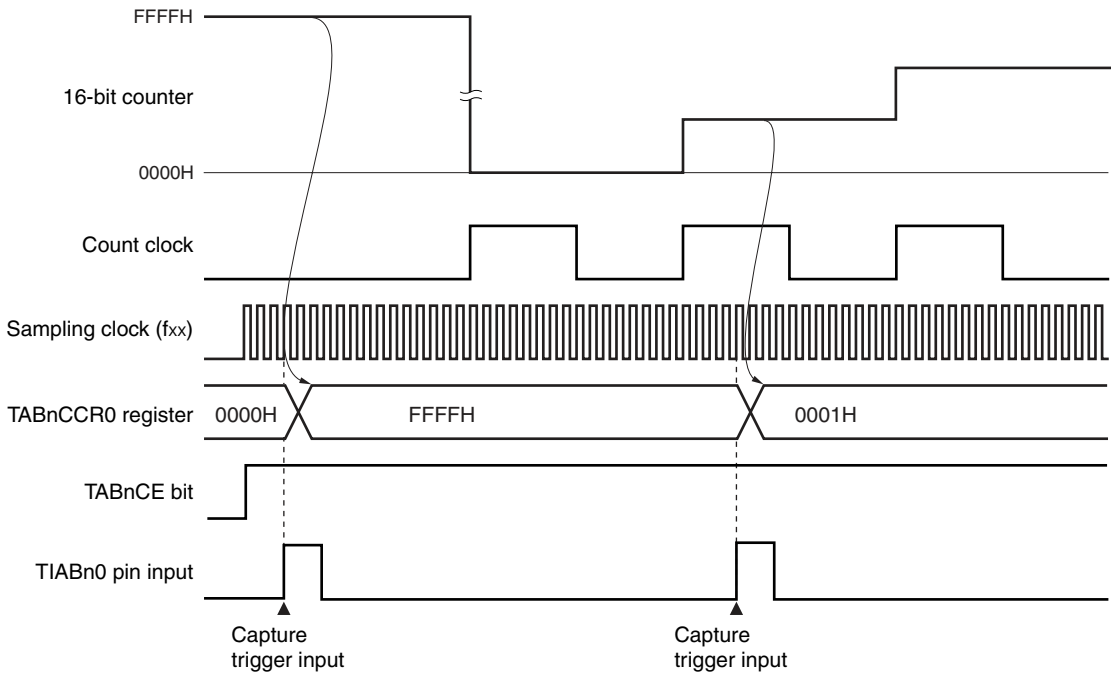
For details of the timer-tuned operation function, see **7.6 Timer-Tuned Operation Function**.

8.7 Cautions

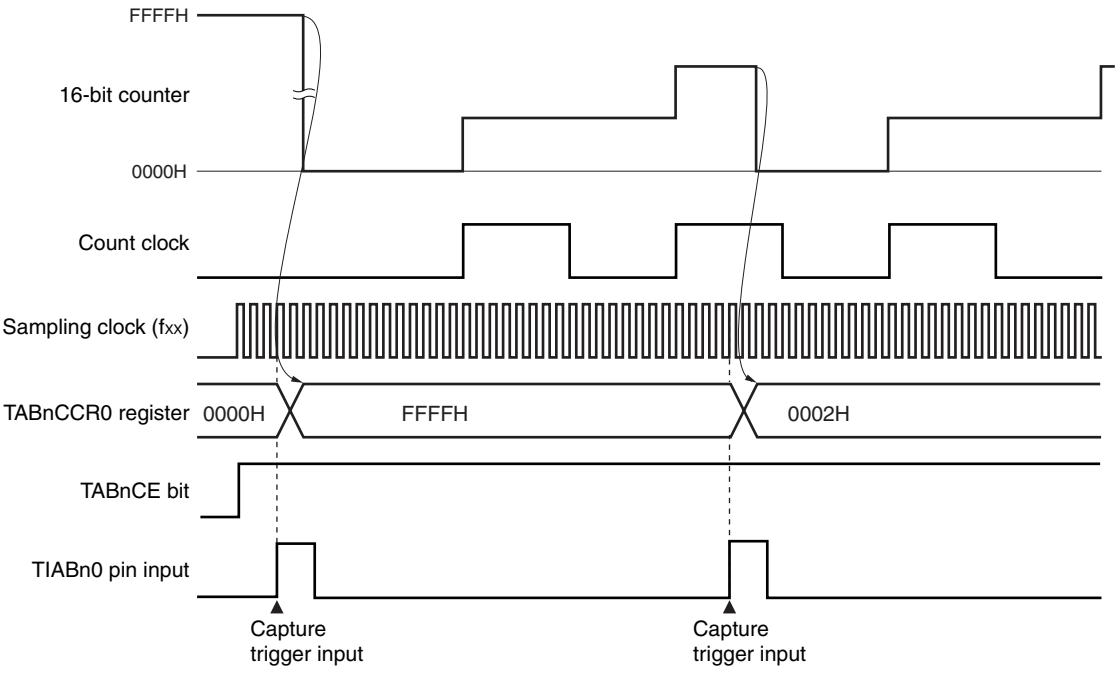
(1) Capture operation

When the capture operation is used and a slow clock is selected as the count clock, FFFFH, not 0000H, may be captured in the TABnCCR0, TABnCCR1, TABnCCR2, and TABnCCR3 registers if the capture trigger is input immediately after the TABnCE bit is set to 1.

(a) Free-running timer mode



(b) Pulse width measurement mode



CHAPTER 9 16-BIT INTERVAL TIMER M (TMM)

9.1 Overview

- Interval function
- 8 clocks selectable
- 16-bit counter × 1
  (The 16-bit counter cannot be read during timer count operation.)
- Compare register × 1
  (The compare register cannot be written during timer counter operation.)
- Compare match interrupt × 1

Timer M supports only the clear & start mode. The free-running timer mode is not supported.

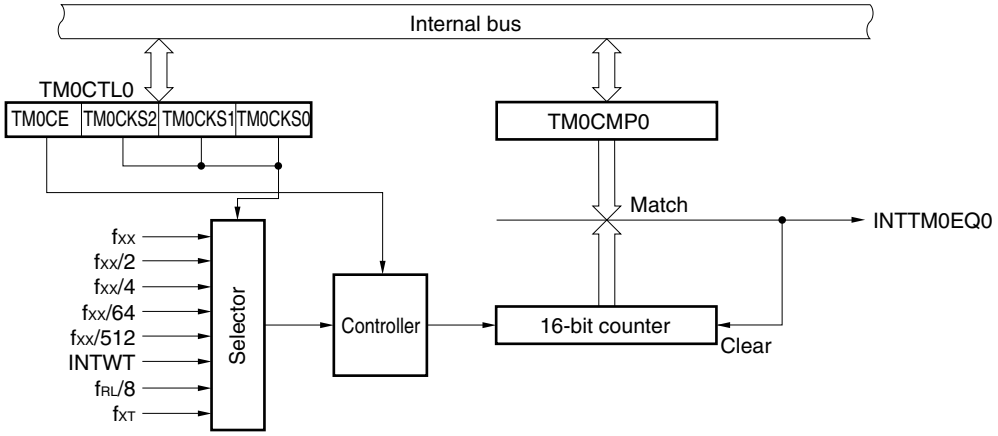
9.2 Configuration

TMM0 includes the following hardware.

Table 9-1. Configuration of TMM0

Item	Configuration
Timer register	16-bit counter
Register	TMM0 compare register 0 (TM0CMP0)
Control register	TMM0 control register 0 (TM0CTL0)

Figure 9-1. Block Diagram of TMM0



Remark f_{XX}: Main clock frequency
f_{RL}: Low-speed internal oscillation clock frequency
f_{XT}: Subclock frequency
INTWT: Watch timer interrupt request signal

(1) 16-bit counter

This is a 16-bit counter that counts the internal clock.

The 16-bit counter cannot be read or written.

(2) TMM0 compare register 0 (TM0CMP0)

The TM0CMP0 register is a 16-bit compare register.

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

Reset sets this register to 0000H.

The same value can always be written to the TM0CMP0 register by software.

TM0CMP0 register rewrite is prohibited when the TM0CTL0.TM0CE bit = 1.

After reset: 0000H R/W Address: FFFFF694H

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TM0CMP0																

9.3 Register

(1) TMM0 control register (TM0CTL0)

The TM0CTL0 register is an 8-bit register that controls the TMM0 operation.

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

Reset sets this register to 00H.

The same value can always be written to the TM0CTL0 register by software. Rewriting this register, except the TM0CE bit, is prohibited while the timer is operating.

After reset: 00H R/W Address: FFFFF690H

	<7>	6	5	4	3	2	1	0
TM0CTL0	TM0CE	0	0	0	0	TM0CKS2	TM0CKS1	TM0CKS0

TM0CE	Internal clock operation enable/disable specification
0	TMM0 operation disabled (16-bit counter reset asynchronously). Operation clock application stopped.
1	TMM0 operation enabled. Operation clock application started. TMM0 operation started.
The internal clock control and internal circuit reset for TMM0 are performed asynchronously with the TM0CE bit. When the TM0CE bit is cleared to 0, the internal clock of TMM0 is 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	$f_{RL}/8$
1	1	1	f_{XT}

Cautions 1. Set the TM0CKS2 to TM0CKS0 bits when TM0CE bit = 0.
When changing the value of TM0CE from 0 to 1, it is not possible to set the value of the TM0CKS2 to TM0CKS0 bits simultaneously.
2. Be sure to clear bits 3 to 6 to "0".

Remark f_{XX}: Main clock frequency
f_{RL}: Low-speed internal oscillation clock frequency
f_{XT}: Subclock frequency

9.4 Operation

Caution Do not set the TM0CMP0 register to FFFFH.

9.4.1 Interval timer mode

In the interval timer mode, an interrupt request signal (INTTM0EQ0) is generated at the specified interval if the TM0CTL0.TM0CE bit is set to 1.

Figure 9-2. Configuration of Interval Timer

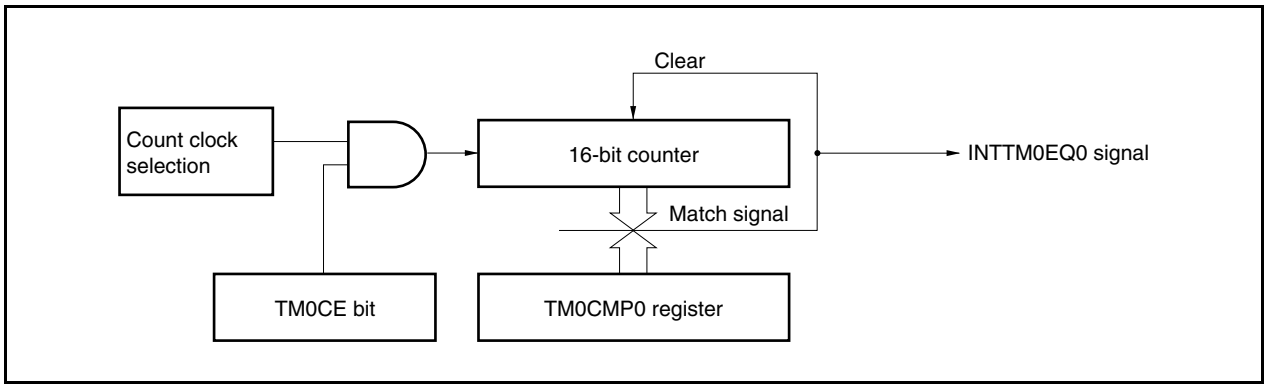
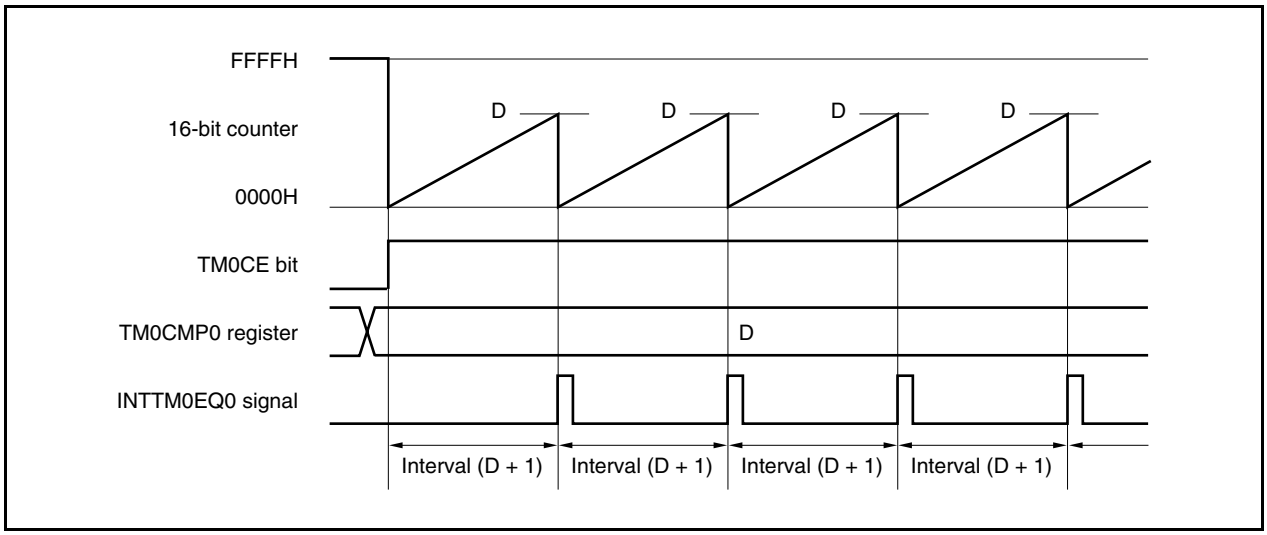


Figure 9-3. Basic Timing of Operation in Interval Timer Mode



When the TM0CE bit is set to 1, the value of the 16-bit counter is cleared from FFFFH to 0000H in synchronization with the count clock, and the counter starts counting.

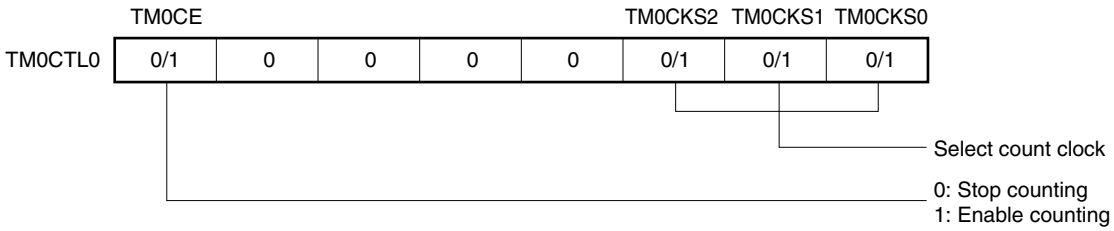
When the count value of the 16-bit counter matches the value of the TM0CMP0 register, the 16-bit counter is cleared to 0000H and a compare match interrupt request signal (INTTM0EQ0) is generated.

The interval can be calculated by the following expression.

Interval = (Set value of TM0CMP0 register + 1) × Count clock cycle

Figure 9-4. Register Setting for Interval Timer Mode Operation

(a) TMM0 control register 0 (TM0CTL0)



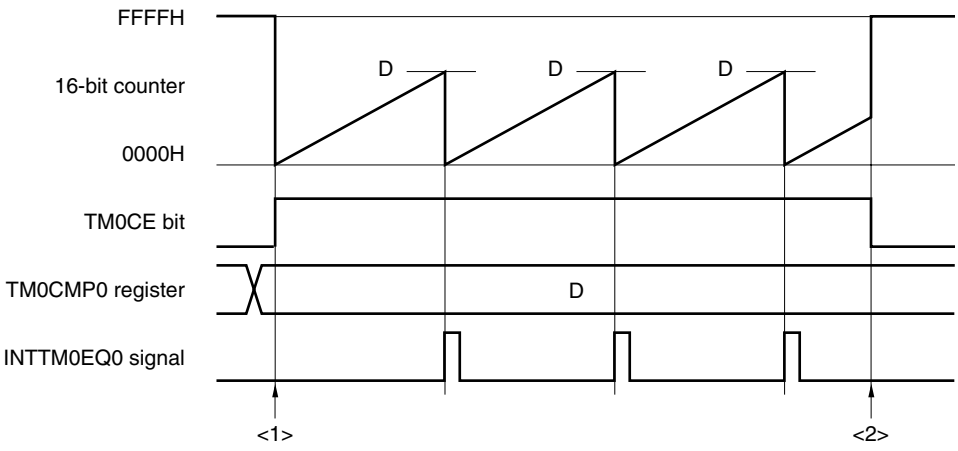
(b) TMM0 compare register 0 (TM0CMP0)

If the TM0CMP0 register is set to D, the interval is as follows.

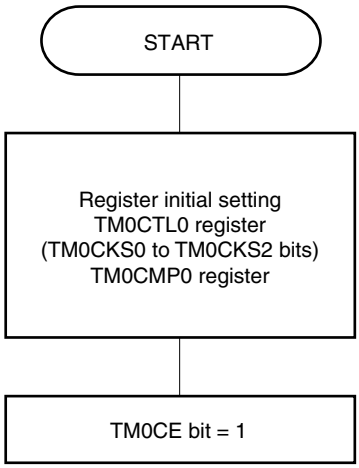
Interval = (D + 1) × Count clock cycle

(1) Interval timer mode operation flow

Figure 9-5. Software Processing Flow in Interval Timer Mode



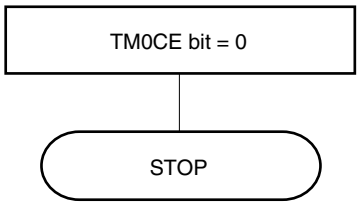
<1> Count operation start flow



Initial setting of these registers is performed before setting the TM0CE bit to 1.

Setting the TM0CKS0 to TM0CKS2 bits is prohibited at the same time when counting has been started (TM0CE bit = 1).

<2> Count operation stop flow



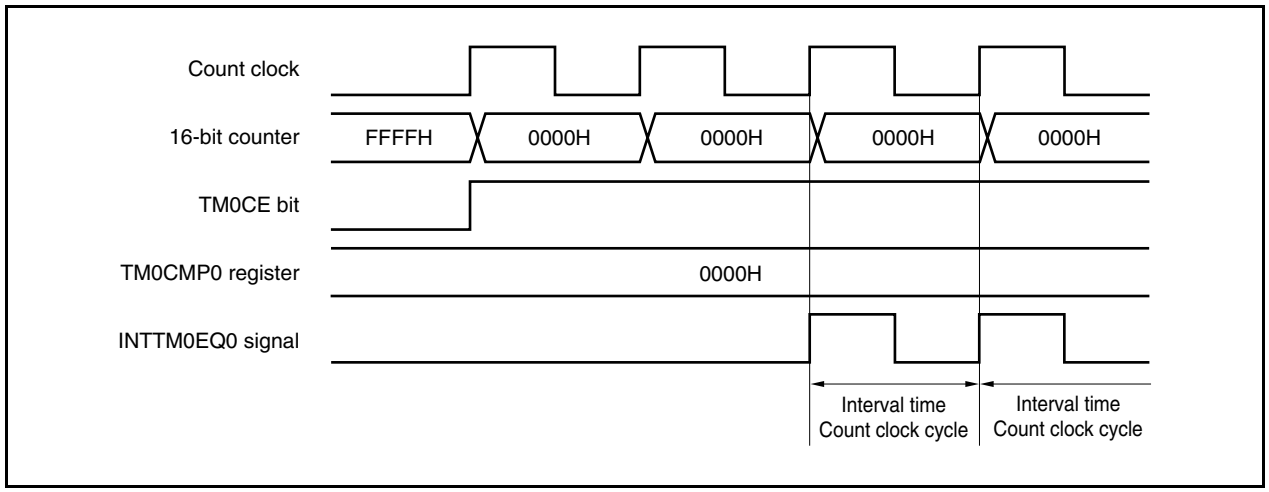
The counter is initialized and counting is stopped by clearing the TM0CE bit to 0.

(2) Interval timer mode operation timing

Caution Do not set the TM0CMP0 register to FFFFH.

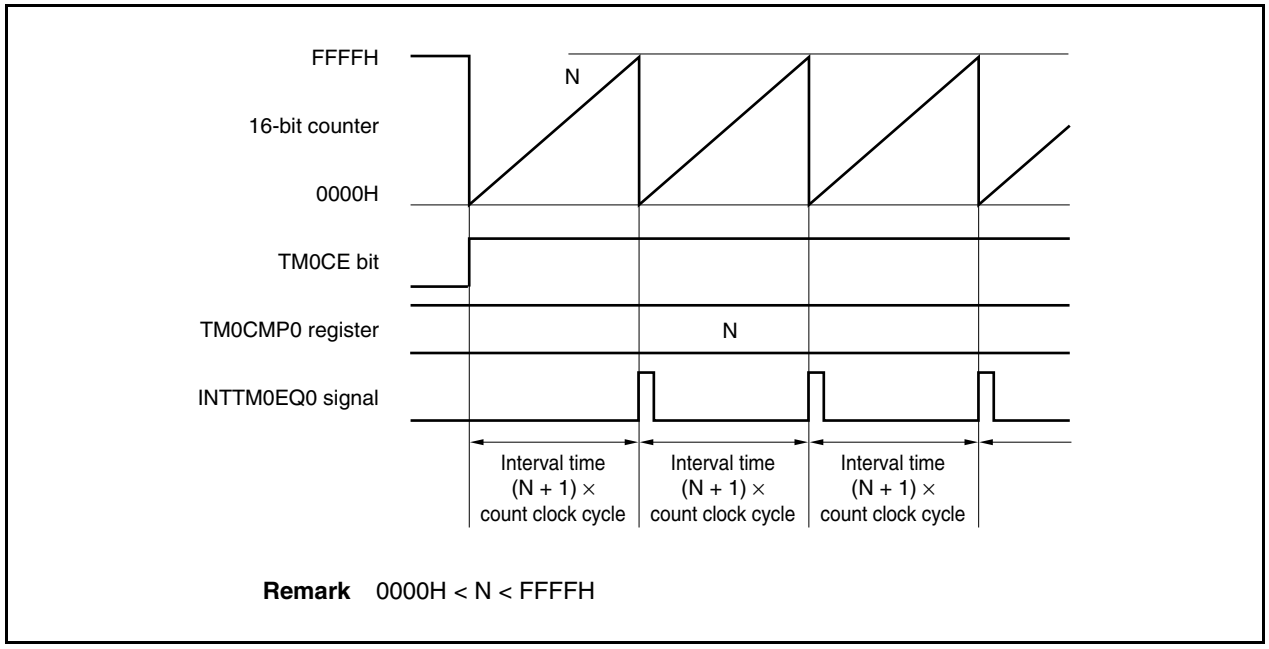
(a) Operation if TM0CMP0 register is set to 0000H

If the TM0CMP0 register is set to 0000H, the INTTM0EQ0 signal is generated at each count clock. The value of the 16-bit counter is always 0000H.



(b) Operation if TM0CMP0 register is set to N

If the TM0CMP0 register is set to N, the 16-bit counter counts up to N. The counter is cleared to 0000H in synchronization with the next count-up timing and the INTTM0EQ0 signal is generated.



Remark 0000H < N < FFFFH

9.4.2 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 Start
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
$f_{RL}/8$	$16/f_{RL}$
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 TM0CE bit is 1, the operation cannot be guaranteed.
If they are rewritten by mistake, clear the TM0CTL0.TM0CE bit to 0, and re-set the registers.

CHAPTER 10 MOTOR CONTROL FUNCTION

10.1 Functional Overview

Timer AB0 (TAB0) and the TAB0 option (TABOP0) can be used as an inverter function that controls a motor. It performs a tuning operation with timer AA4 (TAA4) and A/D conversion of the A/D converter can be started when the value of TAB0 matches the value of TAA4. The following operations can be performed as motor control functions.

- 6-phase PWM output function with 16-bit accuracy (with dead-timer, for upper and lower arms)
- Timer tuning operation function (tunable with TAA4)
- Cycle setting function (cycle can be changed during operation of crest or valley interrupt)
- Compare register rewriting: Anytime rewrite, batch rewrite, or intermittent rewrite (selectable during TAB0 operation)
- Interrupt and transfer culling functions
- Dead-time setting function
- A/D trigger timing function of the A/D converter (four types of timing can be generated)
- 0% output and 100% output available
- 0% output and 100% output selectable by crest interrupt and valley interrupt
- Forced output stop function
  - At valid edge detection by external pin input (INTP1, INTP3)

10.2 Configuration

The motor control function consists of the following hardware.

Item	Configuration
Timer register	Dead-time counters 1 to 3
Compare register	TAB0 dead-time compare register (TAB0DTC register)
Control registers	TAB0 option register 0 (TAB0OPT0) TAB0 option register 1 (TAB0OPT1) TAB0 option register 2 (TAB0OPT2) TAB0 I/O control register 3 (TAB0IOC3) High-impedance output control registers 0, 1 (HZA0CTL0, HZA0CTL1)

- 6-phase PWM output can be produced with dead time by using the output of TAB0 (TOAB01, TOAB02, TOAB03)
- The output level of the 6-phase PWM output can be set individually.
- The 16-bit timer/counter of TAB0 counts up/down triangular waves. When the timer/counter underflows and when a cycle match occurs, an interrupt is generated. Interrupt generation, however, can be suppressed up to 31 times.
- TAA4 can execute counting at the same time as TAB0 (timer tuning operation function). TAA4 can be set in four ways as it can generate two types of A/D trigger sources (INTTAA4CC0 and INTTAA4CC1), and two types of interrupts: on underflow interrupt (INTTAB0OV) and cycle match interrupt (INTTAB0CC0).

Figure 10-1. Block Diagram of Motor Control

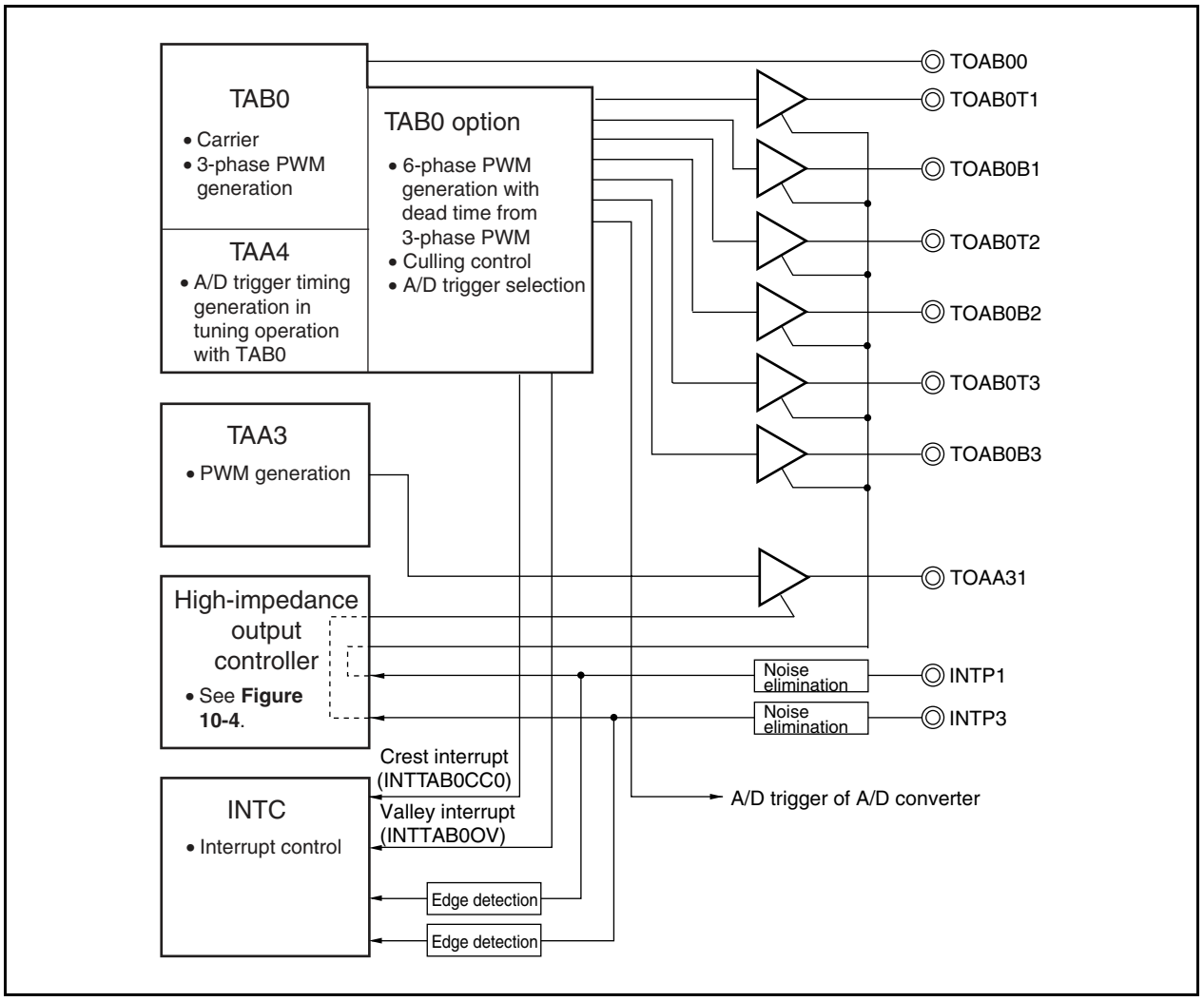
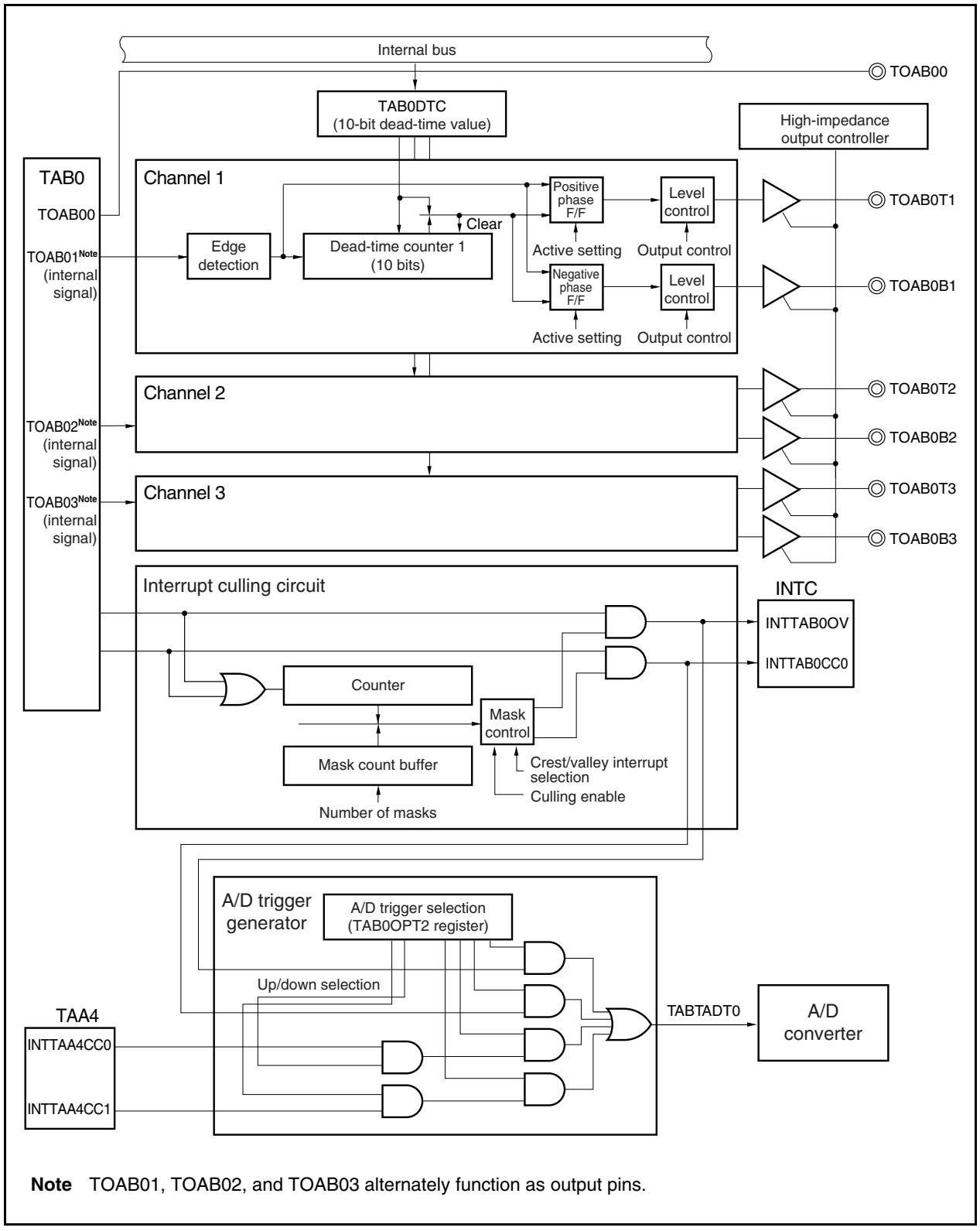


Figure 10-2. TAB0 Option



Note TOAB01, TOAB02, and TOAB03 alternately function as output pins.

(1) TAB0 dead-time compare register (TAB0DTC)

The TAB0DTC register is a 10-bit compare register that specifies a dead-time value.

Rewriting this register is prohibited when the TAB0CTL0.TAB0CE bit = 1.

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

Reset sets this register to 0000H.

After reset: 0000H R/W Address: FFFFF564H

	15	10	9	0
TAB0DTC	000000		TAB0DTC9 to TAB0DTC0	

(2) Dead-time counters 1 to 3

The dead-time counters are 10-bit counters that count dead time.

These counters are cleared or count up at the rising or falling edge of the TOAB0m output signal by TAB0, and are cleared or stopped when their count value matches the value of the TAB0DTC register. The count clock of these counters is the same as that set by the TAB0CTL0.TAB0CKS2 to TAB0CTL0.TAB0CKS0 bits of TAB0.

Remarks 1. The operation differs when the TAB0OPT2.TAB0DTM bit = 1. For details, see **10.4.2 (4) Automatic dead-time width narrowing function (TAB0OPT2.TAB0DTM bit = 1)**.

2. m = 1 to 3

10.3 Control Registers

(1) TAB0 option register 0 (TAB0OPT0)

The TAB0OPT0 register is an 8-bit register that controls the timer AB0 option function.

This register can be read or written in 8-bit or 1-bit units. However, the TAB0CUF bit is read-only.

Reset sets this register to 00H.

Caution The TAB0CMS and TAB0CUF bits can be set only in the 6-phase PWM output mode. Be sure to clear these bits to 0 when TAB0 is used alone.

After reset: 00H R/W Address: FFFFF545H

	7	6	5	4	3	2	1	<0>
TAB0OPT0	TAB0CCS3 Note 1	TAB0CCS2 Note 1	TAB0CCS1 Note 1	TAB0CCS0 Note 1	0	TAB0CMS	TAB0CUF	TAB0OVF Note 2

TAB0CMS	Compare register rewrite mode selection
0	Batch rewrite mode (transfer operation)
1	Anytime rewrite mode
• The TAB0CMS bit is valid only when the 6-phase PWM output mode is set (when the TAB0CTL1.TAB0MD2 to TAB0CTL1.TAB0MD0 bits = 111). Clear the TAB0CMS bit to 0 in any other mode. • The TAB0CMS bit can be rewritten while the timer is operating (when the TAB0CTL0.TAB0CE bit = 1). • The following compare registers are rewritten in the batch write mode. TAB0CCR0 to TAB0CCR3, TAA4CCR0, TAA4CCR1, and TAB0OPT1 registers	

TAB0CUF	Count-up/Count-down flag of timer AB0
0	Timer AB0 is counting up.
1	Timer AB0 is counting down.
The TAB0CUF bit is valid only when the 6-phase PWM output mode is set (when the TAB0CTL1.TAB0MD2 to TAB0CTL1.TAB0MD0 bits = 111).	

Notes 1. Be sure to set the TAB0CCS3 to TAB0CCS0 bits to “0” in the 6-phase PWM output mode.

2. For details of the TAB0OVF bit, see **CHAPTER 8 16-BIT TIMER/EVENT COUNTER AB (TAB)**.

(2) TAB0 option register 1 (TAB0OPT1)

The TAB0OPT1 register is an 8-bit register that controls the interrupt request signal generated by the timer AB0 option function.

This register can be rewritten when the TAB0CTL0.TAB0CE bit is 1.

Two rewrite modes (batch write mode and anytime write mode) can be selected, depending on the setting of the TAB0OPT0.TAB0CMS bit.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: FFFFF560H

	<7>	<6>	5	4	3	2	1	0
TAB0OPT1	TAB0ICE	TAB0IOE	0	TAB0ID4	TAB0ID3	TAB0ID2	TAB0ID1	TAB0ID0

TAB0ICE	Crest interrupt (INTTAB0CC0 signal) enable
0	Do not use INTTAB0CC0 signal (do not use it as count signal for interrupt culling).
1	Use INTTAB0CC0 signal (use it as count signal for interrupt culling).

TAB0IOE	Valley interrupt (INTTAB0OV signal) enable
0	Do not use INTTAB0OV signal (do not use it as count signal for interrupt culling).
1	Use INTTAB0OV signal (use it as count signal for interrupt culling).

TAB0ID4	TAB0ID3	TAB0ID2	TAB0ID1	TAB0ID0	Number of times of interrupt
0	0	0	0	0	Not culled (all interrupts are output)
0	0	0	0	1	1 masked (one of two interrupts is output)
0	0	0	1	0	2 masked (one of three interrupts is output)
0	0	0	1	1	3 masked (one of four interrupts is output)
:	:	:	:	:	:
1	1	1	0	0	28 masked (one of 29 interrupts is output)
1	1	1	0	1	29 masked (one of 30 interrupts is output)
1	1	1	1	0	30 masked (one of 31 interrupts is output)
1	1	1	1	1	31 masked (one of 32 interrupts is output)

(3) TAB0 option register 2 (TAB0OPT2)

The TAB0OPT2 register is an 8-bit register that controls the timer AB0 option function.

This register can be rewritten when the TAB0CTL0.TAB0CE bit is 1. However, rewriting the TAB0DTM bit is prohibited when the TAB0CE bit is 1. The same value can be rewritten.

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

Reset sets this register to 00H.

(1/2)

After reset: 00H R/W Address: FFFFF561H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TAB0OPT2	TAB0RDE	TAB0DTM	TAB0ATM3	TAB0ATM2	TAB0AT3	TAB0AT2	TAB0AT1	TAB0AT0

TAB0RDE	Transfer culling enable
0	Do not cull transfer (transfer timing is generated every time at crest and valley).
1	Cull transfer at the same interval as interrupt culling set by the TAB0OPT1 register.

TAB0DTM	Dead-time counter operation mode selection
0	Dead-time counter counts up normally and, if TOAB0m output of TAB0 is at a narrow interval (TOAB0m output width < dead-time width), the dead-time counter is cleared and counts up again.
1	Dead-time counter counts up normally and, if TOAB0m output of TAB0 is at a narrow interval (TOAB0m output width < dead-time width), the dead-time counter counts down and the dead-time control width is automatically narrowed.
Rewriting the TAB0DTM bit is disabled during timer operation. If it is rewritten by mistake, stop the timer operation by clearing the TAB0CE bit to 0, and re-set the TAB0DTM bit.	

TAB0ATM3	TAB0ATM3 mode selection
0	Output A/D trigger signal (TABTADT0) for INTTAA4CC1 interrupt while dead-time counter is counting up.
1	Output A/D trigger signal (TABTADT0) for INTTAA4CC1 interrupt while dead-time counter is counting down.

TAB0ATM2	TAB0ATM2 mode selection
0	Output A/D trigger signal (TABTADT0) for INTTAA4CC0 interrupt while dead-time counter is counting up.
1	Output A/D trigger signal (TABTADT0) for INTTAA4CC0 interrupt while dead-time counter is counting down.

Caution When using interrupt culling (the TAB0OPT1.TAB0ID4 to TAB0OPT1.TAB0ID0 bits are set to other than 00000), be sure to set the TAB0RDE bit to 1.
Therefore, the interrupt and transfer are generated at the same timing. The interrupt and transfer cannot be set separately. If the interrupt and transfer are set separately (TAB0RDE bit = 0), transfer is not performed normally.

Remark m = 1 to 3

(2/2)

TAB0AT3[Note]	A/D trigger output control 3
0	Disable output of A/D trigger signal (TABTADT0) for INTTAA4CC1 interrupt.
1	Enable output of A/D trigger signal (TABTADT0) for INTRAA4CC1 interrupt.

TAB0AT2[Note]	A/D trigger output control 2
0	Disable output of A/D trigger signal (TABTADT0) for INTTAA4CC0 interrupt.
1	Enable output of A/D trigger signal (TABTADT0) for INTTAA4CC0 interrupt.

TAB0AT1[Note]	A/D trigger output control 1
0	Disable output of A/D trigger signal (TABTADT0) for INTTAB0CC0 (crest interrupt).
1	Enable output of A/D trigger signal (TABTADT0) for INTTAB0CC0 (crest interrupt).

TAB0AT0[Note]	A/D trigger output control 0
0	Disable output of A/D trigger signal (TABTADT0) for INTTAB0OV (valley interrupt).
1	Enable output of A/D trigger signal (TABTADT0) for INTTAB0OV (valley interrupt).

Note For the setting of the TAB0AT3 to TAB0AT0 bits, see **10.4.6 A/D conversion start trigger output function**.

(4) TAB0 I/O control register 3 (TAB0IOC3)

The TAB0IOC3 register is an 8-bit register that controls the output of the timer AB0 option function.

To output from the TOAB0Tm pin, set the TAB0IOC0.TAB0OEm bit to 1 and then set the TAB0IOC3 register.

The TAB0IOC3 register can be rewritten only when the TAB0CTL0.TAB0CE bit is 0.

Rewriting each bit of the TAB0IOC3 register is prohibited when the TAB0CTL0.TAB0CE bit is 1; however the same value can be rewritten to each bit of the TAB0IOC3 register when the TAB0CTL0.TAB0CE bit is 1.

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

Reset sets this register to A8H.

Caution Set the TAB0IOC3 register to the default value (A8H) when the timer is used in a mode other than the 6-phase PWM output mode.

Remark Set the output level of the TOAB0Tm pin by the TAB0IOC0 register.

After reset: A8H R/W Address: FFFFF562H

	<7>	<6>	<5>	<4>	<3>	<2>	1	0
TAB0IOC3	TAB0OLB3	TAB0OEB3	TAB0OLB2	TAB0OEB2	TAB0OLB1	TAB0OEB1	0	0

TAB0OLBm	Setting of TOAB0Bm pin output level
0	Disable inversion of output of TOAB0Bm pin
1	Enable inversion of output of TOAB0Bm pin

TAB0OEBm	TOAB0Bm pin output
0	Disable TOAB0Bm pin output. • When TAB0OLBm bit = 0, low level is output from TOAB0Bm pin. • When TAB0OLBm bit = 1, high level is output from TOAB0Bm pin.
1	Enable TOAB0Bm pin output.

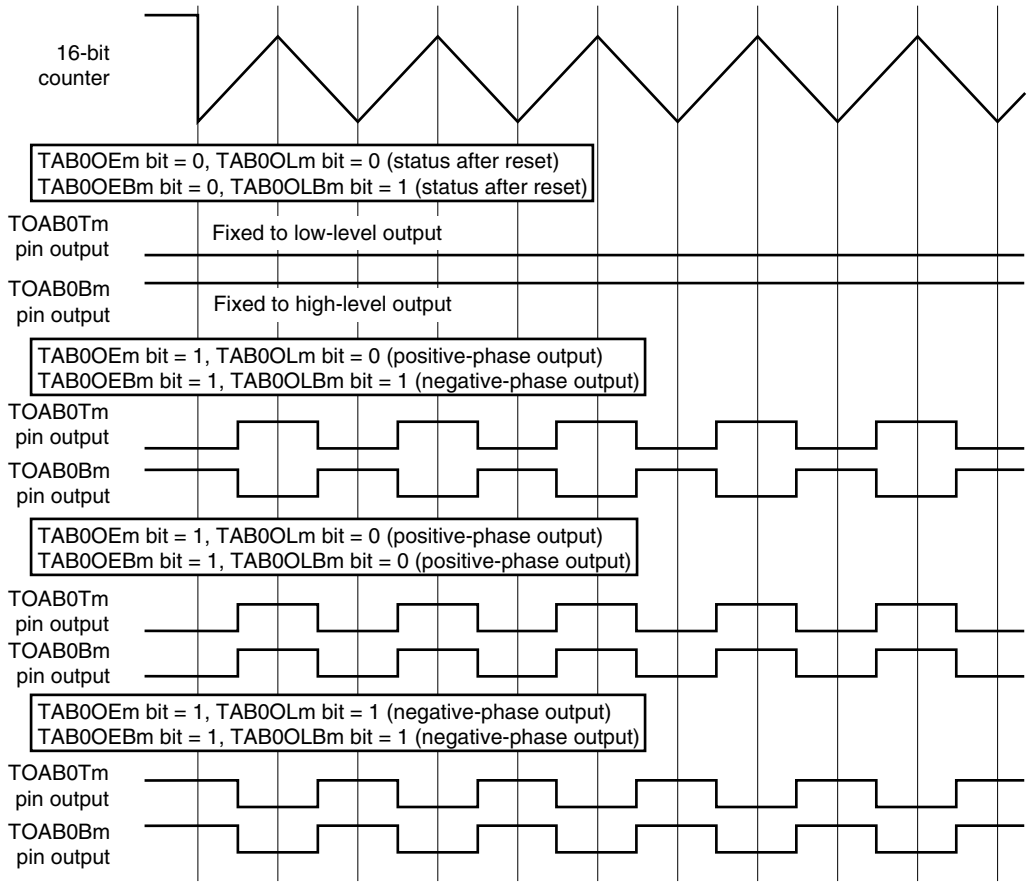
Remark m = 1 to 3

(a) Output from TOAB0Tm and TOAB0Bm pins

The TOAB0Tm pin output is controlled by the TAB0IOC0.TAB0OLm and TAB0IOC0.TAB0OEm bits. The TOAB0Bm pin output is controlled by the TAB0IOC3.TAB0OLBm and TAB0IOC3.TAB0OEBm bits.

The timer output with each setting in the 6-phase PWM output mode is shown below.

Figure 10-3. Output Control of TOAB0Tm and TOAB0Bm Pins



Remark m = 1 to 3

Table 10-1. TOAB0Tm Pin Output

TAB0OLm Bit	TAB0OEm Bit	TAB0CE Bit	TOAB0Tm Pin Output
0	0	x	Low-level output
	1	0	Low-level output
		1	TOAB0Tm positive-phase output
1	0	x	High-level output
	1	0	High-level output
		1	TOAB0Tm negative-phase output

Remark m = 1 to 3

Table 10-2. TOAB0Bm Pin Output

TAB0OLBm Bit	TAB0OEBm Bit	TAB0CE Bit	TOAB0Bm Pin Output
0	0	x	Low-level output
	1	0	Low-level output
		1	TOAB0Bm positive-phase output
1	0	x	High-level output
	1	0	High-level output
		1	TOAB0Bm negative-phase output

Remark m = 1 to 3

(5) High-impedance output control registers 0, 1 (HZA0CTL0, HZA0CTL1)

The HZA0CTL0 and HZA0CTL1 registers are 8-bit registers that control the high-impedance state of the output buffer.

These registers can be read or written in 8-bit or 1-bit units. However, the HZA0DCF0 and HZA0DCF1 bits are read-only bits and cannot be written.

16-bit access is not possible.

Reset sets these registers to 00H.

The same value can be always rewritten to the HZA0CTL0 and HZA0CTL1 registers by software.

The relationship between detection factor and the control registers is shown below.

Pins Subject to High-Impedance Control	High-Impedance Control Factor	Control Register
	External Pin	
When TOAB0T1 to TOAB0T3 are output When TOAB0B1 to TOAB0B3 are output	INTP1	HZA0CTL0
When TOAA31 is output	INTP3	HZA0CTL1

Caution High impedance control is performed only when the target port is specified as a target pin in the above table.

(1/2)

After reset: 00H R/W Address: HZA0CTL0 FFFFF570H, HZA0CTL1 FFFFF571H

HZA0CTLn (n = 0, 1)

<7>	<6>	5	4	<3>	<2>	1	<0>
HZA0DCEn	HZA0DCMn	HZA0DCNn	HZA0DCPn	HZA0DCTn	HZA0DCCn	0	HZA0DCFn

HZA0DCEn	High-impedance output control
0	Disable high-impedance output control operation. Pins can function as output pins.
1	Enable high-impedance output control operation.

HZA0DCMn	Condition of clearing high-impedance state by HZA0DCCn bit
0	Setting of the HZA0DCCn bit is valid regardless of the external pin[Note] input.
1	Setting of the HZA0DCCn bit is invalid while the external pin[Note] holds a level detected as abnormal (active level).
Rewrite the HZA0DCMn bit when the HZA0DCEn bit = 0.	

HZA0DCNn	HZA0DCPn	External pin[Note] input edge specification
0	0	No valid edge (setting the HZA0DCFn bit by external pin[Note] input is prohibited).
0	1	Rising edge of the external pin[Note] is valid (abnormality is detected by rising edge input).
1	0	Falling edge of the external pin[Note] is valid (abnormality is detected by falling edge input).
1	1	Setting prohibited

- Rewrite the HZA0DCNn and HZA0DCPn bits when the HZA0DCEn bit is 0.
- For the valid edge specification of the interrupt of the INTP1 and INTP3 pins, see **CHAPTER 18 INTERRUPT/EXCEPTION FUNCTION**.
- High-impedance output control is performed when the valid edge is input after the operation is enabled (by setting HZA0DCEn bit to 1). If the external pin[Note] is at the active level when the operation is enabled, therefore, high-impedance output control is not performed.

HZA0DCTn	High-impedance output trigger bit
0	No operation
1	Pins are made to go into a high-impedance state by software and the HZA0DCFn bit is set to 1.

- If an edge indicating abnormality is input to the external pin[Note] (which is detected according to the setting of the HZA0DCNn and HZA0DCPn bits), the HZA0DCTn bit is invalid even if it is set to 1.
- The HZA0DCTn bit is always 0 when it is read because it is a software-triggered bit.
- The HZA0DCTn bit is invalid even if it is set to 1 when the HZA0DCEn bit = 0.
- Simultaneously setting the HZA0DCTn and HZA0DCCn bits to 1 is prohibited.

Note HZA0CTL0: INTP1 pin, HZA0CTL1: INTP3 pin

(2/2)

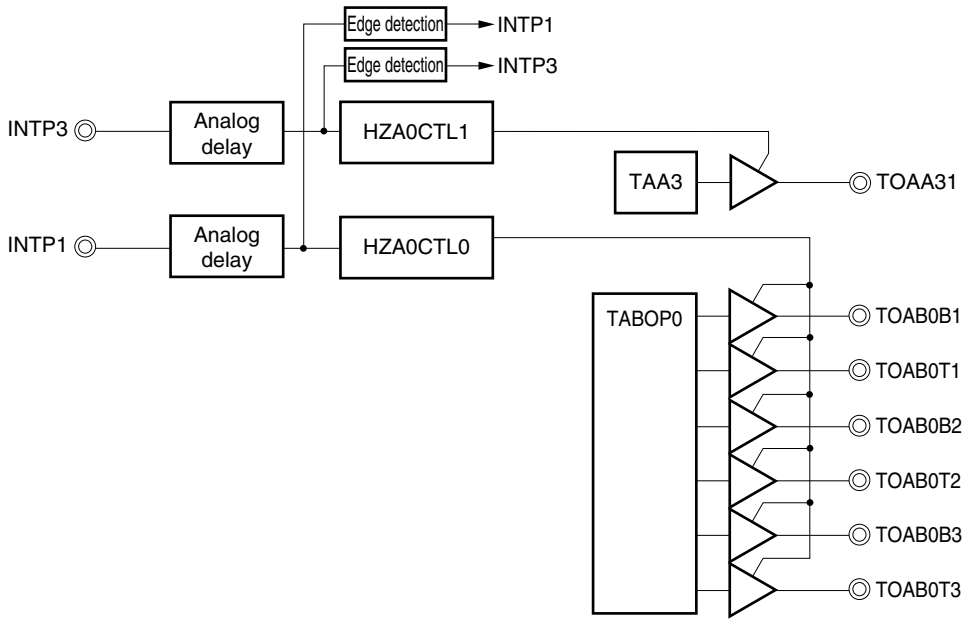
HZA0DCCn	High-impedance output control clear bit
0	No operation
1	Pins that have gone into a high-impedance state are output-enabled by software and the HZA0DCFn bit is cleared to 0.

- Pins can function as output pins when the HZA0DCM bit = 0, regardless of the status of the external pin[Note].
- If an edge indicating abnormality is input to the external pin[Note] (which is set by the HZA0DCNn and HZA0DCPn bits) when the HZA0DCM bit = 1, the HZA0DCCn bit is invalid even if it is set to 1.
- The HZA0DCCn bit is always 0 when it is read.
- The HZA0DCCn bit is invalid even if it is set to 1 when the HZA0DCEn bit = 0.
- Simultaneously setting the HZA0DCTn and HZA0DCCn bits to 1 is prohibited.

HZA0DCFn	High-impedance output status flag
Clear (0)	Indicates that output of the pin is enabled. • This bit is cleared to 0 when the HZA0DCEn bit = 0. • This bit is cleared to 0 when the HZA0DCCn bit = 1.
Set (1)	Indicates that the pin goes into a high-impedance state. • This bit is set to 1 when the HZA0DCTn bit = 1. • This bit is set to 1 when an edge indicating abnormality is input to the external pin[Note] (which is detected according to the setting of the HZA0DCNn and HZA0DCPn bits).

Note HZA0CTL0: INTP1 pin, HZA0CTL1: INTP3 pin

Figure 10-4. High-Impedance Output Controller Configuration



(a) Setting procedure

(i) Setting of high-impedance control operation

<1> Set the HZA0DCMn, HZA0DCNn, and HZA0DCPn bits.

<2> Set the HZA0DCEn bit to 1 (enable high-impedance control).

(ii) Changing setting after enabling high-impedance control operation

<1> Clear the HZA0DCEn bit to 0 (to stop the high-impedance control operation).

<2> Change the setting of the HZA0DCMn, HZA0DCNn, and HZA0DCPn bits.

<3> Set the HZA0DCEn bit to 1 (to enable the high-impedance control operation again).

(iii) Resuming output when pins are in high-impedance state

If the HZA0DCMn bit is 1, set the HZA0DCCn bit to 1 to clear the high-impedance state after the valid edge of the external pin[Note] is detected. However, the high-impedance state cannot be cleared unless this bit is set while the input level of the external pin[Note] is inactive.

<1> Set the HZA0DCCn bit to 1 (command signal to clear the high-impedance state).

<2> Read the HZA0DCFn bit and check the flag status.

<3> Return to <1> if the HZA0DCFn bit is 1. The input level of the external pin[Note] must be checked. The pin can function as an output pin if the HZA0DCFn bit is 0.

(iv) To make the pin to go into a high-impedance state by software

The HZA0DCTn bit must be set to 1 by software to make the pin to go into a high-impedance state while the input level of the external pin[Note] is inactive. The following procedure is an example in which the setting is not dependent upon the setting of the HZA0DCMn bit.

<1> Set the HZA0DCTn bit to 1 (high-impedance output command).

<2> Read the HZA0DCFn bit to check the flag status.

<3> Return to <1> if the HZA0DCFn bit is 0. The input level of the external pin[Note] must be checked. The pin is in a high-impedance state if the HZA0DCFn bit is 1.

However, if the external pin[Note] is not used with the HZA0DCP1 bit and HZA0DCNn bit cleared to 0, the pin goes into a high-impedance state when the HZA0DCTn bit is set to 1.

Note HZA0CTL0: INTP1 pin, HZA0CTL1: INTP3 pin

10.4 Operation

10.4.1 System outline

(1) Outline of 6-phase PWM output

The 6-phase PWM output mode is used to generate a 6-phase PWM output wave, by using TAB0 and the TAB0 option in combination.

The 6-phase PWM output mode is enabled by setting the TAB0CTL1.TAB0MD2 to TAB0CTL1.TAB0MD0 bits of TAB0 to “111”.

One 16-bit counter and four 16-bit compare registers of TAB0 are used to generate a basic 3-phase wave.

The functions of the compare registers are as follows.

TAA4 can perform a tuning operation with TAB0 to start a conversion trigger source for the A/D converter.

Compare Register	Function	Settable Range
TAB0CCR0 register	Setting of cycle	0002H ≤ m ≤ FFFEH
TAB0CCR1 register	Specifying output width of phase U	0000H ≤ i ≤ m + 1
TAB0CCR2 register	Specifying output width of phase V	0000H ≤ j ≤ m + 1
TAB0CCR3 register	Specifying output width of phase W	0000H ≤ k ≤ m + 1

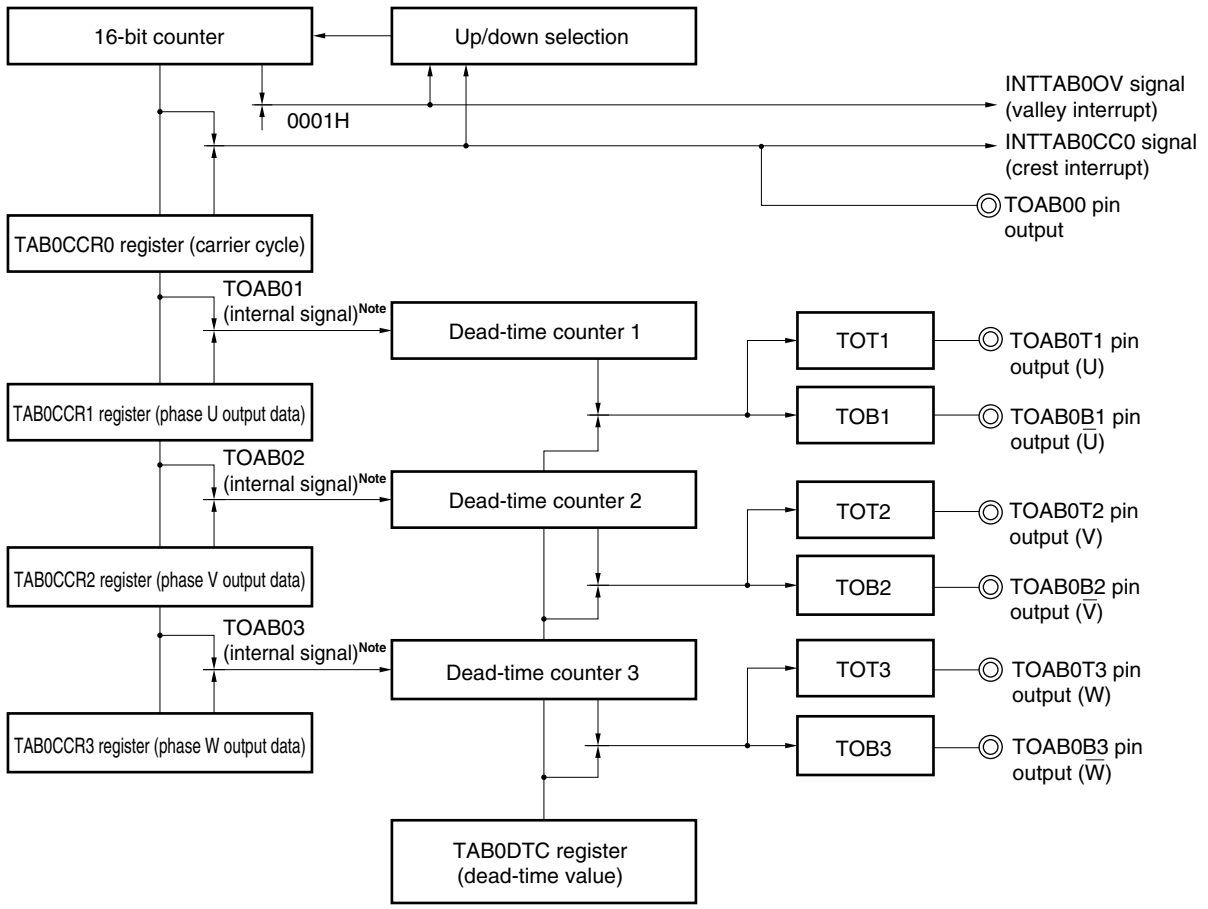
Remark m = Set value of TAB0CCR0 register
i = Set value of TAB0CCR1 register
j = Set value of TAB0CCR2 register
k = Set value of TAB0CCR3 register

A dead-time interval is generated from the basic 3-phase wave generated by using three 10-bit dead-time counters and one compare register to create a wave with a reverse phase to that of the basic 3-phase wave. Then a 6-phase PWM output wave (U, $\overline{U}$, V, $\overline{V}$, W, and $\overline{W}$) is generated.

The 16-bit counter for generating the basic 3-phase wave counts up or down. After the operation has been started, this counter counts up. When its count value matches the cycle set to the TAB0CCR0 register, the counter starts counting down. When the count value matches 0001H, the counter counts up again. This means that a value two times higher than the value set to the TAB0CCR0 register + 1 is the carrier cycle.

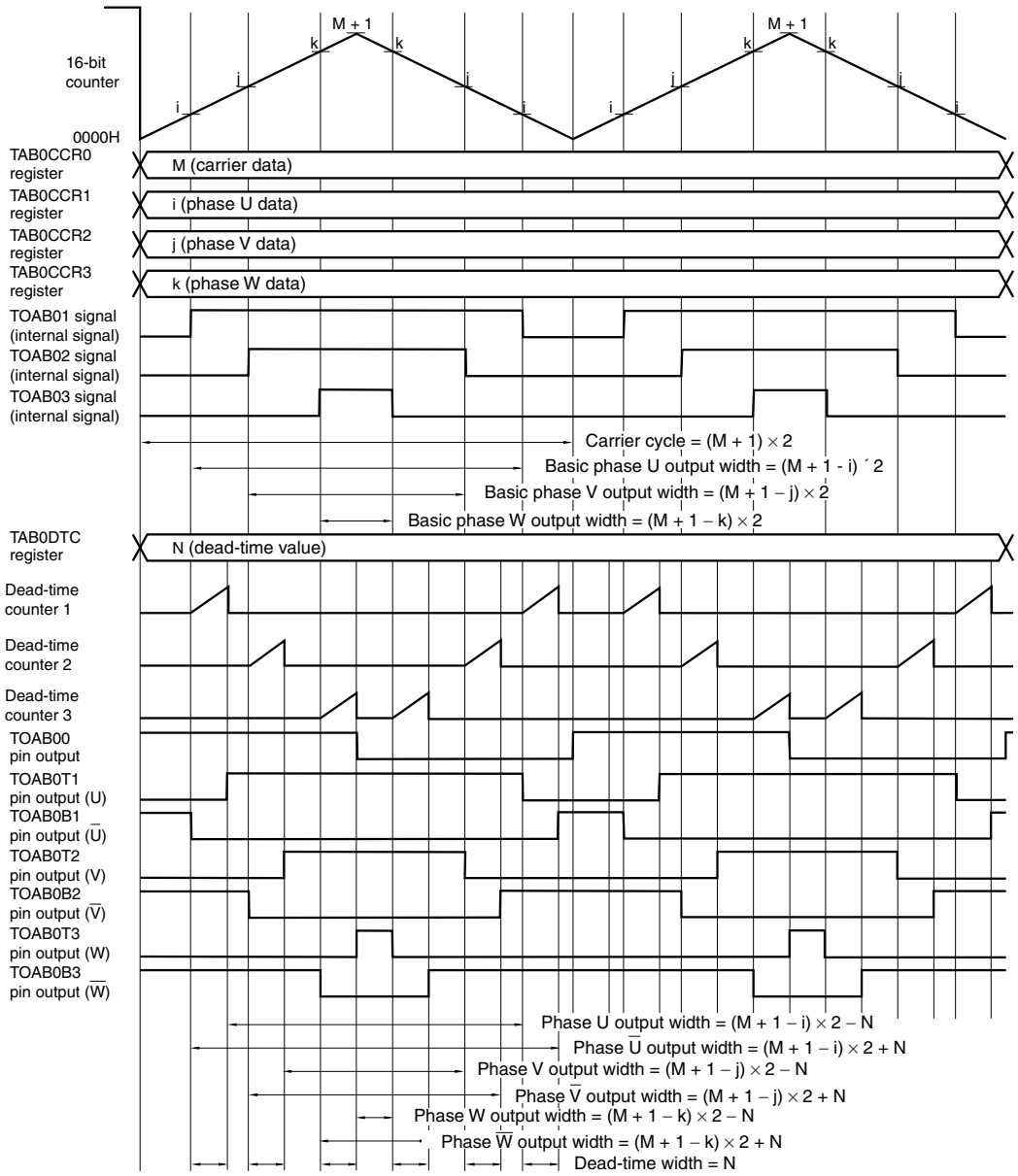
10-bit dead-time counters 1 to 3 that generate the dead-time interval count up. Therefore, the value set to the TAB0 dead-time compare register (TAB0DTC) is used as a dead-time value as is. Because three counters are used, dead time can be generated independently in phases U, V, and W. However, because there is only one register that specifies a dead-time value (TAB0DTC), the same dead-time value is used in the three phases.

Figure 10-5. Outline of 6-Phase PWM Output Mode



Note TOAB01, TOAB02, and TOAB03 alternately function as output pins.

Figure 10-6. Timing Chart of 6-Phase PWM Output Mode



Cautions 1. Set the value "M" of the TAB0CCR0 register in a range of 0002H ≤ M ≤ FFFEH in the 6-phase PWM output mode.

2. Only a value of up to "M + 1" can be set to the TAB0CCR1, TAB0CCR2, and TAB0CCR3 registers.

3. The output is 100% if "0000H" is set to the TAB0CCR1, TAB0CCR2, and TAB0CCR3 registers. The output is 0% if "M + 1" is set to the TAB0CCR1, TAB0CCR2, and TAB0CCR3 registers.

The output (duty 50%) rises at the crest (M + 1) of the 16-bit counter and falls at the valley (0000H) if "M + 2" or higher is set to the TAB0CCR1, TAB0CCR2, and TAB0CCR3 registers.

4. If the operation value of an equation (such as (M + 1 – i) × 2 – N) of the output width of phases U, V, and W is 0 or lower, it is converged to 0 (100% output). If the operation value is higher than "(M + 1) × 2", it is converged to (M + 1) × 2 (0% output).

(2) Interrupt requests

Two types of interrupt requests are available: the INTTAB0CC0 (crest interrupt) signal and INTTAB0OV (valley interrupt) signal.

The INTTAB0CC0 and INTTAB0OV signals can be culled by using the TAB0OPT1 register.

For details of culling interrupts, see **10.4.3 Interrupt culling function**.

- INTTAB0CC0 (crest interrupt) signal: Interrupt signal indicating match between the value of the 16-bit counter that counts up and the value of the TAB0CCR0 register
- INTTAB0OV (valley interrupt) signal: Interrupt signal indicating match between the value of the 16-bit counter that counts down and the value 0001H

(3) Rewriting registers during timer operation

The following registers have a buffer register and can be rewritten in the anytime rewrite mode, batch rewrite mode, or intermittent batch rewrite mode.

Related Unit	Register
Timer AA4	TAA4 capture/compare register 0 (TAA4CCR0) TAA4 capture/compare register 1 (TAA4CCR1)
Timer AB0	TAB0 capture/compare register 0 (TAB0CCR0) TAB0 capture/compare register 1 (TAB0CCR1) TAB0 capture/compare register 2 (TAB0CCR2) TAB0 capture/compare register 3 (TAB0CCR3)
Timer AB0 option	TAB0 option register 1 (TAB0OPT1)

For details of the transfer function of the compare register, see **10.4.4 Operation to rewrite register with transfer function**.

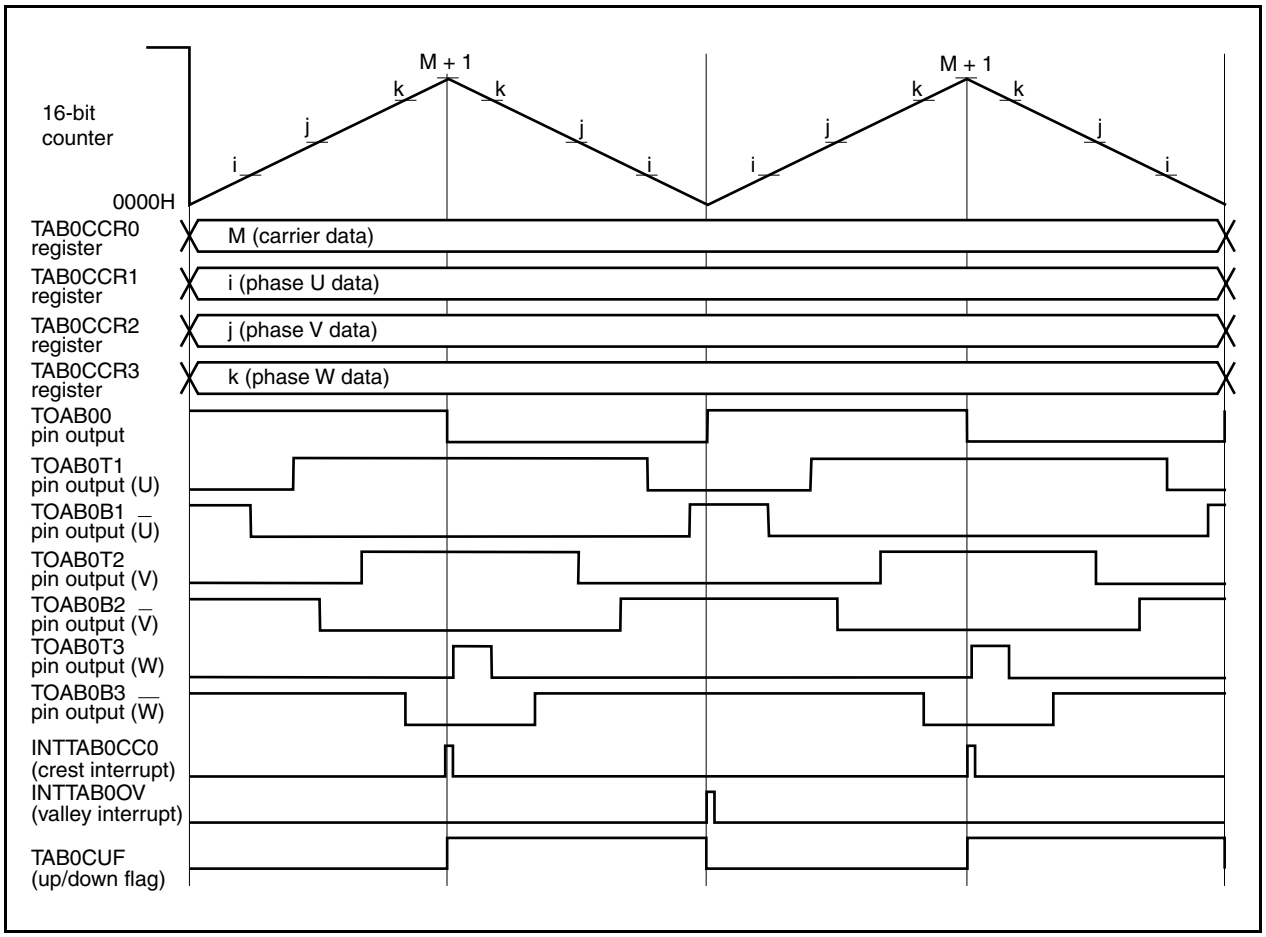
(4) Counting-up/down operation of 16-bit counter

The operation status of the 16-bit counter can be checked by using the TAB0CUF bit of TAB0 option register 0 (TAB0OPT0).

Status of TAB0CUF Bit	Status of 16-Bit Counter	Range of 16-Bit Counter Value
TAB0CUF bit = 0	Counting up	0000H – m
TAB0CUF bit = 1	Counting down	(m + 1) – 0001H

Remark m = Set value of TAB0CCR0 register

Figure 10-7. Interrupt and Up/Down Flag



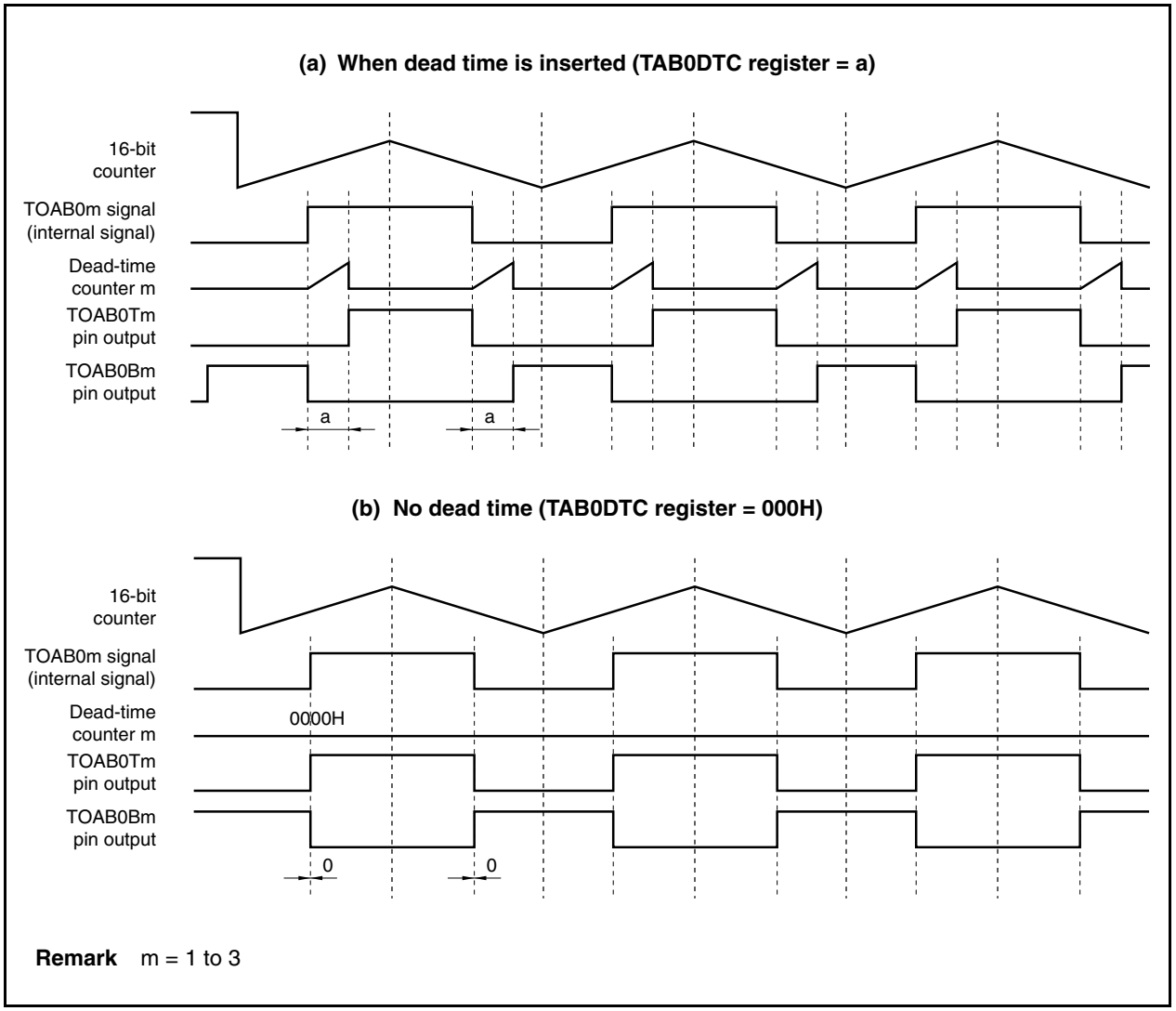
10.4.2 Dead-time control (generation of negative-phase wave signal)

(1) Dead-time control mechanism

In the 6-phase PWM output mode, compare registers 1 to 3 (TAB0CCR1, TAB0CCR2, and TAB0CCR3) are used to set the duty factor, and compare register 0 (TAB0CCR0) is used to set the cycle. By setting these four registers and by starting the operation of TAB, three types of PWM output waves (basic 3-phase waves) with a variable duty factor are generated. These three PWM output waves are input to the timer AB option unit (TABOP0) and their inverted signal with dead-time is created to generate three sets of (six) PWM waves.

The TABOP0 unit consists of three 10-bit counters (dead-time counters 1 to 3) that operate in synchronization with the count clock of TAB0, and a TAB0 dead-time compare register (TAB0DTC) that specifies dead time. If “a” is set to the TAB0DTC register, the dead-time value is “a”, and interval “a” is created between a positive-phase wave and a negative-phase wave.

Figure 10-8. PWM Output Wave with Dead Time (1)



Remark m = 1 to 3

(2) PWM output of 0%/100%

The V850ES/Hx3 is capable of 0% wave output and 100% wave output for PWM output.

A low level is continuously output from TOAB0Tm pin as the 0% wave output. A high level is continuously output from TOAB0Tm pin as the 100% wave output.

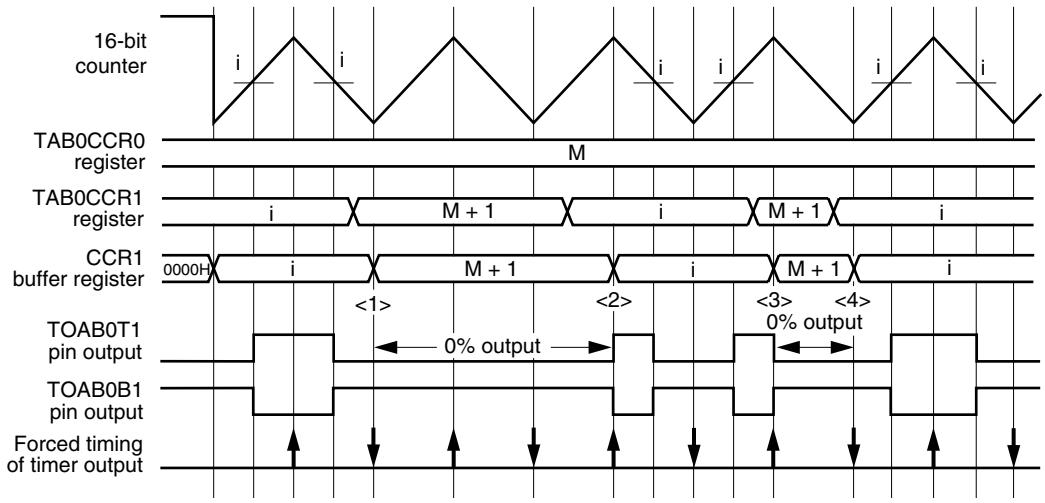
The 0% wave is output by setting the TAB0CCRm register to "M + 1" when the TAB0CCR0 register = M.

The 100% wave is output by setting the TAB0CCRm register to "0000H".

Rewriting the TAB0CCRm register is enabled while the timer is operating, and 0% wave output or 100% wave output can be selected at the point of the crest interrupt (INTTAB0CC0) and valley interrupt (INTTAB0OV).

Remark m = 1 to 3

Figure 10-9. 0% PWM Output Waveform (Without Dead Time)



<1> 0% output is selected by the valley interrupt (without a match with the 16-bit counter).
The valley interrupt forcibly lowers the timer output. This produces the 0% output.

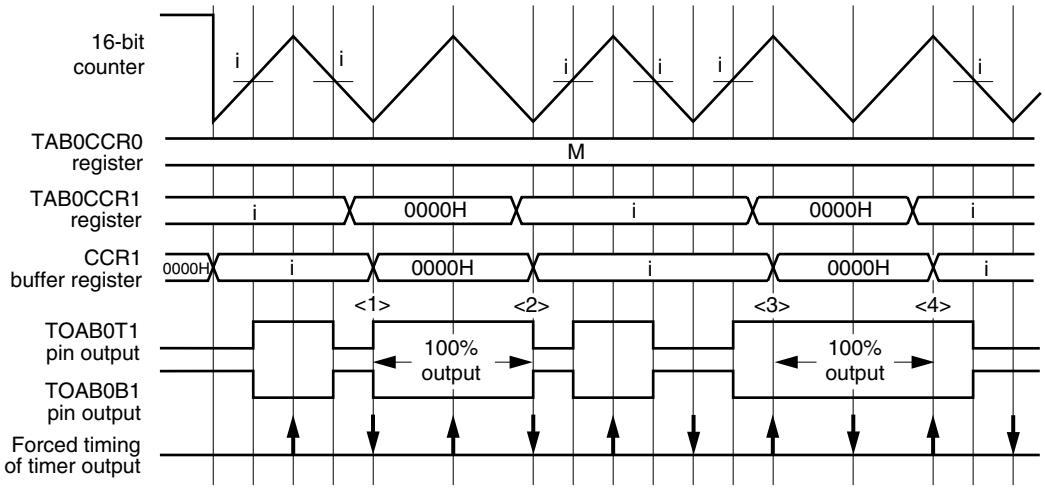
<2> 0% output is canceled by the crest interrupt (without a match with the 16-bit counter).
The crest interrupt forcibly raises the timer output. This cancels the 0% output.

<3> 0% output is selected by the crest interrupt (with a match with the 16-bit counter).
The crest interrupt forcibly raises the timer output, but lowering the timer output takes precedence when the value of the TAB0CCRm register matches the value of the 16-bit counter. As a result, the 0% wave is output.

<4> 0% output is canceled by the valley interrupt (without a match with the 16-bit counter).
The valley interrupt forcibly lowers the timer output. This cancels the 0% output.

Remark ↑ means forced raising and ↓ means forced lowering.

Figure 10-10. 100% PWM Output Waveform (Without Dead Time)



<1> 100% output is selected by the valley interrupt (with a match with the 16-bit counter).
The valley interrupt forcibly lowers the timer output, but raising the timer output takes precedence when the value of the TAB0CCRm register matches the value of the 16-bit counter. As a result, the 100% output is produced.

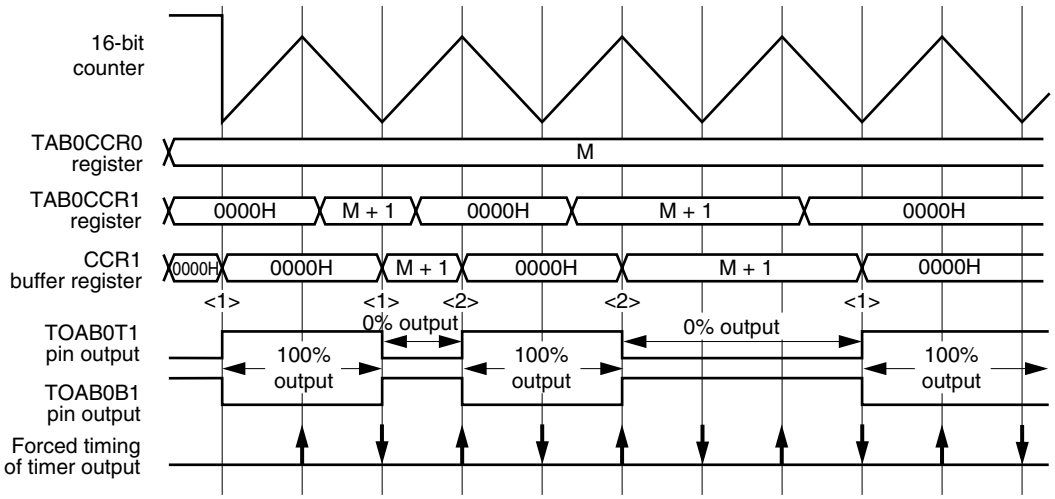
<2> 100% output is canceled by the valley interrupt (without a match with the 16-bit counter).
The valley interrupt forcibly lowers the timer output. This cancels the 100% output.

<3> 100% output is selected by the crest interrupt (without a match with the 16-bit counter).
The crest interrupt forcibly raises the timer output. This produces the 100% output.

<4> 100% output is canceled by the crest interrupt (without a match with the 16-bit counter).
The crest interrupt forcibly raises the timer output. This cancels the 100% output.

Remark ↑ means forced raising and ↓ means forced lowering.

Figure 10-11. PWM Output Waveform from 0% to 100% and from 100% to 0% (Without Dead Time)



<1> The valley interrupt selects 100% ←→ 0% or 0% ←→ 100% output.
Output can be selected from 100% ←→ 0% or 0% ←→ 100% immediately after the timer has been started.

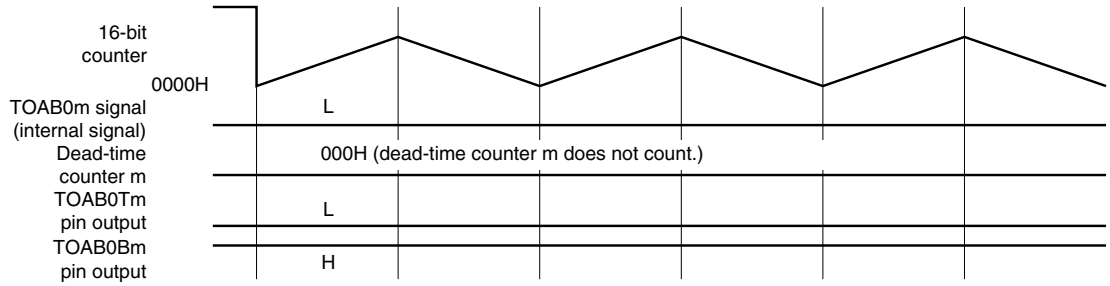
<2> The crest interrupt selects 100% ←→ 0% output.
The crest interrupt selects 100% → 0% output by using the timer output forced raising function and by a match between the 16-bit counter value and the TAB0CCR0 register value.

(3) Output waveform in vicinity of 0% and 100% output

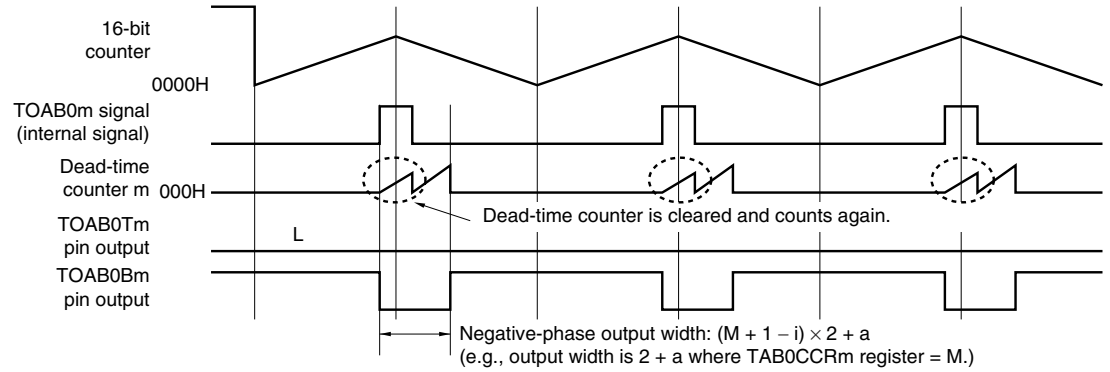
If an interrupt is generated because the value of the 16-bit counter matches the value of the compare register while dead time is being counted, the dead-time counter is cleared and starts its count operation again.
The output waveform of dead-time control in the vicinity of 0% and 100% output is shown below.

Figure 10-12. PWM Output Waveform with Dead Time (2)

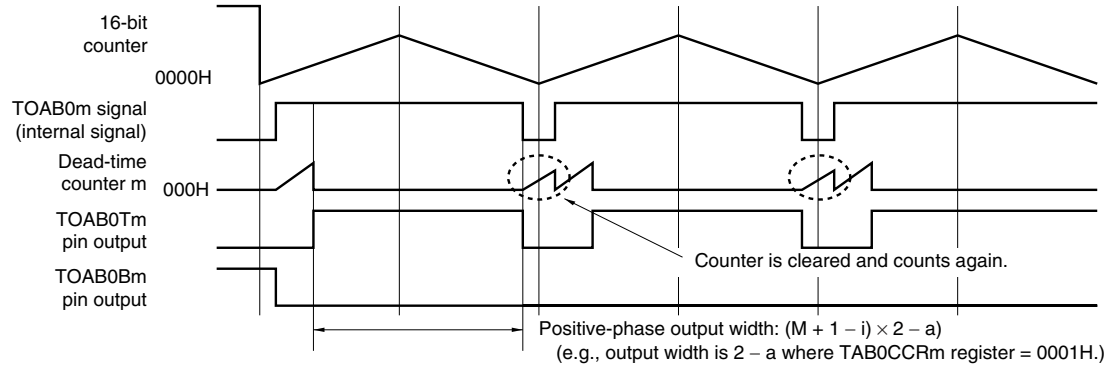
(a) 0% output (TAB0CCRm register = M + 1, TAB0CCR0 register = M, TAB0DTC register = a)



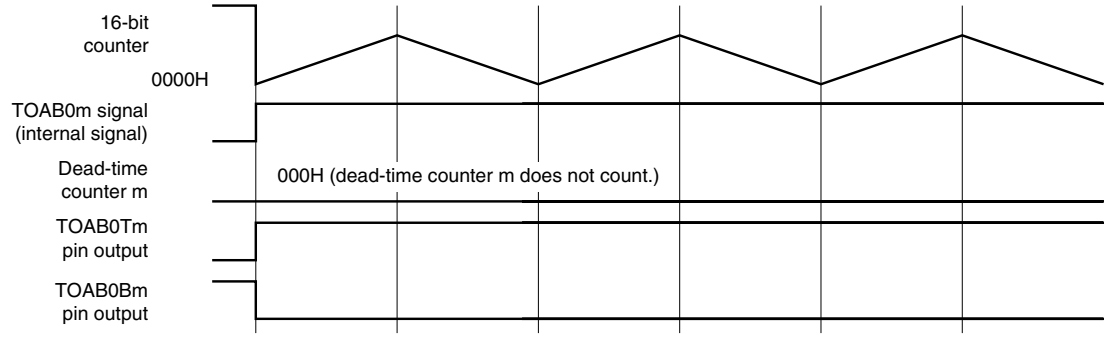
(b) In vicinity of 0% output (TAB0CCRm register = i ≥ M + 1 – a/2, TAB0CCR0 register = M, TAB0DTC register = a)



(c) In vicinity of 100% output (TAB0CCRm register = i ≤ a/2, TAB0CCR0 register = M, TAB0DTC register = a)



(d) 100% output (TAB0CCRm register = 0000H, TAB0CCR0 register = M, TAB0DTC register = a)



Remark m = 1 to 3

(4) Automatic dead-time width narrowing function (TAB0OPT2.TAB0DTM bit = 1)

The dead-time width can be automatically narrowed in the vicinity of 0% output or 100% output by setting the TAB0OPT2.TAB0DTM bit to 1.

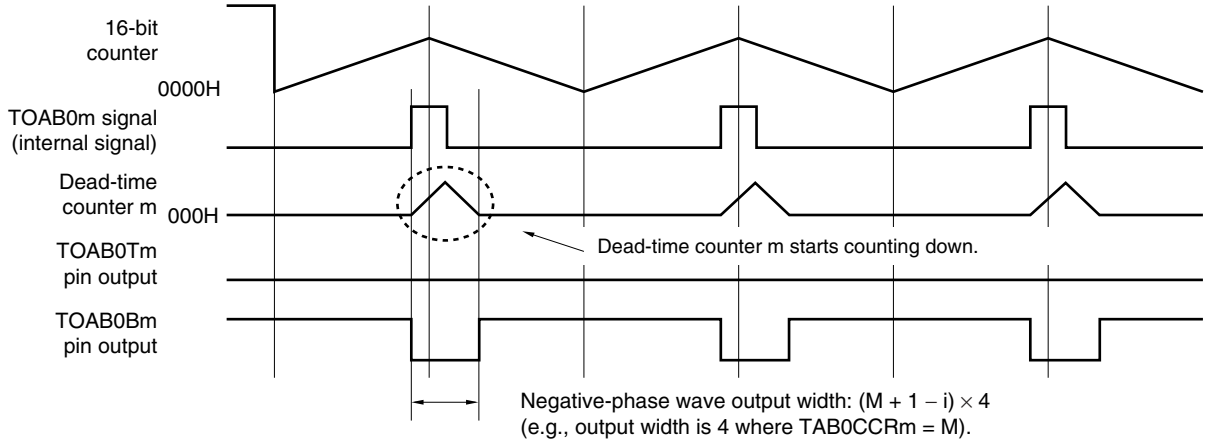
By setting the TAB0DTM bit to 1, the dead-time counter is not cleared, but starts down counting if the TOAB0m (internal signal) output of timer AB changes during dead-time counting.

The following timing chart shows the operation of the dead-time counter when the TAB0DTM bit is set to 1.

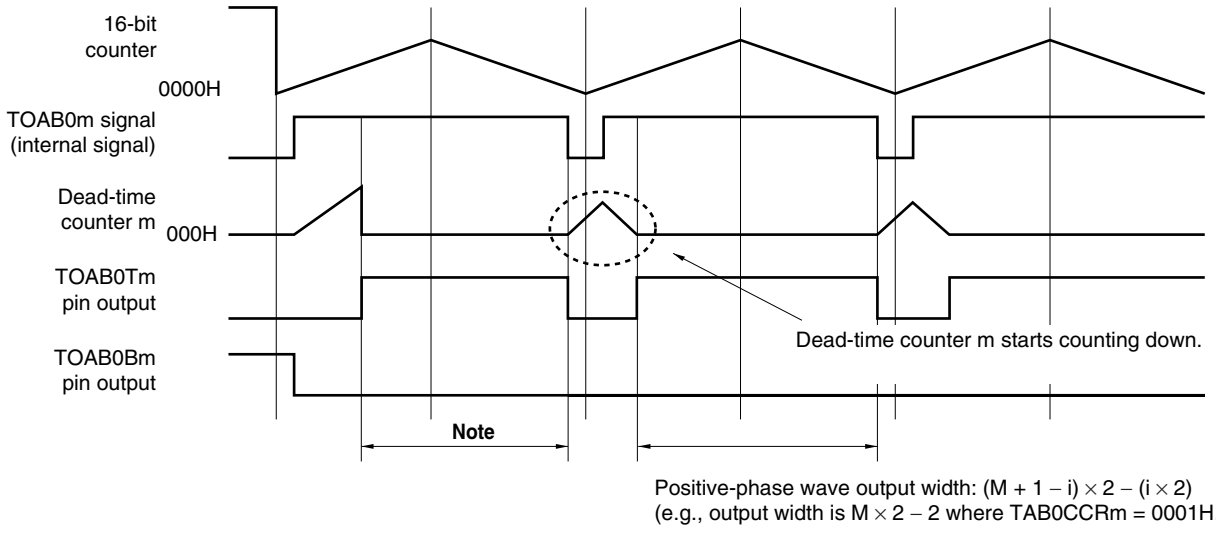
Figure 10-13. Operation of Dead-Time Counter m (1)

(a) In vicinity of 0% output

(TAB0CCRm register = i ≥ M + 1 − a/2, TAB0CCR0 register = M, TAB0DTC register = a)



(b) In vicinity of 100% output (TAB0CCRm register = i ≤ a/2, TAB0CCR0 register = M, TAB0DTC register = a)



Note The output width of the first wave differs from that of the second and subsequent waves immediately after the TAB0CTL0.TAB0CE bit has been set. The first wave is shorter than the second wave because the dead time is fully counted.

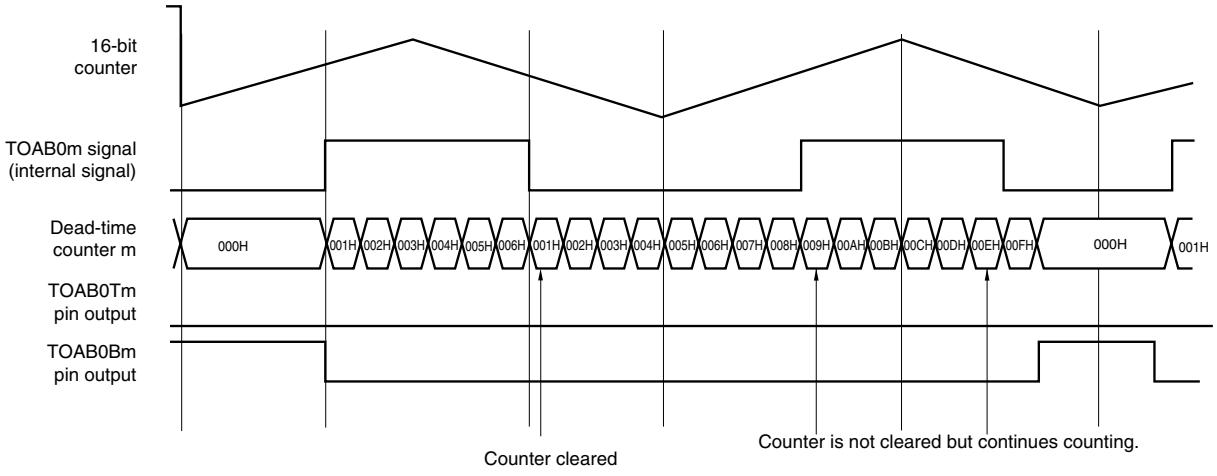
Remark m = 1 to 3

(5) Dead-time control in case of incorrect setting

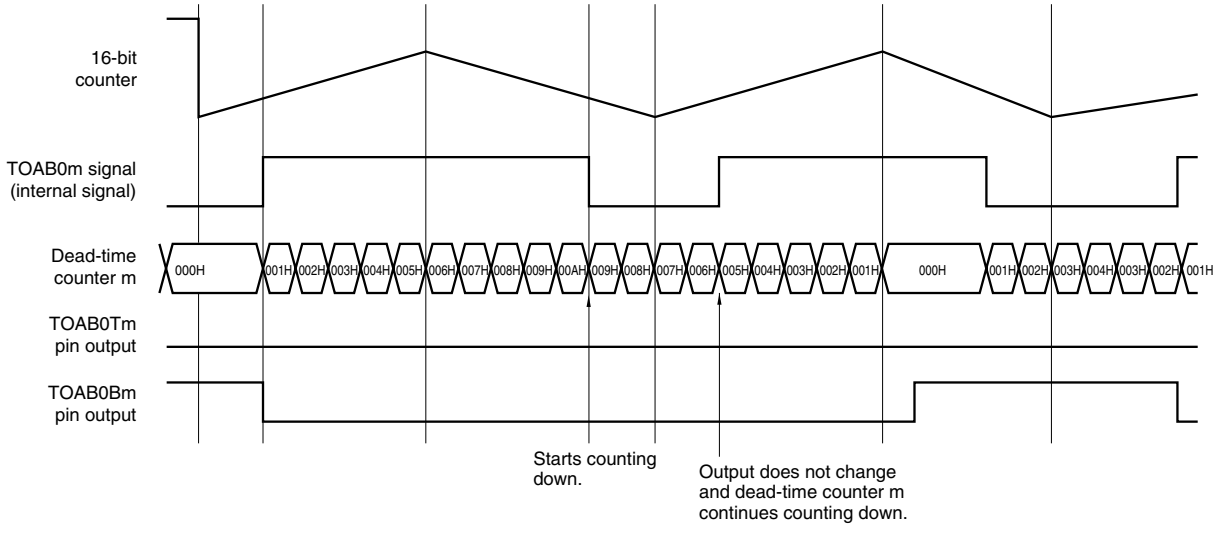
Usually, the TOAB0m (internal signal) output of TAB0 changes only once during dead-time counting, only in the vicinity of 0% and 100% output. This section shows an example where the TAB0CCR0 register (carrier cycle) and TAB0DTC register (dead-time value) are incorrectly set. If these registers are incorrectly set, the TOAB0m (internal signal) output of TAB0 changes more than once during dead-time counting. The following flowchart shows the 6-phase PWM output wave in this case.

Figure 10-14. Operation of Dead-Time Counter m (2)

(a) When TAB0OPT2.TAB0DTM bit = 0, TAB0CCR0 register = 0006H, TAB0DTC register = 000FH, TAB0CCRm register = 0004H



(b) When TAB0OPT2.TAB0DTM bit = 1, TAB0CCR0 register = 0006H, TAB0DTC register = 000FH, TAB0CCRm register = 0002H



Remark m = 1 to 3

10.4.3 Interrupt culling function

- The interrupts to be culled are INTTAB0CC0 (crest interrupt) and INTTAB0OV (valley interrupt).
- The TAB0OPT1.TAB0ICE bit is used to enable output of the INTTAB0CC0 interrupt and the number of times the interrupt is to be culled.
- The TAB0OPT1.TAB0IOE bit is used to enable output of the INTTAB0OV interrupt and the number of times the interrupt is to be culled.
- The TAB0RDE bit of TAB0OPT2 is used to specify whether transfer is to be culled or not.
- The TAB0OPT1.TAB0ID4 to TAB0OPT1.TAB0ID0 bits are used to specify the number of counts by which a specified interrupt is to be culled.

  The interrupt is culled for the duration of the specified number of counts and is generated at the next interrupt timing.

  If it is specified that transfer is to be culled, transfer is executed at the same timing as the interrupt output after culling. If it is specified that transfer is not to be culled, transfer is executed at the transfer timing after the TAB0CCR1 register has been written.
- The TAB0OPT0.TAB0CMS bit is used to specify whether the registers with a transfer function are batch rewritten or anytime rewritten.

  The values of the registers are updated in synchronization with transferring when the TAB0CMS bit is 0. When the TAB0CMS bit is 1, the values of the registers are immediately updated when a new value is written to the registers.

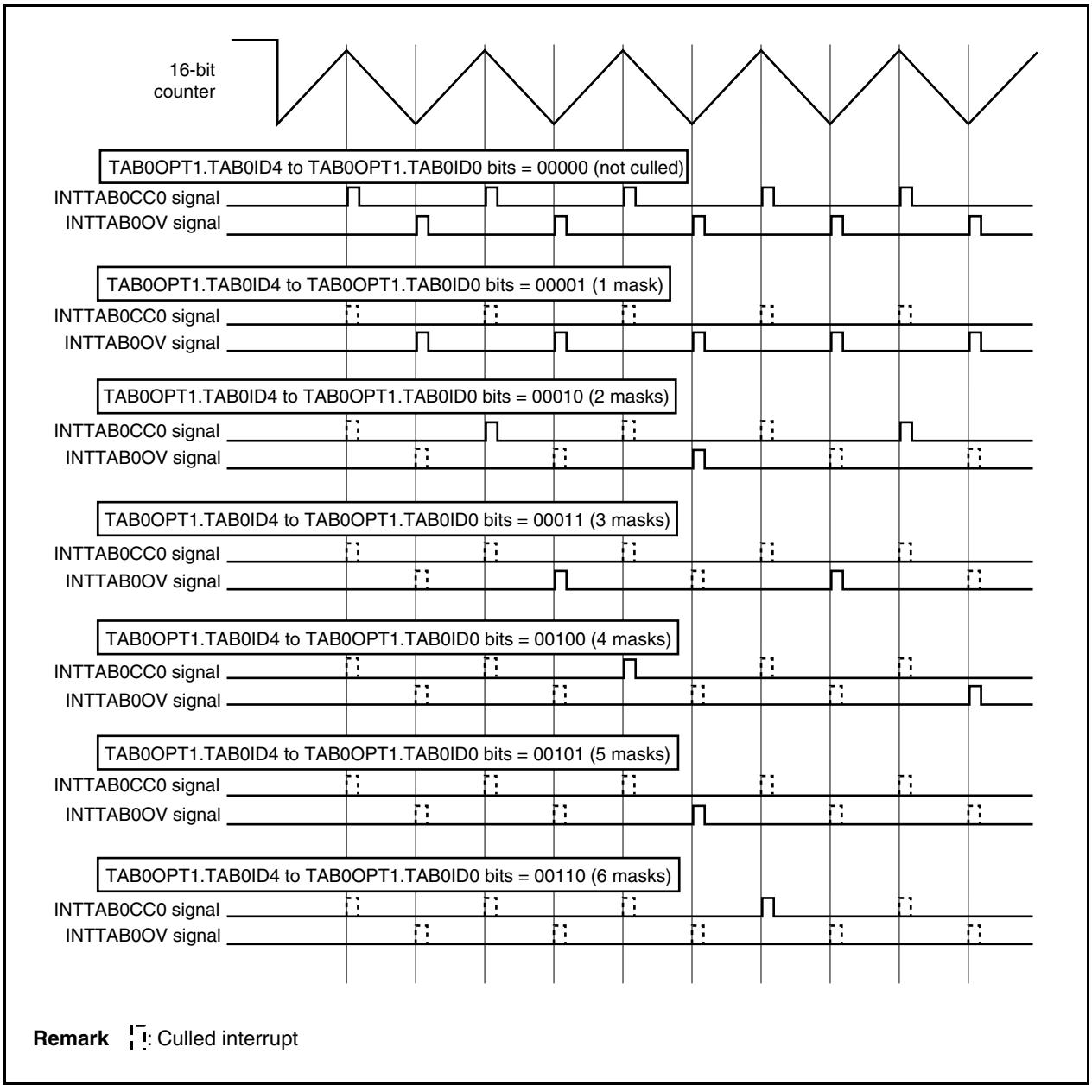
  Transfer is performed from the TAB0CCRm register to the CCRm buffer register in synchronization with interrupt culling timing.

Cautions 1. When using the interrupt culling function in the batch rewrite mode (transfer mode), execute the function in the intermittent batch rewrite mode (transfer culling mode).

2. The interrupt is generated at the timing after culling.

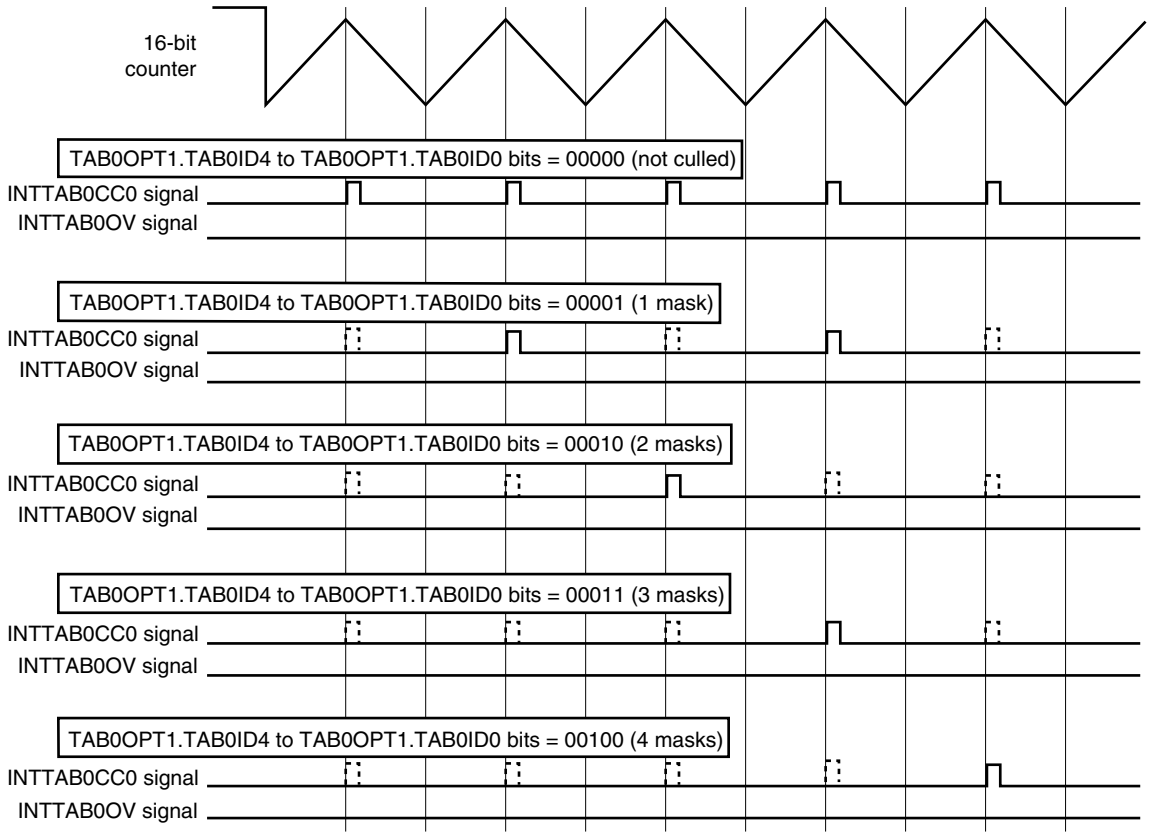
(1) Interrupt culling operation

Figure 10-15. Interrupt Culling Operation When TAB0OPT1.TAB0ICE Bit = 1, TAB0OPT1.TAB0IOE Bit = 1, TAB0OPT2.TAB0RDE Bit = 1 (Crest/Valley Interrupt Output)



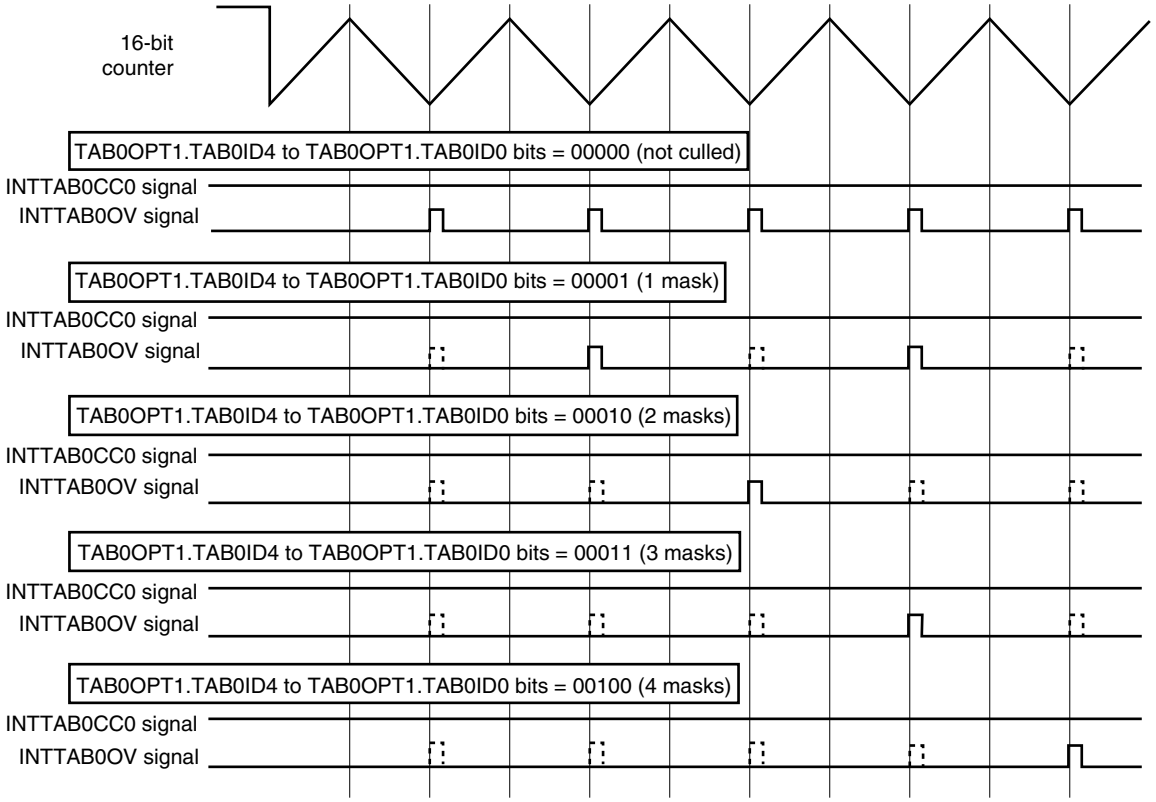
Remark : Culled interrupt

Figure 10-16. Interrupt Culling Operation When TAB0OPT1.TAB0ICE Bit = 1, TAB0OPT1.TAB0IOE Bit = 0, TAB0OPT2.TAB0RDE Bit = 1 (Crest Interrupt Output)



Remark : Culled interrupt

Figure 10-17. Interrupt Culling Operation When TAB0OPT1.TAB0ICE Bit = 0, TAB0OPT1.TAB0IOE Bit = 1, TAB0OPT2.TAB0RDE Bit = 1 (Valley Interrupt Output)



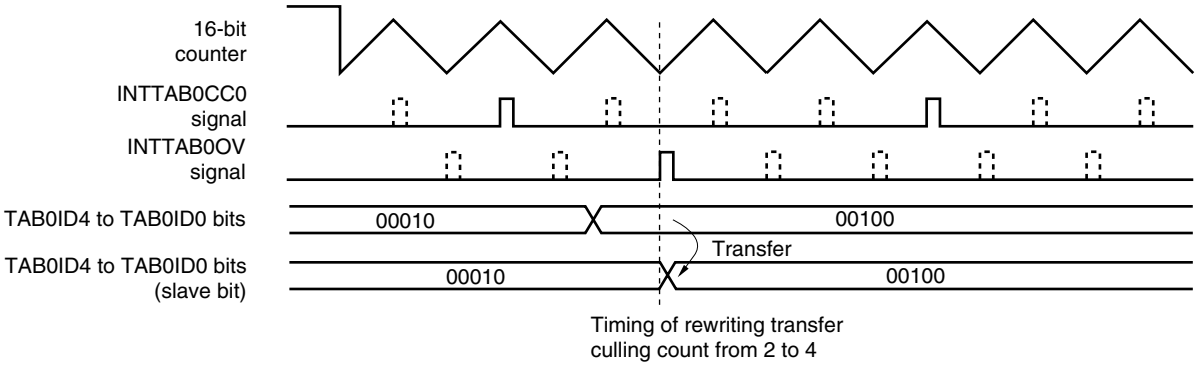
Remark : Culled interrupt

(2) To alternately output crest interrupt (INTTAB0CC0) and valley interrupt (INTTAB0OV)

To alternately output the crest and valley interrupts, set both the TAB0OPT1.TAB0ICE and TAB0OPT1.TAB0IOE bits to 1.

Figure 10-18. Crest/Valley Interrupt Output

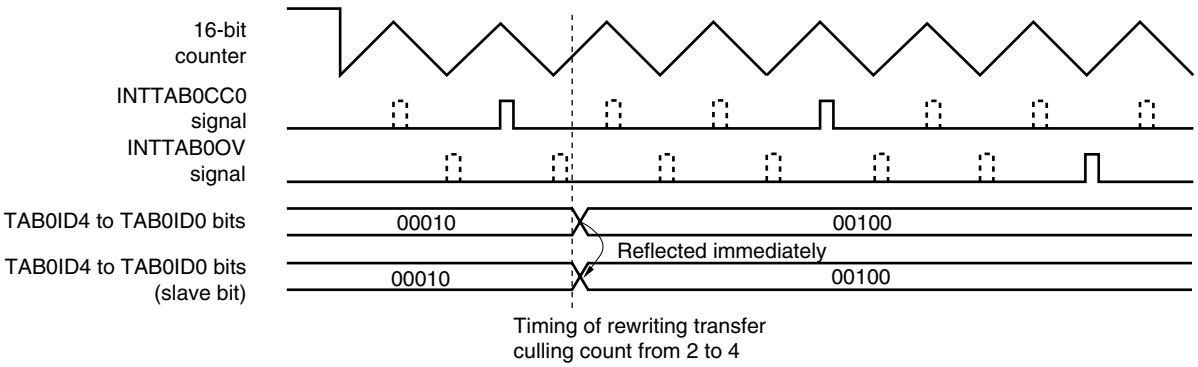
(a) TAB0OPT0.TAB0CMS bit = 0, TAB0OPT2.TAB0RDE bit = 1 (with transfer culling control)



Remarks 1. Transfer is performed at the culled interrupt output timing. The other transfer timing is ignored.

2. : Culled interrupt

(b) TAB0CMS bit = 1, TAB0RDE bit = 0 or 1 (without transfer control)



Remarks 1. Rewriting is reflected immediately. The transfer timing is ignored.

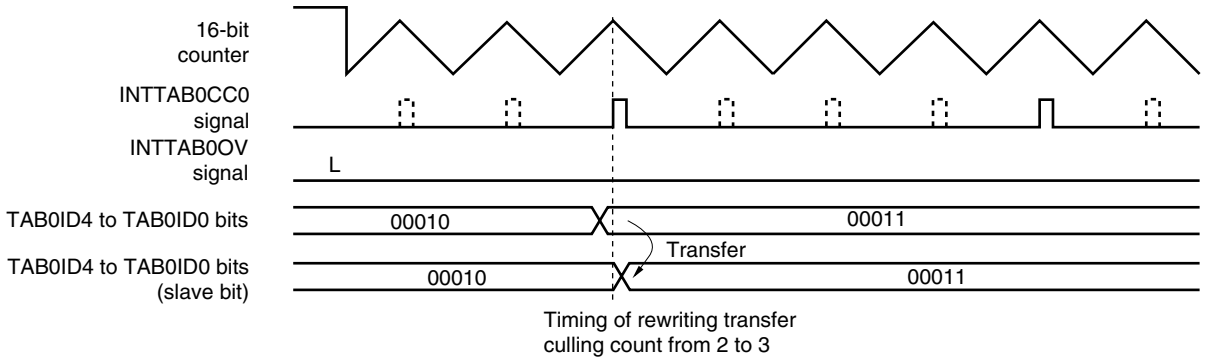
2. : Culled interrupt

(3) To output only crest interrupt (INTTAB0CC0)

Set the TAB0OPT1.TAB0ICE bit to 1 and clear the TAB0OPT1.TAB0IOE bit to 0.

Figure 10-19. Crest Interrupt Output

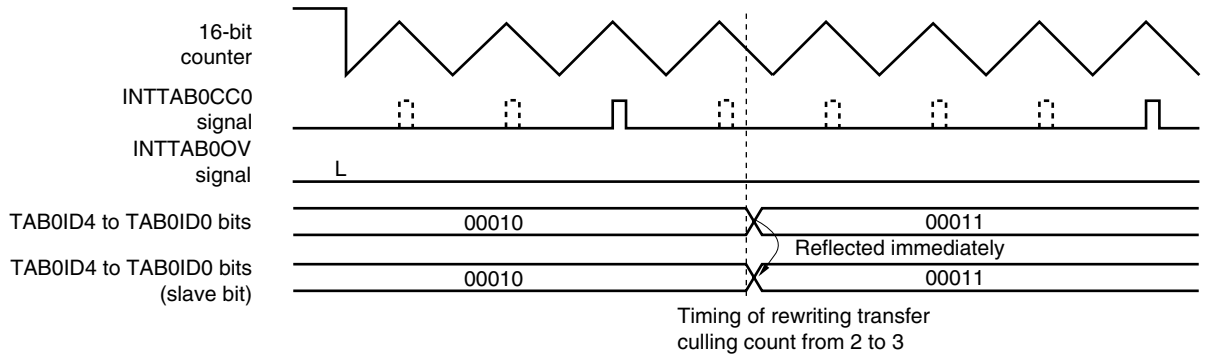
(a) TAB0OPT0.TAB0CMS bit = 0, TAB0OPT2.TAB0RDE bit = 1 (with transfer culling control)



Remarks 1. Transfer is performed at the culled interrupt output timing. The other transfer timing is ignored.

2. ⊓: Culled interrupt

(b) TAB0CMS bit = 1, TAB0RDE bit = 0 or 1 (without transfer control)



Remarks 1. Rewriting is reflected immediately. The transfer timing is ignored.

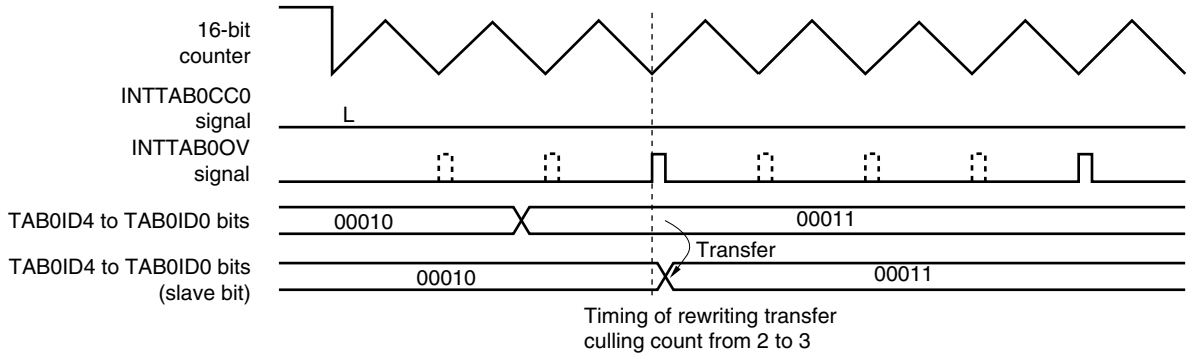
2. ⊓: Culled interrupt

(4) To output only valley interrupt (INTTAB0OV)

Clear the TAB0OPT1.TAB0ICE bit to 0 and set the TAB0IOE bit to 1.

Figure 10-20. Valley Interrupt Output

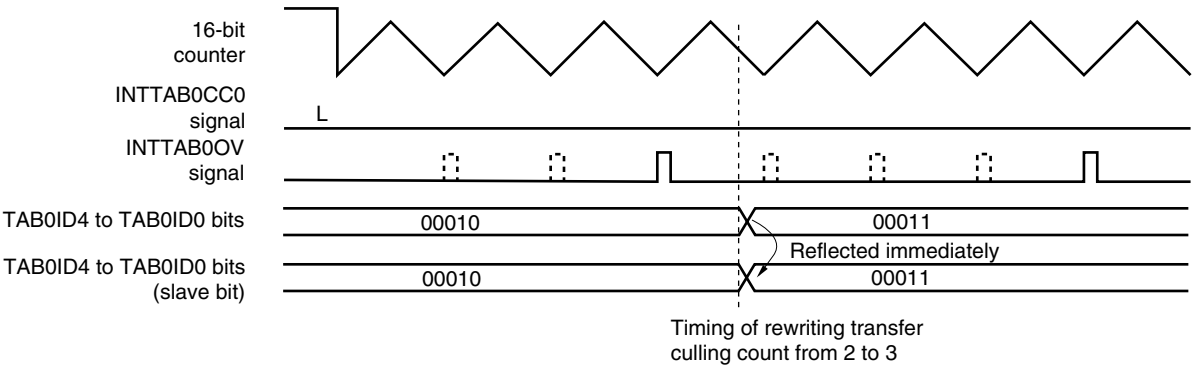
(a) TAB0OPT0.TAB0CMS bit = 0, TAB0OPT2.TAB0RDE bit = 1 (with transfer culling control)



Remarks 1. Transfer is performed at the culled interrupt output timing. The other transfer timing is ignored.

2. : Culled interrupt

(b) TAB0CMS bit = 1, TAB0RDE bit = 0 or 1 (without transfer control)



Remarks 1. Rewriting is reflected immediately. The transfer timing is ignored.

2. : Culled interrupt

10.4.4 Operation to rewrite register with transfer function

The following seven registers are provided with a transfer function and used to control a motor. Each of registers has a buffer register.

- TAB0CCR0: Register that specifies the cycle of the 16-bit counter (TAB)
- TAB0CCR1: Register that specifies the duty factor of TOAB0T1 (U) and TOAB0B1 ($\overline{U}$)
- TAB0CCR2: Register that specifies the duty factor of TOAB0T2 (V) and TOAB0B2 ($\overline{V}$)
- TAB0CCR3: Register that specifies the duty factor of TOAB0T3 (W) and TOAB0B3 ($\overline{W}$)
- TAB0OPT1: Register that specifies the culling of interrupts
- TAA4CCR0: Register that specifies the A/D conversion start trigger generation timing (TAA4 during tuning operation)
- TAA4CCR1: Register that specifies the A/D conversion start trigger generation timing (TAA4 during tuning operation)

The following three rewrite modes are provided in the registers with a transfer function.

- Anytime rewrite mode
  This mode is set by setting the TAB0OPT0.TAB0CMS bit to 1.
  In this mode, each compare register is updated independently, and the value of the compare register is updated as soon as a new value is written to it.

- Batch rewrite mode (transfer mode)
  This mode is set by clearing the TAB0OPT0.TAB0CMS bit to 0, the TAB0OPT1.TAB0ID4 to TAB0OPT1.TAB0ID0 bits to 00000, and the TAB0OPT2.TAB0RDE bit to 0.
  When data is written to the TAB0CCR1 register, data in the seven registers are transferred to the buffer register all at once at the next transfer timing. Unless the TAB0CCR1 register is rewritten, the transfer operation is not performed even if the other six registers are rewritten.
  The transfer timing is the timing of each crest (match between the 16-bit counter value and TAB0CCR0 register value) and valley (match between the 16-bit counter value and 0001H) regardless of the interrupt.

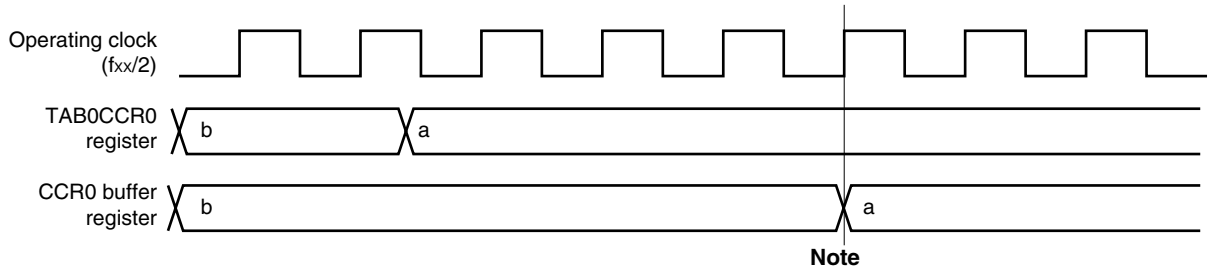
- Intermittent batch rewrite mode (transfer culling mode)
  This mode is set by clearing the TAB0OPT0.TAB0CMS bit to 0 and setting the TAB0OPT2.TAB0RDE bit to 1.
  When data is written to the TAB0CCR1 register, the seven registers are transferred to the buffer register all at once at the next transfer timing. Unless the TAB0CCR1 register is rewritten, the transfer operation is not performed even if the other six registers are rewritten.
  If interrupt culling is specified by the TAB0OPT1 register, the transfer timing is also culled as the interrupts are culled, and the seven registers are transferred all at once at the culled timing of crest interrupt (match between the 16-bit counter value and TAB0CCR0 register value) or valley interrupt (match between the 16-bit counter value and 0001H).
  For details of the interrupt culling function, see **10.4.3 Interrupt culling function**.

(1) Anytime rewrite mode

This mode is set by setting the TAB0OPT0.TAB0CMS bit to 1. The setting of the TAB0OPT2.TAB0RDE bit is ignored.

In this mode, the value written to each register with a transfer function is immediately transferred to an internal buffer register and compared with the value of the counter. If a register with transfer function is rewritten in this mode after the count value of the 16-bit counter matches the value of the TAB0CCRm register, the rewritten value is not reflected because the next match is ignored after the first match has occurred. If the register is rewritten during counting up, the new register value becomes valid after the counter has started counting down.

Figure 10-21. Timing of Reflecting Rewritten Value

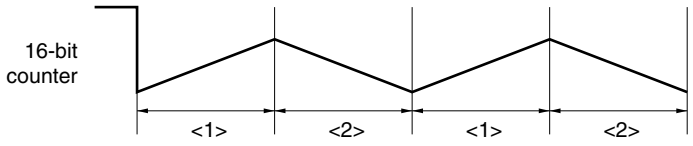


Note After the register (TAB0CCR0, TAB0CCR2, TAB0CCR3, TAB0OPT1, TAA4CCR0, or TAA4CCR1) has been written, the written value is transferred to an internal buffer register after four clocks of the operating clock. However, the value of only the TAB0CCR1 register is transferred after 5 more clocks.

(a) Rewriting TAB0CCR0 register

Even if the TAB0CCR0 register is rewritten in the anytime rewrite mode, the new value may not be reflected in some cases.

Figure 10-22. Example of Rewriting TAB0CCR0 Register



Rewriting during period <1> (rewriting during counting up)

If the newly rewritten value is greater than the value of the 16-bit counter, there is no problem because it will match the value of the 16-bit counter. If the new value is less than the value of the 16-bit counter, it will not match the value of the counter. As a result, the 16-bit counter overflows and continues counting up from 0000H until it matches the register value again, and the correct PWM waveform is not output.

Rewriting during period <2> (rewriting during counting down)

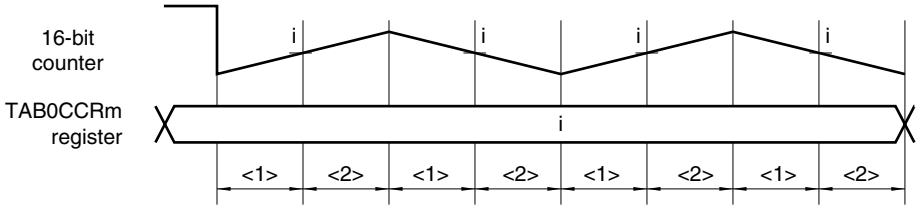
A match with the value of the 16-bit counter is ignored during counting down. Therefore, the rewritten period value is reflected starting from counting up in the next cycle as a match point.

(b) Rewriting TAB0CCRm register

Figure 10-24 shows the timing of rewriting before the value of the 16-bit counter matches the value of the TAB0CCRm register (<1> in Figure 10-23), and Figure 10-25 shows the timing of rewriting after the value of the 16-bit counter matches the value of the TAB0CCRm register (<2> in Figure 10-23).

Figure 10-23. Basic Operation of 16-Bit Counter and TAB0CCRm Register

(a) Basic figure



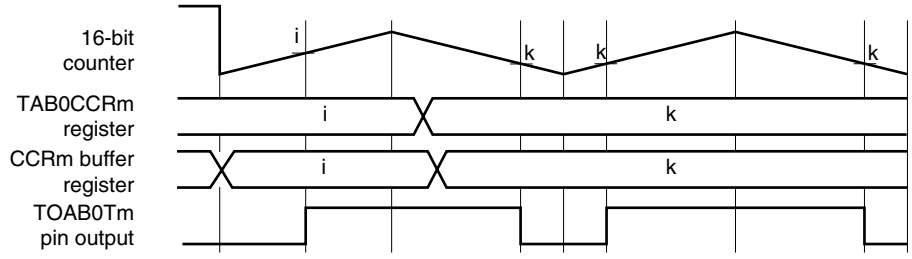
Remarks 1. i = Set value of TAB0CCRm register

2. m = 1 to 3

Figure 10-24. Example of Rewriting TAB0CCR1 to TAB0CCR3 Registers (Rewriting Before Match Occurs)

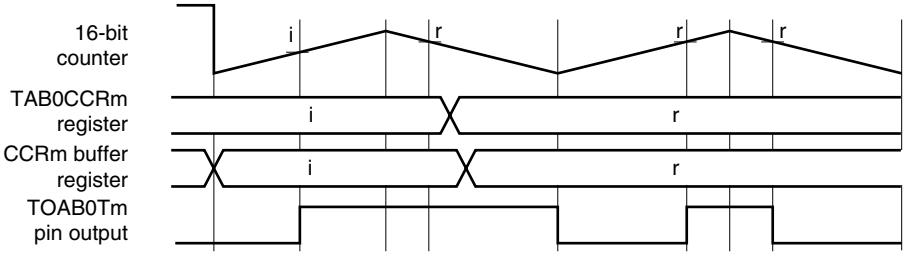
(a)

If the TAB0CCRm register is rewritten before its value matches the value of the 16-bit counter, the register value will match the value of the 16-bit counter after the register has been rewritten. Consequently, the new register value is immediately reflected.



(b)

If a value less than the value of the 16-bit counter (greater if the counter is counting down) is written to the TAB0CCRm register, the output waveform is as follows because the register value does not match the counter value.

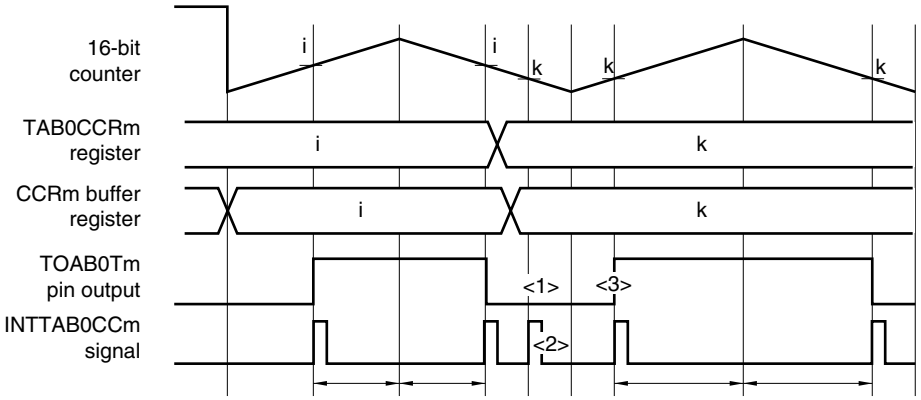


If the register value does not match the counter value, the TOAB0Tm pin output does not change. Even if the value of the 16-bit counter does not match the value of the TAB0CCRm register, the TOAB0Tm pin output always changes to the high level if the crest interrupt occurs and to the low level if the valley interrupt occurs.
This is a function provided for 0% output and 100% output.
For details, see **10.4.2 (2) PWM output of 0%/100%**.

Remarks 1. i, r, k = Set values of TAB0CCRm register

2. m = 1 to 3

Figure 10-25. Example of Rewriting TAB0CCR1 to TAB0CCR3 Registers (Rewriting After Match Occurs)



<1> Matching of the count value of the 16-bit counter and the value of the TAB0CCRm register as a result of rewriting the register is ignored after a match signal has been generated, and the PWM output does not change.

<2> Even if the PWM output does not change, the interrupt generated upon a match between the 16-bit counter value and the TAB0CCRm register value (INTTABCCm) is output.

<3> The next match between the 16-bit counter and TAB0CCRm register is valid after the counter has changed its counting direction to up or down, and the PWM output changes.

If the TAB0CCRm register is rewritten after its value matches the value of the 16-bit counter, the next match is ignored after the first match occurs and the rewritten value is not reflected to the TOAB0Tm pin output. If the register is rewritten while the counter is counting down, the match that occurs after the counter starts counting down is valid (the match that occurs after the counter has started counting up is valid if the register is rewritten while the counter is counting up).

Remarks 1. i, r, k = Set value of TAB0CCRm register

2. m = 1 to 3

(c) Rewriting TAB0OPT1 register

The interrupt culling counter is cleared when the TAB0OPT1 register is written. When the interrupt culling counter has been cleared, the measured number of times the interrupt has occurred is discarded. Consequently, the interrupt generation interval is temporarily extended.

To avoid this operation, rewrite the TAB0OPT1 register in the intermittent batch rewrite mode (transfer culling mode).

For details of rewriting the TAB0OPT1 register, see **10.4.3 Interrupt culling function**.

(2) Batch rewrite mode (transfer mode)

This mode is set by clearing the TAB0OPT0.TAB0CMS bit to 0, the TAB0OPT1.TAB0ID4 to TAB0OPT1.TAB0ID0 bits to 00000, and the TAB0OPT2.TAB0RDE bit to 0.

In this mode, the values written to each compare register are transferred to the internal buffer register all at once at the transfer timing and compared with the counter value.

(a) Rewriting procedure

If data is written to the TAB0CCR1 register, the values set to the TAB0CCR0 to TAB0CCR3, TAB0OPT1, TAA4CCR0, and TAA4CCR1 registers are transferred all at once to the internal buffer register at the next transfer timing. Therefore, write to the TAB0CCR1 register last. Writing to the register is prohibited after the TAB0CCR1 register has been written and before the transfer timing is generated (until the crest (match between the 16-bit counter value and TAB0CCR0 register value) or the valley (match between the 16-bit counter value and 0001H)). The operation procedure is as follows.

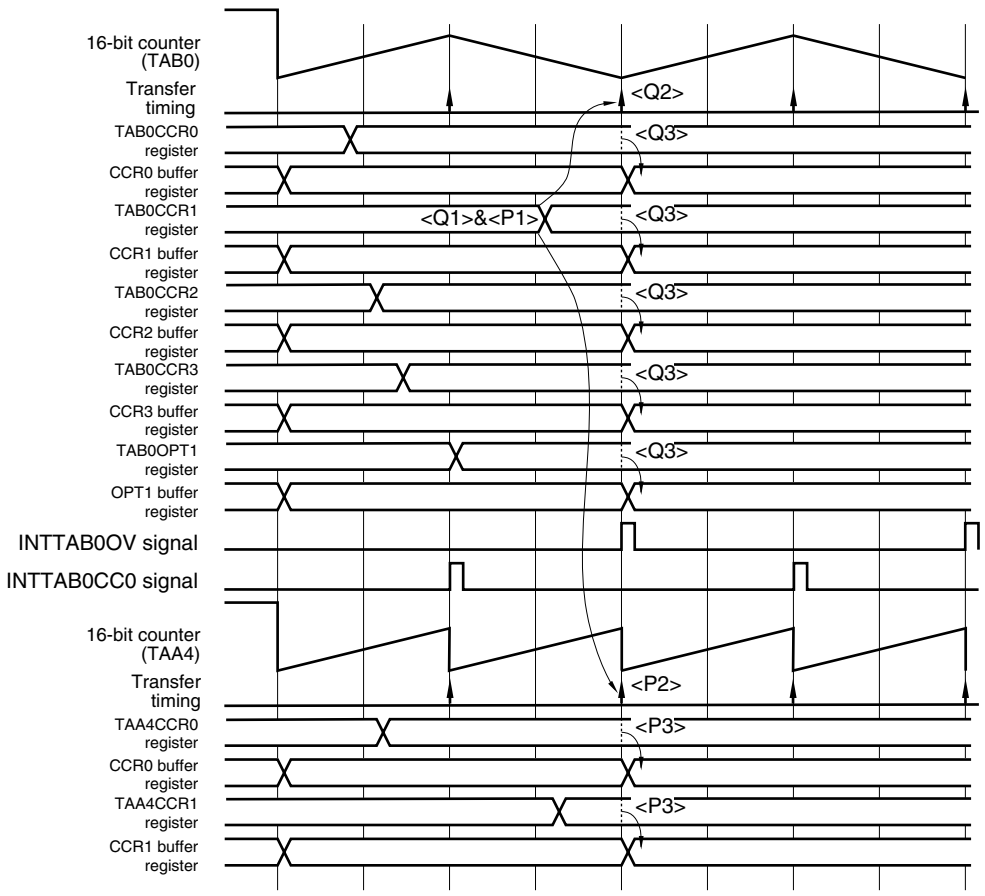
<1> Rewriting the TAB0CCR0, TAB0CCR2, TAB0CCR3, TAB0OPT1, TAA4CCR0, and TAA4CCR1 registers
(Do not rewrite registers that do not have to be rewritten.)

<2> Rewriting the TAB0CCR1 register
(Rewrite the same value to the register even when it is not necessary to rewrite the TAB0CCR1 register.)

<3> Holding the next rewriting pending until the transfer timing is generated
(Rewrite the register next time after the INTTAB0OV or INTTAB0CC0 interrupt has occurred.)

<4> Return to <1>.

Figure 10-26. Basic Operation in Batch Mode



[Operation of TAB0]

<Q1> Write the TAB0CCR1 register

<Q2> The target timing is the first transfer timing after a write to the TAB0CCR1 register.

<Q3> The values are transferred all at once at the transfer timing.

[Operation of TAA4]

<P1> Write the TAB0CCR1 register

<P2> The target timing is the first transfer timing after a write to the TAB0CCR1 register.

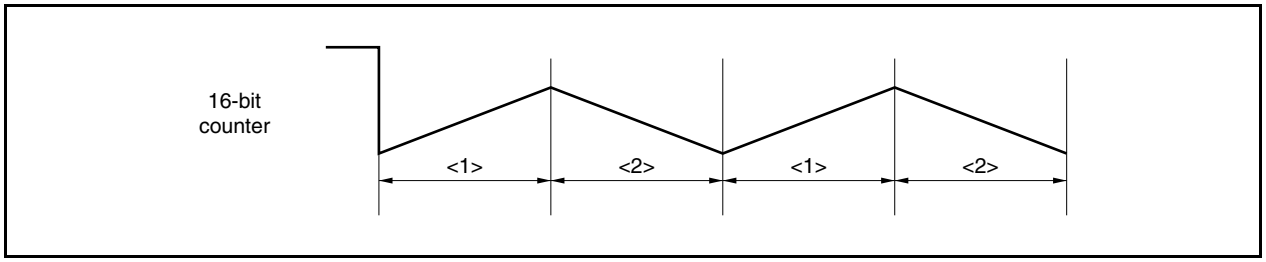
<P3> The values are transferred all at once at the transfer timing.

(b) Rewriting TAB0CCR0 register

When rewriting the TAB0CCR0 register in the batch rewrite mode, the output waveform differs depending on whether transfer occurs at the crest (match between the 16-bit counter value and TAB0CCR0 register value) or at the valley (match between the 16-bit counter value and 0001H). Usually, it is recommended to rewrite the TAB0CCR0 register while the 16-bit counter is counting down, and transfer the register value at the transfer timing of the crest timing.

Figure 10-28 shows an example of rewriting the TAB0CCR0 register while the 16-bit counter is counting up (during period <1> in Figure 10-27). Figure 10-29 shows an example of rewriting the TAB0CCR0 register while the counter is counting down (during period <2> in Figure 10-27).

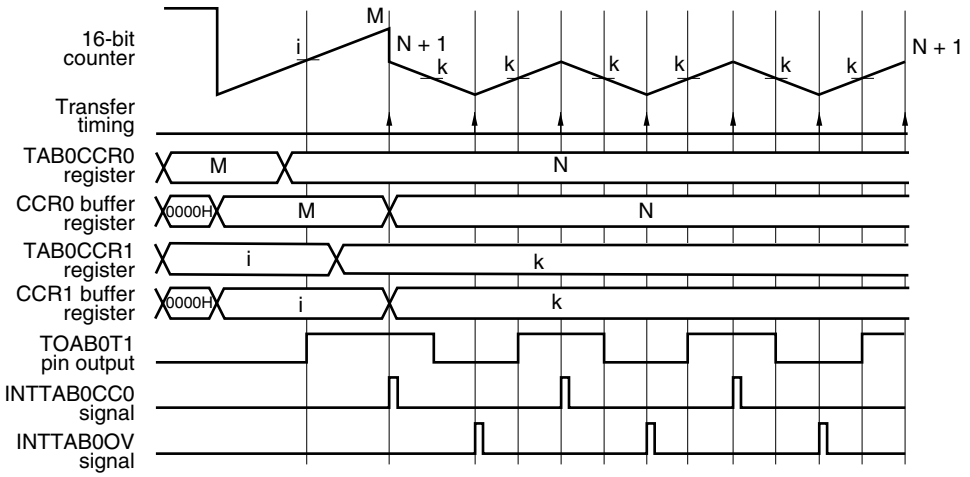
Figure 10-27. Basic Operation of 16-Bit Counter



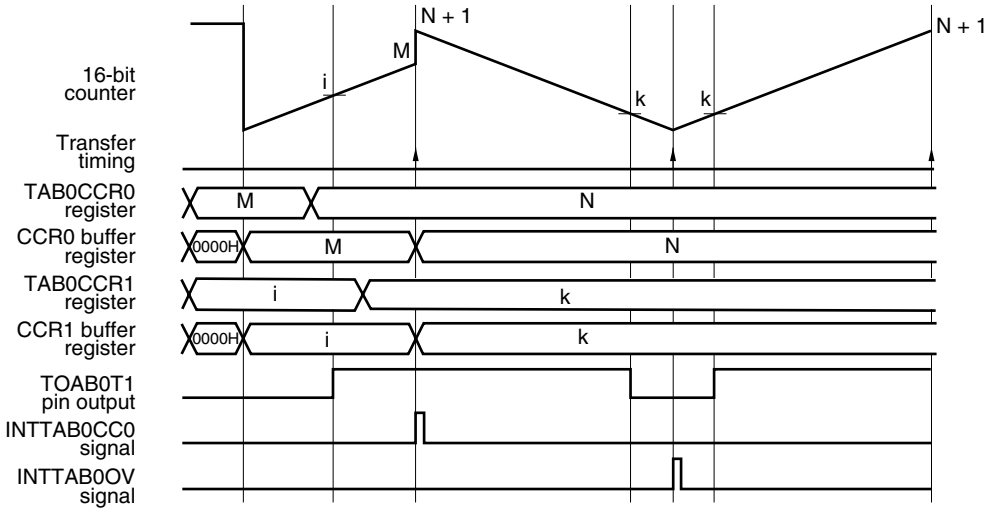
The transfer timing in Figure 10-28 is at the point where the crest timing occurs. While the 16-bit counter is counting down, the cycle changes and an asymmetrical triangular wave is output. Because the cycle changes, rewrite the duty factor (voltage data value).

Figure 10-28. Example of Rewriting TAB0CCR0 Register (During Counting Up)

(a) M > N



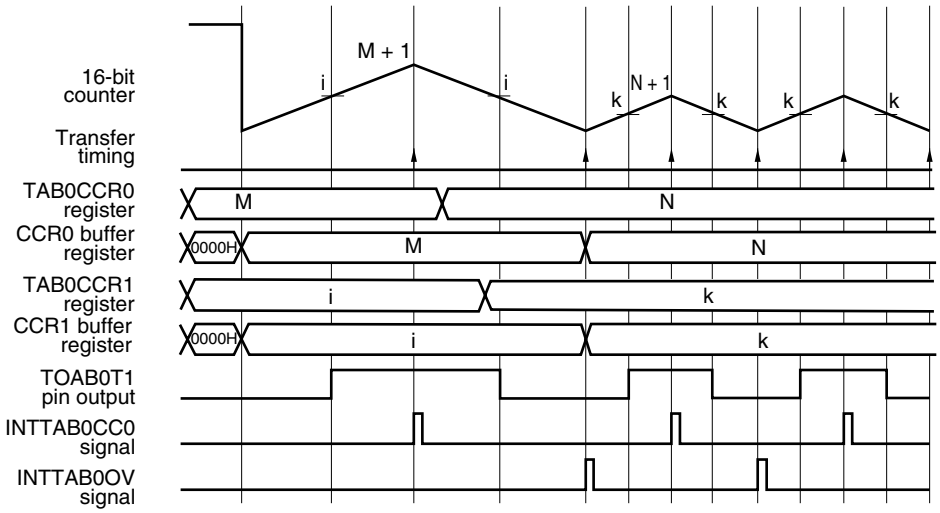
(b) M < N



Remarks 1. If transfer (match between the value of the 16-bit counter and the value of the CCR0 buffer register) occurs in the 6-phase PWM output mode, the value of the TAB0CCR0 register plus 1 is loaded to the 16-bit counter. In this way, the expected wave can be output even if the cycle value is changed at the transfer timing of the crest (match between the 16-bit counter value and the TAB0CCR0 register value) timing.

2. M: Value of CCR0 buffer register before rewriting
N: Value of CCR0 buffer register after rewriting

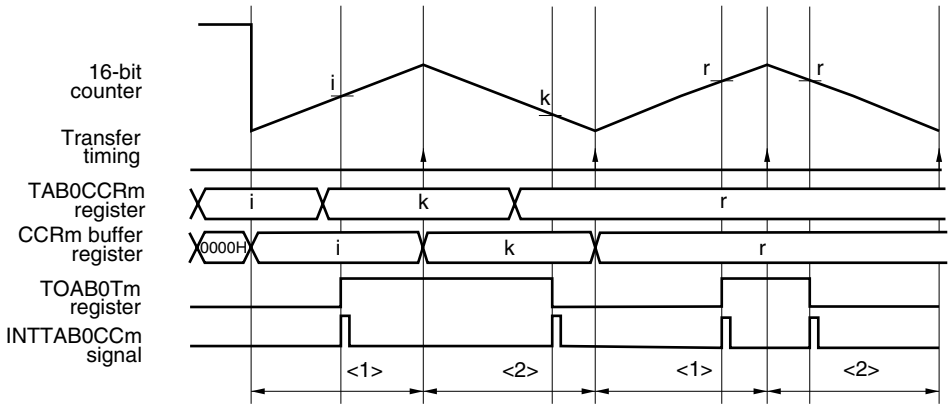
Figure 10-29. Example of Rewriting TAB0CCR0 Register (During Counting Down)



Because the next transfer timing is at the point of the valley (match between the 16-bit counter value and 0001H), the cycle value changes from the next cycle and output of a symmetrical triangular wave is maintained. Because the cycle changes, rewrite the duty value (voltage data value) as required.

(c) Rewriting TAB0CCRm register

Figure 10-30. Example of Rewriting TAB0CCRm Register



Rewriting during period <1> (rewriting during counting up)

Because the TAB0CCRm register value is transferred at the transfer timing of the crest (match between the 16-bit counter value and TAB0CCRm register value), an asymmetrical triangular wave is output.

Rewriting during period <2> (rewriting during counting down)

Because the TAB0CCRm register value is transferred at the transfer timing of the valley (match between the 16-bit counter value and 0001H), a symmetrical triangular wave is output.

Remark m = 1 to 3

(d) Transferring TAB0OPT1 register value

Do not set the TAB0OPT1.TAB0ID4 to TAB0OPT1.TAB0ID0 bits to other than 00000. When using the interrupt culling function, rewrite the TAB0OPT1 register in the intermittent batch rewrite mode (transfer culling mode).

For details of rewriting the TAB0OPT1 register, see **10.4.3 Interrupt culling function**.

(3) Intermittent batch rewrite mode (transfer culling mode)

This mode is set by clearing the TAB0OPT0.TAB0CMS bit to 0 and setting the TAB0OPT2.TAB0RDE bit to 1.

In this mode, the values written to each compare register are transferred to the internal buffer register all at once after the culled transfer timing and compared with the counter value. The transfer timing is the timing at which an interrupt is generated (INTTAB0CC0, INTTAB0OV) by interrupt culling.

For details of the interrupt culling function, see **10.4.3 Interrupt culling function**.

(a) Rewriting procedure

If data is written to the TAB0CCR1 register, the TAB0CCR0 to TAB0CCR3, TAB0OPT1, TAA4CCR0, and TAA4CCR1 registers are transferred all at once to the internal buffer register at the next transfer timing. Therefore, write to the TAB0CCR1 register last. Writing to the register is prohibited after the TAB0CCR1 register has been written until the transfer timing is generated (until the INTTAB0OV or INTTAB0CC0 interrupt occurs). The operation procedure is as follows.

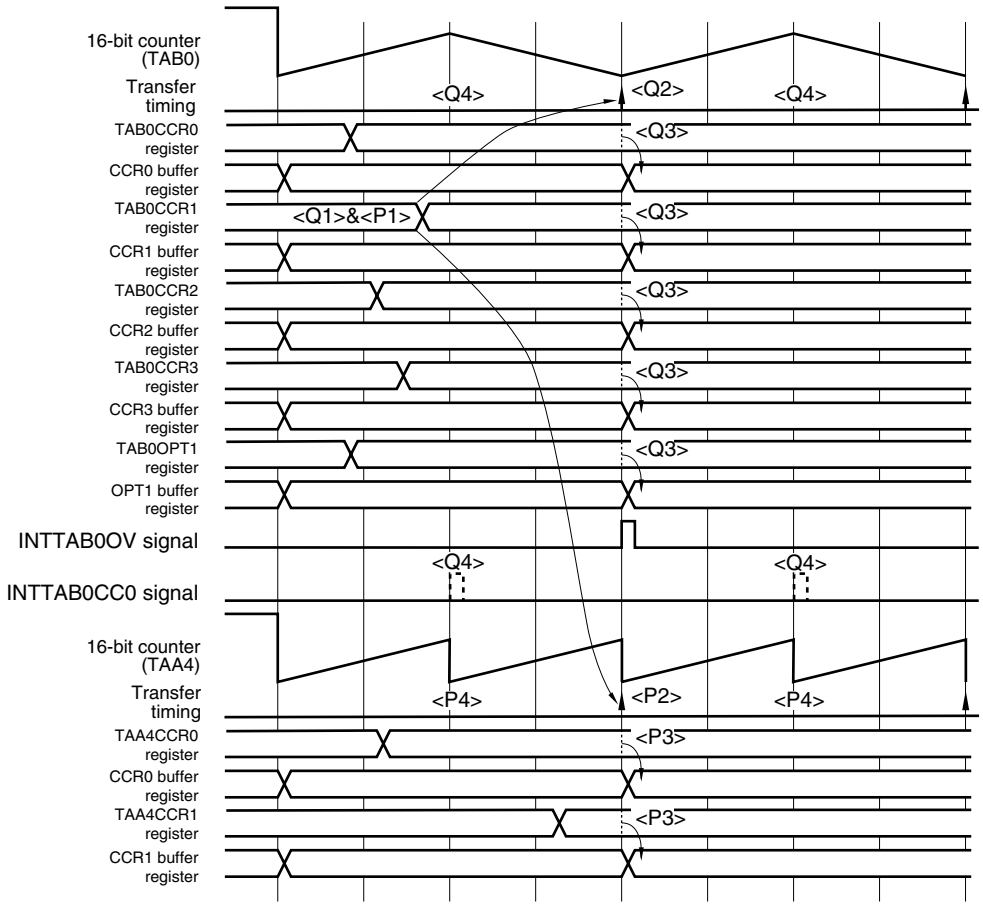
<1> Rewrite the TAB0CCR0, TAB0CCR2, TAB0CCR3, TAB0OPT1, TAA4CCR0, and TAA4CCR1 registers. Do not rewrite registers that do not have to be rewritten.

<2> Rewrite the TAB0CCR1 register.
Rewrite the same value to the register even when it is not necessary to rewrite the TAB0CCR1 register.

<3> Hold the next rewriting pending until the transfer timing is generated.
Perform the next rewrite after the INTTAB0OV or INTTAB0CC0 interrupt has occurred.

<4> Return to <1>.

Figure 10-31. Basic Operation in Intermittent Batch Rewrite Mode



[TAB0 operation]

<Q1> Write the TAB0CCR1 register.

<Q2> Rewrite the register at the transfer timing that is generated after the TAB0CCR1 register has been rewritten.

<Q3> The registers are transferred all at once at the transfer timing.

<Q4> The transfer timing is also culled as the interrupts are culled.

[TAA4 operation]

<P1> Write the TAB0CCR1 register.

<P2> Rewrite the register at the transfer timing that is generated after the TAB0CCR1 register has been rewritten.

<P3> The registers are transferred all at once at the transfer timing.

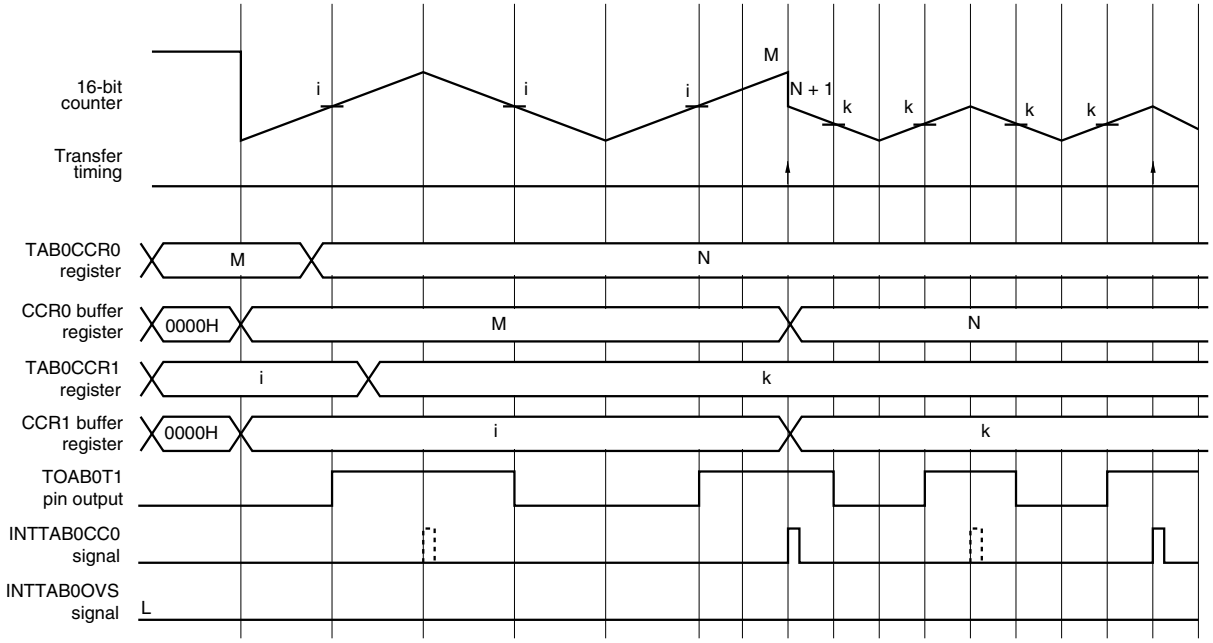
<P4> The transfer timing is also culled as the interrupts are culled.

Remark This is an example of the operation when the TAB0OPT1.TAB0ICE bit = 1, TAB0OPT1.TAB0IOE bit = 1, TAB0OPT1.TAB0ID4 to TAB0OPT1.TAB0ID0 bits = 00001.

(b) Rewriting TAB0CCR0 register

When rewriting the TAB0CCR0 register in the intermittent batch mode, the output waveform differs depending on where the occurrence of the crest or valley interrupt is specified by the interrupt culling setting. The following figure illustrates the change of the output waveform when interrupts are culled.

Figure 10-32. Rewriting TAB0CCR0 Register (When Crest Interrupt Is Set)

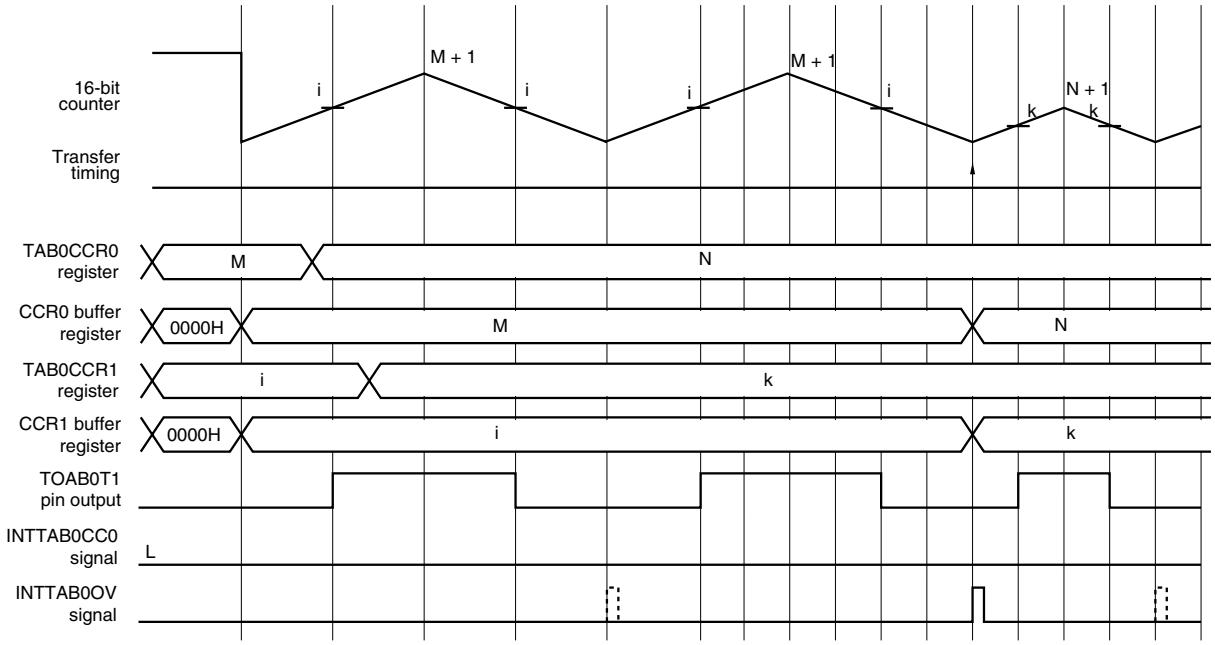


The transfer timing is generated when the crest interrupt occurs, the cycle of counting up and counting down changes, and an asymmetrical triangular wave is output.

Remarks 1. This is an example of the operation when the TAB0OPT1.TAB0ICE bit = 1, TAB0OPT1.TAB0IOE bit = 0, TAB0OPT1.TAB0ID4 to TAB0OPT1.TAB0ID0 bits = 00001.

2. : Culled interrupt

Figure 10-33. Rewriting TAB0CCR0 Register (When Valley Interrupt Is Set)



The transfer timing is generated when the valley interrupt occurs, the cycle of counting up becomes same as cycle of counting down, and a symmetrical triangular wave is output.

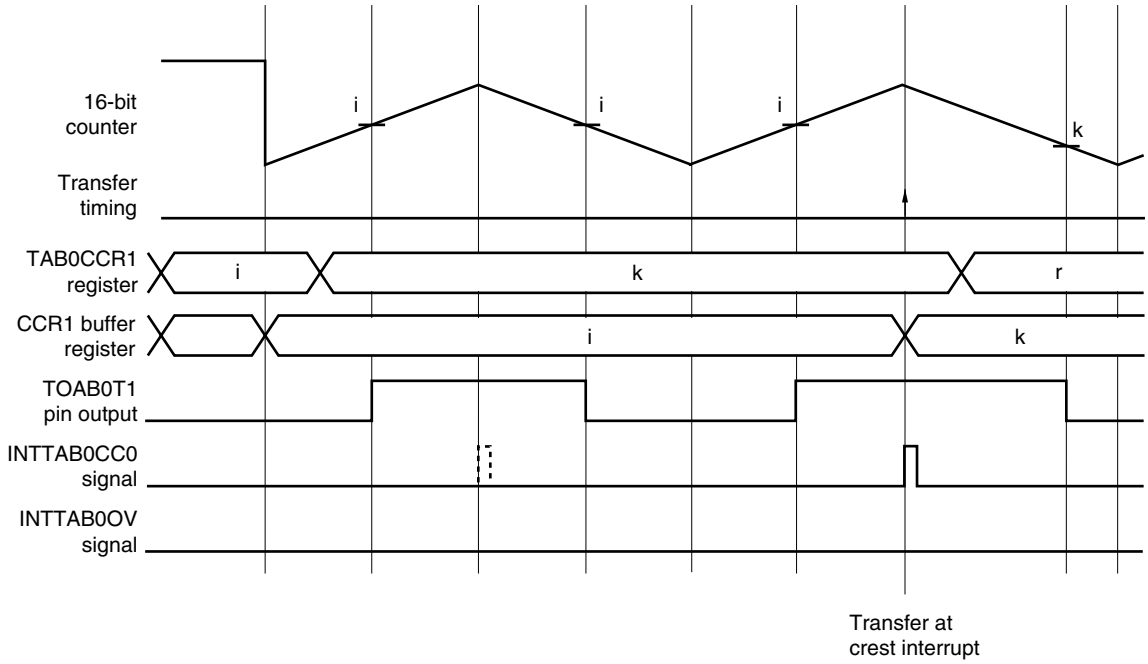
Remarks 1. This is an example of the operation when the TAB0OPT1.TAB0ICE bit = 0, TAB0OPT1.TAB0IOE bit = 1, TAB0OPT1.TAB0ID4 to TAB0OPT1.TAB0ID0 bits = 00001.

2. : Culled interrupt

(c) Rewriting TAB0CCR1 to TAB0CCR3 registers

- Transfer at crest when crest interrupt is set
  Because the register is transferred at the transfer timing of the crest interrupt, an asymmetrical triangular wave is output.

Figure 10-34. Rewriting TAB0CCR1 Register (TAB0OPT1.TAB0ICE Bit = 1, TAB0OPT1.TAB0IOE Bit = 0, TAB0OPT1.TAB0ID4 to TAB0OPT1.TAB0ID0 Bits = 00001)

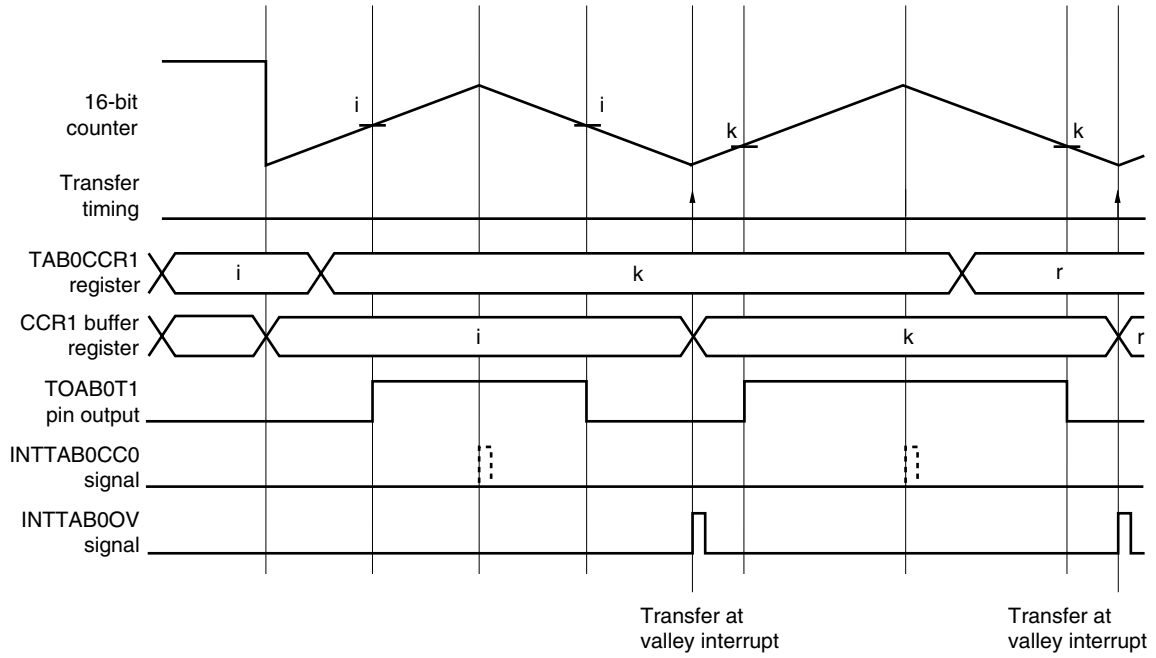


Remark : Culled interrupt

- Transfer at valley when valley interrupt is set

  Because the register is transferred at the transfer timing of the valley interrupt, a symmetrical triangular wave is output.

Figure 10-35. Rewriting TAB0CCR1 Register (TAB0OPT1.TAB0ICE Bit = 1, TAB0OPT1.TAB0IOE Bit = 1, TAB0OPT1.TAB0ID4 to TAB0OPT1.TAB0ID0 Bits = 00001)



Remark : Culled interrupt

(d) Rewriting TAB0OPT1 register

Because a new interrupt culling value is transferred when the value of the interrupt culling counter matches the value of the 16-bit counter, the next interrupt and those that follow occur at the set interval.

For details of rewriting the TAB0OPT1 register, see **10.4.3 Interrupt culling function**.

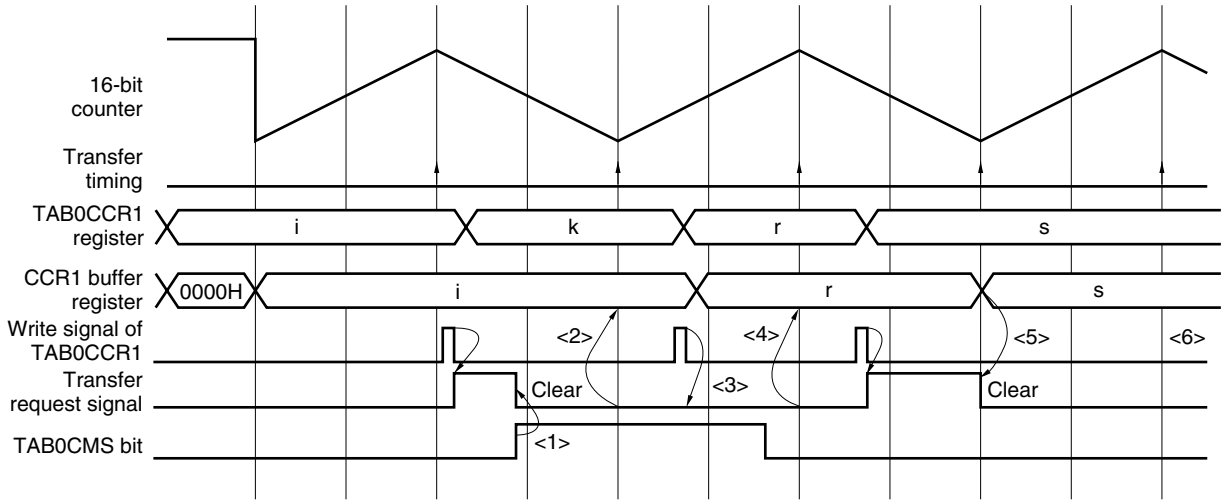
(4) Rewriting TAB0OPT0.TAB0CMS bit

The TAB0CMS bit can select the anytime rewrite mode and batch rewrite mode. This bit can be rewritten during timer operation (when TAB0CTL0.TAB0CE bit = 1). However, the operation and caution illustrated in Figure 10-36 are necessary.

If the TAB0CCR1 register is written when the TAB0CMS bit is cleared to 0, a transfer request signal (internal signal) is set.

When the transfer request signal is set, the register is transferred at the next transfer timing, and the transfer request signal is cleared. This transfer request signal is also cleared when the TAB0CMS bit is set to 1.

Figure 10-36. Rewriting TAB0CMS Bit



<1> If the TAB0CCR1 register is rewritten when the TAB0CMS bit is 0, the transfer request signal is set.
If the TAB0CMS bit is set to 1 in this status, the transfer request signal is cleared.

<2> The register is not transferred because the TAB0CMS bit is set to 1 and the transfer request signal is cleared.

<3> The transfer request signal is not set even if the TAB0CCR1 register is written when the TAB0CMS bit is 1.

<4> The transfer request signal is not set even if the TAB0CCR1 register is written when the TAB0CMS bit is 1, so even if the TAB0CMS bit is cleared to 0, transfer does not occur at the subsequent transfer timing.

<5> The transfer request signal is set if the TAB0CCR1 register is written when the TAB0CMS bit is 0.
Transfer is performed at the subsequent transfer timing and the transfer request signal is cleared.

<6> Once transfer has been performed, the transfer request signal is cleared. Therefore, transfer is not performed at the next transfer timing.

10.4.5 TAA4 tuning operation for A/D conversion start trigger signal output

This section explains the tuning operation of TAA4 and TAB0 in the 6-phase PWM output mode.

In the 6-phase PWM output mode, the tuning operation is performed with TAB0 serving as the master and TAA4 as a slave. The conversion start trigger signal of the A/D converter can be set as the A/D conversion start trigger source by the INTTAA4CC0 and INTTAA4CC1 signals of TAA4 and the INTTAB0OV and INTTAB0CC0 signals of TAB0.

(1) Tuning operation starting procedure

The TAA4 and TAB0 registers should be set using the following procedure to perform the tuning operation.

(a) Setting of TAA4 register (stop the operations of TAB0 and TAA4 (by clearing the TAB0CTL0.TAB0CE bit and TAA4CTL0.TAA4CE bit to 0)).

- Set the TAA4CTL1 register to 85H (set the tuning operation slave mode and free-running timer mode).
- Clear the TAA4OPT0 register to 00H (select the compare register).
- Set an appropriate value to the TAA4CCR0 and TAA4CCR1 registers (set the default value for comparison for starting the operation).

(b) Setting of TAB0 register

- Set the TAB0CTL1 register to 07H (master mode and 6-phase PWM output mode).
- Set an appropriate value to the TAB0IOC0 register (set the output mode of TOAB0T1 to TOAB0T3). However, clear the TAB0OL0 bit to 0 and set the TAB0OE0 bit to 1 (enable positive phase output). Unless this setting is made, the crest interrupt (INTTAB0CC0) and valley interrupt (INTTAB0OV) do not occur. Consequently, the conversion start trigger signal of the A/D converter is not correctly generated.
- Clear the TAB0OPT0 register to 00H (select the compare register).
- Set an appropriate value to the TAB0CCR0 to TAB0CCR3 registers (set the default value for comparison for starting the operation).
- Set the TAB0CTL0 register to 0xH (clear the TAB0CE bit to 0 and set the operating clock of TAB0). The operating clock of TAB0 set by the TAB0CTL0 register is also supplied to TAA4, and the count operation is performed at the same timing. The operating clock of TAA4 set by the TAA4CTL0 register is ignored.

(c) Setting of TABOP0 (TAB0 option) register

- Set an appropriate value to the TAB0OPT1 and TAB0OPT2 registers.
- Set an appropriate value to the TAB0IOC3 register (set TOAB0B1 to TOAB0B3 in the output mode).
- Set an appropriate value to the TAB0DTC register (set the default value for comparison for starting the operation).

(d) Setting of alternate function

- Set the alternate function to the port by setting the port control mode.

(e) Set the TAA4CE bit to 1 and set the TAB0CE bit to 1 immediately after that to start the 6-phase PWM output operation

Rewriting the TAB0CTL0, TAB0CTL1, TAB0IOC1, TAB0IOC2, TAA4CTL0, TAA4CTL1, TAA4IOC0, TAA4IOC1, and TAA4IOC2 registers is prohibited during operation. The operation and the PWM output waveform are not guaranteed if any of these registers is rewritten during operation. However, rewriting the TAB0CTL0.TAB0CE bit to clear it is permitted. Manipulating (reading/writing) the other TAB0, TAA4, and TAB0 option registers is prohibited until the TAA4CTL0.TAA4CE bit is set to 1 and then the TAB0CE bit is set to 1.

Caution When tuning TAA4 in the 6-phase PWM mode, output of the TOAA00 and TOAA01 pins is disabled. Clear the TAA0IOC0.TAA0OE0 and TAA0IOC0.TAA0OE1 bits to 0.

(2) Tuning operation clearing procedure

To clear the tuning operation and exit the 6-phase PWM output mode, set the TAA4 and TAB0 registers using the following procedure.

<1> Clear the TAB0CTL0.TAB0CE bit to 0 and stop the timer operation.
<2> Clear the TAA4CTL0.TAA4CE bit to 0 so that TAA4 can be separated.
<3> Stop the timer output by using the TAB0IOC0 and TAA4IOC0 registers.
<4> Clear the TAA4CTL1.TAA4SYE bit to 0 to clear the tuning operation.

Caution Manipulating (reading/writing) the other TAB0, TAA4, and TAB0 option registers is prohibited until the TAB0CE bit is set to 1 and then the TAA4CE bit is set to 1.

(3) When not tuning TAA4

When the match interrupt signal of TAA4 is not necessary as the conversion trigger source that starts the A/D converter, TAA4 can be used independently as a separate timer without being tuned. In this case, the match interrupt signal of TAA4 cannot be used as a trigger source to start A/D conversion in the 6-phase PWM output mode. Therefore, fix the TAB0OPT2.TAB0AT2 and TAB0OPT2.TAB0AT3 bits to 0.

The other control bits can be used in the same manner as when TAA4 is tuned.

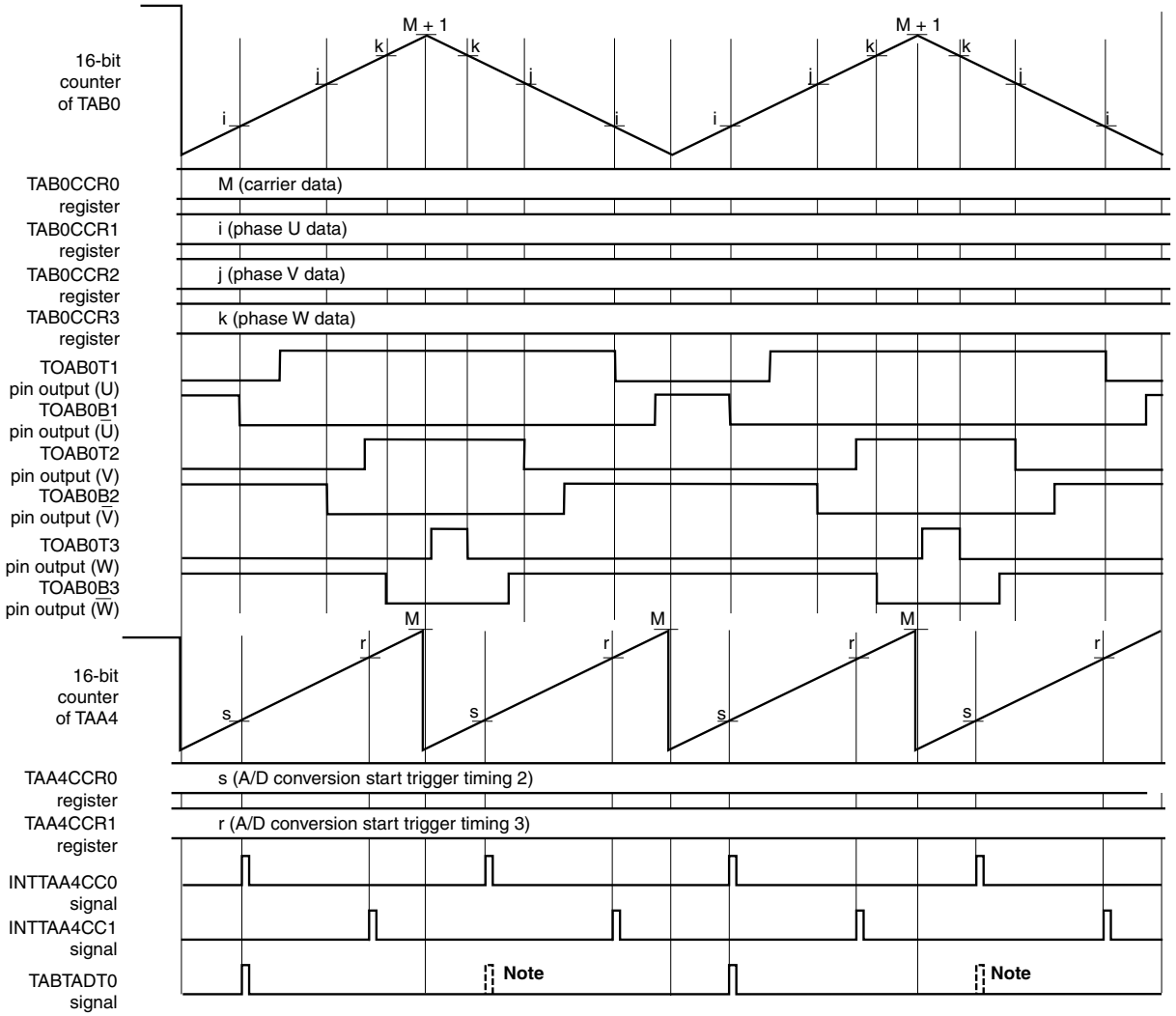
If TAA4 is not tuned, the compare registers (TAA4CCR0 and TAA4CCR1) of TAA4 are not affected by the setting of the TAB0OPT0.TAB0CMS and TAB0OPT2.TAB0RDE bit. For the initialization procedure when TAA4 is not tuned, see (b) to (e) in **10.4.5 (1) Tuning operation starting procedure**. (a) is not necessary because it is a step used to set TAA4 for the tuning operation.

(4) Basic operation of TAA4 during tuning operation

The 16-bit counter of TAA4 only counts up. The 16-bit counter is cleared by the set cycle value of the TAB0CCR0 register and starts counting from 0000H again. The count value of this counter is the same as the value of the 16-bit counter of TAB0 when it counts up. However, it is not the same when the 16-bit counter of TAA4 counts down.

- When TAB0 counts up (same value)
  16-bit counter of TAB0: 0000H → M (counting up)
  16-bit counter of TAA4: 0000H → M (counting up)
- When TAB0 counts down (not same value)
  16-bit counter of TAB0: M + 1 → 0001H (counting down)
  16-bit counter of TAA4: 0000H → M (counting up)

Figure 10-37. TAA4 During Tuning Operation



Note The TABTADT0 signal is masked by the TAB0OPT2.TAB0ATM2 and TAB0OPT2.TAB0ATM3 bits.

10.4.6 A/D conversion start trigger output function

The V850E/Hx3 has a function to select four trigger sources (INTTAB0OV, INTTAB0CC0, INTTAA4CC0, INTTAA4CC1) to generate the A/D conversion start trigger signal (TABTADT0).

The trigger sources are specified by the TAB0OPT2.TAB0AT0 to TAB0OPT2.TAB0AT3 bits.

- TAB0AT0 bit = 1:
  A/D conversion start trigger signal generated when INTTAB0OV (counter underflow) occurs.
- TAB0AT1 bit = 1:
  A/D conversion start trigger signal generated when INTTAB0CC0 (cycle match) occurs.
- TAB0AT2 bit = 1:
  A/D conversion start trigger signal generated when INTTAA4CC0 (match of TAA4CCR0 register of TAA4 during tuning operation) occurs.
- TAB0AT3 bit = 1:
  A/D conversion start trigger signal generated when INTTAA4CC1 (match of TAA4CCR1 register of TAA4 during tuning operation) occurs.

The A/D conversion start trigger signals selected by the TAB0AT0 to TAB0AT3 bits are ORed and output. Therefore, two or more trigger sources can be specified at the same time.

The INTTAB0OV and INTTAB0CC0 signals selected by the TAB0AT0 and TAB0AT1 bits are culled interrupt signals.

Therefore, these signals are output after the interrupts have been culled and, unless interrupt output is enabled (TAB0OPT1.TAB0ICE, TAB0OPT1.TAB0IOE bits), the A/D conversion start trigger signal is not output.

The trigger sources (INTTAA4CC0 and INTTAA4CC1) from TAA4 have a function to mask the A/D conversion start trigger signal depending on the status of the count-up/count-down of the 16-bit counter, if so set by the TAB0AT2 and TAB0AT3 bits.

- TAB0ATM2 bit: Correspond to the TAB0AT2 bit and control INTTAA4CC0 (match interrupt signal) of TAA4.
  - TAB0ATM2 bit = 0: The A/D conversion start trigger signal is output when the 16-bit counter counts up (TAB0OPT0.TAB0CUF bit = 0), and the A/D conversion start trigger signal is not output when the 16-bit counter counts down (TAB0OPT0.TAB0CUF bit = 1).
  - TAB0ATM2 bit = 1: The A/D conversion start trigger signal is output when the 16-bit counter counts up (TAB0OPT0.TAB0CUF bit = 1), and the A/D conversion start trigger signal is not output when the 16-bit counter counts down (TAB0OPT0.TAB0CUF bit = 0).

- TAB0ATM3 bit: Correspond to the TAB0AT3 bit and control INTTAA4CC1 (match interrupt signal) of TAA4.
  - TAB0ATM3 bit = 0: The A/D conversion start trigger signal is output when the 16-bit counter counts up (TAB0OPT0.TAB0CUF bit = 0), and the A/D conversion start trigger signal is not output when the 16-bit counter counts down (TAB0OPT0.TAB0CUF bit = 1).
  - TAB0ATM3 bit = 1: The A/D conversion start trigger signal is output when the 16-bit counter counts up (TAB0OPT0.TAB0CUF bit = 1), and the A/D conversion start trigger signal is not output when the 16-bit counter counts down (TAB0OPT0.TAB0CUF bit = 0).

The TAB0ATM3, TAB0ATM2, and TAB0AT3 to TAB0AT0 bits can be rewritten while the timer is operating. If the bit that sets the A/D conversion start trigger signal is rewritten while the timer is operating, the new setting is immediately reflected on the output status of the A/D conversion start trigger signal. These control bits do not have a transfer function and can be used only in the anytime rewrite mode.

Cautions 1. The A/D conversion start trigger signal output that is set by the TAB0AT2 and TAB0AT3 bits can be used only when TAA4 is performing a tuning operation as the slave timer of TAB0. If TAB0 and TAA4 are not performing a tuning operation, or if a mode other than the 6-phase PWM output mode is used, the output cannot be guaranteed.

2. The TAB0 signal output is internally used to identify whether the 16-bit counter is counting up or down. Therefore, enable TOAB00 pin output by clearing the TAB0IOC0.TAB0OL0 bit to 0 and setting the TAB0IOC0.TAB0OE0 bit to 1.

Figure 10-38. Example of A/D Conversion Start Trigger (TABTADT0) Signal Output (TAB0OPT1.TAB0ICE Bit = 1, TAB0OPT1.TAB0IOE Bit = 1, TAB0OPT1.TAB0ID4 to TAB0OPT1.TAB0ID0 Bits = 00000: Without Interrupt Culling)

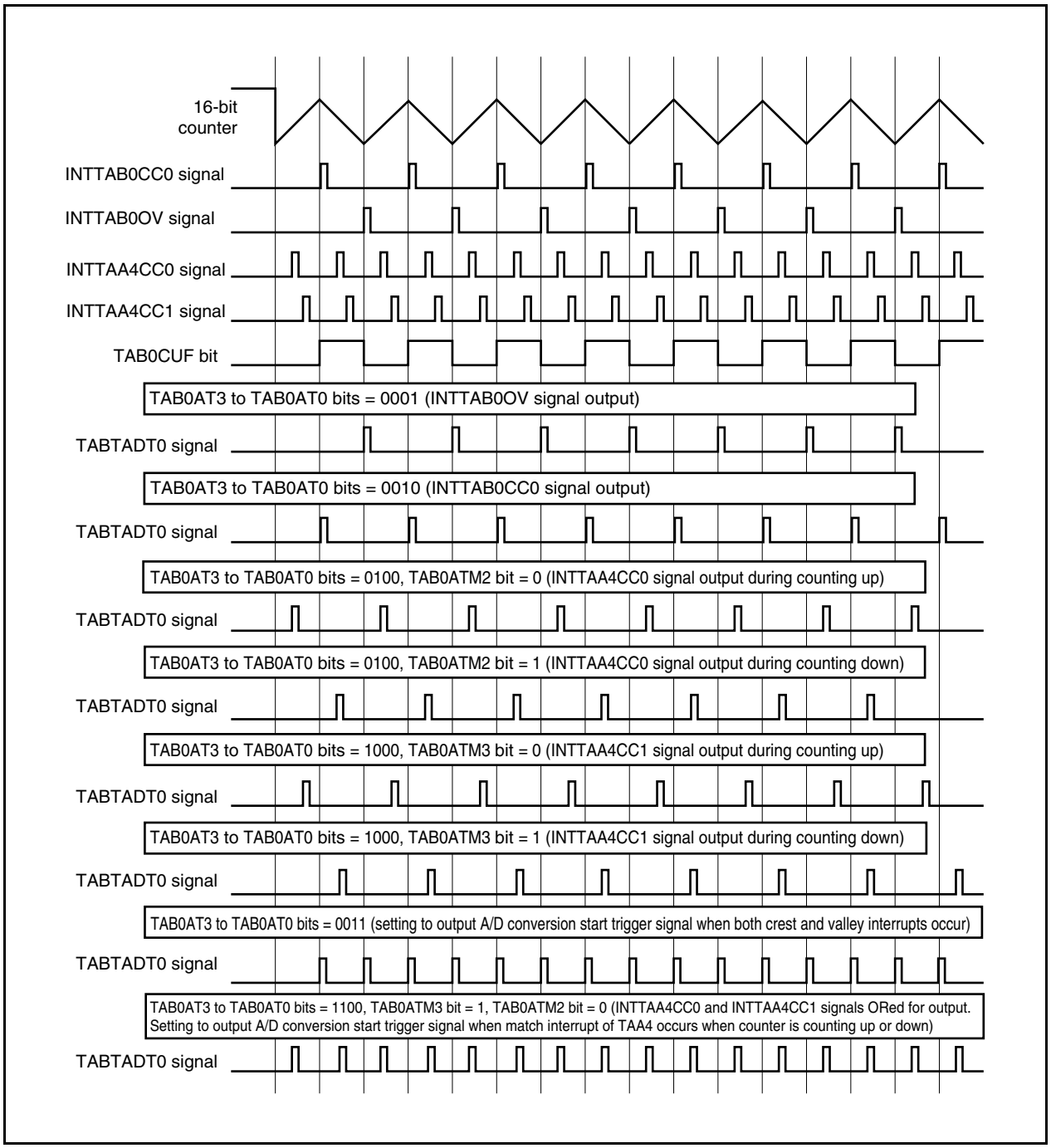
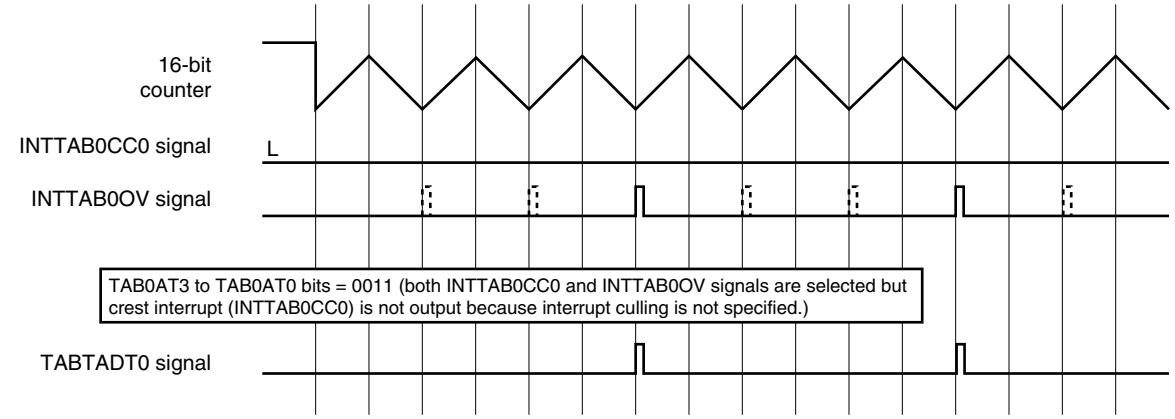
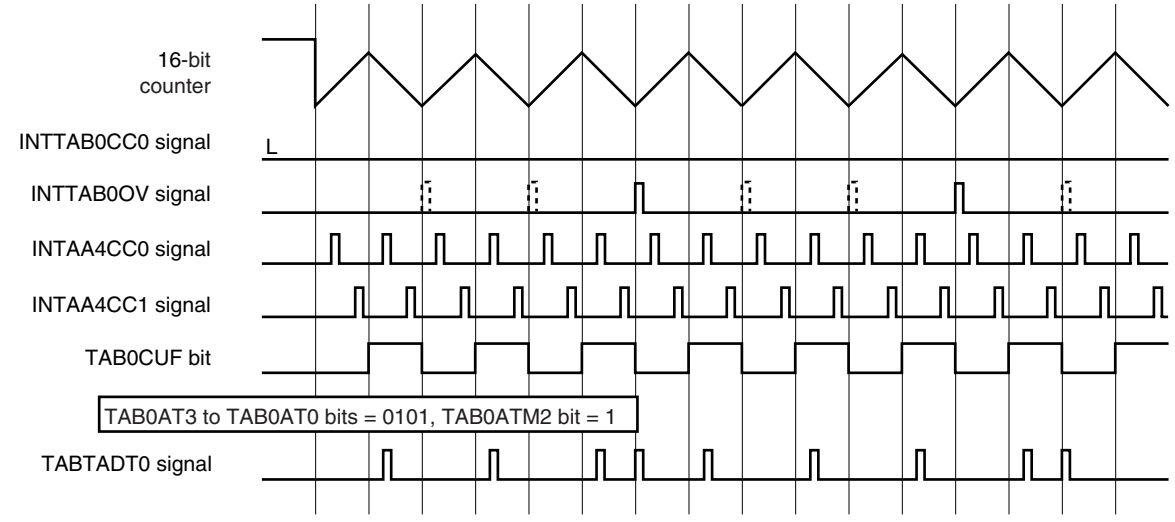


Figure 10-39. Example of A/D Conversion Start Trigger (TABTADT0) Signal Output (TAB0OPT1.TAB0ICE Bit = 0, TAB0OPT1.TAB0IOE Bit = 1, TAB0OPT1.TAB0ID4 to TAB0OPT1.TAB0ID0 Bits = 00010: With Interrupt Culling) (1)



Remark : Culled interrupt

Figure 10-40. Example of A/D Conversion Start Trigger (TABTADT0) Signal Output (TAB0OPT1.TAB0ICE Bit = 0, TAB0OPT1.TAB0IOE Bit = 1, TAB0OPT1.TAB0ID4 to TAB0OPT1.TAB0ID0 Bits = 00010: With Interrupt Culling) (2)



Caution The INTTAB0CC0 signal is culled but the INTTAA4CC0 signal is not.

Remark : Culled interrupt

(1) Operation under boundary condition (operation when 16-bit counter matches INTTAA4CC0 signal)

Table 10-3. Operation When TAB0CCR0 Register = M, TAB0AT2 Bit = 1, TAB0ATM2 Bit = 0 (Counting Up Period Selected)

Value of TAA4CCR0 Register	Value of 16-Bit Counter of TAB0	Value of 16-Bit Counter of TAA4	Status of 16-Bit Counter of TAB0	TABTADT0 Signal Output by INTTAA4CC0 Signal
0000H	0000H	0000H	–	Output
0000H	M + 1	0000H	–	Not output
0001H	0001H	0001H	Count-up	Output
0001H	M	0001H	Count-down	Not output
M	M	M	Count-up	Output
M	0001H	M	Count-down	Not output

Table 10-4. Operation When TAB0CCR0 Register = M, TAB0AT2 Bit = 1, TAB0ATM2 Bit = 1 (Counting Down Period Selected)

Value of TAA4CCR0 Register	Value of 16-Bit Counter of TAB0	Value of 16-Bit Counter of TAA4	Status of 16-Bit Counter of TAB0	TABTADT0 Signal Output by INTTAA4CC0 Signal
0000H	0000H	0000H	–	Not output
0000H	M + 1	0000H	–	Output
0001H	0001H	0001H	Count-up	Not output
0001H	M	0001H	Count-down	Output
M	M	M	Count-up	Not output
M	0001H	M	Count-down	Output

Caution The TAA4CCRm register enables setting of "0" to "M" when the TAB0CCR0 register = M. Setting of a value of "M + 1" or higher is prohibited.
If a value higher than "M + 1" is set, the 16-bit counter of TAA4 is cleared by "M". Therefore, the TABTADT0 signal is not output.

CHAPTER 11 WATCH TIMER FUNCTIONS

11.1 Functions

The watch timer has the following functions.

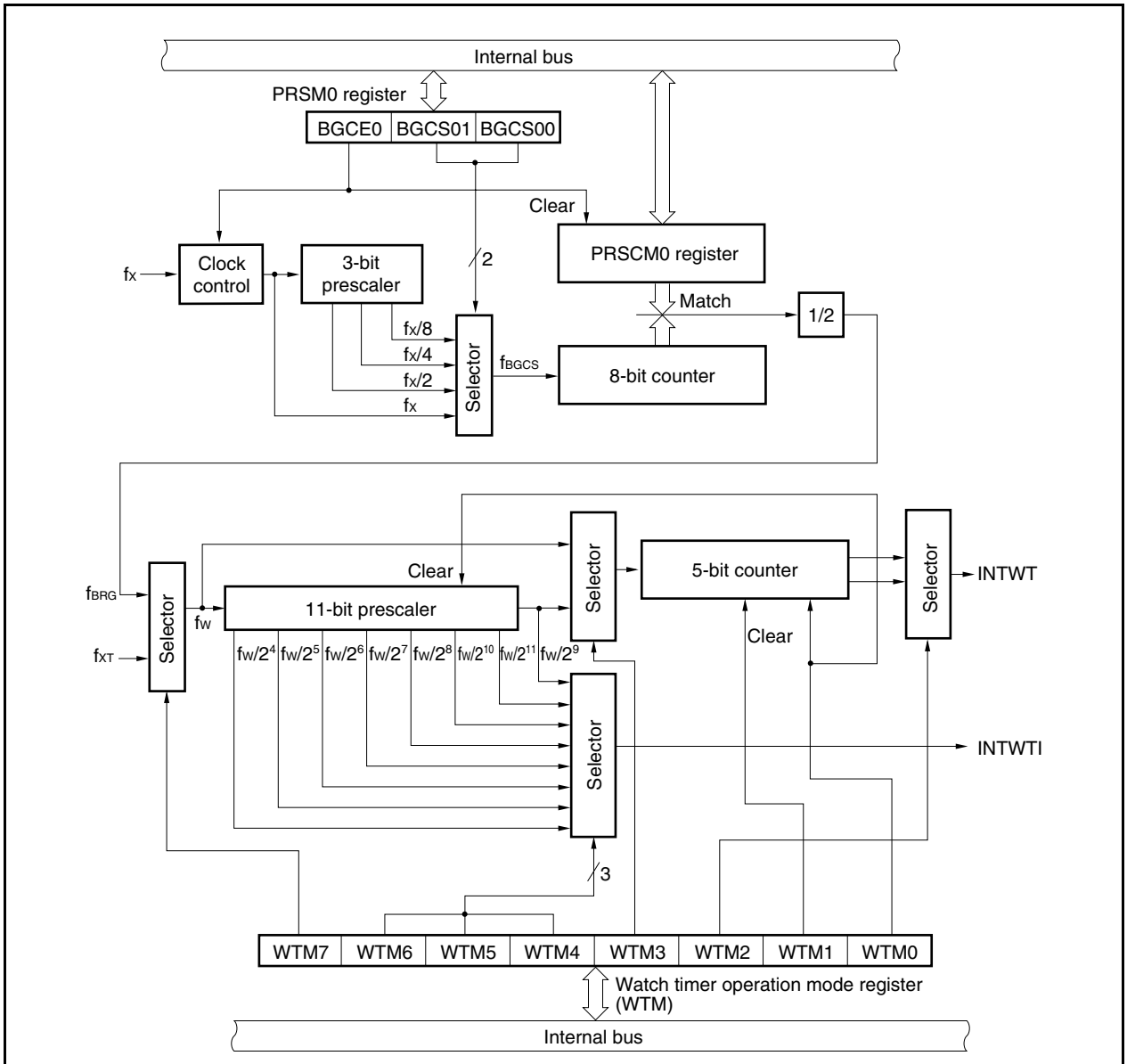
- Watch timer: An interrupt request signal (INTWT) is generated at intervals of 0.5 or 0.25 seconds by using the main clock or subclock.
- Interval timer: An interrupt request signal (INTWTI) is generated at set intervals.

The watch timer and interval timer functions can be used at the same time.

11.2 Configuration

The block diagram of the watch timer is shown below.

Figure 11-1. Block Diagram of Watch Timer



Remark f_X: Main clock oscillation frequency
f_{BGCS}: Watch timer source clock frequency
f_{BRG}: Watch timer count clock frequency
f_{XT}: Subclock frequency
f_W: Watch timer clock frequency
INTWT: Watch timer interrupt request signal
INTWTI: Interval timer interrupt request signal

(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 8-bit 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 (INTWT) at intervals of $2^4/f_W$, $2^5/f_W$, $2^{13}/f_W$, 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) PRSCM register

This is an 8-bit compare register that sets the interval time.

(8) PRSM 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.

11.3 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})		
			10 MHz	8 MHz
0	0	f_X	100 ns	125 ns
0	1	$f_X/2$	200 ns	250 ns
1	0	$f_X/4$	400 ns	500 ns
1	1	$f_X/8$	800 ns	1 μs

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.

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

(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 PRSCM0 register = 1 to 256

However, N = 256 only when 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 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 μs: $f_W = f_{XT}$)
0	0	0	1	$2^5/f_W$ (977 μs: $f_W = f_{XT}$)
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 μs: $f_W = f_{BRG}$)
1	0	0	1	$2^5/f_W$ (977 μs: $f_W = f_{BRG}$)
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 μs: $f_W = f_{XT}$)
0	1	1	$2^4/f_W$ (488 μs: $f_W = f_{XT}$)
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 μs: $f_W = f_{BRG}$)
1	1	1	$2^4/f_W$ (488 μs: $f_W = f_{BRG}$)

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

3. f_{XT}: Subclock frequency

4. f_{BRG}: Watch timer count clock frequency

11.4 Operation

11.4.1 Operation as watch timer

The watch timer generates an interrupt request signal (INTWT) at fixed time intervals. The watch timer operates using time intervals of 0.25 or 0.5 seconds with the subclock (32.768 kHz) or main clock.

The count operation 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 the count operation stops.

The time of the watch timer can be adjusted by clearing the WTM1 bit to 0 and then the 5-bit counter when operating at the same time as the interval timer. At this time, an error of up to 15.6 ms may occur for 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, the 8-bit comparison value using the PRSCM0 register, and the count clock frequency (f_{BRG}) of the watch timer to 32.768 kHz.

When the PRSM0.BGCE0 bit is set (1), f_{BRG} is supplied to the watch timer.

f_{BRG} can be calculated by the following expression.

$$f_{BRG} = f_X/(2^{m+1} \times N)$$

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 = f_X/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 f_X = 8.00 MHz
<1> N = 8,000,000/65,536 = 122.07…, m = 0
<2>, <3> Because N (round up the first decimal place) is even, N = N/2 (= 61), m = m + 1 (= 1).
<4> Set value of PRSCM0 register: 3DH (61), set value of BGCS01 and BGCS00 bits: 01

At this time, the actual f_{BRG} frequency is as follows.

$f_{BRG} = f_X/(2^{m+1} \times N) = 8{,}000{,}000/(4 \times 61)$
$= 32.787$ kHz

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 only when PRSCM0 register is set to 00H.
f_X: Main clock oscillation frequency

11.4.2 Operation as interval timer

The watch timer can also be used as an interval timer that repeatedly generates an interrupt request signal (INTWTI) at intervals specified by a preset count value.

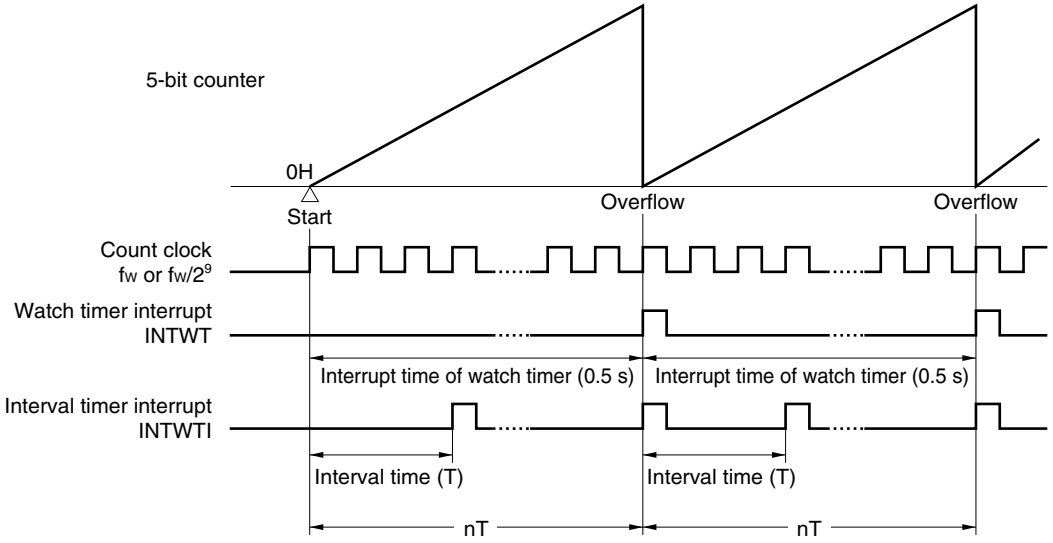
The interval time can be selected by the WTM4 to WTM7 bits of the WTM register.

Table 11-1. Interval Time of Interval Timer

WTM7	WTM6	WTM5	WTM4	Interval Time	
0	0	0	0	$2^4 \times 1/fw$	488 μs (operating at $f_W = f_{XT}$ = 32.768 kHz)
0	0	0	1	$2^5 \times 1/fw$	977 μs (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 μs (operating at $f_W = f_{BRG}$ = 32.768 kHz)
1	0	0	1	$2^5 \times 1/fw$	977 μs (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 f_W: Watch timer clock frequency

Figure 11-2. Operation Timing of Watch Timer/Interval Timer



Remarks 1. When 0.5 seconds of the watch timer interrupt time is set.

2. fw: Watch timer clock frequency

Values in parentheses apply to operation with fw = 32.768 kHz.

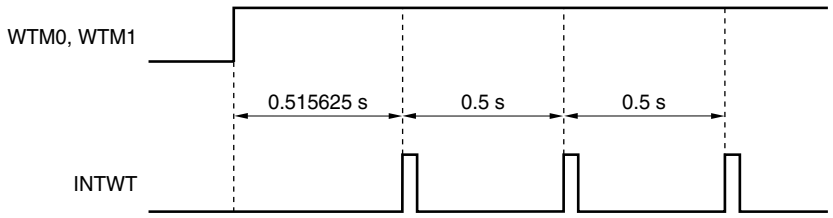
n: Number of interval timer operations

11.4.3 Cautions

Some time is required before the first watch timer interrupt request signal (INTWT) is generated after operation is enabled (WTM.WTM1 and WTM.WTM0 bits = 1).

Figure 11-3. Example of Generation of Watch Timer Interrupt Request Signal (INTWT)
(When Interrupt Cycle = 0.5 s)

It takes 0.515625 seconds (max.) for the first INTWT signal to be generated ($2^9 \times 1/32768 = 0.015625$ seconds longer (max.)). The INTWT signal is then generated every 0.5 seconds.



CHAPTER 12 FUNCTIONS OF WATCHDOG TIMER 2

12.1 Functions

Watchdog timer 2 has the following functions.

- Default-start watchdog timer[Note 1]
  → Reset mode: Reset operation upon overflow of watchdog timer 2 (generation of WDT2RES signal)
  → Non-maskable interrupt request mode: NMI operation upon overflow of watchdog timer 2 (generation of INTWDT2 signal)[Note 2]
- Input selectable from main clock oscillation and low-speed internal oscillation clock as the source clock

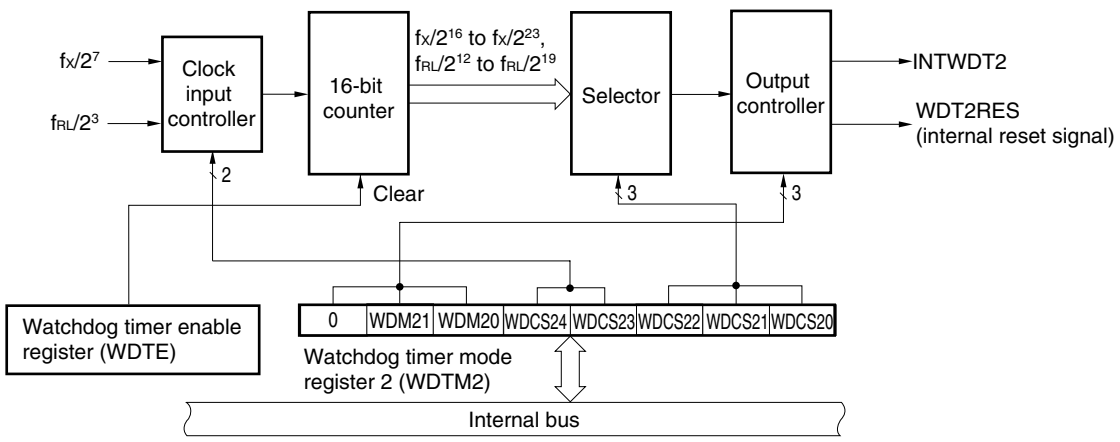
Notes 1. Watchdog timer 2 automatically starts in the reset mode following reset release.
When watchdog timer 2 is not used, either stop its operation before reset is executed via this function, or clear watchdog timer 2 once and stop it within the next interval time.
Also, write to the WDTM2 register for verification purposes only once, even if the default settings (reset mode, interval time: $f_{RL}/2^{19}$) do not need to be changed.

2. For the non-maskable interrupt servicing due to a non-maskable interrupt request signal (INTWDT2), see **18.2.2 (2) INTWDT2 signal**.

12.2 Configuration

The following shows the block diagram of watchdog timer 2.

Figure 12-1. Block Diagram of Watchdog Timer 2



Remark f_X: Main clock oscillation frequency
f_{RL}: Low-speed internal oscillation clock frequency
INTWDT2: Non-maskable interrupt request signal from watchdog timer 2
WDTRES2: Watchdog timer 2 reset signal

Watchdog timer 2 includes the following hardware.

Table 12-1. Configuration of Watchdog Timer 2

Item	Configuration
Control registers	Watchdog timer mode register 2 (WDTM2) Watchdog timer enable register (WDTE)

12.3 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. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- **When the CPU operates with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the internal oscillation 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 12-2 Watchdog Timer 2 Clock Selection.

2. If the WDTM2 register is rewritten twice after reset, an overflow signal is forcibly generated and the counter is reset.

3. To intentionally generate an overflow signal, write to the WDTM2 register only 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 only twice, or a value other than "ACH" is written to the WDTE register only once.

4. To stop the operation of watchdog timer 2, set the RCM.RSTOP bit to 1 (low-speed internal oscillator is stopped), and write 1FH to the WDTM2 register.

Table 12-2. Watchdog Timer 2 Clock Selection

WDCS24	WDCS23	WDCS22	WDCS21	WDCS20	Selected Clock	204 kHz (MIN.)	240 kHz (TYP.)	276 kHz (MAX.)
0	0	0	0	0	$2^{12}/f_{RL}$	20.1 ms	17.1 ms	14.8 ms
0	0	0	0	1	$2^{13}/f_{RL}$	40.2 ms	34.1 ms	29.7 ms
0	0	0	1	0	$2^{14}/f_{RL}$	80.3 ms	68.3 ms	59.4 ms
0	0	0	1	1	$2^{15}/f_{RL}$	160.6 ms	136.5 ms	118.7 ms
0	0	1	0	0	$2^{16}/f_{RL}$	321.3 ms	273.1 ms	237.4 ms
0	0	1	0	1	$2^{17}/f_{RL}$	642.5 ms	546.1 ms	474.9 ms
0	0	1	1	0	$2^{18}/f_{RL}$	1285.0 ms	1092.3 ms	949.8 ms
0	0	1	1	1	$2^{19}/f_{RL}$	2570.0 ms	2184.5 ms	1899.6 ms
						f_X = 8 MHz		f_X = 10 MHz
0	1	0	0	0	$2^{16}/f_X$	8.2 ms		6.6 ms
0	1	0	0	1	$2^{17}/f_X$	16.4 ms		13.1 ms
0	1	0	1	0	$2^{18}/f_X$	32.8 ms		26.2 ms
0	1	0	1	1	$2^{19}/f_X$	65.5 ms		52.4 ms
0	1	1	0	0	$2^{20}/f_X$	131.1 ms		104.9 ms
0	1	1	0	1	$2^{21}/f_X$	262.1 ms		209.7 ms
0	1	1	1	0	$2^{22}/f_X$	524.3 ms		419.4 ms
0	1	1	1	1	$2^{23}/f_X$	1048.6 ms		838.9 ms
1	1	1	1	1	Operation stopped			

(2) Watchdog timer enable register (WDTE)

The counter of watchdog timer 2 is cleared and counting restarted by writing “ACH” to the WDTE register.

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

Reset sets this register to 9AH.

After reset: 9AH R/W Address: FFFFF6D1H

	7	6	5	4	3	2	1	0
WDTE								

Cautions 1. When a value other than “ACH” is written to the WDTE register, an overflow signal is forcibly output.

2. When a 1-bit memory manipulation instruction is executed for the WDTE register, an overflow signal is forcibly output.

3. To intentionally generate an overflow signal, write to the WDTM2 register only twice or write a value other than ACH to the WDTE register once.

However, when the watchdog timer 2 is set to stop operation, an overflow signal is not generated even if data is written to the WDTM2 register only twice, or a value other than “ACH” is written to the WDTE register only once.

4. The read value of the WDTE register is “9AH” (which differs from written value “ACH”).

12.4 Operation

Watchdog timer 2 automatically starts in the reset mode following reset release.

The WDTM2 register can be written to only once following reset using byte access. To use watchdog timer 2, write the operation mode and the interval time 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 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 set values of the WDM21 and WDTM2.WDM20 bits.

When the WDTM2.WDM21 bit is set to 1 (reset mode), if a WDT overflow occurs during oscillation stabilization after a reset or standby is released, no internal reset will occur and the CPU clock will switch to the internal oscillation clock.

To not use watchdog timer 2, write 1FH to the WDTM2 register.

For the non-maskable interrupt servicing while the non-maskable interrupt request mode is set, see **18.2.2 (2) INTWDT2 signal**.

CHAPTER 13 A/D CONVERTER

13.1 Overview

The A/D converter of the V850ES/Hx3 has a resolution of 10 bits and converts an input analog signal into a digital value.

The number of A/D converter in the V850ES/Hx3 is shown below.

Product Name	V850ES/HE3	V850ES/HF3	V850ES/HG3	V850ES/HJ3
Number of ANI channels (m)	10 channels (m = 10)	12 channels (m = 12)	16 channels (m = 16)	24 channels (m = 24)
Number of mounted (n)	ANI0 to ANI9 (n = 0 to 9)	ANI0 to ANI11 (n = 0 to 11)	ANI0 to ANI14 (n = 0 to 14)	ANI0 to ANI23 (n = 0 to 23)

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} = 4.0 to 5.5 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
○ Power-fail monitor function (conversion result compare function)

13.2 Functions

(1) 10-bit resolution A/D conversion

An analog input channel is selected from ANIn, and an A/D conversion operation is repeated at a resolution of 10 bits. Each time A/D conversion has been completed, an interrupt request signal (INTAD) is generated.

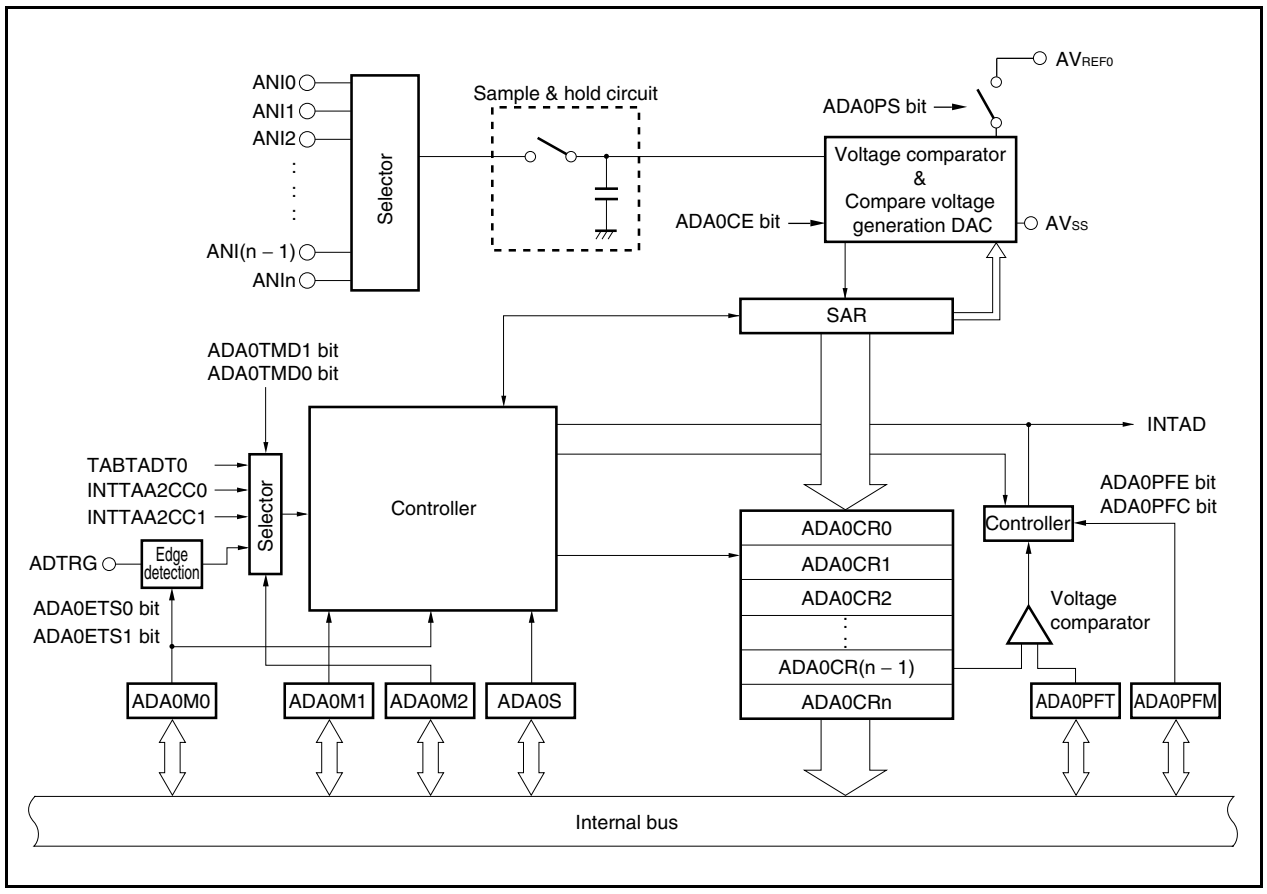
(2) Power-fail detection function

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 a specified comparison condition is satisfied.

13.3 Configuration

The block diagram of the A/D converter is shown below.

Figure 13-1. Block Diagram of A/D Converter



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 register n (ADA0CRn) A/D conversion result register nH (ADCRnH): Only higher 8 bits can be read
Control registers	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)

(1) Successive approximation register (SAR)

The SAR register compares the voltage value of the analog input signal with the output voltage of the compare voltage generation DAC (compare voltage), and holds the comparison result starting from the most significant bit (MSB).

When the comparison result has been held down to the least significant bit (LSB) (i.e., when A/D conversion is complete), the contents of the SAR register are transferred to the ADA0CRn register.

(2) A/D conversion result register n (ADA0CRn), A/D conversion result register nH (ADA0CRnH)

The ADA0CRn register is a 16-bit register that stores the A/D conversion result. ADA0ARn consist of an m number of registers and the A/D conversion result is stored in the 10 higher bits of the AD0CRn register corresponding to analog input. (The lower 6 bits are fixed to 0.)

(3) A/D converter mode register 0 (ADA0M0)

This register specifies the operation mode and controls the conversion operation by the A/D converter.

(4) A/D converter mode register 1 (ADA0M1)

This register sets the conversion time of the analog input signal to be converted.

(5) A/D converter mode register 2 (ADA0M2)

This register sets the hardware trigger mode.

(6) A/D converter channel specification register (ADA0S)

This register sets the input port that inputs the analog voltage to be converted.

(7) Power-fail compare mode register (ADA0PFM)

This register sets the power-fail monitor mode.

(8) Power-fail compare threshold value register (ADA0PFT)

The ADA0PFT register sets a 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).

(9) Controller

The controller compares the result of the 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 a specified comparison condition is satisfied.

(10) 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.

(11) Voltage comparator

The voltage comparator compares a voltage value that has been sampled and held with the voltage value of the compare voltage generation DAC.

(12) Compare voltage generation DAC

This compare voltage generation DAC is connected between AV_{REF0} and AV_{SS} and generates a voltage for comparison with the analog input signal.

(13) ANIn pin

These are analog input pins for an m number of channels of the A/D converter and are used to input analog signals to be converted into digital signals. Pins other than the one selected as the analog input by the ADA0S register can be used as input port pins.

Cautions 1. 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. The analog input pins (ANIn) function alternately as input port pins (P70 to P715, P120 to P127). If any of the ANIn pins is selected to execute A/D conversion, do not execute an instruction that reads an input port or that writes an output port during conversion. If executed, the conversion resolution may be degraded.

(14) AV_{REF0} pin

This is the pin used to input the reference voltage of the A/D converter.

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.

Be sure to satisfy the condition $3.7 \leq AV_{REF0} \leq 5.5$ V even when the A/D converter is not used.

(15) AV_{SS} pin

This is the ground pin of the A/D converter. Always make the potential at this pin the same as that at 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 operations.

This register can be read or written in 8-bit or 1-bit units. However, ADA0EF bit is read-only.

Reset sets this register to 00H.

Caution Accessing the ADA0M0 register is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- **When the CPU operates with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the low-speed internal oscillation clock**

After reset: 00H R/W Address: FFFFF200H

	7	6	5	4	3	2	1	0
ADA0M0	ADA0CE	ADA0PS	ADA0MD1	ADA0MD0	ADA0ETS1	ADA0ETS0	ADA0TMD	ADA0EF

ADA0CE	A/D conversion control
0	Stops A/D conversion
1	Enables A/D conversion

ADA0PS	A/D power control
0	A/D converter power off
1	A/D converter power on
After the ADA0PS bit has been set to 1 and the stabilization time has elapsed, the conversion result becomes valid from the first result if A/D conversion operation is enabled (ADA0CE bit = 1).	

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

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 specification
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. Write operations to bit 0 are ignored.

2. Changing the ADA0M1 register value is prohibited while A/D conversion is enabled (ADA0CE bit = 1).

3. If the ADA0M0, ADA0M2, ADA0S, ADA0PFM, and ADA0PFT registers are written 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 state is set.

4. When not using the A/D converter, stop the operation by setting the ADA0CE bit to 0 and set the ADA0PS bit (power off) to 0 to reduce the power consumption.

5. The resolution for the first conversion of the data of the input pin immediately after the start of A/D conversion may be degraded. For details, see 13.6 (7) AV_{REF0} pin.

(2) A/D converter mode register 1 (ADA0M1)

The ADA0M1 register is an 8-bit register that controls the conversion time specification.

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	0	0	0	0	ADA0FR3	ADA0FR2	ADA0FR1	ADA0FR0

Cautions 1. Be sure to set bits 7 to 4 to "0".

2. Setting the ADA0FR3 to ADA0FR0 bits is prohibited during the conversion operation (ADA0M0.ADA0CE bit = 1).

Remark For A/D conversion time setting examples, see **Table 13-2**.

Table 13-2. Conversion Mode Setting Example

ADA0FR3 to ADA0FR0				A/D Conversion Time							Sampling Time
3	2	1	0		fxx = 32 MHz	fxx = 24 MHz	fxx = 20 MHz	fxx = 16 MHz	fxx = 10 MHz	fxx = 4 MHz	
0	0	0	0	32/fxx	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	3.20 μs	8.00 μs	17/fxx
0	0	0	1	64/fxx	Setting prohibited	Setting prohibited	3.20 μs	4.00 μs	6.40 μs	16.00 μs	34/fxx
0	0	1	0	96/fxx	Setting prohibited	4.00 μs	4.80 μs	6.00 μs	9.60 μs	Setting prohibited	51/fxx
0	0	1	1	128/fxx	4.00 μs	5.34 μs	6.40 μs	8.00 μs	12.80 μs	Setting prohibited	68/fxx
0	1	0	0	160/fxx	5.00 μs	6.67 μs	8.00 μs	10.00 μs	16.00 μs	Setting prohibited	85/fxx
0	1	0	1	192/fxx	6.00 μs	8.00 μs	9.60 μs	12.00 μs	Setting prohibited	Setting prohibited	102/fxx
0	1	1	0	224/fxx	7.00 μs	9.34 μs	11.20 μs	14.00 μs	Setting prohibited	Setting prohibited	119/fxx
0	1	1	1	256/fxx	8.00 μs	10.67 μs	12.80 μs	16.00 μs	Setting prohibited	Setting prohibited	136/fxx
1	0	0	0	288/fxx	9.00 μs	12.00 μs	14.40 μs	Setting prohibited	Setting prohibited	Setting prohibited	153/fxx
1	0	0	1	320/fxx	10.00 μs	13.34 μs	16.00 μs	Setting prohibited	Setting prohibited	Setting prohibited	170/fxx
1	0	1	0	352/fxx	11.00 μs	14.67 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	187/fxx
1	0	1	1	384/fxx	12.00 μs	16.00 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	204/fxx
1	1	0	0	416/fxx	13.00 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	221/fxx
1	1	0	1	448/fxx	14.00 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	238/fxx
1	1	1	0	480/fxx	15.00 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	255/fxx
1	1	1	1	512/fxx	16.00 μs	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	272/fxx

Caution Set as 3.2 μs ≤ conversion time ≤ 16.0 μs.

(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 detected)
0	1	Timer trigger mode 0 (when INTTAA2CC0 interrupt request generated)
1	0	Timer trigger mode 1 (when INTTAA2CC1 interrupt request generated)
1	1	Timer trigger mode 2 (when TABTADT0[Note] signal generated)

Note TABTADT0: Timer trigger signal from 6-phase PWM output circuit (motor control). For details, see **10.4.5 TAA4 tuning operation for A/D conversion start trigger signal output**.

Caution Be sure to set bits 7 to 2 to "0".

(4) A/D converter 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	ADA0S4	ADA0S3	ADA0S2	ADA0S1	ADA0S0

ADA0S4	ADA0S3	ADA0S2	ADA0S1	ADA0S0	Select mode	Scan mode
0	0	0	0	0	ANI0	ANI0
0	0	0	0	1	ANI1	ANI0, ANI1
0	0	0	1	0	ANI2	ANI0 to ANI2
0	0	0	1	1	ANI3	ANI0 to ANI3
0	0	1	0	0	ANI4	ANI0 to ANI4
0	0	1	0	1	ANI5	ANI0 to ANI5
0	0	1	1	0	ANI6	ANI0 to ANI6
0	0	1	1	1	ANI7	ANI0 to ANI7
0	1	0	0	0	ANI8	ANI0 to ANI8
0	1	0	0	1	ANI9	ANI0 to ANI9
0	1	0	1	0	ANI10	ANI0 to ANI10
0	1	0	1	1	ANI11	ANI0 to ANI11
0	1	1	0	0	ANI12	ANI0 to ANI12
0	1	1	0	1	ANI13	ANI0 to ANI13
0	1	1	1	0	ANI14	ANI0 to ANI14
0	1	1	1	1	ANI15	ANI0 to ANI15
1	0	0	0	0	ANI16	ANI0 to ANI16
1	0	0	0	1	ANI17	ANI0 to ANI17
1	0	0	1	0	ANI18	ANI0 to ANI18
1	0	0	1	1	ANI19	ANI0 to ANI19
1	0	1	0	0	ANI20	ANI0 to ANI20
1	0	1	0	1	ANI21	ANI0 to ANI21
1	0	1	1	0	ANI22	ANI0 to ANI22
1	0	1	1	1	ANI23	ANI0 to ANI23
Other than above					Setting prohibited[Note]	

Note If a channel that does not have an analog input is selected, the conversion result is undefined.

Remark The number of analog input function pins (ANIn) differs from one product to another. For details, see **13.1 Overview**.

(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. The 10 bits of the conversion result are read from the higher 10 bits of the ADA0CRn register, and 0 is read from the lower 6 bits. The higher 8 bits of the conversion result are read from the ADA0CRnH register.

The ADA0CRn and ADA0CRnH registers are initialized when the ADA0M0.ADA0PS bit = 0.

Caution Accessing the ADA0CRn and ADA0CRnH registers is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- **When the CPU operates with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the low-speed internal oscillation clock**

After reset: 0000H R Address: ADA0CR0 FFFFF210H, ADA0CR1 FFFFF212H, ADA0CR2 FFFFF214H, ADA0CR3 FFFFF216H ADA0CR4 FFFFF218H, ADA0CR5 FFFFF21AH ADA0CR6 FFFFF21CH, ADA0CR7 FFFFF21EH ADA0CR8 FFFFF220H, ADA0CR9 FFFFF222H ADA0CR10 FFFFF224H, ADA0CR11 FFFFF226H ADA0CR12 FFFFF228H, ADA0CR13 FFFFF22AH ADA0CR14 FFFFF22CH, ADA0CR15 FFFFF22EH ADA0CR16 FFFFF230H, ADA0CR17 FFFFF232H ADA0CR18 FFFFF234H, ADA0CR19 FFFFF236H ADA0CR20 FFFFF238H, ADA0CR21 FFFFF23AH ADA0CR22 FFFFF23CH, ADA0CR23 FFFFF23EH

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADA0CRn	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	0	0	0	0	0	0

After reset: 00H R Address: ADA0CR0H FFFFF211H, ADA0CR1H FFFFF213H, ADA0CR2H FFFFF215H, ADA0CR3H FFFFF217H ADA0CR4H FFFFF219H, ADA0CR5H FFFFF21BH ADA0CR6H FFFFF21DH, ADA0CR7H FFFFF21FH ADA0CR8H FFFFF221H, ADA0CR9H FFFFF223H ADA0CR10H FFFFF225H, ADA0CR11H FFFFF227H ADA0CR12H FFFFF229H, ADA0CR13H FFFFF22BH ADA0CR14H FFFFF22DH, ADA0CR15H FFFFF22FH ADA0CR16H FFFFF231H, ADA0CR17H FFFFF233H ADA0CR18H FFFFF235H, ADA0CR19H FFFFF237H ADA0CR20H FFFFF239H, ADA0CR21H FFFFF23BH ADA0CR22H FFFFF23DH, ADA0CR23H FFFFF23FH

	7	6	5	4	3	2	1	0
ADA0CRnH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2

Caution A write operation to the ADA0M0 to ADA0M2, ADA0S, ADA0PFM, and ADA0PFT registers may cause the contents of the ADA0CRn register to become undefined. After the conversion, read the conversion result before writing to the ADA0M0 and ADA0S registers. The contents of the ADA0CRm register may also become undefined when an external or timer trigger has been acknowledged. Read the ADA0CRm register after conversion and before the next external or timer trigger is acknowledged. 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,

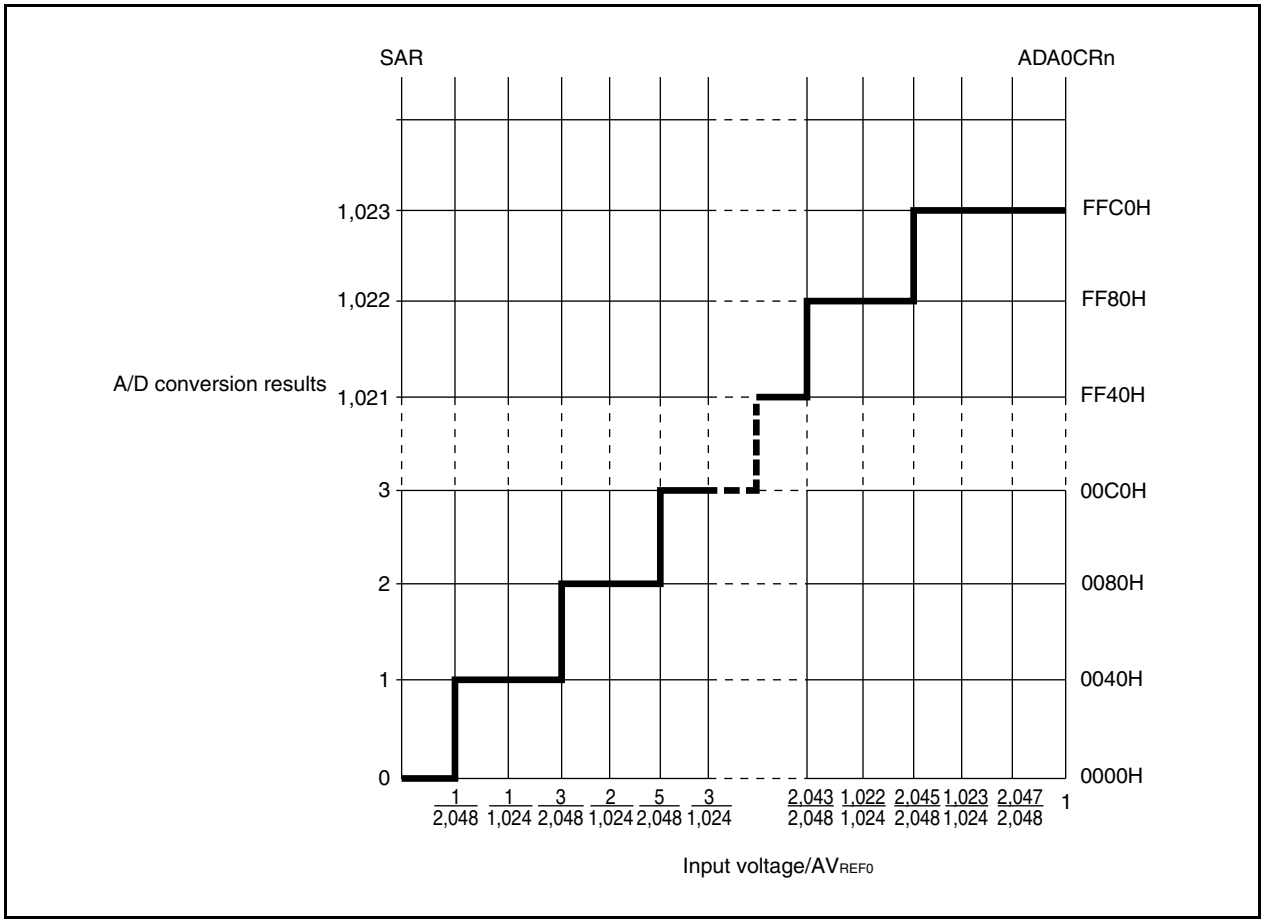
$$(SAR - 0.5) \times \frac{AV_{REF0}}{1{,}024} \leq V_{IN} < (SAR + 0.5) \times \frac{AV_{REF0}}{1{,}024}$$

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	Generates an interrupt request signal (INTAD) when ADA0CRnH ≥ ADA0PFT
1	Generates 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 value of the ADA0CRnH register 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, however, the interrupt signal is not generated.

2. In the scan mode, the 8-bit data set to the ADA0PFT register is compared with the contents of the ADA0CR0H register. 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, however, the INTAD signal is not generated. Regardless of the comparison result, the scan operation is continued and the conversion result is stored in the ADA0CRn register until the scan operation is completed. However, the INTAD signal is not generated after the scan operation has been completed.

(7) Power-fail compare threshold value register (ADA0PFT)

The ADA0PFT register sets the threshold value when comparing with the A/D conversion result register nH (ADA0CRnH).

The 8-bit data set in the ADA0PFT register is compared with the value of the ADA0CRnH register.

The ADA0PFT 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

	7	6	5	4	3	2	1	0
ADA0PFT								

13.5 Operation

13.5.1 Basic operation

<1> Turn on power to the A/D converter by setting the ADA0M0.ADA0PS bit to 1. At this time, make sure that the ADA0M0.ADA0CE bit is 0. The other bits can be set at the same time.

<2> Set the operation mode, trigger mode, and conversion time for executing A/D conversion by using the ADA0M0, ADA0M1, ADA0M2, and ADA0S registers.

<3> When the ADA0CE bit of the ADA0M0 register is set, conversion is started after power to the A/D converter has been turned on (ADA0M0.ADA0PS bit = 0 → 1) and then the stabilization time has elapsed in the software trigger mode. In the external or timer trigger mode, the A/D converter waits for a trigger in the external or timer trigger mode.

<4> When A/D conversion is started, the voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<5> When the sample & hold circuit samples the input channel for a specific time, it enters the hold status, and holds the input analog voltage until A/D conversion is complete.

<6> Set bit 9 of the successive approximation register (SAR) to set the compare voltage generation DAC to (1/2) AV_{REF0}.

<7> The voltage difference between 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. If it is lower than (1/2) AV_{REF0}, the MSB is reset.

<8> 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 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 ≥ Compare voltage: Bit 8 = 1
Analog input voltage ≤ Compare voltage: Bit 8 = 0

<9> This comparison is continued to bit 0 of the SAR register.

<10> When comparison of the 10 bits is complete, the valid digital result is stored in the SAR register, which is then transferred to and stored in the ADA0CRn register. After that, an A/D conversion end interrupt request signal (INTAD) is generated.

<11> Conversion is stopped in the one-shot select mode (at this time, the ADA0M0.ADA0CE bit remains 1 and is not cleared automatically).

In the one-shot scan mode, the A/D converter stops after it has made one round of scanning. In the successive select mode or successive scan mode, the converter repeats steps <4> to <10> until the ADA0M0.ADA0CE bit is cleared to 0.

13.5.2 Trigger mode

The timing of starting the conversion operation is specified by setting a trigger mode. The trigger mode includes a software trigger mode and hardware trigger modes. Four trigger modes are available in the hardware trigger mode: timer trigger modes 0 to 2, 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.

(1) Software trigger mode

When the ADA0M0.ADA0CE bit is set to 1, the signal of the analog input pin (ANIn) 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.

(2) External trigger mode

In this mode, converting the signal of the analog input pin (ANIn) 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 (i.e., the rising edge, falling edge, or both rising and falling edges) can be specified by using the ADA0M0.ADA0ETS1 and ADA0M0.ATA0ETS0 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, 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 the valid trigger is input during the conversion operation, the conversion is aborted and started again from the beginning.

If the ADA0M0, ADA0M2, ADA0S, ADA0PFM, or ADA0PFT register is written during the conversion operation, the conversion is not aborted, and the A/D converter waits for the trigger again.

(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 (INTTAA2CC0, INTTAA2CC1, or TABTADT0) of the capture/compare register connected to the timer. The INTTAA2CC0, INTTAA2CC1, or TABTADT0 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, 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 the valid trigger is input during the conversion operation, 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.

13.5.3 Operation mode

Four operation modes are available as the modes in which to set the ANIn pins: continuous select mode, continuous scan mode, one-shot select mode, and one-shot scan mode.

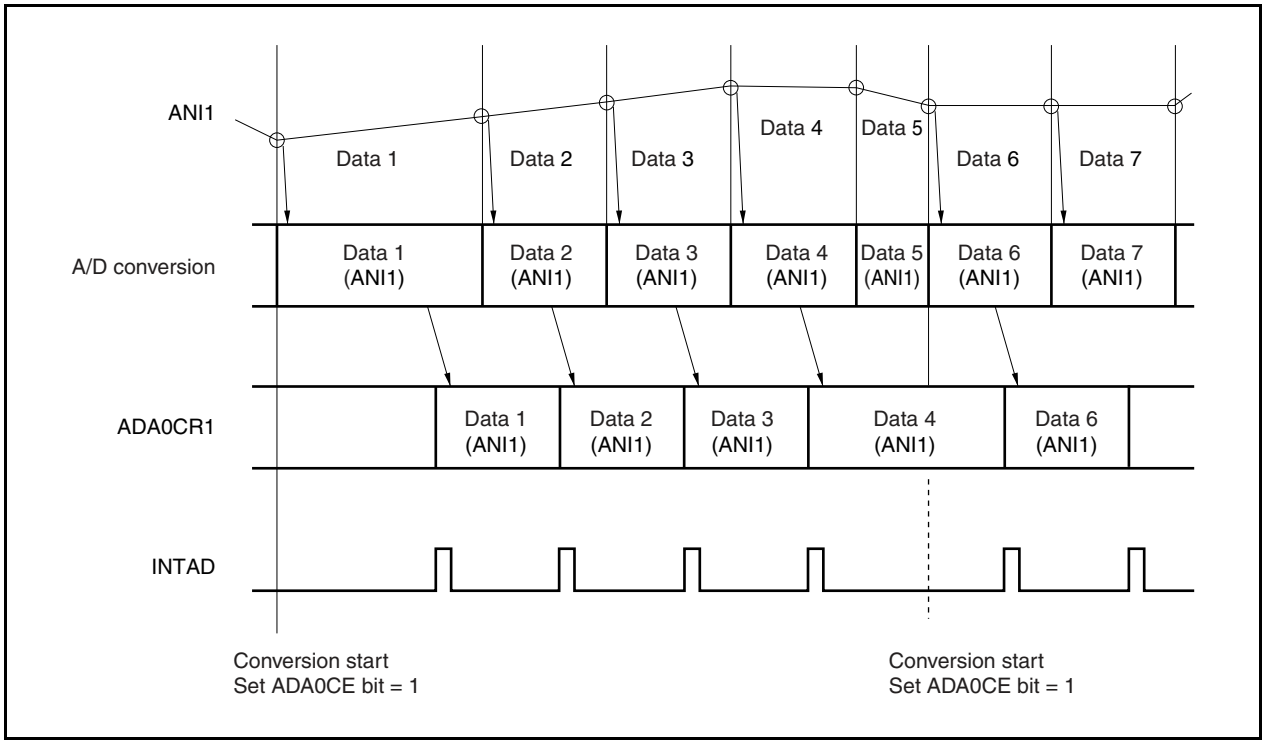
The operation mode is selected by the ADA0M0.ADA0MD1 and ADA0M0.ADA0MD0 bits.

(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. In this mode, an analog input pin corresponds to an ADA0CRn register on a one-to-one basis. 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 13-3. Timing Example of Continuous Select Mode Operation (ADA0S Register = 01H)



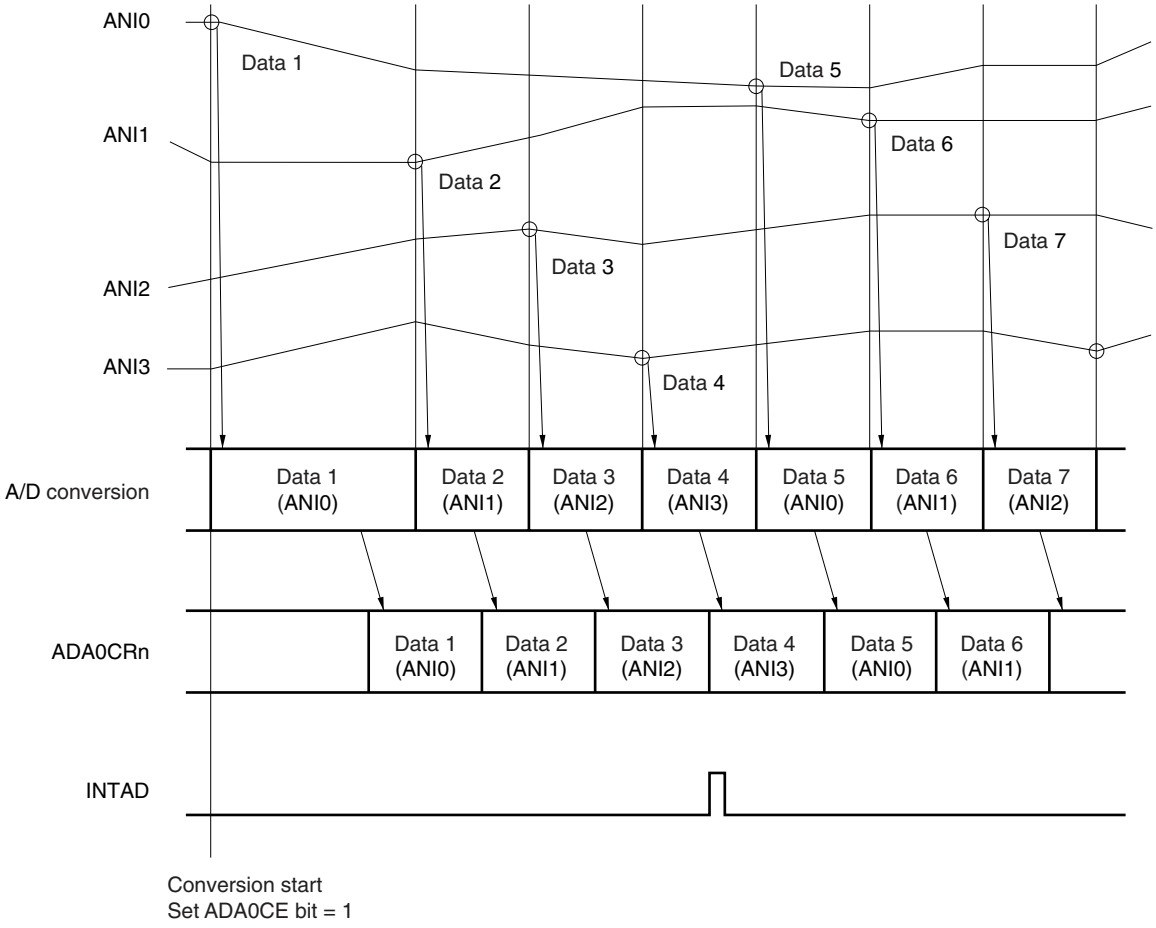
(2) Continuous scan mode

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.

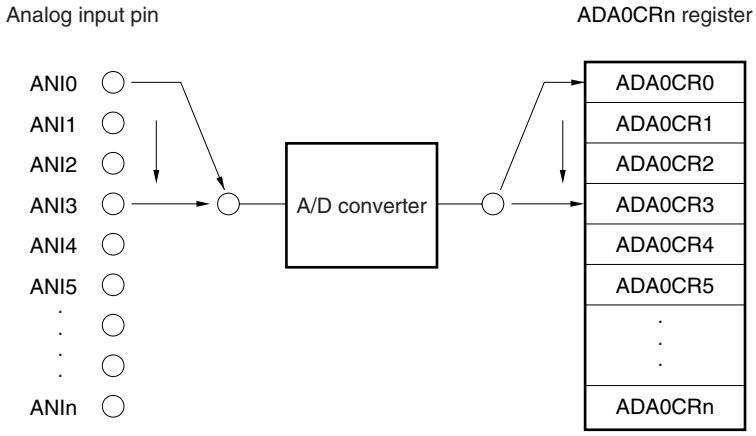
The result of each conversion is stored in the ADA0CRn register corresponding to the analog input pin. When conversion of the analog input pin specified by the ADA0S register is complete, the INTAD signal is generated, and A/D conversion is started again from the ANI0 pin, unless the ADA0CE bit is cleared to 0.

Figure 13-4. Timing Example of Continuous Scan Mode Operation (ADA0S Register = 03H)

(a) Timing example



(b) Block diagram

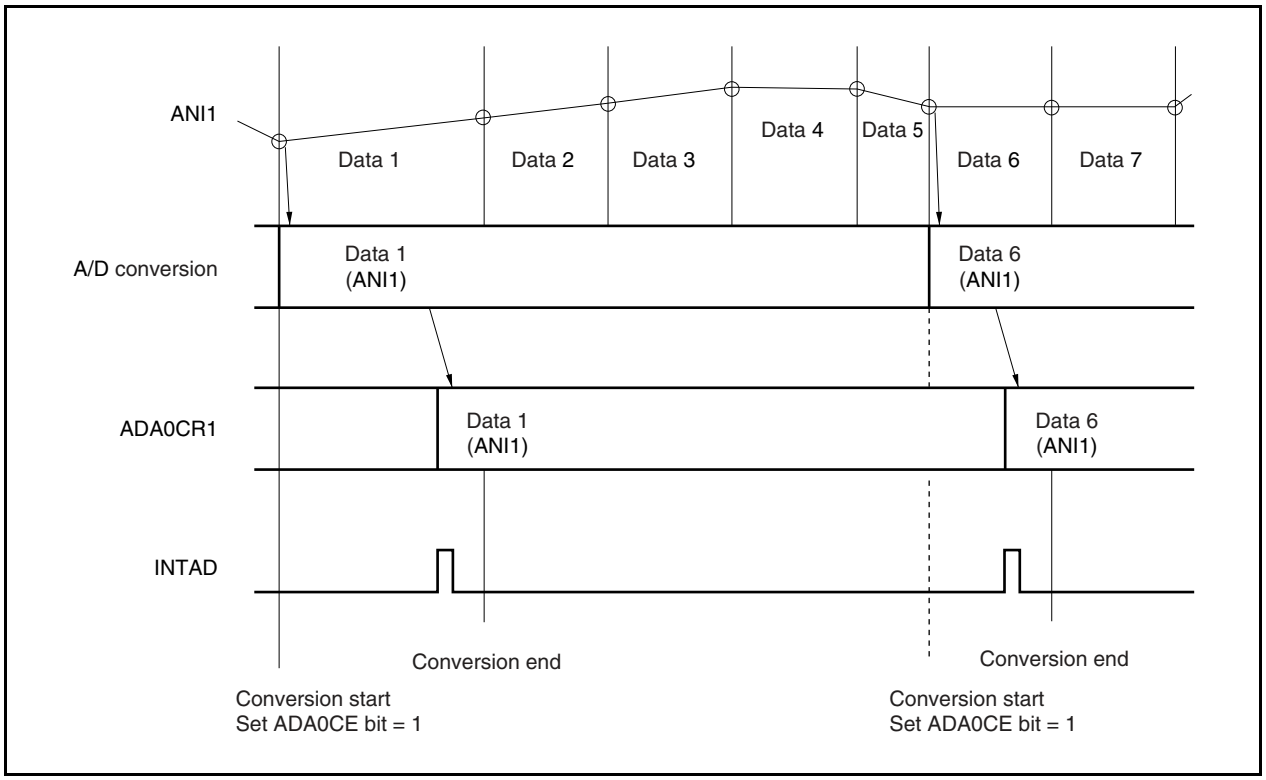


(3) One-shot select mode

In this mode, the voltage on 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. The A/D conversion operation is stopped after it has been completed.

Figure 13-5. Timing Example of One-Shot Select Mode Operation (ADA0S Register = 01H)



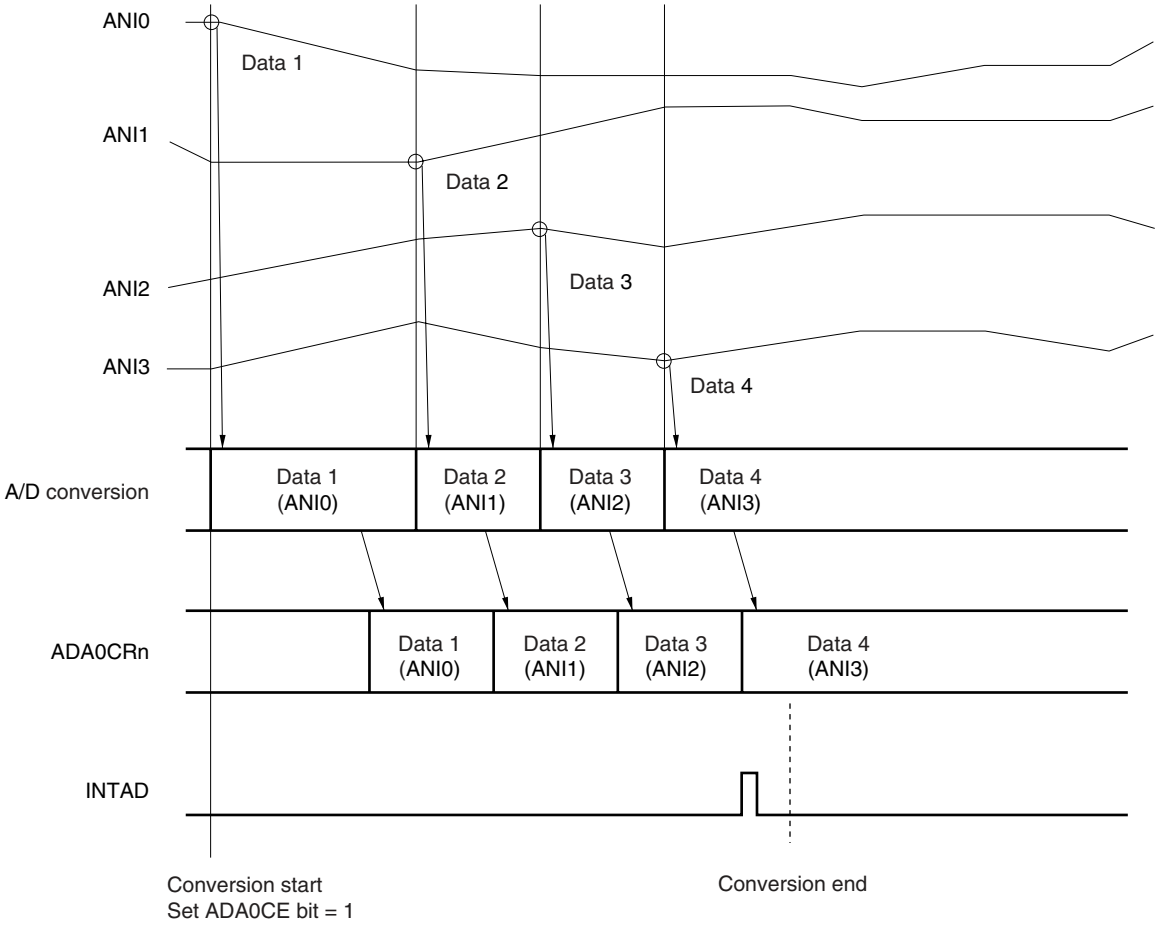
(4) One-shot scan mode

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.

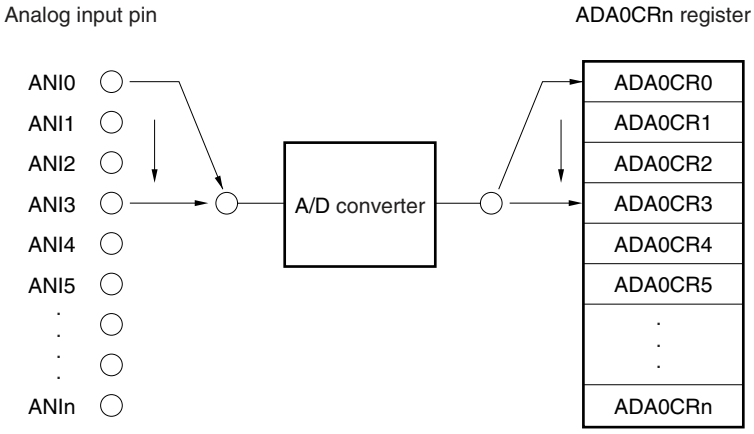
Each conversion result is stored in the ADA0CRn register corresponding to the analog input pin. When conversion of the analog input pin specified by the ADA0S register is complete, the INTAD signal is generated. A/D conversion is stopped after it has been completed.

Figure 13-6. Timing Example of One-Shot Scan Mode Operation (ADA0S Register = 03H)

(a) Timing example



(b) Block diagram



13.5.4 Power-fail compare mode

The A/D conversion end interrupt request signal (INTAD) can be controlled as follows by the ADA0PFM and ADA0PFT registers.

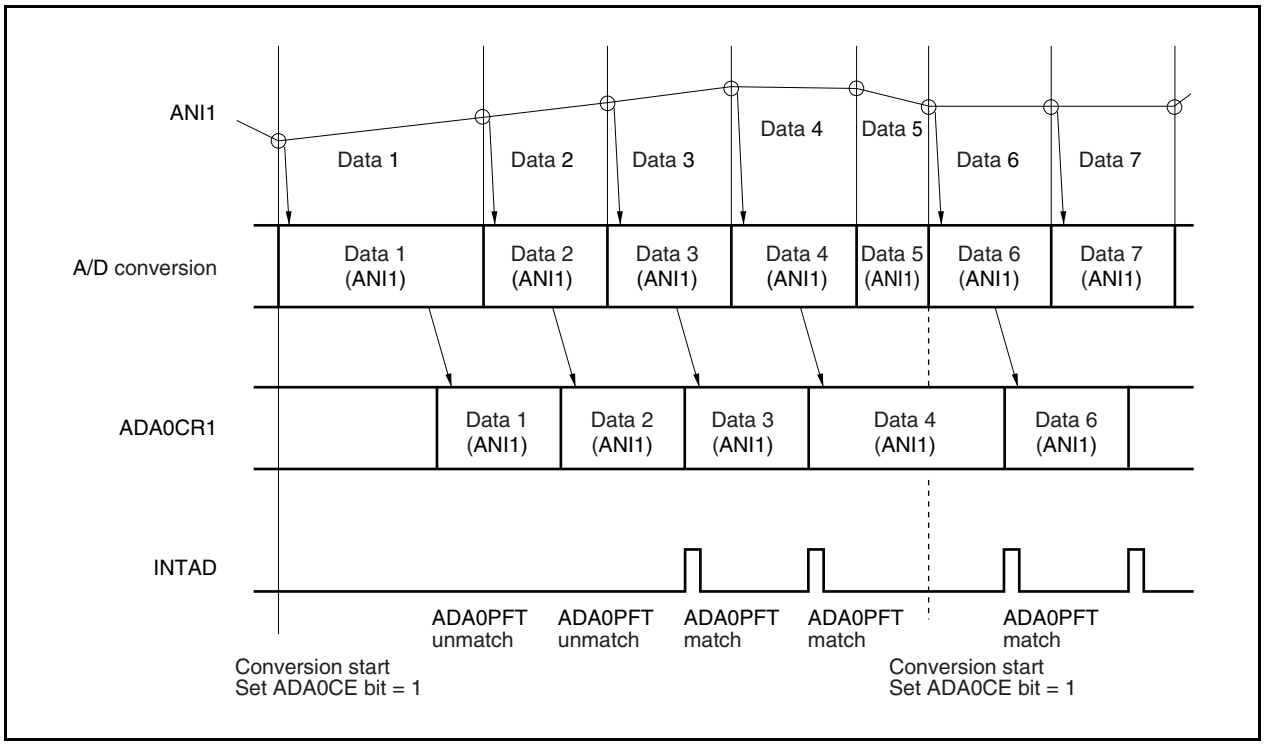
- When the ADA0PFM.ADA0PFE bit = 0, the INTAD signal is generated each time conversion is completed (normal use of the A/D converter).
- When the ADA0PFE bit = 1 and when the ADA0PFM.ADA0PFC bit = 0, the value of the ADA0CRnH register is compared with the value of the ADA0PFT register when conversion is completed, and the INTAD signal is generated only if ADA0CRnH ≥ ADA0PFT.
- When the ADA0PFE bit = 1 and when the ADA0PFC bit = 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 as modes in which to set the ANIn pins: 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 13-7. Timing Example of Continuous Select Mode Operation (When Power-Fail Comparison Is Made: ADA0S Register = 01H)



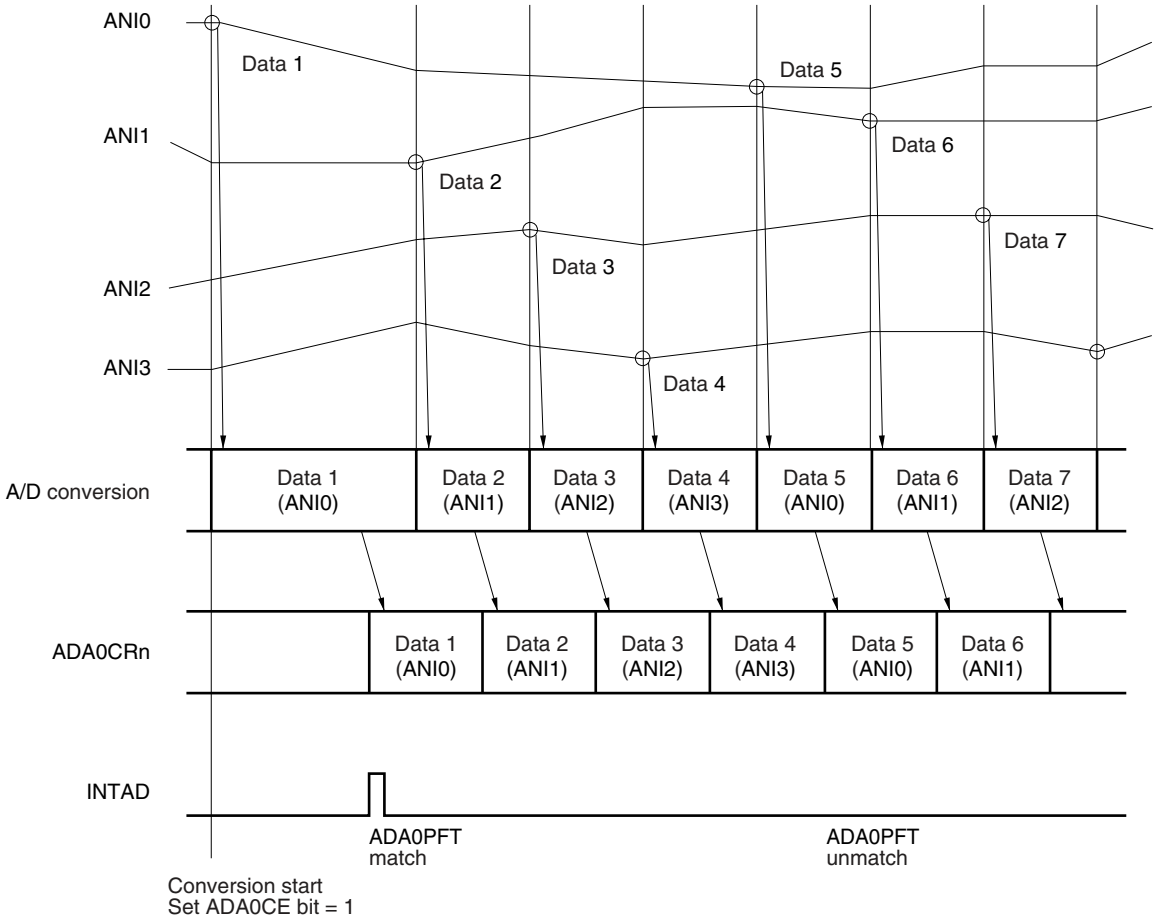
(2) Continuous scan mode

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 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 INTAD signal is not generated.

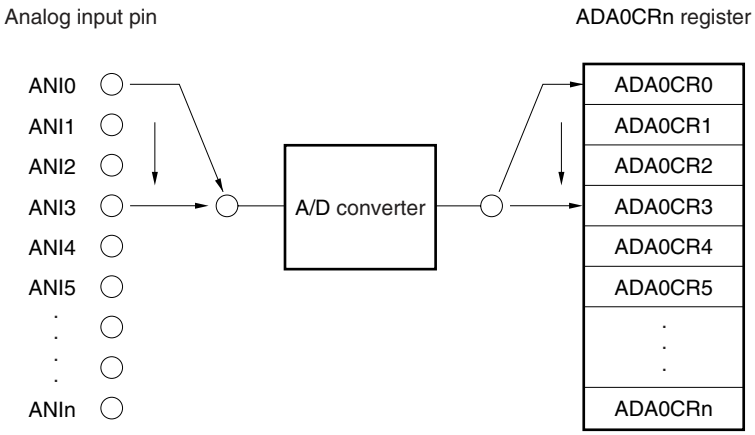
After the result of the first conversion has been stored in the ADA0CR0 register, the results of sequentially converting the voltages on the analog input pins up to the pin specified by the ADA0S register are continuously stored. After completion of conversion, the next conversion is started from the ANI0 pin again, unless the ADA0CE bit is cleared to 0.

Figure 13-8. Timing Example of Continuous Scan Mode Operation (When Power-Fail Comparison Is Made: ADA0S Register = 03H)

(a) Timing example



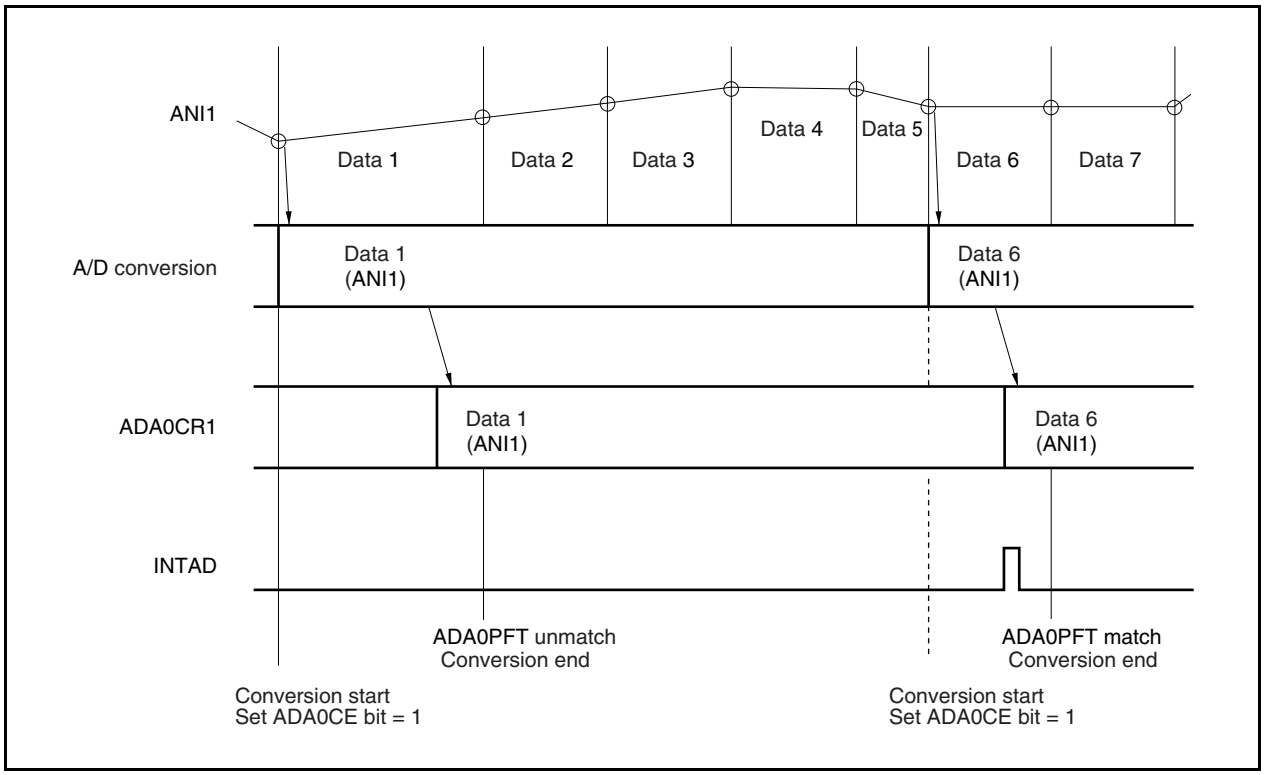
(b) Block diagram



(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 13-9. Timing Example of One-Shot Select Mode Operation (When Power-Fail Comparison Is Made: ADA0S Register = 01H)

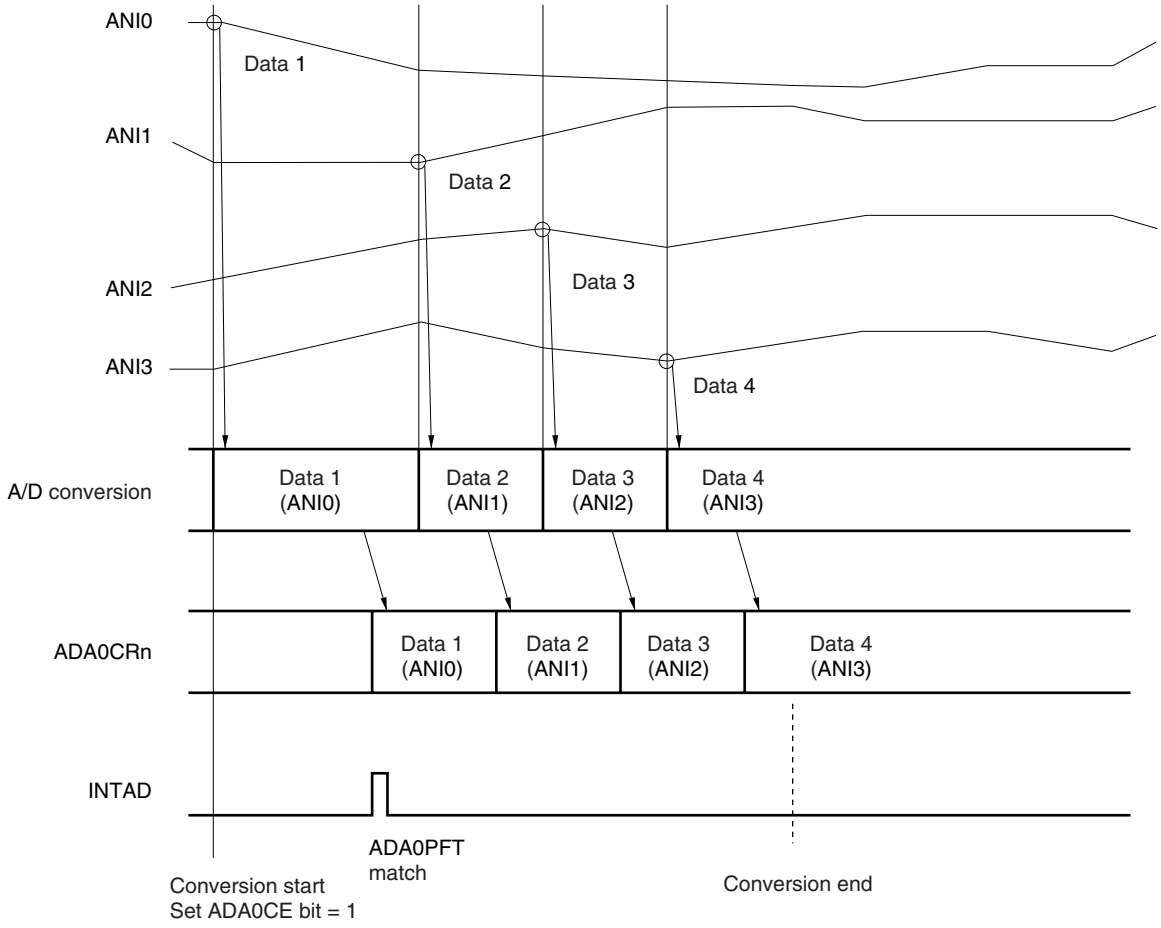


(4) One-shot scan mode

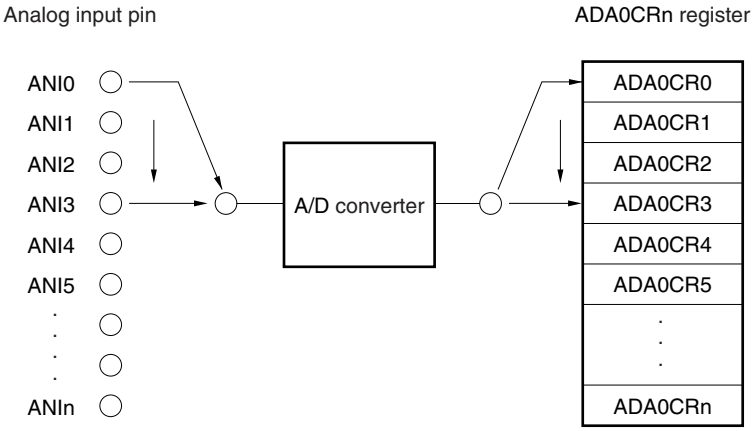
In this mode, the results of converting the voltages of the analog input pins sequentially selected from the ANI0 pin to the pin specified by the ADA0S register are stored, and the set value of the ADA0CR0H register of channel 0 is compared with the set value of the ADA0PFT register. If the result of 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. The conversion is stopped after it has been completed.

Figure 13-10. Timing Example of One-Shot Scan Mode Operation (When Power-Fail Comparison Is Made: ADA0S Register = 03H)

(a) Timing example



(b) Block diagram



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 and ADA0M0.ADA0PS bits to 0.

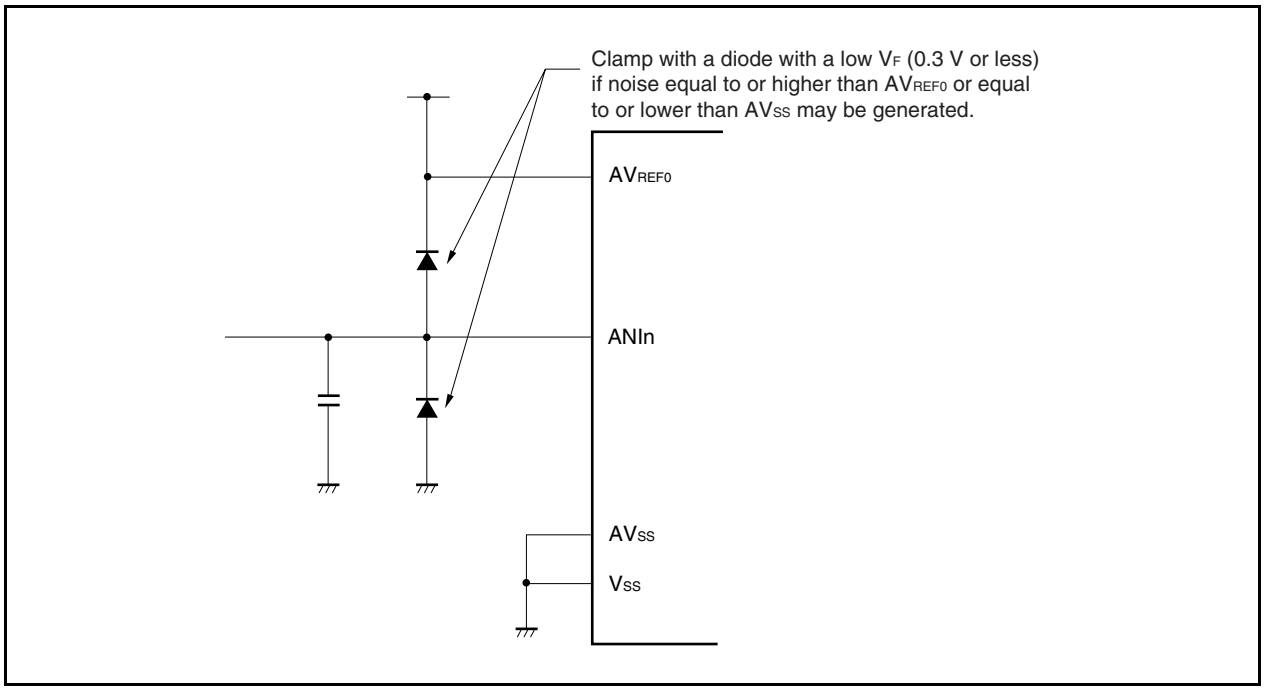
(2) Input range of ANIn pins

Input the voltage within the specified range 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 13-11 is recommended.

Figure 13-11. Processing 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.

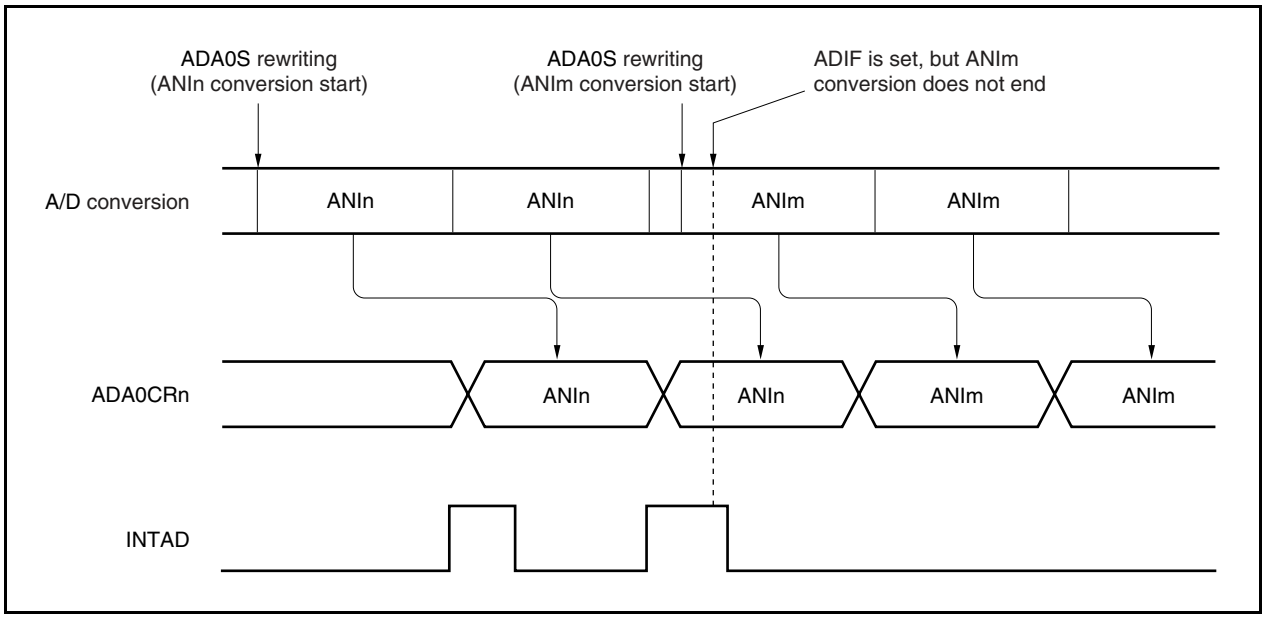
Also the conversion resolution may drop at the pins set as output port pins during A/D conversion if the output current fluctuates due to the influence of the external circuit connected to the port pins.

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 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 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 even though the A/D conversion of the newly selected analog input pin has not been completed. When A/D conversion is stopped, clear the ADIF flag before resuming conversion.

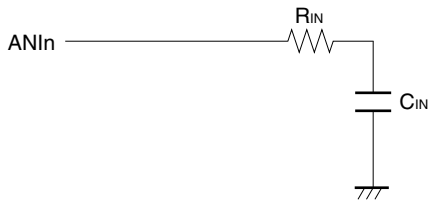
Figure 13-12. Generation Timing of A/D Conversion End Interrupt Request



(6) Internal equivalent circuit

The following shows the equivalent circuit of the analog input block.

Figure 13-13. Internal Equivalent Circuit of ANIn Pin



R_{IN}	C_{IN}
2.6 kΩ	6.2 pF

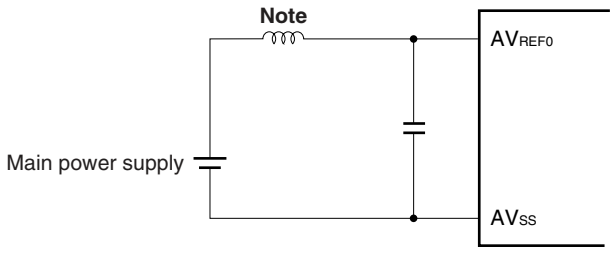
(7) AV_{REF0} pin

(a) The AV_{REF0} 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, therefore, be sure to supply the voltage defined in electrical specifications to the AV_{REF0} pin as shown in Figure 13-14.

(b) The AV_{REF0} pin is also used as the reference voltage pin of the A/D converter. If the source supplying power to the AV_{REF0} pin has a high impedance or if the power supply has a low current supply capability, the reference voltage may fluctuate due to the current that flows during conversion (especially, immediately after the conversion operation enable 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 AV_{REF0} and AV_{SS} pins to suppress the reference voltage fluctuation as shown in Figure 13-14.

(c) If the source supplying power to the AV_{REF0} pin has a high DC resistance (for example, because of insertion of a diode), the voltage when conversion is enabled may be lower than the voltage when conversion is stopped, because of a voltage drop caused by the A/D conversion current.

Figure 13-14. AV_{REF0} Pin Processing Example



Note Parasitic inductance

(8) Reading ADA0CRn result

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) Variation of A/D conversion results

If there is noise at the analog input pins and at the reference voltage input pins, that noise may generate an illegal conversion result. Software processing will be needed to avoid a negative effect on the system from this illegal conversion result. An example of this software processing is shown below.

- Take the average result of a number of A/D conversions and use that as the A/D conversion result.
- Execute a number of A/D conversions consecutively and use those results, omitting any exceptional results that may have been obtained.
- If an A/D conversion result that is judged to have generated a system malfunction is obtained, be sure to recheck the system malfunction before performing malfunction processing.

(10) Standby mode

Because the A/D converter stops operating in the STOP mode, conversion results are invalid, so power consumption can be reduced. 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 and ADA0M0.ADA0PS bits to 0, and then set the ADA0M0.ADA0PS bit to 0 and the ADA0M0.ADA0CE bit to 1 after releasing the STOP mode.

In the IDLE1, IDLE2, or subclock operation mode, operation continues. To lower the power consumption, therefore, clear the ADA0M0.ADA0CE and ADA0M0.ADA0PS bits to 0. 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.

(11) 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 the A/D conversion has finished, the analog input voltage remains in the internal sample & hold capacitor. As a result, the following phenomena may occur.

- When the same channel is used for A/D conversions, if the voltage is higher or lower than the previous A/D conversion, then hysteresis characteristics may appear where the conversion result is affected by the previous value.
- 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.

Therefore, to obtain 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, i.e., 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.

1%FSR = (Maximum value of convertible analog input voltage – Minimum value of convertible analog input voltage)/100
= (AV_{REF0} – 0)/100
= AV_{REF0}/100

When the resolution is 10 bits, 1 LSB is as follows:

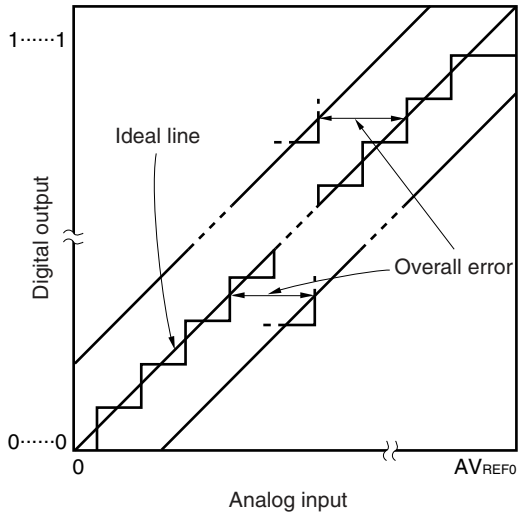
1 LSB = $1/2^{10}$ = 1/1,024
= 0.098%FSR

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 13-15. Overall Error

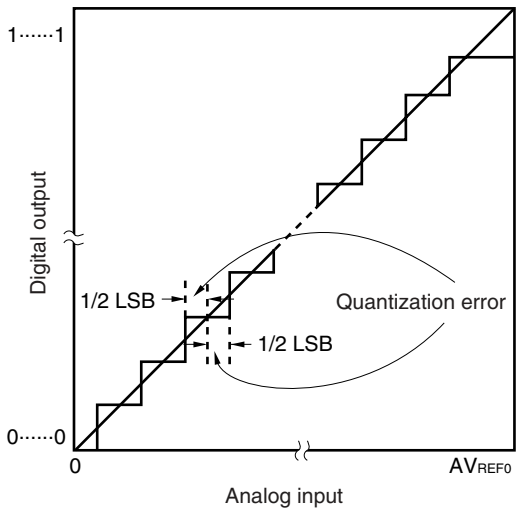


(3) Quantization error

This is an error of ±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 ±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.

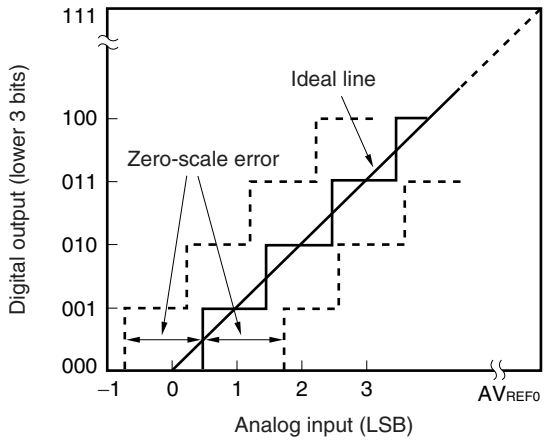
Figure 13-16. Quantization Error



(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).

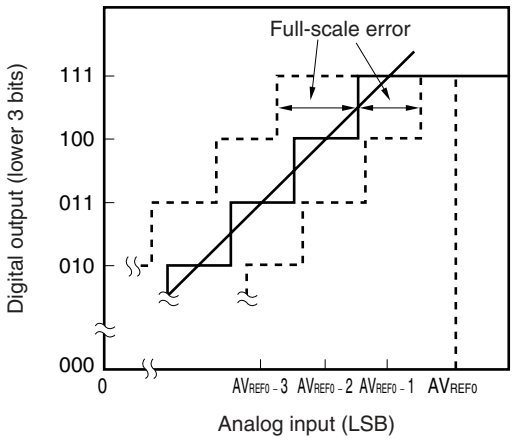
Figure 13-17. Zero-Scale Error



(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).

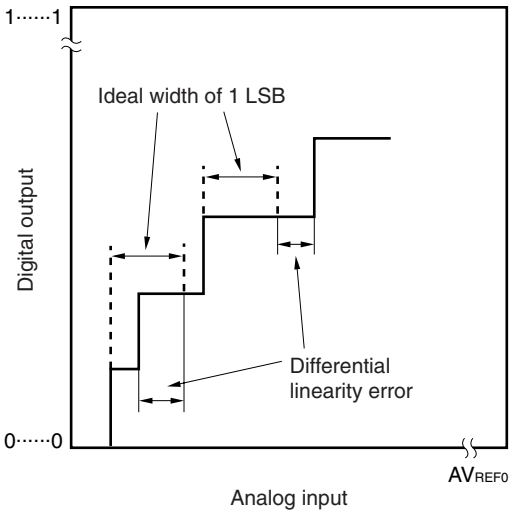
Figure 13-18. Full-Scale Error



(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**.

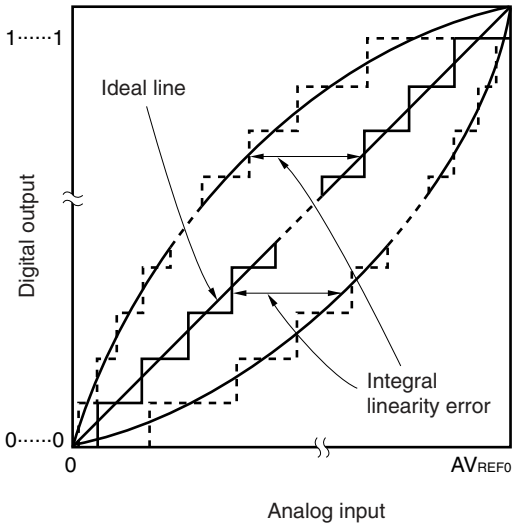
Figure 13-19. Differential Linearity Error



(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 13-20. Integral Linearity Error



(8) Conversion time

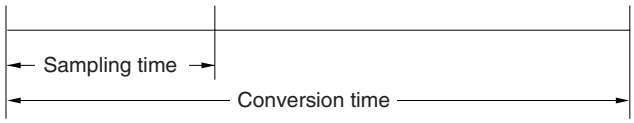
This is the time required to obtain a digital output after each trigger has been generated.

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 13-21. Sampling Time



CHAPTER 14 ASYNCHRONOUS SERIAL INTERFACE D (UARTD)

The V850ES/Hx3 includes asynchronous serial interface D (UARTD). The number of channels differs depending on the product. Table 14-1 shows the number of channels of each product.

Table 14-1. Number of Channels of Asynchronous Serial Interface D

Part Number	V850ES/HE3	V850ES/HF3	V850ES/HG3	V850ES/HJ3	
				μPD70F3755	μPD70F3757
Number of Channels	2 channels UARTD0, UARTD1	2 channels UARTD0, UARTD1	3 channels UARTD0 to UARTD2	3 channels UARTD0 to UARTD2	6 channels UARTD0 to UARTD5

In this chapter, the number of channels is expressed as n.

14.1 Features

○ Transfer rate: 300 bps to 1.5 Mbps (using internal system clock of 32 MHz and dedicated baud rate generator)
○ Full-duplex communication: Internal UARTDn receive data register (UDnRX)
Internal UARTDn transmit data register (UDnTX)
○ 2-pin configuration: TXDDn: Transmit data output pin
RXDDn: Receive data input pin
○ Reception error detection function
- Parity error
- Framing error
- Overrun error
- LIN communication data consistency error detection function
- SBF reception success detection function

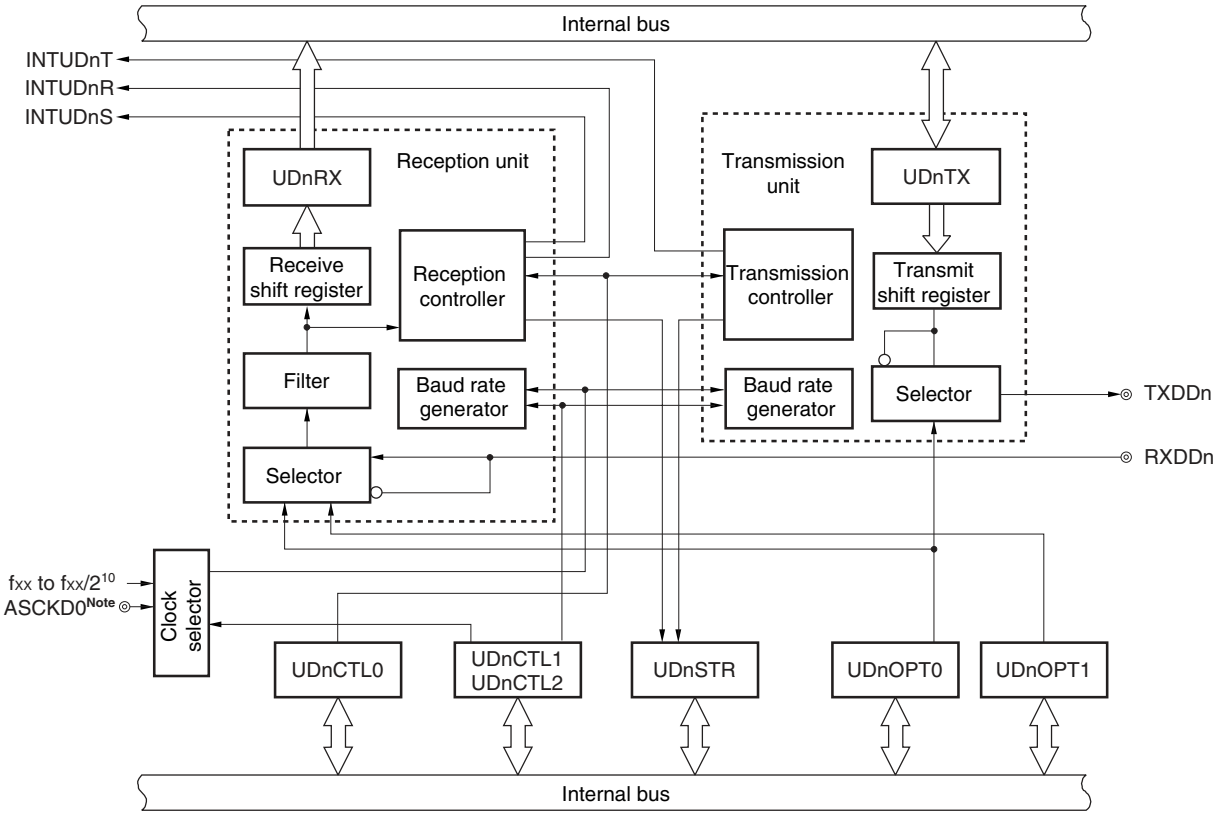
○ Interrupt sources: 3
- Reception complete interrupt (INTUDnR): This interrupt occurs upon transfer of receive data from the receive shift register to the receive data register after completion of serial transfer in the reception enabled status.
- Transmission enable interrupt (INTUDnT): This interrupt occurs upon transfer of transmit data from the transmit data register to the transmit shift register in the transmission enabled status.
- Status interrupt (INTUDnS): This interrupt occurs upon detection of a reception error, a LIN communication data consistency error, or SBF reception success.

○ Character length: 7, 8 bits
○ Parity function: Odd, even, 0, none
○ Transmission stop bit: 1, 2 bits
○ On-chip dedicated baud rate generator
○ MSB-/LSB-first transfer selectable
○ Transmit/receive data inverted input/output possible
○ SBF (Sync Break Field) transmission/reception in the LIN (Local Interconnect Network) communication format possible
- 13 to 20 bits selectable for SBF transmission
- Recognition of 11 bits or more possible for SBF reception in the LIN communication format
- SBF reception flag provided
- New SBF reception can be detected during data communication
- Function to check consistency of transmit data (the function to compare transmit data and receive data, and detect mismatch) is available

14.2 Configuration

The block diagram of the UARTDn is shown below.

Figure 14-1. Block Diagram of Asynchronous Serial Interface Dn



Note UARTD0 only

Remark For the configuration of the baud rate generator, see **Figure 14-32**.

UARTDn includes the following hardware units.

Table 14-2. Configuration of UARTDn

Item	Configuration
Registers	UARTDn control register 0 (UDnCTL0) UARTDn control register 1 (UDnCTL1) UARTDn control register 2 (UDnCTL2) UARTDn option control register 0 (UDnOPT0) UARTDn option control register 1 (UDnOPT1) UARTDn status register (UDnSTR) UARTDn receive shift register UARTDn receive data register (UDnRX) UARTDn transmit shift register UARTDn transmit data register (UDnTX)

(1) UARTDn control register 0 (UDnCTL0)

The UDnCTL0 register is an 8-bit register used to specify the UARTDn operation.

(2) UARTDn control register 1 (UDnCTL1)

The UDnCTL1 register is an 8-bit register used to select the input clock for the UARTDn.

(3) UARTDn control register 2 (UDnCTL2)

The UDnCTL2 register is an 8-bit register used to control the baud rate for the UARTDn.

(4) UARTDn option control register 0 (UDnOPT0)

The UDnOPT0 register is an 8-bit register used to control serial transfer for the UARTDn.

(5) UARTDn option control register 1 (UDnOPT1)

The UDnOPT1 register is an 8-bit register used to control serial transfer for the UARTDn.

(6) UARTDn status register (UDnSTR)

The UDnSTRn register consists of flags indicating the error contents when a reception error occurs. Each one of the reception error flags is set (1) upon occurrence of a reception error and is reset (0) by reading the UDnSTR register.

(7) UARTDn receive shift register

This is a shift register used to convert the serial data input to the RXDDn pin into parallel data. Upon reception of 1 byte of data and detection of the stop bit, the receive data is transferred to the UDnRX register.

This register cannot be manipulated directly.

(8) UARTDn receive data register (UDnRX)

The UDnRX register is an 8-bit buffer register that holds receive data. When 7 characters are received, 0 is stored in the highest bit (when data is received LSB first).

In the reception enabled status, receive data is transferred from the UARTDn receive shift register to the UDnRX register in synchronization with the completion of shift-in processing of 1 frame.

Transfer to the UDnRX register also causes the reception complete interrupt request signal (INTUDnR) to be output.

(9) UARTDn transmit shift register

The transmit shift register is a shift register used to convert the parallel data transferred from the UDnTX register into serial data.

When 1 byte of data is transferred from the UDnTX register, the shift register data is output from the TXDDn pin.

This register cannot be manipulated directly.

(10) UARTDn transmit data register (UDnTX)

The UDnTX register is an 8-bit transmit data buffer. Transmission starts when transmit data is written to the UDnTX register. When data can be written to the UDnTX register (when data of one frame is transferred from the UDnTX register to the UARTDn transmit shift register), the transmission enable interrupt request signal (INTUDnT) is generated.

14.3 Assignment of Serial Interface (μPD70F3757 Only)

The μPD70F3757 of the V850ES/Hx3 is assigned RXDD3 and TXDD3 function pins for UARTD3 at two places, as shown below.

Table 14-3. Assignment of Function Pins of UARTD3

Function Name	Pin No.	Alternate Function Pin
RXDD3	22	P40/SIB0/KR0/INTP14
	59	P80/INTP14
TXDD3	23	P41/SOB0/KR1
	60	P81

To use UARTD3, do not use pins 22 and 59 or 23 and 60 at the same time.
Use pins 22 and 23, and 59 and 60 as a set.

14.3.1 To use UARTD3 and CSIB0 at the same time

In the μPD70F3757, UARTD3 and CSIB0 share one pin of port 4 and cannot be used at the same time. In this case, use port 8 for UARTD3.

Cautions 1. The operation related to transmission/reception of UARTD3 or CSIB0 is not guaranteed if the mode is changed during transmission or reception. To change the mode, be sure to disable the operation.

2. When using pins 22 and 23 as the RXDD3 and TXDD3 pins of UARTD3, do not set pins 59 and 60 as the RXDD3 and TXDD3 pins.

Figure 14-2. Example of Setting for Using UARTD3 and CSIB0 at the Same Time

(1) Setting of CSIB0

						PMC42	PMC41	PMC40
PMC4	0	0	0	0	0	1	1	1

						PFC42	PFC41	PFC40
PFC4	0	0	0	0	0	0	0	0

							PFCE41	PFCE40
PFCE4	0	0	0	0	0	0	0	0

(2) Setting of UARTD3

							PMC81	PMC80
PMC8	0	0	0	0	0	0	1	1Note

Note The P80 pin alternately functions as an RXDD3 input function and INTP14 input function. When using as the RXDD3 input function, disable edge detection for the INTP14 pin (set the INTF8.INTF80 bit and INTR8.INTR80 bits to 0).

14.3.2 Mode switching between UARTD5 and I²C00

In the μPD70F3757, UARTD5 and I²C00 function alternately, and their pins cannot be used at the same time. To use UARTD5, the PMC9, PFC9, and PFCE9 registers must be set in advance (see **4.3.9 Port 9**).

Caution The operations related to transmission and reception of UARTD5 or I²C00 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 14-3. Mode Switch Settings of UARTD5 and I²C00

PMC9

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

7	6	5	4	3	2	1	0
PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

PFC9

15	14	13	12	11	10	9	8
PFC915	PFC914	PFC913	PFC912	PFC911	PFC910	PFC99	PFC98

7	6	5	4	3	2	1	0
PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90

PFCE9

15	14	13	12	11	10	9	8
PFCE915	PFCE914	PFCE913	PFCE912	0	0	PFCE99	PFCE98

7	6	5	4	3	2	1	0
PFCE97	PFCE96	PFCE95	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90

PMC915	PFCE915	PFC915	Operation mode
0	×	×	Port I/O mode
1	1	0	SCL00 I/O
1	1	1	TXDD4 output

PMC914	PFCE914	PFC914	Operation mode
0	×	×	Port I/O mode
1	1	0	SDA00 I/O
1	1	1	RXDD4 output

14.3.3 Mode switching between UARTD5 and CSIB2

In the μPD70F3757, UARTD5 and CSIB2 function alternately, and their pins cannot be used at the same time. To use UARTD5, the PMC9, PFC9, and PFCE9 registers must be set in advance (see **4.3.9 Port 9**).

Caution The operations related to transmission and reception of UARTD5 or CSIB2 are not guaranteed if the mode is switched during transmission or reception. Be sure to disable the unit that is not used.

Figure 14-4. Mode Switch Settings of UARTD5 and CSIB2

PMC9

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

7	6	5	4	3	2	1	0
PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

PFC9

15	14	13	12	11	10	9	8
PFC915	PFC914	PFC913	PFC912	PFC911	PFC910	PFC99	PFC98

7	6	5	4	3	2	1	0
PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90

PFCE9

15	14	13	12	11	10	9	8
PFCE915	PFCE914	PFCE913	PFCE912	0	0	PFCE99	PFCE98

7	6	5	4	3	2	1	0
PFCE97	PFCE96	PFCE95	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90

PMC912	PFCE912	PFC912	Operation mode
0	×	×	Port I/O mode
1	0	1	$\overline{\text{SCKB2}}$ I/O
1	1	1	TXDD5 output

14.4 Registers

(1) UARTDn control register 0 (UDnCTL0)

The UDnCTL0 register is an 8-bit register that controls the UARTDn 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: UD0CTL0 FFFFFA00H, UD1CTL0 FFFFFA10H, UD2CTL0 FFFFFA20H, UD3CTL0 FFFFFA30H, UD4CTL0 FFFFFA40H, UD5CTL0 FFFFFA50H

	<7>	<6>	<5>	<4>	3	2	1	0
UDnCTL0	UDnPWR	UDnTXE	UDnRXE	UDnDIR	UDnPS1	UDnPS0	UDnCL	UDnSL

UDnPWR	UARTDn operation control
0	Disable UARTDn operation (reset UARTDn asynchronously)
1	Enable UARTDn operation

The UARTDn operation is controlled by the UDnPWR bit. The TXDDn pin output is fixed to high level by clearing the UDnPWR bit to 0 (fixed to low level if UDnOPT0.UDnTDL bit = 1).

UDnTXE	Transmission operation enable
0	Disable transmission operation
1	Enable transmission operation

- If the UDnTXE bit is cleared to “0” while the UDnTDL bit of the UDnOPT0 register is “0”, the output of the TXDDn pin is fixed to the high level.
  If the UDnTXE bit is cleared to “0” while the UDnTDL bit of the UDnOPT0 register is “1”, the output of the TXDDn pin is fixed to the low level.
- To start transmission, set the UDnPWR bit to 1 and then set the UDnTXE bit to 1. To stop transmission, clear the UDnTXE bit to 0 and then clear the UDnPWR bit to 0.
- To initialize the transmission unit, clear the UDnTXE bit to 0, wait for two cycles of the base clock, and then set the UDnTXE bit to 1 again. Otherwise, initialization may not be executed (for the base clock, see **14.7 (1) (a) Base clock**).

(2/2)

UDnRXE	Reception operation enable
0	Disable reception operation
1	Enable reception operation

- To start reception, set the UDnPWR bit to 1 and then set the UDnRXE bit to 1. To stop reception, clear the UDnRXE bit to 0 and then clear the UDnPWR bit to 0.
- Reception is enabled after the UDnRXE bit has been set to 1 and two cycles of the basic clock have elapsed. Detection of the rising edge of the RXDD pin is enabled after the UDnRXE bit has been set to 1 and four cycles of the basic clock have elapsed.
- To initialize the reception unit, clear the UDnRXE bit to 0, wait for two cycles of the base clock, and then set the UDnRXE bit to 1 again. Otherwise, initialization may not be executed (for the base clock, see **14.7 (1) (a) Base clock**).

UDnDIR	Transfer direction selection
0	MSB-first transfer
1	LSB-first transfer

- This register can be rewritten only when the UDnPWR bit = 0, the UDnTXE bit = 0, or the UDnRXE bit = 0.
- When transmission and reception are performed in the LIN format, set the UDnDIR bit to 1.

UDnPS1	UDnPS0	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 UDnPWR bit = 0, the UDnTXE bit = 0, or the UDnRXE bit = 0.
- If “Reception with 0 parity” is selected for reception, a parity check is not performed. Therefore, since the UDnPE bit of the UDnSTR register is not set, the status interrupt (INTUDnS) due to a parity error is not generated.
- When transmission and reception are performed in the LIN format, clear the UDnPS1 and UDnPS0 bits to 00.

UDnCL	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 UDnPWR bit = 0 or the UDnTXE bit = the UDnRXE bit = 0.
- When transmission and reception are performed in the LIN format, set the UDnCL bit to 1.

UDnSL	Specification of length of stop bit for transmit data
0	1 bit
1	2 bits

This register can be rewritten only when the UDnPWR bit = 0 or the UDnTXE bit = the UDnRXE bit = 0.

Remark For details of parity, see **14.6.11 Parity types and operations**.

(2) UARTDn control register 1 (UDnCTL1)

For details, see **14.7 (2) UARTDn control register 1 (UDnCTL1)**.

(3) UARTDn control register 2 (UDnCTL2)

For details, see **14.7 (3) UARTDn control register 2 (UDnCTL2)**.

(4) UARTDn option control register 0 (UDnOPT0)

The UDnOPT0 register is an 8-bit register that controls the serial transfer operation of the UARTDn register.

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: UD0OPT0 FFFFFA03H, UD1OPT0 FFFFFA13H, UD2OPT0 FFFFFA23H, UD3OPT0 FFFFFA33H, UD4OPT0 FFFFFA43H, UD5OPT0 FFFFFA53H

	<7>	6	5	4	3	2	1	0
UDnOPT0	UDnSRF	UDnSRT	UDnSTT	UDnSLS2	UDnSLS1	UDnSLS0	UDnTDL	UDnRDL

UDnSRF	SBF reception flag
0	When the UDnCTL0.UDnPWR bit = 0 or UDnCTL0.UDnRXE bit = 0 are set. Also upon normal end of SBF reception.
1	During SBF reception
• SBF (Sync Break Field) reception is judged during LIN communication. • The UDnSRF bit is held at 1 when an SBF reception error occurs, and then SBF reception is started again. • UDnSRF bit is a read-only bit.	

UDnSRT	SBF reception trigger
0	–
1	SBF reception trigger
• This is the SBF reception trigger bit during LIN communication, and when read, “0” is always read. For SBF reception, set the UDnSRT bit to 1 to enable SBF reception. • Set the UDnSRT bit after setting the UDnPWR bit =1 and the UDnRXE bit = 1.	

UDnSTT	SBF transmission trigger
0	–
1	SBF transmission trigger
• This is the SBF transmission trigger bit during LIN communication, and when read, “0” is always read. • Set the UDnSTT bit after setting the UDnPWR bit = 1 and the UDnTXE bit = 1.	

Caution Do not set the UDnSRT and UDnSTT bits (1) during SBF reception (UDnSRF bit = 1).

(2/2)

UDnSLS2	UDnSLS1	UDnSLS0	SBF transmit length selection
1	0	1	13-bit output (reset 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 UDnPWR bit = 0 or when the UDnTXE bit = 0.			

UDnTDL	Transmit data level bit
0	Normal output of transfer data
1	Inverted output of transfer data
• The output level of the TXDDn pin can be inverted using the UDnTDL bit. • This register can be set when the UDnPWR bit = 0 or when the UDnTXE bit = 0.	

UDnRDL	Receive data level bit
0	Normal input of transfer data
1	Inverted input of transfer data
• The input level of the RXDDn pin can be inverted using the UDnRDL bit. • This register can be set when the UDnPWR bit = 0 or the UDnRXE bit = 0.	

(5) UARTDn option control register 1 (UDnOPT1)

The UDnOPT1 register is an 8-bit register that controls the serial transfer operation of the UARTDn register.

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

Reset sets this register to 00H.

After reset: 14H R/W Address: UD0OPT1 FFFFFA05H, UD1OPT1 FFFFFA15H, UD2OPT1 FFFFFA25H, UD3OPT1 FFFFFA35H, UD4OPT1 FFFFFA45H, UD5OPT1 FFFFFA55H

	7	6	5	4	3	2	1	0
UDnOPT1	0	0	0	0	0	0	UDnSRS	UDnDCS

UDnSRS	SBF reception mode select bit
0	New SBF is not detected during data communication (if a low level is detected at the position of the stop bit, it is recognized as a framing error).
1	New SBF is detected during data communication (if a low level is detected at the position of the stop bit, detection of the next high level is waited. If the low-level period is of 11 bits or more, it is recognized as a new SBF).

- Set the UDnSRS bit when using LIN communication. Otherwise, be sure to clear this bit to “0”.
- When setting the UDnSRS bit to 1, the UDnDCS bit must also be set to 1.

UDnDCS	Data consistency check select bit
0	Does not check data consistency.
1	Checks data consistency.

- The UDnDCS bit selects whether data consistency is to be checked when data is transmitted by LIN communication.
- When the UDnDCS bit = 1, transmit data and receive data are compared when data is transmitted by LIN communication. If a mismatch is detected, a data consistency error flag is set and a status interrupt request signal (INTUDnS) is generated.
- To use LIN communication, set the UDnDCS bit. Otherwise, be sure to clear this bit to “0”.
- To set the UDnDCS bit to 1, fix the data bit length to 8 bits and disable appending the parity bit.

(6) UARTDn status register (UDnSTR)

The UDnSTR register is an 8-bit register that displays the UARTDn transfer status and reception error contents. This register can be read or written in 8-bit or 1-bit units, but the UDnTSF bit is a read-only bit, while the UDnPE, UDnFE, and UDnOVE bits can be read and written. However, these bits can only be cleared by writing 0; they cannot be set by writing 1 (even if 1 is written to them, the value is retained).

The initialization conditions are shown below.

Register/Bit	Initialization Conditions
UDnSTR register	• Reset • UDnCTL0.UDnPWR bit = 0
UDnSSF bit	• UDnRXE bit of UDnCTL0 register = 0 • UDnSRS bit of UDnOPT1 register = 0
UDnDCE bit	• UDnDCS bit of UDnOPT1 register = 0 • UDnTXE bit of UDnCTL0 register = 0
UDnTSF bit	• UDnCTL0.UDnTXE bit = 0
UDnPE, UDnFE, UDnOVE bits	• 0 write • UDnCTL0.UDnRXE bit = 0

Caution To clear the status flag, use a 1-bit manipulation instruction, or write the inverted value of a read value by using an 8-bit manipulation instruction and clear all the bits that have been set when read at once.

(1/2)

After reset: 00H R/W Address: UD0STR FFFFFA04H, UD1STR FFFFFA14H, UD2STR FFFFFA24H, UD3STR FFFFFA34H, UD4STR FFFFFA44H, UD5STR FFFFFA54H

	<7>	6	5	4	3	<2>	<1>	<0>
UDnSTR	UDnTSF	0	0	UDnSSF	UDnDCE	UDnPE	UDnFE	UDnOVE

UDnTSF	Transfer status flag
0	• When the UDnPWR bit = 0 or the UDnTXE bit = 0 is set. • When, following transfer completion, there is no next data transfer from UDnTX register • When the UDnTX bit does not have the next transmit data after SBF transmission is completed.
1	• Write to UDnTX register • When the SBF transmission trigger bit (UDnSST) is set.
The UDnTSF bit is always 1 when performing continuous transmission. When initializing the transmission unit, check that the UDnTSF bit = 0 before performing initialization. The transmit data is not guaranteed when initialization is performed while the UDnTSF bit = 1.	

UDnSSF	SBF reception success flag
0	When the UDnPWR bit = 0, the UDnTXE bit = 0, the UDnSRS bit = 0, or the UDnSSF bit = 0 is set.
1	When a low level of 11 consecutive bits or more (SBF) is received
• When the LIN communication mode is set by the SBF reception mode select bit (UDnSRS = 1), the start of a new frame slot must be confirmed by reading the UDnSSF bit by status interrupt servicing. • The UDnSSF bit holds the current status until 0 is written to it. When “1” is written to this bit, it holds the current status.	

UDnDCE	Data consistency error flag
0	When the UDnPWR bit = 0, the UDnTXE bit = 0, the UDnDCS bit = 0, or the UDnDCE bit = 0 is set.
1	When transmit data does not match receive data in the LIN communication mode
• When the data consistency check select bit is set (UDnDCS = 1), transmit data and receive data are compared when data is transmitted. If a mismatch is detected, the UDnDCE bit is set to “1”. • The UDnDCE bit holds the current status until 0 is written to it. If “1” is written to this bit, it holds the current status.	

(2/2)

UDnPE	Parity error flag
0	• When the UDnPWR bit = 0 or the UDnRXE bit = 0 is set. • When “0” is written
1	When parity of data and parity bit do not match during reception.

- The operation of the UDnPE bit is controlled by the settings of the UDnCTL0.UDnPS1 and UDnCTL0.UDnPS0 bits.
- The UDnPE bit can be read and written, but it can only be cleared by writing “0” to it, and it cannot be set by writing 1 to it. When “1” is written to this bit, the value is retained.

UDnFE	Framing error flag
0	• When the UDnPWR bit = 0 or the UDnRXE bit = 0 is set • When “0” is written
1	When no stop bit is detected during reception

- Only the first bit of the receive data stop bits is checked, regardless of the value of the UDnCTL0.UDnSL bit.
- The UDnFE bit can be read and written, but it can only be cleared by writing “0” to it, and it cannot be set by writing “1” to it. When “1” is written to this bit, the value is retained.

UDnOVE	Overrun error flag
0	• When the UDnPWR bit = 0 or the UDnRXE bit = 0 is set. • When “0” is written
1	When receive data is set to the UDnRX register and the next receive operation is completed before that receive data has been read

- When an overrun error occurs, the data is discarded without the next receive data being written to the receive buffer.
- The UDnOVE bit can be read and written, but it can only be cleared by writing “0” to it, and it cannot be set by writing “1” to it. When “1” is written to this bit, the value is retained.

(7) UARTDn receive data register (UDnRX)

The UDnRX 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 UDnRX register upon completion of reception of 1 byte of data.

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

In addition to reset input, the UDnRX register can be set to FFH by clearing the UDnCTL0.UDnPWR bit to 0.

After reset: FFH R Address: UD0RX FFFFFA06H, UD1RX FFFFFA16H, UD2RX FFFFFA26H, UD3RX FFFFFA36H, UD4RX FFFFFA46H, UD5RX FFFFFA56H

	7	6	5	4	3	2	1	0
UDnRX								

When the data character length has been specified as 7 bits (UDnCL = 0):

- During LSB-first reception, the receive data is transferred to bits 6 to 0 of the UDnRX register and the MSB always becomes 0.
- During MSB-first reception, the receive data is transferred to bits 7 to 1 of the UDnRX register and the LSB always becomes 0.
- When an overrun error occurs (UDnOVE = 1), the receive data at this time is not transferred to the UDnRX register and is discarded.

(8) UARTDn transmit data register (UDnTX)

The UDnTX register is an 8-bit register used to set transmit data.

Data can be continuously transmitted by writing the next transmit data before transmission is completed and after the transmission interrupt request signal (INTUDnT) is generated.

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

Reset sets this register to FFH.

After reset: FFH R/W Address: UD0TX FFFFFA07H, UD1TX FFFFFA17H, UD2TX FFFFFA27H, UD3TX FFFFFA37H, UD4TX FFFFFA47H, UD5TX FFFFFA57H

	7	6	5	4	3	2	1	0
UDnTX								

When the data length has been specified as 7 bits (UDnCL = 0):

- During LSB-first transmission, the transmit data is transferred to bits 6 to 0 of the UDnTX register.
- During MSB-first transmission, the transmit data is transferred to bits 7 to 1 of the UDnTX register.

Caution Writing to the UDnTX register triggers the start of transmission when transmission is enabled (UDnPWR = 1 and UDnTXE = 1). If a value which is the same as that written previously is written to the UDnTX register, therefore, the same data is transmitted two times.

Be sure to write transmit data during transmission after the transmission interrupt request (INTUDnT) has been generated. Writing to the UDnTX register does not trigger the start of transmission if transmission is disabled. Consequently, transmission is not started even if transmission is enabled after the UDnTX register has been written while transmission is disabled.

14.5 Interrupt Request Signals

The following three interrupt request signals are generated from UARTDn.

- Reception complete interrupt request signal (INTUDnR)
- Transmission enable interrupt request signal (INTUDnT)
- Status interrupt request signal (INTUDnS)

The default priority for these three interrupt request signals is shown below.

Table 14-4. Generated Interrupts and Their Default Priorities

Interrupt	Priority
Status	High
Reception complete	|
Transmission enable	Low

(1) Status interrupt request signal (INTUDnS)

A status interrupt request signal is generated if an error condition is detected during reception. A flag corresponding to the detected error (UDnPE, UDnFE, or UDnOVE bit) is set in the UDnSTR register.

- When the LIN communication mode is selected by the SBF reception mode select bit (UDnSRS bit = 1), the status interrupt request signal is generated if a low level of 11 consecutive bits or more (SBF) is received.
- When the data consistency check select bit is set (UDnDCS bit = 1), transmit data and receive data are compared when data is transmitted. If a mismatch is detected, the status interrupt request is generated.

(2) Reception complete interrupt request signal (INTUDnR)

A reception complete interrupt request signal is output when data is shifted into the receive shift register and transferred to the UDnRX register in the reception enabled status.

No reception complete interrupt request signal (INTUDnR) is output, and a status interrupt request signal (INTUDnS) is output when a reception error occurs.

No reception complete interrupt request signal is generated in the reception disabled status.

(3) Transmission enable interrupt request signal (INTUDnT)

If transmit data is transferred from the UDnTX register to the UARTDn transmit shift register with transmission enabled, the transmission enable interrupt request signal is generated.

14.6 Operation

14.6.1 Data format

Full-duplex serial data reception and transmission is performed.

As shown in Figure 14-5, one data frame of transmit/receive data consists of a start bit, character bits, parity bit, and stop bit(s).

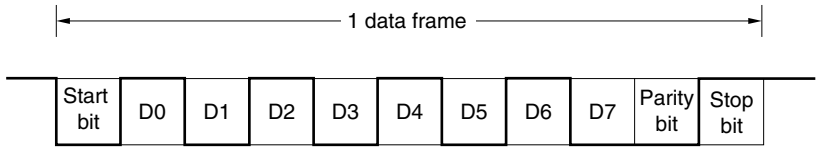
Specification of the character bit length within 1 data frame, parity selection, specification of the stop bit length, and specification of MSB/LSB-first transfer are performed using the UDnCTL0 register.

Moreover, control of UART output/inverted output for the TXDDn bit is performed using the UDnOPT0.UDnTDL bit.

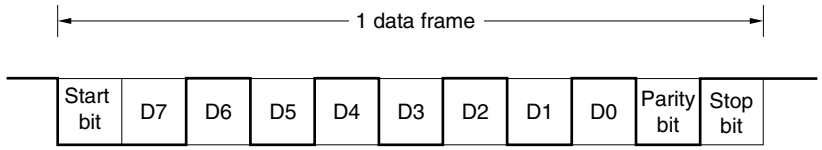
- Start bit ..... 1 bit
- Character bits ..... 7 bits/8 bits
- Parity bit ..... Even parity/odd parity/0 parity/no parity
- Stop bit ..... 1 bit/2 bits
- Output logic ..... Forward/reverse
- Communication direction ..... LSB/MSB

Figure 14-5. UARTD Transmit/Receive Data Format

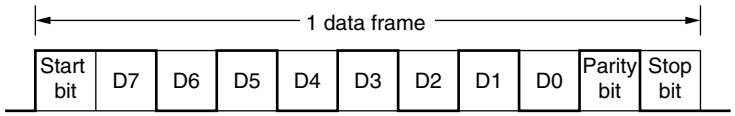
(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55H



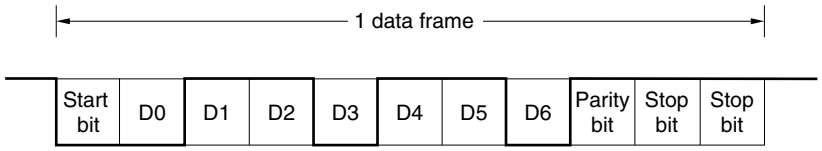
(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H



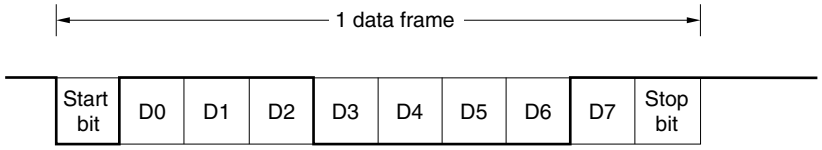
(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55H, TXDDn inversion



(d) 7-bit data length, LSB first, odd parity, 2 stop bits, transfer data: 36H



(e) 8-bit data length, LSB first, no parity, 1 stop bit, transfer data: 87H



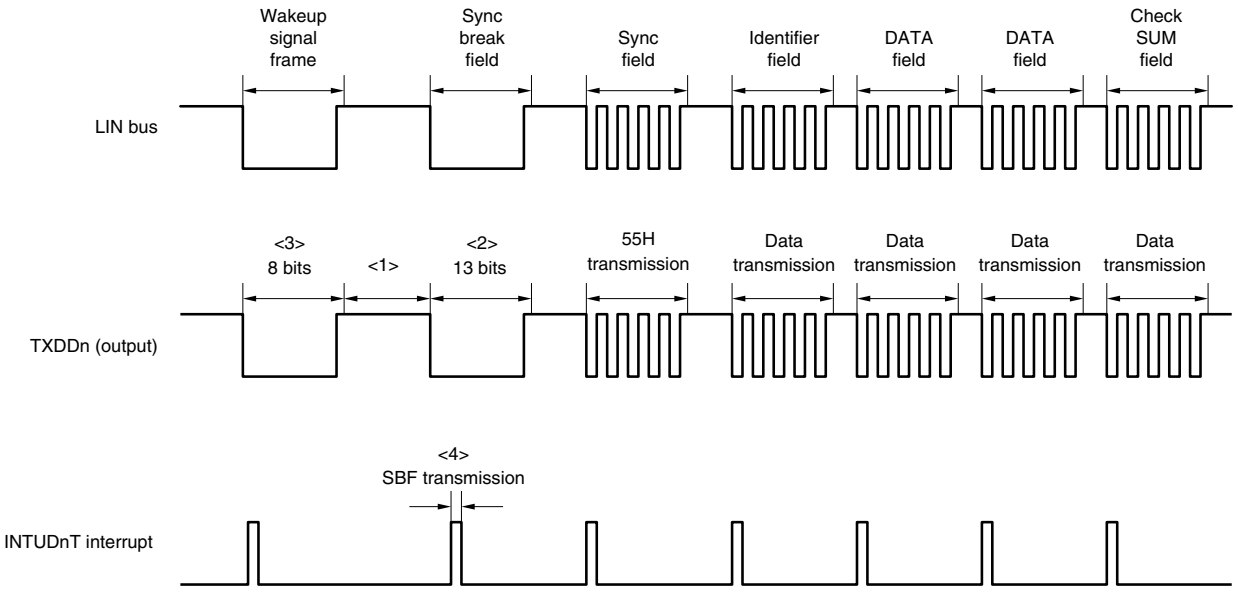
14.6.2 SBF transmission/reception format

The V850ES/Hx3 has 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 aid the cost reduction of an automotive network.
LIN communication is single-master communication, and up to 15 slaves can be connected to one 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 ±15% or less.

Figures 14-6 and 14-7 outline the transmission and reception manipulations of LIN.

Figure 14-6. LIN Transmission Manipulation Outline



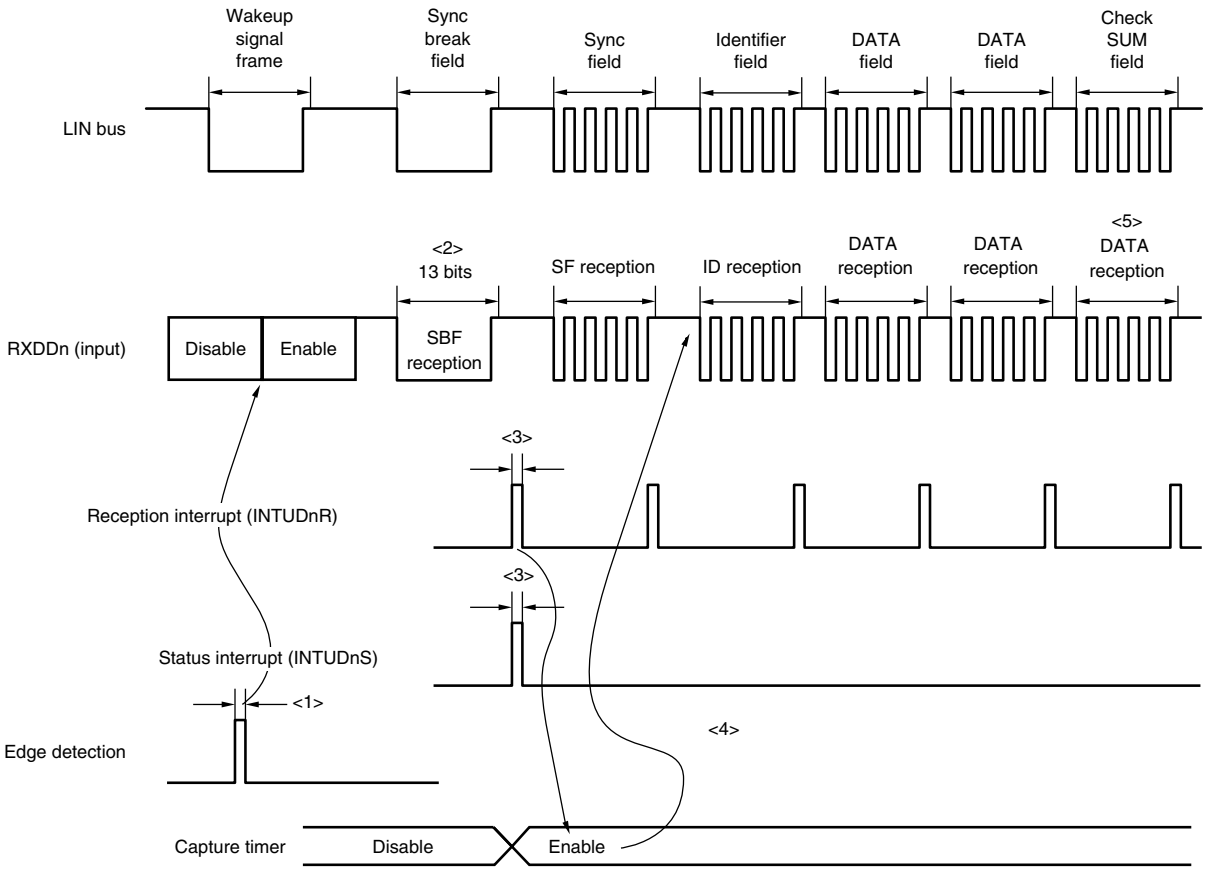
<1> The interval between each field is controlled by software.

<2> SBF output is performed by hardware. The output width is the bit length set by the UDnOPT0.UDnSLS2 to UDnOPT0.UDnSLS0 bits. If even finer output width adjustments are required, such adjustments can be performed using the UDnCTL2.UDnBRS7 to UDnCTL2.UDnBRS0 bits.

<3> 80H transfer in the 8-bit mode is substituted for the wakeup signal frame.

<4> A transmission enable interrupt request signal (INTUDnT) is output at the start of each transmission. The INTUDnT signal is also output at the start of each SBF transmission.

Figure 14-7. LIN Reception Manipulation Outline



<1> The wakeup signal is sent by the pin edge detector, UARTDn is enabled, and the SBF reception mode is set.

<2> The receive operation is performed until detection of the stop bit. Upon detection of SBF reception of 11 or more bits, normal SBF reception end is judged, and an interrupt signal is output. Upon detection of SBF reception of less than 11 bits, an SBF reception error is judged, no interrupt signal is output, and the mode returns to the SBF reception mode.

<3> If SBF reception ends normally, the reception complete interrupt request signal (INTUDnR) is generated when the SBF reception mode select bit (UDnSRS) is “0”. When UDnSRS is “1”, the status interrupt request signal (INTUDnS) is generated and the SBF reception success flag (UDnSSF) is set. When the SBF reception trigger bit (UDnSRT) is “1”, detection of errors such as overrun, parity, and framing errors (UDnOVE, UDnPE, and UDnFE) is suppressed during SBF reception. In addition, data is not transferred from the receive shift register to the receive data register (UDnRX). At this time, UDnRX holds the previous value.

<4> The RXDDn pin is connected to TI (capture input) of the timer, the transfer rate is calculated, and the baud rate error is calculated. The value of the UDnCTL2 register obtained by correcting the baud rate error after stopping UARTD reception operation is set again following SBF reception, causing the status to become the reception status.

<5> Check sum field distinctions are made by software. UARTDn is initialized following CSF reception, and the processing for setting the SBF reception mode again is performed by software. When the UDnOPT1.UDnSRS bit = 1, however, SBF can be automatically received afterward even if the SBS reception mode is not set again.

14.6.3 SBF transmission

First, enable transmission by the following operations.

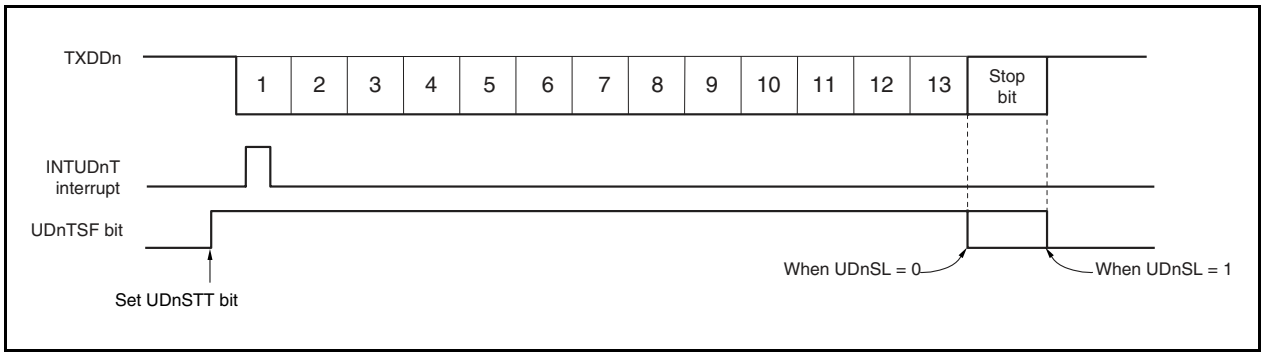
- Specify an operating clock by using UARTD control register 1 (UDnCTL1).
- Specify a baud rate by using UARTD control register 2 (UDnCTL2).
- Specify an output logic level and an SBF length by using UARTD option control register 0 (UDnOPT0).
- Specify whether data consistency is to be checked and a SBF reception mode by using UARTD option control register 1 (UDnOPT1).
- Specify a communication direction, parity, data character length, and stop bit length by using UARTD control register 0 (UDnCTL0).
- Set the power bit and transmission enable bit (UDnPWR = 1 and UDnTXE = 1).

Next, set the SBF transmission trigger (UDnSTT) and start SBF transmission. A low level of 13 to 20 bits specified as an SBF length (UDnSLS2 to UDnSLS0) is output to the TXDDn output pin. When SBF transmission is started, a transmission interrupt request signal (INTUDnT) is generated. After SBF transmission has been completed, the SBF transmission status is automatically canceled and the normal UART transmission mode is restored.

Transmission stands by until data to be transmitted is written to the UDnTX register or the SBF transmission trigger (UDnSTT) is set. Confirm normal reception of SBF by using the reception complete interrupt (INTUDnR) or status interrupt (INTUDnS) during SBF transmission, and then start the next transmission operation.

Caution The following settings must be made.
Normal output (UDnTDL = 0) as the output logic level
LSB first (UDnDIR = 1) as the communication direction
No parity bit (UDnPS1 = 0, UDnPS0 = 0) as the parity
8 bits (UDnCL = 1) as the data character length

Figure 14-8. SBF Transmission



14.6.4 SBF reception

First, enable reception by the following operations and monitor the RXDDn input and detect the start bit.

- Specify an operating clock by using UARTD control register 1 (UDnCTL1).
- Specify a baud rate by using UARTD control register 2 (UDnCTL2).
- Specify an input logic level by using UARTD option control register 0 (UDnOPT0).
- Specify whether data consistency is to be checked and a SBF reception mode by using UARTD option control register 1 (UDnOPT1).
- Specify a communication direction, parity, data character length, and stop bit length by using UARTD control register 0 (UDnCTL0).
- Set the power bit and reception enable bit (UDnPWR = 1 and UDnRXE = 1).

When the SBF reception trigger bit (UDnSRT) is set, reception of SBF is waited for, the RXDDn input is monitored, and the start bit is detected.

When the falling edge of the RXDDn input level is detected, sampling of data input to RXDDn is started. The length of SBF is measured by counting up an internal counter in accordance with the set baud rate, until the stop bit is detected. As soon as the stop bit has been received, SBF is judged to be normal if its length is 11 bits or more. If the SBF reception mode select bit (UDnSRS) is “0”, the reception complete interrupt request signal (INTUDnR) is generated. If the UDnSRS bit is “1”, the status interrupt request signal (INTUDnS) is generated and, at the same time, the SBF reception success flag (UDnSSF) is set. The SBF reception flag (UDnSRF) is automatically cleared and SBF reception is completed.

Detection of errors such as overrun, parity, and framing errors (UDnOVE, UDnPE, and UDnFE) is suppressed. In addition, data is not transferred from the receive shift register to the receive data register (UDnRX). SBF whose width is 10 bits or less is judged to be abnormal. In this case, reception is completed with neither the reception complete interrupt request signal (INTUDnR) nor status interrupt request signal (INTUDnS) generated, and the SBF reception mode is restored again. At this time, the SBF reception flag (UDnSRF) is not cleared.

If transmission is executed with data consistency checked (UDnDCS = 1), the data consistency error flag (UDnDCE) is set and status interrupt request signal (INTUDnS) is output, regardless of whether SBF reception has been successful or failed, if a mismatch between the transmit data and receive data is detected. At this time, INTUDnR is not output.

UARTD can detect reception of a new SBF even during data communication.

Cautions 1. The following settings must be made.
Normal input (UDnRDL = 0) as the input logic level
LSB first (UDnDIR = 1) as the communication direction
No parity bit (UDnPS1 = 0, UDnPS0 = 0) as the parity
8 bits (UDnCL = 1) as the data character length

2. If SBF is transmitted during a data reception, a framing error occurs.

3. Do not set the SBF reception trigger bit (UDnSRT) and SBF transmission trigger bit (UDnSTT) to 1 during an SBF reception (UDnSRF = 1).

Figure 14-9. SBF Reception (UDnSRS = 0)

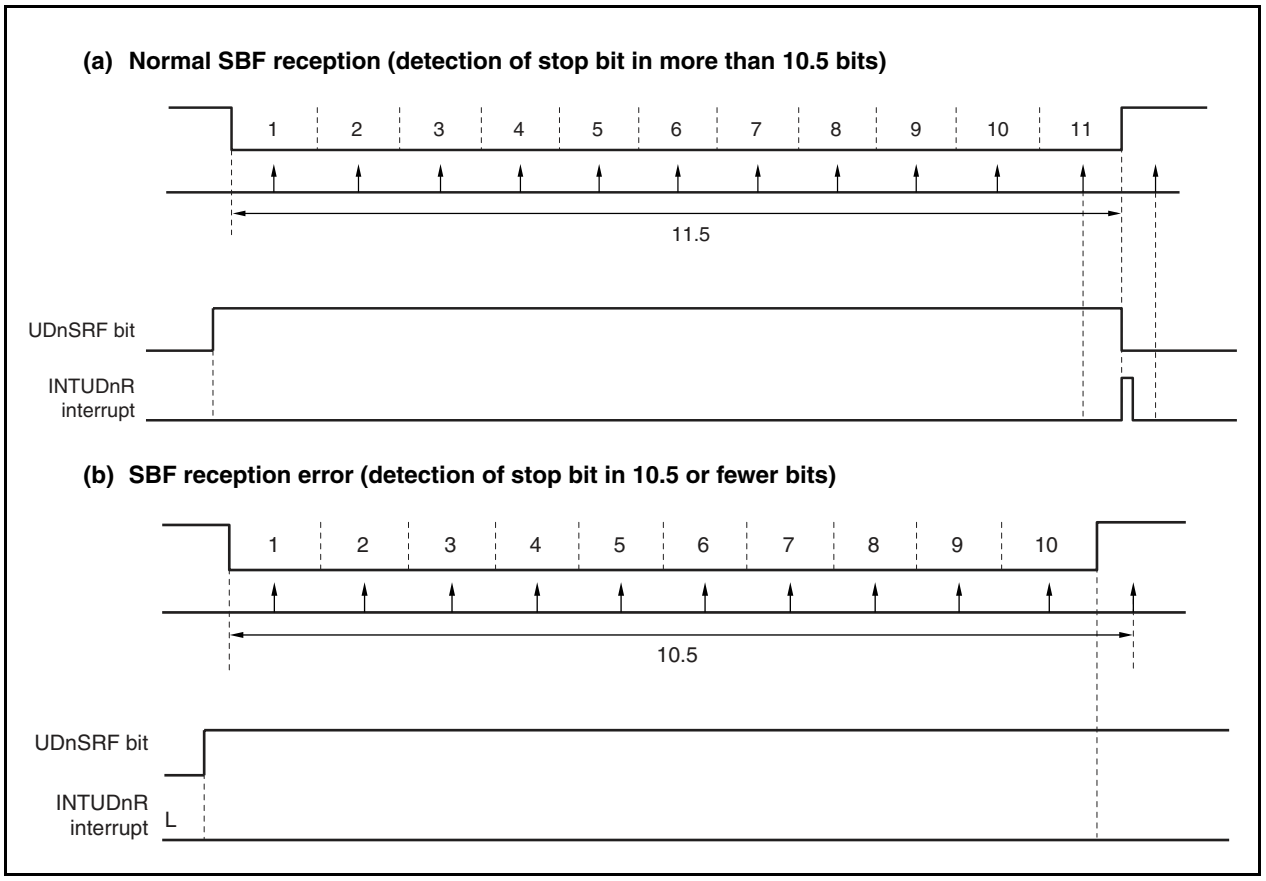
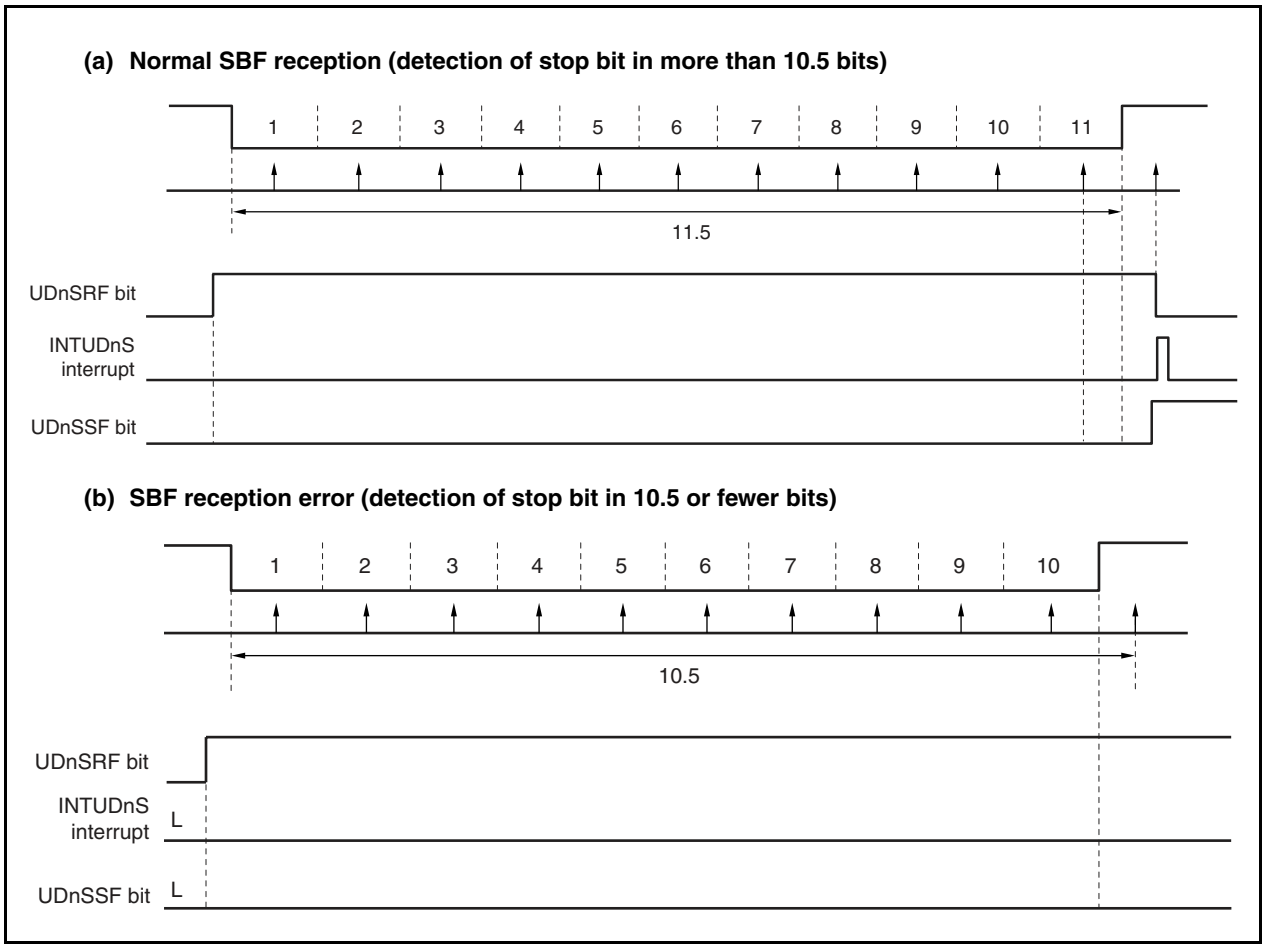


Figure 14-10. SBF Reception (UDnSRS = 1)



14.6.5 Data consistency check function

When the data consistency check select bit (UDnDCS) is set to “1”, transmit data and receive data are compared bitwise when data including a Sync Break Field, Sync Field, Identifier Field, and Check SUM Field is transmitted. If a mismatch is detected and if reception is completed before transmission is completed to correct a mismatch between the transmit data and receive data due to an error between the transmission and reception operations, the status interrupt request signal (INTUDnS) is output and the data consistency error flag (UDnDCE) is set at the end of the frame.

In addition, the next transmission is not executed even if the next transmit data has already been written to the transmit data register (UDnTX) (the written data in UDnTX is ignored). Even if the SBF transmission trigger bit (UDnSTT) is set, SBTT is cleared and SBF is not transmitted. To resume transmission, clear the data consistency error flag (UDnDCE) and then write transmit data to the transmit data register (UDnTX) or set the SBF transmission trigger bit (UDnSTT).

Consistency of data is checked from the start bit of transmission to the first stop bit during SBF transmission. Even if the stop bit length is specified by the stop bit length select bit (UDnSL) to be 2 bits, consistency of the second stop bit is not checked.

When only reception is executed (without transmission), data is not checked for consistency. When transmission is executed, however, consistency between transmit data and input data pin level is checked even if the reception enable bit is disabled (UDnRXE = 0). When UDnRXE = 0, reception itself is not executed. Consequently, the reception complete interrupt request signal (INTUDnR) is not generated when receive data is stored, nor is the status interrupt request signal (INTUDnS) generated when UDnSSF, UDnFE, or UDnOVE is set. It is therefore not necessary to read receive data.

A data consistency error is detected in the following cases.

- If a mismatch between transmit data and receive data is detected during transmission (from the start bit to the first stop bit)
- If reception is completed before transmission is completed when UDnSRF = 0[Note]
- If the rising edge of input data is detected during SBF transmission when UDnSRF = 1 and UDnSRS = 0
- If “1” is detected in the input data during SBF transmission when UDnSRF = 1 and UDnSRS = 1
- If “0” is detected in the input data when the first stop bit is transmitted

Note Except when UDnRXE = 0

Cautions 1. If data consistency check select bit UDnDCS = 0, the data consistency error flag (UDnDCE) is fixed to “0”.

2. Occurrence of a data consistency error does not affect the operation to store receive data in the UDnRX register. However, the data is stored in the register if there is a possibility of a framing error.

3. If SBF is transmitted when UDnSRS = 0, UDnDCS = 1, and UDnSRF = 0, reception ends at the position of the stop bit (10th bit) of data (reception ends before transmission ends). Consequently, a consistency error occurs even if the transmit data and receive data do not mismatch.

Figure 14-11. Timing Example of Data Consistency Error (UDnSRF = 0)

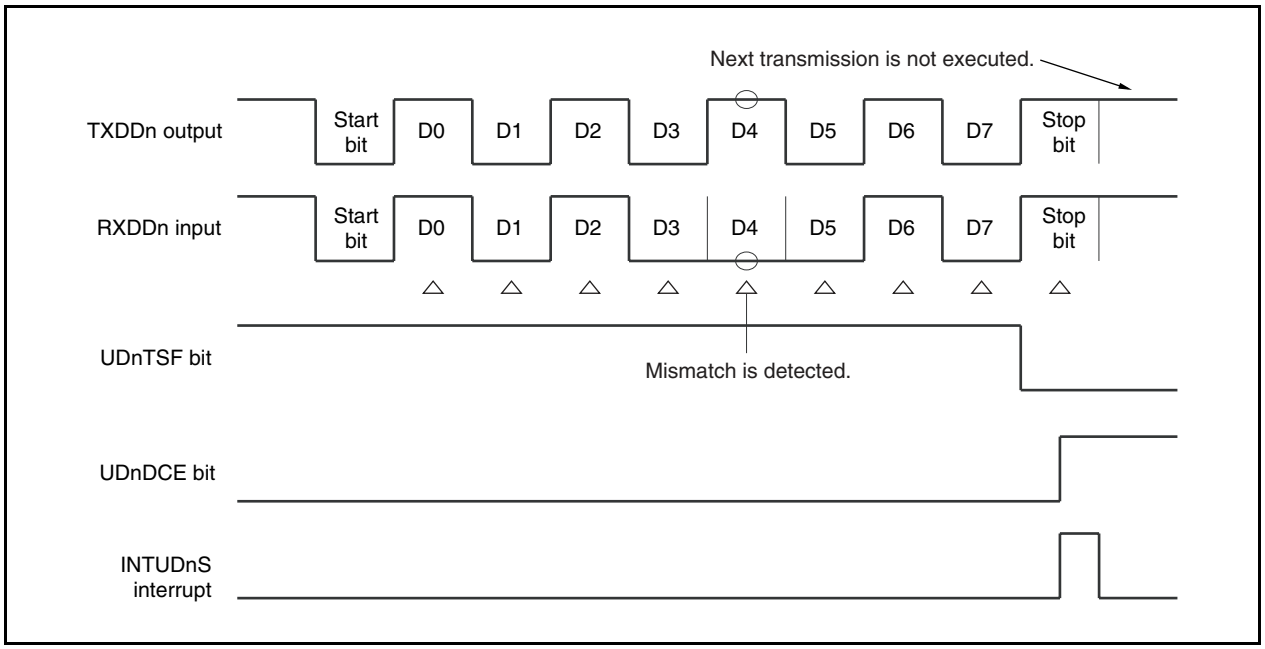
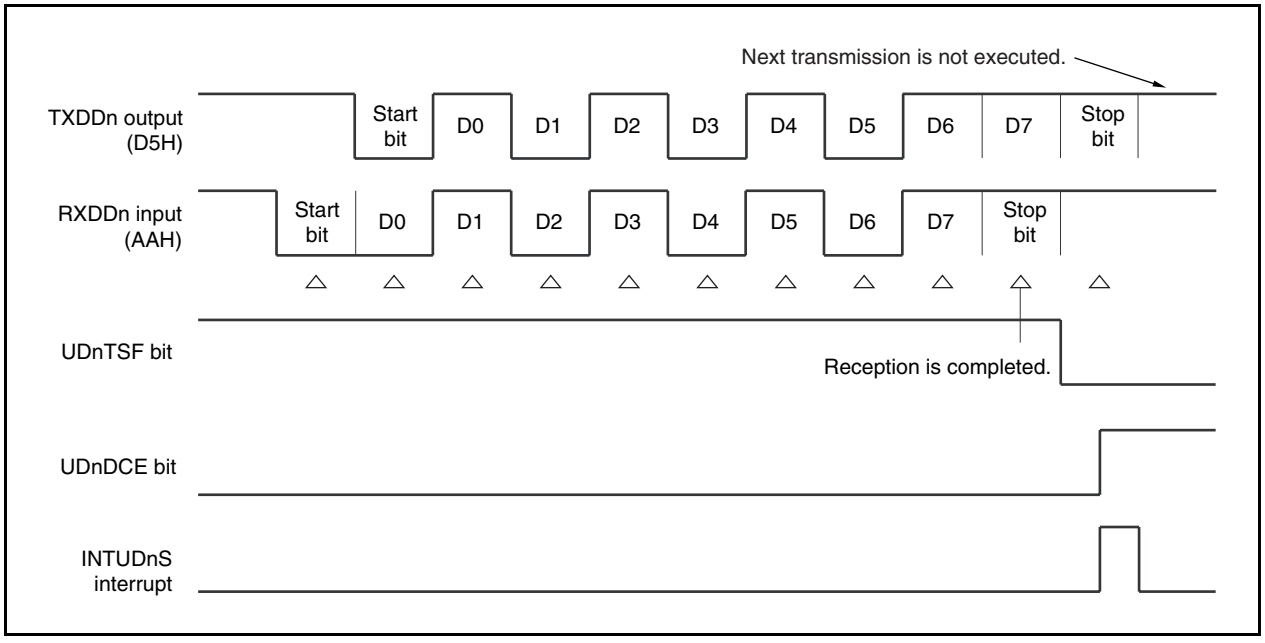


Figure 14-12. Timing Example of Data Consistency Error If There Is Delay Between Transmission and Reception



14.6.6 Selecting SBF reception mode

To receive SBF (Sync Break Field), two types of modes are available, which are selected by SBF reception mode select bit (UDnSRS). The UDnSRS bit can be set only when the data consistency check select bit (UDnDCS) is set to "1".

(1) When UDnSRS = 0

When the SBF reception mode select bit (UDnSRS) is cleared to "0", a mode in which a new SBF is recognized only when the device is waiting for successful SBF reception (UDnSRF = 1) is set.

If the device is not waiting for successful SBF reception (UDnSRF = 0), a framing error or overrun error is identified at the position of the stop bit of data (10th bit) (see **Figure 14-13**). If an overrun error does not occur, receive data is stored in the UDnRX register. If the device is waiting for successful SBF reception (UDnSRF = 1), a framing error or overrun error is not detected nor is the receive data stored in the UDnRX register.

If reception is stopped when UDnSRF = 0 and when transmission of data or the stop bit of the SBF is started, the data consistency error interrupt and flag change their status when transmission of the bit next to the stop bit is started (see **Figure 14-12**). If reception is in progress when transmission of the stop bit is started, the interrupt and flag change their status at the position of the stop bit (see **Figure 14-12**). If reception is stopped when UDnSRF = 1 and when transmission of the stop bit is started, the interrupt and flag change their status when transmission of the bit next to the stop bit is started. During reception, the change takes place when the rising edge of the input data following the stop bit is detected.

Caution The SBF reception success flag (UDnSSF) is fixed to "0" when the SBF reception mode select bit (UDnSRS) = 0.

Figure 14-13. Timing of Identifying Framing/Overrun Error When UDnSRS = 0

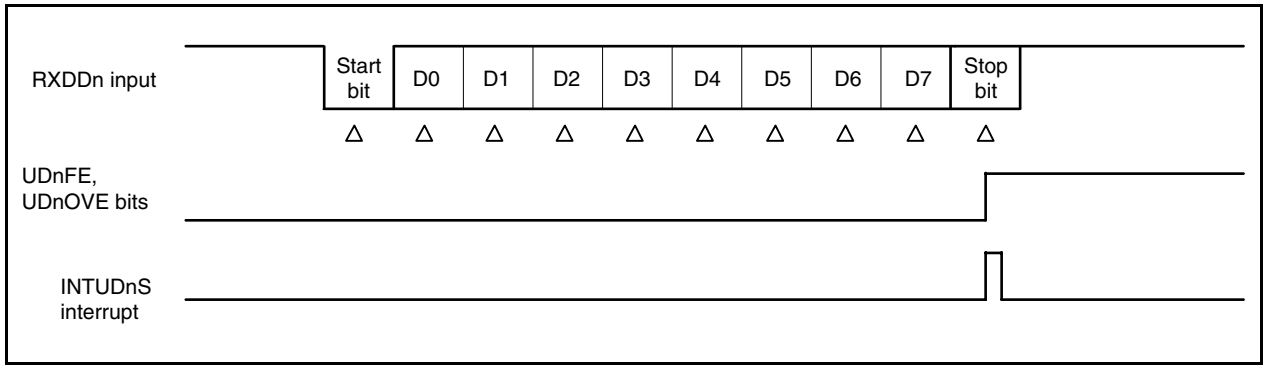


Figure 14-14. Timing Example of Consistency Error Occurrence During SBF Transmission When UDnSRF = 1 (if Reception Is Stopped Before Transmission of Stop Bit Is Started (Data Input Immediately Before Is "1"))

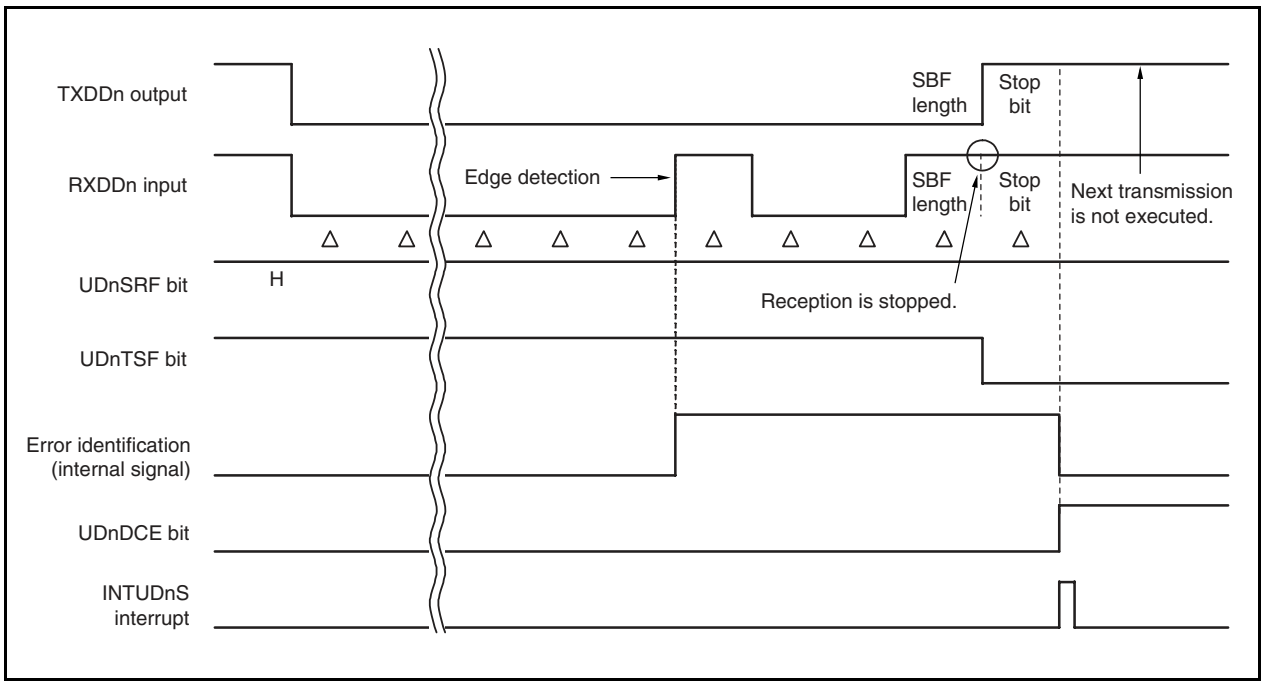
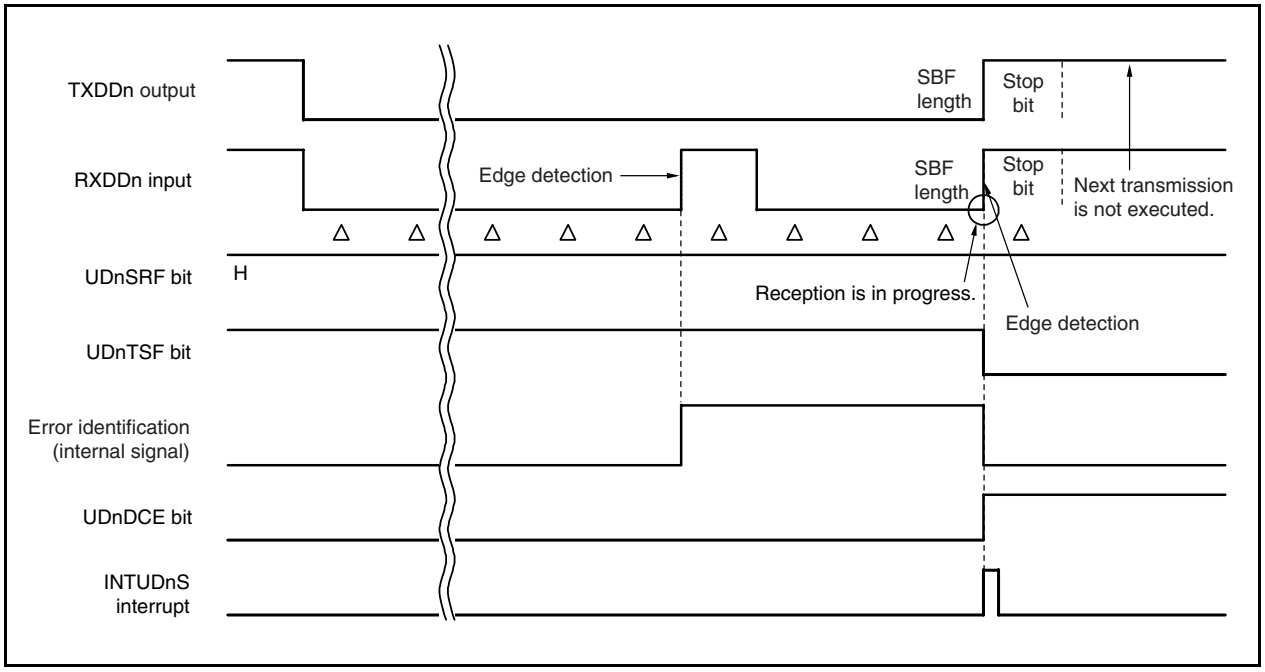


Figure 14-15. Timing Example of Consistency Error Occurrence During SBF Transmission When UDnSRF = 1 (if Reception Is in Progress When Transmission of Stop Bit Is Started (Data Input Immediately Before Is "0"))



(2) When UDnSRS = 1

When the SBF reception mode select bit (UDnSRS) is set to “1”, a mode in which a new SBF is recognized during data communication as well as when the waiting for successful SBF reception (UDnSRF = 1) is set. If a low level is detected at the position of the stop bit of data (10th bit) when not waiting for successful SBF reception (UDnSRF = 0), there is a possibility that a new SBF is being received. Therefore, UARTD waits for a framing or overrun error to be identified until the input data goes high. If a continuous low-level period is of less than 11 bits, it is identified that an error has been detected (see **Figure 14-16**). If the error is not an overrun error, the first 8 bits of the receive data are stored in the UDnRX register. At this time, the SBF reception success flag (UDnSSF) is not set. While the device is waiting for successful SBF reception (UDnSRF = 1), a framing error and overrun error are not detected nor is the receive data stored in the UDnRX register.

If the continuous low-level period is of 11 bits or more, it is judged that a new SBF has been successfully received, and the SBF reception success flag (UDnSSF) is set (see **Figure 14-17**). A framing error and overrun error are not detected. At this time, the receive data is not stored in the UDnRX register.

The data consistency error interrupt and flag change their status, if reception is stopped, when transmission of the stop bit of data or an SBF is started, when UDnSRF = 0, and when transmission of the bit next to the stop bit is started (see **Figure 14-12**). If reception is in progress when transmission of the stop bit is started, the interrupt and flag change their status when input data “1” is detected after the stop bit (see **Figures 14-11** and **14-18**).

If input data “1” is detected when UDnSRF = 1 and after the stop bit has been transmitted, the change takes place when transmission of the next bit is started if reception is stopped (see **Figure 14-19**). During reception, it takes place on the bit that has detected “1” (see **Figure 14-20**).

Figure 14-16. Timing of Identifying Framing/Overrun Error If SBF Reception Fails (When UDnSRF = 0)

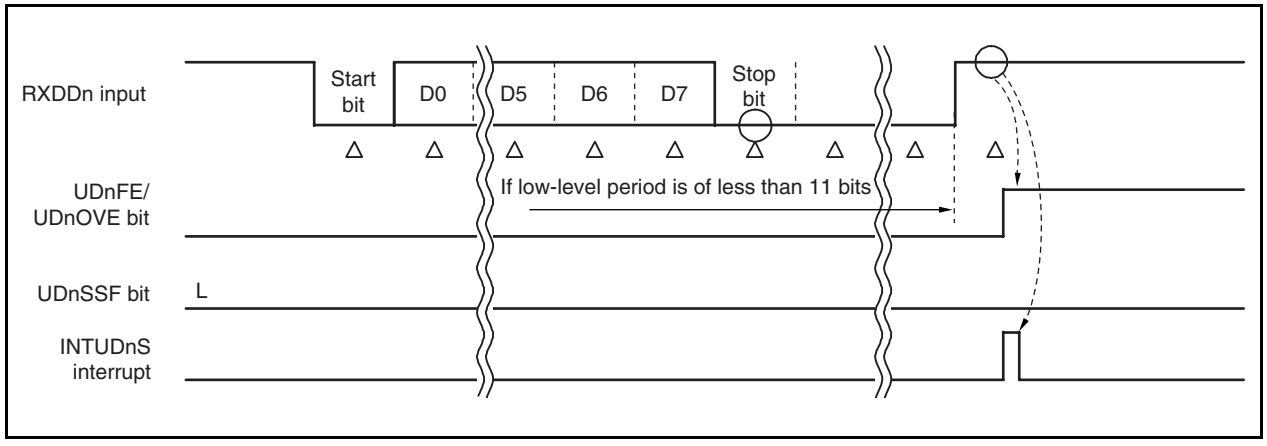


Figure 14-17. Generation Timing of Status Interrupt If SBF Reception Is Successful

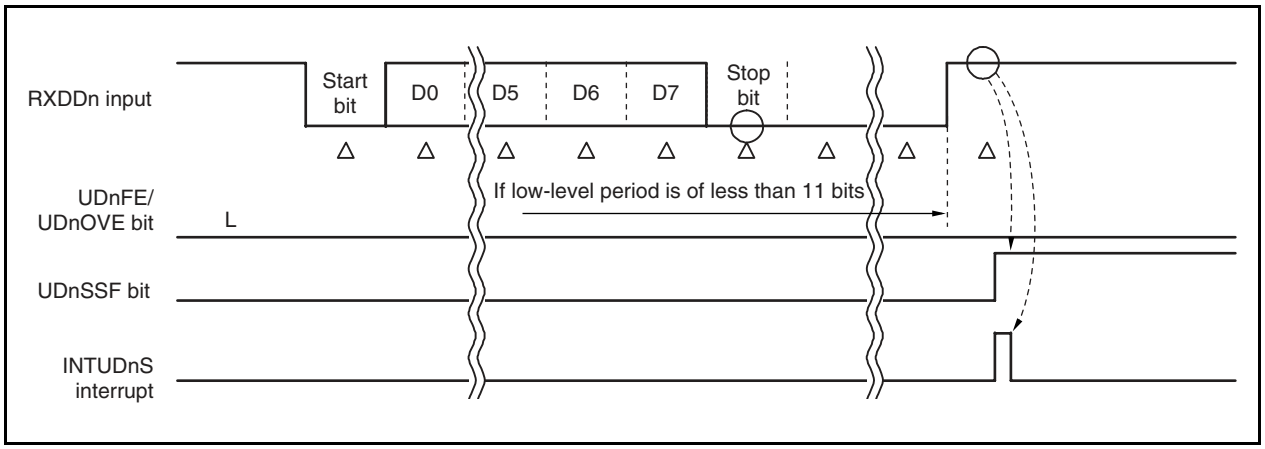


Figure 14-18. Timing Example of Data Consistency Error Occurrence When UDnSRF = 0

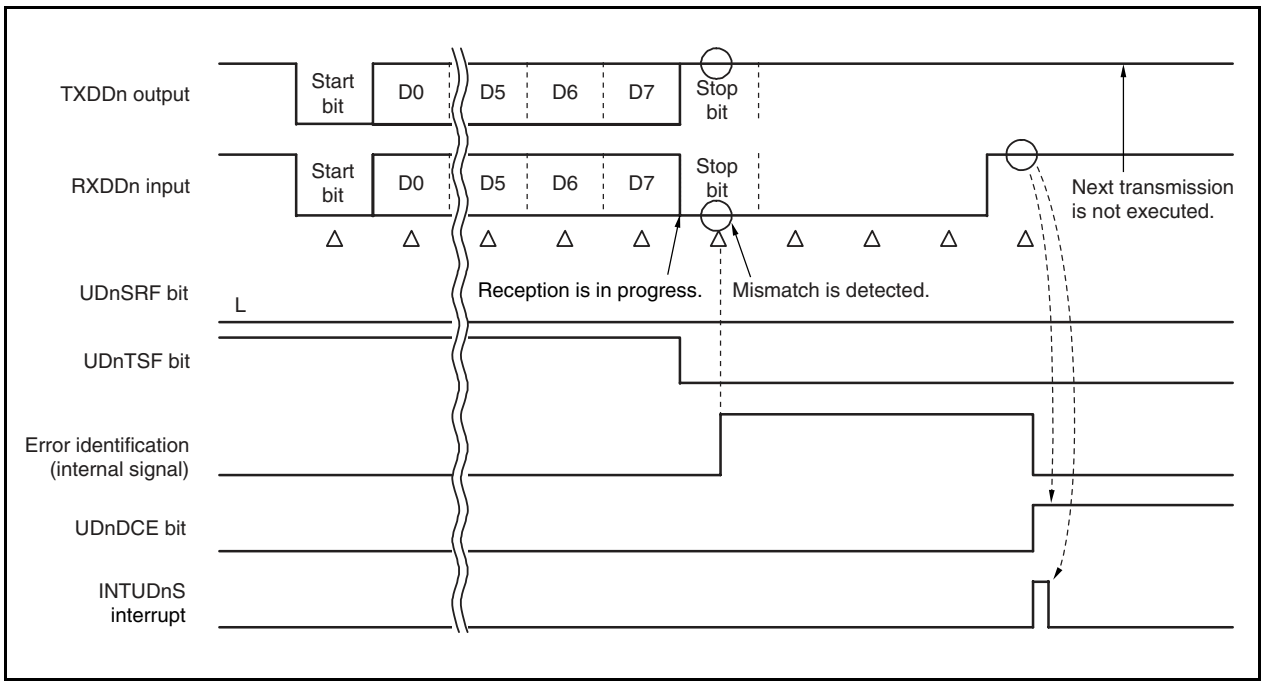


Figure 14-19. Timing Example of Data Consistency Error Occurrence When UDnSRF = 1 (if Reception Is Stopped When Input Data "1" Is Detected After Stop Bit (if Preceding Bit Is "1"))

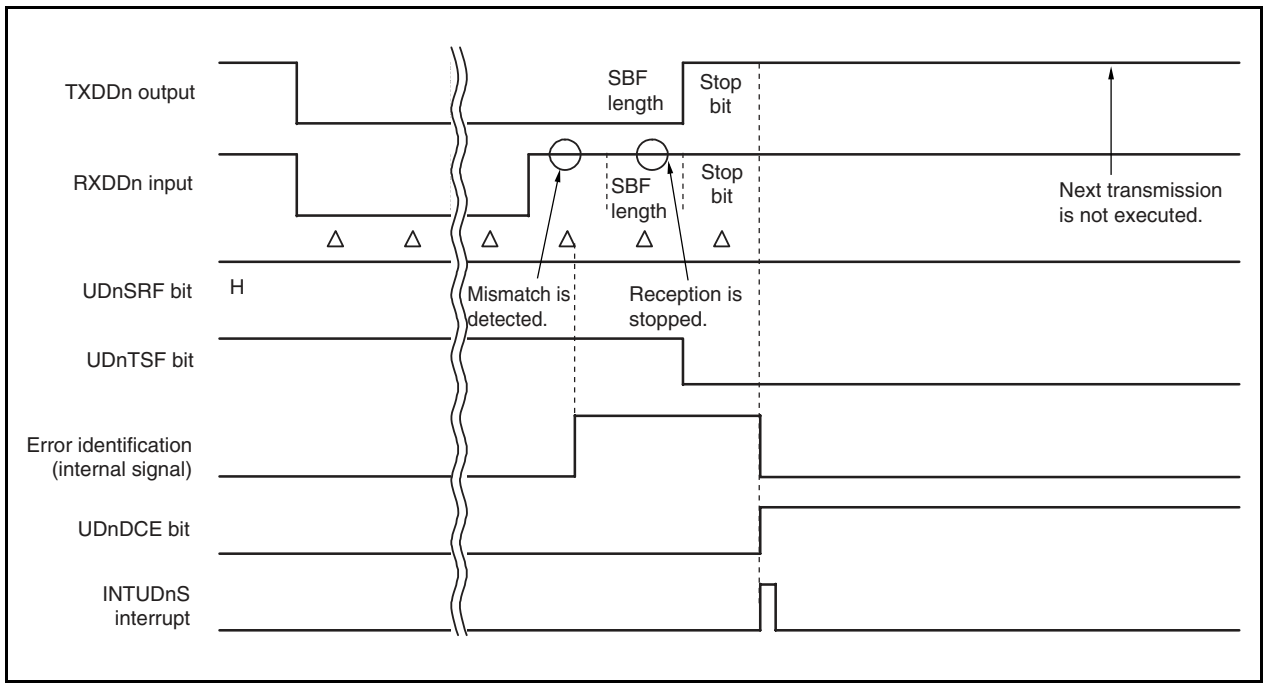
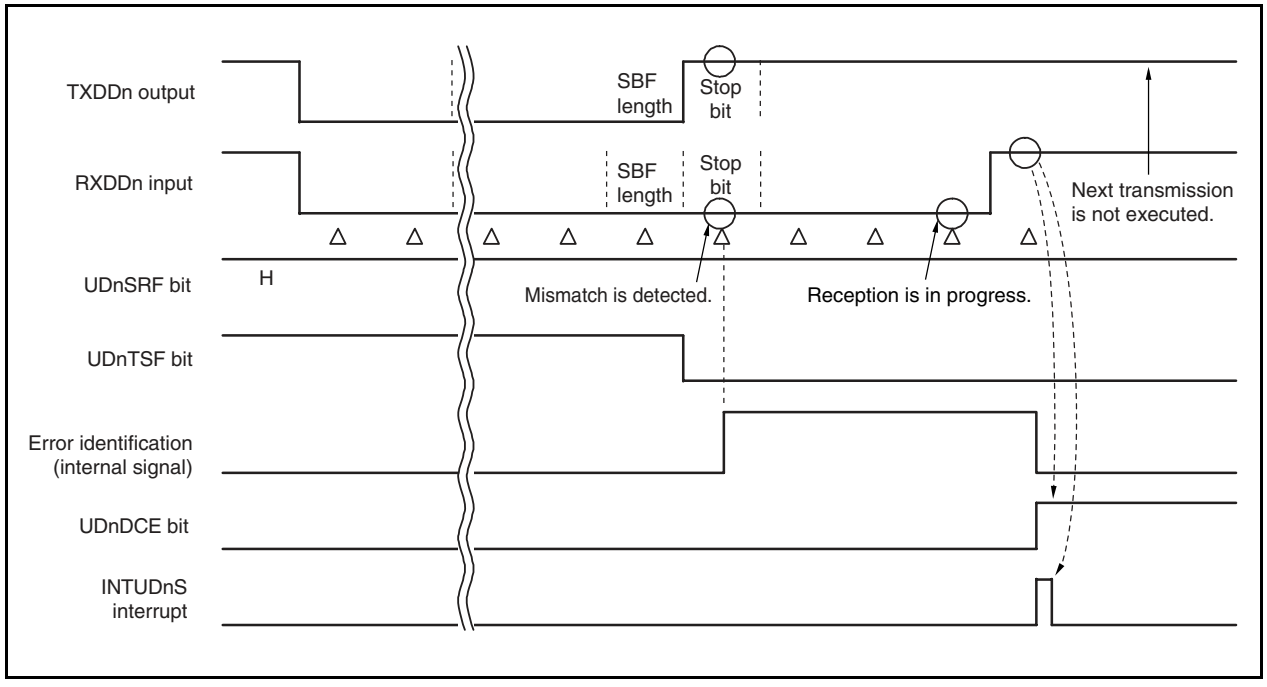


Figure 14-20. Timing Example of Data Consistency Error Occurrence When UDnSRF = 1 (if Reception Is in Progress When Input Data "1" Is Detected After Stop Bit (if Preceding Bit Is "0"))



(3) Recognizing SBF

How SBF is recognized in each of the two modes of SBF reception is explained below.

When UDnSRS = 1, a mode in which SBF is recognized even in the middle of data is set. To recognize the SBF, receive data is sampled at a sampling point and a counter is counted up if the low level of the data is recognized. If the high level is recognized, the counter is cleared to 0. It is judged that SBF has been successfully received when the count value has reached “11”. If SBF is transmitted in the middle of data reception, therefore, it is unclear whether reception is successful or not if the SBF length is of 10 bits + 1 clock to less than 11 bits, depending on the position where the SBF is sampled. However, SBF is correctly received if it is 11 bits or more long.

SBF data is recognized as a SBF if its length is 10.5 bits from the start of reception. When this mode is used, therefore, SBF may be identified by the difference in fluctuation errors of the oscillator between nodes, if transmit data conflicts with some other node or if data “0x00H” is transmitted.

Figure 14-21. Example of Reception of SBF with Low Level 10.5 Bits Long (Success)

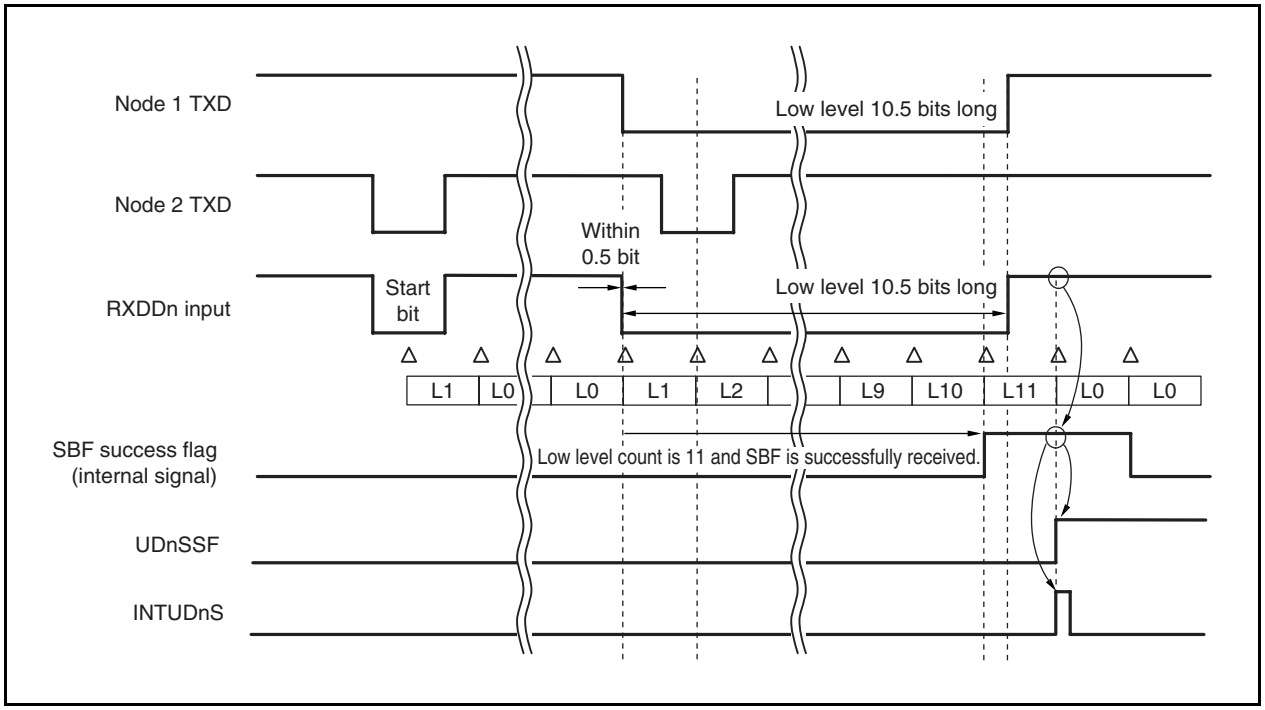


Figure 14-22. Example of Reception of SBF with Low Level 10.5 Bits Long (Failure)

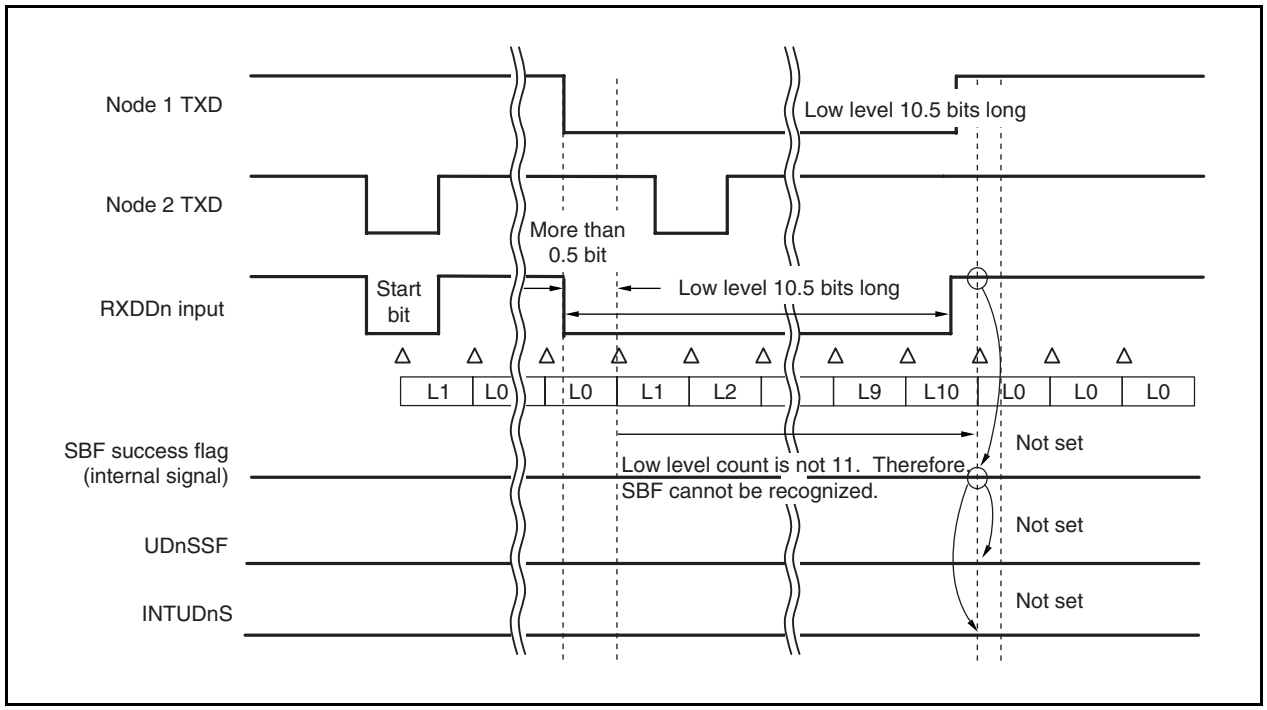


Figure 14-23. Example of Successful Reception of SBF with the Shortest Width

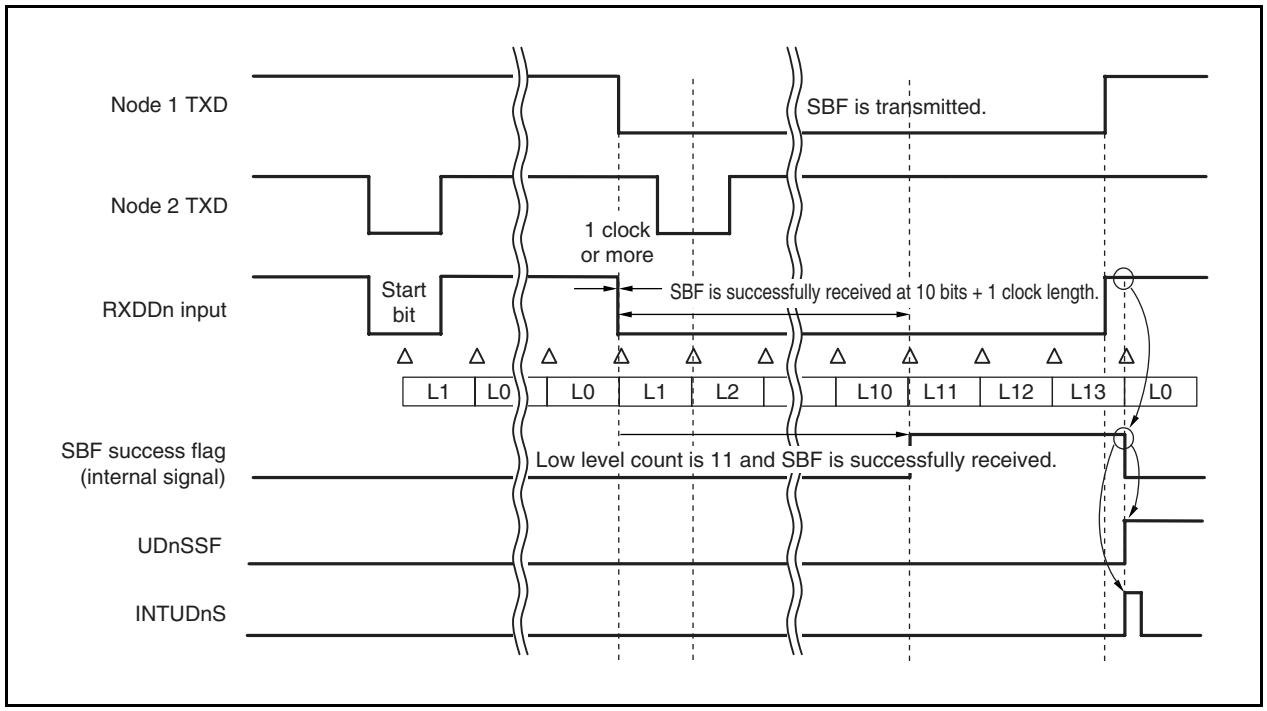
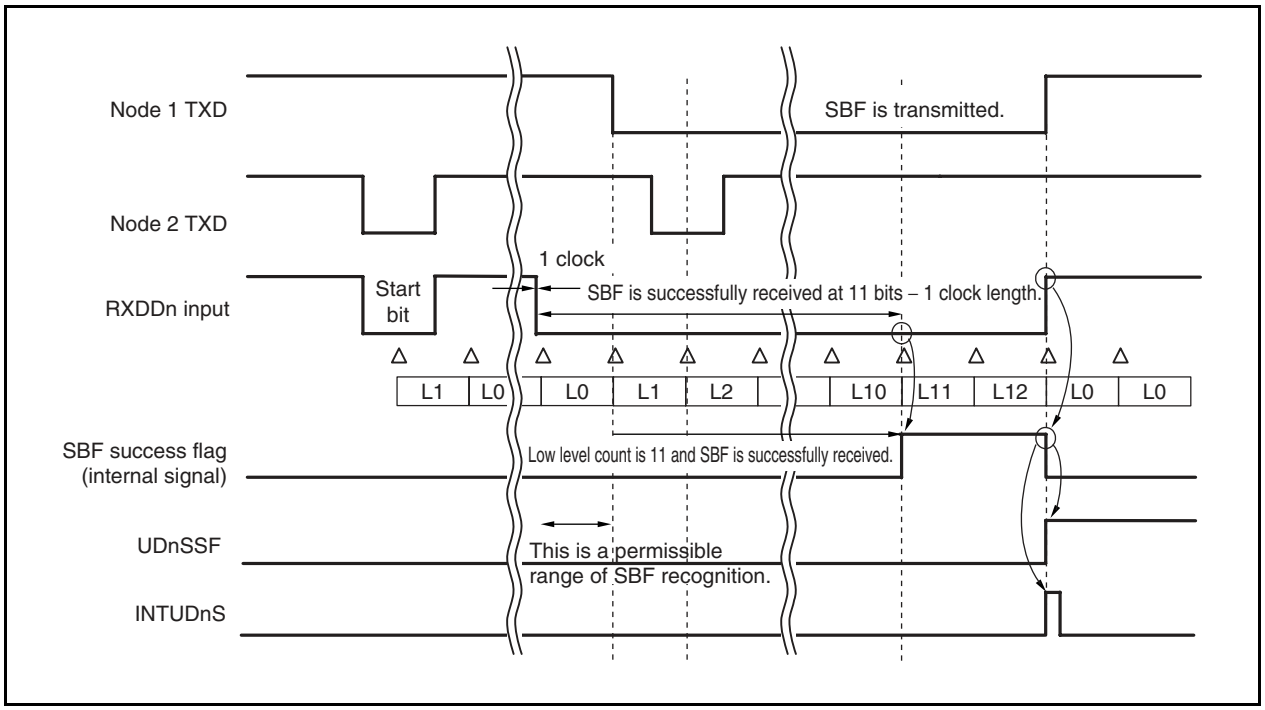


Figure 14-24. Example of Successful Reception of SBF with the Longest Width



14.6.7 UART transmission

First, enable transmission by the following operations.

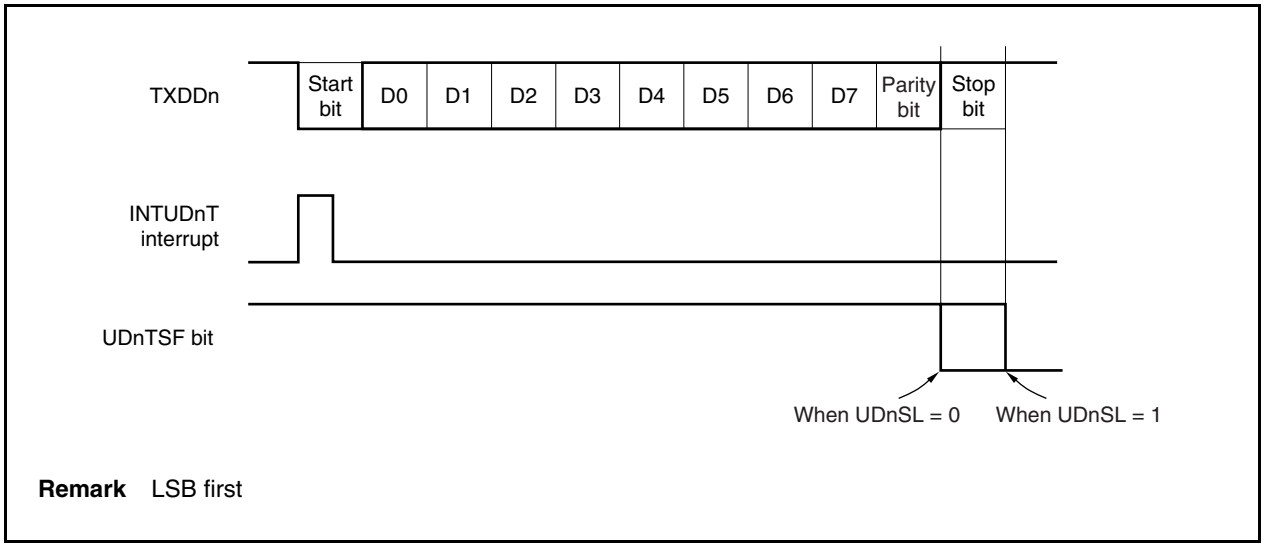
- Specify an operating clock by using UARTD control register 1 (UDnCTL1).
- Specify a baud rate by using UARTD control register 2 (UDnCTL2).
- Specify an output logic level by using UARTD option control register 0 (UDnOPT0).
- Specify a communication direction, parity, data character length, and stop bit length by using UARTD control register 0 (UDnCTL0).
- Set the power bit and transmission enable bit (UDnPWR = 1 and UDnTXE = 1).

When transmit data is written to the transmit buffer register (UDnTX), transmission is started. To change the communication direction, parity, data character length, and/or stop bit length, clear the power bit (UDnPWR = 0) or clear both the transmission enable bit and reception enable bit (UDnTXE = 0 and UDnRXE = 0).

The data stored in the UDnTX register is transferred to the transmit shift register (UDnTXS); is then appended with a start bit, parity bit, and stop bit; and is serially output from the TXDDn pin. In addition, a transmission interrupt request signal (INTUDnT) is generated when transfer of the data stored in the UDnTX register to the UDnTXS shift register is completed.

When the INTUDnT interrupt has been generated, the next transmit data can be written to the UDnTX register.

Figure 14-25. UART Transmission



Remark LSB first

14.6.8 Continuous transmission procedure

UARTDn can write the next transmit data to the UDnTX register when the UARTDn transmit shift register starts the shift operation. The transmit timing of the UARTDn transmit shift register can be judged from the transmission enable interrupt request signal (INTUDnT).

An efficient communication rate is achieved by writing the data to be transmitted next to the UDnTX register during transfer.

During continuous transmission, do not write the next transmit data to the UDnTX register before a transmission interrupt request signal (INTUDnT) is generated after transmit data is written to the UDnTX register and transferred to the UARTDn transmit shift register. If a value is written to the UDnTX register before a transmission interrupt request signal is generated, the previously set transmit data is overwritten by the latest transmit data.

Cautions 1. Before initializing the transmission unit, confirm that the transmission status flag is reset (UDnTSF = 0). If the transmission unit is initialized while UDnTSF = 1, transmission is aborted in midway.

2. Even when data is continuously transmitted, it takes two operating clocks to transmit the next start bit after the stop bit has been transmitted. However, this does not pose any problem to communication because the reception unit initializes the timing by detection of the start bit.

Figure 14-26. Continuous Transmission Processing Flow

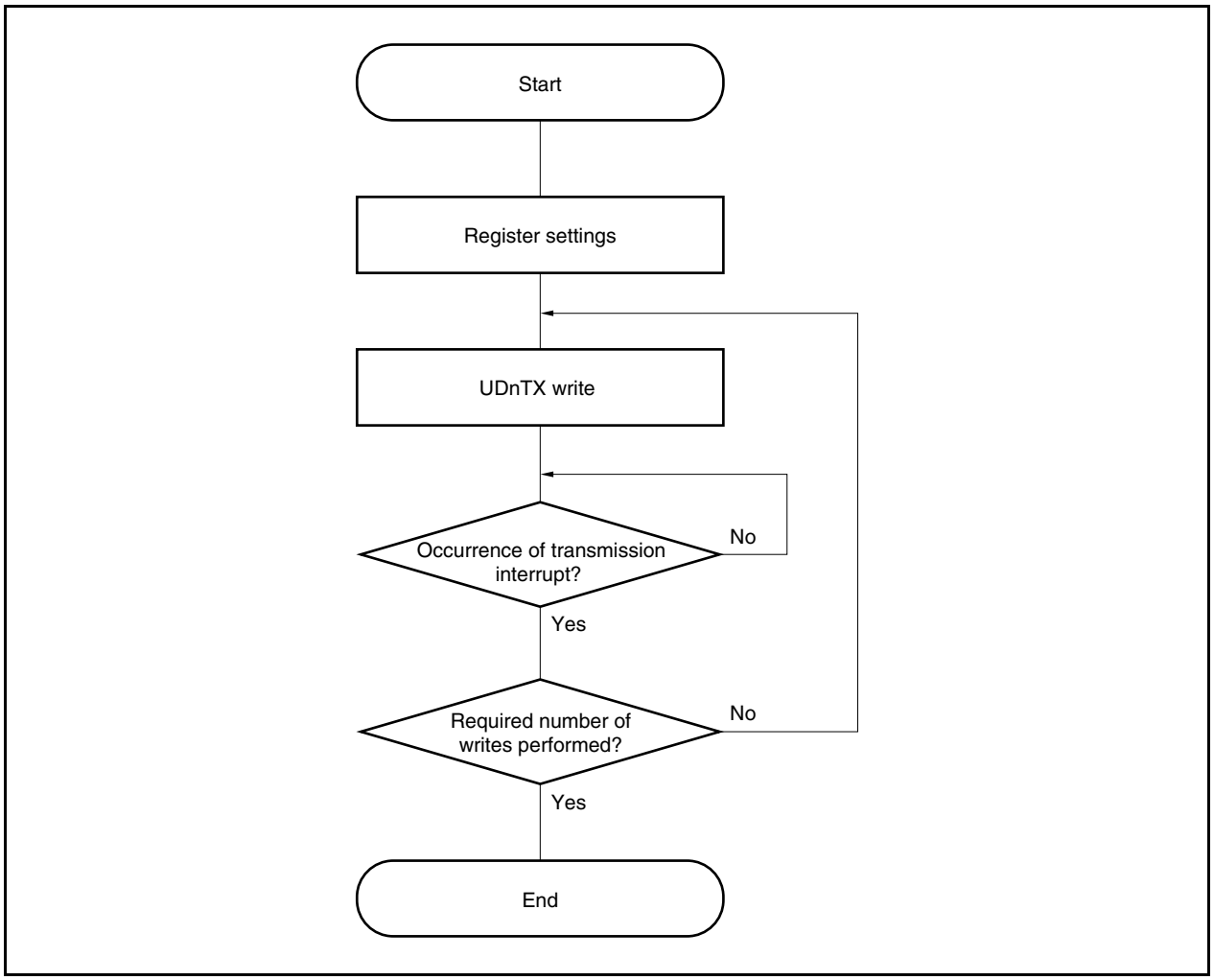
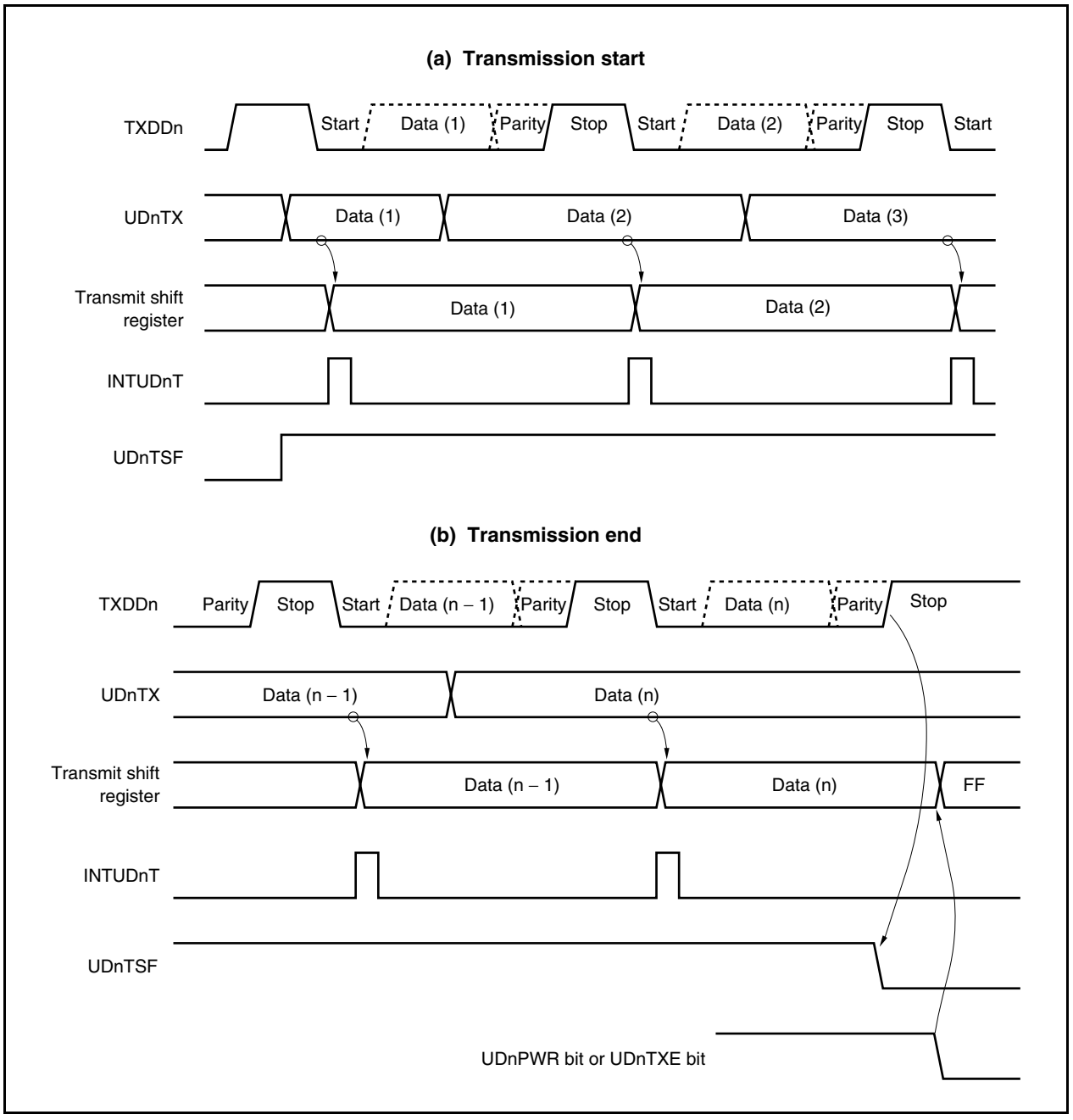


Figure 14-27. Continuous Transmission Operation Timing



14.6.9 UART reception

First, enable reception by the following operations and monitor the RXDDn input and detect the start bit.

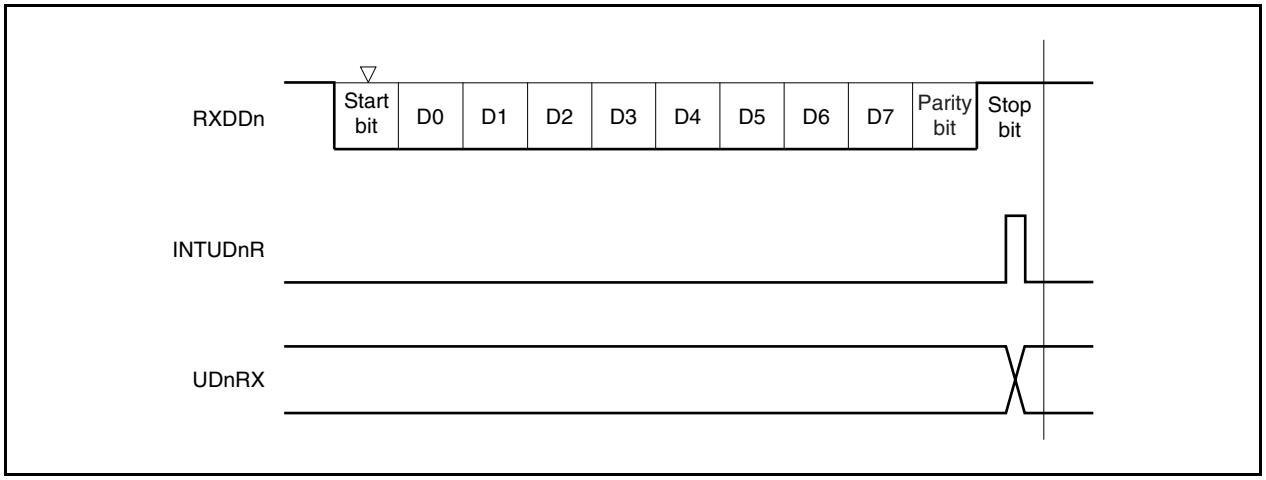
- Specify an operating clock by using UARTD control register 1 (UDnCTL1).
- Specify a baud rate by using UARTD control register 2 (UDnCTL2).
- Specify an output logic level by using UARTD option control register 0 (UDnOPT0).
- Specify a communication direction, parity, data character length, and stop bit length by using UARTD control register 0 (UDnCTL0).
- Set the power bit and reception enable bit (UDnPWR = 1 and UDnRXE = 1).

To change the communication direction, parity, data character length, and/or stop bit length, clear the power bit (UDnPWR = 0) or clear both the transmission enable bit and reception enable bit (UDnTXE = 0 and UDnRXE = 0) beforehand.

The level input to the RXDDn pin is sampled with the operating clock. If the falling edge is detected, sampling of data input to RXDDn is started. If the data is at low level after time of 1/2 bits since detection of the falling edge (indicated by ▽ in Figure 14-28), 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 a specified baud rate. When the stop bit has been received, the reception complete interrupt request signal (INTUDnR) is generated and, at the same time, the data stored in the receive shift register is transferred to the receive data register (UDnRX).

If an overrun error occurs (UDnOVE = 1), however, the receive data is not transferred to UDnRX but discarded. Even if a parity error (UDnPE = 1) or framing error (UDnFE = 1) occurs, on the other hand, reception continues up to the position at which the stop bit is to be received, and the receive data is transferred to the UDnRX register. No matter which reception error has occurred, the INTUDnS interrupt is generated after completion of reception but the INTUDnR interrupt is not generated.

Figure 14-28. UART Reception



Cautions 1. Be sure to read the receive data register (UDnRX) register even when a reception error occurs. If the UDnRX register is not read, an overrun error occurs during reception of the next data.

2. The operation during reception is performed assuming that there is only one stop bit. A second stop bit is ignored.

3. If a low level is always input to the RXDDn pin, it is not recognized as a start bit.

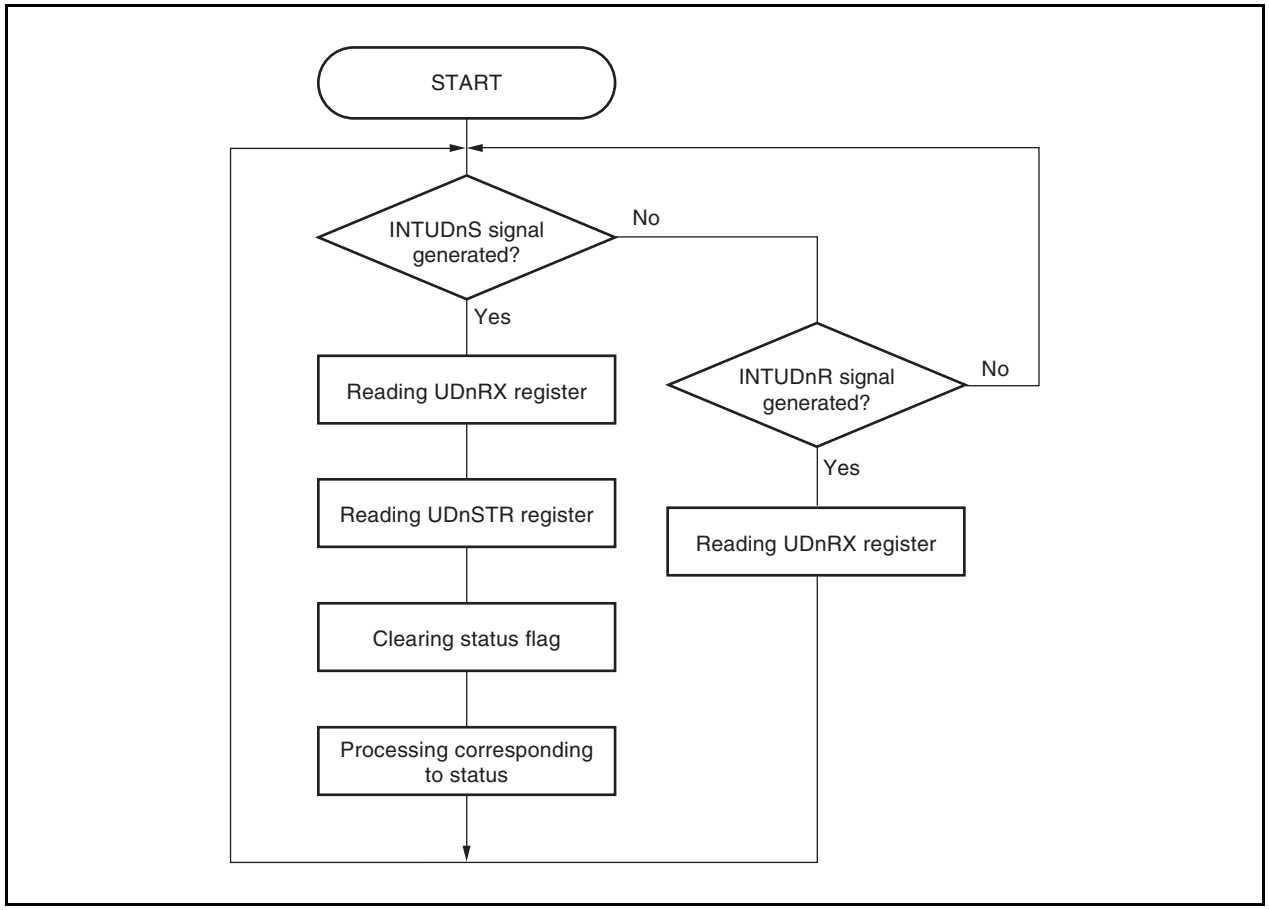
4. If data are continuously received, the next start bit can be detected immediately after the stop bit of the first receive data has been detected (when the reception complete interrupt has been generated).

14.6.10 Procedure of processing in case of reception interrupt

The procedure of the processing that is performed if a reception interrupt occurs is as follows.

The status flag in the processing procedure shown below is a flag other than the UDnTSF bit.

Figure 14-29. Flow of Reading Receive Data



(1) Example of processing corresponding to status

An example of processing corresponding to the status shown in Figure 14-29 during LIN communication is shown below.

Table 14-5. Example of Processing Corresponding to Status During LIN Communication (When UDnSRS = 1, UDnDCS = 1)

UDnSSF	UDnDCE	UDnFE	UDnOVE	Status	Processing Example
1	1	×	×	Detects a mismatch between transmit and receive data when it serves as the master. However, because continuous low levels of 11 bits or more have been received, the next data is not transmitted even if it is ready to be transmitted.	• Does not transmit the next data (Sync Field) but waits for the next time schedule because the other party of communication may not be able to recognize the SBF. • Clears all the status flags and writes the next data to transmit the next data (Sync Field) even though the other party of communication may not be able to recognize the SBF.
1	0	×	×	Succeeded in SBF transmission and SBF reception as the master.	Next data (Sync Field) is transmitted.
				Succeeded in SBF reception as the slave.	Next data (Sync Field) is received.
0	1	×	×	Failed to transmit SBF or data as the master. The next data or SBF is not transmitted even if it is ready to be transmitted.	Discards the subsequent transmit/receive data and waits for the next time schedule.
				Failed to transmit data as the slave. The next data is not transmitted even if it is ready to be transmitted.	Discards the subsequent transmit/receive data and waits for the next time schedule.
0	0	1	×	Framing error is detected when data is received.	Processing of the framing error is performed.
0	0	×	1	Overrun error is detected when data is received. One data received immediately before is discarded.	Processing of overrun error is performed.

Cautions 1. Clear all the status flags that have been set for any processing.

2. If an error is detected during LIN communication or if SBF has been successfully received when UDnSRS = 1, the status interrupt request signal (INTUDnS), instead of the reception complete interrupt request signal (INTUDnR), is generated and the status flag is set in accordance with the communication status.

Remark ×: Don't care

(2) Source of generating status interrupt

The status interrupt is generated by a parity error, framing error, and overrun error, and a data consistency error that occurs only during LIN communication, and successful SBF reception.

The status interrupt request signal (INTUDnS) is generated if any of the above sources is detected. The source of generation can be referenced by using the status register (UDnSTR). Reference UDnSTR by the status interrupt servicing routine and determine the processing to be performed.

The status flags except UDnTSF must be cleared by writing "0" to them via software.

Table 14-6. Source of Generation of Status Interrupt

Status Flag	Generation Source	Description
UDnPE	Parity error	Result of calculating parity of receive data does not match the value of the received parity bit.
UDnFE	Framing error	Stop bit is not detected (low level is detected at the position of the stop bit).
UDnOVE	Overrun error	Reception of the next data is completed before receive data transferred to the receive data register is read.
UDnDCE	Data consistency error	The values of transmit data and receive data do not match when the data consistency check select bit (UDnDCS) is set and when data is transmitted.
UDnSSF	Successful SBF reception	New SBF has been successfully received when the SBF reception mode select bit (UDnSRS) is set (this also happens when the master transmits SBF).

Any of the following processing must be performed depending on the source of generation of the status interrupt.

- Parity error or data consistency error
  Discard the receive data and execute communication again because wrong data has been received. In case of a data consistency error, conflict of data may occur.

- Framing error
  The stop bit may not have been correctly detected or a bit error may have occurred due to erroneous detection of the start bit. In the case of LIN communication, the baud rate may be different from that of the transmission unit or SBF of insufficient length may have been received. If a framing error often occurs, initialize both the transmission unit and reception unit and start communication again.

- Overrun error
  Data of one frame that has been received immediately before has been discarded because the next reception has been completed before receive data is read. If the data is necessary, execute communication again.

- Successful SBF reception
  A new SBF has been successfully received. Prepare to start a new frame slot.

Caution The status flag does not reflect the newest status but accumulates all the sources that have been generated after the status flag has been cleared. Therefore, complete the corresponding processing and clear the status flag before the next reception is completed.

Table 14-7. Timing of Status Interrupt Generation and Timing of Change in Status Flag

Status Flag	Mode Setting			Timing of Change in Status Interrupt and Flag
	SBF Reception Mode Select (UDnSRS)	Data Consistency Check (UDnDCS)	SBF Reception Flag (UDnSRF)	
UDnPE	0	0	0	Reception sample point of stop bit of data (Select “no parity” in any other mode.)
UDnFE, UDnOVE	0	×	0	Reception sample point of stop bit of data
	0	×	1	No change
	1	0	×	Setting prohibited
	1	1	0	Reception sample point at which input data “1” is detected after stop bit of data
	1	1	1	No change
UDnDCE	×	0	×	No change
	0	1	0	Reception sample point of stop bit if reception is being executed when transmission of stop bit is started
				When transmission of bit next to stop bit is started if reception is stopped when transmission of stop bit is started
	0	1	1	When rising edge of data input after stop bit is detected if data input immediately before is “0” when transmission of stop bit is started
				When transmission of bit next to stop bit is started if data input immediately before is “1” when transmission of stop bit is started
	1	1	0	Reception sample point at which input data “1” is detected after stop bit if reception is being executed when transmission of stop bit is started
				When transmission of bit next to stop bit is started if reception is stopped when transmission of stop bit is started
	1	1	1	Reception sample point of bit at which “1” is detected if preceding bit is “0” when input data “1” is detected after transmission of stop bit
				When transmission of next bit is started if preceding bit is “1” when input data “1” is detected after transmission of stop bit
UDnSSF	0	×	×	No change
	1	0	×	Setting prohibited
	1	1	×	Reception sample point at which “1” is detected after input data “0” of 11 bits or more is detected

Remark ×: Don’t care

14.6.11 Parity types and operations

Caution When using the LIN function, fix the UDnCTL0.UDnPS1 and UDnCTL0.UDnPS0 bits to 00.

The parity bit is used to detect bit errors in the communication data. Normally the same parity is used on the transmission side and the reception side.

In the case of even parity and odd parity, it is possible to detect odd-count bit errors. In the case of 0 parity and no parity, errors cannot be detected.

(a) Even parity

(i) During transmission

The number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so as to be an even number. The parity bit values are as follows.

- Odd number of bits whose value is “1” among transmit data: 1
- Even number of bits whose value is “1” among transmit data: 0

(ii) During reception

The number of bits whose value is “1” among the reception data, including the parity bit, is counted, and if it is an odd number, a parity error is output.

(b) Odd parity

(i) During transmission

Opposite to even parity, the number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so that it is an odd number. The parity bit values are as follows.

- Odd number of bits whose value is “1” among transmit data: 0
- Even number of bits whose value is “1” among transmit data: 1

(ii) During reception

The number of bits whose value is “1” among the receive data, including the parity bit, is counted, and if it is an even number, a parity error is output.

(c) 0 parity

During transmission, the parity bit is always made “0”, regardless of the transmit data.

During reception, parity bit check is not performed. Therefore, no parity error occurs, regardless of whether the parity bit is 0 or 1.

(d) No parity

No parity bit is added to the transmit data.

Reception is performed assuming that there is no parity bit. No parity error occurs since there is no parity bit.

14.6.12 Receive data noise filter

This filter samples the RXDDn pin using the base clock of the prescaler output.

When the same sampling value is read twice, the match detector output changes and the RXDDn signal is sampled as the input data. Therefore, data not exceeding 2 clock width is judged to be noise and is not delivered to the internal circuit (see **Figure 14-31**). See **14.7 (1) (a) Base clock** regarding the base clock.

Moreover, since the circuit is as shown in Figure 14-30, the processing that goes on within the receive operation is delayed by 3 clocks in relation to the external signal status.

Figure 14-30. Noise Filter Circuit

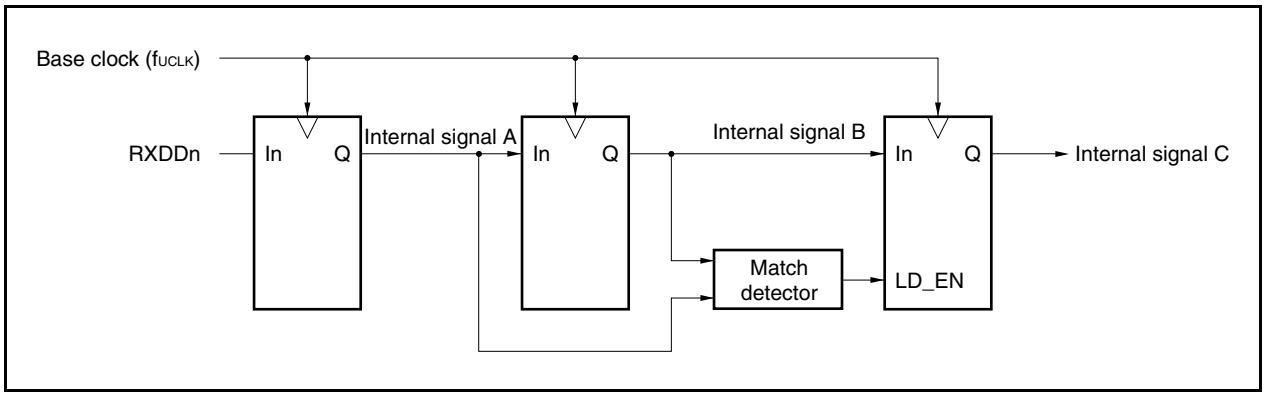
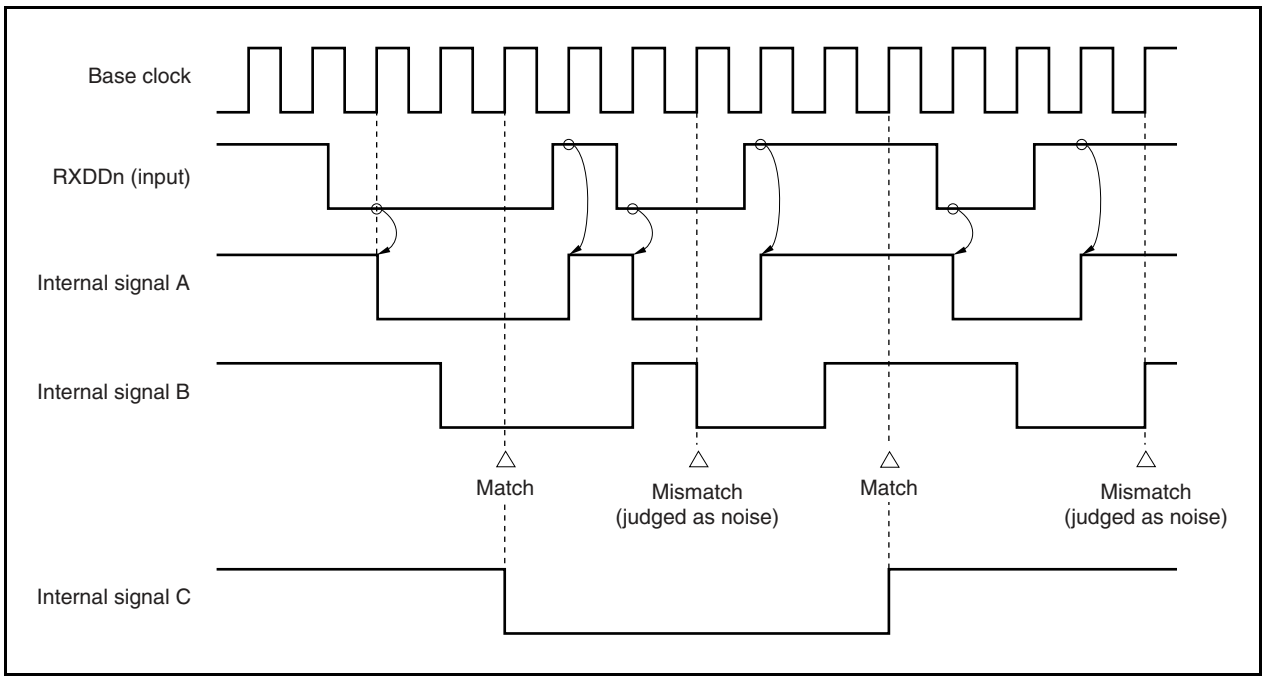


Figure 14-31. Timing of RXDDn Signal Judged as Noise



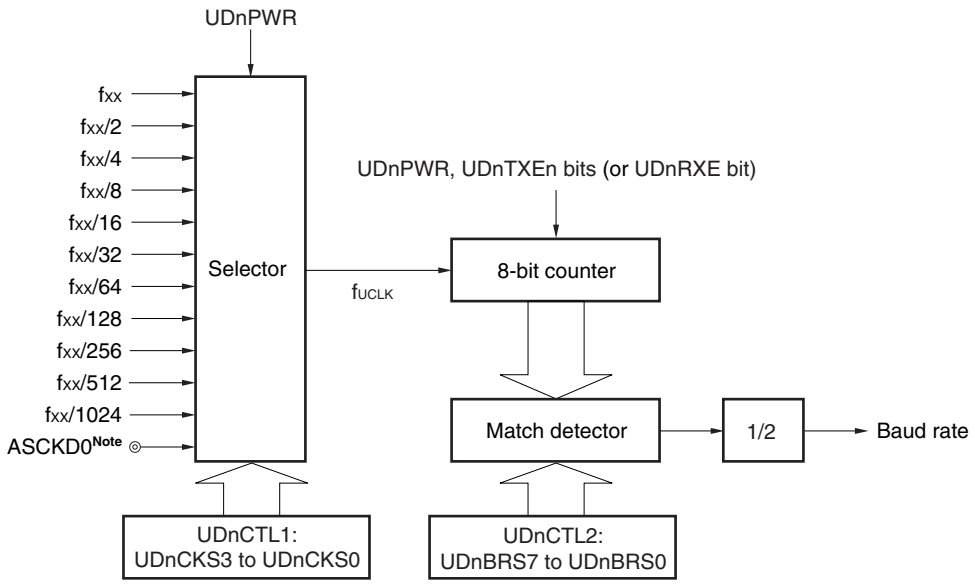
14.7 Dedicated Baud Rate Generator

The dedicated baud rate generator consists of a source clock selector block and an 8-bit programmable counter, and generates a serial clock during transmission and reception with UARTDn. Regarding the serial clock, a dedicated baud rate generator output can be selected for each channel.

There is an 8-bit counter for transmission and another one for reception.

(1) Baud rate generator configuration

Figure 14-32. Configuration of Baud Rate Generator



Note Only UARTD0 is valid; setting UARTD1 to UARTD5 is prohibited.

Remarks 1. fXX: Main clock frequency

2. fUCLK: Base clock frequency

(a) Base clock

When the UDnCTL0.UDnPWR bit is 1, the clock selected by the UDnCTL1.UDnCKS3 to UDnCTL1.UDnCKS0 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 UDnCTL1 register and the UDnCTL2 register.

The base clock is selected by UDnCTL1.UDnCKS3 to UDnCTL1.UDnCKS0 bits.

The frequency division value for the 8-bit counter can be set using the UDnCTL2.UDnBRS7 to UDnCTL2.UDnBRS0 bits.

(2) UARTDn control register 1 (UDnCTL1)

The UDnCTL1 register is an 8-bit register that selects the UARTDn base clock.

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

Reset sets this register to 00H.

Caution Clear the UDnCTL0.UDnPWR bit to 0 before rewriting the UDnCTL1 register.

After reset: 00H R/W Address: UD0CTL1 FFFFFA01H, UD1CTL1 FFFFFA11H, UD2CTL1 FFFFFA21H, UD3CTL1 FFFFFA31H, UD4CTL1 FFFFFA41H, UD5CTL1 FFFFFA51H

	7	6	5	4	3	2	1	0
UDnCTL1	0	0	0	0	UDnCKS3	UDnCKS2	UDnCKS1	UDnCKS0

UDnCKS3	UDnCKS2	UDnCKS1	UDnCKS0	Base clock (f_{UCLK}) selection
0	0	0	0	fxx
0	0	0	1	fxx/2
0	0	1	0	fxx/4
0	0	1	1	fxx/8
0	1	0	0	fxx/16
0	1	0	1	fxx/32
0	1	1	0	fxx/64
0	1	1	1	fxx/128
1	0	0	0	fxx/256
1	0	0	1	fxx/512
1	0	1	0	fxx/1,024
1	0	1	1	External clock[Note] (ASCKD0 pin)
Other than above				Setting prohibited

Note Only UARTD0 is valid; setting UARTD1 to UARTD5 is prohibited.

Remark fxx: Main clock frequency

(3) UARTDn control register 2 (UDnCTL2)

The UDnCTL2 register is an 8-bit register that selects the baud rate (serial transfer speed) clock of UARTDn.

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

Reset sets this register to FFH.

Caution Clear the UDnCTL0.UDnPWR bit to 0 or clear the UDnTXE and UDnRXE bits to 00 before rewriting the UDnCTL2 register.

After reset FFH R/W Address: UD0CTL2 FFFFFA02H, UD1CTL2 FFFFFA12H, UD2CTL2 FFFFFA22H, UD3CTL2 FFFFFA32H, UD4CTL2 FFFFFA42H, UD5CTL2 FFFFFA52H

	7	6	5	4	3	2	1	0
UDnCTL2	UDnBRS7	UDnBRS6	UDnBRS5	UDnBRS4	UDnBRS3	UDnBRS2	UDnBRS1	UDnBRS0

UDn BRS7	UDn BRS6	UDn BRS5	UDn BRS4	UDn BRS3	UDn BRS2	UDn BRS1	UDn 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 UDnCTL1.UDnCKS3 to UDnCTL1.UDnCKS0 bits

(4) Baud rate

The baud rate is obtained by the following equation.

$$\text{Baud rate} = \frac{f_{UCLK}}{2 \times k} \text{ [bps]}$$

When using the internal clock, the equation will be as follows (when using the ASCKD0 pin as clock at UARTD0, calculate using the above equation).

$$\text{Baud rate} = \frac{f_{XX}}{2^{m+1} \times k} \text{ [bps]}$$

Remark f_{UCLK} = Frequency of base clock selected by the UDnCTL1.UDnCKS3 to UDnCTL1.UDnCKS0 bits
f_{XX}: Main clock frequency
m = Value set using the UDnCTL1.UDnCKS3 to UDnCTL1.UDnCKS0 bits (m = 0 to 10)
k = Value set using the UDnCTL2.UDnBRS7 to UDnCTL2.UDnBRS0 bits (k = 4 to 255)

The baud rate error is obtained by the following equation.

$$\text{Error (\%)} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Target baud rate (correct baud rate)}} - 1 \right) \times 100 \text{ [\%]}$$

$$= \left(\frac{f_{UCLK}}{2 \times k \times \text{Target baud rate}} - 1 \right) \times 100 \text{ [\%]}$$

When using the internal clock, the equation will be as follows (when using the ASCKD0 pin as clock at UARTD0, calculate the baud rate error using the above equation).

$$\text{Error (\%)} = \left(\frac{f_{XX}}{2^{m+1} \times k \times \text{Target baud rate}} - 1 \right) \times 100 \text{ [\%]}$$

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 and set the UDnCTL1 and UDnCTL2 registers (when using internal clock).

<1> Set k = fxx/(2 × Target baud rate). Set m = 0.
<2> Set k = k/2 and m = m + 1 where k ≥ 256.
<3> Repeat <2> until k < 256.
<4> Roundup the first decimal place of k.
If k = 256 by the roundup, perform <2> again (k will become 128).
<5> Set m to the UDnCTL1 register and k to the UDnCTL2 register.

Example: When fxx = 20 MHz and target baud rate = 153,600 bps
<1> k = 20,000,000/(2 × 153,600) = 65.10…, m = 0
<2>, <3> k = 65.10… < 256, m = 0
<4> Set value of UDnCTL2 register: k = 65 = 41H, set value of UDnCTL1 register: m = 0

Actual baud rate = 20,000,000/(2 × 65)
= 153,846 [bps]

Baud rate error = {20,000,000/(2 × 65 × 153,600) − 1} × 100
= 0.160 [%]

The representative examples of baud rate settings are shown below.

Table 14-8. Baud Rate Generator Setting Data

Baud Rate (bps)	fxx = 32 MHz			fxx = 20 MHz			fxx = 16 MHz		
	UDnCTL1	UDnCTL2	ERR (%)	UDnCTL1	UDnCTL2	ERR (%)	UDnCTL1	UDnCTL2	ERR (%)
300	08H	D0H	0.16	08H	82H	0.16	07H	D0H	0.16
600	07H	D0H	0.16	07H	82H	0.16	06H	D0H	0.16
1200	06H	D0H	0.16	06H	82H	0.16	05H	D0H	0.16
2400	05H	D0H	0.16	05H	82H	0.16	04H	D0H	0.16
4800	04H	D0H	0.16	04H	82H	0.16	03H	D0H	0.16
9600	03H	D0H	0.16	03H	82H	0.16	02H	D0H	0.16
19200	02H	D0H	0.16	02H	82H	0.16	01H	D0H	0.16
31250	02H	80H	0.00	01H	A0H	0.00	01H	80H	0.00
38400	01H	D0H	0.16	01H	82H	0.16	00H	D0H	0.16
76800	00H	D0H	0.16	00H	82H	0.16	00H	68H	0.16
115200	00H	8BH	−0.08	00H	57H	−0.22	00H	45H	0.64
153600	00H	68H	0.16	00H	41H	0.16	00H	34H	0.16
312500	00H	33H	0.39	00H	20H	0.00	00H	1AH	−1.54
625000	00H	1AH	−1.54	00H	10H	0.00	00H	0DH	−1.54
1250000	00H	0DH	−1.54	00H	08H	0.00	−	−	−

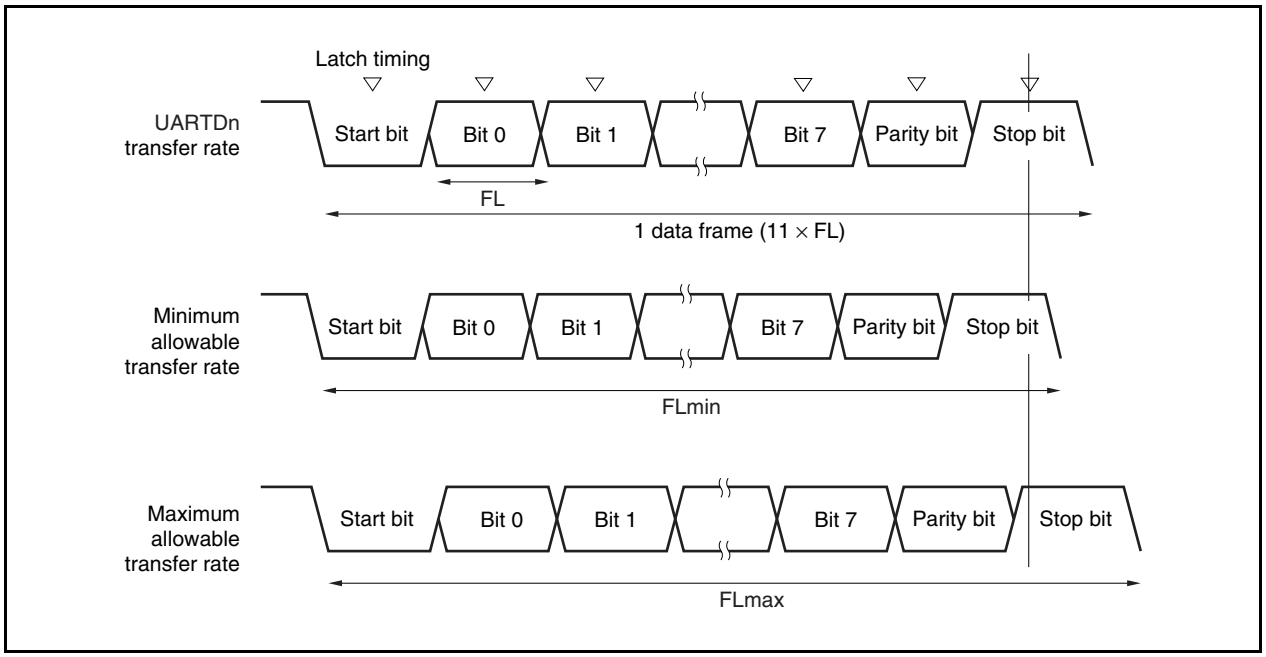
Remark fxx: Main clock frequency
ERR: Baud rate error (%)

(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 during reception must be set within the allowable error range using the following equation.

Figure 14-33. Allowable Baud Rate Range During Reception



As shown in Figure 14-33, the receive data latch timing is determined by the counter set using the UDnCTL2 register following start bit detection. The transmit data can be normally received if up to the last data (stop bit) can be received in time for this latch timing.

When this is applied to 11-bit reception, the following is the theoretical result.

$FL = (Brate)^{-1}$

Brate: UARTDn baud rate (n = 0 to 3)

k: Setting value of UDnCTL2.UDnBRS7 to UDnCTL2.UDnBRS0 bits (n = 0 to 3)

FL: 1-bit data length

Latch timing margin: 2 clocks

Minimum allowable transfer rate: $FLmin = 11 \times FL - \frac{k-2}{2k} \times FL = \frac{21k+2}{2k} FL$

Therefore, the maximum baud rate that can be received by the destination is as follows.

$$BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2} \text{ Brate}$$

Similarly, obtaining the following maximum allowable transfer rate yields the following.

$$\frac{10}{11} \times FLmax = 11 \times FL - \frac{k + 2}{2 \times k} \times FL = \frac{21k - 2}{2 \times k} FL$$

$$FLmax = \frac{21k - 2}{20\ k} FL \times 11$$

Therefore, the minimum baud rate that can be received by the destination is as follows.

$$BRmin = (FLmax/11)^{-1} = \frac{20k}{21k - 2} \text{ Brate}$$

Obtaining the allowable baud rate error for UARTDn and the destination from the above-described equations for obtaining the minimum and maximum baud rate values yields the following.

Table 14-9. Maximum/Minimum Allowable Baud Rate Error

Division Ratio (k)	Maximum Allowable Baud Rate Error	Minimum Allowable Baud Rate Error
4	+2.32%	–2.43%
8	+3.53%	–3.61%
20	+4.26%	–4.31%
50	+4.56%	–4.58%
100	+4.66%	–4.67%
255	+4.72%	–4.73%

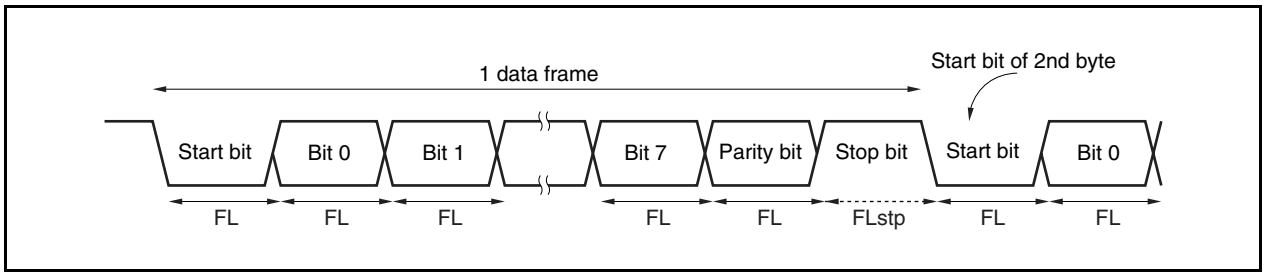
Remarks 1. The reception accuracy depends on the bit count in 1 frame, the input clock frequency, and the division ratio (k). The higher the input clock frequency and the larger the division ratio (k), the higher the accuracy.

2. k: Setting value of UDnCTL2.UDnBRS7 to UDnCTL2.UDnBRS0 bits

(6) Baud rate during continuous transmission

During continuous transmission, the transfer rate from the stop bit to the next start bit is usually 2 base clocks longer. However, timing initialization is performed via start bit detection by the receiving side, so this has no influence on the transfer result.

Figure 14-34. Transfer Rate During Continuous Transfer



Assuming 1 bit data length: FL; stop bit length: FLstp; and base clock frequency: f_{UCLK}, we obtain the following equation.

$$FLstp = FL + 2/f_{UCLK}$$

Therefore, the transfer rate during continuous transmission is as follows.

$$\text{Transfer rate} = 11 \times FL + (2/f_{UCLK})$$

14.8 Cautions

(1) When the clock supply to UARTDn 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 TXDDn 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 UDnCTL0.UDnPWR, UDnCTL0.UDnRXEn, and UDnCTL0.UDnTXEn bits to 000.

(2) UARTD is multiplexed as follows.

(a) The RXDD0 and INTP7 functions cannot be used at the same time. To use the RXDD0 function, do not use the INTP7 function (clear the INTF3.INTF31 bit and INTR3.INTR31 bit to 0). To use the INTP7 function, do not use the RXDD0 function (clear the UD0CTL0.UD0RXE bit to 0).

(b) The RXDD1 and KR7 functions cannot be used at the same time. To use the RXDD1 function, do not use the KR7 function (clear the KRM.KRM7 bit to 0). To use the KR7 function, set the PFC91 bit to 1 and clear the PFCE91 bit to 0, or do not use the RXDD1 function (clear the UD1CTL0.UD1RXE bit to 0).

(c) The RXDD2 and INTP8 functions cannot be used at the same time. To use the RXDD2 function, do not use the INTP8 function (clear the INTF3.INTF39 bit and INTR3.INTR39 bit to 0). To use the INTP8 function, do not use the RXDD2 function (clear the UD2CTL0.UD2RXE bit to 0).

(d) The RXDD3 and INTP14 functions cannot be used at the same time. To use the RXDD3 function, do not use the INTP14 function (clear the INTF4.INTF40 bit and INTR4.INTR40 bit to 0). To use the INTP14 function, do not use the RXDD3 function (clear the UD3CTL0.UD3RXE bit to 0) (μPD70F3757 only).

(e) Pin function RXDD3 for UARTD3 is assigned to pins 22 and 59, and TXDD3 is assigned to pins 23 and 60 and is shared by INTP14.
To use pins 22 and 23 as the RXDD3 and TXDD3 functions of UARTD3, do not set pins 59 and 60 for RXDD3 and TXDD3 functions. Also specify that the edge of INTP14 is not detected (No edge detection) (clear the INT4.INTF40 and INTR4.INTR40 bits to 0 or INTF8.INTF80 and INTR8.INTR80 bits to 0) (μPD70F3757 only).

(f) The RXDD4 and INTP5 functions cannot be used at the same time. To use the RXDD4 function, do not use the INTP5 function (clear the INTF9H.INTF914 bit and INTR9H.INTR914 bit to 0). To use the INTP5 function, do not use the RXDD4 function (it is recommended to set the PFC914 bit to 1 and clear the PFCE914 bit to 0) (μPD70F3757 only).

(g) The RXDD5 and INTP4 functions cannot be used at the same time. To use the RXDD5 function, do not use the INTP4 function (clear the INTF9H.INTF913 bit and INTR9H.INTR913 bit to 0). To use the INTP4 function, do not use the RXDD5 function (it is recommended to set the PFC913 bit to 1 and clear the PFCE913 bit to 0) (μPD70F3757 only).

(3) Start up the UARTDn in the following sequence.
<1> Set the UDnCTL0.UDnPWR bit to 1.
<2> Set the ports.
<3> Set the UDnCTL0.UDnTXE bit to 1, UDnCTL0.UDnRXE bit to 1.

(4) Stop the UARTDn in the following sequence.
<1> Set the UDnCTL0.UDnTXE bit to 0, UDnCTL0.UDnRXE bit to 0.
<2> Set the ports and set the UDnCTL0.UDnPWR bit to 0 (it is not a problem if port setting is not changed).

(5) In transmission mode (UDnCTL0.UDnPWR bit = 1 and UDnCTL0.UDnTXE bit = 1), do not overwrite the same value to the UDnTX register by software because transmission starts by writing to this register. To transmit the same value continuously, overwrite the same value.

(6) In continuous transmission, the communication rate from the stop bit to the next start bit is extended 2 base clocks more than usual. However, the reception side initializes the timing by detecting the start bit, so the reception result is not affected.

(7) If the break command is executed in the on-chip debug (OCD) mode and if UARTD receives data, an overrun error occurs.

CHAPTER 15 3-WIRE VARIABLE-LENGTH SERIAL I/O (CSIB)

The V850ES/Hx3 includes 3-wire variable serial interface B (CSIB). The number of channels differs depending on the product. Table 15-1 shows the number of channels of each product.

Table 15-1. Number of Channels of 3-Wire Variable Serial Interface B (CSIB)

Part Number	V850ES/HE3	V850ES/HF3	V850ES/HG3	V850ES/HJ3
Number of Channels	2 channels (CSIB0, CSIB1)	2 channels (CSIB0, CSIB1)	3 channels (CSIB0, CSIB1)	3 channels (CSIB0 to CSIB2)

In this chapter, the number of channels is expressed as n.

15.1 Features

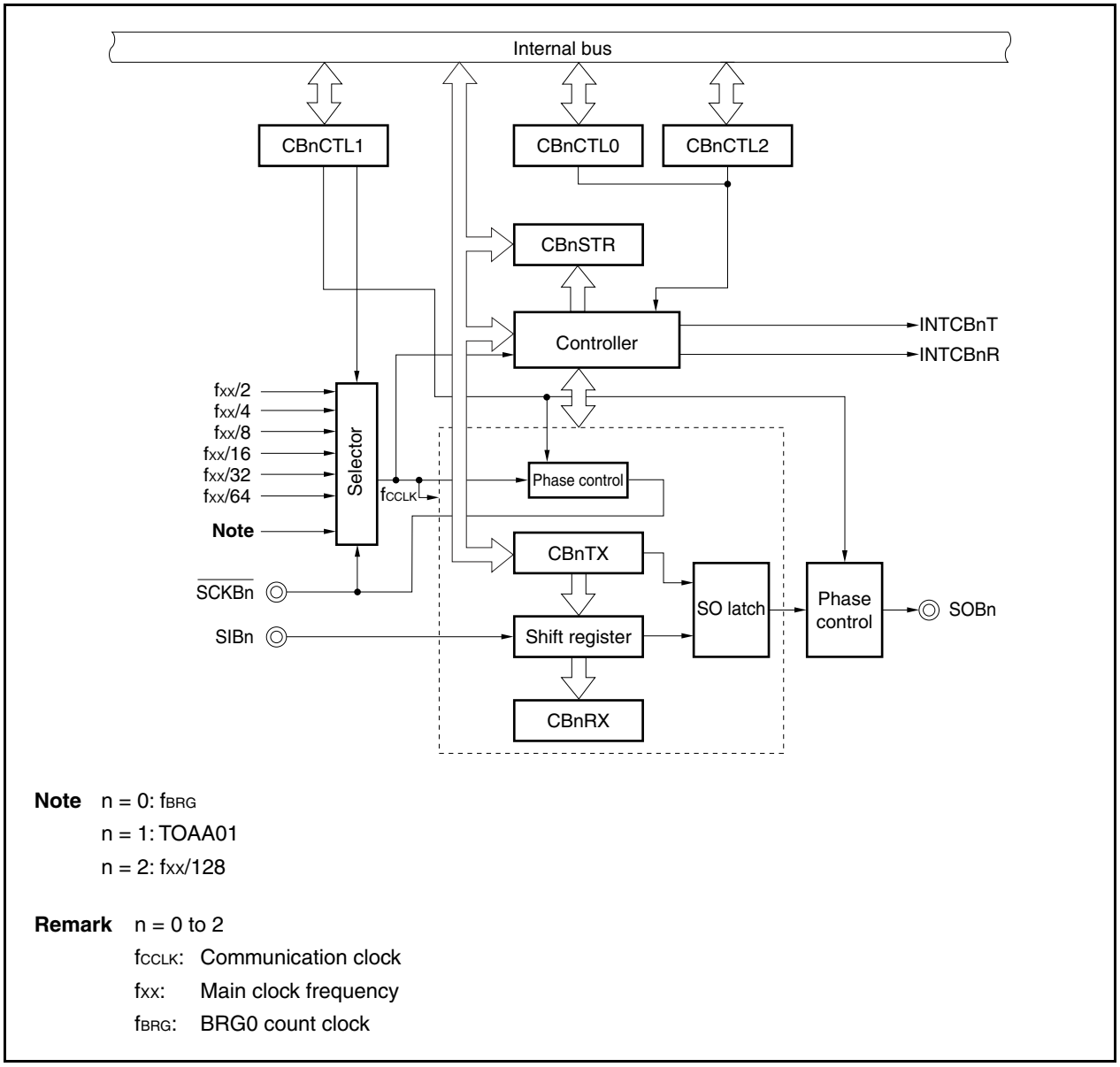
- ○ Transfer rate: Max. 8 Mbps (f_{XX} = 32 MHz, using internal clock)
- ○ Master mode and slave mode selectable
- ○ 8-bit to 16-bit transfer, 3-wire serial interface
- ○ Interrupt request signals (INTCBnT, INTCBnR)
- ○ Serial clock and data phase switchable
- ○ Transfer data length selectable in 1-bit units between 8 and 16 bits
- ○ Transfer data MSB-first/LSB-first switchable
- ○ 3-wire transfer SOBn: Serial data output
  SIBn: Serial data input
  $\overline{\text{SCKBn}}$: Serial clock I/O

  Transmission mode, reception mode, and transmission/reception mode specifiable

15.2 Configuration

The following shows the block diagram of CSIBn.

Figure 15-1. Block Diagram of CSIBn



Note n = 0: f_{BRG}
n = 1: TOAA01
n = 2: $f_{XX}/128$

Remark n = 0 to 2
f_{CCLK}: Communication clock
f_{XX}: Main clock frequency
f_{BRG}: BRG0 count clock

CSIBn includes the following hardware.

Table 15-2. 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)

15.3 Mode Switching of CSIB and Other Serial Interfaces

15.3.1 CSIB0 and UARTD3 mode switching

In the μPD70F3757, CSIB0 and UARTD3 are alternate functions of the same pin and therefore cannot be used simultaneously. Set CSIB0, in advance, using the PMC4, PFC4, and PFCE4 registers, before use (see **4.3.4 Port 4**).

Caution The transmit/receive operation of CSIB0 and UARTD3 is not guaranteed if these functions are switched during transmission or reception. Be sure to disable the one that is not used.

Figure 15-2. CSIB0 and UARTD3 Mode Switch Settings

						PMC42	PMC41	PMC40
PMC4	0	0	0	0	0	1	1	1

						PFC42	PFC41	PFC40
PFC4	0	0	0	0	0	0	0	0

							PFCE41	PFCE40
PFCE4	0	0	0	0	0	0	0	0

PMC41	PFC41	PFCE41	Operation mode
0	×	×	Port I/O mode
1	0	0	CSIB mode (SOB0 output)
	0	1	Key return function
	1	0	UARTD3 mode (TXDD3 output)
	1	1	Setting prohibited

PMC40	PFC40	PFCE40	Operation mode
0	×	×	Port I/O mode
1	0	0	CSIB mode (SIB0 input)
	0	1	Key return function
	1	0	UARTD3 mode (RXDD3 input)/interrupt function
	1	1	Setting prohibited

15.3.2 CSIB2 and UARTD5 mode switching

In the μPD70F3757, CSIB2 and UARTD5 are alternate functions of the same pin and therefore cannot be used simultaneously. Set CSIB2, in advance, using the PMC9, PFC9, and PFCE9 registers, before use (see **4.3.9 Port 9**).

Caution The transmit/receive operation of CSIB2 and UARTD5 is not guaranteed if these functions are switched during transmission or reception. Be sure to disable operation before switching the mode.

Figure 15-3. CSIB2 and UARTD5 Mode Switch Settings

PMC9

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

7	6	5	4	3	2	1	0
PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

PFC9

15	14	13	12	11	10	9	8
PFC915	PFC914	PFC913	PFC912	PFC911	PFC910	PFC99	PFC98

7	6	5	4	3	2	1	0
PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90

PFCE9

15	14	13	12	11	10	9	8
PFCE915	PFCE914	PFCE913	PFCE912	0	0	PFCE99	PFCE98

7	6	5	4	3	2	1	0
PFCE97	PFCE96	PFCE95	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90

PMC912	PFCE912	PFC912	Operation mode
0	×	×	Port I/O mode
1	0	1	$\overline{\text{SCKB2}}$ I/O
1	1	1	TXDD5 output

15.4 Registers

The following registers are used to control CSIBn.

- CSIBn receive data register (CBnRX)
- CSIB 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.

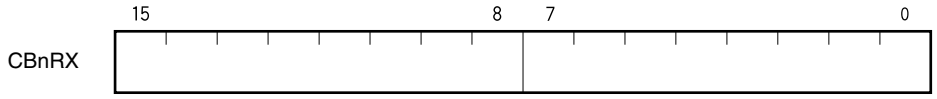
The receive operation is started by reading the CBnRX register in the reception enabled status.

If the transfer data length is 8 bits, the lower 8 bits of this register are read-only in 8-bit units as the CBnRXL register.

Reset sets this register to 0000H.

In addition to reset input, the CBnRX register can be initialized by clearing (to 0) the CBnPWR bit of the CBnCTL0 register.

After reset: 0000H R Address: CB0RX FFFFFD04H, CB1RX FFFFFD14H, CB2RX FFFFFD24H



(2) CSIB transmit data register (CBnTX)

The CBnTX register is a 16-bit buffer register used to write the CSIBn transfer data.

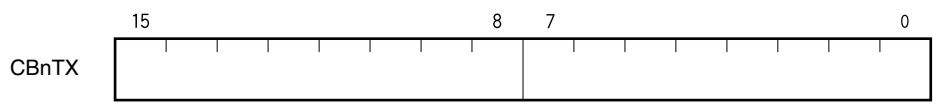
This register can be read or written in 16-bit units.

The transmit operation is started by writing data to the CBnTX register in the transmission enabled status.

If the transfer data length is 8 bits, the lower 8 bits of this register are read-only in 8-bit units as the CBnTXL register.

Reset sets this register to 0000H.

After reset 0000H R/W Address: CB0TX FFFFFD06H, CB1TX FFFFFD16H, CB2TX FFFFFD26H



Remark The communication start conditions are shown below.

Transmission mode (CBnTXE bit = 1, CBnRXE bit = 0):	Write to CBnTX register
Transmission/reception mode (CBnTXE bit = 1, CBnRXE bit = 1):	Write to CBnTX register
Reception mode (CBnTXE bit = 0, CBnRXE bit = 1):	Read from CBnRX register

(3) CSIBn control register 0 (CBnCTL0)

CBnCTL0 is a register that controls the CSIBn serial transfer operation.

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

Reset sets this register to 01H.

(1/3)

After reset: 01H R/W Address: CB0CTL0 FFFFFD00H, CB1CTL0 FFFFFD10H, CB2CTL0 FFFFFD20H

	<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
• The CBnPWR bit controls the CSIBn operation and resets the internal circuit.	

CBnTXE Note	Specification of transmit operation disable/enable
0	Disable transmit operation
1	Enable transmit operation
• The SOBn output is low level when the CBnTXE bit is 0.	

CBnRXE Note	Specification of receive operation disable/enable
0	Disable receive operation
1	Enable receive operation
• When the CBnRXE bit is cleared to 0 in order to disable the receive operation, no reception complete interrupt is output even when the prescribed data is transferred, and the receive data (CBnRX register) is not updated.	

Note These bits can be rewritten only when the CBnPWR bit = 0. However, the CBnPWR bit can be set to 1 at the same time as rewriting these bits.

Caution To forcibly suspend transmission/reception, clear the CBnPWR bit instead of the CBnRXE bit and CBnTXE bit to 0. 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]
The reception complete interrupt request signal (INTCBnR) is generated.
Even if transmission is enabled (CBnTXE bit = 1), the transmission enable interrupt request signal (INTCBnT) is not generated.
If the next transmit data is written during communication (CBnSTR.CBnTSF bit = 1), it is ignored and the next communication is not started. Also, if reception-only communication is set (CBnTXE bit = 0, CBnRXE bit = 1), the next communication is not started even if the receive data is read during communication (CBnSTR. CBbTSF bit = 1).
[In continuous transfer mode]
The continuous transmission is enabled by writing the next transmit data during communication (CBnSTR.CBnTSF bit = 1). Writing the next transmission data is enabled after a transmission enable interrupt (INTCBnT) occurrence.
If reception-only communication is set (CBnTXE bit = 0, CBnRXE bit = 1) in the continuous transfer mode, the next reception is started continuously after a reception complete interrupt (INTCBnR) regardless of the read operation of the CBnRX register.
Therefore, read immediately the receive data from the CBnRX register. If this read operation is delayed, an overrun error (CBnOVE bit = 1) occurs.

Note These bits can be rewritten only when the CBnPWR bit = 0. However, the CBnPWR bit can be set to 1 at the same time as rewriting these bits.

(3/3)

CBnSCE	Specification of start transfer disable/enable
0	Communication start trigger invalid
1	Communication start trigger valid

- In master mode
  This bit enables or disables the communication start trigger.
  (a) In single transmission or transmission/reception mode, or continuous transmission or continuous transmission/reception mode
  The setting of the CBnSCE bit has no influence on communication operation.
  (b) In single reception mode
  Clear the CBnSCE bit to 0 before reading the last receive data because reception is started by reading the receive data (CBnRX register) to disable the reception startup[Note 1].
  (c) In continuous reception mode
  Clear the CBnSCE bit to 0 one communication clock before reception of the last data is completed to disable the reception startup after the last data is received[Note 2].
- In slave mode
  This bit enables or disables the communication start trigger.
  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 the CBnSTR.CBnTSF bit = 0, clear the CBnRXE bit to 0 to disable reception.
  To continue reception, set the CBnSCE bit to 1 to start up the next reception by dummy-reading the CBnRX register.
- In continuous reception mode
  <1> Clear the CBnSCE bit to 0 during the reception of the last data by INTCBnR interrupt servicing.
  <2> Read the CBnRX register.
  <3> Read the last reception data by reading the CBnRX register after acknowledging the CBnTIR interrupt.
  <4> After confirming the CBnSTR.CBnTSF bit = 0, clear the CBnRXE bit to 0 to disable reception.
  To continue reception, set the CBnSCE bit to 1 to wait for the next reception by dummy-reading the CBnRX register.

Notes 1. If the CBnSCE bit is read while it is 1, the next communication operation is started.

2. The CBnSCE bit is not cleared to 0 one communication clock before the completion of the last data reception, the next communication operation is automatically started.

Caution Be sure to set bits 3 and 2 to "0".

(4) CSIBn control register 1 (CBnCTL1)

CBnCTL1 is an 8-bit register that controls the CSIBn serial transfer operation.

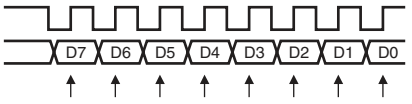
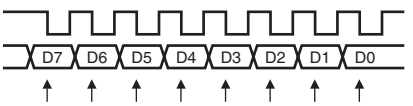
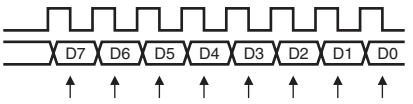
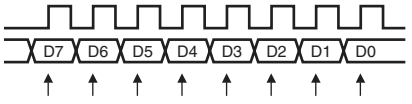
This register can be read or written in 8-bit or 1-bit units.

Reset sets this register to 00H.

Caution The CBnCTL1 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0.

After reset 00H R/W Address: CB0CTL1 FFFFFD01H, CB1CTL1 FFFFFD11H, CB2CTL1 FFFFFD21H

	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 $\overline{\text{SCKBn}}$
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 1]			Mode
			n = 0	n = 1	n = 2	
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_{BRG}[Note 2]	TAA0 (TOAA01)	$f_{XX}/128$	Master mode
1	1	1	External clock ($\overline{\text{SCKBn}}$)			Slave mode

Notes 1. Set so that communication clock (f_{CCLK}) is 8 MHz or less.

2. Baud rate generator output is also used for watch timer. For details, see **15.8 Baud Rate Generator**.

(5) CSIBn control register 2 (CBnCTL2)

CBnCTL2 is an 8-bit register that controls the number of CSIBn serial transfer bits.

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

Reset sets this register to 00H.

Caution The CBnCTL2 register can be rewritten only when the CBnCTL0.CBnPWR bit = 0 or when both the CBnTXE and CBnRXE bits = 0.

After reset: 00H R/W Address: CB0CTL2 FFFFFD02H, CB1CTL2 FFFFFD12H, CB2CTL2 FFFFFD22H

	7	6	5	4	3	2	1	0
CBnCTL2	0	0	0	0	CBnCL3	CBnCL2	CBnCL1	CBnCL0

CBnCL3	CBnCL2	CBnCL1	CBnCL0	Serial register bit length
0	0	0	0	8 bits
0	0	0	1	9 bits
0	0	1	0	10 bits
0	0	1	1	11 bits
0	1	0	0	12 bits
0	1	0	1	13 bits
0	1	1	0	14 bits
0	1	1	1	15 bits
1	×	×	×	16 bits

Remarks 1. If the number of transfer bits is other than 8 or 16, prepare and use data aligned from the LSB of the CBnTX and CBnRX registers.

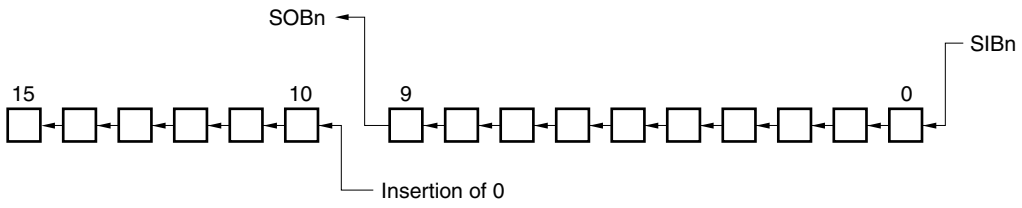
2. ×: don't care

(a) Transfer data length change function

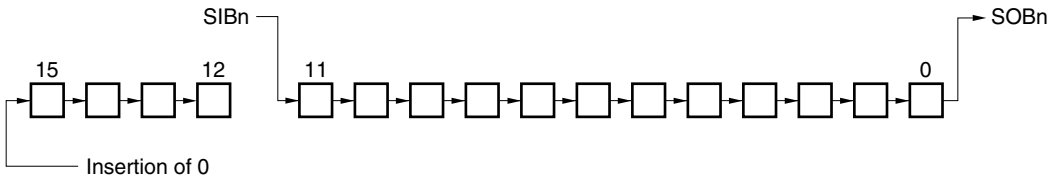
The CSIBn transfer data length can be set in 1-bit units between 8 and 16 bits using the CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits.

When the transfer bit length is set to a value other than 16 bits, set the data to the CBnTX or CBnRX register starting from the LSB, regardless of whether the transfer start bit is the MSB or LSB. Any data can be set for the higher bits that are not used, but the receive data becomes 0 following serial transfer.

(i) Transfer bit length = 10 bits, MSB first



(ii) Transfer bit length = 12 bits, LSB first



(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 FFFFFD03H, CB1STR FFFFFD13H, CB2STR FFFFFD23H

	<7>	6	5	4	3	2	1	<0>
CBnSTR	CBnTSF	0	0	0	0	0	0	CBnOVE

CBnTSF	Communication status flag
0	Communication stopped
1	Communicating

- During transmission, this register is set when data is prepared in the CBnTX register, and during reception, it is set when a dummy read of the CBnRX register is performed.
  When transfer ends, this flag is cleared to 0 at the last edge of the clock.

CBnOVE	Overrun error flag
0	No overrun
1	Overrun

- An overrun error occurs when the next reception ends without reading the value of the receive buffer by CPU, upon completion of the 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.
  - Do not check the CBnOVE flag.
  - Read this bit even if reading the reception data is not required.
- The CBnOVE flag is cleared by writing 0 to it. It cannot be set even by writing 1 to it.

15.5 Interrupt Request Signals

CSIBn can generate the following two types of 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 15-3. Interrupts and Their Default Priority

Interrupt	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.

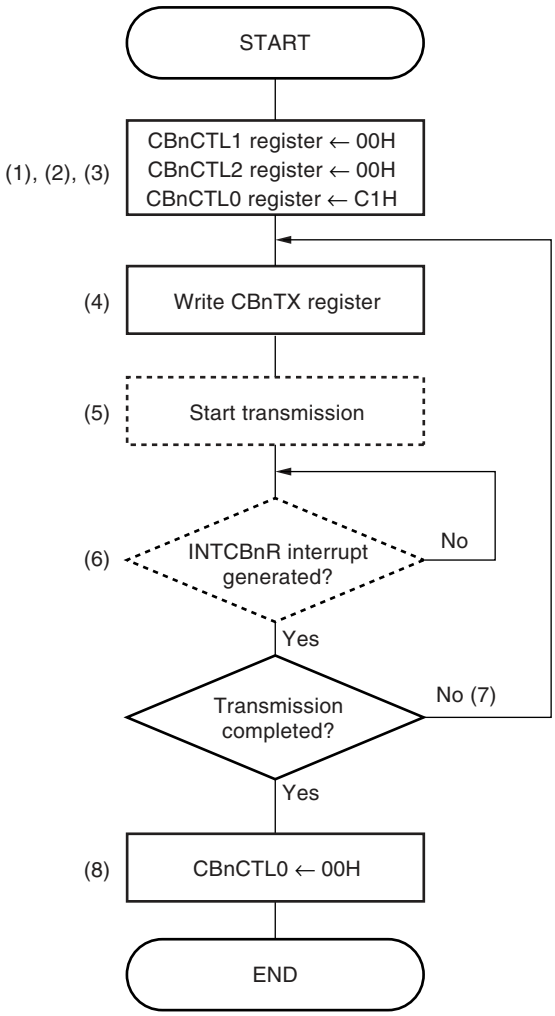
In the single transmission and single transmission/reception modes, the INTCBnT interrupt is not generated.

15.6 Operation

15.6.1 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_{CCLK}) = f_{XX}/2 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

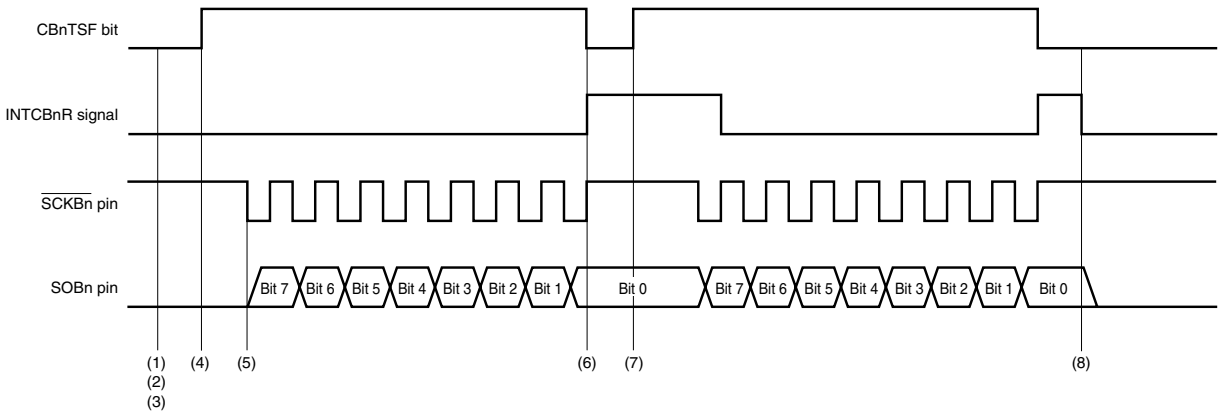
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/2, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write C1H to the CBnCTL0 register, and select the transmission mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and transmission is started.

(5) When transmission is started, output the serial clock to the $\overline{\text{SCKBn}}$ pin, and output the transmit data from the SOBn pin in synchronization with the serial clock.

(6) When transmission of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output and transmit data output, generate the reception complete interrupt request signal (INTCBnR) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

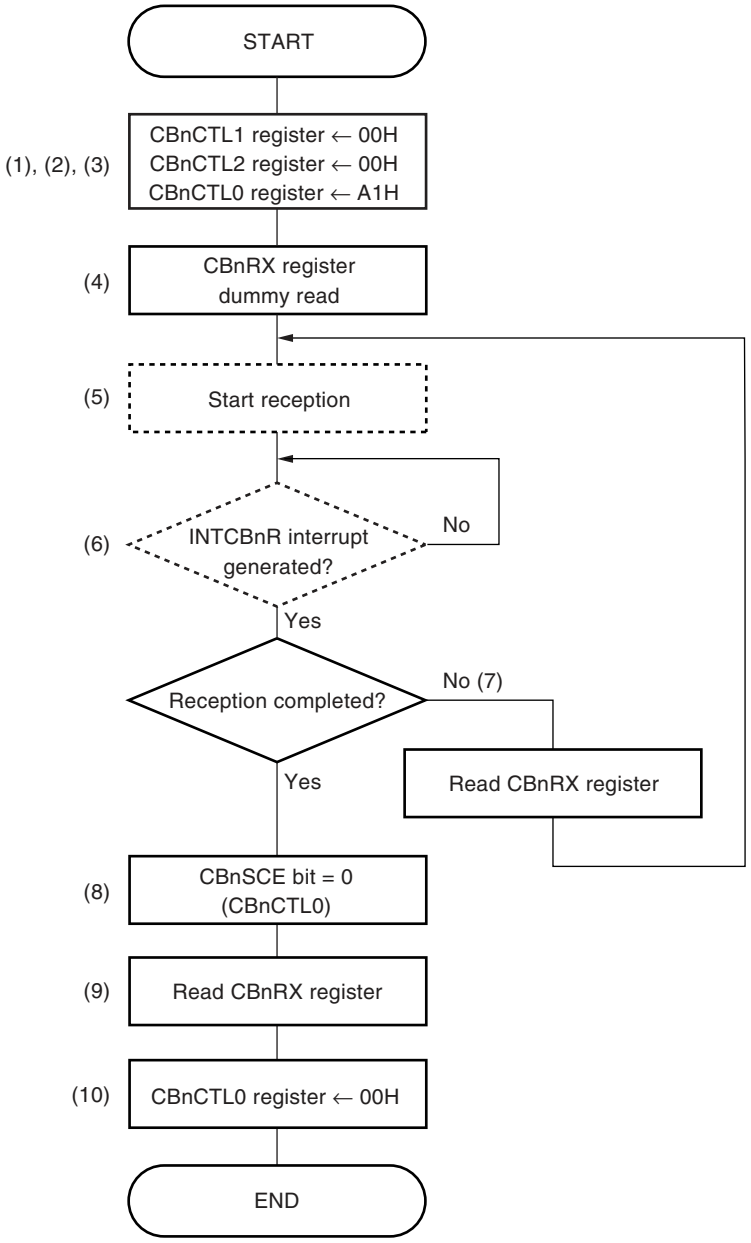
(7) To continue transmission, start the next transmission by writing the transmit data to the CBnTX register again after the INTCBnR signal is generated.

(8) To end transmission, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnTXE bit = 0.

15.6.2 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_{CCLK}) = f_{XX}/2 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

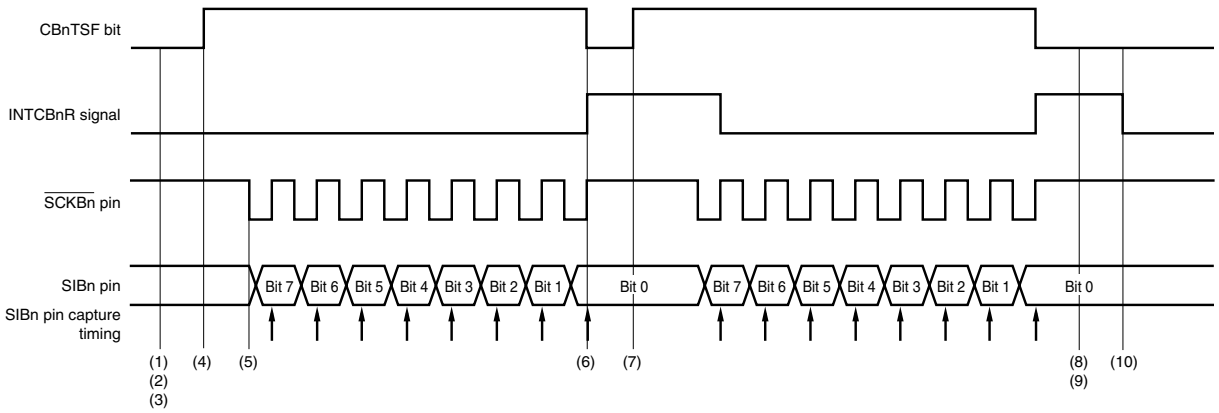
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/2, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write A1H to the CBnCTL0 register, and select the reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by performing a dummy read of the CBnRX register, and reception is started.

(5) When reception is started, output the serial clock to the $\overline{SCKBn}$ pin, and capture the receive data of the SIBn pin in synchronization with the serial clock.

(6) When reception of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output and data capturing, generate the reception complete interrupt request signal (INTCBnR) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

(7) To continue reception, read the CBnRX register with the CBnCTL0.CBnSCE bit = 1 remained after the INTCBnR signal is generated.

(8) To read the CBnRX register without starting the next reception, write the CBnSCE bit = 0.

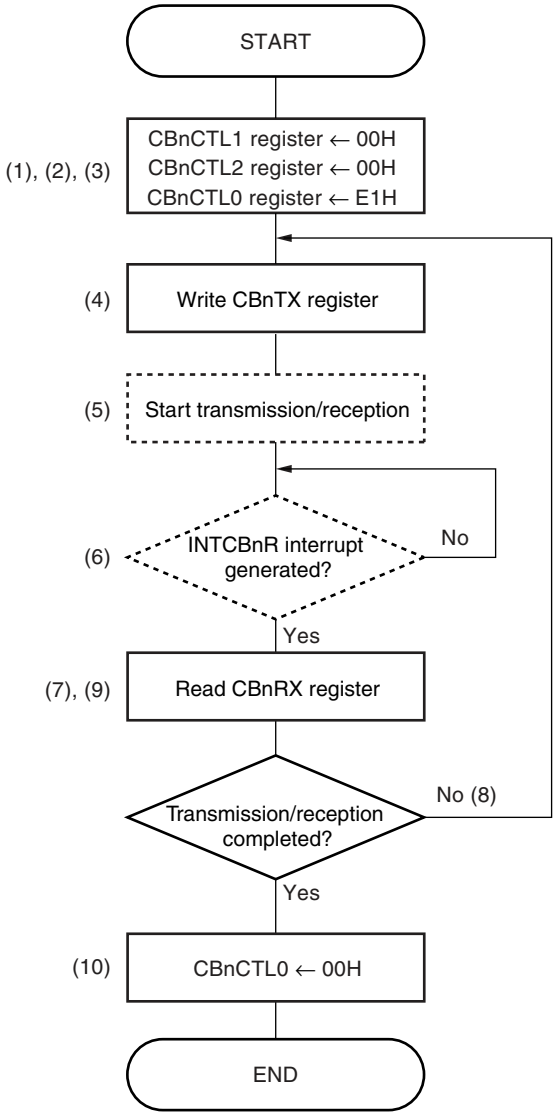
(9) Read the CBnRX register.

(10) To end reception, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnRXE bit = 0.

15.6.3 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)

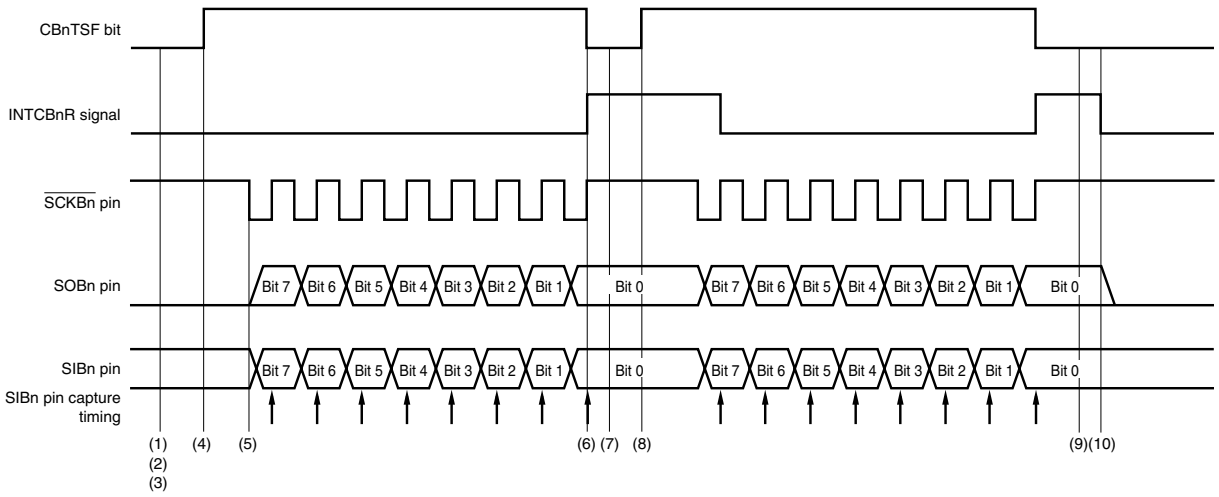
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/2, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write E1H to the CBnCTL0 register, and select the transmission/reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and transmission/reception is started.

(5) When transmission/reception is started, output the serial clock to the $\overline{SCKBn}$ pin, output the transmit data to the SOBn pin in synchronization with the serial clock, and capture the receive data of the SIBn pin.

(6) When transmission/reception of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock output, transmit data output, and data capturing, generate the reception complete interrupt request signal (INTCBnR) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

(7) Read the CBnRX register.

(8) To continue transmission/reception, write the transmit data to the CBnTX register again.

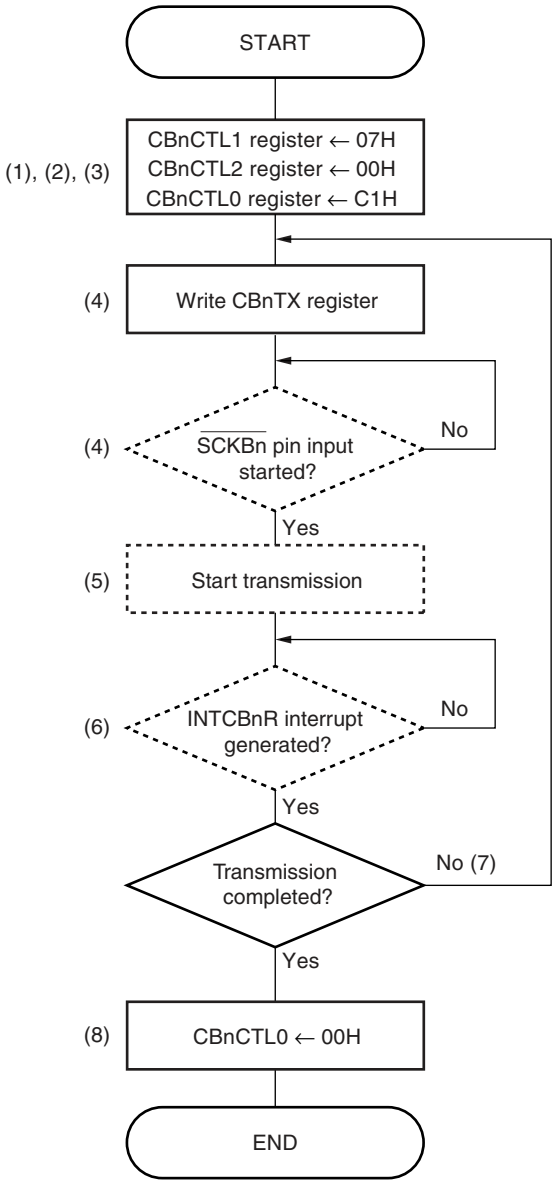
(9) Read the CBnRX register.

(10) To end transmission/reception, write the CBnCTL0.CBnPWR bit = 0, the CBnCTL0.CBnTXE bit = 0, and the CBnCTL0.CBnRXE bit = 0.

15.6.4 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_{CCLK}) = external clock ($\overline{SCKBn}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

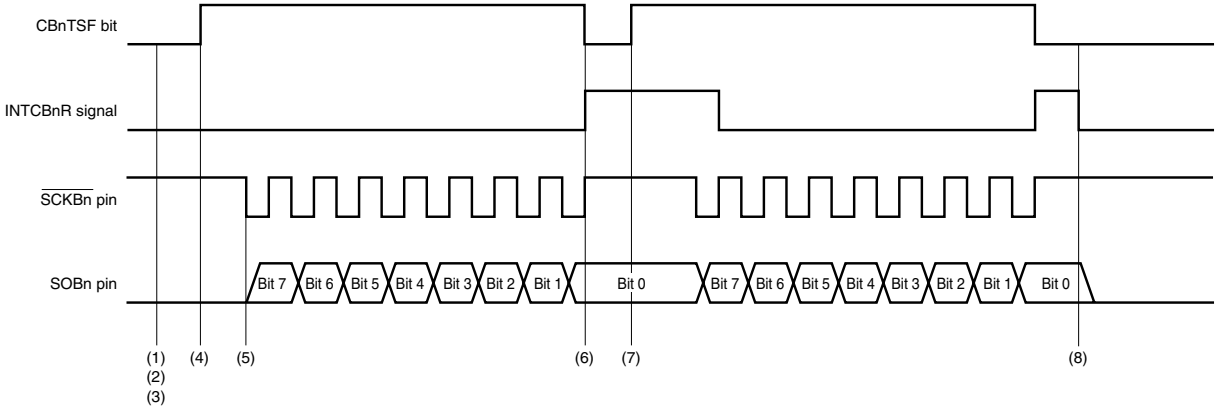
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$), and slave mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write C1H to the CBnCTL0 register, and select the transmission mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data from the SOBn pin in synchronization with the serial clock.

(6) When transmission of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock input and transmit data output, generate the reception complete interrupt request signal (INTCBnR) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

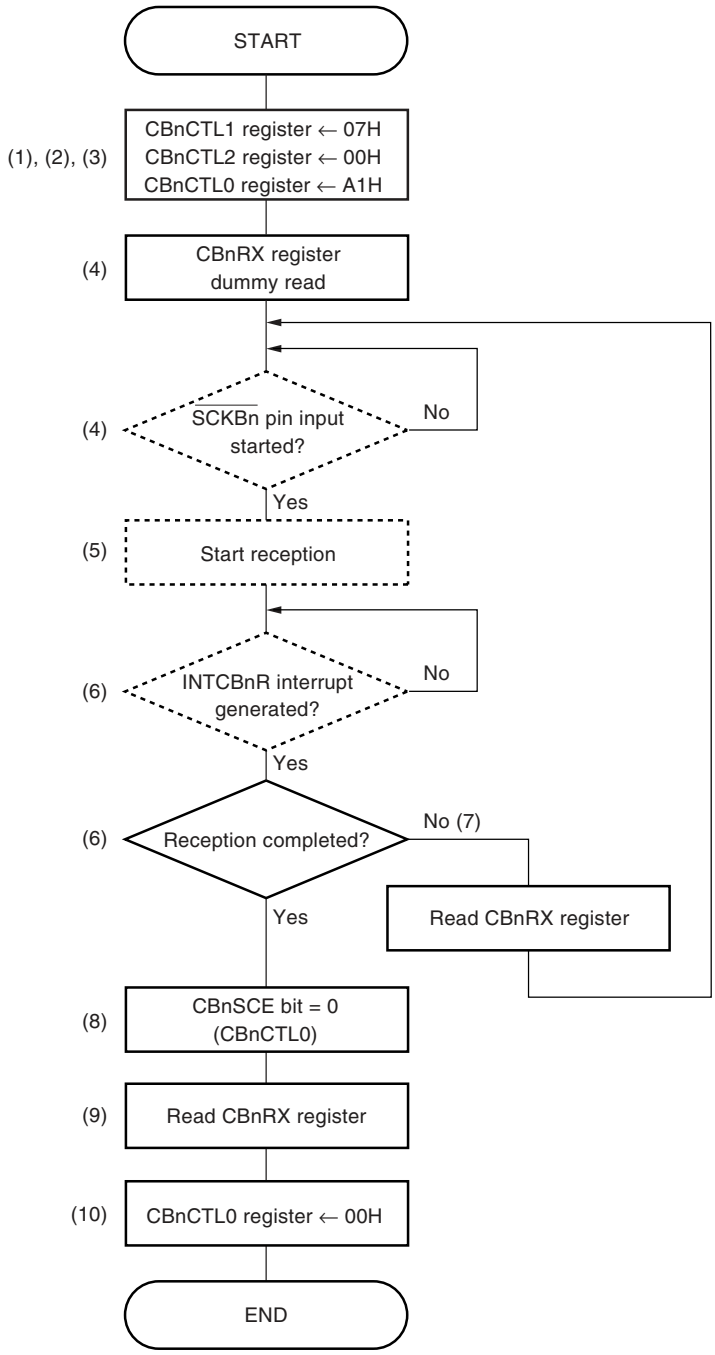
(7) To continue transmission, write the transmit data to the CBnTX register again after the INTCBnR signal is generated, and wait for a serial clock input.

(8) To end transmission, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnTXE bit = 0.

15.6.5 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_{CCLK}) = external clock ($\overline{SCKBn}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

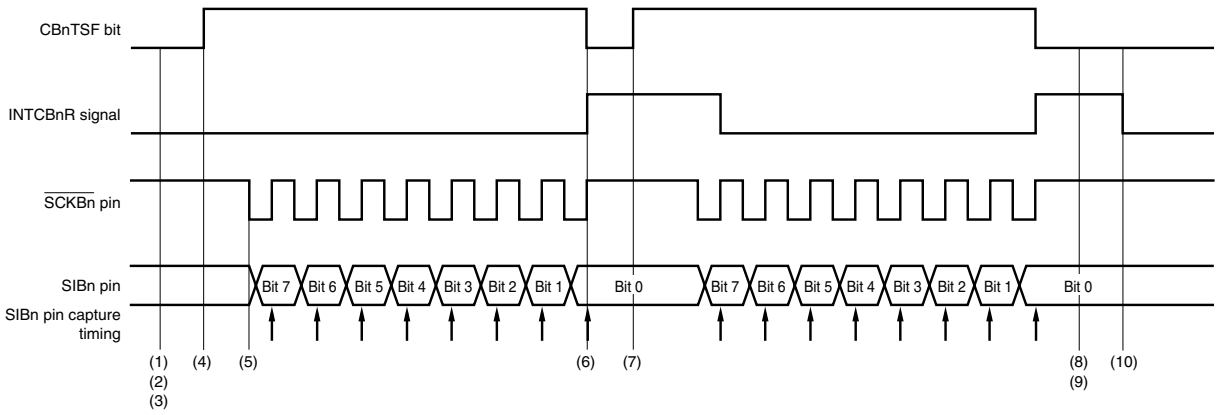
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$), and slave mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write A1H to the CBnCTL0 register, and select the reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by performing a dummy read of the CBnRX register, and the device waits for a serial clock input.

(5) When a serial clock is input, capture the receive data of the SIBn pin in synchronization with the serial clock.

(6) When reception of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock input and data capturing, generate the reception complete interrupt request signal (INTCBnR) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

(7) To continue reception, read the CBnRX register with the CBnCTL0.CBnSCE bit = 1 remained after the INTCBnR signal is generated, and wait for a serial clock input.

(8) To end reception, write the CBnSCE bit = 0.

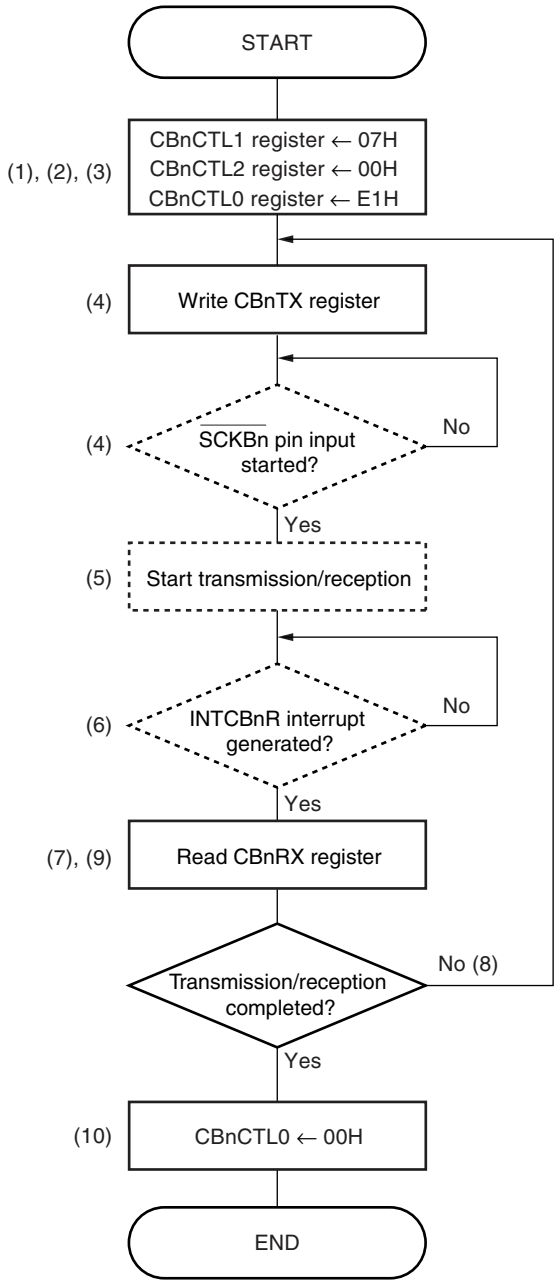
(9) Read the CBnRX register.

(10) To end reception, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnRXE bit = 0.

15.6.6 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_{CCLK}) = external clock ($\overline{SCKBn}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

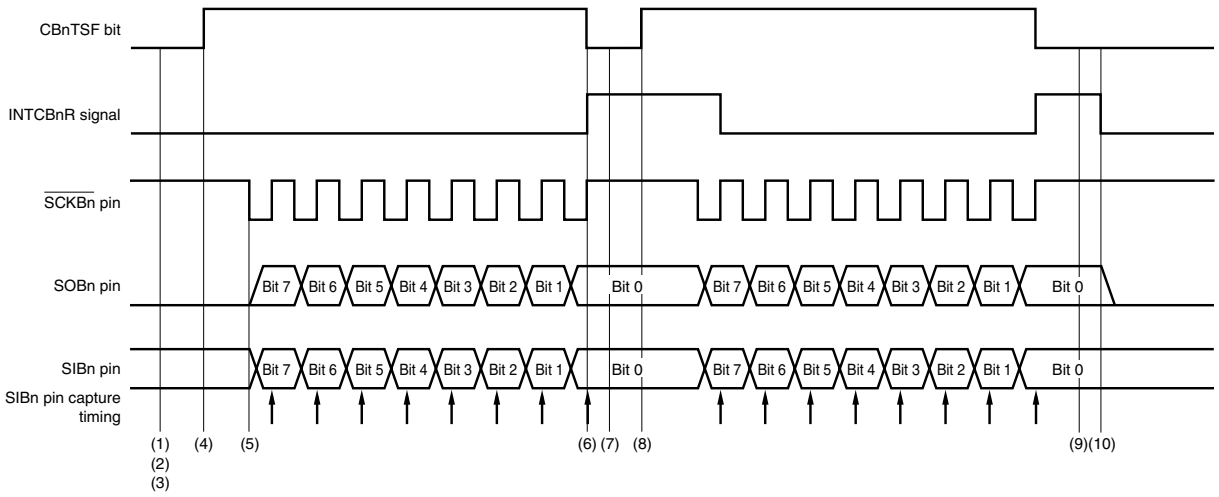
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$), and slave mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write E1H to the CBnCTL0 register, and select the transmission/reception mode and MSB first at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data to the SOBn pin in synchronization with the serial clock, and capture the receive data of the SIBn pin.

(6) When transmission/reception of the transfer data length set with the CBnCTL2 register is completed, stop the serial clock input, transmit data output, and data capturing, generate the reception complete interrupt request signal (INTCBnR) at the last edge of the serial clock, and clear the CBnTSF bit to 0.

(7) Read the CBnRX register.

(8) To continue transmission/reception, write the transmit data to the CBnTX register again, and wait for a serial clock input.

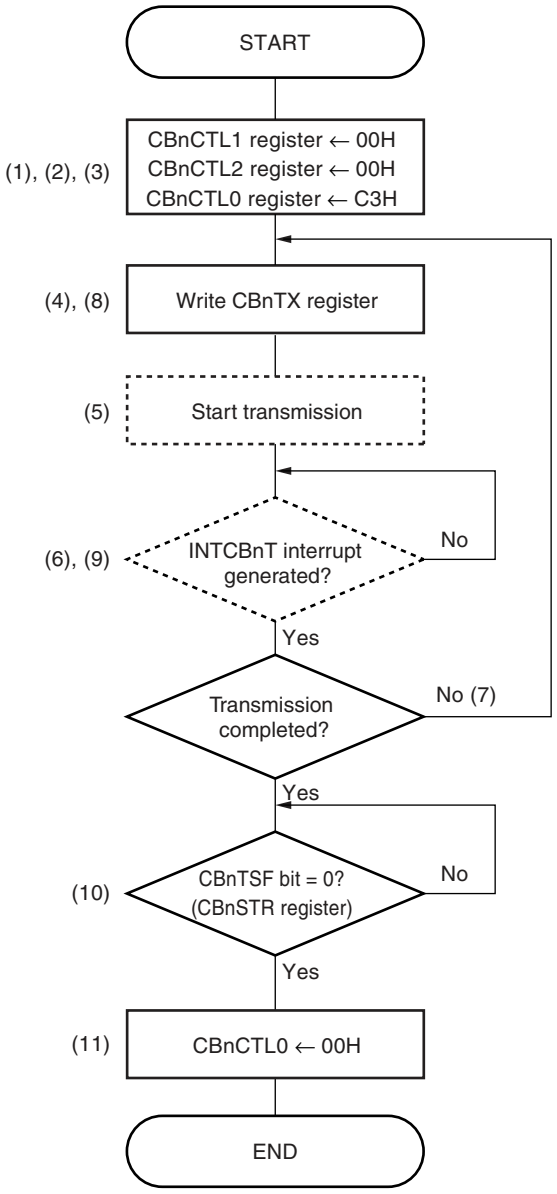
(9) Read the CBnRX register.

(10) To end transmission/reception, write the CBnCTL0.CBnPWR bit = 0, the CBnCTL0.CBnTXE bit = 0, and the CBnCTL0.CBnRXE bit = 0.

15.6.7 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)

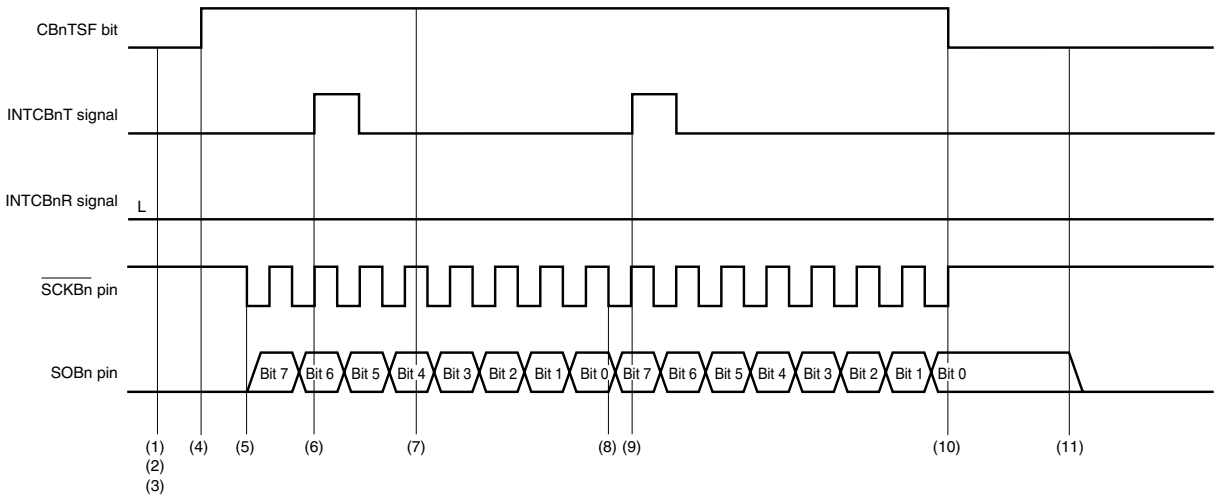
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/2, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write C3H to the CBnCTL0 register, and select the transmission mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and transmission is started.

(5) When transmission is started, output the serial clock to the $\overline{\text{SCKBn}}$ pin, and output the transmit data from the SOBn pin in synchronization with the serial clock.

(6) When transfer of the transmit data from the CBnTX register to the shift register is completed and writing to the CBnTX register is enabled, the transmission enable interrupt request signal (INTCBnT) is generated.

(7) To continue transmission, write the transmit data to the CBnTX register again after the INTCBnT signal is generated.

(8) When a new transmit data is written to the CBnTX register before communication completion, the next communication is started following communication completion.

(9) The transfer of the transmit data from the CBnTX register to the shift register is completed and the INTCBnT signal is generated. To end continuous transmission with the current transmission, do not write to the CBnTX register.

(10) When the next transmit data is not written to the CBnTX register before transfer completion, stop the serial clock output to the $\overline{\text{SCKBn}}$ pin after transfer completion, and clear the CBnTSF bit to 0.

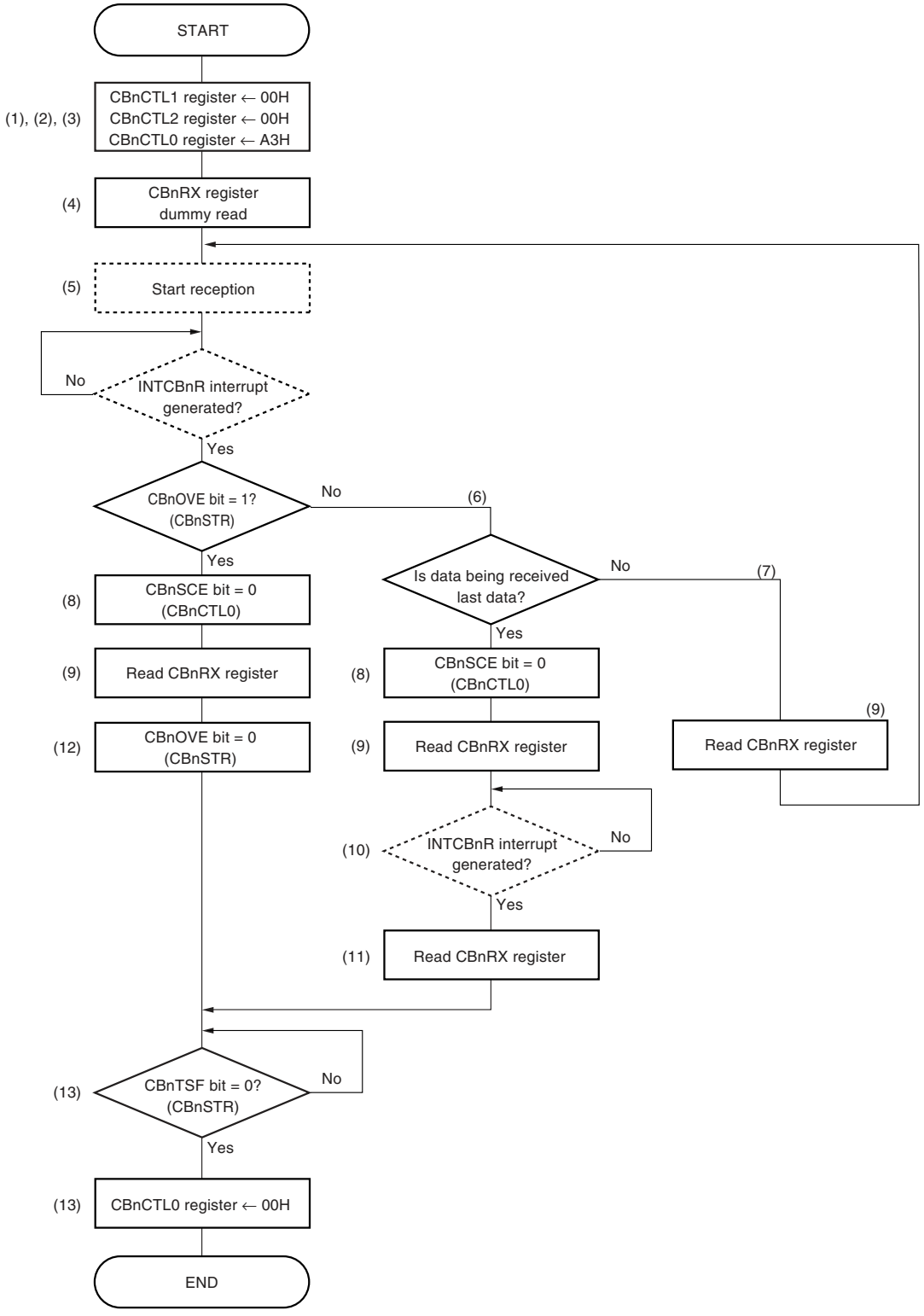
(11) To release the transmission enable status, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnTXE bit = 0 after checking that the CBnTSF bit = 0.

Caution In continuous transmission mode, the reception complete interrupt request signal (INTCBnR) is not generated.

15.6.8 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}) = $f_{XX}/2$ (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

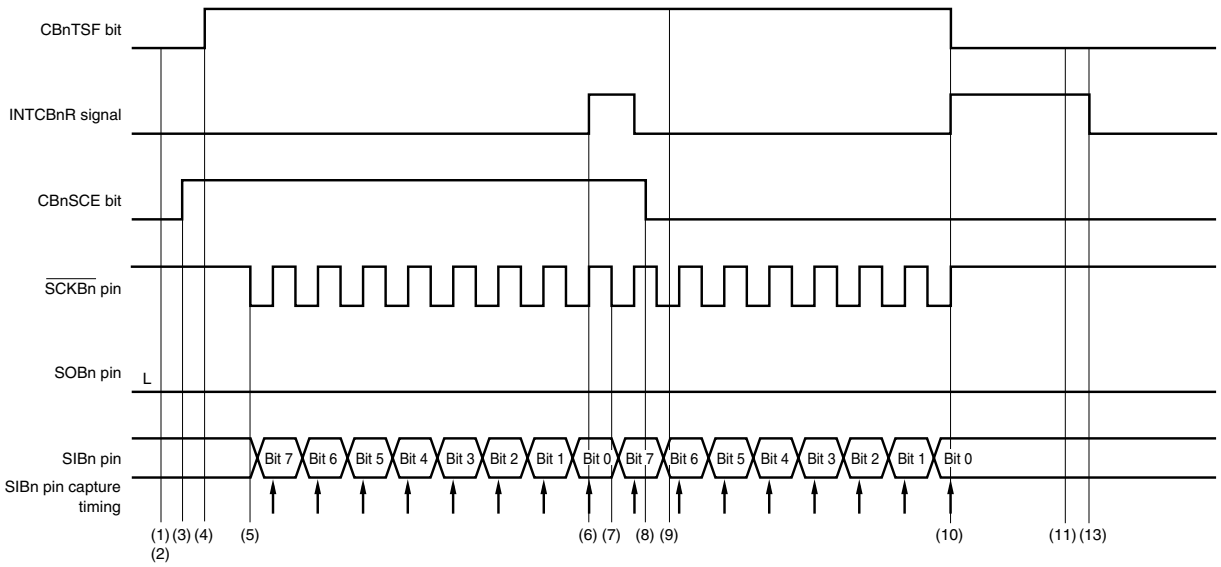
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/2, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write A3H to the CBnCTL0 register, and select the reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by performing a dummy read of the CBnRX register, and reception is started.

(5) When reception is started, output the serial clock to the $\overline{\text{SCKBn}}$ pin, and capture the receive data of the SIBn pin in synchronization with the serial clock.

(6) When reception is completed, the reception complete interrupt request signal (INTCBnR) is generated, and reading of the CBnRX register is enabled.

(7) When the CBnCTL0.CBnSCE bit = 1 upon communication completion, the next communication is started following communication completion.

(8) To end continuous reception with the current reception, write the CBnSCE bit = 0.

(9) Read the CBnRX register.

(10) When reception is completed, the INTCBnR signal is generated, and reading of the CBnRX register is enabled. When the CBnSCE bit = 0 is set before communication completion, stop the serial clock output to the $\overline{\text{SCKBn}}$ pin, and clear the CBnTSF bit to 0, to end the receive operation.

(11) Read the CBnRX register.

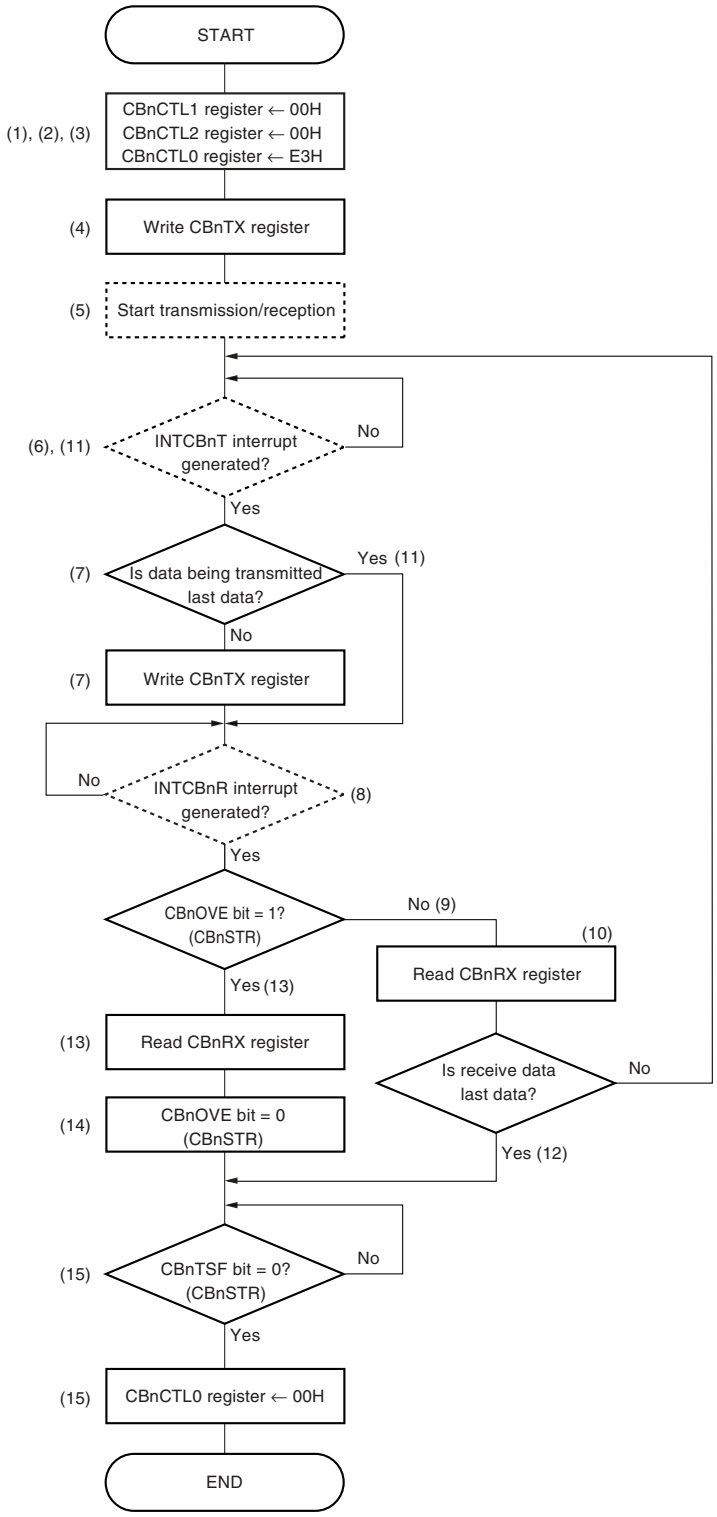
(12) If an overrun error occurs, write the CBnSTR.CBnOVE bit = 0, and clear the error flag.

(13) To release the reception enable status, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnRXE bit = 0 after checking that the CBnTSF bit = 0.

15.6.9 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_{CCLK}) = f_{XX}/2 (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 000), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

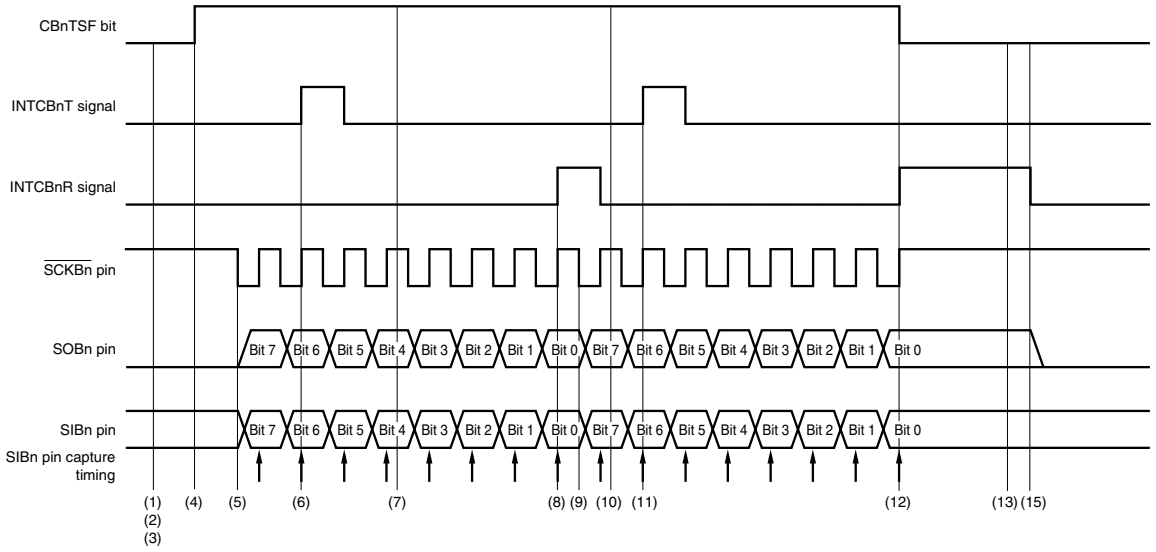
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 00H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = f_{XX}/2, and master mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write E3H to the CBnCTL0 register, and select the transmission/reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and transmission/reception is started.

(5) When transmission/reception is started, output the serial clock to the $\overline{\text{SCKBn}}$ pin, output the transmit data to the SOBn pin in synchronization with the serial clock, and capture the receive data of the SIBn pin.

(6) When transfer of the transmit data from the CBnTX register to the shift register is completed and writing to the CBnTX register is enabled, the transmission enable interrupt request signal (INTCBnT) is generated.

(7) To continue transmission/reception, write the transmit data to the CBnTX register again after the INTCBnT signal is generated.

(8) When one transmission/reception is completed, the reception complete interrupt request signal (INTCBnR) is generated, and reading of the CBnRX register is enabled.

(9) When a new transmit data is written to the CBnTX register before communication completion, the next communication is started following communication completion.

(10) Read the CBnRX register.

(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) When the next transmit data is not written to the CBnTX register before transfer completion, stop the serial clock output to the $\overline{\text{SCKBn}}$ pin after transfer completion, and clear the CBnTSF bit to 0.

(13) When the reception error interrupt request signal (INTCBnR) is generated, read the CBnRX register.

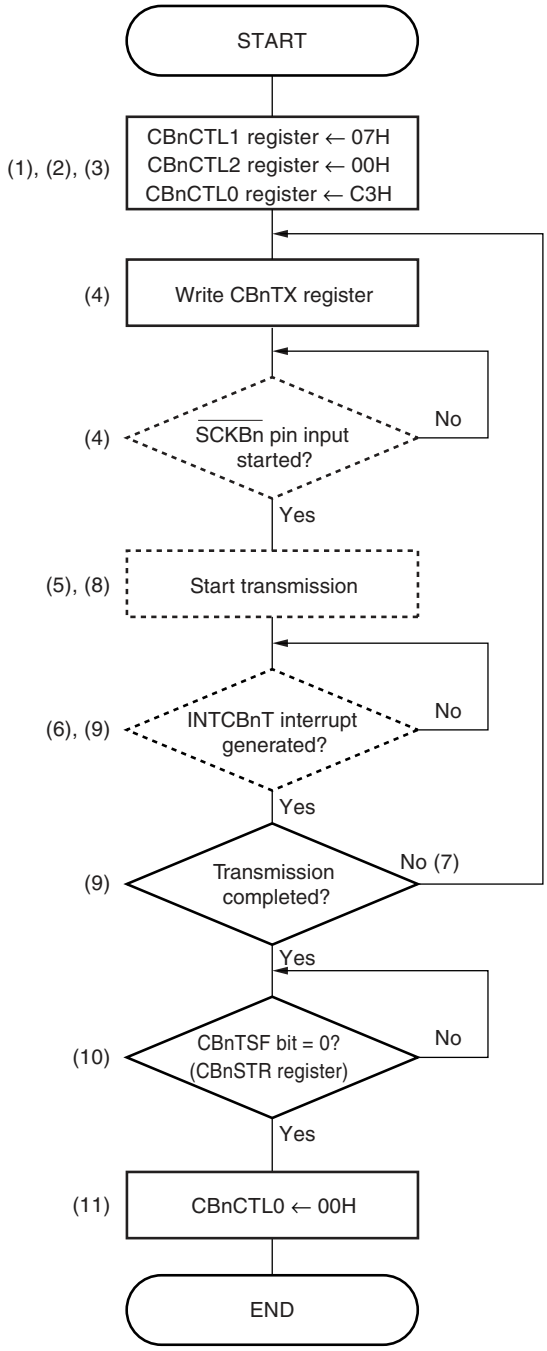
(14) If an overrun error occurs, write the CBnSTR.CBnOVE bit = 0, and clear the error flag.

(15) To release the transmission/reception enable status, write the CBnCTL0.CBnPWR bit = 0, the CBnCTL0.CBnTXE bit = 0, and the CBnCTL0.CBnRXE bit = 0 after checking that the CBnTSF bit = 0.

15.6.10 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_{CCLK}) = external clock ($\overline{SCKBn}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

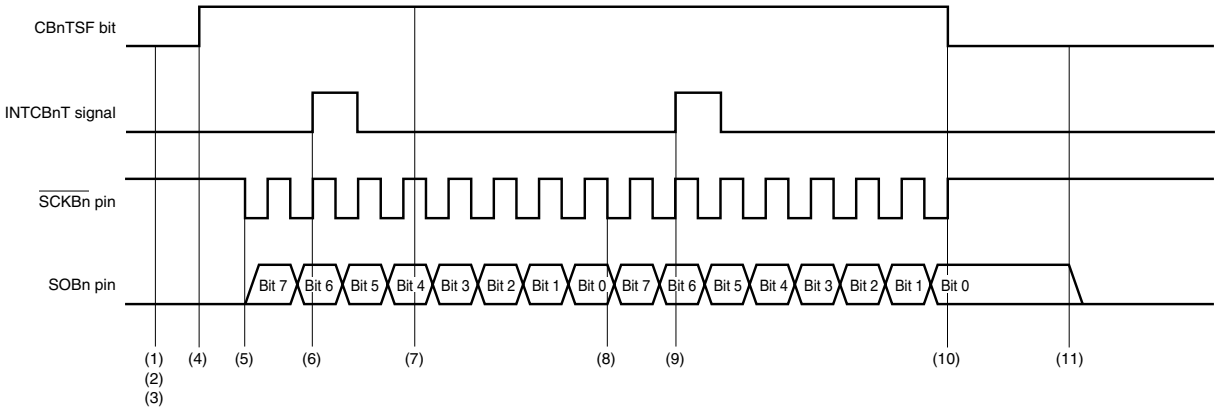
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$), and slave mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write C3H to the CBnCTL0 register, and select the transmission mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data from the SOBn pin in synchronization with the serial clock.

(6) When transfer of the transmit data from the CBnTX register to the shift register is completed and writing to the CBnTX register is enabled, the transmission enable interrupt request signal (INTCBnT) is generated.

(7) To continue transmission, write the transmit data to the CBnTX register again after the INTCBnT signal is generated.

(8) When a serial clock is input following completion of the transmission of the transfer data length set with the CBnCTL2 register, continuous transmission is started.

(9) When transfer of the transmit data from the CBnTX register to the shift register is completed and writing to the CBnTX register is enabled, the INTCBnT signal is generated. To end continuous transmission with the current transmission, do not write to the CBnTX register.

(10) When the clock of the transfer data length set with the CBnCTL2 register is input without writing to the CBnTX register, clear the CBnTSF bit to 0 to end transmission.

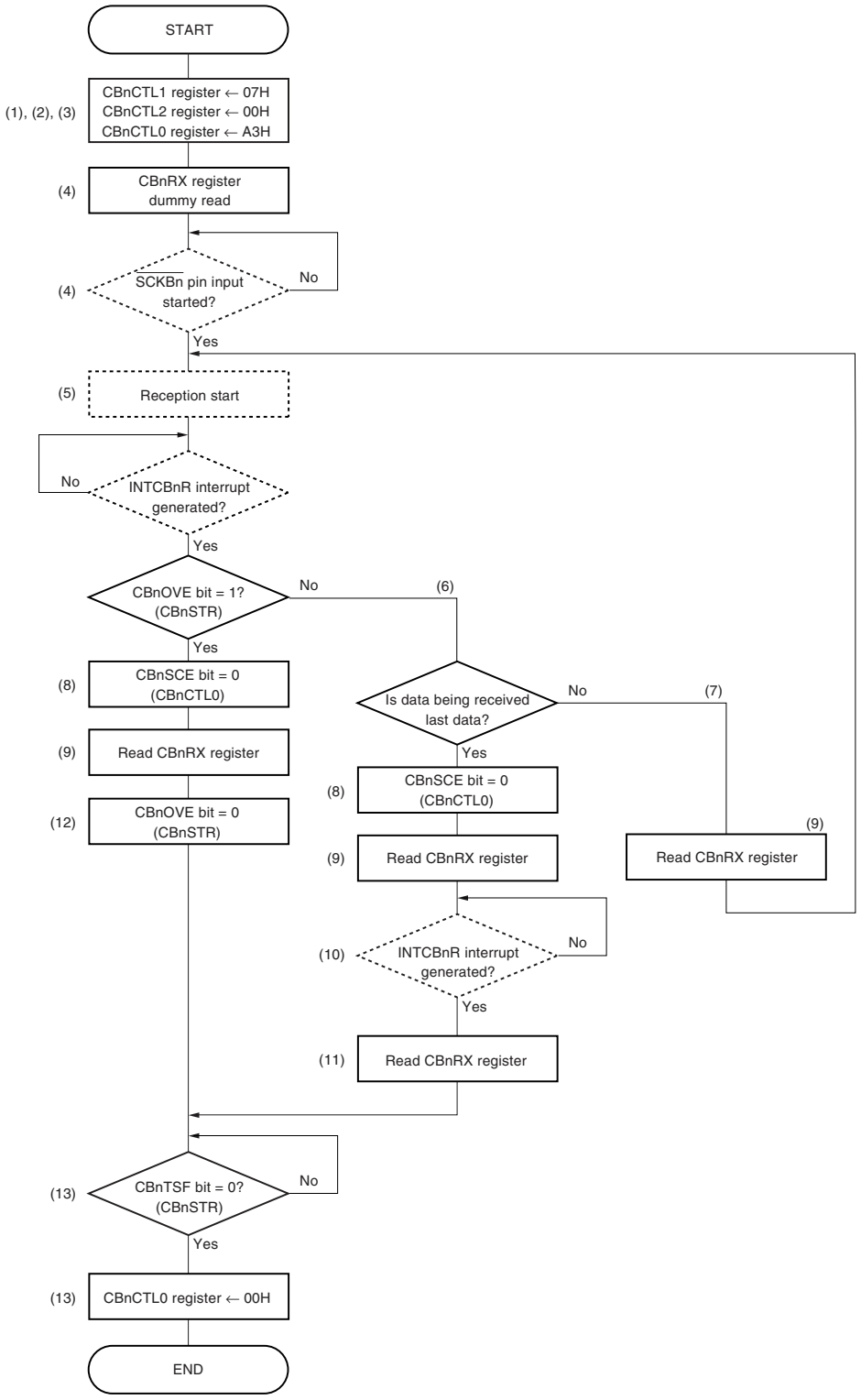
(11) To release the transmission enable status, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnTXE bit = 0 after checking that the CBnTSF bit = 0.

Caution In continuous transmission mode, the reception complete interrupt request signal (INTCBnR) is not generated.

15.6.11 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_{CCLK}) = external clock ($\overline{SCKBn}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

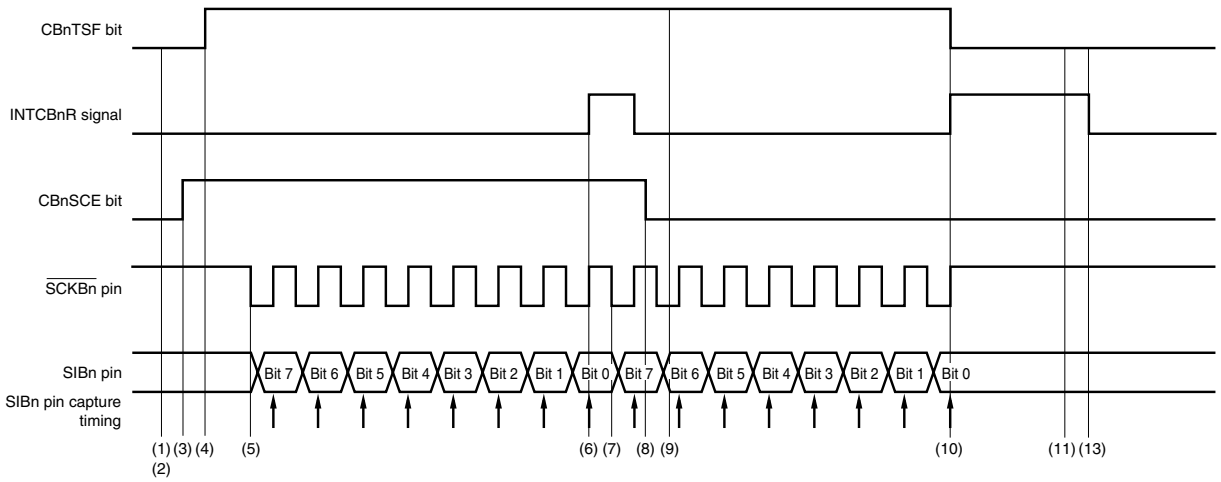
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$), and slave mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write A3H to the CBnCTL0 register, and select the reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by performing a dummy read of the CBnRX register, and the device waits for a serial clock input.

(5) When a serial clock is input, capture the receive data of the SIBn pin in synchronization with the serial clock.

(6) When reception is completed, the reception complete interrupt request signal (INTCBnR) is generated, and reading of the CBnRX register is enabled.

(7) When a serial clock is input in the CBnCTL0.CBnSCE bit = 1 status, continuous reception is started.

(8) To end continuous reception with the current reception, write the CBnSCE bit = 0.

(9) Read the CBnRX register.

(10) When reception is completed, the INTCBnR signal is generated, and reading of the CBnRX register is enabled. When the CBnSCE bit = 0 is set before communication completion, clear the CBnTSF bit to 0 to end the receive operation.

(11) Read the CBnRX register.

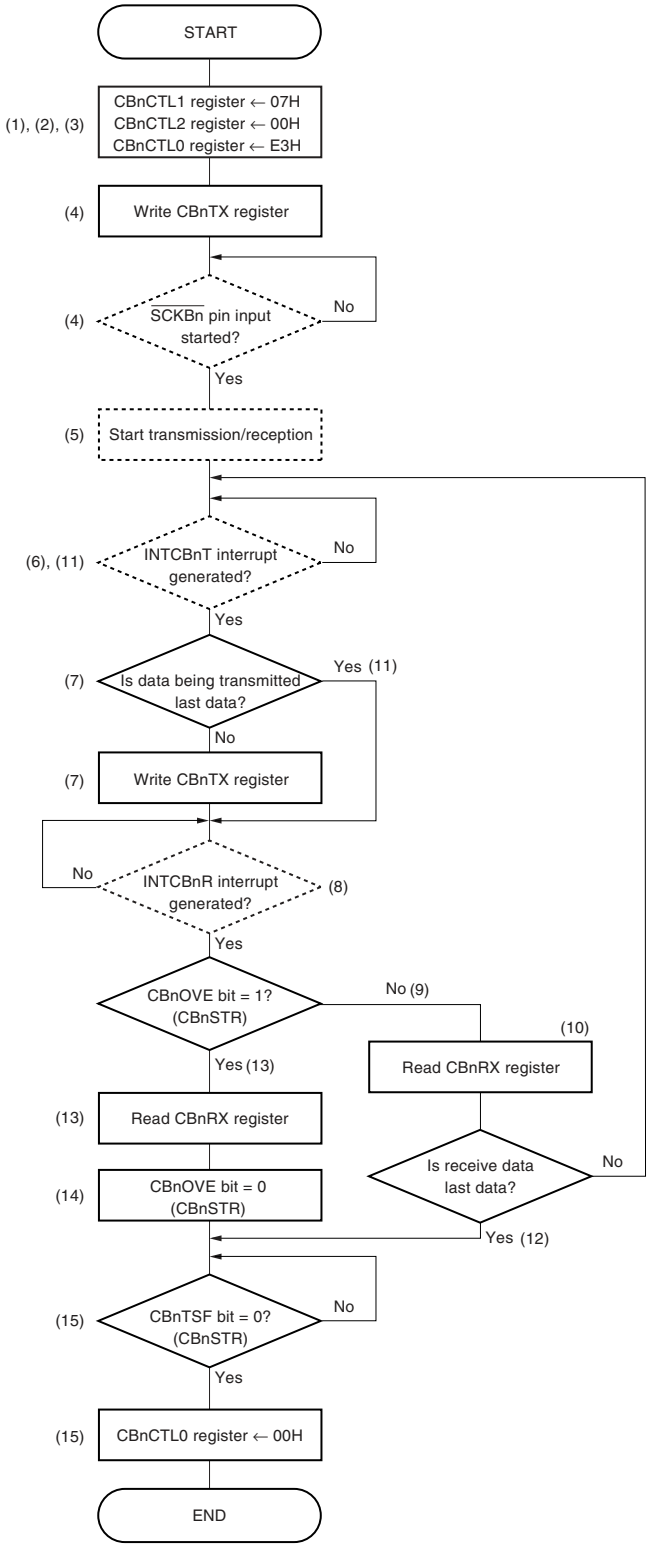
(12) If an overrun error occurs, write the CBnSTR.CBnOVE bit = 0, and clear the error flag.

(13) To release the reception enable status, write the CBnCTL0.CBnPWR bit = 0 and the CBnCTL0.CBnRXE bit = 0 after checking that the CBnTSF bit = 0.

15.6.12 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_{CCLK}) = external clock ($\overline{SCKBn}$) (CBnCTL1.CBnCKS2 to CBnCTL1.CBnCKS0 bits = 111), transfer data length = 8 bits (CBnCTL2.CBnCL3 to CBnCTL2.CBnCL0 bits = 0000)

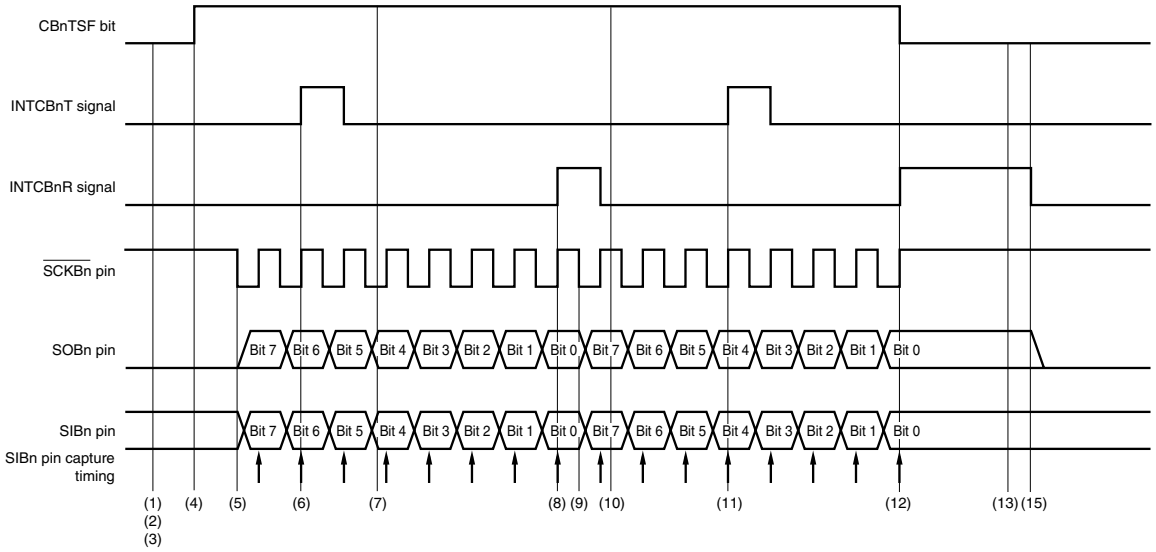
(1) Operation flow



Remarks 1. The broken lines indicate the hardware processing.

2. The numbers in this figure correspond to the processing numbers in **(2) Operation timing**.

(2) Operation timing



(1) Write 07H to the CBnCTL1 register, and select communication type 1, communication clock (f_{CCLK}) = external clock ($\overline{SCKBn}$), and slave mode.

(2) Write 00H to the CBnCTL2 register, and set the transfer data length to 8 bits.

(3) Write E3H to the CBnCTL0 register, and select the transmission/reception mode, MSB first, and continuous transfer mode at the same time as enabling the operation of the communication clock (f_{CCLK}).

(4) The CBnSTR.CBnTSF bit is set to 1 by writing the transmit data to the CBnTX register, and the device waits for a serial clock input.

(5) When a serial clock is input, output the transmit data to the SOBn pin in synchronization with the serial clock, and capture the receive data of the SIBn pin.

(6) When transfer of the transmit data from the CBnTX register to the shift register is completed and writing to the CBnTX register is enabled, the transmission enable interrupt request signal (INTCBnT) is generated.

(7) To continue transmission, write the transmit data to the CBnTX register again after the INTCBnT signal is generated.

(8) When reception of the transfer data length set with the CBnCTL2 register is completed, the reception complete interrupt request signal (INTCBnR) is generated, and reading of the CBnRX register is enabled.

(9) When a serial clock is input continuously, continuous transmission/reception is started.

(10) Read the CBnRX register.

(11) When transfer of the transmit data from the CBnTX register to the shift register is completed and writing to the CBnTX register is enabled, the INTCBnT signal is generated. To end continuous transmission/reception with the current transmission/reception, do not write to the CBnTX register.

(12) When the clock of the transfer data length set with the CBnCTL2 register is input without writing to the CBnTX register, the INTCBnR signal is generated. Clear the CBnTSF bit to 0 to end transmission/reception.

(13) When the INTCBnR signal is generated, read the CBnRX register.

(14) If an overrun error occurs, write the CBnSTR.CBnOVE bit = 0, and clear the error flag.

(15) To release the transmission/reception enable status, write the CBnCTL0.CBnPWR bit = 0, the CBnCTL0.CBnTXE bit = 0, and the CBnCTL0.CBnRXE bit = 0 after checking that the CBnTSF bit = 0.

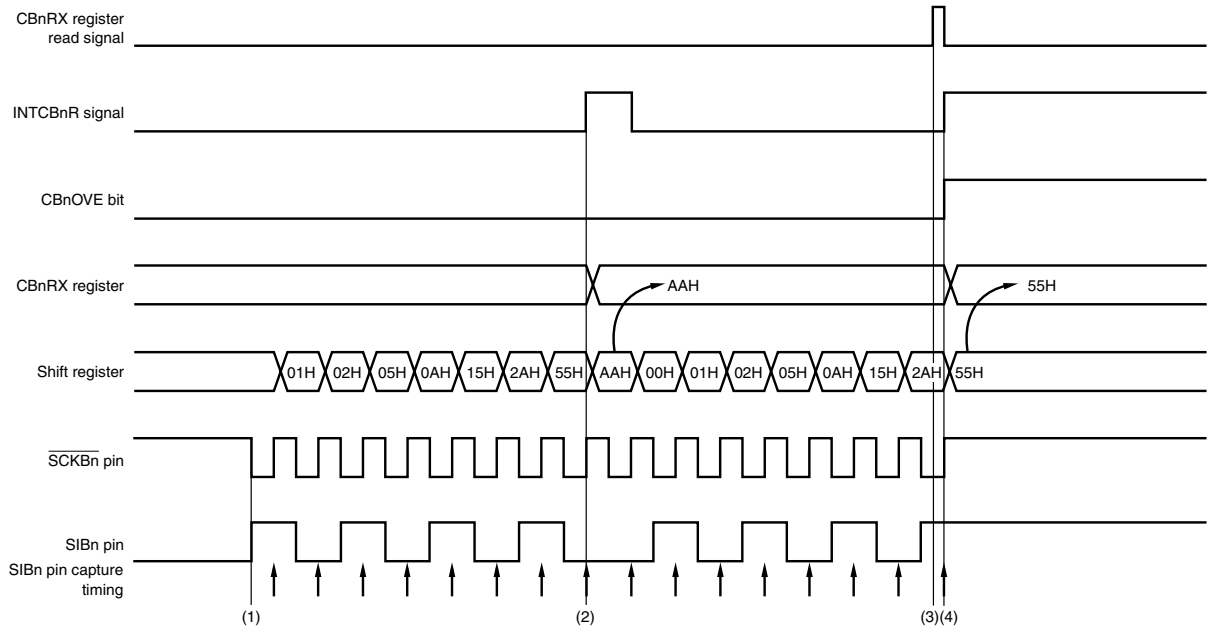
15.6.13 Reception error

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 when 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.

Even if an overrun error has occurred, the previous receive data is lost since the CBnRX register is updated. Even if a reception error has occurred, the INTCBnR signal is generated again upon the next reception completion if the CBnRX register is not read.

To avoid an overrun error, complete reading the CBnRX register until one half clock before sampling the last bit of the next receive data from the INTCBnR signal generation.

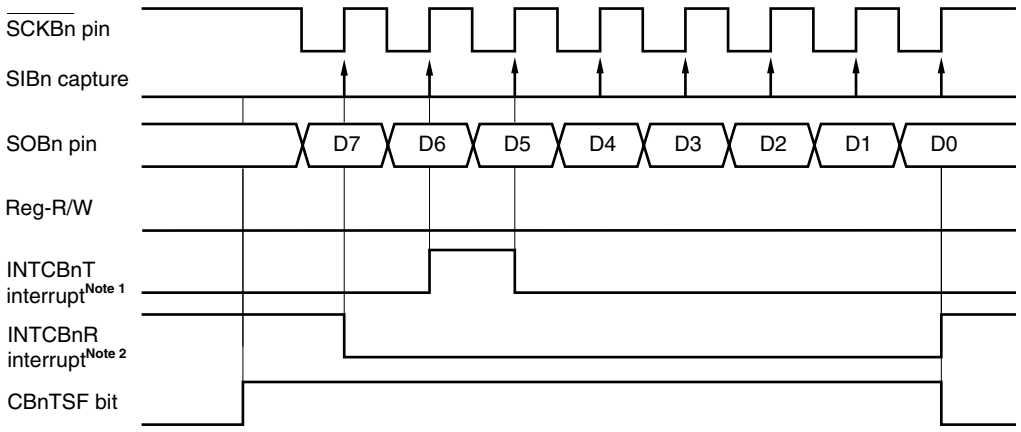
(1) Operation timing



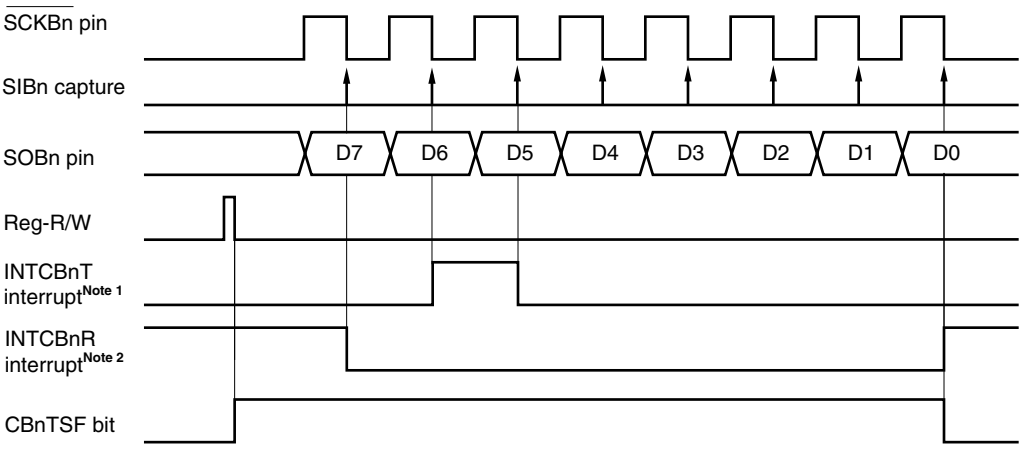
(1) Start continuous transfer.
(2) Completion of the first transfer
(3) The CBnRX register cannot be read until one half clock before the completion of the second transfer.
(4) An overrun error occurs, and the reception complete interrupt request signal (INTCBnR) is generated, and then the overrun error flag (CBnSTR.CBnOVE) is set to 1. The receive data is overwritten.

15.6.14 Clock timing

(i) Communication type 1 (CBnCKP and CBnDAP bits = 00)



(ii) Communication type 3 (CBnCKP and CBnDAP bits = 10)



Notes 1. The INTCBnT interrupt is set when the data written to the CBnTX register is transferred to the data shift register in the continuous transmission or continuous transmission/reception mode. In the single transmission or single transmission/reception mode, the INTCBnT interrupt request signal is not generated, but the INTCBnR interrupt request signal is generated upon end of communication.

2. The INTCBnR interrupt occurs if reception is correctly ended and receive data is ready in the CBnRX register while reception is enabled. In the single mode, the INTCBnR interrupt request signal is generated even in the transmission mode, upon end of communication.

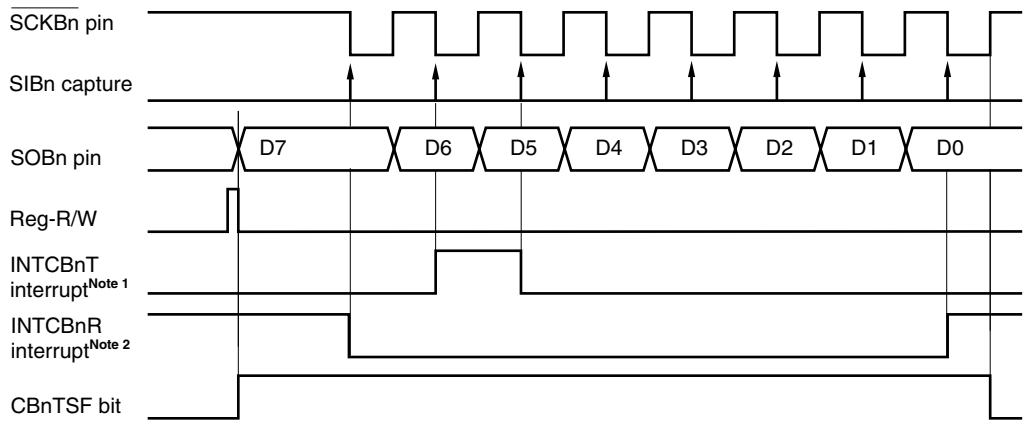
Caution In single transfer mode, writing to the CBnTX register with the CBnTSF bit set to 1 is ignored. This has no influence on the operation during transfer.

For example, if the next data is written to the CBnTX register when DMA is started by generating the INTCBnR signal, the written data is not transferred because the CBnTSF bit is set to 1.

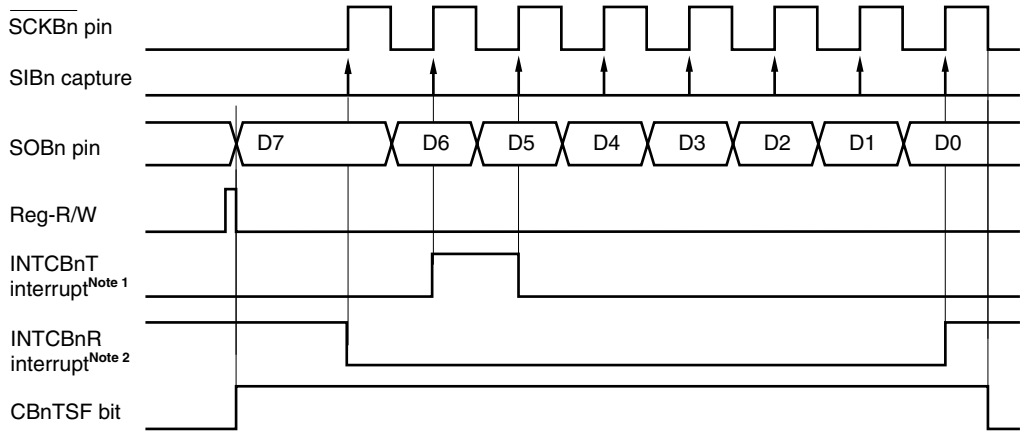
Use the continuous transfer mode, not the single transfer mode, for such applications.

(2/2)

(iii) Communication type 2 (CBnCKP and CBnDAP bits = 01)



(iv) Communication type 4 (CBnCKP and CBnDAP bits = 11)



Notes 1. The INTCBnT interrupt is set when the data written to the CBnTX register is transferred to the data shift register in the continuous transmission or continuous transmission/reception modes. In the single transmission or single transmission/reception modes, the INTCBnT interrupt request signal is not generated, but the INTCBnR interrupt request signal is generated upon end of communication.

2. The INTCBnR interrupt occurs if reception is correctly ended and receive data is ready in the CBnRX register while reception is enabled. In the single mode, the INTCBnR interrupt request signal is generated even in the transmission mode, upon end of communication.

Caution In single transfer mode, writing to the CBnTX register with the CBnTSF bit set to 1 is ignored. This has no influence on the operation during transfer.

For example, if the next data is written to the CBnTX register when DMA is started by generating the INTCBnR signal, the written data is not transferred because the CBnTSF bit is set to 1.

Use the continuous transfer mode, not the single transfer mode, for such applications.

15.7 Output Pin Status with Operation Disabled

(1) $\overline{\text{SCKBn}}$ pin

When CSIBn operation is disabled (CBnCTL0.CBnPWR bit = 0), the $\overline{\text{SCKBn}}$ pin output status is as follows.

CBnCKS2	CBnCKS1	CBnCKS0	CBnCKP	$\overline{\text{SCKBn}}$ Pin Output
1	1	1	×	High impedance
Other than above			0	Fixed to high level
			1	Fixed to low level

Remarks 1. The output level of the $\overline{\text{SCKBn}}$ pin changes if any of the CBnCTL1.CBnCKP and CBnCKS2 to CBnCKS0 bits is rewritten.

2. ×: don't care

(2) SOBn pin

When CSIBn operation is disabled (CBnPWR bit = 0), the SOBn pin output status is as follows.

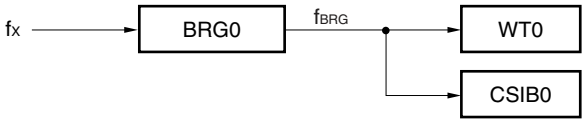
CBnTXE	CBnDAP	CBnDIR	SOBn Pin Output
0	×	×	Fixed to low level
1	0	×	SOBn latch value (low level)
	1	0	CBnTX register value (MSB)
		1	CBnTX register value (LSB)

Remarks 1. The SOBn pin output changes when any one of the CBnCTL0.CBnTXE, CBnCTL0.CBnDIR bits, and CBnCTL1.CBnDAP bit is rewritten.

2. ×: don't care

15.8 Baud Rate Generator

The clock generated by the baud rate generator (prescaler 3) is supplied to the watch timer and CSIB0.



(1) Prescaler mode register 0 (PRSM0)

The PRSM0 register controls generation of the baud rate signal for CSIB.

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	Baud rate output
0	Disabled
1	Enabled

BGCS01	BGCS00	Watch timer source clock (f_{BGCS}) selection		
			10 MHz	8 MHz
0	0	f_X	100 ns	125 ns
0	1	$f_X/2$	200 ns	250 ns
1	0	$f_X/4$	400 ns	500 ns
1	1	$f_X/8$	800 ns	1 μs

Cautions 1. Do not rewrite the PRSM0 register while watch timer and CSIB0 are operating.

2. Set the PRSM0 register before setting the BGCE0 bit to 1.

(2) Prescaler compare register 0 (PRSCM0)

The PRSCM0 register is an 8-bit compare registers.

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 while watch timer and CSIB are operating.

2. Set the PRSCM0 register before setting the PRSM0.BGCE0 bit to 1.

15.8.1 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.

$$f_{BRG} = \frac{f_{XX}}{2^{k+1} \times N}$$

Caution Set so that the f_{BRG} is 8 MHz or less.

Remark f_{BRG}: BRG count clock
f_{XX}: Main clock oscillation frequency
k: PRSM0 register setting value = 0 to 3
N: PRSCM0 register setting value = 1 to 256
However, N = 256 only when PRSCM0 register is set to 00H.

15.9 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 has been completed.

(2) In regards to registers that are forbidden from being rewritten during operations (CBnCTL0.CBnPWR bit is 1), if rewriting has been carried out by mistake during operations, set the CBnCTL0.CBnPWR bit to 0 once, then initialize CSIBn.
Registers to which rewriting during operation are prohibited 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 and 4 (CBnCTL1.CBnDAP bit = 1), the CBnSTR.CBnTSF bit is cleared half a $\overline{\text{SCKBn}}$ clock 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 CBnTSF bit = 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 CBnTSF bit = 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 for using DMA.

CHAPTER 16 I²C BUS

To use the I²C bus function, use the P914/SDA00 and P915/SCL00 pins as the serial transmit/receive data I/O pin (SDA00) and serial clock I/O pin (SCL00), respectively, and set them to N-ch open-drain output.

In the V850ES/Hx3, one channel of I²C bus is provided.

16.1 Mode Switching of I²C Bus and Other Serial Interfaces

16.1.1 UARTD4 and I²C00 mode switching

In the V850ES/HJ3, UARTD4 and I²C00 are alternate functions of the same pin and therefore cannot be used simultaneously. Set I²C00 in advance, using the PMC9, PFC9, and PFCE9 registers, before use.

Caution The transmit/receive operation of UARTD4 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 16-1. UARTD4 and I²C00 Mode Switch Settings

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

	15	14	13	12	11	10	9	8
PMC9	PMC915	PMC914	PMC913	PMC912	PMC911	PMC910	PMC99	PMC98
	7	6	5	4	3	2	1	0
	PMC97	PMC96	PMC95	PMC94	PMC93	PMC92	PMC91	PMC90

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

	15	14	13	12	11	10	9	8
PFC9	PFC915	PFC914	PFC913	PFC912	PFC911	PFC910	PFC99	PFC98
	7	6	5	4	3	2	1	0
	PFC97	PFC96	PFC95	PFC94	PFC93	PFC92	PFC91	PFC90

After reset: 0000H R/W Address: FFFFF712H, FFFFF713H

	15	14	13	12	11	10	9	8
PFCE9	PFCE915	PFCE914	PFCE913	PFCE912	PFCE911	PFCE910	PFCE99	PFCE98
	7	6	5	4	3	2	1	0
	PFCE97	PFCE96	PFCE95	PFCE94	PFCE93	PFCE92	PFCE91	PFCE90

PMC9n	PFC9n	PFCE9n	Operation mode
0			Port I/O mode
1	0	0	Setting prohibited
	0	1	Interrupt function
	1	0	I²C00 mode
	1	1	UARTD4 mode

Remarks 1. n = 14, 15

2. = don't care

16.2 Features

The I²C00 has the following two modes.

- Operation stop mode
- I²C (Inter IC) bus mode (multimaster supported)

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

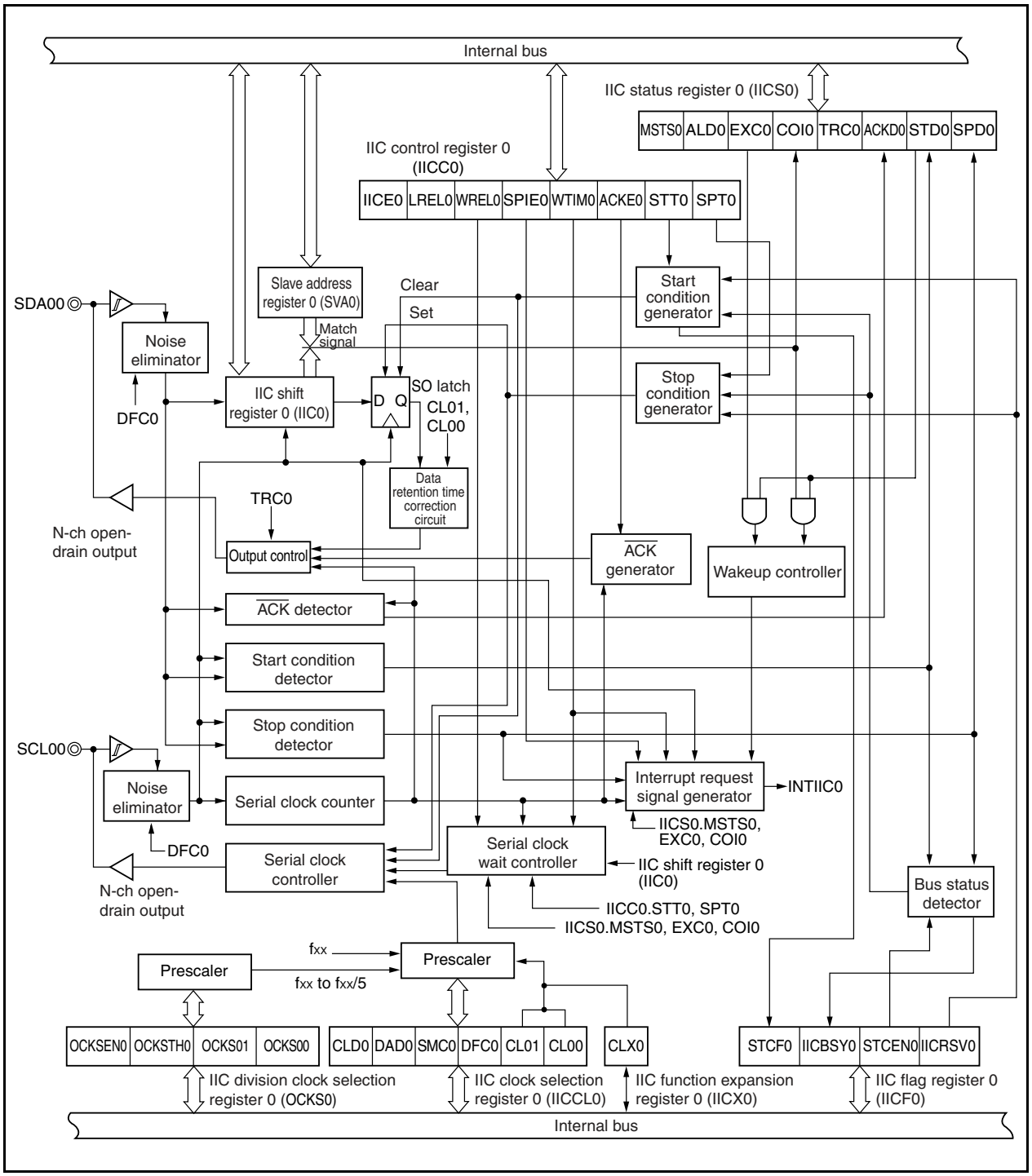
(2) I²C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCL00) line and a serial data bus (SDA00) line.

This mode complies with the I²C bus format and the master device can generate “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received state and data by hardware. This function can simplify the part of application program that controls the I²C bus.

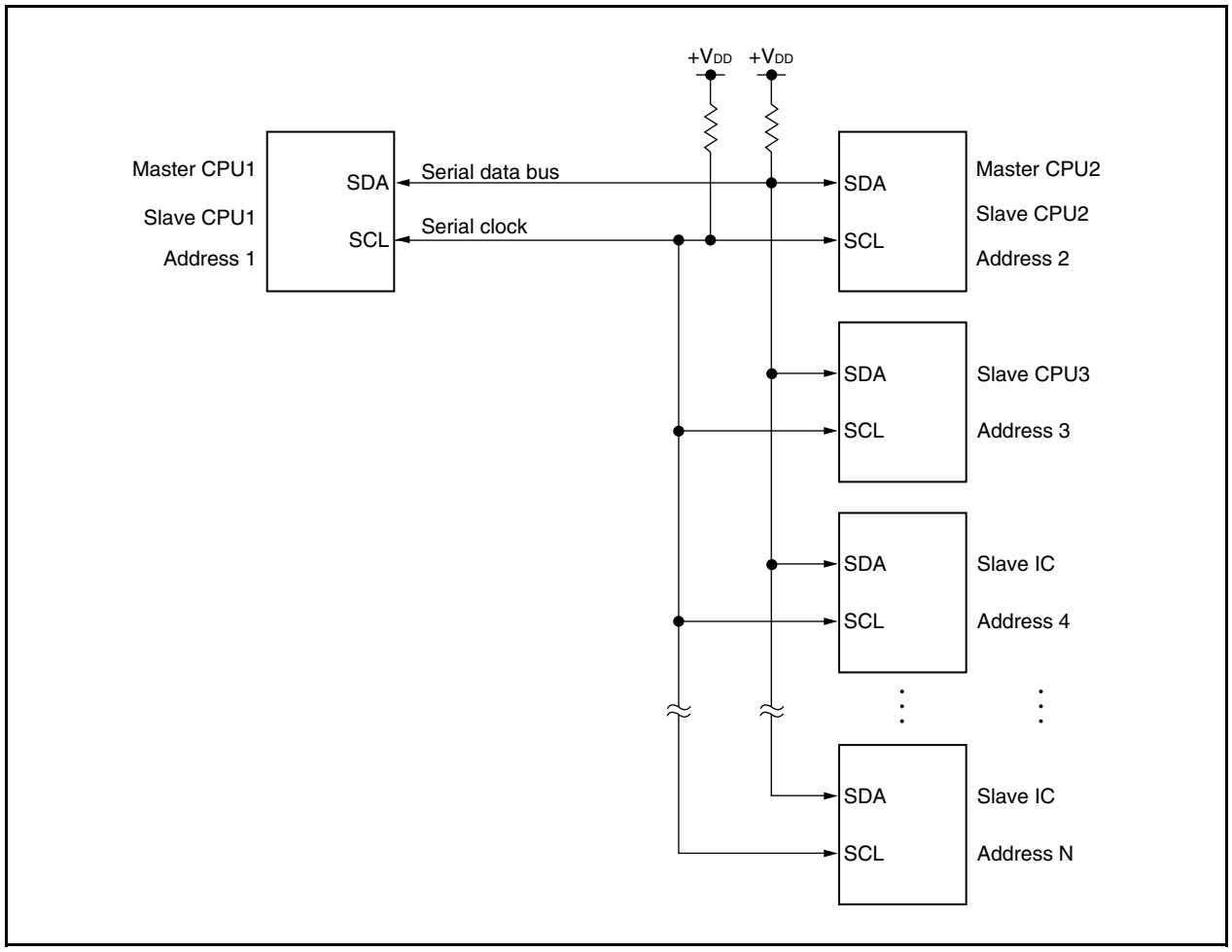
Since the SCL00 and SDA00 pins are used for N-ch open drain outputs, I²C00 requires pull-up resistors for the serial clock line and the serial data bus line.

Figure 16-2. Block Diagram of I²C00



A serial bus configuration example is shown below.

Figure 16-3. Serial Bus Configuration Example Using I²C Bus



16.3 Configuration

I²C00 includes the following hardware.

Table 16-1. Configuration of I²C00

Item	Configuration
Registers	IIC shift register 0 (IIC0) Slave address register 0 (SVA0)
Control registers	IIC control register 0 (IICC0) IIC status register 0 (IICS0) IIC flag register 0 (IICF0) IIC clock selection register 0 (IICCL0) IIC function expansion register 0 (IICX0) IIC division clock selection register 0 (OCKS0)

(1) IIC shift register 0 (IIC0)

The IIC0 register is used to convert 8-bit serial data to 8-bit parallel data and to convert 8-bit parallel data to 8-bit serial data. The IIC0 register can be used for both transmission and reception.

Write and read operations to the IIC0 register are used to control the actual transmit and receive operations.

The IIC0 register can be read or written in 8-bit units.

Reset sets IIC0 to 00H.

(2) Slave address register 0 (SVA0)

The SVA0 register sets local addresses when in slave mode.

The SVA0 register can be read or written in 8-bit units.

Reset sets SVA0 to 00H.

(3) SO latch

The SO latch is used to retain the SDA00 pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request signal (INTIIC0) when the address received by this register matches the address value set to the SVA0 register or when an extension code is received.

(5) Prescaler

This selects the sampling clock to be used.

(6) Serial clock counter

This counter counts the serial clocks that are output and the serial clocks that are input during transmit/receive operations and is used to verify that 8-bit data was sent or received.

(7) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIIC0).

An I²C interrupt is generated following either of two triggers.

- Falling of the eighth or ninth clock of the serial clock (set by IICC0.WTIM0 bit)
- Interrupt request generated when a stop condition is detected (set by IICC0.SPIE0 bit)

(8) Serial clock controller

In master mode, this circuit generates the clock output via the SCL00 pin from a sampling clock.

(9) Serial clock wait controller

This circuit controls the wait timing.

(10) $\overline{\text{ACK}}$ generator, stop condition detector, start condition detector, and $\overline{\text{ACK}}$ detector

These circuits are used to generate and detect various statuses.

(11) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

(12) Start condition generator

This circuit generates a start condition when the IICC0.STT0 bit is set.

However, in the communication reservation disabled status (IICF0.IICRSV0 bit = 1), when the bus is not released (IICF0.IICBSY0 bit = 1), start condition requests are ignored and the IICF0.STCF0 bit is set to 1.

(13) Stop condition generator

A stop condition is generated when the IICC0.SPT0 bit is set (1).

(14) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the IICF0.STCEN0 bit.

16.4 Registers

I²C00 is controlled by the following registers.

- IIC control register 0 (IICC0)
- IIC status register 0 (IICS0)
- IIC flag register 0 (IICF0)
- IIC clock selection register 0 (IICCL0)
- IIC function expansion register 0 (IICX0)
- IIC division clock selection register 0 (OCKS0)

The following registers are also used.

- IIC shift register 0 (IIC0)
- Slave address register 0 (SVA0)

Remark For the alternate-function pin settings, see **Table 4-25 Using Port Pin as Alternate-Function Pin**.

(1) IIC control register 0 (IICC0)

The IICC0 register is used to enable/stop I²C00 operations, set wait timing, and set other I²C operations.

This register can be read or written in 8-bit or 1-bit units. However, set the SPIE0, WTIM0, and ACKE0 bits when the IICE0 bit is 0 or during the wait period. When setting the IICE0 bit from “0” to “1”, these bits can also be set at the same time.

Reset sets this register to 00H.

(1/4)

After reset: 00H R/W Address: IICC0 FFFFFD82H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICC0	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I²C00 operation enable/disable specification
0	Stops operation. Resets the IICS0 register[Note 1]. Stops internal operation.
1	Enables operation.

Be sure to set this bit to 1 when the SCL00 and SDA00 lines are high level.

Condition for clearing (IICE0 bit = 0)	Condition for setting (IICE0 bit = 1)
• Cleared by instruction • Reset	• Set by instruction

LREL0[Note 2]	Exit from communications
0	Normal operation
1	Exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. This is used in cases in which an extension code that is irrelevant for the local unit has been received. The SCL00 and SDA00 lines are set to high impedance. The STT0, SPT0, IICS0.MSTS0, IICS0.EXC0, IICS0.COI0, IICS0.TRC0, IICS0.ACKD0, and IICS0.STD0 bits are cleared to 0.

The standby mode following exit from communications remains set as it is until the following communications entry conditions are met.
- After a stop condition is detected, restart is in master mode.
- An address match or extension code reception occurs after the start condition.

Condition for clearing (LREL0 bit = 0)	Condition for setting (LREL0 bit = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WREL0[Note 2]	Wait cancellation control
0	Does not cancel wait
1	Cancels wait. This setting is automatically cleared to 0 after wait is canceled.

Condition for clearing (WREL0 bit = 0)	Condition for setting (WREL0 bit = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Notes 1. The IICS0 register, and the IICF0.STCF0, IICF0.IICBSY0, IICCL0.CLD0, and IICCL0.DAD0 bits are reset.

2. This flag's signal is invalid when the IICE0 bit = 0.

Caution If the I²C00 operation is enabled (IICE0 bit = 1) when the SCL00 line is high level and the SDA00 line is low level, the start condition is detected immediately. To avoid this, after enabling the I²C00 operation, immediately set the LREL0 bit to 1 with a bit manipulation instruction.

(2/4)

SPIE0[Note]	Enable/disable generation of interrupt request when stop condition is detected
0	Disables
1	Enables

Condition for clearing (SPIE0 bit = 0)	Condition for setting (SPIE0 bit = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIM0[Note]	Control of wait and interrupt request generation
0	Interrupt request is generated at the eighth clock's falling edge. Master mode: After output of eight clocks, clock output is set to low level and wait is set. Slave mode: After input of eight clocks, the clock is set to low level and wait is set for master device.
1	Interrupt request is generated at the ninth clock's falling edge. Master mode: After output of nine clocks, clock output is set to low level and wait is set. Slave mode: After input of nine clocks, the clock is set to low level and wait is set for master device.

An interrupt is generated at the falling of the 9th clock during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. When in master mode, a wait is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a wait is inserted at the falling edge of the ninth clock after $\overline{ACK}$ is issued. However, when the slave device has received an extension code, a wait is inserted at the falling edge of the eighth clock.

Condition for clearing (WTIM0 bit = 0)	Condition for setting (WTIM0 bit = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKE0[Note]	Acknowledgment control
0	Disables acknowledgment.
1	Enables acknowledgment. During the ninth clock period, the SDA00 line is set to low level.

The ACKE0 bit setting is invalid for address reception. In this case, $\overline{ACK}$ is generated when the addresses match. However, the ACKE0 bit setting is valid for address reception of the extension code.

Condition for clearing (ACKE0 bit = 0)	Condition for setting (ACKE0 bit = 1)
• Cleared by instruction • Reset	• Set by instruction

Note This flag's signal is invalid when the IICE0 bit = 0.

(3/4)

STT0	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in STOP mode): Generates a start condition (for starting as master). The SDA00 line is changed from high level to low level while the SCL00 line is high level and then the start condition is generated. Next, after the rated amount of time has elapsed, the SCL00 line is changed to low level (wait status). When a third party is communicating • When communication reservation function is enabled (IICF0.IICRSV0 bit = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. • When communication reservation function is disabled (IICRSV0 bit = 1) The IICF0.STCF0 bit is set to 1 and the information set (1) to the STT0 bit is cleared. No start condition is generated. In the wait state (when master device): Generates a restart condition after releasing the wait.

Cautions concerning set timing

For master reception: Cannot be set to 1 during transfer. Can be set to 1 only when the ACKE0 bit has been cleared to 0 and slave has been notified of final reception.

For master transmission: A start condition may not be generated normally during the $\overline{\text{ACK}}$ period. Set to 1 during the wait period that follows output of the ninth clock.

- Cannot be set to 1 at the same time as the SPT0 bit.
- When the STT0 bit is set to 1, setting the STT0 bit to 1 again is disabled until the setting is cleared to 0.

Condition for clearing (STT0 bit = 0)	Condition for setting (STT0 bit = 1)
• When the STT0 bit is set to 1 in the communication reservation disabled status • Cleared when start condition is generated by master device • When the LREL0 bit = 1 (exit from communications) • When the IICE0 bit = 1 → 0 (operation stop) • Reset	• Set by instruction

Remark The STT0 bit is 0 if it is read after data setting.

(4/4)

SPT0	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master device's transfer). After the SDA00 line goes to low level, either set the SCL00 line to high level or wait until the SCL00 pin goes to high level. Next, after the rated amount of time has elapsed, the SDA00 line is changed from low level to high level and a stop condition is generated.

Cautions concerning setting timing

For master reception: Cannot be set to 1 during transfer. Can be set to 1 only when the ACKE0 bit has been cleared to 0 and during the wait period after slave has been notified of final reception.

For master transmission: A stop condition may not be generated normally during the $\overline{\text{ACK}}$ period. Set to 1 during the wait period that follows output of the ninth clock.

- Cannot be set to 1 at the same time as the STT0 bit.
- The SPT0 bit can be set to 1 only when in master mode[Note].
- When the WTIM0 bit has been cleared to 0, if the SPT0 bit is set to 1 during the wait period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock.
  The WTIM0 bit should be changed from 0 to 1 during the wait period following output of eight clocks, and the SPT0 bit should be set to 1 during the wait period that follows output of the ninth clock.
- When the SPT0 bit is set to 1, setting the SPT0 bit to 1 again is disabled until the setting is cleared to 0.

Condition for clearing (SPT0 bit = 0)	Condition for setting (SPT0 bit = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • When the LREL0 bit = 1 (exit from communications) • When the IICE0 bit = 1 → 0 (operation stop) • Reset	• Set by instruction

Note Set the SPT0 bit to 1 only in master mode. However, the SPT0 bit must be set to 1 and a stop condition generated before the first stop condition is detected following the switch to operation enable status. For details, see **16.15 Cautions**.

Caution When the IICS0.TRC0 bit is set to 1, the WREL0 bit is set to 1 during the ninth clock and wait is canceled, after which the TRC0 bit is cleared to 0 and the SDA00 line is set to high impedance.

Remark The SPT0 bit is 0 if it is read after data setting.

(2) IIC status register 0 (IICS0)

The IICS0 register indicates the status of the I²C00 bus.

This register is read-only in 8-bit or 1-bit units.

However, the IICS0 register can only be read when the IICC0.STT0 bit is 1 or during the wait period.

Reset sets this register to 00H.

Caution Accessing the IICS0 register is prohibited in the following statuses. For details, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

- **When the CPU operates with the subclock and the main clock oscillation is stopped**
- **When the CPU operates with the low-speed internal oscillation clock**

(1/3)

After reset: 00H R Address: IICS0 FFFFFD86H

	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	Master device status
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS0 bit = 0)	Condition for setting (MSTS0 bit = 1)
• When a stop condition is detected • When the ALD0 bit = 1 (arbitration loss) • Cleared by the IICC0.LREL0 bit = 1 (exit from communications) • When the IICC0.IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD0	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. The MSTS0 bit is cleared to 0.

Condition for clearing (ALD0 bit = 0)	Condition for setting (ALD0 bit = 1)
• Automatically cleared after the IICS0 register is read[Note] • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

Note This bit is also cleared when a bit manipulation instruction is executed for another bit in the IICS0 register.

(2/3)

EXC0	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC0 bit = 0)	Condition for setting (EXC0 bit = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by the LREL0 bit = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

COI0	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI0 bit = 0)	Condition for setting (COI0 bit = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by the LREL0 bit = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 • Reset	• When the received address matches the local address (SVA0 register) (set at the rising edge of the eighth clock).

TRC0	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDA00 line is set for high impedance.
1	Transmit status. The value in the SO latch is enabled for output to the SDA00 line (valid starting at the rising edge of the first byte’s ninth clock).

Condition for clearing (TRC0 bit = 0)	Condition for setting (TRC0 bit = 1)
• When a stop condition is detected • Cleared by the LREL0 bit = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Cleared by the IICC0.WREL0 bit = 1[Note] (wait release) • When the ALD0 bit changes from 0 to 1 (arbitration loss) • Reset Master • When “1” is output to the first byte’s LSB (transfer direction specification bit) Slave • When a start condition is detected When not used for communication	Master • When a start condition is generated • When “0” is output to the first byte’s LSB (transfer direction specification bit) Slave • When “1” is input in the first byte’s LSB (transfer direction specification bit)

Note The IICS0.TRC0 bit is cleared to 0 and the SDA00 line become high impedance when the IICC0.WREL0 bit is set to 1 and wait state is released at the ninth clock with the TRC0 bit = 1.

(3/3)

ACKD0	Detection of $\overline{ACK}$
0	$\overline{ACK}$ was not detected.
1	$\overline{ACK}$ was detected.

Condition for clearing (ACKD0 bit = 0)	Condition for setting (ACKD0 bit = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by the LREL0 bit = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• After the SDA00 pin is set to low level at the rising edge of the SCL00 pin's ninth clock

STD0	Detection of start condition
0	Start condition was not detected.
1	Start condition was detected. This indicates that the address transfer period is set.

Condition for clearing (STD0 bit = 0)	Condition for setting (STD0 bit = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by the LREL0 bit = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPD0	Detection of stop condition
0	Stop condition was not detected.
1	Stop condition was detected. The master device's communication is terminated and the bus is released.

Condition for clearing (SPD0 bit = 0)	Condition for setting (SPD0 bit = 1)
• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

(3) IIC flag register 0 (IICF0)

IICF0 is a register that set the operation mode of I²C00 and indicate the status of the I²C bus.

The IICF0 register can be read or written in 8-bit or 1-bit units. However, the STCF0 and IICBSY0 bits are read-only.

The IICRSV0 bit can be used to enable/disable the communication reservation function (see **16.14 Communication Reservation**).

The STCEN0 bit can be used to set the initial value of the IICBSY0 bit (see **16.15 Cautions**).

The IICRSV0 and STCEN0 bits can be written only when the operation of I²C00 is disabled (IICC0.IICE0 bit = 0). When operation is enabled, the IICF0 register can be read.

Reset sets this register to 00H.

After reset: 00H R/W[Note] Address: IICF0 FFFFFD8AH

	<7>	<6>	5	4	3	2	<1>	<0>
IICF0	STCF0	IICBSY0	0	0	0	0	STCEN0	IICRSV0

STCF0	IICC0.STT0 clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear STT0 flag

Condition for clearing (STCF0 bit = 0)	Condition for setting (STCF0 bit = 1)
• Clearing by setting the STT0 bit = 1 • When the IICE0 bit = 1 → 0 (operation stop) • Reset	• Generating start condition unsuccessful and the STT0 bit cleared to 0 when communication reservation is disabled (IICRSV0 bit = 1).

IICBSY0	I²C00 bus status flag
0	Bus release status (initial communication status when STCEN0 bit = 1)
1	Bus communication status (initial communication status when STCEN0 bit = 0)

Condition for clearing (IICBSY0 bit = 0)	Condition for setting (IICBSY0 bit = 1)
• Detection of stop condition • When the IICE0 bit = 1 → 0 (operation stop) • Reset	• Detection of start condition • Setting of the IICE0 bit when the STCEN0 bit = 0

STCEN0	Initial start enable trigger
1	After operation is enabled (IICE0 bit = 1), enable generation of a start condition upon detection of a stop condition.
	After operation is enabled (IICE0 bit = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCEN0 bit = 0)	Condition for setting (STCEN0 bit = 1)
• Detection of start condition • Reset	• Setting by instruction

IICRSV0	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSV0 bit = 0)	Condition for setting (IICRSV0 bit = 1)
• Clearing by instruction • Reset	• Setting by instruction

Note Bits 6 and 7 are read-only bits.

Cautions 1. Write to the STCEN0 bit only when the operation is stopped (IICE0 bit = 0).

2. As the bus release status (IICBSY0 bit = 0) is recognized regardless of the actual bus status when the STCEN0 bit = 1, when generating the first start condition (STT0 bit = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

3. Write to the IICRSV0 bit only when the operation is stopped (IICE0 bit = 0).

(4) IIC clock selection register 0 (IICCL0)

The IICCL0 register is used to set the transfer clock for the I²C00 bus.

This register can be read or written in 8-bit or 1-bit units. However, the CLD0 and DAD0 bits are read-only.

The SMC0, CL01 and CL00 bits are set in combination with the IICX0.CLX0 and OCKS0.OCKSTH0 bits (see **16.4 (6) I²C00 transfer clock setting method**).

Set the IICCL0 register when the IICC0.IICE0 bit = 0.

Reset sets this register to 00H.

After reset: 00H R/W[Note] Address: IICCL0 FFFFFD84H

	7	6	<5>	<4>	3	2	1	0
IICCL0	0	0	CLD0	DAD0	SMC0	DFC0	CL01	CL00

CLD0	Detection of SCL00 pin level (valid only when IICC0.IICE0 bit = 1)
0	The SCL00 pin was detected at low level.
1	The SCL00 pin was detected at high level.

Condition for clearing (CLD0 bit = 0)	Condition for setting (CLD0 bit = 1)
• When the SCL00 pin is at low level • When the IICE0 bit = 1 → 0 (operation stop) • Reset	• When the SCL00 pin is at high level

DAD0	Detection of SDA00 pin level (valid only when IICE0 bit = 1)
0	The SDA00 pin was detected at low level.
1	The SDA00 pin was detected at high level.

Condition for clearing (DAD0 bit = 0)	Condition for setting (DAD0 bit = 1)
• When the SDA00 pin is at low level • When the IICE0 bit = 1 → 0 (operation stop) • Reset	• When the SDA00 pin is at high level

SMC0	Operation mode switching
0	Operates in standard mode.
1	Operates in high-speed mode.

DFC0	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Digital filter can be used only in high-speed mode. In high-speed mode, the transfer clock does not vary regardless of DFC0 bit set/clear. The digital filter is used for noise elimination in high-speed mode.	

Note Bits 4 and 5 are read-only bits.

Caution Be sure to set bits 7 and 6 to "0".

(5) IIC function expansion register 0 (IICX0)

The IICX0 register is used to set the function expansion of I²C00 (valid only in high-speed mode).

This register can be read or written in 8-bit or 1-bit units. The CLX0 bit is set in combination with the IICCL0.SMC0, IICCL0.CL01, IICCL0.CL00, and OCKS0.OCKSTH0 bits (see **16.4 (6) I²C00 transfer clock setting method**).

Set the IICX0 register when the IICC0.IICE0 bit = 0.

Reset sets this register to 00H.

After reset: 00H R/W Address: IICX0 FFFFFD85H

	7	6	5	4	3	2	1	<0>
IICX0	0	0	0	0	0	0	0	CLX0

(6) I²C00 transfer clock setting method

The I²C00 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, 90, 96, 120, 132, 172, 176, 198, 220, 258, 264, 330, 344, 430 (see **Table 16-2 Selection Clock Setting**.)

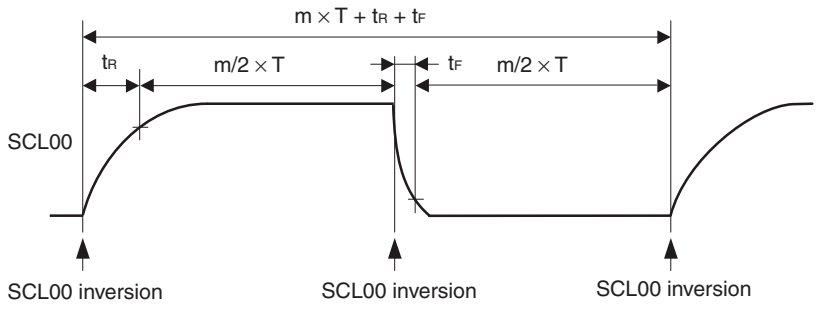
T: $1/f_{XX}$

t_R: SCL00 rise time

t_F: SCL00 fall time

For example, the I²C00 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 selection clock is set using a combination of the IICCL0.SMC0, IICCL0.CL01, IICCL0.CL00, IICX0.CLX0 and OCKS0.OCKSTH0 bit.

Table 16-2. Selection Clock Setting

IICX0	IICCL0			Selection Clock	Transfer Clock	Settable Main Clock Frequency (f_{XX}) Range	Operating Mode
Bit 0	Bit 3	Bit 1	Bit 0				
CLX0	SMC0	CL01	CL00				
0	0	0	0	f_{XX} (when OCKS0 = 18H set)	$f_{XX}/44$	4.00 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} ≤ 20.95 MHz	
0	0	0	1	$f_{XX}/2$ (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} ≤ 25.14 MHz	
				$f_{XX}/4$ (when OCKS0 = 12H set)	$f_{XX}/344$	16.76 MHz ≤ f_{XX} ≤ 32.00 MHz	
				$f_{XX}/5$ (when OCKS0 = 13H set)	$f_{XX}/430$	20.95 MHz ≤ f_{XX} ≤ 32.00 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)	$f_{XX}/66$	6.40 MHz	
				$f_{XX}/2$ (when OCKS0 = 10H set)	$f_{XX}/132$	12.80 MHz	
				$f_{XX}/3$ (when OCKS0 = 11H set)	$f_{XX}/198$	19.20 MHz	
				$f_{XX}/4$ (when OCKS0 = 12H set)	$f_{XX}/264$	25.60 MHz	
				$f_{XX}/5$ (when OCKS0 = 13H set)	$f_{XX}/330$	32.00 MHz	
0	1	0	×	$f_{XX}/2$ (when OCKS0 = 10H set)	$f_{XX}/48$	8.00 MHz ≤ f_{XX} ≤ 16.76 MHz	High-speed mode (SMC0 bit = 1)
				$f_{XX}/3$ (when OCKS0 = 11H set)	$f_{XX}/72$	12.00 MHz ≤ f_{XX} ≤ 25.14 MHz	
				$f_{XX}/4$ (when OCKS0 = 12H set)	$f_{XX}/96$	16.00 MHz ≤ f_{XX} ≤ 32.00 MHz	
				$f_{XX}/5$ (when OCKS0 = 13H set)	$f_{XX}/120$	20.00 MHz ≤ f_{XX} ≤ 32.00 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)	$f_{XX}/18$	6.40 MHz	
				$f_{XX}/2$ (when OCKS0 = 10H set)	$f_{XX}/36$	12.80 MHz	
				$f_{XX}/3$ (when OCKS0 = 11H set)	$f_{XX}/54$	19.20 MHz	
				$f_{XX}/4$ (when OCKS0 = 12H set)	$f_{XX}/72$	25.60 MHz	
				$f_{XX}/5$ (when OCKS0 = 13H set)	$f_{XX}/90$	32.00 MHz	
1	1	0	×	$f_{XX}/2$ (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)	$f_{XX}/60$	20.00 MHz ≤ f_{XX} ≤ 20.95 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

(7) IIC division clock selection registers 0 (OCKS0)

The OCKS0 register controls the I²C00 division clock.

This register controls the I²C00 division clock via the OCKS0 register.

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

Reset sets this register to 00H.

After reset: 00H R/W Address: OCKS0 FFFFF340H

	7	6	5	4	3	2	1	0
OCKS0	0	0	0	OCKSEN0	OCKSTH0	0	OCKS01	OCKS00

OCKSEN0	Operation setting of I²C division clock
0	Disable I²C division clock operation
1	Enable I²C division clock operation

OCKSTH0	OCKS01	OCKS00	Selection of I²C division clock
0	0	0	$f_{XX}/2$
0	0	1	$f_{XX}/3$
0	1	0	$f_{XX}/4$
0	1	1	$f_{XX}/5$
1	×	×	f_{XX}

(8) IIC shift register 0 (IIC0)

The IIC0 shift register is used for serial transmission/reception (shift operations) that is synchronized with the serial clock.

This register can be read or written in 8-bit units, but data should not be written to the IIC0 shift register during a data transfer.

Access (read/write) the IIC0 shift register only during the wait period. Accessing this register in communication states other than the wait period is prohibited. However, for the master device, the IIC0 shift register can be written once only after the transmission trigger bit (IICC0.STT0 bit) has been set to 1.

When the IIC0 shift register is written during wait, the wait is cancelled and data transfer is started.

Reset sets this register to 00H.

After reset: 00H R/W Address: IIC0 FFFFFD80H

	7	6	5	4	3	2	1	0
IIC0								

(9) Slave address register 0 (SVA0)

The SVA0 register holds the I²C bus's slave addresses.

However, rewriting this register is prohibited when the IICS0.STD0 bit = 1 (start condition detection).

This register can be read or written in 8-bit units, but bit 0 is fixed to 0.

Reset sets this register to 00H.

After reset: 00H R/W Address: SVA0 FFFFFD83H

	7	6	5	4	3	2	1	0
SVA0								0

16.5 Functions

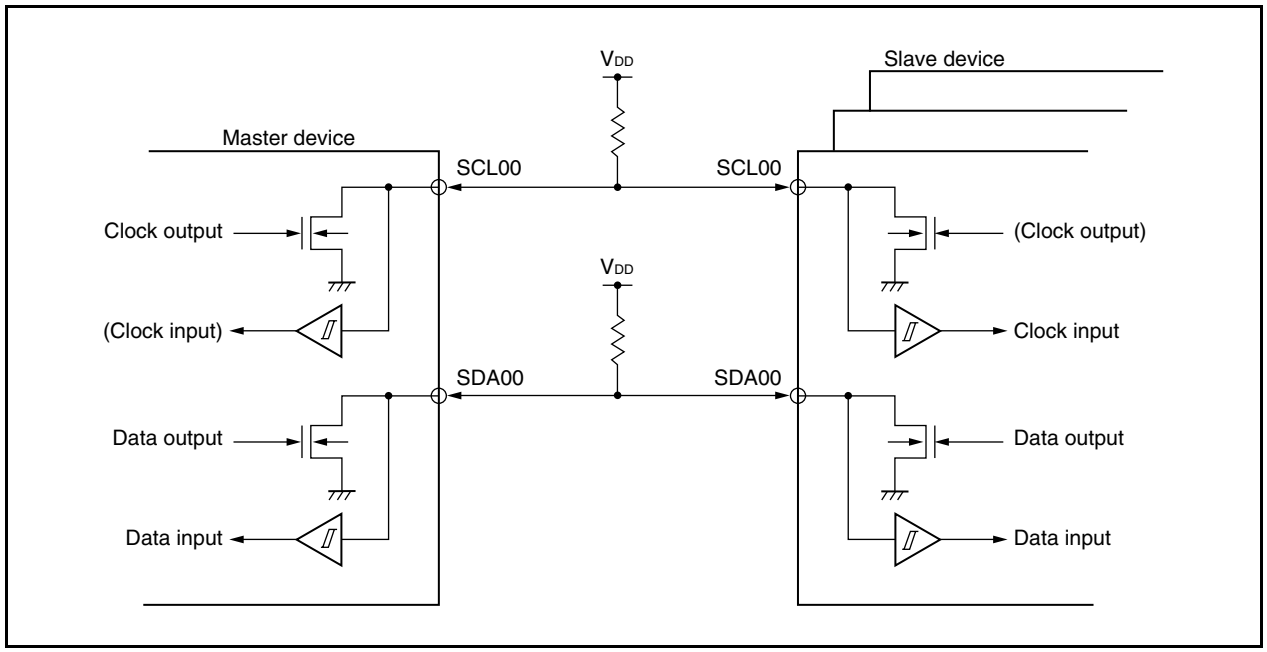
16.5.1 Pin configuration

The serial clock pin (SCL00) and serial data bus pin (SDA00) are configured as follows.

SCL00 .....This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.
SDA00 .....This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

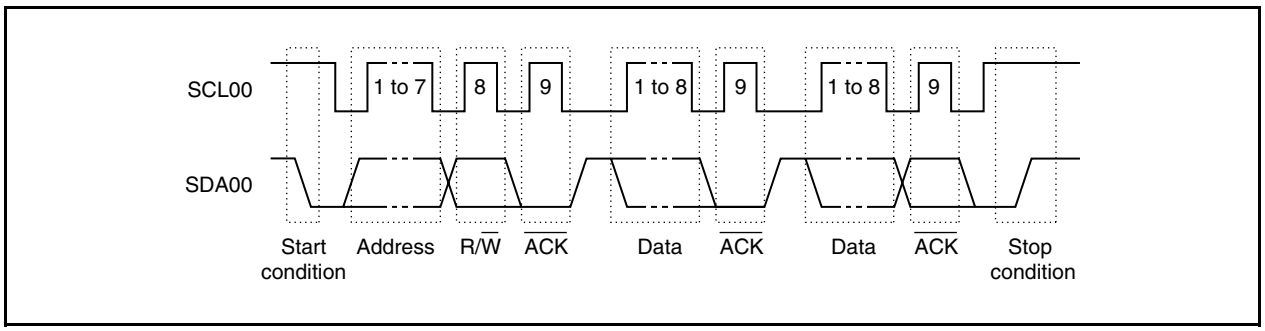
Figure 16-4. Pin Configuration Diagram



16.6 I²C Bus Definitions and Control Methods

The following section describes the I²C bus's serial data communication format and the status generated by the I²C bus. The transfer timing for the "start condition", "address", "transfer direction specification", "data", and "stop condition" generated via the I²C bus's serial data bus is shown below.

Figure 16-5. I²C Bus's Serial Data Transfer Timing



The master device generates the start condition, slave address, and stop condition.

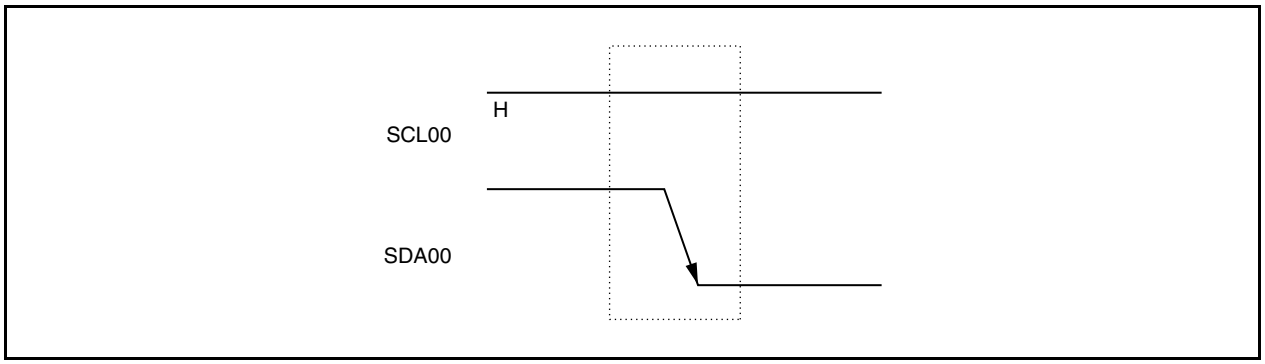
$\overline{\text{ACK}}$ can be generated by either the master or slave device (normally, it is generated by the device that receives 8-bit data).

The serial clock (SCL00) is continuously output by the master device. However, in the slave device, the SCL00's low-level period can be extended and a wait can be inserted.

16.6.1 Start condition

A start condition is met when the SCL00 pin is at high level and the SDA00 pin changes from high level to low level. The start conditions for the SCL00 pin and SDA00 pin are generated when the master device starts a serial transfer to the slave device. Start conditions can be detected when the device is used as a slave.

Figure 16-6. Start Conditions



A start condition is generated when the IICC0.STT0 bit is set to 1 after a stop condition has been detected (IICS0.SPD0 bit = 1). When a start condition is detected, IICS0.STD0 bit is set to 1.

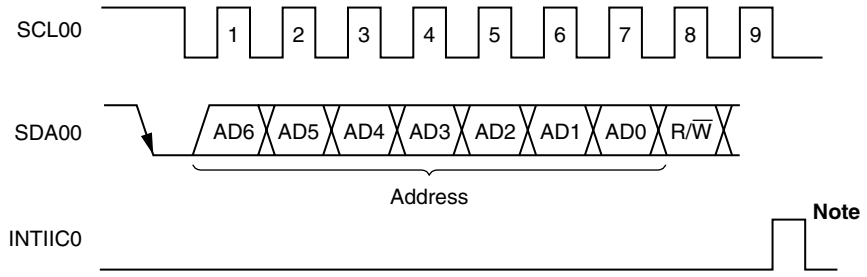
16.6.2 Addresses

The 7 bits of data that follow the start condition are defined as an address.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the SVA0 register. If the address data matches the SVA0 values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 16-7. Address



Note The interrupt request signal (INTIIC0) is generated if a local address or extension code is received during slave device operation.

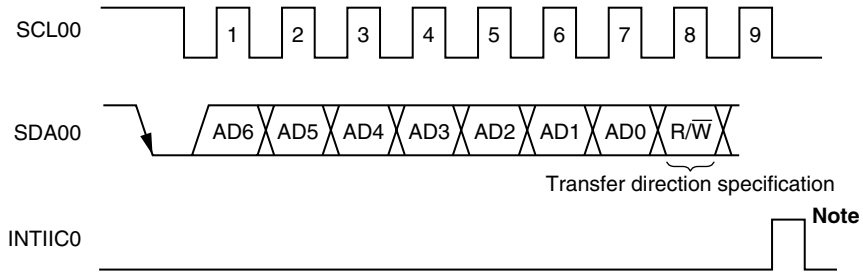
The slave address and the eighth bit, which specifies the transfer direction as described in **16.6.3 Transfer direction specification** below, are together written to the IIC0 register and are then output. Received addresses are written to the IIC0 register.

The slave address is assigned to the higher 7 bits of the IIC0 register.

16.6.3 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 16-8. Transfer Direction Specification



Note The interrupt request signal (INTIIC0) is generated if a local address or extension code is received during slave device operation.

16.6.4 $\overline{\text{ACK}}$

$\overline{\text{ACK}}$ is used to confirm the serial data status of the transmitting and receiving devices.

The receiving device returns $\overline{\text{ACK}}$ for every 8 bits of data it receives.

The transmitting device normally receives $\overline{\text{ACK}}$ after transmitting 8 bits of data. When $\overline{\text{ACK}}$ is returned from the receiving device, the reception is judged as normal and processing continues. The detection of $\overline{\text{ACK}}$ is confirmed with the IICS0.ACKD0 bit.

When the master device is the receiving device, after receiving the final data, it does not return $\overline{\text{ACK}}$ and generates the stop condition. When the slave device is the receiving device and does not return $\overline{\text{ACK}}$, the master device generates either a stop condition or a restart condition, and then stops the current transmission. Failure to return $\overline{\text{ACK}}$ may be caused by the following factors.

(a) Reception was not performed normally.
(b) The final data was received.
(c) The receiving device (slave) does not exist for the specified address.

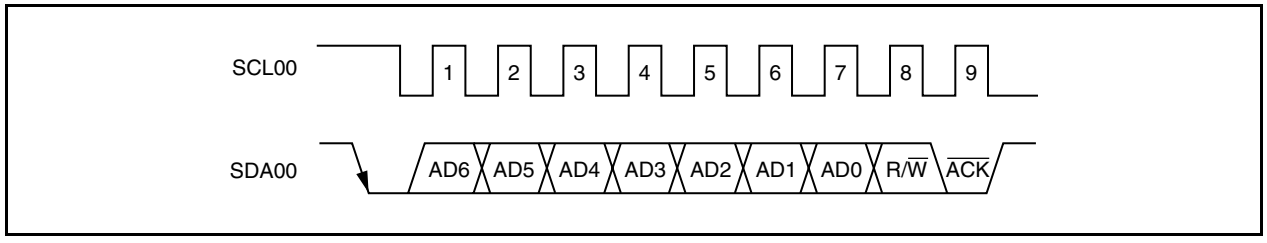
When the receiving device sets the SDA00 line to low level during the ninth clock, $\overline{\text{ACK}}$ is generated (normal reception).

When the IICC0.ACKE0 bit is set to 1, automatic $\overline{\text{ACK}}$ generation is enabled. Transmission of the eighth bit following the 7 address data bits causes the IICS0.TRC0 bit to be set. Normally, set the ACKE0 bit to 1 for reception (TRC0 bit = 0).

When the slave device is receiving (when TRC0 bit = 0), if the slave device cannot receive data or does not need to receive any more data, clear the ACKE0 bit to 0 to indicate to the master that no more data can be received.

Similarly, when the master device is receiving (when TRC0 bit = 0) and the subsequent data is not needed, clear the ACKE0 bit to 0 to prevent $\overline{\text{ACK}}$ from being generated. This notifies the slave device (transmitting device) of the end of the data transmission (transmission stopped).

Figure 16-9. $\overline{\text{ACK}}$



When the local address is received, $\overline{\text{ACK}}$ is automatically generated regardless of the value of the ACKE0 bit. No $\overline{\text{ACK}}$ is generated if the received address is not a local address (NACK).

When receiving the extension code, set the ACKE0 bit to 1 in advance to generate $\overline{\text{ACK}}$.

The $\overline{\text{ACK}}$ generation method during data reception is based on the wait timing setting, as described by the following.

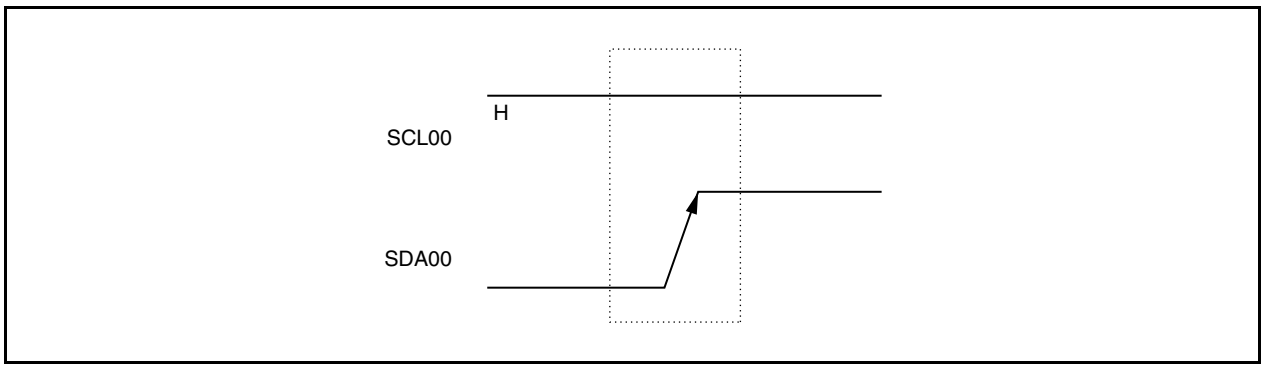
- When 8-clock wait is selected (IICC0.WTIM0 bit = 0):
  $\overline{\text{ACK}}$ is generated at the falling edge of the SCL00 pin's eighth clock if the ACKE0 bit is set to 1 before the wait state cancellation.
- When 9-clock wait is selected (IICC0.WTIM0 bit = 1):
  $\overline{\text{ACK}}$ is generated if the ACKE0 bit is set to 1 in advance.

16.6.5 Stop condition

When the SCL00 pin is at high level, changing the SDA00 pin from low level to high level generates a stop condition.

A stop condition is generated when serial transfer from the master device to the slave device has been completed. Stop conditions can be detected when the device is used as a slave.

Figure 16-10. Stop Condition



A stop condition is generated when the IICC0.SPT0 bit is set to 1. When the stop condition is detected, the IICS0.SPD0 bit is set to 1 and the interrupt request signal (INTIIC0) is generated when the IICC0.SPIE0 bit is set to 1.

16.6.6 Wait state

The wait state is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCL00 pin to low level notifies the communication partner of the wait status. When wait status has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 16-11. Wait State (1/2)

(a) When master device has a nine-clock wait and slave device has an eight-clock wait (master: transmission, slave: reception, and IICC0.ACKE0 bit = 1)

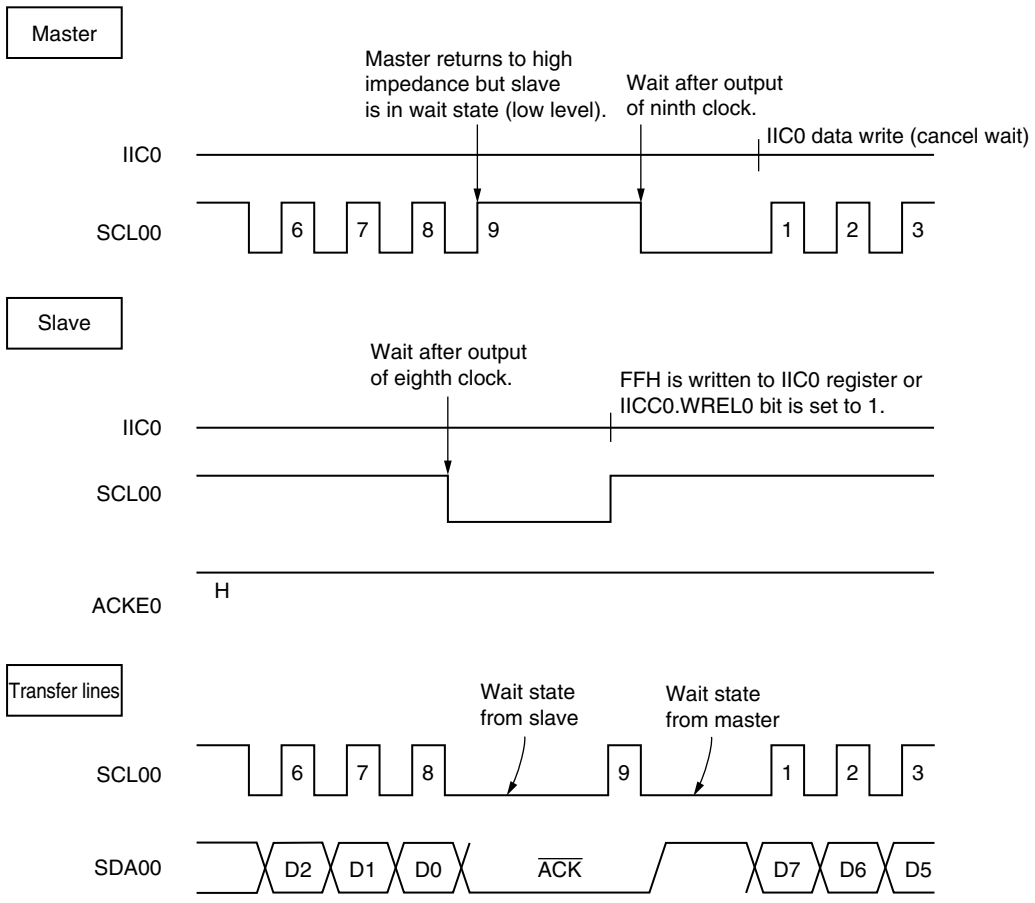
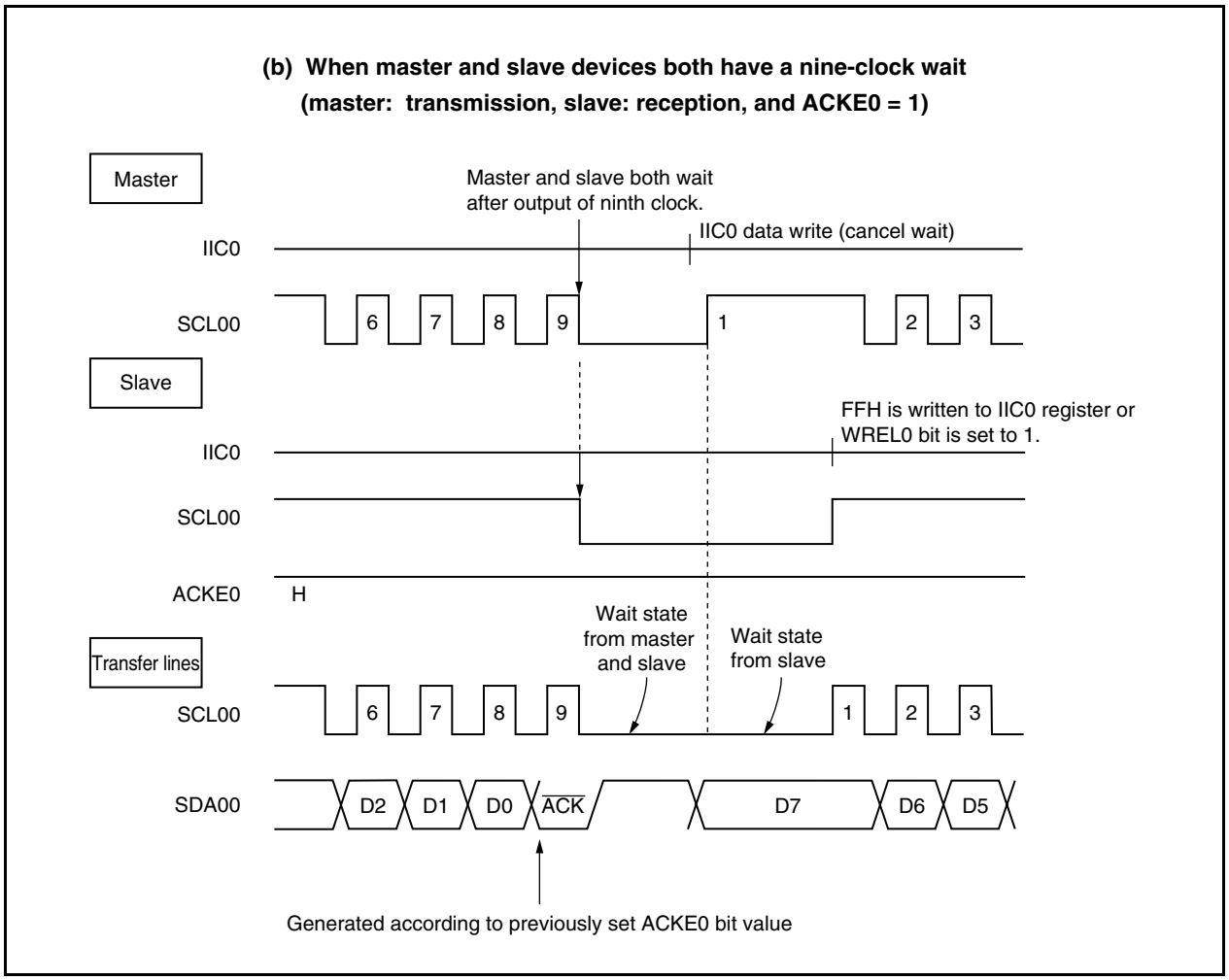


Figure 16-11. Wait State (2/2)



A wait state is automatically generated after a start condition is generated. Moreover, a wait state is automatically generated depending on the setting of the IICC0.WTIM0 bit.

Normally, when the IICC0.WREL0 bit is set to 1 or when FFH is written to the IIC0 register, the wait status is canceled and the transmitting side writes data to the IIC0 register to cancel the wait status.

The master device can also cancel the wait status via either of the following methods.

- By setting the IICC0.STT0 bit to 1
- By setting the IICC0.SPT0 bit to 1

16.6.7 Wait state cancellation method

In the case of I²C00, wait state can be canceled normally in the following ways.

- By writing data to the IIC0 register
- By setting the IICC0.WREL0 bit to 1 (wait state cancellation)
- By setting the IICC0.STT0 bit to 1 (start condition generation)[Note]
- By setting the IICC0.SPT0 bit to 1 (stop condition generation)[Note]

Note Master only

If any of these wait state cancellation actions is performed, I²C00 will cancel wait state and restart communication.

When canceling wait state and sending data (including address), write data to the IIC0 register.

To receive data after canceling wait state, or to complete data transmission, set the WREL0 bit to 1.

To generate a restart condition after canceling wait state, set the STT0 bit to 1.

To generate a stop condition after canceling wait state, set the SPT0 bit to 1.

Execute cancellation only once for each wait state.

For example, if data is written to the IIC0 register following wait state cancellation by setting the WREL0 bit to 1, conflict between the SDA00 line change timing and IIC0 register write timing may result in the data output to the SDA00 line may be incorrect.

Even in other operations, if communication is stopped halfway, clearing the IICC0.IICE0 bit to 0 will stop communication, enabling wait state to be cancelled.

If the I²C bus dead-locks due to noise, etc., setting the IICC0.LREL0 bit to 1 causes the communication operation to be exited, enabling wait state to be cancelled.

16.7 I²C Interrupt Request Signals (INTIIC0)

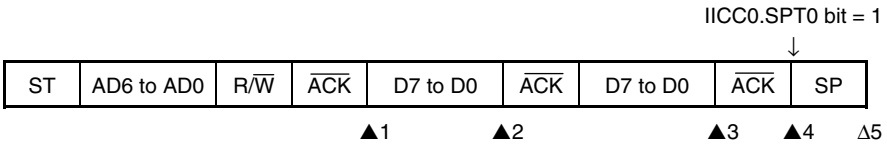
The following shows the value of the IICS0 register at the INTIIC0 interrupt request signal generation timing and at the INTIIC0 signal timing.

Remark ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
$\overline{ACK}$: Acknowledge
D7 to D0: Data
SP: Stop condition

16.7.1 Master device operation

(1) Start ~ Address ~ Data ~ Data ~ Stop (normal transmission/reception)

<1> When IICC0.WTIM0 bit = 0



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X000B

▲3: IICS0 register = 1000X000B (WTIM0 bit = 1[Note])

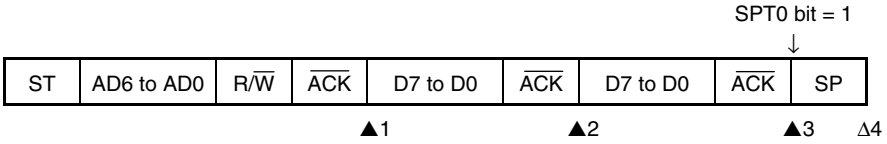
▲4: IICS0 register = 1000XX00B

Δ 5: IICS0 register = 00000001B

Note To generate a stop condition, set the WTIM0 bit to 1 and change the timing of the generation of the interrupt request signal (INTIIC0).

Remark ▲: Always generated
Δ: Generated only when IICC0.SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X100B

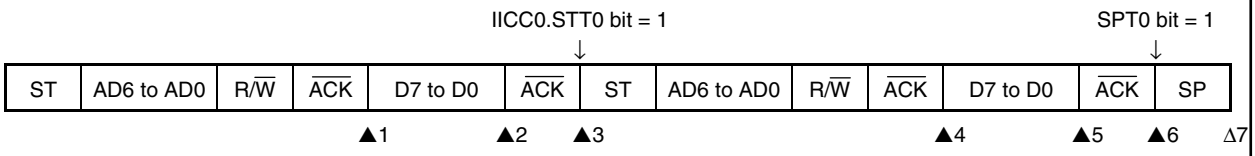
▲3: IICS0 register = 1000XX00B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

<1> When WTIM0 bit = 0



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X000B (WTIM0 bit = 1[Note 1])

▲3: IICS0 register = 1000XX00B (WTIM0 bit = 0[Note 2])

▲4: IICS0 register = 1000X110B

▲5: IICS0 register = 1000X000B (WTIM0 bit = 1[Note 3])

▲6: IICS0 register = 1000XX00B

Δ 7: IICS0 register = 00000001B

Notes 1. To generate a start condition, set the WTIM0 bit to 1 and change the timing of the generation of the interrupt request signal (INTIIC0).

2. Clear the WTIM0 bit to 0 to make the settings original.

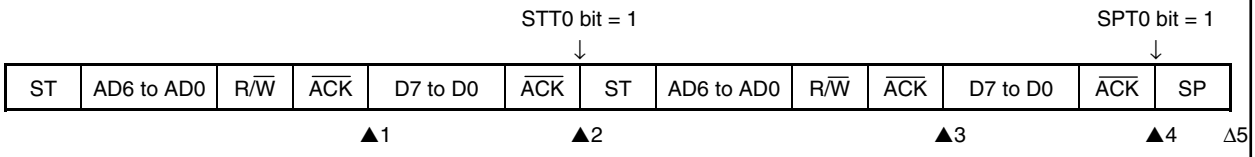
3. To generate a stop condition, set the WTIM0 bit to 1 and change the timing of the generation of the interrupt request signal (INTIIC0).

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000XX00B

▲3: IICS0 register = 1000X110B

▲4: IICS0 register = 1000XX00B

Δ 5: IICS0 register = 00000001B

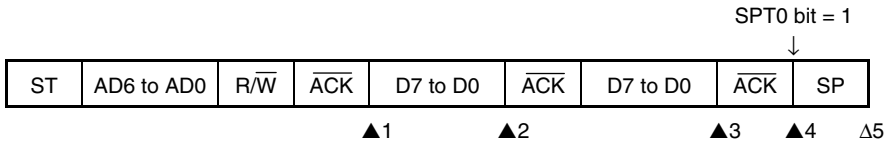
Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

(3) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

<1> When WTIM0 bit = 0



▲1: IICS0 register = 1010X110B

▲2: IICS0 register = 1010X000B

▲3: IICS0 register = 1010X000B (WTIM0 bit = 1[Note])

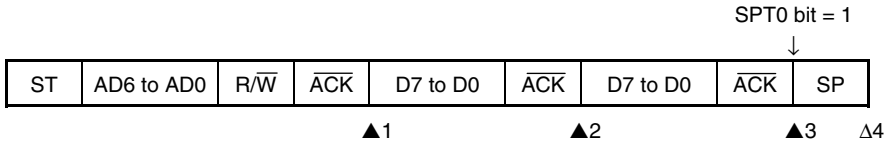
▲4: IICS0 register = 1010XX00B

Δ 5: IICS0 register = 00000001B

Note To generate a stop condition, set the WTIM0 bit to 1 and change the timing of the generation of the interrupt request signal (INTIIC0).

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 1010X110B

▲2: IICS0 register = 1010X100B

▲3: IICS0 register = 1010XX00B

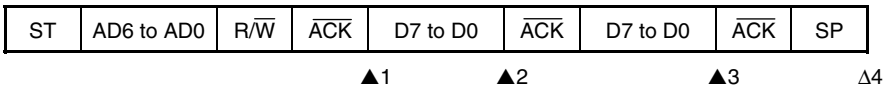
Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

16.7.2 Slave device operation (when receiving slave address data (address match))

(1) Start ~ Address ~ Data ~ Data ~ Stop

<1> When IICC0.WTIM0 bit = 0



▲1: IICS0 register = 0001X110B

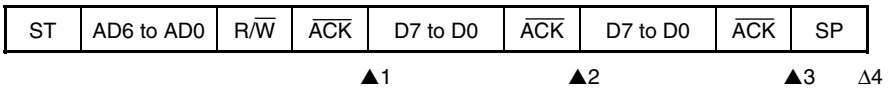
▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0001X000B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when IICC0.SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X100B

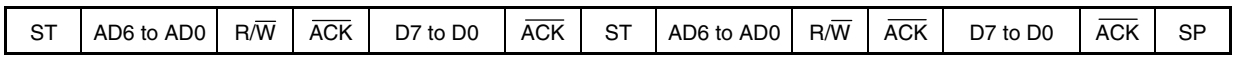
▲3: IICS0 register = 0001XX00B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, address match)



▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0001X110B

▲4: IICS0 register = 0001X000B

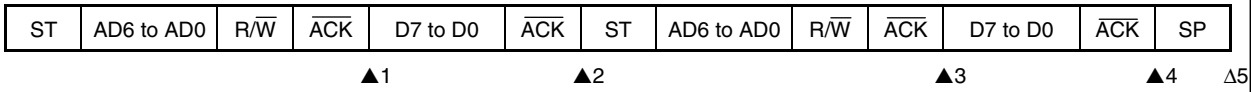
Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

<2> When WTIM0 bit = 1 (after restart, address match)



▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001XX00B

▲3: IICS0 register = 0001X110B

▲4: IICS0 register = 0001XX00B

Δ 5: IICS0 register = 00000001B

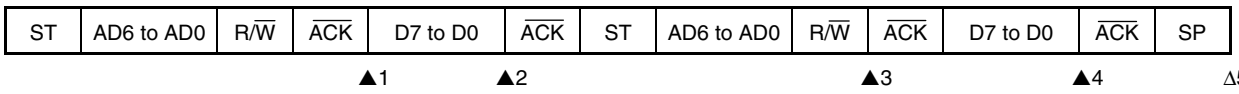
Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

(3) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, extension code reception)



▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0010X010B

▲4: IICS0 register = 0010X000B

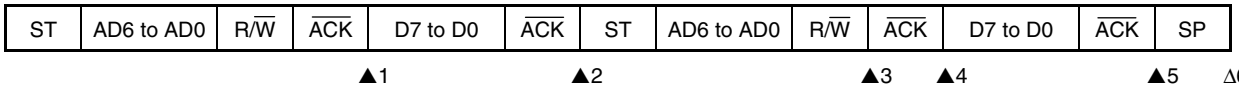
Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

<2> When WTIM0 bit = 1 (after restart, extension code reception)



▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001XX00B

▲3: IICS0 register = 0010X010B

▲4: IICS0 register = 0010X110B

▲5: IICS0 register = 0010XX00B

Δ 6: IICS0 register = 00000001B

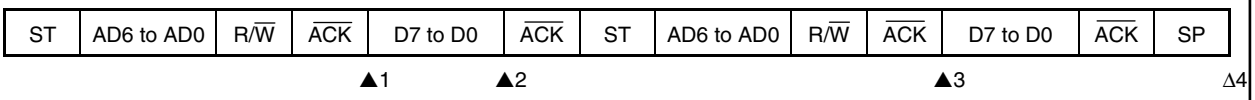
Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

(4) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, address mismatch (= not extension code))



▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 00000110B

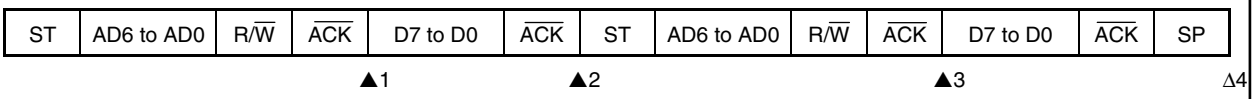
Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

<2> When WTIM0 bit = 1 (after restart, address mismatch (= not extension code))



▲1: IICS0 register = 0001X110B

▲2: IICS0 register = 0001XX00B

▲3: IICS0 register = 00000110B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

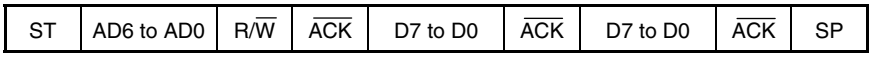
X: don't care

16.7.3 Slave device operation (when receiving extension code)

Always under communication when receiving the extension code.

(1) Start ~ Code ~ Data ~ Data ~ Stop

<1> When IICC0.WTIM0 bit = 0



▲1 ▲2 ▲3 Δ4

▲1: IICS0 register = 0010X010B

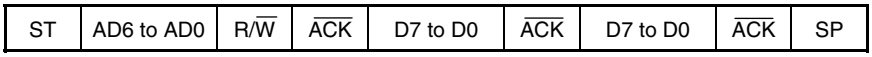
▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0010X000B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when IICC0.SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010X100B

▲4: IICS0 register = 0010XX00B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(2) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, address match)

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0001X110B

▲4: IICS0 register = 0001X000B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1 (after restart, address match)

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 ▲4 ▲5 Δ6

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010XX00B

▲4: IICS0 register = 0001X110B

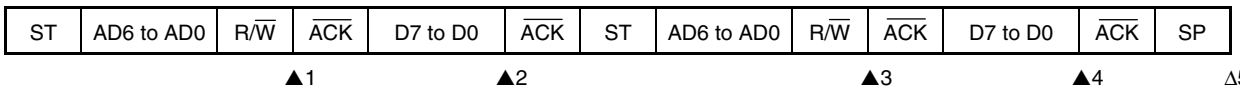
▲5: IICS0 register = 0001XX00B

Δ 6: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(3) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, extension code reception)



▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0010X010B

▲4: IICS0 register = 0010X000B

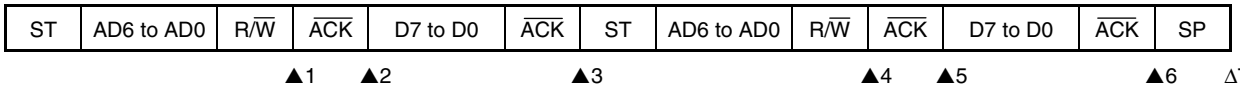
Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

<2> When WTIM0 bit = 1 (after restart, extension code reception)



▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010XX00B

▲4: IICS0 register = 0010X010B

▲5: IICS0 register = 0010X110B

▲6: IICS0 register = 0010XX00B

Δ 7: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

(4) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When WTIM0 bit = 0 (after restart, address mismatch (= not extension code))

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 00000110B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

<2> When WTIM0 bit = 1 (after restart, address mismatch (= not extension code))

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICS0 register = 0010X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010XX00B

▲4: IICS0 register = 00000110B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

16.7.4 Operation without communication

(1) Start ~ Code ~ Data ~ Data ~ Stop

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

Δ1

Δ 1: IICS0 register = 00000001B

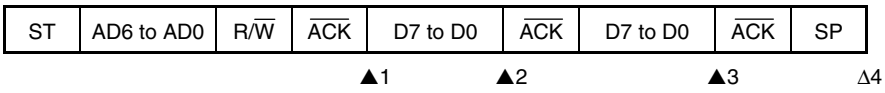
Remark Δ: Generated only when IICC0.SPIE0 bit = 1

16.7.5 Arbitration loss operation (operation as slave after arbitration loss)

When used as master in the multi-master system, check the arbitration result by reading the IICS0.MSTS0 bit for checking arbitration result by each INTIIC0 interrupt occurrence.

(1) When arbitration loss occurs during transmission of slave address data

<1> When IICC0.WTIM0 bit = 0



▲1: IICS0 register = 0101X110B

▲2: IICS0 register = 0001X000B

▲3: IICS0 register = 0001X000B

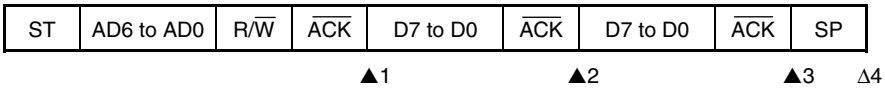
Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when IICC0.SPIE0 bit = 1

X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 0101X110B

▲2: IICS0 register = 0001X100B

▲3: IICS0 register = 0001XX00B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

(2) When arbitration loss occurs during transmission of extension code

<1> When WTIM0 bit = 0

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 Δ4

▲1: IICS0 register = 0110X010B

▲2: IICS0 register = 0010X000B

▲3: IICS0 register = 0010X000B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 ▲3 ▲4 Δ5

▲1: IICS0 register = 0110X010B

▲2: IICS0 register = 0010X110B

▲3: IICS0 register = 0010X100B

▲4: IICS0 register = 0010XX00B

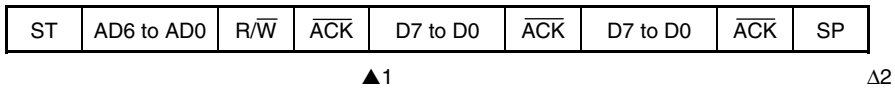
Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

16.7.6 Operation when arbitration loss occurs (no communication after arbitration loss)

When used as master in the multi-master system, check the arbitration result by reading the IICS0.MSTS0 bit for checking arbitration result by each INTIIC0 interrupt occurrence.

(1) When arbitration loss occurs during transmission of slave address data



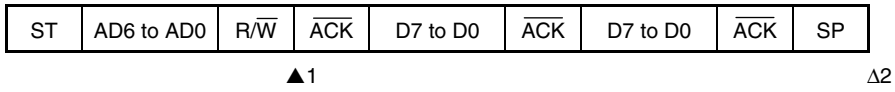
▲1: IICS0 register = 01000110B

Δ 2: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when IICC0.SPIE0 bit = 1

(2) When arbitration loss occurs during transmission of extension code



▲1: IICS0 register = 0110X010B

IICC0.LREL0 bit is set to 1 by software

Δ 2: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

X: don't care

(3) When arbitration loss occurs during data transfer

<1> When IICC0.WTIM0 bit = 0

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 Δ3

▲1: IICS0 register = 10001110B

▲2: IICS0 register = 01000000B

Δ 3: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

<2> When WTIM0 bit = 1

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP

▲1 ▲2 Δ3

▲1: IICS0 register = 10001110B

▲2: IICS0 register = 01000100B

Δ 3: IICS0 register = 00000001B

Remark ▲: Always generated

Δ: Generated only when SPIE0 bit = 1

(4) When arbitration loss occurs due to restart condition during data transfer

<1> Not extension code (Example: Address mismatch)

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to Dn	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1					▲2			Δ3

▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 01000110B

Δ 3: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care
Dn = D6 to D0

<2> Extension code

ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to Dn	ST	AD6 to AD0	R/$\overline{W}$	$\overline{ACK}$	D7 to D0	$\overline{ACK}$	SP
			▲1					▲2			Δ3

▲1: IICS0 register = 1000X110B

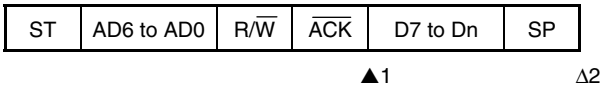
▲2: IICS0 register = 0110X010B

IICC0.LREL0 bit is set to 1 by software

Δ 3: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care
Dn = D6 to D0

(5) When arbitration loss occurs due to stop condition during data transfer



▲1: IICS0 register = 1000X110B

Δ 2: IICS0 register = 01000001B

Remark ▲: Always generated

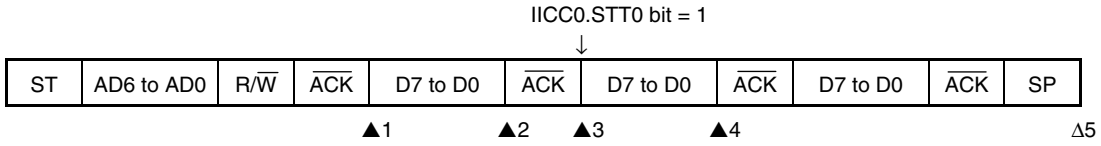
Δ: Generated only when SPIE0 bit = 1

X: don't care

Dn = D6 to D0

(6) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

<1> When WTIM0 bit = 0



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X000B (WTIM0 bit = 1)

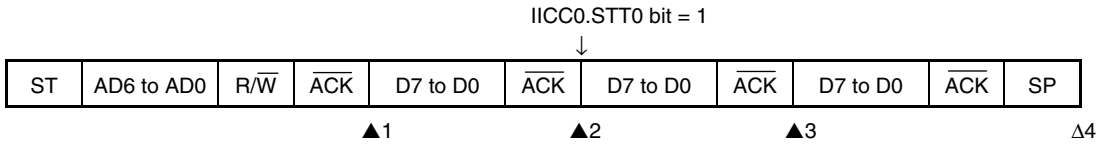
▲3: IICS0 register = 1000X100B (WTIM0 bit = 0)

▲4: IICS0 register = 01000000B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X100B

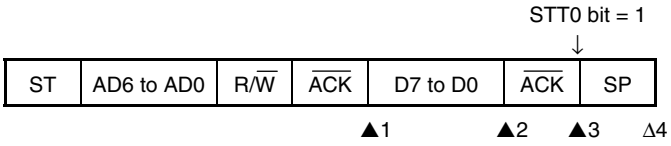
▲3: IICS0 register = 01000100B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(7) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

<1> When WTIM0 bit = 0



▲1: IICS0 register = 1000X110B

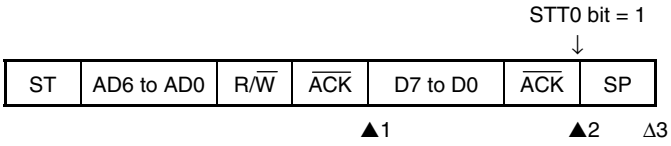
▲2: IICS0 register = 1000X000B (WTIM0 bit = 1)

▲3: IICS0 register = 1000XX00B

Δ 4: IICS0 register = 01000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 1000X110B

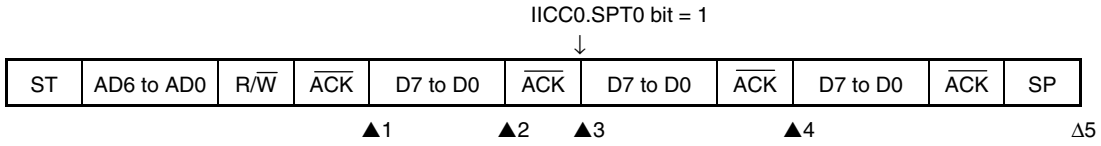
▲2: IICS0 register = 1000XX00B

Δ 3: IICS0 register = 01000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

(8) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

<1> When WTIM0 bit = 0



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X000B (WTIM0 bit = 1)

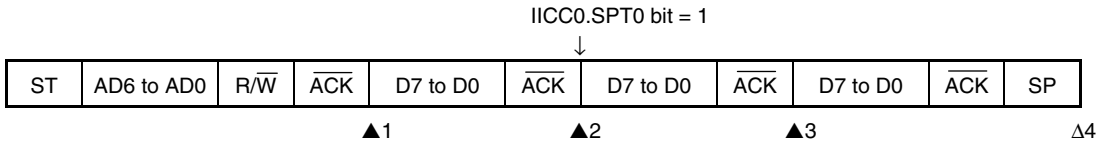
▲3: IICS0 register = 1000X100B (WTIM0 bit = 0)

▲4: IICS0 register = 01000100B

Δ 5: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

<2> When WTIM0 bit = 1



▲1: IICS0 register = 1000X110B

▲2: IICS0 register = 1000X100B

▲3: IICS0 register = 01000100B

Δ 4: IICS0 register = 00000001B

Remark ▲: Always generated
Δ: Generated only when SPIE0 bit = 1
X: don't care

16.8 Interrupt Request Signal (INTIIC0) Generation Timing and Wait Control

The setting of the IICC0.WTIM0 bit determines the timing by which the INTIIC0 signal is generated and the corresponding wait control, as shown below.

Table 16-3. INTIIC0 Signal Generation Timing and Wait Control

WTIM0 Bit	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9 Notes 1, 2	8 Note 2	8 Note 2	9	8	8
1	9 Notes 1, 2	9 Note 2	9 Note 2	9	9	9

Notes 1. The slave device's INTIIC0 signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to the SVA0 register.
At this point, $\overline{ACK}$ is generated regardless of the value set to the IICC0.ACKE0 bit. For a slave device that has received an extension code, the INTIIC0 signal occurs at the falling edge of the eighth clock.
When the address does not match after restart, the INTIIC0 signal is generated at the falling edge of the ninth clock, but no wait occurs.

2. If the received address does not match the contents of the SVA0 register and extension codes have not been received, neither the INTIIC0 signal nor a wait occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- Slave device operation: Interrupt and wait timing are determined depending on the conditions in Notes 1 and 2 above regardless of the WTIM0 bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM0 bit.

(2) During data reception

Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(3) During data transmission

Master/slave device operation: Interrupt and wait timing are determined according to the WTIM0 bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- By writing data to the IIC0 register
- By setting the IICC0.WREL0 bit (canceling wait state)
- By setting the IICC0.STT0 bit (generating start condition) Note
- By setting the IICC0.SPT0 bit (generating stop condition) Note

Note Master only

When an 8-clock wait has been selected (WTIM0 bit = 0), whether or not $\overline{ACK}$ has been generated must be determined prior to wait cancellation.

(5) Stop condition detection

The INTIIC0 signal is generated when a stop condition is detected.

16.9 Address Match Detection Method

When in I²C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match detection is performed automatically by hardware. An INTIIC0 interrupt request signal occurs when a local address has been set to the SVA0 register and when the address set to the SVA0 register matches the slave address sent by the master device, or when an extension code has been received.

16.10 Error Detection

In I²C bus mode, the status of the serial data bus (SDA00) during data transmission is captured by the IIC0 register of the transmitting device, so the IIC0 register data prior to transmission can be compared with the transmitted IIC0 register data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

16.11 Extension Code

(1) When the higher 4 bits of the receive address are either 0000 or 1111, the extension code flag (EXC0) is set for extension code reception and an interrupt request signal (INTIIC0) is issued at the falling edge of the eighth clock.
The local address stored in the SVA0 register is not affected.

(2) If 11110xx0 is set to the SVA0 register by a 10-bit address transfer and 11110xx0 is transferred from the master device, the results are as follows. Note that the INTIIC0 signal occurs at the falling edge of the eighth clock.

- Higher 4 bits of data match: IICS0.EXC0 bit = 1
- 7 bits of data match: IICS0.COI0 bit = 1

(3) Since the processing after the INTIIC0 signal occurs differs according to the data that follows the extension code, such processing is performed by software. The slave that has received an extension code is always under communication, even if the addresses mismatch.
For example, when operation as a slave is not desired after the extension code is received, set the IICC0.LREL0 bit to 1 and the CPU will enter the next communication wait state.

Table 16-4. Extension Code Bit Definitions

Slave Address	R/$\overline{W}$ Bit	Description
0000 000	0	General call address
0000 000	1	Start byte
0000 001	X	CBUS address
0000 010	X	Address that is reserved for different bus format
1111 0xx	X	10-bit slave address specification

16.12 Arbitration

When several master devices simultaneously generate a start condition (when the IICC0.STT0 bit is set to 1 before the IICS0.STD0 bit is set to 1), communication among the master devices is performed as the number of clocks is adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (IICS0.ALD0 bit) is set (1) via the timing by which the arbitration loss occurred, and the SCL00 and SDA00 lines are both set for high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request signal (INTIIC0) (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD0 bit = 1 setting that has been made by software.

For details of interrupt request timing, see **16.7 I²C Interrupt Request Signals (INTIIC0)**.

Figure 16-12. Arbitration Timing Example

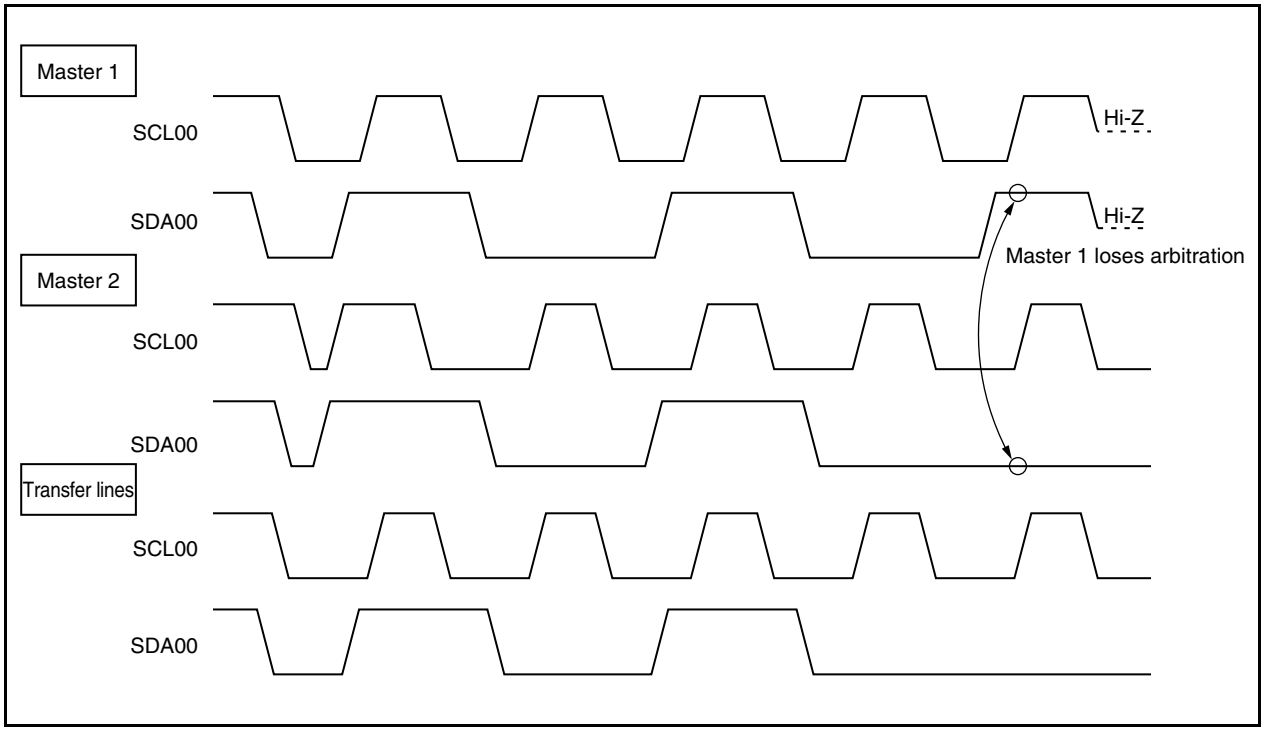


Table 16-5. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{ACK}$ transfer period after data reception	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when IICC0.SPIE0 bit = 1)[Note 2]
When the SDA00 pin is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIE0 bit = 1)[Note 2]
When the SDA00 pin is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When the SCL00 pin is at low level while attempting to generate a restart condition	

Notes 1. When the IICC0.WTIM0 bit = 1, an interrupt request occurs at the falling edge of the ninth clock. When the WTIM0 bit = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

2. When there is a possibility that arbitration will occur, set the SPIE0 bit = 1 for master device operation.

16.13 Wakeup Function

The I²C bus slave function is a function that generates an interrupt request signal (INTIIC0) when a local address or extension code has been received.

This function makes processing more efficient by preventing unnecessary interrupt requests from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

However, when a stop condition is detected, the IICC0.SPIE0 bit is set regardless of the wake up function, and this determines whether interrupt requests are enabled or disabled.

16.14 Communication Reservation

16.14.1 When communication reservation function is enabled (IICF0.IICRSV0 bit = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released when the IICC0.LREL0 bit was set to “1”).

If the IICC0.STT0 bit is set (1) while the bus is not used, a start condition is automatically generated and wait status is set after the bus is released (after a stop condition is detected).

A communication is automatically started as the master by setting the IICC0.SPIE0 bit to 1, detecting the bus release due to an interrupt request (INTIIC0) occurrence (detecting a stop condition), and then writing the address to the IIC0 register. Before detecting a stop condition, data written to the IIC0 register is set to invalid.

When the STT0 bit has been set (1), the operation mode (as start condition or as communication reservation) is determined according to the bus status.

If the bus has been released ..... a start condition is generated
If the bus has not been released (standby mode) ..... communication reservation

To detect which operation mode has been determined for the STT0 bit, set the STT0 bit (1), wait for the wait period, then check the IICS0.MSTS0 bit.

Wait periods, which should be set via software, are listed in Table 16-6. These wait periods can be set via the settings for the IICX0.CLX0, IICCL0.SMC0, IICCL0.CL01, and IICCL0.CL00 bits.

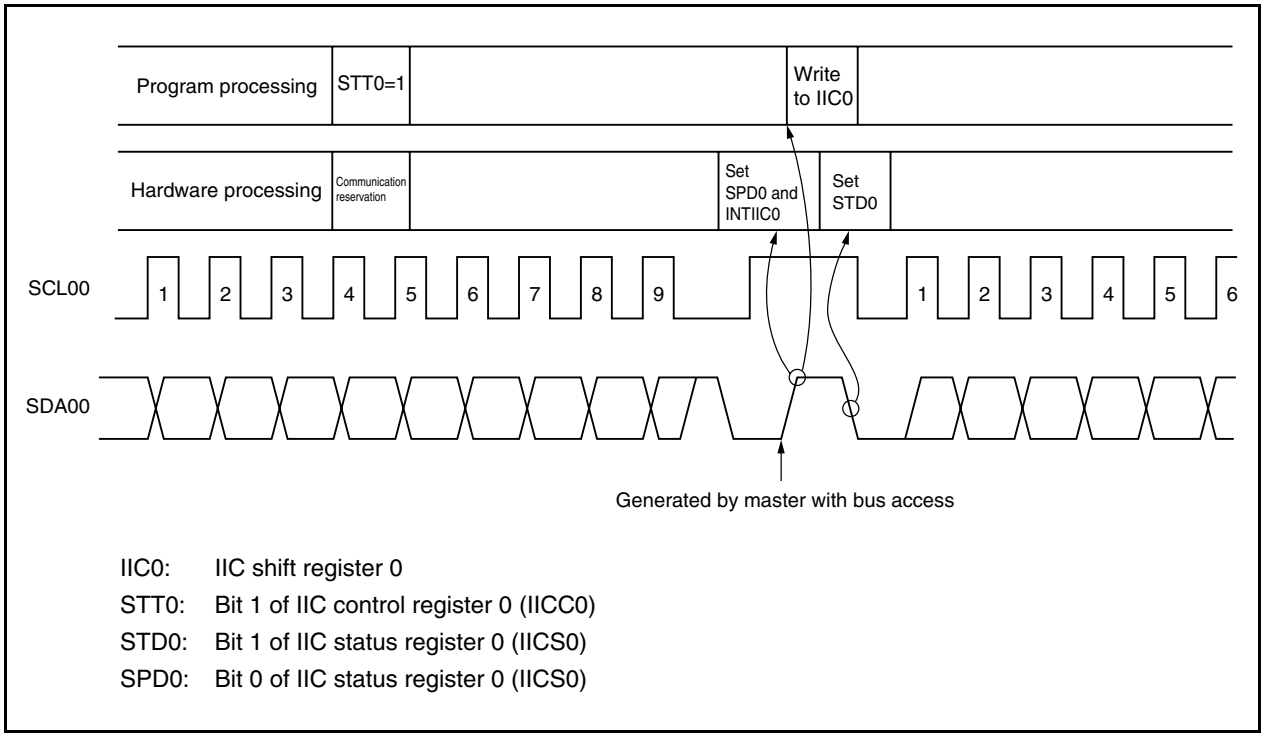
Table 16-6. Wait Periods

Selected Clock	CLX0	SMC0	CL01	CL00	Wait Period
fxx (when OCKS0 = 18H set)	0	0	0	0	26 clocks
fxx/2 (when OCKS0 = 10H set)	0	0	0	0	52 clocks
fxx/3 (when OCKS0 = 11H set)	0	0	0	0	78 clocks
fxx/4 (when OCKS0 = 12H set)	0	0	0	0	104 clocks
fxx/5 (when OCKS0 = 13H set)	0	0	0	0	130 clocks
fxx/2 (when OCKS0 = 10H set)	0	0	0	1	94 clocks
fxx/3 (when OCKS0 = 11H set)	0	0	0	1	141 clocks
fxx/4 (when OCKS0 = 12H set)	0	0	0	1	188 clocks
fxx/5 (when OCKS0 = 13H set)	0	0	0	1	253 clocks
fxx	0	0	1	0	47 clocks
fxx (when OCKS0 = 18H set)	0	0	1	1	37 clocks
fxx/2 (when OCKS0 = 10H set)	0	0	1	1	74 clocks
fxx/3 (when OCKS0 = 11H set)	0	0	1	1	111 clocks
fxx/4 (when OCKS0 = 12H set)	0	0	1	1	148 clocks
fxx/5 (when OCKS0 = 13H set)	0	0	1	1	185 clocks
fxx/2 (when OCKS0 = 10H set)	0	1	0	×	32 clocks
fxx/3 (when OCKS0 = 11H set)	0	1	0	×	48 clocks
fxx/4 (when OCKS0 = 12H set)	0	1	0	×	64 clocks
fxx/5 (when OCKS0 = 13H set)	0	1	0	×	80 clocks
fxx	0	1	1	0	16 clocks
fxx (when OCKS0 = 18H set)	0	1	1	1	13 clocks
fxx/2 (when OCKS0 = 10H set)	0	1	1	1	26 clocks
fxx/3 (when OCKS0 = 11H set)	0	1	1	1	39 clocks
fxx/4 (when OCKS0 = 12H set)	0	1	1	1	52 clocks
fxx/5 (when OCKS0 = 13H set)	0	1	1	1	65 clocks
fxx/2 (when OCKS0 = 10H set)	1	1	0	×	20 clocks
fxx/3 (when OCKS0 = 11H set)	1	1	0	×	30 clocks
fxx/4 (when OCKS0 = 12H set)	1	1	0	×	40 clocks
fxx/5 (when OCKS0 = 13H set)	1	1	0	×	50 clocks
fxx	1	1	1	0	10 clocks

Remark ×: don't care

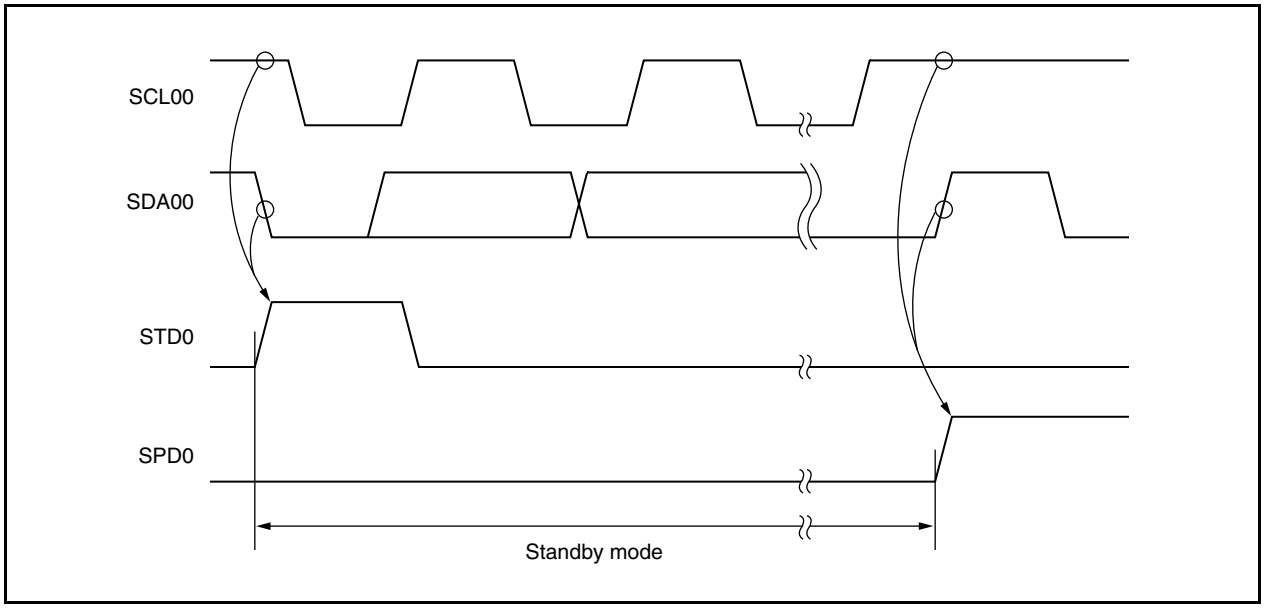
The communication reservation timing is shown below.

Figure 16-13. Communication Reservation Timing



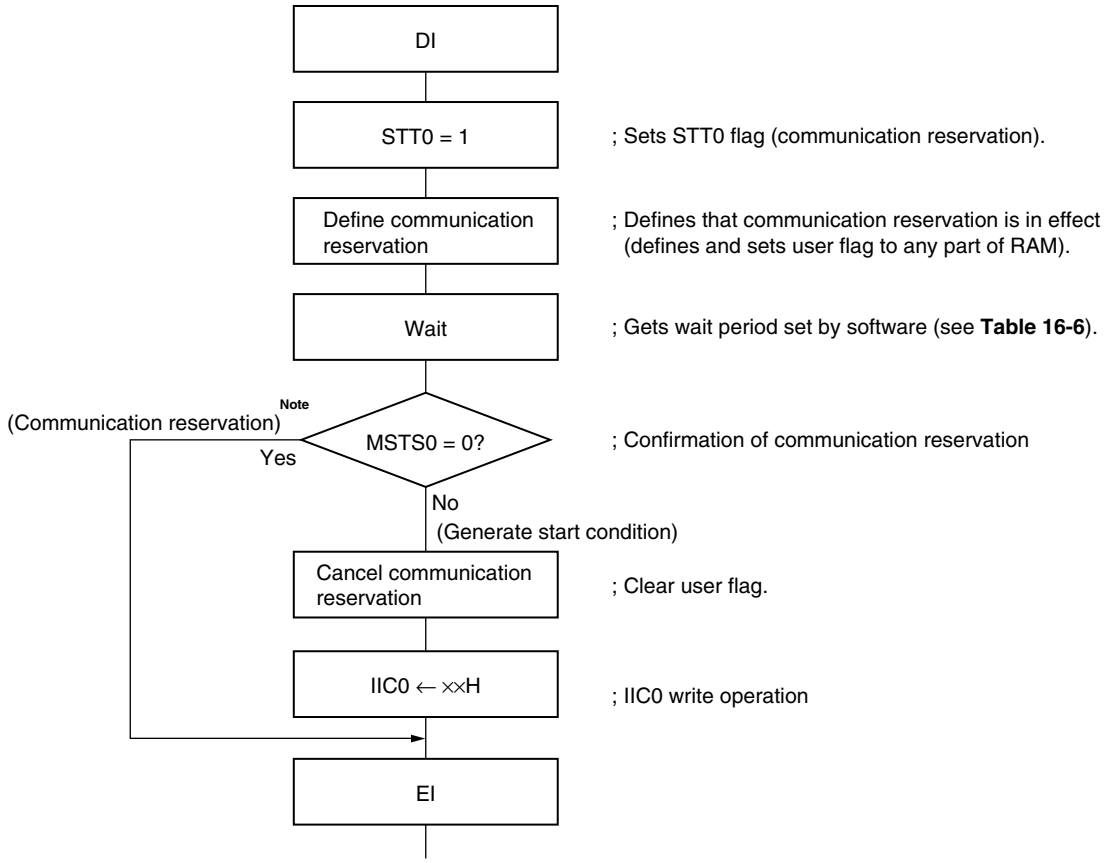
Communication reservations are accepted via the following timing. After the IICS0.STD0 bit is set to 1, a communication reservation can be made by setting the IICC0.STT0 bit to 1 before a stop condition is detected.

Figure 16-14. Timing for Accepting Communication Reservations



The communication reservation flowchart is illustrated below.

Figure 16-15. Communication Reservation Flowchart



Note The communication reservation operation executes a write to the IIC0 register when a stop condition interrupt occurs.

16.14.2 When communication reservation function is disabled (IICF0.IICRSV0 bit = 1)

When the IICC0.STT0 bit is set when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released when the IICC0.LREL0 bit was set to 1)

To confirm whether the start condition was generated or request was rejected, check the IICF0.STCF0 flag. The time shown in Table 16-7 is required until the STCF0 flag is set after setting the STT0 bit = 1. Therefore, secure the time by software.

Table 16-7. Wait Periods

OCKSEN0	OCKS01	OCKS00	CL01	CL00	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

Remark ×: don't care

16.15 Cautions

(1) When IICF0.STCEN0 bit = 0

Immediately after I²C00 operation is enabled, the bus communication status (IICF0.IICBSY0 bit = 1) is recognized regardless of the actual bus status. To execute master communication in the status where a stop condition has not been detected, generate a stop condition and then release the bus before starting the master communication.

Use the following sequence for generating a stop condition.

<1> Set the IICCL0 register.
<2> Set the IICC0.IICE0 bit.
<3> Set the IICC0.SPT0 bit.

(2) When IICF0.STCEN0 bit = 1

Immediately after I²C00 operation is enabled, the bus released status (IICBSY0 bit = 0) is recognized regardless of the actual bus status. To generate the first start condition (IICC0.STT0 bit = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) Determine the operation clock frequency by the IICCL0, IICX0, and OCKS0 registers before enabling the operation (IICC0.IICE0 bit = 1). To change the operation clock frequency, clear the IICC0.IICE0 bit to 0 once.

(4) After the IICC0.STT0 and IICC0.SPT0 bits have been set to 1, they must not be re-set without being cleared to 0 first.

(5) If transmission has been reserved, set the IICC0.SPIE0 bit to 1 so that an interrupt request is generated by the detection of a stop condition. After an interrupt request has been generated, the wait state will be released by writing communication data to I²C00, then transferring will begin. If an interrupt is not generated by the detection of a stop condition, transmission will halt in the wait state because an interrupt request was not generated. However, it is not necessary to set the SPIE0 bit to 1 for the software to detect the IICS0.MSTS0 bit.

16.16 Communication Operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the V850ES/Hx3 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 I²C00 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/Hx3 takes part in a communication with bus released state.

This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the V850ES/Hx3 looses in arbitration and is specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

(3) Slave operation

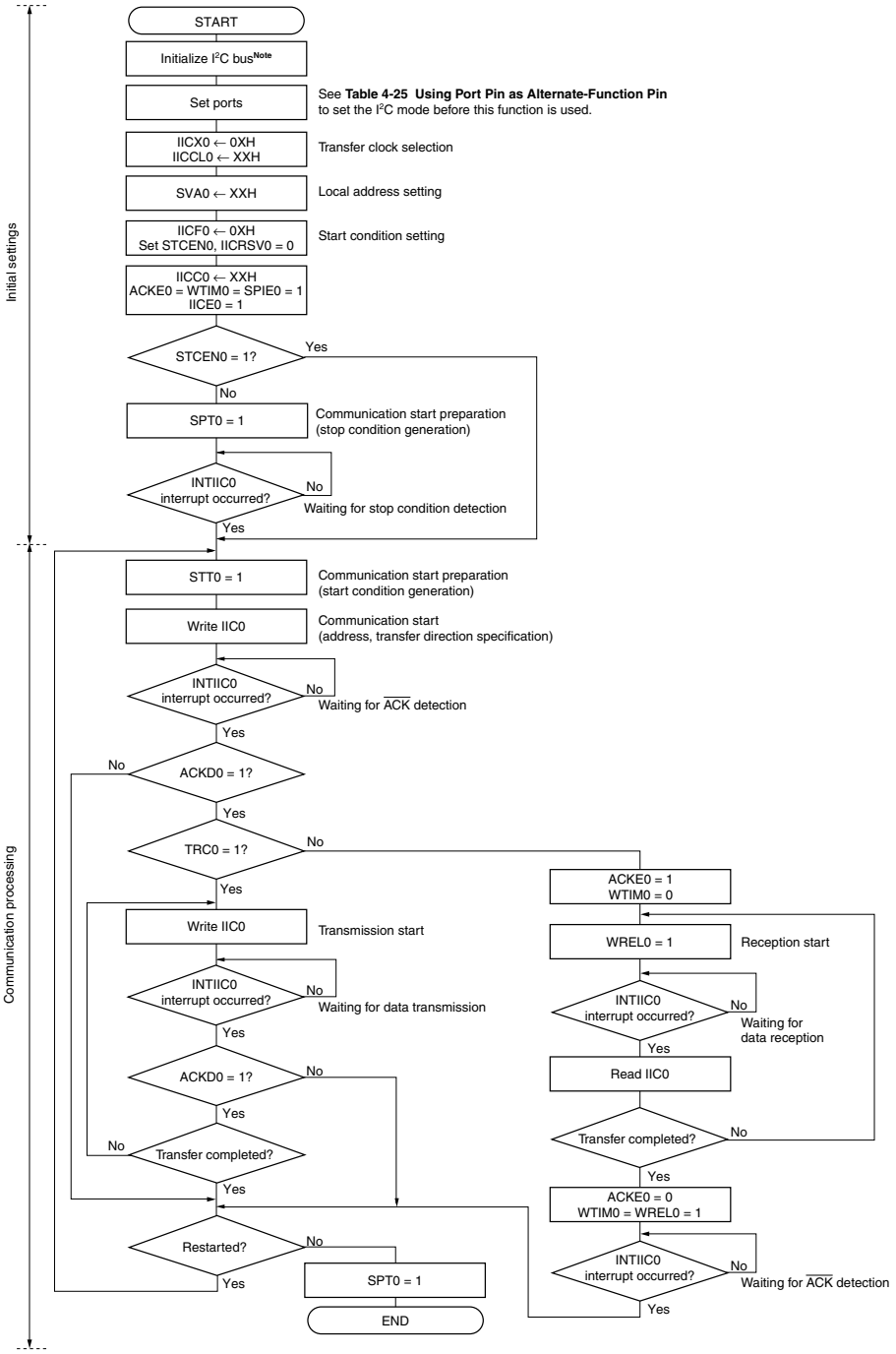
An example of when the I²C00 bus is used as the slave is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIIC0 interrupt occurrence (communication waiting). When the INTIIC0 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.

16.16.1 Master operation in single master system

Figure 16-16. Master Operation in Single Master System



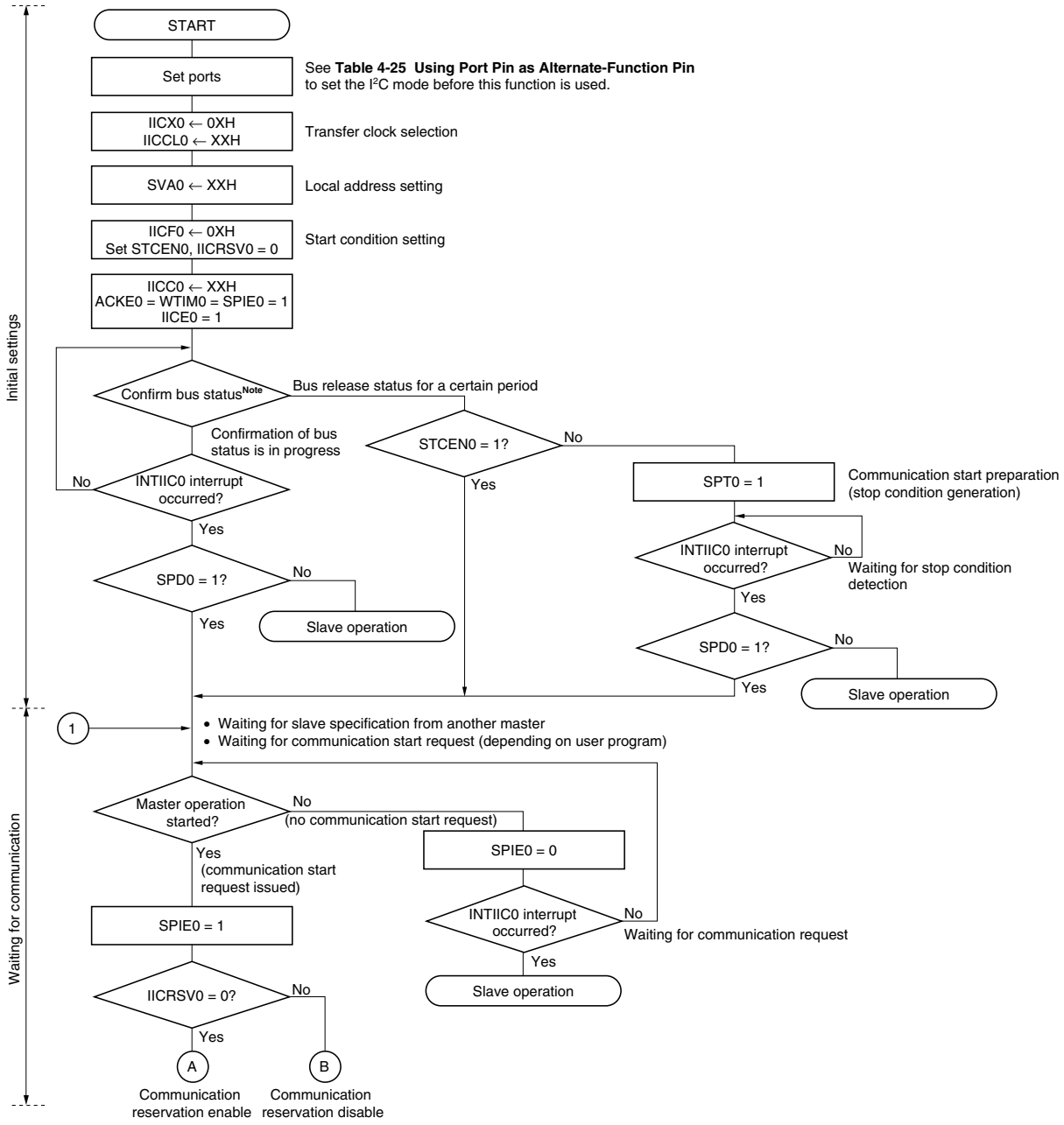
Note Release the I²C00 bus (SCL00, SDA00 pins = high level) in conformity with the specifications of the product in communication.

For example, when the EEPROM™ outputs a low level to the SDA00 pin, set the SCL00 pin to the output port and output clock pulses from that output port until when the SDA00 pin is constantly high level.

Remark For the transmission and reception formats, conform to the specifications of the product in communication.

16.16.2 Master operation in multimaster system

Figure 16-17. Master Operation in Multimaster System (1/3)



Note Confirm that the bus release status (IICCL0.CLD0 bit = 1, IICCL0.DAD0 bit = 1) has been maintained for a certain period (1 frame, for example). When the SDA00 pin is constantly low level, determine whether to release the I²C00 bus (SCL00, SDA00 pins = high level) by referring to the specifications of the product in communication.

Figure 16-17. Master Operation in Multimaster System (2/3)

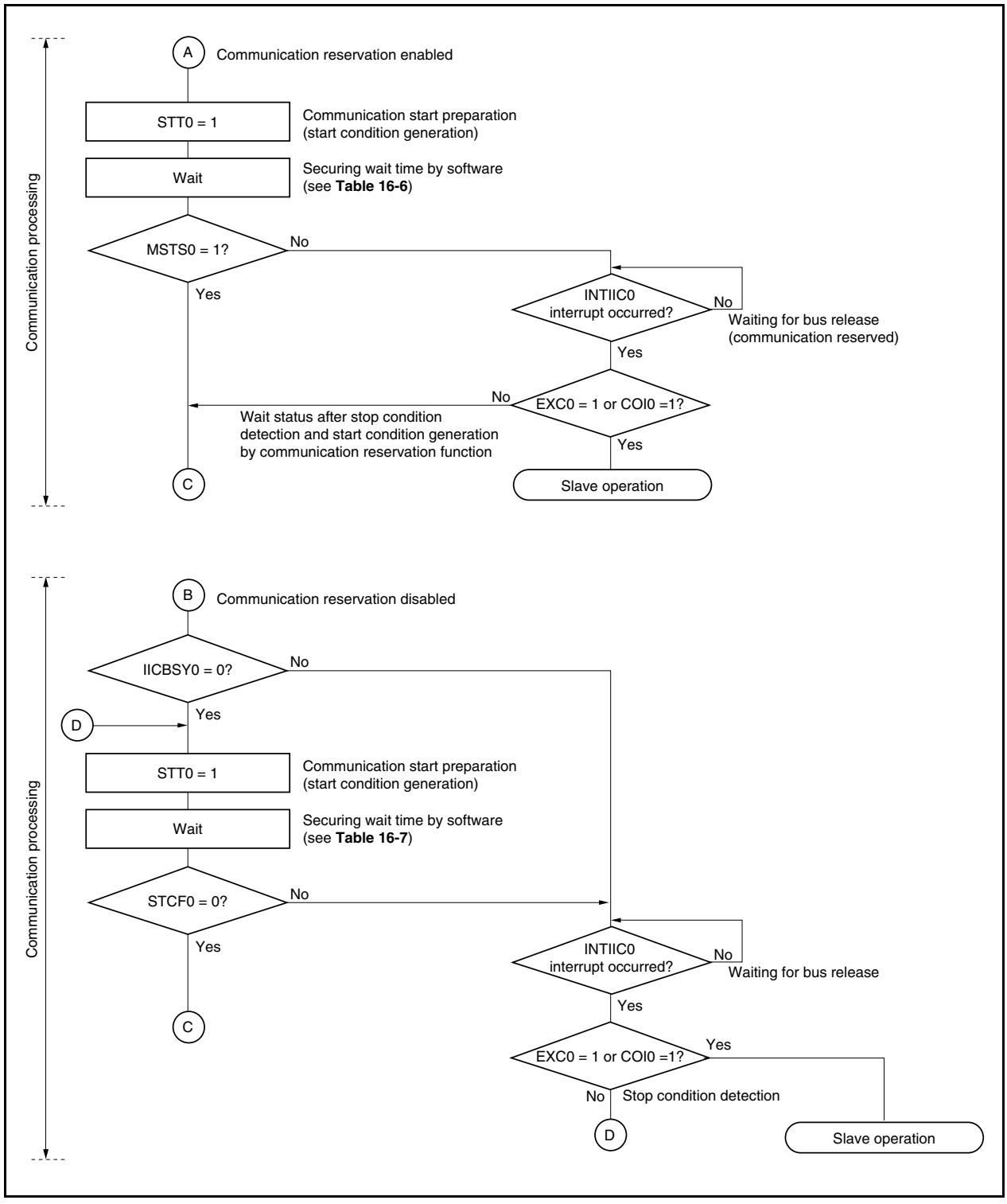
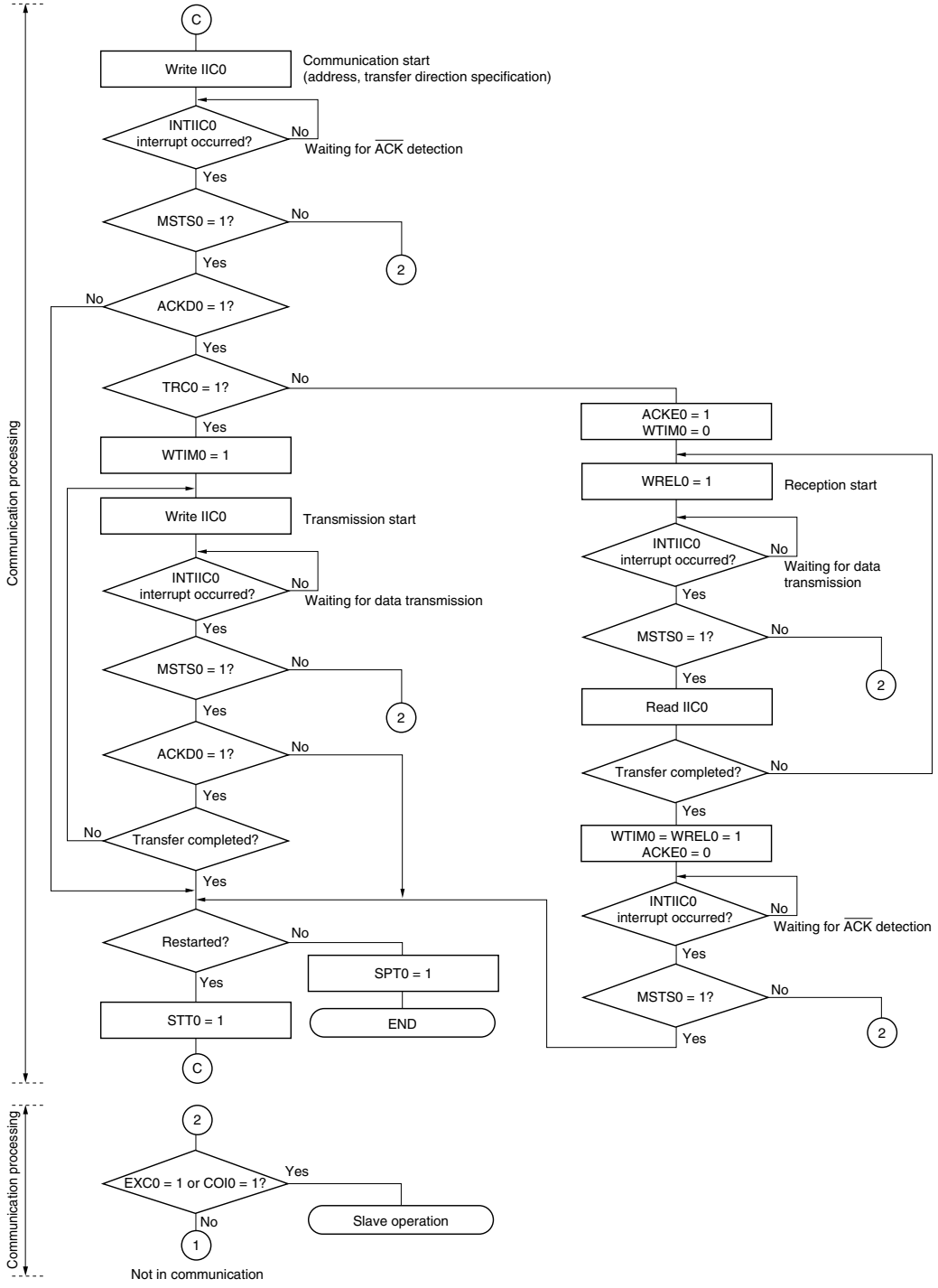


Figure 16-17. Master Operation in Multimaster System (3/3)



Remarks 1. Conform the transmission and reception formats to the specifications of the product in communication.

2. When using the V850ES/Hx3 as the master in the multimaster system, read the IICS0.MSTS0 bit for each INTIIC0 interrupt occurrence to confirm the arbitration result.

3. When using the V850ES/Hx3 as the slave in the multimaster system, confirm the status using the IICS0 and IICF0 registers for each INTIIC0 interrupt occurrence to determine the next processing.

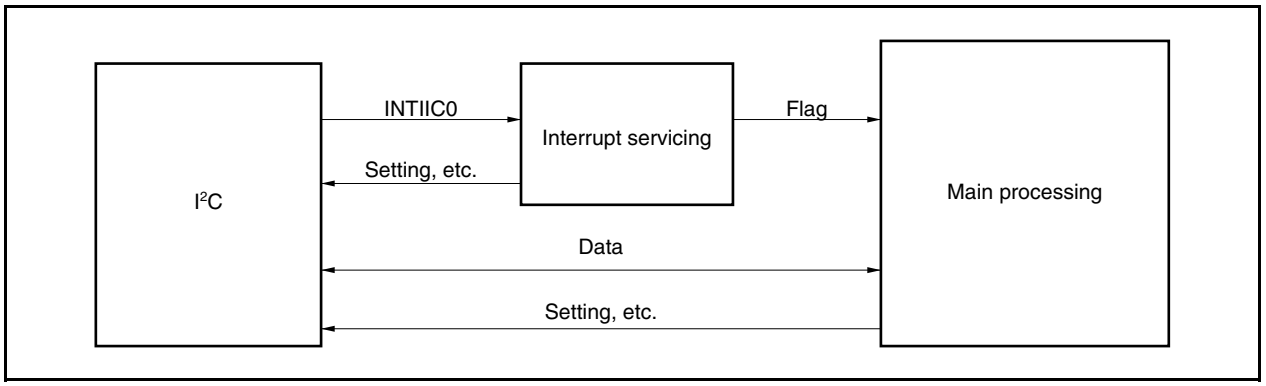
16.16.3 Slave operation

The following shows the processing procedure of the slave operation.

Basically, the operation of the slave device is event-driven. Therefore, processing by an INTIIC0 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 INTIIC0 interrupt servicing performs only status change processing and that the actual data communication is performed during the main processing.

Figure 16-18. Software Outline During Slave Operation



Therefore, the following three flags are prepared so that the data transfer processing can be performed by transmitting these flags to the main processing instead of the INTIIC0 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 INTIIC0 interrupt during normal data transfer. This flag is set in the interrupt servicing block and cleared in the main processing block. The ready flag for the first data for transmission is not set in the interrupt servicing block, so the first data is transmitted without clearance processing (the address match is regarded as a request for the next data).

(3) Communication direction flag

This flag indicates the direction of communication and is the same as the value of the IICS0.TRC0 bit.

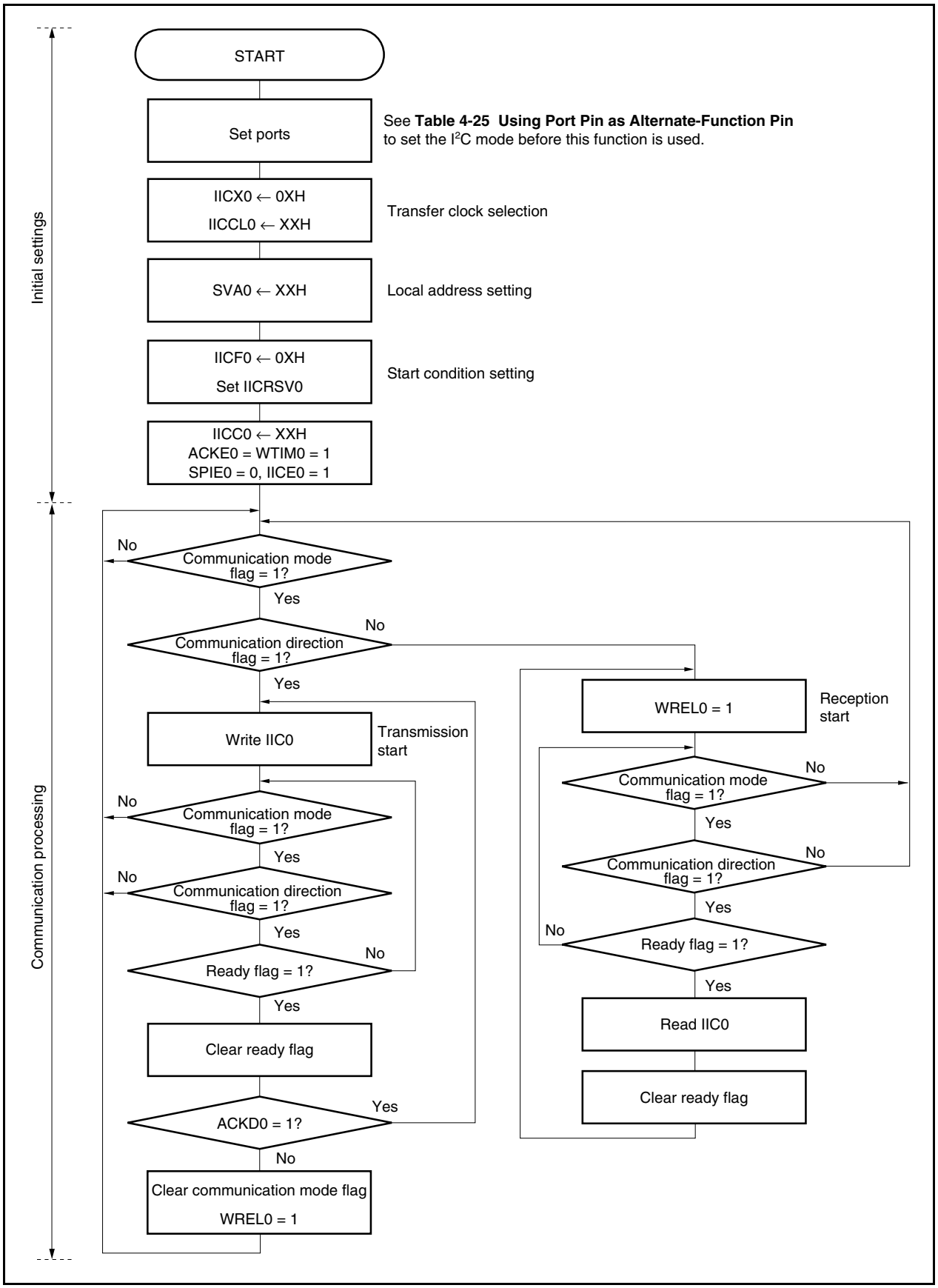
The following shows the operation of the main processing block during slave operation.

Start I²C00 and wait for the communication enabled status. When communication is enabled, perform transfer using the communication mode flag and ready flag (the processing of the stop condition and start condition is performed by interrupts, conditions are confirmed by flags).

For transmission, repeat the transmission operation until the master device stops returning $\overline{\text{ACK}}$. When the master device stops returning $\overline{\text{ACK}}$, transfer is complete.

For reception, receive the required number of data and do not return $\overline{\text{ACK}}$ for the next data immediately after transfer is complete. After that, the master device generates the stop condition or restart condition. This causes exit from communications.

Figure 16-19. Slave Operation Flowchart (1)

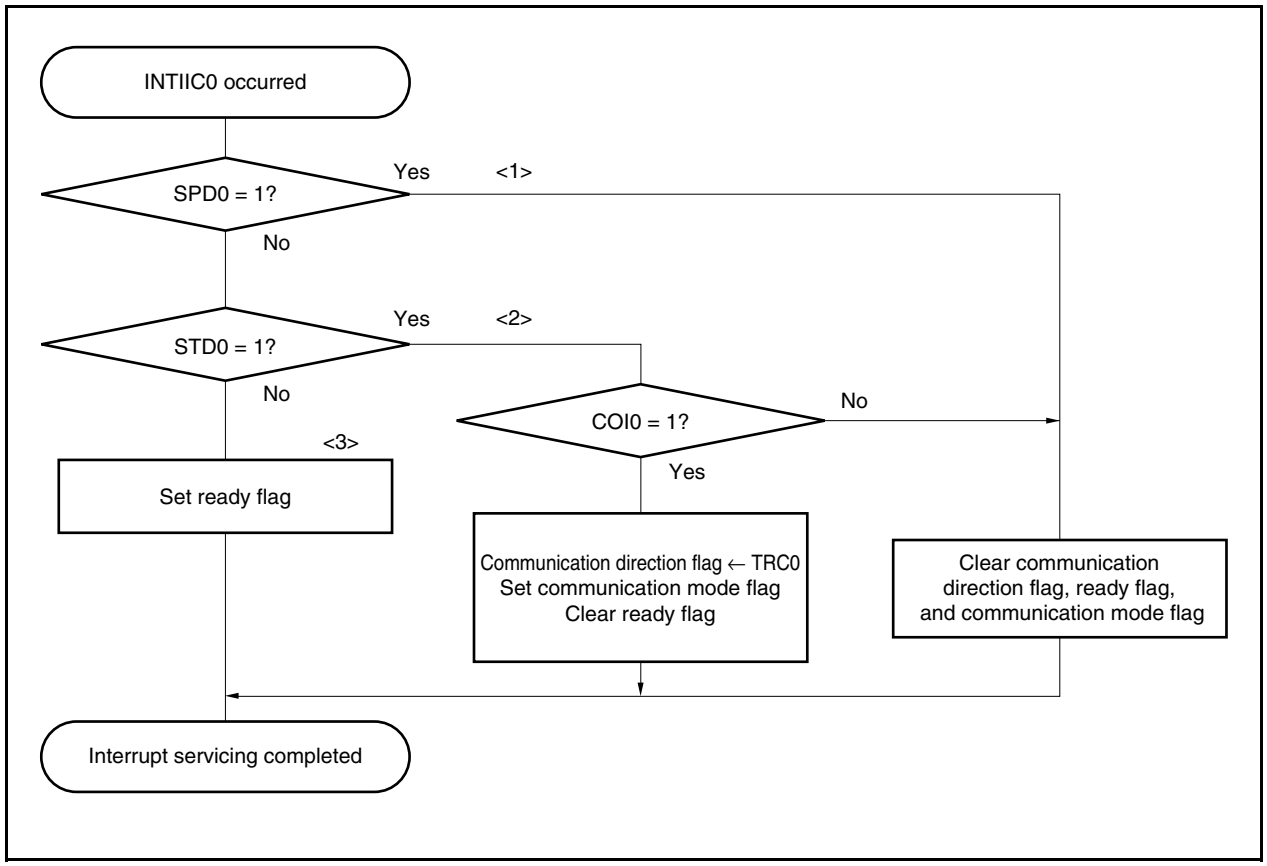


The following shows an example of the processing of the slave device by an INTIIC0 interrupt (it is assumed that no extension codes are used here). During an INTIIC0 interrupt, the status is confirmed and the following steps are executed.

<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 I²C00 bus remains in the wait status.

Remark <1> to <3> in the above correspond to <1> to <3> in **Figure 16-20 Slave Operation Flowchart (2)**.

Figure 16-20. Slave Operation Flowchart (2)



16.17 Timing of Data Communication

When using I²C bus mode, the master device generates an address via the serial bus to select one of several slave devices as its communication partner.

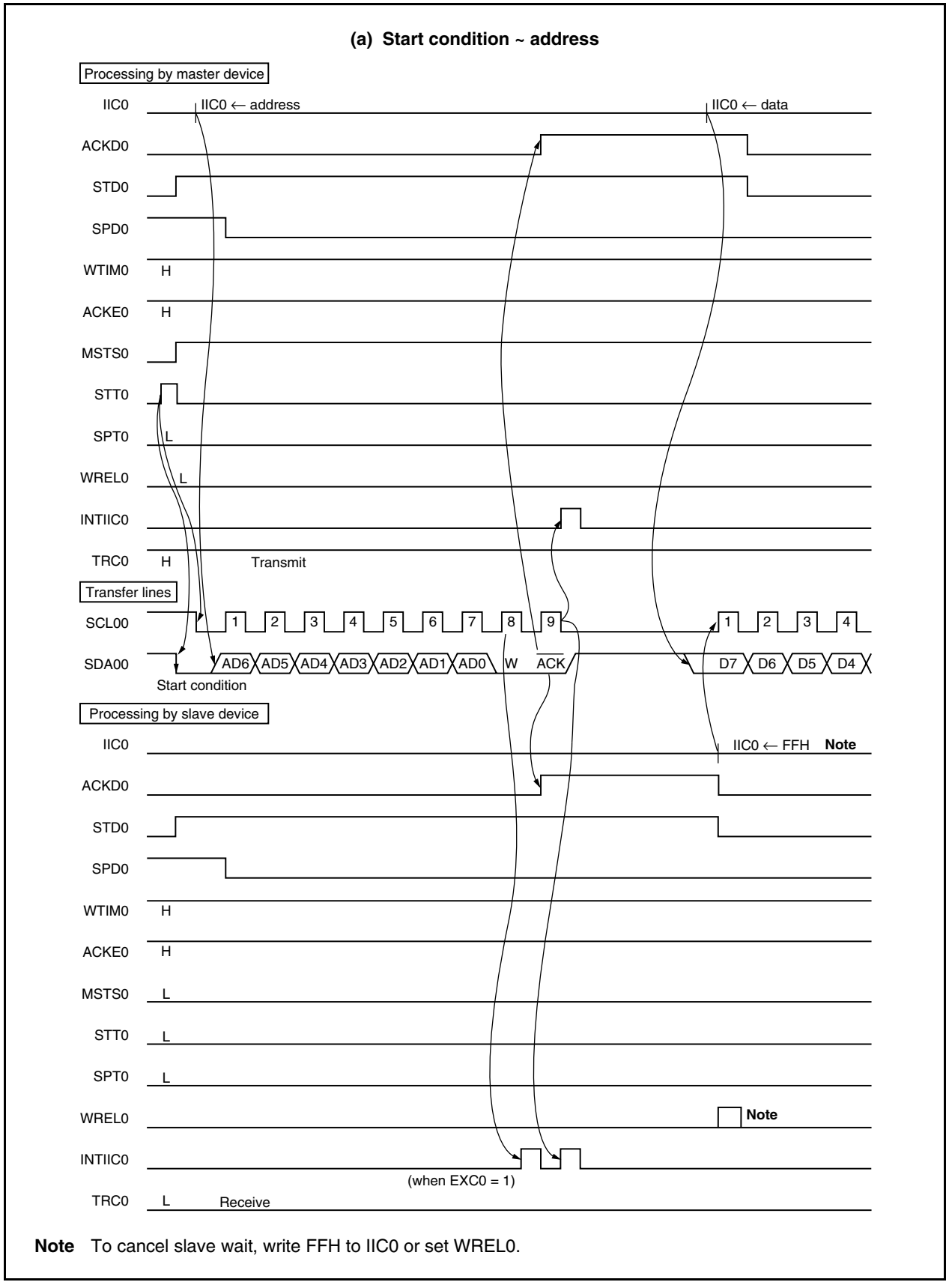
After outputting the slave address, the master device transmits the IICS0.TRC0 bit that specifies the data transfer direction and then starts serial communication with the slave device.

The IIC0 register's shift operation is synchronized with the falling edge of the serial clock (SCL00 pin). The transmit data is transferred to the SO latch and is output (MSB first) via the SDA00 pin.

Data input via the SDA00 pin is captured by the IIC0 register at the rising edge of the SCL00 pin.

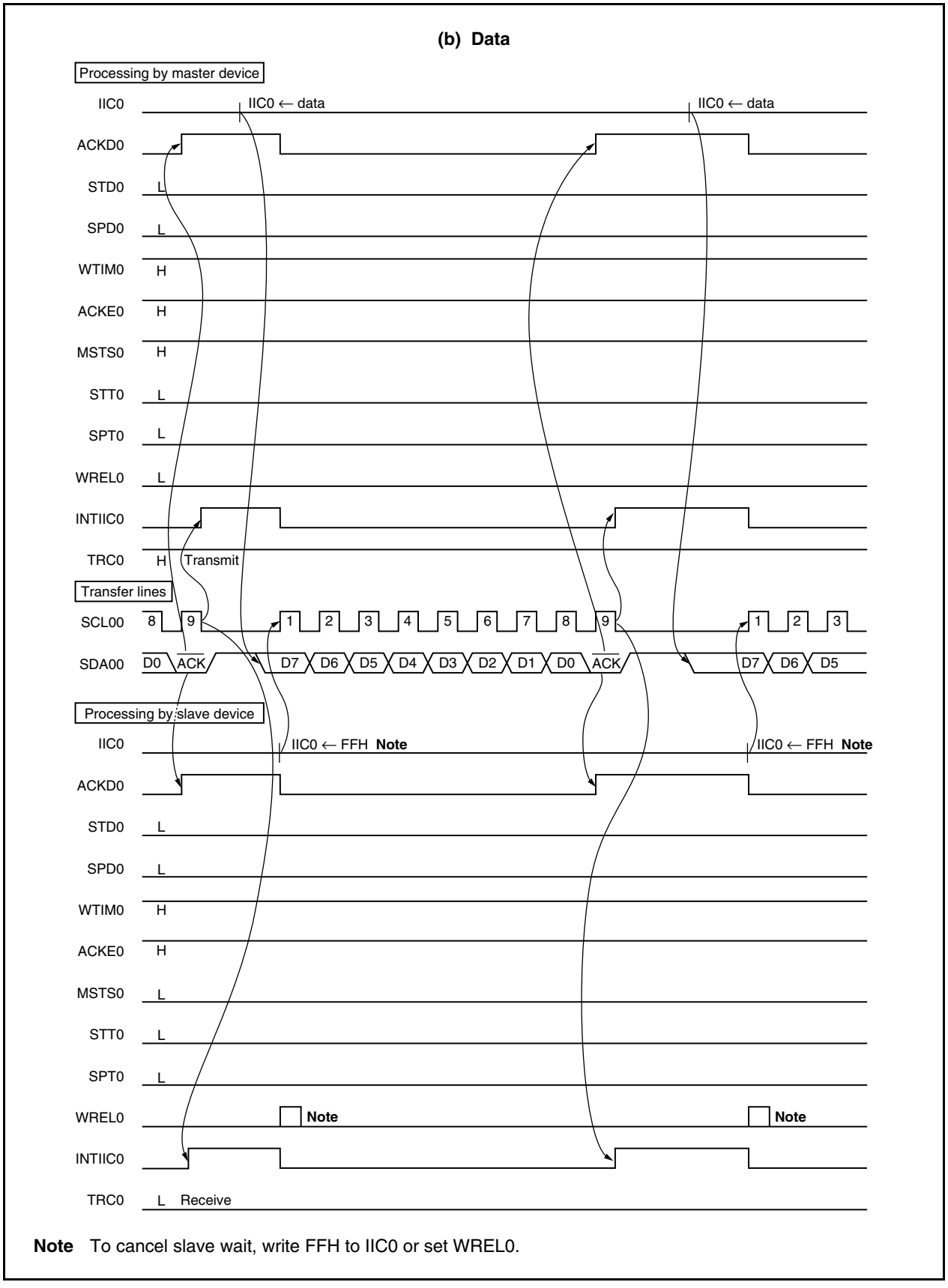
The data communication timing is shown below.

Figure 16-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (1/3)



Note To cancel slave wait, write FFH to IIC0 or set WREL0.

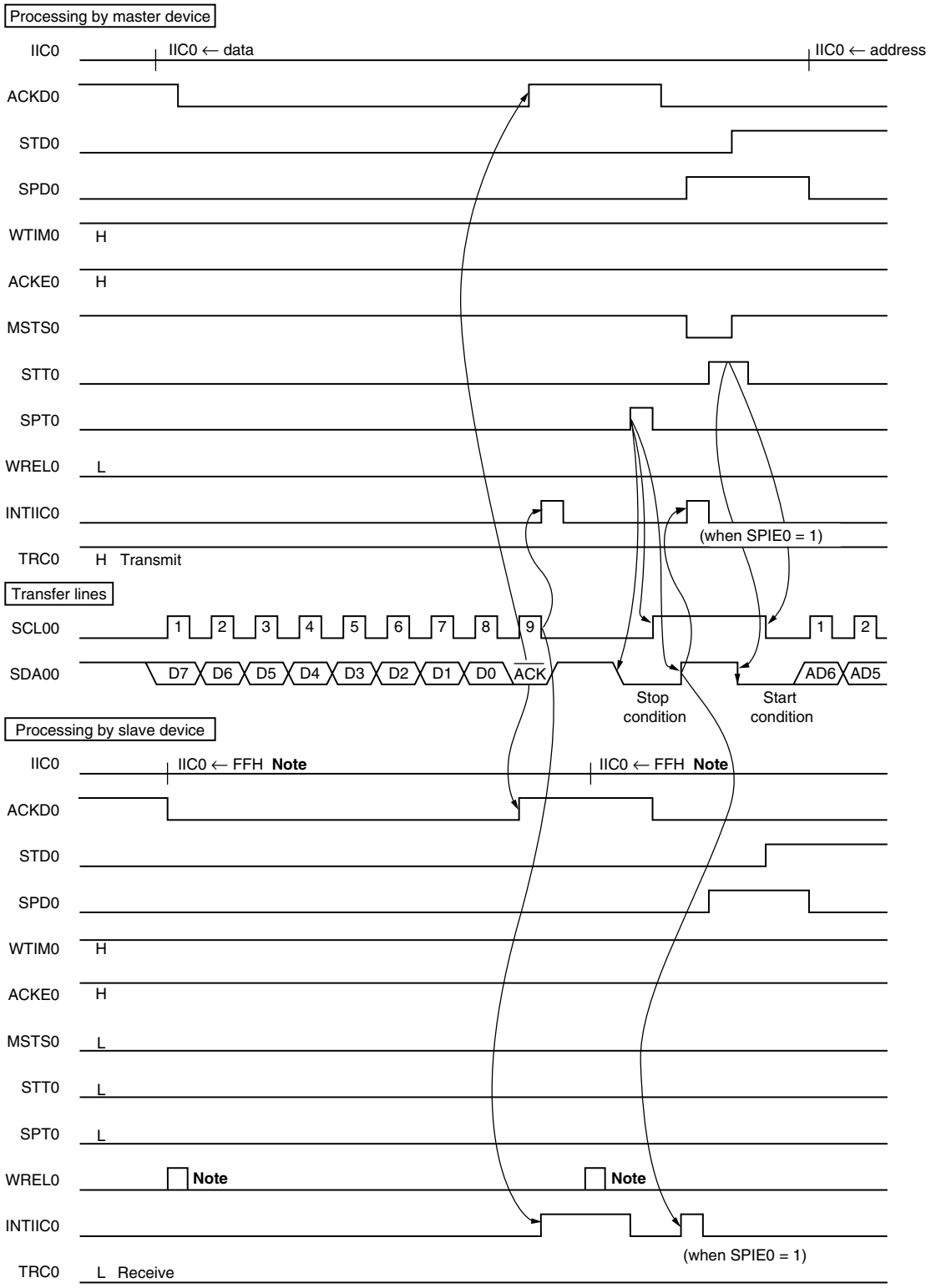
Figure 16-21. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Both Master and Slave) (2/3)



Note To cancel slave wait, write FFH to IIC0 or set WREL0.

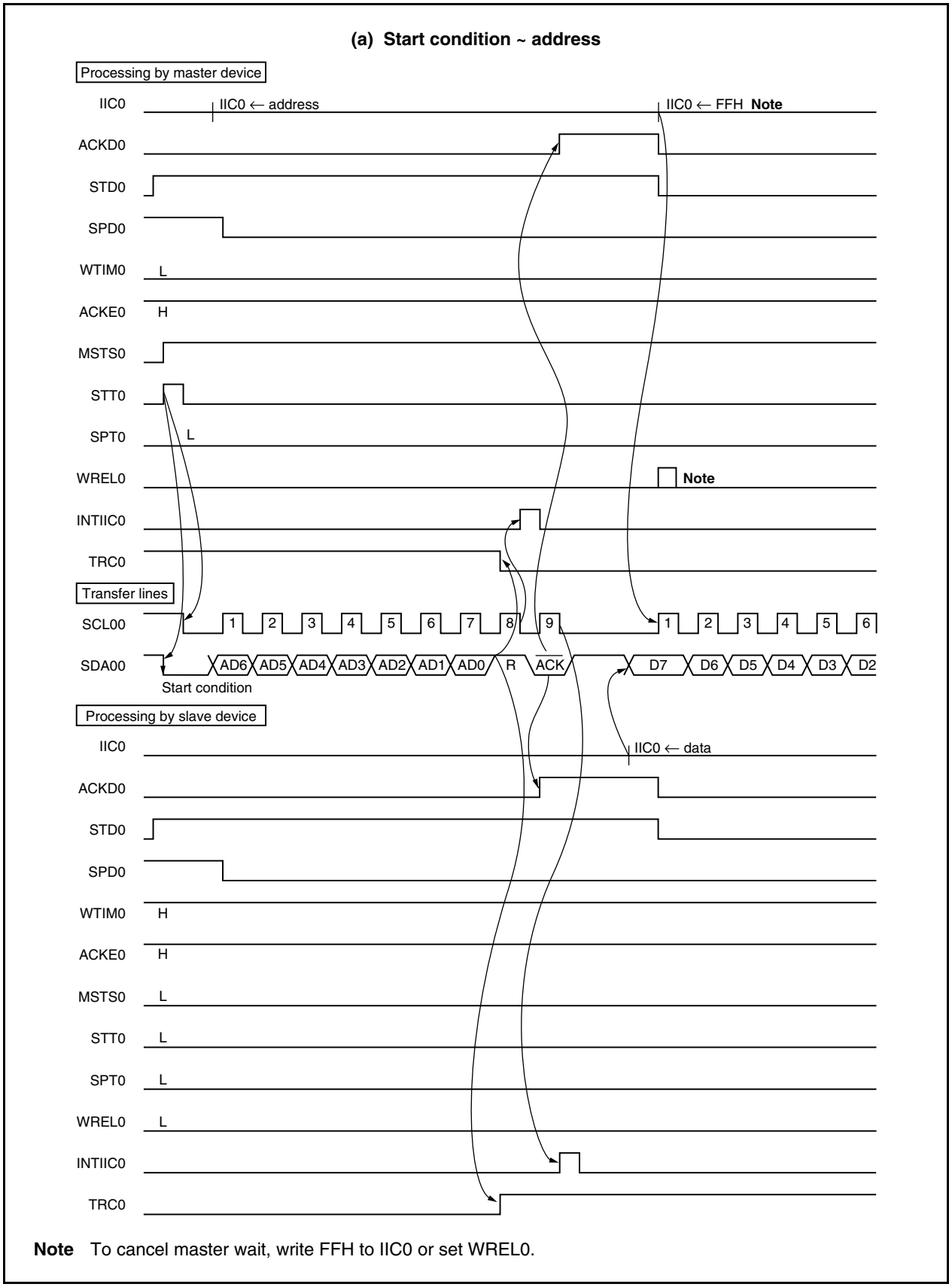
Figure 16-21. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Both Master and Slave) (3/3)

(c) Stop condition



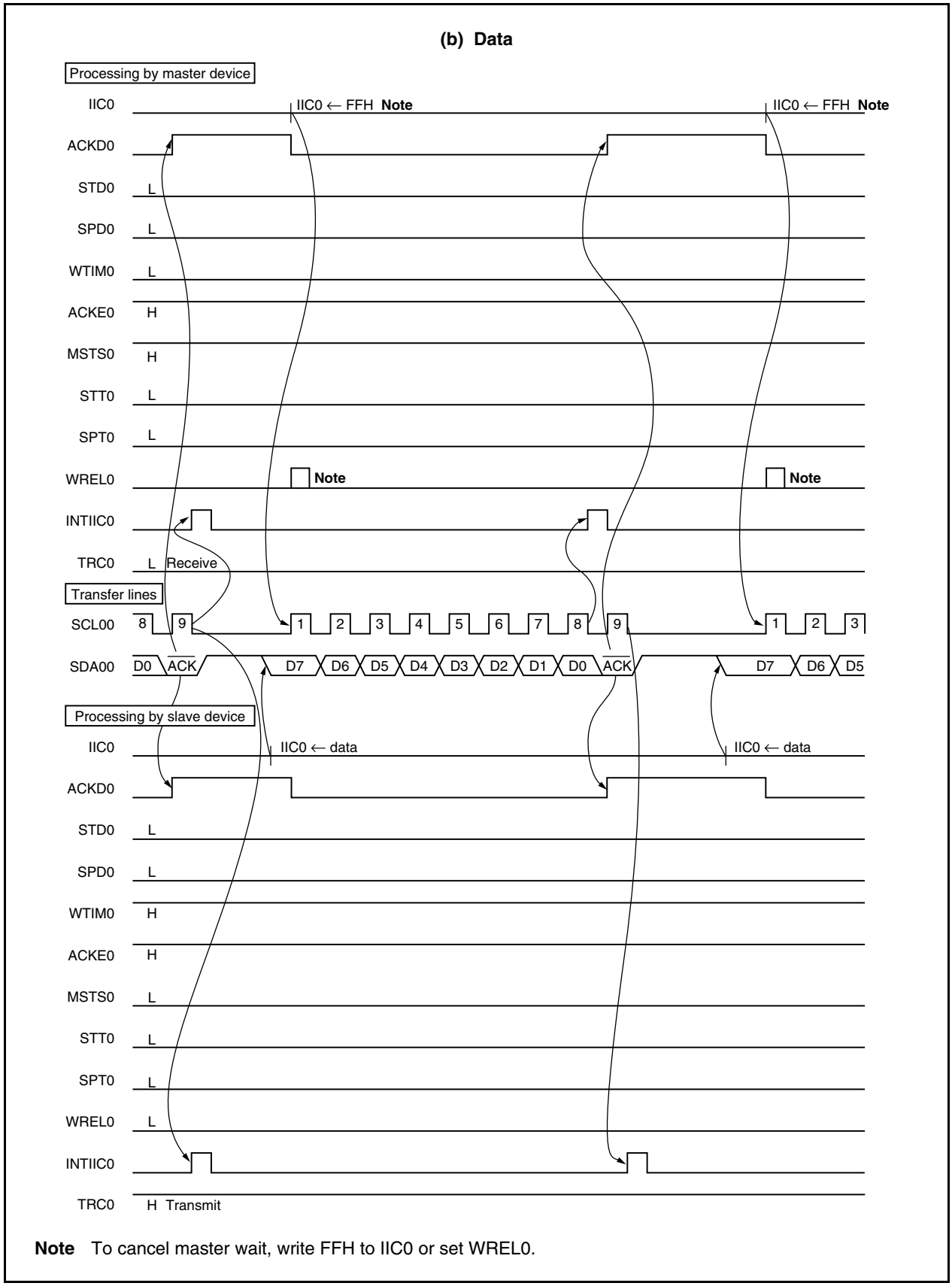
Note To cancel slave wait, write FFH to IIC0 or set WREL0.

Figure 16-22. Example of Slave to Master Communication
(When 8-Clock Wait for Master and 9-Clock Wait for Slave Are Selected) (1/3)



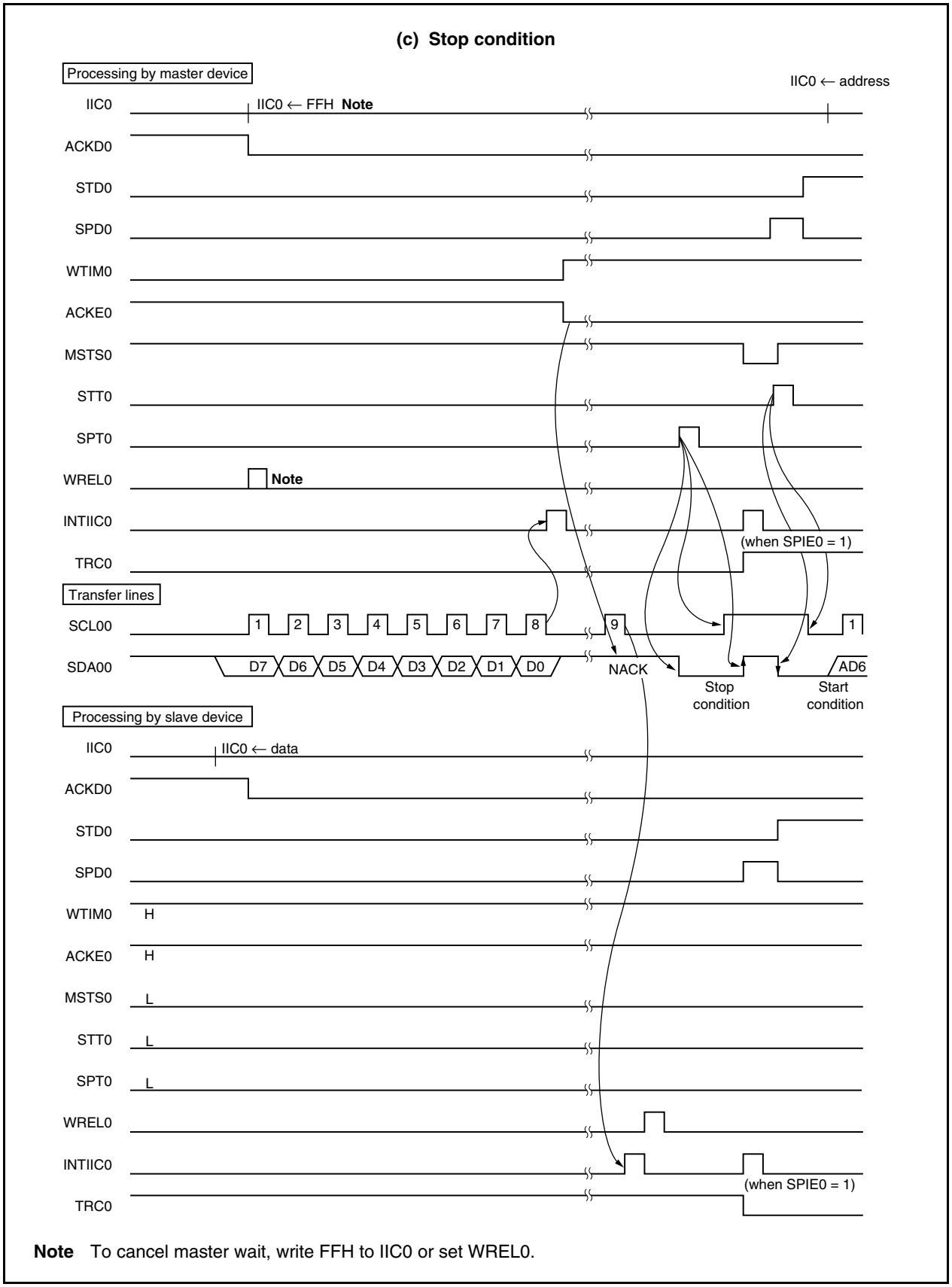
Note To cancel master wait, write FFH to IIC0 or set WREL0.

Figure 16-22. Example of Slave to Master Communication
(When 8-Clock Wait for Master and 9-Clock Wait for Slave Are Selected) (2/3)



Note To cancel master wait, write FFH to IIC0 or set WREL0.

Figure 16-22. Example of Slave to Master Communication
(When 8-Clock Wait for Master and 9-Clock Wait for Slave Are Selected) (3/3)



Note To cancel master wait, write FFH to IIC0 or set WREL0.

CHAPTER 17 DMA FUNCTION (DMA CONTROLLER)

The V850ES/Hx3 includes a direct memory access (DMA) controller (DMAC) that executes and controls DMA transfer.

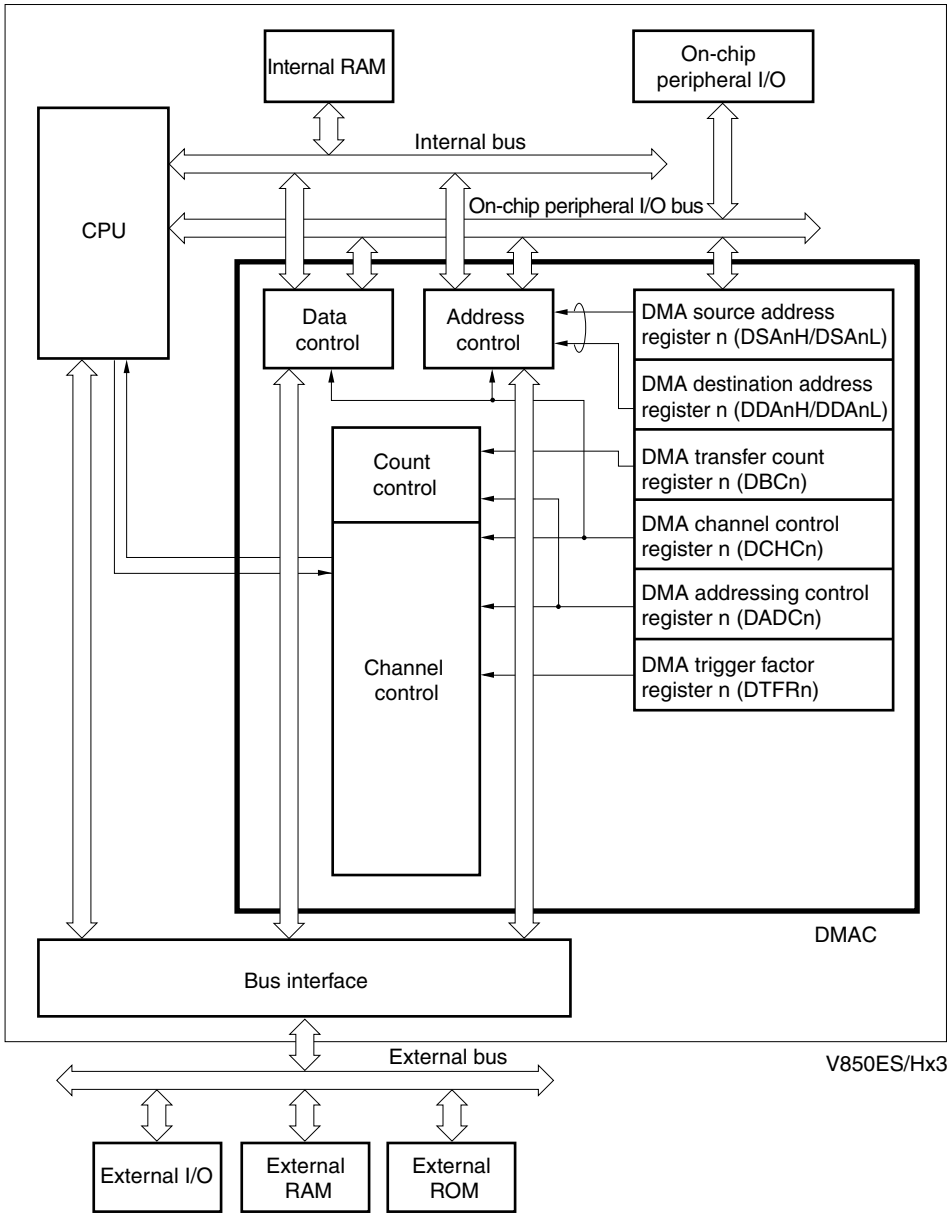
The DMAC controls data transfer between memory and I/O, between memories, or between I/Os based on DMA requests issued by the on-chip peripheral I/O (serial interface, timer/counter, and A/D converter), interrupts from external input pins, or software triggers (memory refers to internal RAM or external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

17.1 Features

- 4 independent DMA channels
- Transfer unit: 8/16 bits
- Maximum transfer count: 65,536 (2^{16})
- Transfer type: Two-cycle transfer
- Transfer mode: Single transfer mode
- Transfer requests
  - Request by interrupts from on-chip peripheral I/O (serial interface, timer/counter, A/D converter) or interrupts from external input pin
  - Requests by software trigger
- Transfer targets
  - Internal RAM ↔ Peripheral I/O
  - Peripheral I/O ↔ Peripheral I/O
  - Internal RAM ↔ External memory[Note]
  - External memory ↔ Peripheral I/O[Note]
  - External memory ↔ External memory[Note]

Note V850ES/HJ3 only

17.2 Configuration



Remark n = 0 to 3

17.3 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	External memory or on-chip peripheral I/O
1	Internal RAM

SA25 to SA16	Set the address (A25 to A16) of the DMA transfer source (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 (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 set bits 14 to 10 of the DSAnH register to "0".

2. Set the DSAnH and DSAnL registers at the following timing when 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 17.13 Cautions).

4. Following reset, set the DSAnH, DSAnL, DDAnH, DDAnL, and DBCn registers before starting DMA transfer. If these registers are not set, the operation when DMA transfer is started is 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, DDA2H FFFFF096H, DDA3H FFFFF09EH, DDA0L FFFFF084H, DDA1L FFFFF08CH, DDA2L FFFFF094H, DDA3L FFFFF09CH

DDAnH (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	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16

DDAnL (n = 0 to 3)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

IR	Specification of DMA transfer destination
0	External memory or on-chip peripheral I/O
1	Internal RAM

DA25 to DA16	Set an address (A25 to A16) of DMA transfer destination (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 an address (A15 to A0) of DMA transfer destination (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 1. Be sure to set bits 14 to 10 of the DDAnH register to "0".

2. Set the DDAnH and DDAnL registers at the following timing when 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 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 17.13 Cautions).

4. Following reset, set the DSAnH, DSAnL, DDAnH, DDAnL, and DBCn registers before starting DMA transfer. If these registers are not set, the operation when DMA transfer is started is not guaranteed.

(3) DMA byte 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 one transfer regardless of the transfer data unit (8/16 bits), and the transfer is terminated if a borrow occurs.

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

After reset: Undefined R/W Address: DBC0 FFFFF0C0H, DBC1 FFFFF0C2H, DBC2 FFFFF0C4H, DBC3 FFFFF0C6H

DBCn (n = 0 to 3)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0

BC15 to BC0	Byte transfer count setting or remaining byte transfer count during DMA transfer
0000H	Byte transfer count 1 or remaining byte transfer count
0001H	Byte transfer count 2 or remaining byte transfer count
:	:
FFFFH	Byte transfer count 65,536 (2^{16}) or remaining byte transfer count
The number of transfer data set first is held when DMA transfer is complete.	

Cautions 1. Set the DBCn register at the following timing when 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 reset, set the DSAnH, DSAnL, DDAnH, DDAnL, and DBCn registers before starting DMA transfer. If these registers are not set, the operation when DMA transfer is started is 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 1. Be sure to set bits 15, 13 to 8, and 3 to 0 of the DADCn register to "0".

2. Set the DADCn register at the following timing when 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. The DS0 bit specifies the size of the transfer data, and does not control bus sizing. If 8-bit data (DS0 bit = 0) is set, therefore, the lower data bus is not always used.

4. 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.

5. 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 the DMA transfer operating mode 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 read value 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 indicates whether DMA transfer through DMA channel n has completed or not
0	DMA transfer had not completed.
1	DMA transfer had completed.
It is set to 1 on 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. When re-setting the DMA transfer status (re-setting the DDAnH, DDAnL, DSAnH, DSAnL, DBCn, and DADCn registers) before DMA transfer is completed (before the TCn bit is set to 1), be sure to initialize the DMA channel. When initializing the DMA controller, however, be sure to observe the procedure described in **17.13 Cautions**.

STGn Note 2	This is a software startup trigger of DMA transfer. If this bit is set to 1 in the DMA transfer enable state (TCn bit = 0, Enn bit = 1), DMA transfer is started.

Enn	Setting of whether DMA transfer through 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, set the Enn bit to 1 again. When aborting or resuming DMA transfer, however, be sure to observe the procedure described in **17.13 Cautions**.	

Notes 1. The TCn bit is read-only.

2. The INITn and STGn bits are write-only.

Cautions 1. Be sure to set 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, DFn 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

	7	6	5	4	3	2	1	0
DTFRn	DFn	0	IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0

(n = 0 to 3)

DFn[Note]	DMA transfer request status flag
0	No DMA transfer request
1	DMA transfer request

Note Do not set the DFn bit to 1 by software. Write 0 to this bit to clear a DMA transfer request if an interrupt that is specified as the cause of starting DMA transfer occurs while DMA transfer is disabled.

Cautions 1. Set the IFCn5 to IFCn0 bits at the following timing when 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. An interrupt request that is generated in the standby mode (IDEL1, IDLE2, STOP, or sub-IDLE mode) does not start the DMA transfer cycle (nor is the DFn bit set to 1).

3. If a DMA start factor is selected by the IFCn5 to IFCn0 bits, the DFn bit is set to 1 when an interrupt occurs from the selected on-chip peripheral I/O, regardless of whether the DMA transfer is enabled or disabled. If DMA is enabled in this status, DMA transfer is immediately started.

Remark For the IFCn5 to IFCn0 bits, see **Table 17-1 DMA Start Factors**.

Table 17-1. DMA Start Factors (1/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt Source	V850ES/HE3	V850ES/HF3	V850ES/HG3	V850ES/HJ3
0	0	0	0	0	0	DMA request by interrupt disabled	√	√	√	√
0	0	0	0	0	1	INTLVIL	√	√	√	√
0	0	0	0	1	0	INTP0	√	√	√	√
0	0	0	0	1	1	INTP1	√	√	√	√
0	0	0	1	0	0	INTP2	√	√	√	√
0	0	0	1	0	1	INTP3	√	√	√	√
0	0	0	1	1	0	INTP4	√	√	√	√
0	0	0	1	1	1	INTP5	√	√	√	√
0	0	1	0	0	0	INTP6	√	√	√	√
0	0	1	0	0	1	INTP7	√	√	√	√
0	0	1	0	1	0	INTTAB0OV	√	√	√	√
0	0	1	0	1	1	INTTAB0CC0	√	√	√	√
0	0	1	1	0	0	INTTAB0CC1	√	√	√	√
0	0	1	1	0	1	INTTAB0CC2	√	√	√	√
0	0	1	1	1	0	INTTAB0CC3	√	√	√	√
0	0	1	1	1	1	INTTAA0OV	√	√	√	√
0	1	0	0	0	0	INTTAA0CC0	√	√	√	√
0	1	0	0	0	1	INTTAA0CC1	√	√	√	√
0	1	0	0	1	0	INTTAA1OV	√	√	√	√
0	1	0	0	1	1	INTTAA1CC0	√	√	√	√
0	1	0	1	0	0	INTTAA1CC1	√	√	√	√
0	1	0	1	0	1	INTTAA2OV	√	√	√	√
0	1	0	1	1	0	INTTAA2CC0	√	√	√	√
0	1	0	1	1	1	INTTAA2CC1	√	√	√	√
0	1	1	0	0	0	INTTAA3OV	√	√	√	√
0	1	1	0	0	1	INTTAA3CC0	√	√	√	√
0	1	1	0	1	0	INTTAA3CC1	√	√	√	√
0	1	1	0	1	1	INTTM0EQ0	√	√	√	√
0	1	1	1	0	0	INTCB0R	√	√	√	√
0	1	1	1	0	1	INTCB0T	√	√	√	√
0	1	1	1	1	0	INTCB1R	√	√	√	√
0	1	1	1	1	1	INTCB1T	√	√	√	√
1	0	0	0	0	0	INTUD0R	√	√	√	√
1	0	0	0	0	1	INTUD0T	√	√	√	√
1	0	0	0	1	0	INTUD1R	√	√	√	√
1	0	0	0	1	1	INTUD1T	√	√	√	√
1	0	0	1	0	0	INTAD	√	√	√	√
1	0	0	1	0	1	INTTAA4OV	√	√	√	√
1	0	0	1	1	0	INTTAA4CC0	√	√	√	√
1	0	0	1	1	1	INTTAA4CC1	√	√	√	√
1	0	1	0	0	0	INTIIC0	√	√	√	√

Remark n = 0 to 3

Table 17-1. DMA Start Factors (2/2)

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt Source	V850ES/HE3	V850ES/HF3	V850ES/HG3	V850ES/HJ3
1	0	1	0	0	1	INTKR	√	√	√	√
1	0	1	0	1	0	INTTAB1OV	–	–	√	√
1	0	1	0	1	1	INTTAB1CC0	–	–	√	√
1	0	1	1	0	0	INTTAB1CC1	–	–	√	√
1	0	1	1	0	1	INTTAB1CC2	–	–	√	√
1	0	1	1	1	0	INTTAB1CC3	–	–	√	√
1	0	1	1	1	1	INTUD2R	–	–	√	√
1	1	0	0	0	0	INTUD2T	–	–	√	√
1	1	0	0	0	1	INTLVIH	√	√	√	√
1	1	0	0	1	0	INTUD3R	–	–	–	√Note
1	1	0	0	1	1	INTUD3T	–	–	–	√Note
1	1	0	1	0	0	INTTAB2OV	–	–	–	√
1	1	0	1	0	1	INTTAB2CC0	–	–	–	√
1	1	0	1	1	0	INTTAB2CC1	–	–	–	√
1	1	0	1	1	1	INTTAB2CC2	–	–	–	√
1	1	1	0	0	0	INTTAB2CC3	–	–	–	√
1	1	1	0	0	1	INTCB2R	–	–	–	√
1	1	1	0	1	0	INTCB2T	–	–	–	√
1	1	1	0	1	1	INTUD4R	–	–	–	√Note
1	1	1	1	0	0	INTUD4T	–	–	–	√Note
1	1	1	1	0	1	INTUD5R	–	–	–	√Note
1	1	1	1	1	0	INTUD5T	–	–	–	√Note
Other than above						Setting prohibited				

Note μPD70F3757 only

Remark n = 0 to 3

17.4 Transfer Targets

Table 17-2 shows the relationship between the transfer targets (√: Transfer enabled, ×: Transfer disabled).

Table 17-2. Relationship Between Transfer Targets

		Transfer Destination			
		Internal ROM	On-Chip Peripheral I/O	Internal RAM	External Memory
Source	On-chip peripheral I/O	×	√	√	√
	Internal RAM	×	√	×	√
	External memory	×	√	√	√
	Internal ROM	×	×	×	×

Caution The operation is not guaranteed for combinations of transfer destination and source marked with "×" in Table 17-2.

17.5 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.

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

If a new transfer request of the same channel and a transfer request of another channel with a lower priority are generated in a transfer cycle, DMA transfer of the channel with the lower priority is executed after the bus is released to the CPU (the new transfer request of the same channel is ignored in the transfer cycle).

17.6 Transfer Types

As a transfer type, the 2-cycle transfer is supported.

In two-cycle transfer, data transfer is performed in two cycles, a read cycle and a write cycle.

In the read cycle, the transfer source address is output and reading is performed from the source to the DMAC. In the write cycle, the transfer destination address is output and writing is performed 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 for DMA transfer of two cycles, the operation is performed as follows.

<16-bit data transfer>

<1> Transfer from 32-bit bus → 16-bit bus
A read cycle (the higher 16 bits are in a high-impedance state) is generated, followed by generation of 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 between 16-bit bus and 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 to an on-chip peripheral I/O register (transfer source/destination), be sure to specify the same transfer size as the register size. For example, for DMA transfer to an 8-bit register, be sure to specify byte (8-bit) transfer.

Remark The bus width of each transfer target (transfer source/destination) is as follows.

- On-chip peripheral I/O: 16-bit bus width
- Internal RAM: 32-bit bus width
- External memory: 8-bit or 16-bit bus width

17.7 DMA Channel Priorities

The DMA channel priorities are fixed as follows.

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

The priorities are checked for every transfer cycle.

17.8 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>)

DMA Cycle		Minimum Number of Execution Clocks
<1> DMA request response time		4 clocks (MIN.) + Noise elimination time[Note 2]
<2> Memory access	External memory access	Depends on connected memory.
	Internal RAM access	2 clocks[Note 3]
	Peripheral I/O register access	3 clocks + Number of wait cycles specified by VSWC register[Note 4]

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 14).

3. Two clocks are required for a DMA cycle.

4. More wait cycles are necessary for accessing a specific peripheral I/O register (for details, see **3.4.8 (2)**).

17.9 DMA Transfer Start Factors

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

(1) Request by software

If the STGn bit is set to 1 while the DCHCn.TCn bit = 1 and Enn bit = 1 (DMA transfer enabled), DMA transfer is started.

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 INTDMAn 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 DCHCn.TCn bit = 0 and Enn bit = 1 (DMA transfer enabled), DMA transfer is started.

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. The start factor that is valid cannot be identified.

2. A new transfer request that is generated after the preceding DMA transfer request was generated or in the preceding DMA transfer cycle is ignored (cleared).

3. The transfer request interval of the same DMA channel varies depending on the setting of bus wait in the DMA transfer cycle, the start status of the other channels, or the external bus hold request. In particular, as described in Caution 2, a new transfer request that is generated for the same channel before the DMA transfer cycle or during the DMA transfer cycle is ignored. Therefore, the transfer request intervals for the same DMA channel must be sufficiently separated by the system. When the software trigger is used, completion of the DMA transfer cycle that was generated before can be checked by updating the DBCn register.

17.10 DMA Abort Factors

DMA transfer is aborted if a bus hold occurs.

The same applies if transfer is executed between the internal memory/on-chip peripheral I/O and internal memory/on-chip peripheral I/O.

When the bus hold is cleared, DMA transfer is resumed.

17.11 End of DMA Transfer

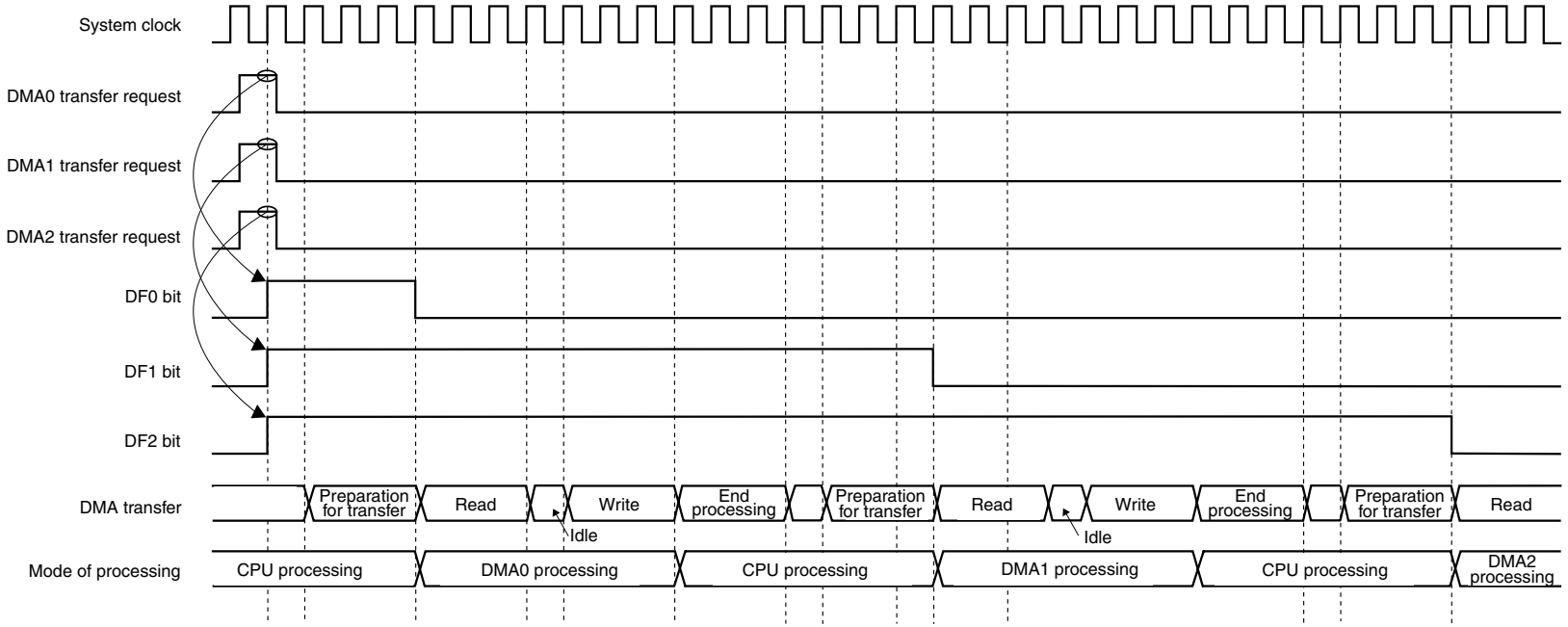
When DMA transfer has been completed the number of times set to the DBCn register and when the DCHCn.Enn bit is cleared to 0 and TCn bit is set to 1, a DMA transfer end interrupt request signal (INTDMAn) is generated for the interrupt controller (INTC) (n = 0 to 3).

The V850ES/Hx3 does 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.

17.12 Operation Timing

Figures 17-1 to 17-4 show DMA operation timing.

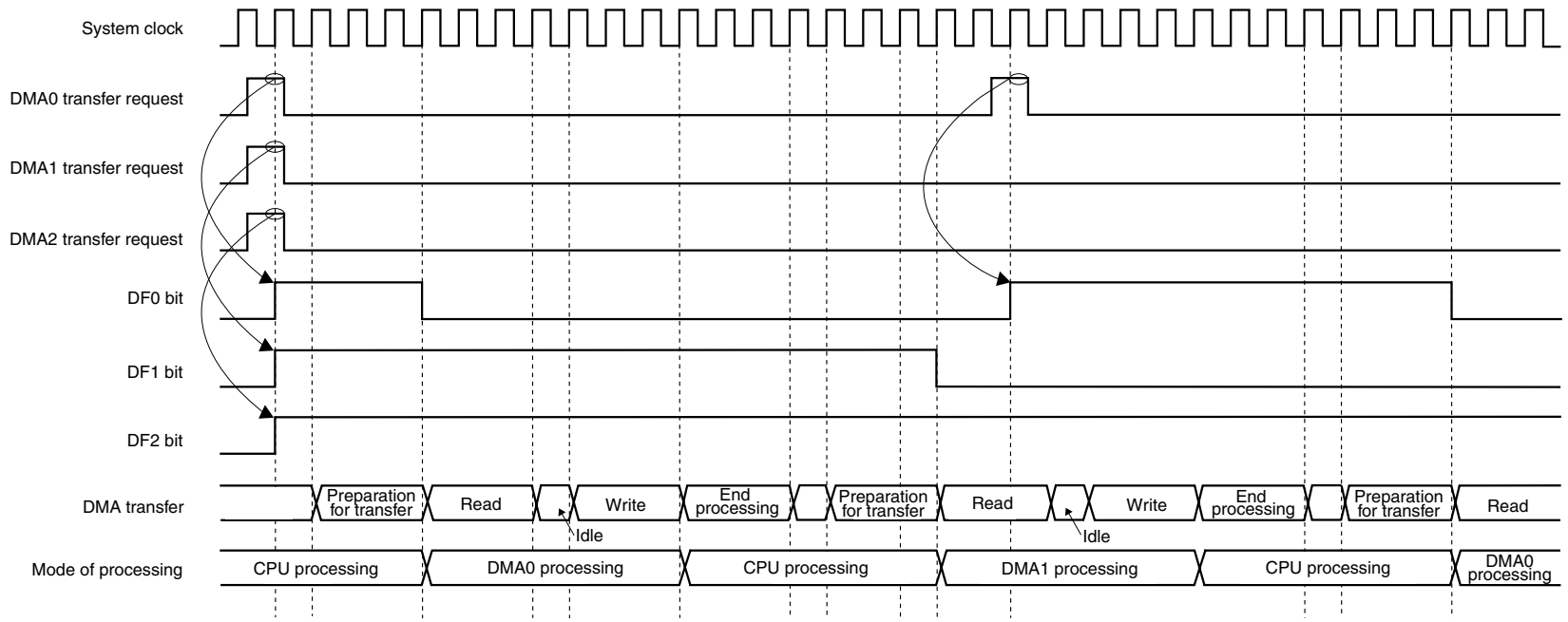
Figure 17-1. Priority of DMA (1)



Remarks 1. Transfer in the order of DMA0 → DMA1 → DMA2

2. In the case of transfer between external memory spaces (multiplexed bus, no wait)

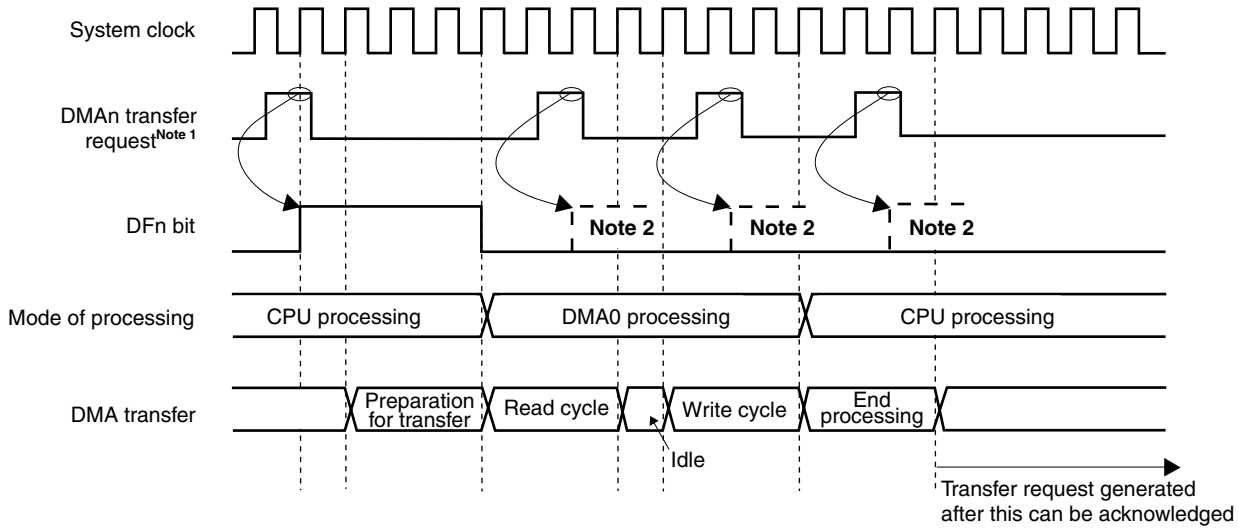
Figure 17-2. Priority of DMA (2)



Remarks 1. Transfer in the order of DMA0 → DMA1 → DMA0 (DMA2 is held pending.)

2. In the case of transfer between external memory spaces (multiplexed bus, no wait)

Figure 17-3. Period in Which DMA Transfer Request Is Ignored (1)

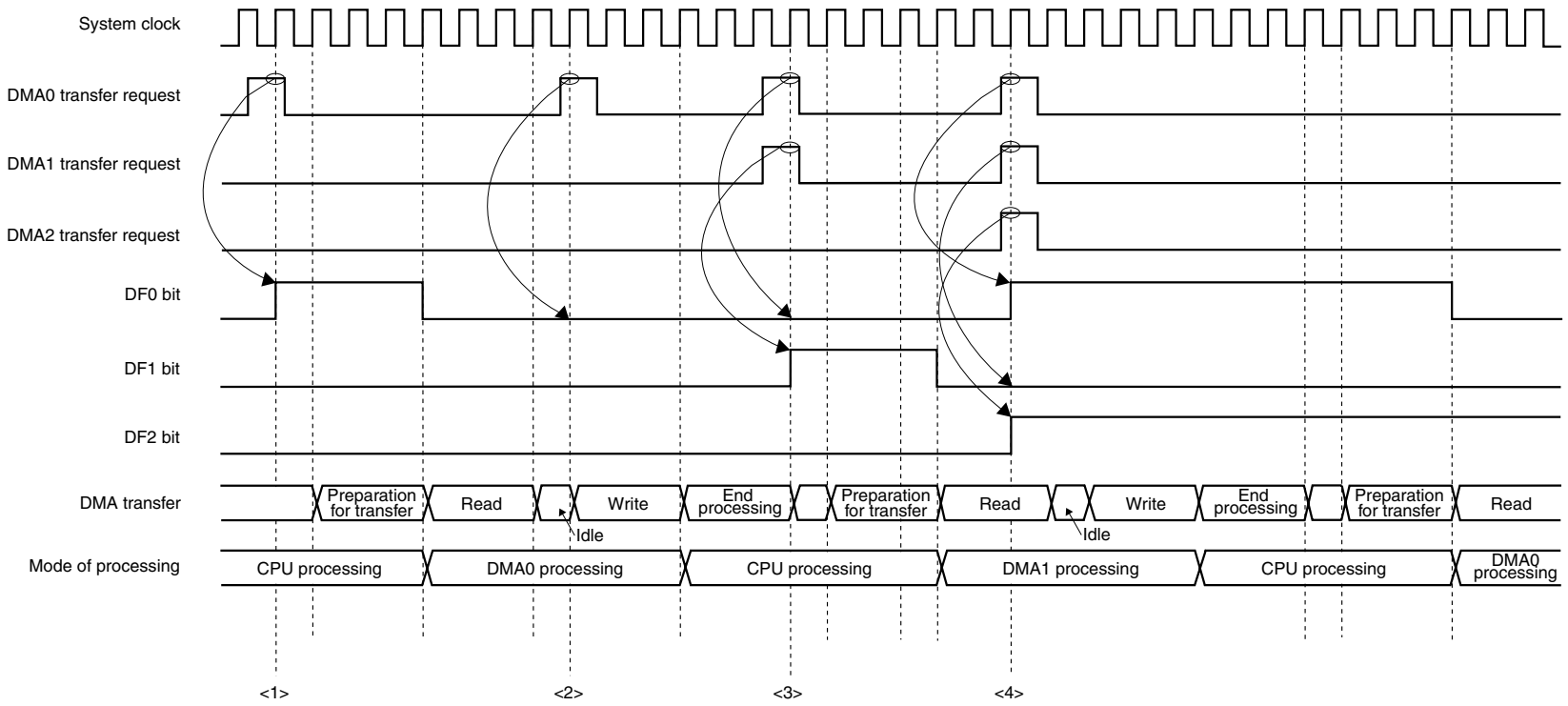


Notes 1. Interrupt from on-chip peripheral I/O, or software trigger (STGn bit)

2. New DMA request of the same channel is ignored between when the first request is generated and the end processing is complete.

Remark In the case of transfer between external memory spaces (multiplexed bus, no wait)

Figure 17-4. Period in Which DMA Transfer Request Is Ignored (2)



<1> DMA0 transfer request

<2> New DMA0 transfer request is generated during DMA0 transfer.

→ A DMA transfer request of the same channel is ignored during DMA transfer.

<3> Requests for DMA0 and DMA1 are generated at the same time.

→ DMA0 request is ignored (a DMA transfer request of the same channel during transfer is ignored).

→ DMA1 request is acknowledged.

<4> Requests for DMA0, DMA1, and DMA2 are generated at the same time.

→ DMA1 request is ignored (a DMA transfer request of the same channel during transfer is ignored).

→ DMA0 request is acknowledged according to priority. DMA2 request is held pending (transfer of DMA2 occurs next).

17.13 Cautions

(1) Caution for VSWC register

When using the DMAC, be sure to set an appropriate value, in accordance with the operating frequency, to the VSWC register.

When the default value (77H) of the VSWC register is used, or if an inappropriate value is set to the VSWC register, the operation is not correctly performed (for details of the VSWC register, see **3.4.8 (1) (a) System wait control register (VSWC)**).

(2) Caution for DMA transfer executed on internal RAM

When executing the following instructions located in the internal RAM, do not execute a DMA transfer that transfers data to/from the internal RAM (transfer source/destination), because the CPU may not operate correctly afterward.

- Data access instruction to misaligned address located in internal RAM

Conversely, when executing a DMA transfer to transfer data to/from the internal RAM (transfer source/destination), do not execute the above instruction.

(3) Caution for reading DCHCn.TCn bit (n = 0 to 3)

The TCn bit is cleared to 0 when it is read, but it is not automatically cleared even if it is read at a specific timing. To accurately 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 servicing routine

Execute reading 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 accurately initialize the channel, execute either of the following two procedures.

<R> **(a) Temporarily stop transfer of 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 of DMA channels other than the one to be forcibly terminated, and transfer the value to a general-purpose register.

<3> Clear the Enn bit of the DMA channels used (including the channel to be forcibly terminated) to 0. To clear the Enn bit of the last DMA channel, execute the clear instruction twice. If the target of DMA transfer (transfer source/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 target of transfer is not the internal RAM).

- Write DCHC0 = 00H (clear E00 bit).
- Write DCHC1 = 00H (clear E11 bit)
- Write DCHC2 = 00H (clear E22 bit).
- Write DCHC2 = 00H again (clear E22 bit).

<4> Write DCHCn = 04 to the channel to be forcibly terminated (set 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 whose DMA transfer has been normally completed between <2> and <3>.

2. When a bit manipulation instruction is used, steps <3> and <4> (Enn bit clear (0) and INITn bit set (1)) are prohibited because the TCn bit is cleared to 0.

(b) Repeatedly execute setting INITn bit until transfer is forcibly terminated correctly

<1> Suppress a request from the DMA request source of the channel to be forcibly terminated (stop operation of the on-chip peripheral I/O).
<2> Check that the DMA transfer request of 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 request is completed.
<3> When it has been confirmed that the DMA request of the channel to be forcibly terminated is not held pending, clear the Enn bit to 0.
<4> Again, clear the Enn bit of the channel to be forcibly terminated.
If the target of transfer for the channel to be forcibly terminated (transfer source/destination) is the internal RAM, execute this operation once more.
<5> Copy the initial number of transfers of the channel to be forcibly terminated to a general-purpose register.
<6> Set the INITn bit of the channel to be forcibly terminated to 1.
<7> Read the value of the DBCn register of the channel to be forcibly terminated, and compare it with the value copied in <5>. 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 of 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 the 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 pending, wait until execution of the pending DMA transfer request is completed.
<3> If it has been confirmed that no DMA transfer request is held pending, clear the Enn bit to 0 (this operation stops DMA transfer).
<4> Set the Enn bit to 1 to resume DMA transfer.
<5> Resume the operation of the DMA request source that has been stopped (start the 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 target (external memory, internal RAM, or on-chip peripheral I/O) during DMA transfer.

(7) Transferring misaligned data

DMA transfer of misaligned data with a 16-bit bus width is not supported.
If an odd address is specified as the transfer source or destination, the least significant bit of the address is forcibly assumed to be 0.

(8) Bus arbitration for CPU

Because the DMA controller has a higher priority bus mastership 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 internal RAM to/from which DMA transfer is not being executed.

[Example]

- The CPU can access the internal ROM and internal RAM when DMA transfer is being executed between the external memory and on-chip peripheral I/O.
- The CPU can access the internal ROM when DMA transfer is being executed between the on-chip peripheral I/O and internal RAM.

(9) Registers/bits that must not be rewritten during DMA operation

Set the following registers at the following timing when a DMA operation is not under execution.

[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
- Time 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.

- 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 two or more DMA channels with the same start factor. If two or more channels are started with the same factor, DMA for which a channel has already been set may be started or a DMA channel with a lower priority may be acknowledged earlier than a DMA channel with a higher priority. The operation cannot be guaranteed.

(12) Read values of DSAn and DDAn registers

Values in the middle of updating may be read from the DSAn and DDAn registers during DMA transfer (n = 0 to 3).

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 read

<1> Read value of DSAnH register: DSAnH = 0000H

<2> Read value of DSAnL register: DSAnL = FFFFH

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

<1> Read value of DSAnH register: DSAnH = 0000H

<2> Occurrence of DMA transfer

<3> Incrementing DSAn register: DSAn = 00100000H

<4> Read value of DSAnL register: DSAnL = 0000H

CHAPTER 18 INTERRUPT/EXCEPTION PROCESSING FUNCTION

The V850ES/Hx3 is provided with a dedicated interrupt controller (INTC) for interrupt servicing and can process a total of 52 to 80 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/Hx3 can process interrupt request signals from the on-chip peripheral hardware and external sources. Moreover, exception processing can be started by the TRAP instruction (software exception) or by generation of an exception event (i.e. fetching of an illegal opcode) (exception trap).

18.1 Features

○ Interrupts
- Non-maskable interrupts: 2 sources (external: 1 source, internal: 1 source)
- Maskable interrupts: See **Table 18-1**.
- 8 levels of programmable priorities (maskable interrupts)
- Multiple interrupt control according to priority
- Masks can be specified for each maskable interrupt request.
- Noise elimination, edge detection, and valid edge specification for external interrupt request signals.

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

Maskable interrupt sources of the V850ES/Hx3 are listed in Table 18-1.

Table 18-1. Maskable Interrupts of V850ES/Hx3

		Internal Source	External Source	Total
V850ES/HE3		42 sources	8 sources	50 sources
V850ES/HF3		42 sources	8 sources	50 sources
V850ES/HG3		50 sources	11 sources	61 sources
V850ES/HJ3	μPD70F3755	57 sources	15 sources	72 sources
	μPD70F3757	63 sources	15 sources	78 sources

Interrupt/exception sources of the V850ES/Hx3 are listed in Tables 18-2 to 18-4.

Table 18-2. Interrupt Source List of V850ES/HE3 and V850ES/HF3 (1/3)

Type	Classification	DP[Note 1]	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Reset	Interrupt	–	RESET	$\overline{\text{RESET}}$ pin input Reset input by internal source	RESET	0000H	00000000H	Undefined	–
Non-maskable	Interrupt	–	NMI	NMI pin valid edge input	Pin	0010H	00000010H	nextPC	–
		–	INTWDT2	WDT2 overflow	WDT2	0020H	00000020H	**Note 2**	–
Software exception	Exception	–	TRAP0n[Note 3]	TRAP instruction	–	004nH[Note 3]	00000040H	nextPC	–
		–	TRAP1n[Note 3]	TRAP instruction	–	005nH[Note 3]	00000050H	nextPC	–
Exception trap	Exception	–	ILGOP/ DBG0	Illegal opcode/ DBTRAP instruction	–	0060H	00000060H	nextPC	–
Maskable	Interrupt	0	INTLVIL	Low voltage detection (voltage under reference level)	POCLVI	0080H	00000080H	nextPC	LVILIC
		1	INTLVIH	Low voltage detection (voltage over reference level)	POCLVI	0090H	00000090H	nextPC	LVIHIC
		2	INTP0	External interrupt pin input edge detection (INTP0)	Pin	00A0H	000000A0H	nextPC	PIC0
		3	INTP1	External interrupt pin input edge detection (INTP1)	Pin	00B0H	000000B0H	nextPC	PIC1
		4	INTP2	External interrupt pin input edge detection (INTP2)	Pin	00C0H	000000C0H	nextPC	PIC2
		5	INTP3	External interrupt pin input edge detection (INTP3)	Pin	00D0H	000000D0H	nextPC	PIC3
		6	INTP4	External interrupt pin input edge detection (INTP4)	Pin	00E0H	000000E0H	nextPC	PIC4
		7	INTP5	External interrupt pin input edge detection (INTP5)	Pin	00F0H	000000F0H	nextPC	PIC5
		8	INTP6	External interrupt pin input edge detection (INTP6)	Pin	0100H	00000100H	nextPC	PIC6
		9	INTP7	External interrupt pin input edge detection (INTP7)	Pin	0110H	00000110H	nextPC	PIC7
		10	INTTAB0OV	TAB0 overflow	TAB0	0120H	00000120H	nextPC	TAB0OVIC
		11	INTTAB0CC0	TAB0 capture 0/compare 0 match	TAB0	0130H	00000130H	nextPC	TAB0CCIC0
		12	INTTAB0CC1	TAB0 capture 1/compare 1 match	TAB0	0140H	00000140H	nextPC	TAB0CCIC1
		13	INTTAB0CC2	TAB0 capture 2/compare 2 match	TAB0	0150H	00000150H	nextPC	TAB0CCIC2
		14	INTTAB0CC3	TAB0 capture 3/compare 3 match	TAB0	0160H	00000160H	nextPC	TAB0CCIC3

Notes 1. DP: Default priority

2. For the restoring in the case of INTWDT2, see **18.2.2 (2) From INTWDT2 signal**.

3. n = 0H to FH

Table 18-2. Interrupt Source List of V850ES/HE3 and V850ES/HF3 (2/3)

Type	Classification	DP[Note]	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	15	INTTAA0OV	TAA0 overflow	TAA0	0170H	00000170H	nextPC	TAA0OVIC
		16	INTTAA0CC0	TAA0 capture 0/compare 0 match	TAA0	0180H	00000180H	nextPC	TAA0CCIC0
		17	INTTAA0CC1	TAA0 capture 1/compare 1 match	TAA0	0190H	00000190H	nextPC	TAA0CCIC1
		18	INTTAA1OV	TAA1 overflow	TAA1	01A0H	000001A0H	nextPC	TAA1OVIC
		19	INTTAA1CC0	TAA1 capture 0/compare 0 match	TAA1	01B0H	000001B0H	nextPC	TAA1CCIC0
		20	INTTAA1CC1	TAA1 capture 1/compare 1 match	TAA1	01C0H	000001C0H	nextPC	TAA1CCIC1
		21	INTTAA2OV	TAA2 overflow	TAA2	01D0H	000001D0H	nextPC	TAA2OVIC
		22	INTTAA2CC0	TAA2 capture 0/compare 0 match	TAA2	01E0H	000001E0H	nextPC	TAA2CCIC0
		23	INTTAA2CC1	TAA2 capture 1/compare 1 match	TAA2	01F0H	000001F0H	nextPC	TAA2CCIC1
		24	INTTAA3OV	TAA3 overflow	TAA3	0200H	00000200H	nextPC	TAA3OVIC
		25	INTTAA3CC0	TAA3 capture 0/compare 0 match	TAA3	0210H	00000210H	nextPC	TAA3CCIC0
		26	INTTAA3CC1	TAA3 capture 1/compare 1 match	TAA3	0220H	00000220H	nextPC	TAA3CCIC1
		27	INTTAA4OV	TAA4 overflow	TAA4	0230H	00000230H	nextPC	TAA4OVIC
		28	INTTAA4CC0	TAA4 capture 0/compare 0 match	TAA4	0240H	00000240H	nextPC	TAA4CCIC0
		29	INTTAA4CC1	TAA4 capture 1/compare 1 match	TAA4	0250H	00000250H	nextPC	TAA4CCIC1
		30	INTTM0EQ0	TMM0 compare match	TMM0	0260H	00000260H	nextPC	TM0EQIC0
		31	INTCB0R	CSIB0 reception completion/error	CSIB0	0270H	00000270H	nextPC	CB0RIC
		32	INTCB0T	CSIB0 consecutive transmission write enable	CSIB0	0280H	00000280H	nextPC	CB0TIC
		33	INTCB1R	CSIB1 reception completion/error	CSIB1	0290H	00000290H	nextPC	CB1RIC
		34	INTCB1T	CSIB1 consecutive transmission write enable	CSIB1	02A0H	000002A0H	nextPC	CB1TIC
		35	INTUD0S	UARTD0 status	UARTD0	02B0H	000002B0H	nextPC	UD0SIC
		36	INTUD0R	UARTD0 reception completion	UARTD0	02C0H	000002C0H	nextPC	UD0RIC
		37	INTUD0T	UARTD0 transmission enable	UARTD0	02D0H	000002D0H	nextPC	UD0TIC
		38	INTUD1S	UARTD1 status	UARTD1	02E0H	000002E0H	nextPC	UD1SIC
		39	INTUD1R	UARTD1 reception completion	UARTD1	02F0H	000002F0H	nextPC	UD1RIC
		40	INTUD1T	UARTD1 transmission enable	UARTD1	0300H	00000300H	nextPC	UD1TIC
		41	INTIIC0	IIC0 transfer end	IIC0	0310H	00000310H	nextPC	IIC0IC
		42	INTAD	A/D conversion end	A/D	0320H	00000320H	nextPC	ADIC
		43	INTDMA0	DMA0 transfer end	DMA	0370H	00000370H	nextPC	DMAIC0

Note DP: Default priority

Table 18-2. Interrupt Source List of V850ES/HE3 and V850ES/HF3 (3/3)

Type	Classification	DP[Note]	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	44	INTDMA1	DMA1 transfer end	DMA	0380H	00000380H	nextPC	DMAIC1
		45	INTDMA2	DMA2 transfer end	DMA	0390H	00000390H	nextPC	DMAIC2
		46	INTDMA3	DMA3 transfer end	DMA	03A0H	000003A0H	nextPC	DMAIC3
		47	INTKR	Key return interrupt request	KR	03B0H	000003B0H	nextPC	KRIC
		48	INTWTI	Watch timer interval	WT	03C0H	000003C0H	nextPC	WTIIC
		49	INTWT	Watch timer reference time	WT	03D0H	000003D0H	nextPC	WTIC

Note DP: Default priority

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

The priority order of non-maskable interrupt is INTWDT2 > NMI.

Restored PC: The value of the program counter (PC) saved to EIPC, FEPC, or DBPC when interrupt servicing is started. Note, however, that the restored PC when a non-maskable or maskable interrupt is acknowledged while one of the following instructions is being executed does not become the nextPC (if an interrupt is acknowledged during interrupt execution, execution stops, and then resumes after the interrupt servicing has finished).

- Load instructions (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)
- Division instructions (DIV, DIVH, DIVU, DIVHU)
- PREPARE, DISPOSE instructions (only if an interrupt is generated before the stack pointer is updated)

nextPC: The PC value that starts the processing following interrupt/exception processing.

2. The execution address of the illegal instruction when an illegal opcode exception occurs is calculated by (Restored PC – 4).

Table 18-3. Interrupt Source List of V850ES/HG3 (1/3)

Type	Classification	DP[Note 1]	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Reset	Interrupt	–	RESET	$\overline{\text{RESET}}$ pin input Reset input by internal source	RESET	0000H	00000000H	Undefined	–
Non-maskable	Interrupt	–	NMI	NMI pin valid edge input	Pin	0010H	00000010H	nextPC	–
		–	INTWDT2	WDT2 overflow	WDT2	0020H	00000020H	**Note 2**	–
Software exception	Exception	–	TRAP0n[Note 3]	TRAP instruction	–	004nH[Note 3]	00000040H	nextPC	–
		–	TRAP1n[Note 3]	TRAP instruction	–	005nH[Note 3]	00000050H	nextPC	–
Exception trap	Exception	–	ILGOP/ DBG0	Illegal opcode/ DBTRAP instruction	–	0060H	00000060H	nextPC	–
Maskable	Interrupt	0	INTLVIL	Low voltage detection (voltage under reference level)	POCLVI	0080H	00000080H	nextPC	LVILIC
		1	INTLVIH	Low voltage detection (voltage over reference level)	POCLVI	0090H	00000090H	nextPC	LVIHIC
		2	INTP0	External interrupt pin input edge detection (INTP0)	Pin	00A0H	000000A0H	nextPC	PIC0
		3	INTP1	External interrupt pin input edge detection (INTP1)	Pin	00B0H	000000B0H	nextPC	PIC1
		4	INTP2	External interrupt pin input edge detection (INTP2)	Pin	00C0H	000000C0H	nextPC	PIC2
		5	INTP3	External interrupt pin input edge detection (INTP3)	Pin	00D0H	000000D0H	nextPC	PIC3
		6	INTP4	External interrupt pin input edge detection (INTP4)	Pin	00E0H	000000E0H	nextPC	PIC4
		7	INTP5	External interrupt pin input edge detection (INTP5)	Pin	00F0H	000000F0H	nextPC	PIC5
		8	INTP6	External interrupt pin input edge detection (INTP6)	Pin	0100H	00000100H	nextPC	PIC6
		9	INTP7	External interrupt pin input edge detection (INTP7)	Pin	0110H	00000110H	nextPC	PIC7
		10	INTTAB0OV	TAB0 overflow	TAB0	0120H	00000120H	nextPC	TAB0OVIC
		11	INTTAB0CC0	TAB0 capture 0/compare 0 match	TAB0	0130H	00000130H	nextPC	TAB0CCIC0
		12	INTTAB0CC1	TAB0 capture 1/compare 1 match	TAB0	0140H	00000140H	nextPC	TAB0CCIC1
		13	INTTAB0CC2	TAB0 capture 2/compare 2 match	TAB0	0150H	00000150H	nextPC	TAB0CCIC2
		14	INTTAB0CC3	TAB0 capture 3/compare 3 match	TAB0	0160H	00000160H	nextPC	TAB0CCIC3

Notes 1. DP: Default priority

2. For the restoring in the case of INTWDT2, see **18.2.2 (2) From INTWDT2 signal**.

3. n = 0H to FH

Table 18-3. Interrupt Source List of V850ES/HG3 (2/3)

Type	Classification	DP[Note]	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	15	INTTAA0OV	TAA0 overflow	TAA0	0170H	00000170H	nextPC	TAA0OVIC
		16	INTTAA0CC0	TAA0 capture 0/compare 0 match	TAA0	0180H	00000180H	nextPC	TAA0CCIC0
		17	INTTAA0CC1	TAA0 capture 1/compare 1 match	TAA0	0190H	00000190H	nextPC	TAA0CCIC1
		18	INTTAA1OV	TAA1 overflow	TAA1	01A0H	000001A0H	nextPC	TAA1OVIC
		19	INTTAA1CC0	TAA1 capture 0/compare 0 match	TAA1	01B0H	000001B0H	nextPC	TAA1CCIC0
		20	INTTAA1CC1	TAA1 capture 1/compare 1 match	TAA1	01C0H	000001C0H	nextPC	TAA1CCIC1
		21	INTTAA2OV	TAA2 overflow	TAA2	01D0H	000001D0H	nextPC	TAA2OVIC
		22	INTTAA2CC0	TAA2 capture 0/compare 0 match	TAA2	01E0H	000001E0H	nextPC	TAA2CCIC0
		23	INTTAA2CC1	TAA2 capture 1/compare 1 match	TAA2	01F0H	000001F0H	nextPC	TAA2CCIC1
		24	INTTAA3OV	TAA3 overflow	TAA3	0200H	00000200H	nextPC	TAA3OVIC
		25	INTTAA3CC0	TAA3 capture 0/compare 0 match	TAA3	0210H	00000210H	nextPC	TAA3CCIC0
		26	INTTAA3CC1	TAA3 capture 1/compare 1 match	TAA3	0220H	00000220H	nextPC	TAA3CCIC1
		27	INTTAA4OV	TAA4 overflow	TAA4	0230H	00000230H	nextPC	TAA4OVIC
		28	INTTAA4CC0	TAA4 capture 0/compare 0 match	TAA4	0240H	00000240H	nextPC	TAA4CCIC0
		29	INTTAA4CC1	TAA4 capture 1/compare 1 match	TAA4	0250H	00000250H	nextPC	TAA4CCIC1
		30	INTTM0EQ0	TMM0 compare match	TMM0	0260H	00000260H	nextPC	TM0EQIC0
		31	INTCB0R	CSIB0 reception completion/error	CSIB0	0270H	00000270H	nextPC	CB0RIC
		32	INTCB0T	CSIB0 consecutive transmission write enable	CSIB0	0280H	00000280H	nextPC	CB0TIC
		33	INTCB1R	CSIB1 reception completion/error	CSIB1	0290H	00000290H	nextPC	CB1RIC
		34	INTCB1T	CSIB1 consecutive transmission write enable	CSIB1	02A0H	000002A0H	nextPC	CB1TIC
		35	INTUD0S	UARTD0 status	UARTD0	02B0H	000002B0H	nextPC	UD0SIC
		36	INTUD0R	UARTD0 reception completion	UARTD0	02C0H	000002C0H	nextPC	UD0RIC
		37	INTUD0T	UARTD0 transmission enable	UARTD0	02D0H	000002D0H	nextPC	UD0TIC
		38	INTUD1S	UARTD1 status	UARTD1	02E0H	000002E0H	nextPC	UD1SIC
		39	INTUD1R	UARTD1 reception completion	UARTD1	02F0H	000002F0H	nextPC	UD1RIC
		40	INTUD1T	UARTD1 transmission enable	UARTD1	0300H	00000300H	nextPC	UD1TIC
		41	INTIIC0	IIC0 transfer end	IIC0	0310H	00000310H	nextPC	IIC0IC
		42	INTAD	A/D conversion end	A/D	0320H	00000320H	nextPC	ADIC
		43	INTDMA0	DMA0 transfer end	DMA	0370H	00000370H	nextPC	DMAIC0

Note DP: Default priority

Table 18-3. Interrupt Source List of V850ES/HG3 (3/3)

Type	Classification	DP[Note]	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	44	INTDMA1	DMA1 transfer end	DMA	0380H	00000380H	nextPC	DMAIC1
		45	INTDMA2	DMA2 transfer end	DMA	0390H	00000390H	nextPC	DMAIC2
		46	INTDMA3	DMA3 transfer end	DMA	03A0H	000003A0H	nextPC	DMAIC3
		47	INTKR	Key return interrupt request	KR	03B0H	000003B0H	nextPC	KRIC
		48	INTWTI	Watch timer interval	WT	03C0H	000003C0H	nextPC	WTIIC
		49	INTWT	Watch timer reference time	WT	03D0H	000003D0H	nextPC	WTIC
		50	INTP8	External interrupt pin input edge detection (INTP8)	Pin	0400H	00000400H	nextPC	PIC8
		51	INTP9	External interrupt pin input edge detection (INTP9)	Pin	0410H	00000410H	nextPC	PIC9
		52	INTP10	External interrupt pin input edge detection (INTP10)	Pin	0420H	00000420H	nextPC	PIC10
		53	INTTAB1OV	TAB1 overflow	TAB1	0430H	00000430H	nextPC	TAB1OVIC
		54	INTTAB1CC0	TAB1 capture 0/compare 0 match	TAB1	0440H	00000440H	nextPC	TAB1CCIC0
		55	INTTAB1CC1	TAB1 capture 1/compare 1 match	TAB1	0450H	00000450H	nextPC	TAB1CCIC1
		56	INTTAB1CC2	TAB1 capture 2/compare 2 match	TAB1	0460H	00000460H	nextPC	TAB1CCIC2
		57	INTTAB1CC3	TAB1 capture 3/compare 3 match	TAB1	0470H	00000470H	nextPC	TAB1CCIC3
		58	INTUD2S	UARTD2 status	UARTD2	0480H	00000480H	nextPC	UA2SIC
		59	INTUD2R	UARTD2 reception completion/error	UARTD2	0490H	00000490H	nextPC	UA2RIC
		60	INTUD2T	UARTD2 transmission enable	UARTD2	04A0H	000004A0H	nextPC	UA2TIC

Note DP: Default priority

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

The priority order of non-maskable interrupt is INTWDT2 > NMI.

Restored PC: The value of the program counter (PC) saved to EIPC, FEPC, or DBPC when interrupt servicing is started. Note, however, that the restored PC when a non-maskable or maskable interrupt is acknowledged while one of the following instructions is being executed does not become the nextPC (if an interrupt is acknowledged during interrupt execution, execution stops, and then resumes after the interrupt servicing has finished).

- Load instructions (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)
- Division instructions (DIV, DIVH, DIVU, DIVHU)
- PREPARE, DISPOSE instructions (only if an interrupt is generated before the stack pointer is updated)

nextPC: The PC value that starts the processing following interrupt/exception processing.

2. The execution address of the illegal instruction when an illegal opcode exception occurs is calculated by (Restored PC − 4).

Table 18-4. Interrupt Source List of V850ES/HJ3 (1/4)

Type	Classification	DP[Note 1]	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Reset	Interrupt	–	RESET	$\overline{\text{RESET}}$ pin input Reset input by internal source	RESET	0000H	00000000H	Undefined	–
Non-maskable	Interrupt	–	NMI	NMI pin valid edge input	Pin	0010H	00000010H	nextPC	–
		–	INTWDT2	WDT2 overflow	WDT2	0020H	00000020H	**Note 2**	–
Software exception	Exception	–	TRAP0n[Note 3]	TRAP instruction	–	004nH[Note 3]	00000040H	nextPC	–
		–	TRAP1n[Note 3]	TRAP instruction	–	005nH[Note 3]	00000050H	nextPC	–
Exception trap	Exception	–	ILGOP/ DBG0	Illegal opcode/ DBTRAP instruction	–	0060H	00000060H	nextPC	–
Maskable	Interrupt	0	INTLVIL	Low voltage detection (voltage under reference level)	POCLVI	0080H	00000080H	nextPC	LVILIC
		1	INTLVIH	Low voltage detection (voltage over reference level)	POCLVI	0090H	00000090H	nextPC	LVIHIC
		2	INTP0	External interrupt pin input edge detection (INTP0)	Pin	00A0H	000000A0H	nextPC	PIC0
		3	INTP1	External interrupt pin input edge detection (INTP1)	Pin	00B0H	000000B0H	nextPC	PIC1
		4	INTP2	External interrupt pin input edge detection (INTP2)	Pin	00C0H	000000C0H	nextPC	PIC2
		5	INTP3	External interrupt pin input edge detection (INTP3)	Pin	00D0H	000000D0H	nextPC	PIC3
		6	INTP4	External interrupt pin input edge detection (INTP4)	Pin	00E0H	000000E0H	nextPC	PIC4
		7	INTP5	External interrupt pin input edge detection (INTP5)	Pin	00F0H	000000F0H	nextPC	PIC5
		8	INTP6	External interrupt pin input edge detection (INTP6)	Pin	0100H	00000100H	nextPC	PIC6
		9	INTP7	External interrupt pin input edge detection (INTP7)	Pin	0110H	00000110H	nextPC	PIC7
		10	INTTAB0OV	TAB0 overflow	TAB0	0120H	00000120H	nextPC	TAB0OVIC
		11	INTTAB0CC0	TAB0 capture 0/compare 0 match	TAB0	0130H	00000130H	nextPC	TAB0CCIC0
		12	INTTAB0CC1	TAB0 capture 1/compare 1 match	TAB0	0140H	00000140H	nextPC	TAB0CCIC1
		13	INTTAB0CC2	TAB0 capture 2/compare 2 match	TAB0	0150H	00000150H	nextPC	TAB0CCIC2
		14	INTTAB0CC3	TAB0 capture 3/compare 3 match	TAB0	0160H	00000160H	nextPC	TAB0CCIC3

Notes 1. DP: Default priority

2. For the restoring in the case of INTWDT2, see **18.2.2 (2) From INTWDT2 signal**.

3. n = 0H to FH

Table 18-4. Interrupt Source List of V850ES/HJ3 (2/4)

Type	Classification	DP[Note 1]	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	15	INTTAA0OV	TAA0 overflow	TAA0	0170H	00000170H	nextPC	TAA0OVIC
		16	INTTAA0CC0	TAA0 capture 0/compare 0 match	TAA0	0180H	00000180H	nextPC	TAA0CCIC0
		17	INTTAA0CC1	TAA0 capture 1/compare 1 match	TAA0	0190H	00000190H	nextPC	TAA0CCIC1
		18	INTTAA1OV	TAA1 overflow	TAA1	01A0H	000001A0H	nextPC	TAA1OVIC
		19	INTTAA1CC0	TAA1 capture 0/compare 0 match	TAA1	01B0H	000001B0H	nextPC	TAA1CCIC0
		20	INTTAA1CC1	TAA1 capture 1/compare 1 match	TAA1	01C0H	000001C0H	nextPC	TAA1CCIC1
		21	INTTAA2OV	TAA2 overflow	TAA2	01D0H	000001D0H	nextPC	TAA2OVIC
		22	INTTAA2CC0	TAA2 capture 0/compare 0 match	TAA2	01E0H	000001E0H	nextPC	TAA2CCIC0
		23	INTTAA2CC1	TAA2 capture 1/compare 1 match	TAA2	01F0H	000001F0H	nextPC	TAA2CCIC1
		24	INTTAA3OV	TAA3 overflow	TAA3	0200H	00000200H	nextPC	TAA3OVIC
		25	INTTAA3CC0	TAA3 capture 0/compare 0 match	TAA3	0210H	00000210H	nextPC	TAA3CCIC0
		26	INTTAA3CC1	TAA3 capture 1/compare 1 match	TAA3	0220H	00000220H	nextPC	TAA3CCIC1
		27	INTTAA4OV	TAA4 overflow	TAA4	0230H	00000230H	nextPC	TAA4OVIC
		28	INTTAA4CC0	TAA4 capture 0/compare 0 match	TAA4	0240H	00000240H	nextPC	TAA4CCIC0
		29	INTTAA4CC1	TAA4 capture 1/compare 1 match	TAA4	0250H	00000250H	nextPC	TAA4CCIC1
		30	INTTM0EQ0	TMM0 compare match	TMM0	0260H	00000260H	nextPC	TM0EQIC0
		31	INTCB0R	CSIB0 reception completion/error	CSIB0	0270H	00000270H	nextPC	CB0RIC
		32	INTCB0T	CSIB0 consecutive transmission write enable	CSIB0	0280H	00000280H	nextPC	CB0TIC
		33	INTCB1R	CSIB1 reception completion/error	CSIB1	0290H	00000290H	nextPC	CB1RIC
		34	INTCB1T	CSIB1 consecutive transmission write enable	CSIB1	02A0H	000002A0H	nextPC	CB1TIC
		35	INTUD0S	UARTD0 status	UARTD0	02B0H	000002B0H	nextPC	UD0SIC
		36	INTUD0R	UARTD0 reception completion	UARTD0	02C0H	000002C0H	nextPC	UD0RIC
		37	INTUD0T	UARTD0 transmission enable	UARTD0	02D0H	000002D0H	nextPC	UD0TIC
		38	INTUD1S	UARTD1 status	UARTD1	02E0H	000002E0H	nextPC	UD1SIC
		39	INTUD1R	UARTD1 reception completion	UARTD1	02F0H	000002F0H	nextPC	UD1RIC
		40	INTUD1T	UARTD1 transmission enable	UARTD1	0300H	00000300H	nextPC	UD1TIC
		41	INTIIC0	IIC0 transfer end	IIC0	0310H	00000310H	nextPC	IIC0IC
			INTUD4S[Note 2]	UATD4 status[Note 2]	UARTD4[Note 2]				UD4SIC[Note 2]
		42	INTAD	A/D conversion end	A/D	0320H	00000320H	nextPC	ADIC
		43	INTDMA0	DMA0 transfer end	DMA	0370H	00000370H	nextPC	DMAIC0

Notes 1. DP: Default priority

2. μPD70F3757 only

Table 18-4. Interrupt Source List of V850ES/HJ3 (3/4)

Type	Classification	DP[Note 1]	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	44	INTDMA1	DMA1 transfer end	DMA	0380H	00000380H	nextPC	DMAIC1
		45	INTDMA2	DMA2 transfer end	DMA	0390H	00000390H	nextPC	DMAIC2
		46	INTDMA3	DMA3 transfer end	DMA	03A0H	000003A0H	nextPC	DMAIC3
		47	INTKR	Key return interrupt request	KR	03B0H	000003B0H	nextPC	KRIC
		48	INTWTI	Watch timer interval	WT	03C0H	000003C0H	nextPC	WTIIC
		49	INTWT	Watch timer reference time	WT	03D0H	000003D0H	nextPC	WTIC
		50	INTP8	External interrupt pin input edge detection (INTP8)	Pin	0400H	00000400H	nextPC	PIC8
		51	INTP9	External interrupt pin input edge detection (INTP9)	Pin	0410H	00000410H	nextPC	PIC9
		52	INTP10	External interrupt pin input edge detection (INTP10)	Pin	0420H	00000420H	nextPC	PIC10
		53	INTTAB1OV	TAB1 overflow	TAB1	0430H	00000430H	nextPC	TAB1OVIC
		54	INTTAB1CC0	TAB1 capture 0/compare 0 match	TAB1	0440H	00000440H	nextPC	TAB1CCIC0
		55	INTTAB1CC1	TAB1 capture 1/compare 1 match	TAB1	0450H	00000450H	nextPC	TAB1CCIC1
		56	INTTAB1CC2	TAB1 capture 2/compare 2 match	TAB1	0460H	00000460H	nextPC	TAB1CCIC2
		57	INTTAB1CC3	TAB1 capture 3/compare 3 match	TAB1	0470H	00000470H	nextPC	TAB1CCIC3
		58	INTUD2S	UARTD2 status	UARTD2	0480H	00000480H	nextPC	UA2SIC
		59	INTUD2R	UARTD2 reception completion/error	UARTD2	0490H	00000490H	nextPC	UA2RIC
		60	INTUD2T	UARTD2 transmission enable	UARTD2	04A0H	000004A0H	nextPC	UA2TIC
		61	INTP11	External interrupt pin input edge detection (INTP11)	Pin	04F0H	000004F0H	nextPC	PIC11
		62	INTP12	External interrupt pin input edge detection (INTP12)	Pin	0500H	00000500H	nextPC	PIC12
		63	INTP13	External interrupt pin input edge detection (INTP13)	Pin	0510H	00000510H	nextPC	PIC13
		64	INTP14	External interrupt pin input edge detection (INTP14)	Pin	0520H	00000520H	nextPC	PIC14
		65	INTUD3S[Note 2]	UARTD3 status[Note 2]	UARTD3[Note 2]	0530H	00000530H	nextPC	UA3SIC[Note 2]
		66	INTUD3R[Note 2]	UARTD3 reception completion[Note 2]	UARTD3[Note 2]	0540H	00000540H	nextPC	UA3RIC[Note 2]
		67	INTUD3T[Note 2]	UARTD3 transmission enable[Note 2]	UARTD3[Note 2]	0550H	00000550H	nextPC	UA3TIC[Note 2]
		68	INTUD4R[Note 2]	UARTD4 reception completion[Note 2]	UARTD4[Note 2]	0560H	00000560H	nextPC	UA4RIC[Note 2]
		69	INTUD4T[Note 2]	UARTD4 transmission enable[Note 2]	UARTD4[Note 2]	0570H	00000570H	nextPC	UA4TIC[Note 2]
		70	INTTAB2OV	TAB2 overflow	TAB2	0580H	00000580H	nextPC	TAB2OVIC

Notes 1. DP: Default priority

2. μPD70F3757 only

Table 18-4. Interrupt Source List of V850ES/HJ3 (4/4)

Type	Classification	DP[Note 1]	Name	Trigger	Generating Unit	Exception Code	Handler Address	Restored PC	Interrupt Control Register
Maskable	Interrupt	71	INTTAB2CC0	TAB2 capture 0/compare 0 match	TAB2	0590H	00000590H	nextPC	TAB2CCIC0
		72	INTTAB2CC1	TAB2 capture 1/compare 1 match	TAB2	05A0H	000005A0H	nextPC	TAB2CCIC1
		73	INTTAB2CC2	TAB2 capture 2/compare 2 match	TAB2	05B0H	000005B0H	nextPC	TAB2CCIC2
		74	INTTAB2CC3	TAB2 capture 3/compare 3 match	TAB2	05C0H	000005C0H	nextPC	TAB2CCIC3
		75	INTUD5S[Note 2]	UARTD5 status[Note 2]	UARTD5[Note 2]	05D0H	000005D0H	nextPC	UA5SIC[Note 2]
		76	INTUD5R[Note 2]	UARTD5 reception completion[Note 2]	UARTD5[Note 2]	05E0H	000005E0H	nextPC	UA5RIC[Note 2]
			INTCB2R	CSIB2 reception completion/error	CSIB2				CB2RIC
		77	INTUD5T[Note 2]	UARTD5 transmission enable[Note 2]	UARTD5[Note 2]	05F0H	000005F0H	nextPC	UA5TIC[Note 2]
			INTCB2T	CSIB2 consecutive transmission write enable	CSIB2				CB2TIC

Notes 1. DP: Default priority

2. μPD70F3757 only

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

The priority order of non-maskable interrupt is INTWDT2 > NMI.

Restored PC: The value of the program counter (PC) saved to EIPC, FEPC, or DBPC when interrupt servicing is started. Note, however, that the restored PC when a non-maskable or maskable interrupt is acknowledged while one of the following instructions is being executed does not become the nextPC (if an interrupt is acknowledged during interrupt execution, execution stops, and then resumes after the interrupt servicing has finished).

- Load instructions (SLD.B, SLD.BU, SLD.H, SLD.HU, SLD.W)
- Division instructions (DIV, DIVH, DIVU, DIVHU)
- PREPARE, DISPOSE instructions (only if an interrupt is generated before the stack pointer is updated)

nextPC: The PC value that starts the processing following interrupt/exception processing.

2. The execution address of the illegal instruction when an illegal opcode exception occurs is calculated by (Restored PC − 4).

18.2 Non-Maskable Interrupts

A non-maskable interrupt request signal is acknowledged unconditionally, even when interrupts are in the interrupt disabled (DI) status. An NMI is not subject to priority control and takes precedence over all the other interrupt request signals.

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”.

The function of the NMI pin is enabled by setting the PMC0.PMC02 bit to 1 and the INTF0.INTF02 bit and INTR0.INTR02 bit to a desired value, and specifying a desired 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 an NMI is being serviced, it is serviced as follows.

(1) If new NMI request signal is issued while NMI 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 NMI currently under execution has been serviced (after the RETI instruction has been executed).

(2) If INTWDT2 request signal is issued while NMI is being serviced

The INTWDT2 request signal is held pending if the NP bit remains set (1) while the NMI is being serviced. The pending INTWDT2 request signal is acknowledged after the NMI currently under execution has been serviced (after the RETI instruction has been executed).

If the NP bit is cleared (0) while the NMI is being serviced, the newly generated INTWDT2 request signal is executed (the NMI servicing is stopped).

Caution For the non-maskable interrupt servicing executed by the non-maskable interrupt request signal (INTWDT2), see 18.2.2 (2) From INTWDT2 signal.

Figure 18-1. Non-Maskable Interrupt Request Signal Acknowledgment Operation (1/2)

(a) NMI and INTWDT2 request signals generated at the same time

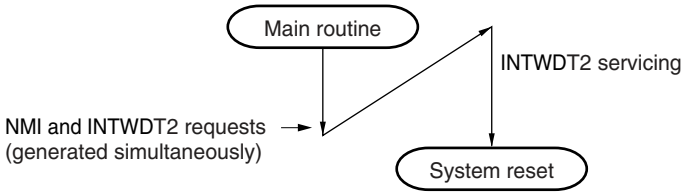
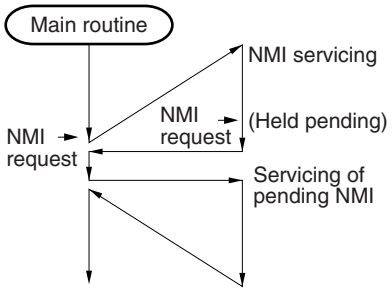
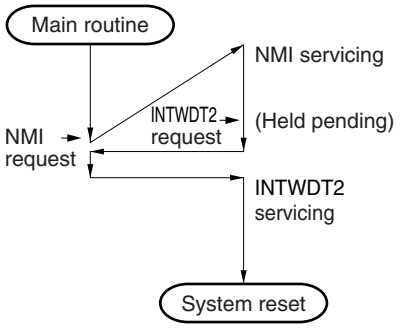
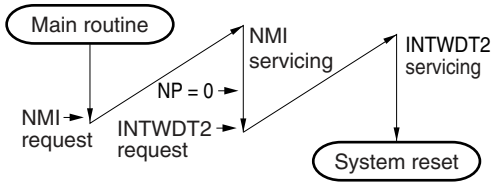
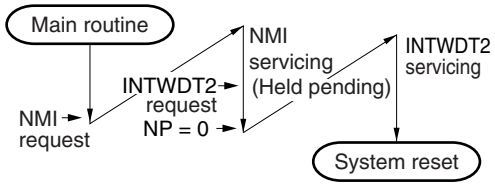
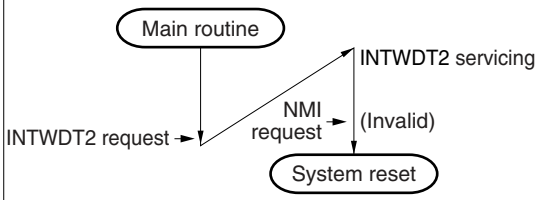
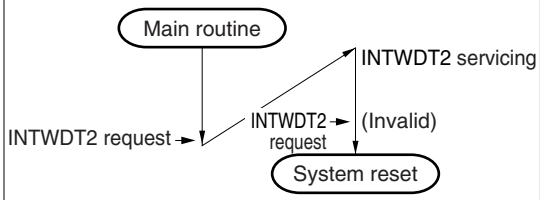


Figure 18-1. Non-Maskable Interrupt Request Signal Acknowledgment Operation (2/2)

(b) Non-maskable interrupt request signal generated during non-maskable interrupt servicing

Non-maskable interrupt being serviced	Non-maskable interrupt request signal generated during non-maskable interrupt servicing	
	NMI	INTWDT2
NMI	• NMI request generated during NMI servicing 	• INTWDT2 request generated during NMI servicing (NP bit = 1 retained before INTWDT2 request)  • INTWDT2 request generated during NMI servicing (NP bit = 0 set before INTWDT2 request)  • INTWDT2 request generated during NMI servicing (NP = 0 set after INTWDT2 request) 
INTWDT2	• NMI request generated during INTWDT2 servicing 	• INTWDT2 request generated during INTWDT2 servicing 

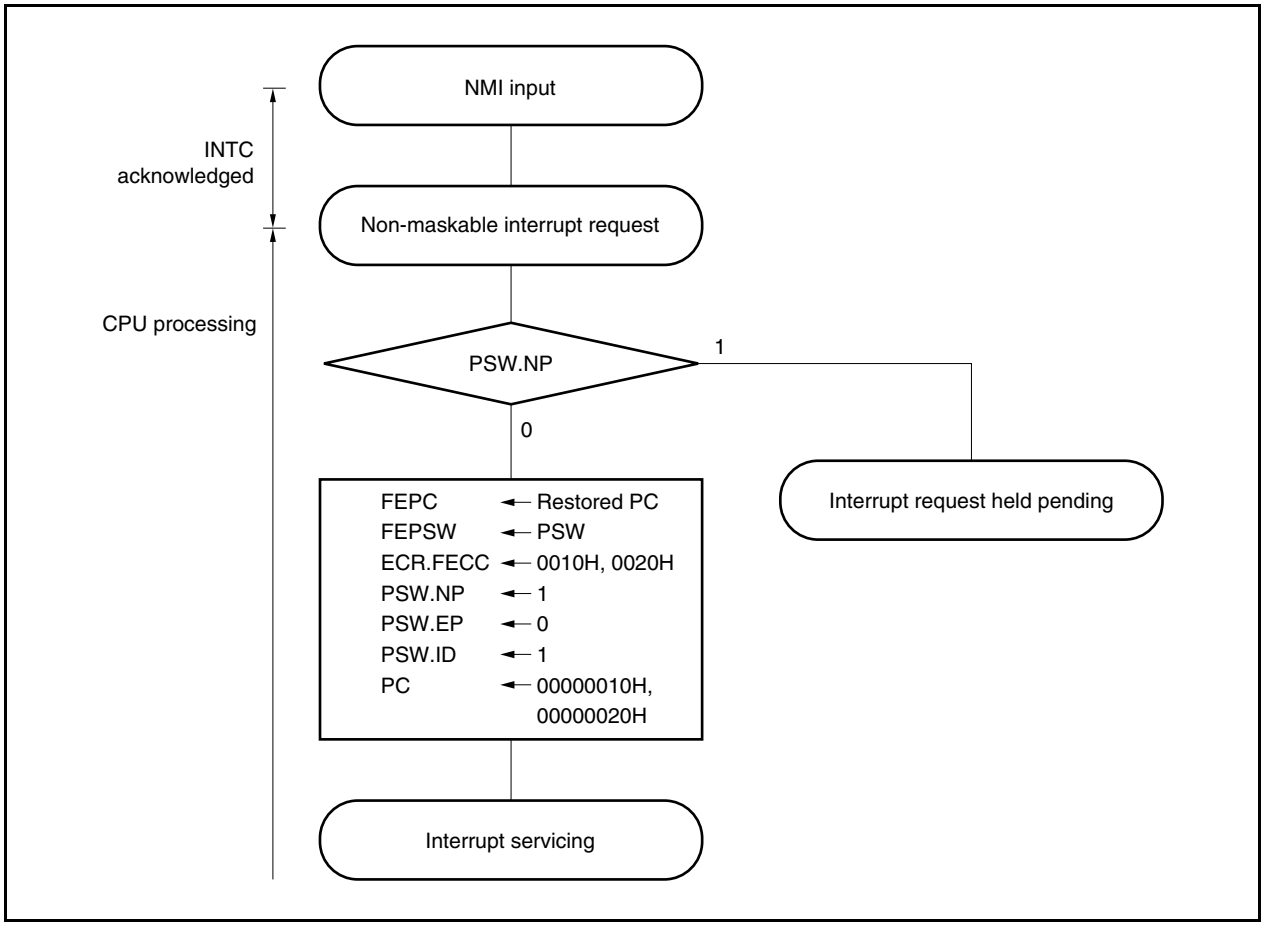
18.2.1 Operation

If a non-maskable interrupt request signal is generated, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to FEPC.
<2> Saves the current PSW to FEPSW.
<3> Writes 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 configuration of a non-maskable interrupt is shown in Figure 18-2.

Figure 18-2. Servicing Configuration of Non-Maskable Interrupt



18.2.2 Restore

(1) From NMI pin input

Execution is restored from the NMI servicing by the RETI instruction.

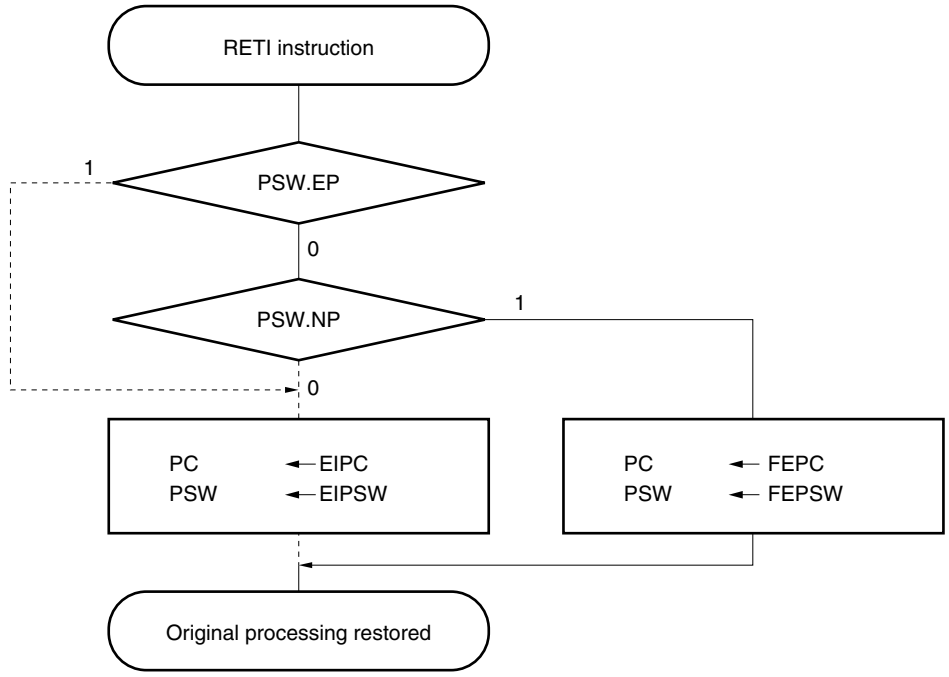
When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

<1> Loads the restored 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.

Figure 18-3 illustrates how the RETI instruction is processed.

Figure 18-3. RETI Instruction Processing



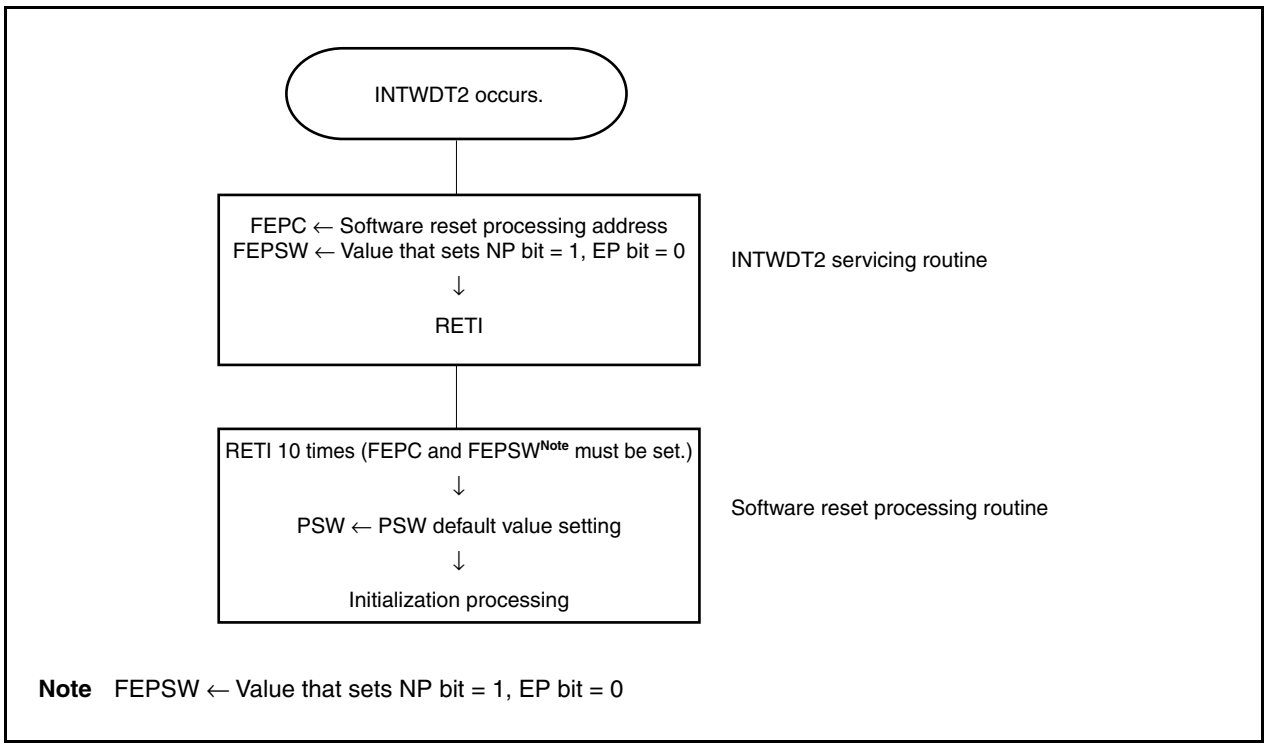
Caution When the EP and NP bits are changed by the LDSR instruction during non-maskable interrupt servicing, in order to restore the PC and PSW correctly during restoration by the RETI instruction, it is necessary to set the EP bit back to 0 and the NP bit back to 1 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

(2) From INTWDT2 signal

Restoring from non-maskable interrupt servicing executed by the non-maskable interrupt request (INTWDT2) by using the RETI instruction is disabled. Execute the following software reset processing.

Figure 18-4. Software Reset Processing



Note FEPSW ← Value that sets NP bit = 1, EP bit = 0

18.2.3 NP flag

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

This flag is set when a non-maskable interrupt request signal has been acknowledged, and masks non-maskable interrupt requests to prohibit multiple interrupts from being acknowledged.

After reset: 00000020H

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

NP	Non-maskable interrupt servicing status
0	No non-maskable interrupt servicing
1	Non-maskable interrupt currently being serviced

18.3 Maskable Interrupts

Maskable interrupt request signals can be masked by interrupt control registers. The V850ES/Hx3 has 50 to 78 maskable interrupt sources.

If two or more maskable interrupt request signals are generated at the same time, they are acknowledged according to the default priority. In addition to the default priority, eight levels of priorities can be specified by using the interrupt control registers (programmable priority control).

When an interrupt request signal has been acknowledged, the acknowledgment of other maskable interrupt request signals is disabled and the interrupt disabled (DI) status is set.

When the EI instruction is executed in an interrupt service routine, the interrupt enabled (EI) status is set, which enables servicing of interrupts having a higher priority than the interrupt request signal in progress (specified by the interrupt control register). Note that only interrupts with a higher priority will have this capability; interrupts with the same priority level cannot be nested.

To enable multiple interrupts, however, save EIPC and EIPSW to memory or general-purpose registers before executing the EI instruction, and execute the DI instruction before the RETI instruction to restore the original values of EIPC and EIPSW.

18.3.1 Operation

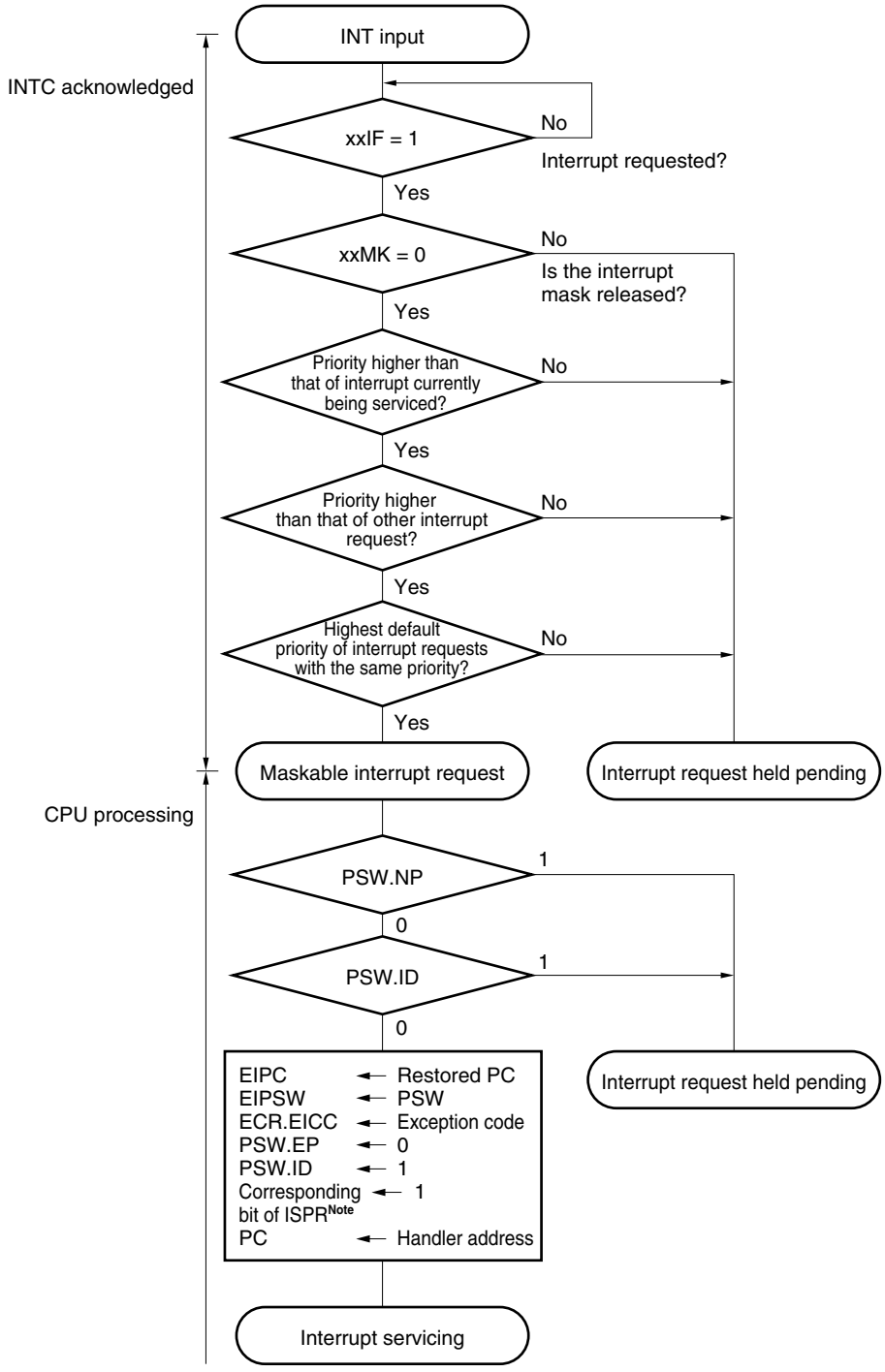
If a maskable interrupt occurs, the CPU performs the following processing, and transfers control to a handler routine.

<1> Saves the restored PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower halfword of ECR (EICC).
<4> Sets the PSW. ID bit to 1 and clears the PSW. EP bit to 0.
<5> Sets the handler address corresponding to each interrupt to the PC, and transfers control.

The maskable interrupt request signal masked by INTC and the maskable interrupt request signal generated while another interrupt is being serviced (while the PSW.NP bit = 1 or the PSW.ID bit = 1) are held pending inside INTC. In this case, servicing a new maskable interrupt is started in accordance with the priority of the pending maskable interrupt request signal if either the maskable interrupt is unmasked or the NP and ID bits are cleared to 0 by using the RETI or LDSR instruction.

How maskable interrupts are serviced is illustrated below.

Figure 18-5. Maskable Interrupt Servicing



Note For the ISPR register, see **18.3.6 In-service priority register (ISPR)**.

18.3.2 Restore

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

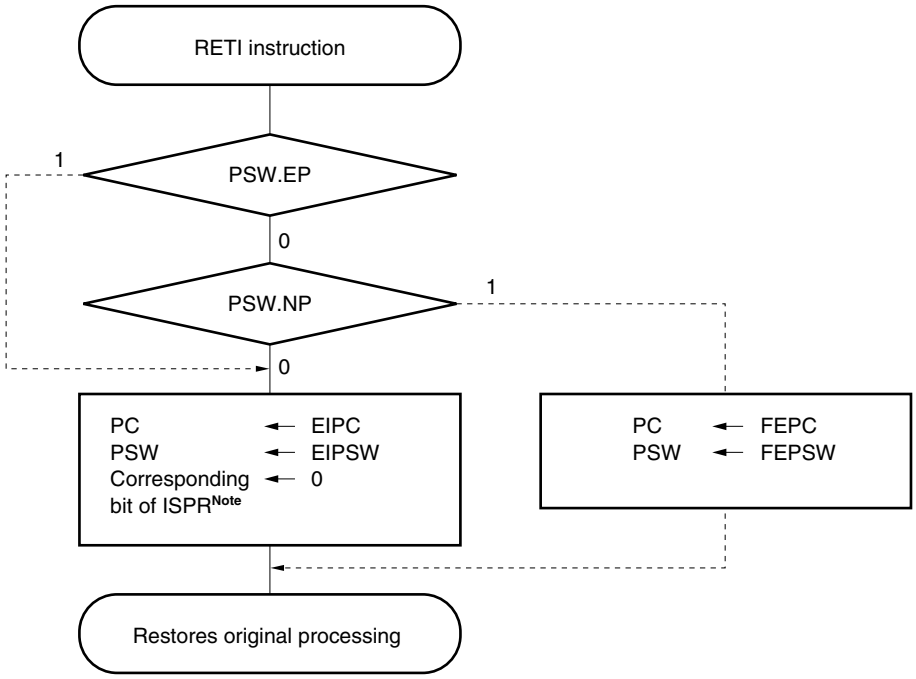
When the RETI instruction is executed, the CPU performs the following steps, and transfers control to the address of the restored PC.

<1> Loads the restored PC and PSW from EIPC and EIPSW because the PSW.EP bit is 0 and the PSW.NP bit is 0.

<2> Transfers control to the address of the restored PC and PSW.

Figure 18-6 illustrates the processing of the RETI instruction.

Figure 18-6. RETI Instruction Processing



Note For the ISPR register, see **18.3.6 In-service priority register (ISPR)**.

Caution When the EP and NP bits are changed by the LDSR instruction during maskable interrupt servicing, in order to restore the PC and PSW correctly during restoration by the RETI instruction, it is necessary to set the EP bit back to 0 and the NP bit back to 0 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

18.3.3 Priorities of maskable interrupts

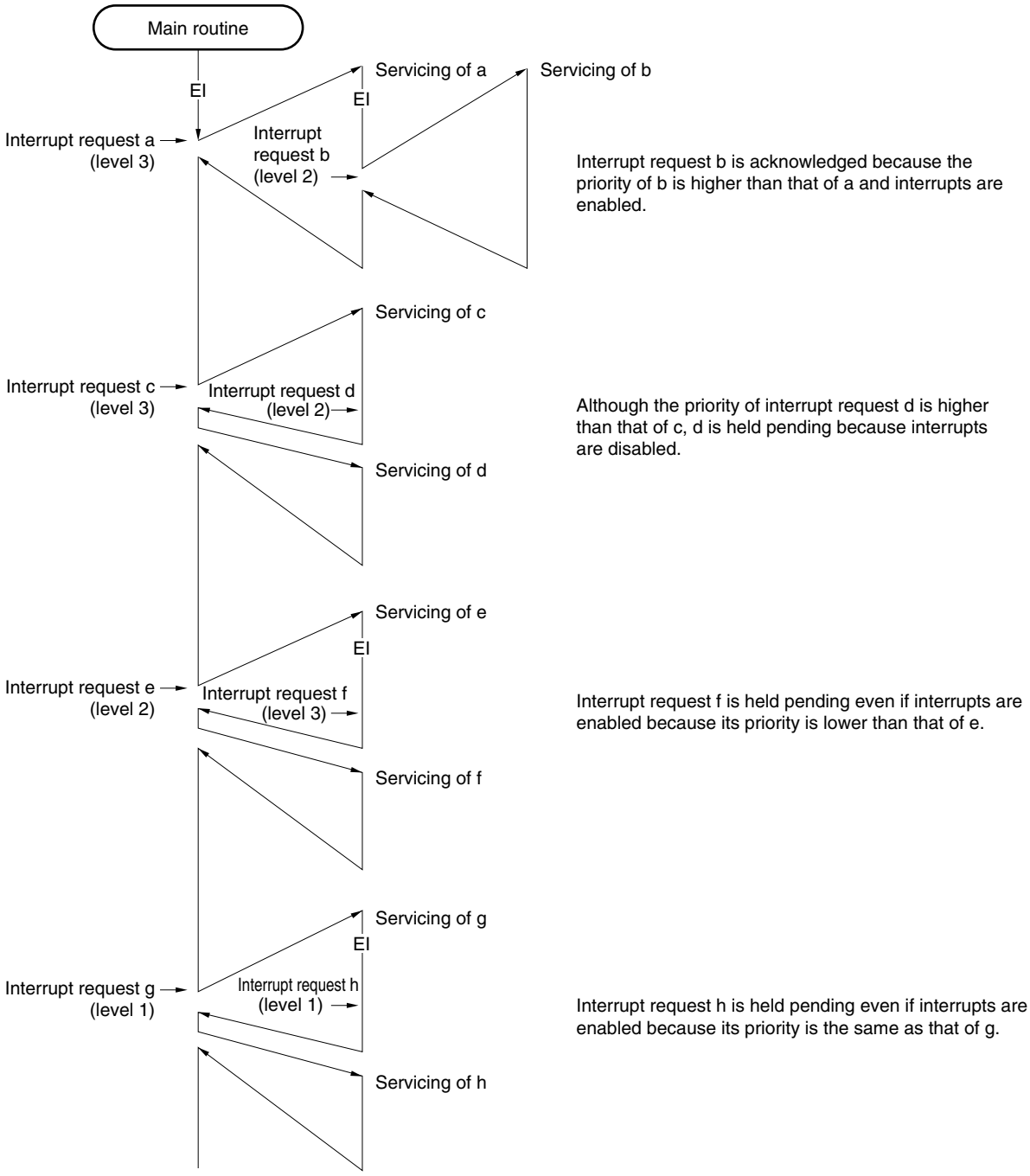
The INTC performs multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced. Multiple interrupts can be controlled by priority levels.

There are two types of priority level control: control based on the default priority levels, and control based on the programmable priority levels that are specified by the interrupt priority level specification bit (xxPRn) of the interrupt control register (xxICn). When two or more interrupts having the same priority level specified by the xxPRn bit are generated at the same time, interrupt request signals are serviced in order depending on the priority level allocated to each interrupt request type (default priority level) beforehand. For more information, see **Tables 18-2** to **18-4 Interrupt/Exception Source List**. The programmable priority control customizes interrupt request signals into eight levels by setting the priority level specification flag.

Note that when an interrupt request signal is acknowledged, the PSW.ID flag is automatically set to 1. Therefore, when multiple interrupts are to be used, clear the ID flag to 0 beforehand (for example, by placing the EI instruction in the interrupt service program) to set the interrupt enable mode.

Remark xx: Identification name of each peripheral unit (see **Table 18-5 Interrupt Control Register (xxICn)**)
n: Peripheral unit number (see **Table 18-5 Interrupt Control Register (xxICn)**).

Figure 18-7. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (1/2)

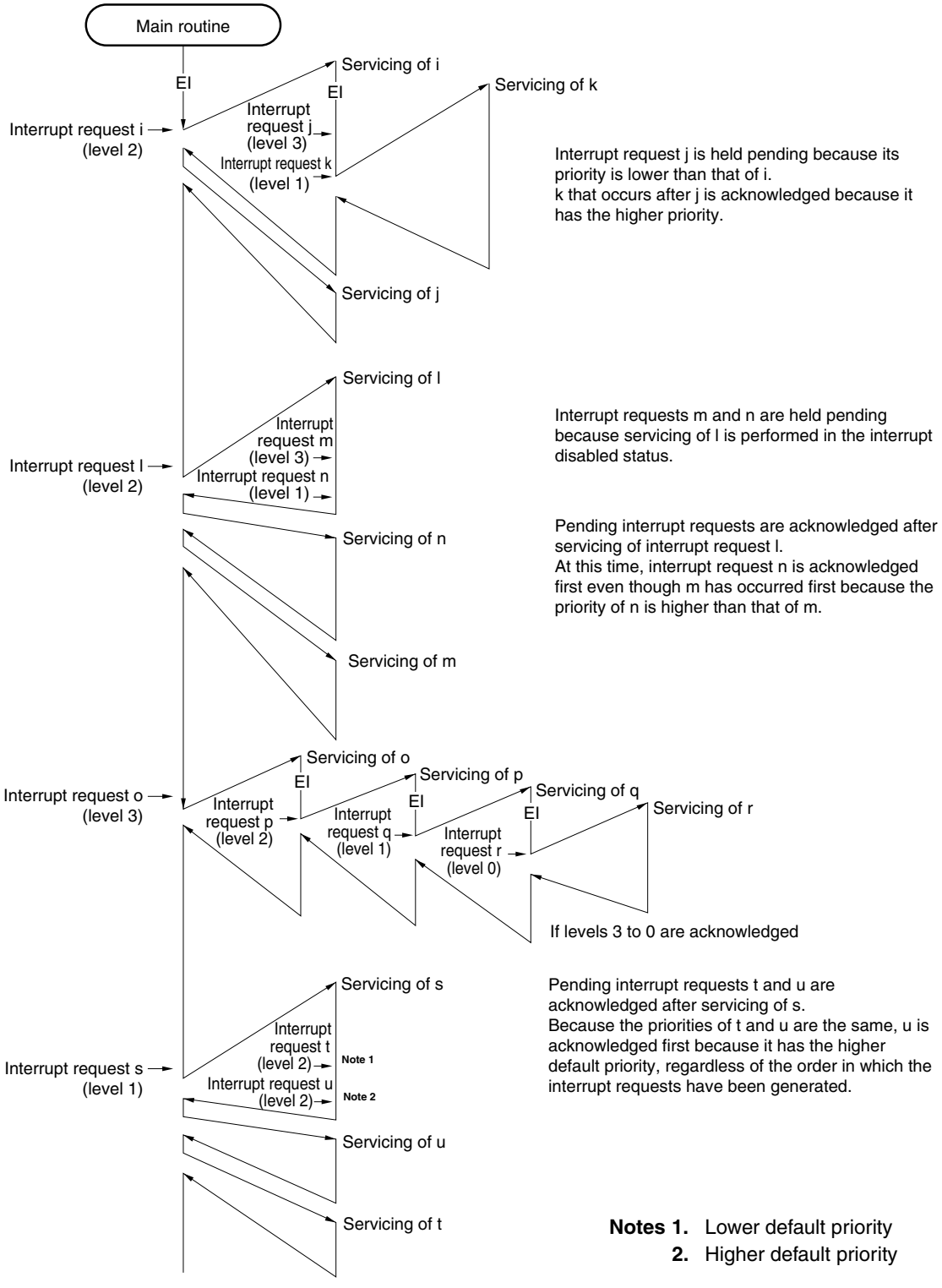


Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Remarks 1. a to u in the figure are the temporary names of interrupt request signals shown for the sake of explanation.

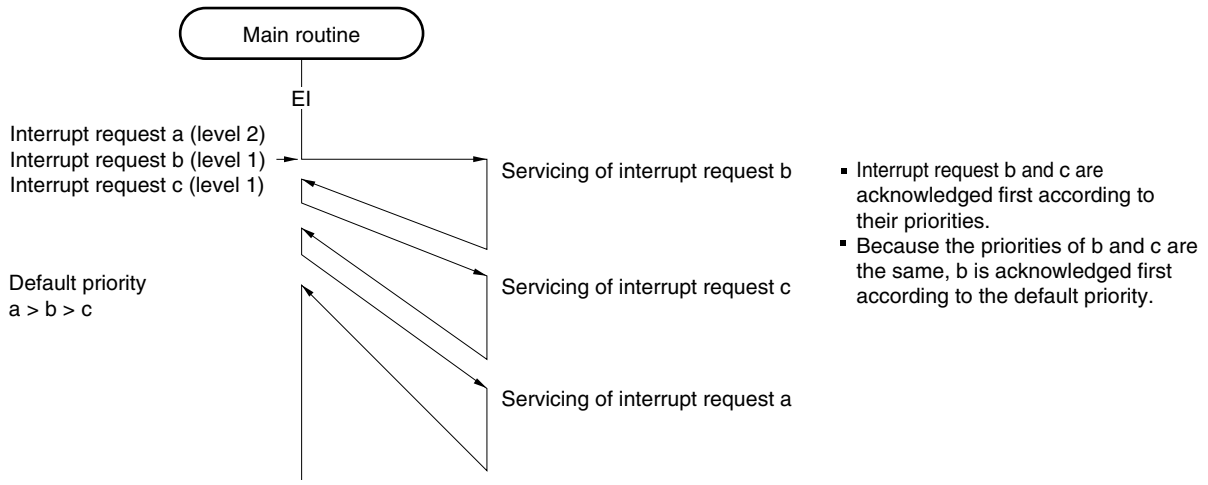
2. The default priority in the figure indicates the relative priority between two interrupt request signals.

Figure 18-7. Example of Processing in Which Another Interrupt Request Signal Is Issued While an Interrupt Is Being Serviced (2/2)



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

Figure 18-8. Example of Servicing Interrupt Request Signals Simultaneously Generated



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

Remarks 1. a to c in the figure are the temporary names of interrupt request signals shown for the sake of explanation.

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

18.3.4 Interrupt control register (xxICn)

The 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.

Caution Disable interrupts (DI) or mask the interrupt to read the xxICn.xxIFn bit. If the xxIFn bit is read while interrupts are enabled (EI) or while the interrupt is unmasked, the correct value may not be read when acknowledging an interrupt and reading the bit conflict.

After reset: 47H R/W Address: FFFFF110H to FFFFF1A2H

	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 18-5 Interrupt Control Registers (xxICn)**)
n: Peripheral unit number (see **Table 18-5 Interrupt Control Registers (xxICn)**).

The addresses and bits of the interrupt control registers are as follows.

Table 18-5. Interrupt Control Registers (xxICn) (1/2)

Address	Register	Bit								Target Product			
		7	6	5	4	3	2	1	0	V850ES/HE3	V850ES/HF3	V850ES/HG3	V850ES/HJ3
FFFFF110H	LVILIC	LVILIF	LVILMK	0	0	0	LVILPR2	LVILPR1	LVILPR0	√	√	√	√
FFFFF112H	LVIHIC	LVIHIF	LVIHMK	0	0	0	LVIHPR2	LVIHPR1	LVIHPR0	√	√	√	√
FFFFF114H	PIC0	PIF0	PMK0	0	0	0	PPR02	PPR01	PPR00	√	√	√	√
FFFFF116H	PIC1	PIF1	PMK1	0	0	0	PPR12	PPR11	PPR10	√	√	√	√
FFFFF118H	PIC2	PIF2	PMK2	0	0	0	PPR22	PPR21	PPR20	√	√	√	√
FFFFF11AH	PIC3	PIF3	PMK3	0	0	0	PPR32	PPR31	PPR30	√	√	√	√
FFFFF11CH	PIC4	PIF4	PMK4	0	0	0	PPR42	PPR41	PPR40	√	√	√	√
FFFFF11EH	PIC5	PIF5	PMK5	0	0	0	PPR52	PPR51	PPR50	√	√	√	√
FFFFF120H	PIC6	PIF6	PMK6	0	0	0	PPR62	PPR61	PPR60	√	√	√	√
FFFFF122H	PIC7	PIF7	PMK7	0	0	0	PPR72	PPR71	PPR70	√	√	√	√
FFFFF124H	TAB0OVIC	TAB0OVIF	TAB0OVMK	0	0	0	TAB0OVPR2	TAB0OVPR1	TAB0OVPR0	√	√	√	√
FFFFF126H	TAB0CCIC0	TAB0CCIF0	TAB0CCMK0	0	0	0	TAB0CCPR02	TAB0CCPR01	TAB0CCPR00	√	√	√	√
FFFFF128H	TAB0CCIC1	TAB0CCIF1	TAB0CCMK1	0	0	0	TAB0CCPR12	TAB0CCPR11	TAB0CCPR10	√	√	√	√
FFFFF12AH	TAB0CCIC2	TAB0CCIF2	TAB0CCMK2	0	0	0	TAB0CCPR22	TAB0CCPR21	TAB0CCPR20	√	√	√	√
FFFFF12CH	TAB0CCIC3	TAB0CCIF3	TAB0CCMK3	0	0	0	TAB0CCPR32	TAB0CCPR31	TAB0CCPR30	√	√	√	√
FFFFF12EH	TAA0OVIC	TAA0OVIF	TAA0OVMK	0	0	0	TAA0OVPR2	TAA0OVPR1	TAA0OVPR0	√	√	√	√
FFFFF130H	TAA0CCIC0	TAA0CCIF0	TAA0CCMK0	0	0	0	TAA0CCPR02	TAA0CCPR01	TAA0CCPR00	√	√	√	√
FFFFF132H	TAA0CCIC1	TAA0CCIF1	TAA0CCMK1	0	0	0	TAA0CCPR12	TAA0CCPR11	TAA0CCPR10	√	√	√	√
FFFFF134H	TAA1OVIC	TAA1OVIF	TAA1OVMK	0	0	0	TAA1OVPR2	TAA1OVPR1	TAA1OVPR0	√	√	√	√
FFFFF136H	TAA1CCIC0	TAA1CCIF0	TAA1CCMK0	0	0	0	TAA1CCPR02	TAA1CCPR01	TAA1CCPR00	√	√	√	√
FFFFF138H	TAA1CCIC1	TAA1CCIF1	TAA1CCMK1	0	0	0	TAA1CCPR12	TAA1CCPR11	TAA1CCPR10	√	√	√	√
FFFFF13AH	TAA2OVIC	TAA2OVIF	TAA2OVMK	0	0	0	TAA2OVPR2	TAA2OVPR1	TAA2OVPR0	√	√	√	√
FFFFF13CH	TAA2CCIC0	TAA2CCIF0	TAA2CCMK0	0	0	0	TAA2CCPR02	TAA2CCPR01	TAA2CCPR00	√	√	√	√
FFFFF13EH	TAA2CCIC1	TAA2CCIF1	TAA2CCMK1	0	0	0	TAA2CCPR12	TAA2CCPR11	TAA2CCPR10	√	√	√	√
FFFFF140H	TAA3OVIC	TAA3OVIF	TAA3OVMK	0	0	0	TAA3OVPR2	TAA3OVPR1	TAA3OVPR0	√	√	√	√
FFFFF142H	TAA3CCIC0	TAA3CCIF0	TAA3CCMK0	0	0	0	TAA3CCPR02	TAA3CCPR01	TAA3CCPR00	√	√	√	√
FFFFF144H	TAA3CCIC1	TAA3CCIF1	TAA3CCMK1	0	0	0	TAA3CCPR12	TAA3CCPR11	TAA3CCPR10	√	√	√	√
FFFFF146H	TAA4OVIC	TAA4OVIF	TAA4OVMK	0	0	0	TAA4OVPR2	TAA4OVPR1	TAA4OVPR0	√	√	√	√
FFFFF148H	TAA4CCIC0	TAA4CCIF0	TAA4CCMK0	0	0	0	TAA4CCPR02	TAA4CCPR01	TAA4CCPR00	√	√	√	√
FFFFF14AH	TAA4CCIC1	TAA4CCIF1	TAA4CCMK1	0	0	0	TAA4CCPR12	TAA4CCPR11	TAA4CCPR10	√	√	√	√
FFFFF14CH	TM0EQIC0	TM0EQIF0	TM0EQMK0	0	0	0	TM0EQPR02	TM0EQPR01	TM0EQPR00	√	√	√	√
FFFFF14EH	CB0RIC	CB0RIF	CB0RMK	0	0	0	CB0RPR2	CB0RPR1	CB0RPR0	√	√	√	√
FFFFF150H	CB0TIC	CB0TIF	CB0TMK	0	0	0	CB0TPR2	CB0TPR1	CB0TPR0	√	√	√	√
FFFFF152H	CB1RIC	CB1RIF	CB1RMK	0	0	0	CB1RPR2	CB1RPR1	CB1RPR0	√	√	√	√
FFFFF154H	CB1TIC	CB1TIF	CB1TMK	0	0	0	CB1TPR2	CB1TPR1	CB1TPR0	√	√	√	√
FFFFF156H	UD0SIC	UD0SIF	UD0SMK	0	0	0	UD0SPR2	UD0SPR1	UD0SPR0	√	√	√	√
FFFFF158H	UD0RIC	UD0RIF	UD0RMK	0	0	0	UD0RPR2	UD0RPR1	UD0RPR0	√	√	√	√
FFFFF15AH	UD0TIC	UD0TIF	UD0TMK	0	0	0	UD0TPR2	UD0TPR1	UD0TPR0	√	√	√	√
FFFFF15CH	UD1SIC	UD1SIF	UD1SMK	0	0	0	UD1SPR2	UD1SPR1	UD1SPR0	√	√	√	√
FFFFF15EH	UD1RIC	UD1RIF	UD1RMK	0	0	0	UD1RPR2	UD1RPR1	UD1RPR0	√	√	√	√
FFFFF160H	UD1TIC	UD1TIF	UD1TMK	0	0	0	UD1TPR2	UD1TPR1	UD1TPR0	√	√	√	√

Table 18-5. Interrupt Control Registers (xxICn) (2/2)

Address	Register	Bit								Target Product			
		7	6	5	4	3	2	1	0	V850ES/HE3	V850ES/HF3	V850ES/HG3	V850ES/HJ3
FFFFF162H	IIC0IC	IIC0IF	IIC0MK	0	0	0	IIC0PR2	IIC0PR1	IIC0PR0	√	√	√	√
	UD4SIC	UD4SIF	UD4SMK	0	0	0	UD4SPR2	UD4SPR1	UD4SPR0				√Note
FFFFF164H	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0	√	√	√	√
FFFFF16EH	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00	√	√	√	√
FFFFF170H	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10	√	√	√	√
FFFFF172H	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20	√	√	√	√
FFFFF174H	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30	√	√	√	√
FFFFF176H	KRIC	KRIF	KRMK	0	0	0	KRPR2	KRPR1	KRPR0	√	√	√	√
FFFFF178H	WTIIC	WTIIF	WTIMK	0	0	0	WTIPR2	WTIPR1	WTIPR0	√	√	√	√
FFFFF17AH	WTIC	WTIF	WTMK	0	0	0	WTPR2	WTPR1	WTPR0	√	√	√	√
FFFFF180H	PIC8	PIF8	PMK8	0	0	0	PPR82	PPR81	PPR80			√	√
FFFFF182H	PIC9	PIF9	PMK9	0	0	0	PPR92	PPR91	PPR90			√	√
FFFFF184H	PIC10	PIF10	PMK10	0	0	0	PPR102	PPR101	PPR100			√	√
FFFFF186H	TAB1OVIC	TAB1OVIF	TAB1OVMK	0	0	0	TAB1OVPR2	TAB1OVPR1	TAB1OVPR0			√	√
FFFFF188H	TAB1CCIC0	TAB1CCIF0	TAB1CCMK0	0	0	0	TAB1CCPR02	TAB1CCPR01	TAB1CCPR00			√	√
FFFFF18AH	TAB1CCIC1	TAB1CCIF1	TAB1CCMK1	0	0	0	TAB1CCPR12	TAB1CCPR11	TAB1CCPR10			√	√
FFFFF18CH	TAB1CCIC2	TAB1CCIF2	TAB1CCMK2	0	0	0	TAB1CCPR22	TAB1CCPR21	TAB1CCPR20			√	√
FFFFF18EH	TAB1CCIC3	TAB1CCIF3	TAB1CCMK3	0	0	0	TAB1CCPR32	TAB1CCPR31	TAB1CCPR30			√	√
FFFFF190H	UD2SIC	UD2SIF	UD2SMK	0	0	0	UD2SPR2	UD2SPR1	UD2SPR0			√	√
FFFFF192H	UD2RIC	UD2RIF	UD2RMK	0	0	0	UD2RPR2	UD2RPR1	UD2RPR0			√	√
FFFFF194H	UD2TIC	UD2TIF	UD2TMK	0	0	0	UD2TPR2	UD2TPR1	UD2TPR0			√	√
FFFFF19EH	PIC11	PIF11	PMK11	0	0	0	PPR112	PPR111	PPR110			√	√
FFFFF1A0H	PIC12	PIF12	PMK12	0	0	0	PPR122	PPR121	PPR120			√	√
FFFFF1A2H	PIC13	PIF13	PMK13	0	0	0	PPR132	PPR131	PPR130			√	√
FFFFF1A4H	PIC14	PIF14	PMK14	0	0	0	PPR142	PPR141	PPR140			√	√
FFFFF1A6H	UD3SIC	UD3SIF	UD3SMK	0	0	0	UD3SPR2	UD3SPR1	UD3SPR0				√Note
FFFFF1A8H	UD3RIC	UD3RIF	UD3RMK	0	0	0	UD3RPR2	UD3RPR1	UD3RPR0				√Note
FFFFF1AAH	UD3TIC	UD3TIF	UD3TMK	0	0	0	UD3TPR2	UD3TPR1	UD3TPR0				√Note
FFFFF1ACH	UD4RIC	UD4RIF	UD4RMK	0	0	0	UD4RPR2	UD4RPR1	UD4RPR0				√Note
FFFFF1AEH	UD4TIC	UD4TIF	UD4TMK	0	0	0	UD4TPR2	UD4TPR1	UD4TPR0				√Note
FFFFF1B0H	TAB2OVIC	TAB2OVIF	TAB2OVMK	0	0	0	TAB2OVPR2	TAB2OVPR1	TAB2OVPR0				√
FFFFF1B2H	TAB2CCIC0	TAB2CCIF0	TAB2CCMK0	0	0	0	TAB2CCPR02	TAB2CCPR01	TAB2CCPR00				√
FFFFF1B4H	TAB2CCIC1	TAB2CCIF1	TAB2CCMK1	0	0	0	TAB2CCPR12	TAB2CCPR11	TAB2CCPR10				√
FFFFF1B6H	TAB2CCIC2	TAB2CCIF2	TAB2CCMK2	0	0	0	TAB2CCPR22	TAB2CCPR21	TAB2CCPR20				√
FFFFF1B8H	TAB2CCIC3	TAB2CCIF3	TAB2CCMK3	0	0	0	TAB2CCPR32	TAB2CCPR31	TAB2CCPR30				√
FFFFF1BAH	UD5SIC	UD5SIF	UD5SMK	0	0	0	UD5SPR2	UD5SPR1	UD5SPR0				√Note
FFFFF1BCH	CB2RIC	CB2RIF	CB2RMK	0	0	0	CB2RPR2	CB2RPR1	CB2RPR0				√
	UD5RIC	UD5RIF	UD5RMK	0	0	0	UD5RPR2	UD5RPR1	UD5RPR0				√Note
FFFFF1BEH	CB2TIC	CB2TIF	CB2TMK	0	0	0	CB2TPR2	CB2TPR1	CB2TPR0				√
	UD5TIC	UD5TIF	UD5TMK	0	0	0	UD5TPR2	UD5TPR1	UD5TPR0				√Note

Note μPD70F3757 only

18.3.5 Interrupt mask registers 0 to 5 (IMR0 to IMR5)

The IMR0 to IMR4 registers set the interrupt mask state for the maskable interrupts. The xxMKn bit of the IMR0 to IMR5 registers is equivalent to the xxICn.xxMKn bit.

The IMRm register can be read or written in 16-bit units (m = 0 to 5).

If the higher 8 bits of the IMRm register are used as 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 5).

Reset sets these registers to FFFFH.

Caution The device file defines the xxICn.xxMKn bit as a reserved word. If a bit is manipulated using the name of xxMKn, the contents of the xxICn register, instead of the IMRm register, are rewritten (as a result, the contents of the IMRm register are also rewritten).

(1) V850ES/HE3, V850ES/HF3

After reset: FFFFH R/W Address: IMR3 FFFFF106H, IMR3L FFFFF106H, IMR3H FFFFF107H

	15	14	13	12	11	10	9	8
IMR3 (IMR3H[Note])	1	1	1	1	1	1	1	1
	7	6	5	4	3	2	1	0
IMR3L	1	1	WTMK	WTIMK	KRMK	DMAMK3	DMAMK2	DMAMK1

After reset: FFFFH R/W Address: IMR2 FFFFF104H, IMR2L FFFFF104H, IMR2H FFFFF105H

	15	14	13	12	11	10	9	8
IMR2 (IMR2H[Note])	DMAMK0	1	1	1	1	ADMK	IIC0MK	UD1TMK
	7	6	5	4	3	2	1	0
IMR2L	UD1RMK	UD1SMK	UD0TMK	UD0RMK	UD0SMK	CB1TMK	CB1RMK	CB0TMK

After reset: FFFFH R/W Address: IMR1 FFFFF102H, IMR1L FFFFF102H, IMR1H FFFFF103H

	15	14	13	12	11	10	9	8
IMR1 (IMR1H[Note])	CB0RMK	TM0EQMK0	TAA4CCMK1	TAA4CCMK0	TAA4OVMK	TAA3CCMK1	TAA3CCMK0	TAA3OVMK
	7	6	5	4	3	2	1	0
IMR1L	TAA2CCMK1	TAA2CCMK0	TAA2OVMK	TAA1CCMK1	TAA1CCMK0	TAA1OVMK	TAA0CCMK1	TAA0CCMK0

After reset: FFFFH R/W Address: IMR0 FFFFF100H, IMR0L FFFFF100H, IMR0H FFFFF101H

	15	14	13	12	11	10	9	8
IMR0 (IMR0H[Note])	TAA0OVMK	TAB0CCMK3	TAB0CCMK2	TAB0CCMK1	TAB0CCMK0	TAB0OVMK	PMK7	PMK6
	7	6	5	4	3	2	1	0
IMR0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIHMK	LVILMK

xxMKn	Setting of interrupt mask flag
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Note To read 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.

Caution Set bits 15 to 6 of the IMR3 register, and bits 14 to 11 of the IMR2 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 18-5 Interrupt Control Registers (xxICn)**).

n: Peripheral unit number (see **Table 18-5 Interrupt Control Registers (xxICn)**)

(2) V850ES/HG3

After reset: FFFFH R/W Address: IMR4 FFFFF108H, IMR4L FFFFF108H, IMR4H FFFFF109H

	15	14	13	12	11	10	9	8
IMR4 (IMR4H[Note])	1	1	1	1	1	1	1	1
	7	6	5	4	3	2	1	0
IMR4L	1	1	1	1	1	UD2TMK	UD2RMK	UD2SMK

After reset: FFFFH R/W Address: IMR3 FFFFF106H, IMR3L FFFFF106H, IMR3H FFFFF107H

	15	14	13	12	11	10	9	8
IMR3 (IMR3H[Note])	TAB1CCMK3	TAB1CCMK2	TAB1CCMK1	TAB1CCMK0	TAB1OVMK	PIC10	PIC9	PIC8
	7	6	5	4	3	2	1	0
IMR3L	1	1	WTMK	WTIMK	KRMK	DMAMK3	DMAMK2	DMAMK1

After reset: FFFFH R/W Address: IMR2 FFFFF104H, IMR2L FFFFF104H, IMR2H FFFFF105H

	15	14	13	12	11	10	9	8
IMR2 (IMR2H[Note])	DMAMK0	1	1	1	1	ADMK	IIC0MK	UD1TMK
	7	6	5	4	3	2	1	0
IMR2L	UD1RMK	UD1SMK	UD0TMK	UD0RMK	UD0SMK	CB1TMK	CB1RMK	CB0TMK

After reset: FFFFH R/W Address: IMR1 FFFFF102H, IMR1L FFFFF102H, IMR1H FFFFF103H

	15	14	13	12	11	10	9	8
IMR1 (IMR1H[Note])	CB0RMK	TM0EQMK0	TAA4CCMK1	TAA4CCMK0	TAA4OVMK	TAA3CCMK1	TAA3CCMK0	TAA3OVMK
	7	6	5	4	3	2	1	0
IMR1L	TAA2CCMK1	TAA2CCMK0	TAA2OVMK	TAA1CCMK1	TAA1CCMK0	TAA1OVMK	TAA0CCMK1	TAA0CCMK0

After reset: FFFFH R/W Address: IMR0 FFFFF100H, IMR0L FFFFF100H, IMR0H FFFFF101H

	15	14	13	12	11	10	9	8
IMR0 (IMR0H[Note])	TAA0OVMK	TAB0CCMK3	TAB0CCMK2	TAB0CCMK1	TAB0CCMK0	TAB0OVMK	PMK7	PMK6
	7	6	5	4	3	2	1	0
IMR0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIHMK	LVILMK

xxMKn	Setting of interrupt mask flag
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Note To read bits 8 to 15 of the IMR0 to IMR4 registers in 8-bit or 1-bit units, specify them as bits 0 to 7 of IMR0H to IMR4H registers.

Caution Set bits 15 to 3 of the IMR4 register, bits 7 and 6 of the IMR3 register, and bits 14 to 11 of the IMR2 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 18-5 Interrupt Control Registers (xxICn)**).

n: Peripheral unit number (see **Table 18-5 Interrupt Control Registers (xxICn)**)

(3) V850ES/HJ3

After reset: FFFFH R/W Address: IMR5 FFFFF10AH, IMR5L FFFFF10AH, IMR5H FFFFF10BH

	15	14	13	12	11	10	9	8
IMR5 (IMR5H[Note 1])	1	1	1	1	1	1	1	1
	7	6	5	4	3	2	1	0
IMR5L	CB2TMK/UD5TMK[Note 2]	CB2RMK/UD5RMK[Note 2]	UD5SMK[Note 2]	TAB2CCMK3	TAB2CCMK2	TAB2CCMK1	TAB2CCMK0	TAB2OVMK

After reset: FFFFH R/W Address: IMR4 FFFFF108H, IMR4L FFFFF108H, IMR4H FFFFF109H

	15	14	13	12	11	10	9	8
IMR4 (IMR4H[Note 1])	UD4TMK[Note 2]	UD4RMK[Note 2]	UD3TMK[Note 2]	UD3RMK[Note 2]	UD3SMK[Note 2]	PMK14	PMK13	PMK12
	7	6	5	4	3	2	1	0
IMR4L	PMK11	1	1	1	1	UD2TMK	UD2RMK	UD2SMK

After reset: FFFFH R/W Address: IMR3 FFFFF106H, IMR3L FFFFF106H, IMR3H FFFFF107H

	15	14	13	12	11	10	9	8
IMR3 (IMR3H[Note 1])	TAB1CCMK3	TAB1CCMK2	TAB1CCMK1	TAB1CCMK0	TAB1OVMK	PIC10	PIC9	PIC8
	7	6	5	4	3	2	1	0
IMR3L	1	1	WTMK	WTIMK	KRMK	DMAMK3	DMAMK2	DMAMK1

After reset: FFFFH R/W Address: IMR2 FFFFF104H, IMR2L FFFFF104H, IMR2H FFFFF105H

	15	14	13	12	11	10	9	8
IMR2 (IMR2H[Note 1])	DMAMK0	1	1	1	1	ADMK	IIC0MK/UD4SMK[Note 2]	UD1TMK
	7	6	5	4	3	2	1	0
IMR2L	UD1RMK	UD1SMK	UD0TMK	UD0RMK	UD0SMK	CB1TMK	CB1RMK	CB0TMK

After reset: FFFFH R/W Address: IMR1 FFFFF102H, IMR1L FFFFF102H, IMR1H FFFFF103H

	15	14	13	12	11	10	9	8
IMR1 (IMR1H[Note 1])	CB0RMK	TM0EQMK0	TAA4CCMK1	TAA4CCMK0	TAA4OVMK	TAA3CCMK1	TAA3CCMK0	TAA3OVMK
	7	6	5	4	3	2	1	0
IMR1L	TAA2CCMK1	TAA2CCMK0	TAA2OVMK	TAA1CCMK1	TAA1CCMK0	TAA1OVMK	TAA0CCMK1	TAA0CCMK0

After reset: FFFFH R/W Address: IMR0 FFFFF100H, IMR0L FFFFF100H, IMR0H FFFFF101H

	15	14	13	12	11	10	9	8
IMR0 (IMR0H[Note 1])	TAA0OVMK	TAB0CCMK3	TAB0CCMK2	TAB0CCMK1	TAB0CCMK0	TAB0OVMK	PMK7	PMK6
	7	6	5	4	3	2	1	0
IMR0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIHMK	LVILMK

xxMKn	Setting of Interrupt mask flag
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Notes 1. To read bits 8 to 15 of the IMR0 to IMR5 registers in 8-bit or 1-bit units, specify them as bits 0 to 7 of IMR0H to IMR5H registers.

2. μPD70F3757 only

Caution Set bits 15 to 8 of the IMR5 register, bits 6 to 3 of the IMR4 register, bits 7 and 6 of the IMR3 register, and bits 14 to 11 of the IMR2 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 18-5 Interrupt Control Registers (xxICn)**).

n: Peripheral unit number (see **Table 18-5 Interrupt Control Registers (xxICn)**)

18.3.6 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).

After reset: 00H R Address: FFFFF1FAH

	7	6	5	4	3	2	1	0
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0

ISPRn	Priority of interrupt currently acknowledged
0	Interrupt request signal with priority n not acknowledged
1	Interrupt request signal with priority n acknowledged

Remark n = 0 to 7 (priority level)

18.3.7 ID flag

This flag controls the maskable interrupt's operating state, and stores control information regarding enabling or disabling of interrupt request signals. An interrupt disable flag (ID) is assigned to the PSW.

Reset sets this flag 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 (pending)

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 modified by the RETI instruction or LDSR instruction when referencing the PSW.

Non-maskable interrupt request signals and exceptions are acknowledged regardless of this flag. When a maskable interrupt request signal is acknowledged, the ID flag is automatically set to 1 by hardware.

The 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.

18.3.8 Watchdog timer mode register 2 (WDTM2)

This register can be read or written in 8-bit units (for details, see **CHAPTER 12 FUNCTIONS OF 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 operation
0	1	Non-maskable interrupt request mode
1	×	Reset mode (initial-value)

18.4 Software Exception

A software exception is generated when the CPU executes the TRAP instruction, and can always be acknowledged.

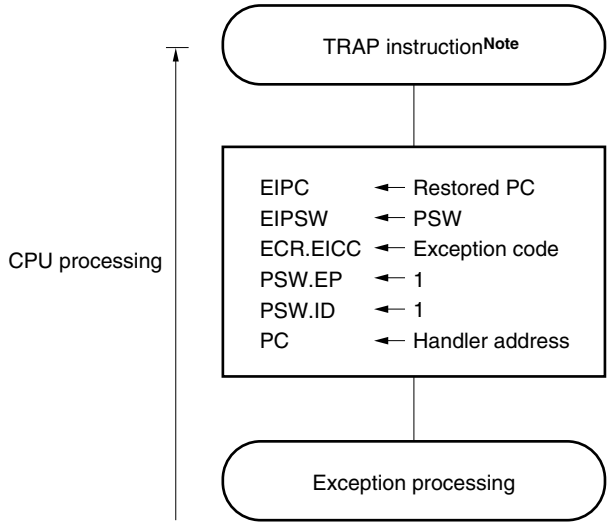
18.4.1 Operation

If a software exception occurs, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to EIPC.
<2> Saves the current PSW to EIPSW.
<3> Writes an exception code to the lower 16 bits (EICC) of ECR (interrupt source).
<4> Sets the PSW.EP and PSW.ID bits to 1.
<5> Sets the handler address (00000040H or 00000050H) corresponding to the software exception to the PC, and transfers control.

Figure 18-9 illustrates the processing of a software exception.

Figure 18-9. Software Exception Processing



Note TRAP instruction format: TRAP vector (the vector is a value from 00H to 1FH.)

The handler address is determined by the TRAP instruction's operand (vector). If the vector is 00H to 0FH, it becomes 00000040H, and if the vector is 10H to 1FH, it becomes 00000050H.

18.4.2 Restore

Restoration from software exception processing is carried out by the RETI instruction.

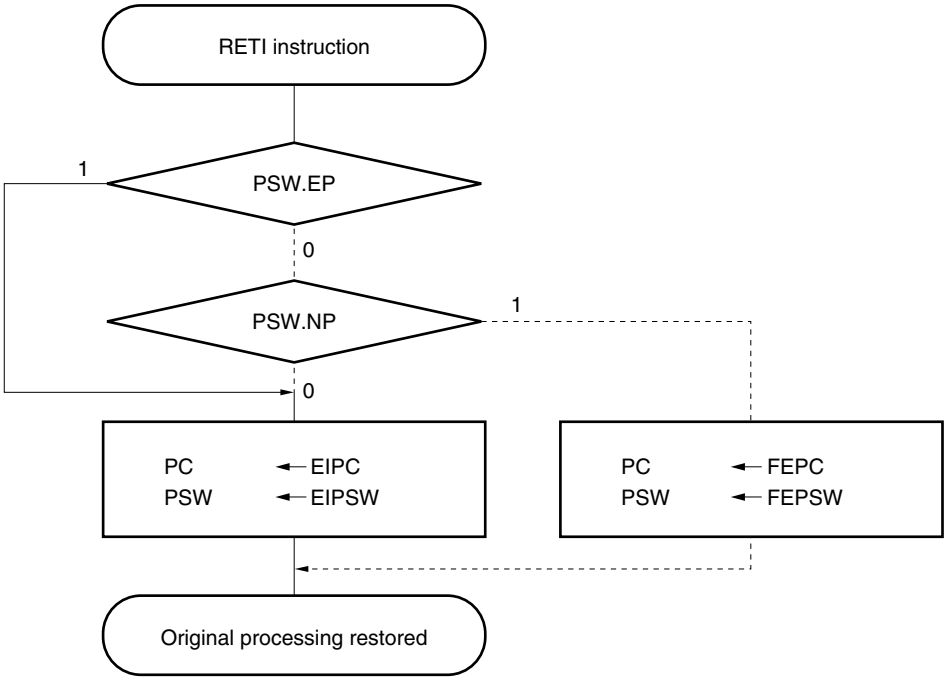
By executing the RETI instruction, the CPU carries out the following processing and shifts control to the restored PC's address.

<1> Loads the restored PC and PSW from EIPC and EIPSW because the PSW.EP bit is 1.

<2> Transfers control to the address of the restored PC and PSW.

Figure 18-10 illustrates the processing of the RETI instruction.

Figure 18-10. RETI Instruction Processing



Caution When the EP and NP bits are changed by the LDSR instruction during the software exception processing, in order to restore the PC and PSW correctly during restoration by the RETI instruction, it is necessary to set the EP bit back to 1 and the NP bit back to 0 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

18.4.3 EP flag

The EP flag is a status flag used to indicate that exception processing is in progress. It is set when an exception occurs.

After reset: 00000020H

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

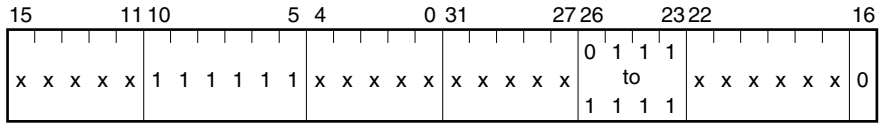
EP	Exception processing status
0	Exception processing not in progress.
1	Exception processing in progress.

18.5 Exception Trap

An exception trap is an interrupt that is requested when the illegal execution of an instruction takes place. In the V850ES/Hx3, an illegal opcode exception (ILGOP: Illegal Opcode Trap) is considered as an exception trap.

18.5.1 Illegal opcode definition

The illegal instruction has an opcode (bits 10 to 5) of 111111B, a sub-opcode (bits 26 to 23) of 0111B to 1111B, and a sub-opcode (bit 16) of 0B. An exception trap is generated when an instruction applicable to this illegal instruction is executed.



x: Arbitrary

Caution Since it is possible to assign this instruction to an illegal opcode in the future, it is recommended that it not be used.

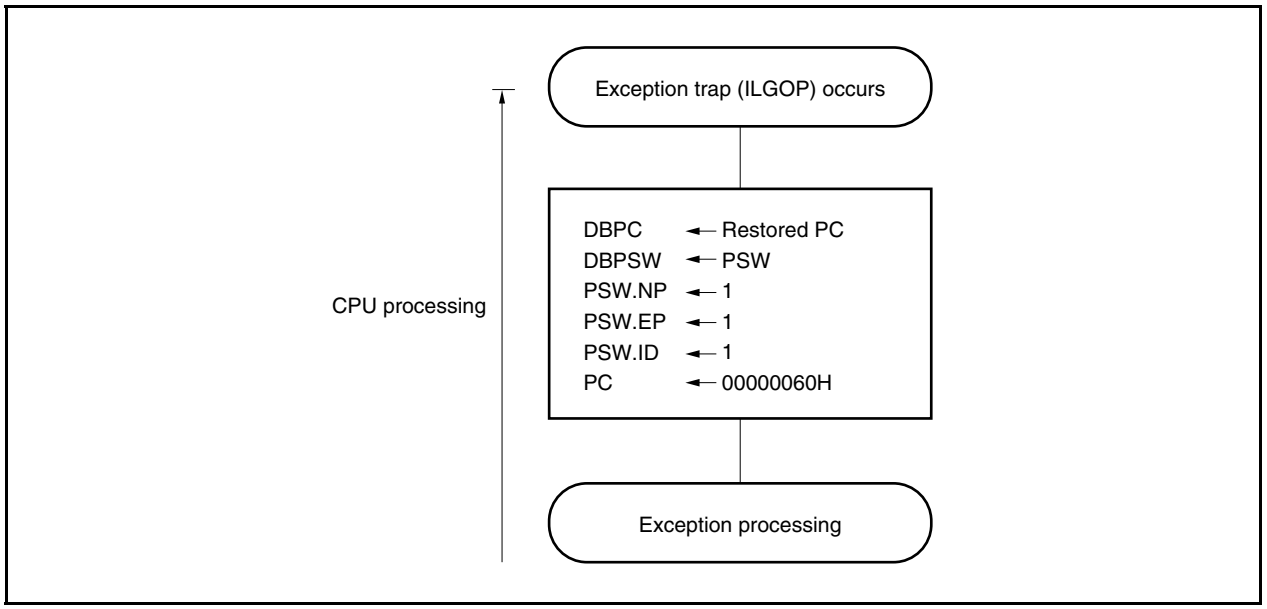
(1) Operation

If an exception trap occurs, the CPU performs the following processing, and transfers control to the handler routine.

<1> Saves the restored PC to DBPC.
<2> Saves the current PSW to DBPSW.
<3> Sets the PSW.NP, PSW.EP, and PSW.ID bits to 1.
<4> Sets the handler address (00000060H) corresponding to the exception trap to the PC, and transfers control.

Figure 18-11 illustrates the processing of the exception trap.

Figure 18-11. Exception Trap Processing



(2) Restore

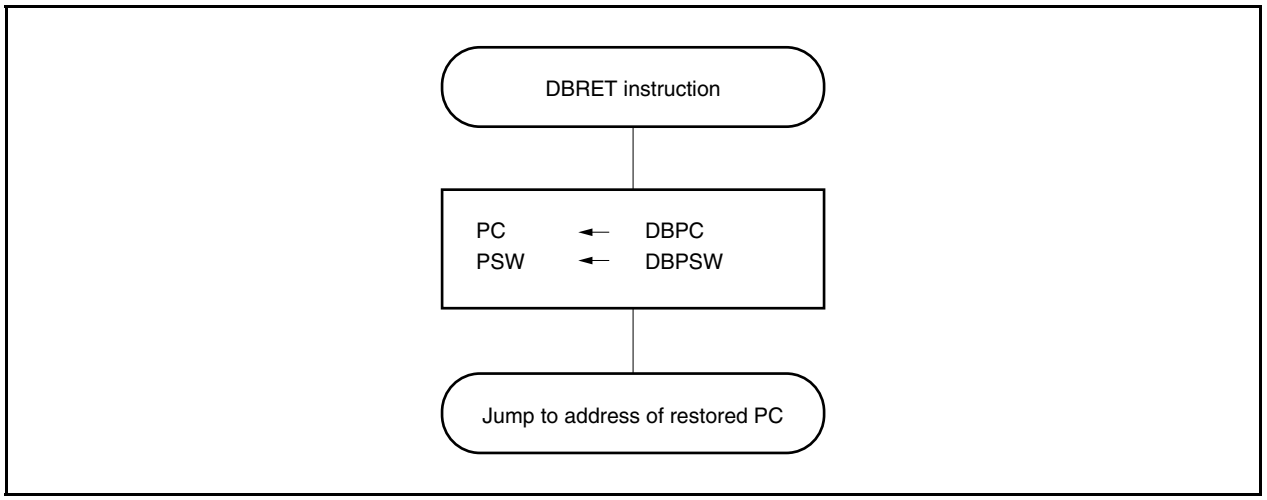
Restoration from an exception trap is carried out by the DBRET instruction. By executing the DBRET instruction, the CPU carries out the following processing and controls the address of the restored PC.

<1> Loads the restored PC and PSW from DBPC and DBPSW.
<2> Transfers control to the address indicated by the restored PC and PSW.

Caution DBPC and DBPSW can be accessed only during the interval between the execution of the illegal opcode and the DBRET instruction.

Figure 18-12 illustrates the restore processing from an exception trap.

Figure 18-12. Restore Processing from Exception Trap



18.5.2 Debug trap

A debug trap is an exception that is generated when the DBTRAP instruction is executed and is always acknowledged.

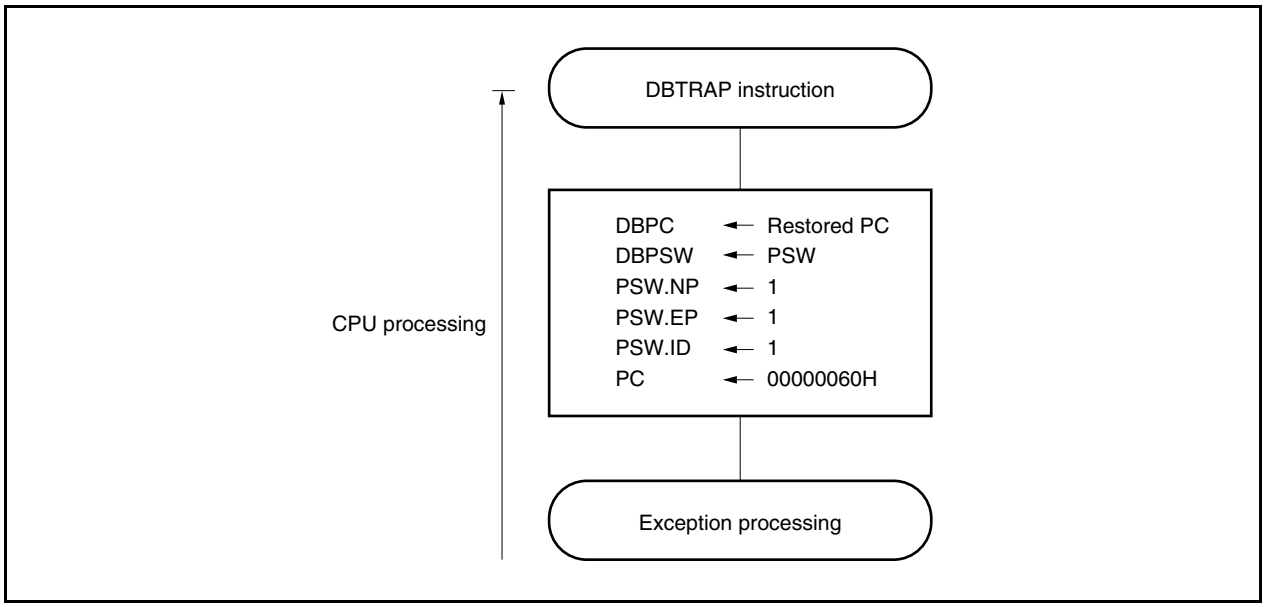
(1) Operation

Upon occurrence of a debug trap, the CPU performs the following processing.

<1> Saves restored PC to DBPC.
<2> Saves current PSW to DBPSW.
<3> Sets the PSW.NP, PSW.EP, and PSW.ID bits to 1.
<4> Sets handler address (00000060H) for debug trap to PC and transfers control.

Figure 18-13 shows the debug trap processing format.

Figure 18-13. Debug Trap Processing Format



(2) Restore

Restoration from a debug trap is executed with the DBRET instruction.

With the DBRET instruction, the CPU performs the following steps and transfers control to the address of the restored PC.

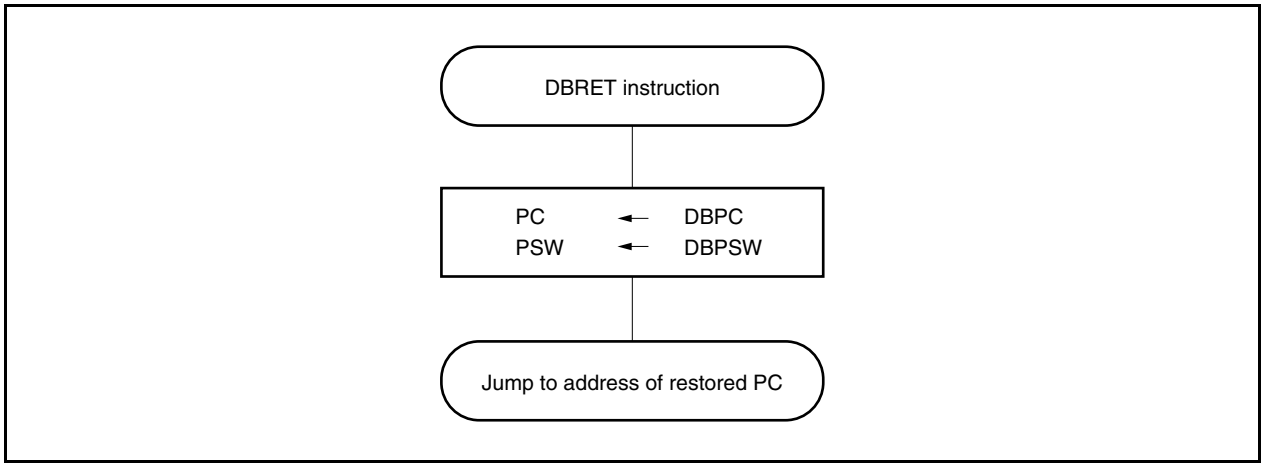
<1> The restored PC and PSW are read from DBPC and DBPSW.

<2> Control is transferred to the fetched 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.

Figure 18-14 shows the processing format for restoration from a debug trap.

Figure 18-14. Processing Format of Restoration from Debug Trap



18.6 External Interrupt Request Input Pins (NMI and INTP0 to INTP14)

18.6.1 Noise elimination

(1) Eliminating noise on NMI pin

The NMI pin has an internal noise elimination circuit that uses analog delay. Therefore, the input level of the NMI pin is not detected as an edge unless it is maintained for a specific time or longer. Therefore, an edge is detected after specific time.

The NMI pin can be used to release the STOP mode. In the STOP mode, noise is not eliminated by using the system clock because the internal system clock is stopped.

(2) Eliminating noise on INTP0 to INTP2 and INTP4 to INTP14 pins

The INTP0 to INTP2 and INTP4 to INTP14 pins have an internal noise elimination circuit that uses analog delay. Therefore, the input level of the NMI pin is not detected as an edge unless it is maintained for a specific time or longer. Therefore, an edge is detected after specific time.

(3) Eliminating noise on INTP3

The INTP3 pin has an internal noise elimination circuit that uses analog delay and an internal digital noise elimination circuit. Either can be selected by using the noise elimination control register (NFC) (see **18.6.2 (8)**).

18.6.2 Edge detection

The valid edge of each of the NMI and INTP0 to INTP14 pins can be selected from the following four.

- Rising edge
- Falling edge
- Both rising and falling edges
- No edge detected

The edge of the NMI pin is not detected after reset. Therefore, the interrupt request signal is not acknowledged unless a valid edge is enabled by using the INTF0 and INTR0 register (the NMI pin functions as a normal port pin).

(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 the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF0n and INTR0n bits to 00, and then set the port mode.

After reset: 00H R/W Address: INTF0 FFFFFC00H, INTR0 FFFFFC20H

	7	6	5	4	3	2	1	0
INTF0	0	INTF06	INTF05	INTF04	INTF03	INTF02	0	0
		INTP3	INTP2	INTP1	INTP0	NMI		

	7	6	5	4	3	2	1	0
INTR0	0	INTR06	INTR05	INTR04	INTR03	INTR02	0	0
		INTP3	INTP2	INTP1	INTP0	NMI		

Remark For the valid edge specification combinations, see **Table 18-6**.

Table 18-6. 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 set the INTF0n and INTR0n bits to 00 if the corresponding pin is not used as 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 rising, falling edge specification register 1 (INTR1, INTF1)

The INTR1 and INTF1 registers are 8-bit registers that specify detection of the rising and falling edges of the INTP9 and INTP10 pins.

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

Reset sets these registers to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF1n and INTR1n bits to 00, and then set the port mode.

After reset: 00H R/W Address: INTF1 FFFFFC02H, INTR1 FFFFFC22H

	7	6	5	4	3	2	1	0
INTF1	0	0	0	0	0	0	INTF11	INTF10
							INTP10	INTP9
INTR1	0	0	0	0	0	0	INTR11	INTR10
							INTP10	INTP9

Remark For the valid edge specification combinations, see **Table 18-7**.

Table 18-7. Valid Edge Specification

INTF1n	INTR1n	Valid Edge Specification (n = 0, 1)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution Be sure to set the INTF1n and INTR1n bits to 00 if the corresponding pin is not used as the INTP9 and INTP10 pins.

Remark n = 0: Control of INTP9 pin
n = 1: Control of INTP10 pin

(3) External interrupt rising, falling edge specification register 3 (INTR3, INTF3)

The INTR3 and INTF3 registers are 8-bit registers that specify detection of the rising and falling edges of the INTP7 and INTP8 pins.

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

However, when the higher 8 bits of INTF3 register are used as the INTF3H register and the lower 8 bits as the INTF3L register, they can be read or written in 8-bit or 1-bit units.

Reset sets these registers to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF3n and INTR3n bits to 00, and then set the port mode.

After reset: 0000H R/W Address: INTF3 FFFFFC06H, INTF3L FFFFFC06H, INTF3H FFFFFC07H

	7	6	5	4	3	2	1	0
INTF3 (INTF3H[Note])	0	0	0	0	0	0	INTF39	0
							INTP8	
(INTF3L)	0	0	0	0	0	0	INTF31	0
							INTP7	

After reset: 0000H R/W Address: INTR3 FFFFFC26H, INTR3L FFFFFC26H, INTR3H FFFFFC27H

	7	6	5	4	3	2	1	0
INTR3 (INTR3H[Note])	0	0	0	0	0	0	INTR39	0
							INTP8	
(INTR3L)	0	0	0	0	0	0	INTR31	0
							INTP7	

Caution When bits 8 to 15 of the INTF3 and INTR3 registers are read or written in 8-bit or 1-bit units, specify them as bits 0 to 7 of the INTF3H and INTR3H registers.

Remark For the valid edge specification combinations, see **Table 18-8**.

Table 18-8. Valid Edge Specification

INTF3n	INTR3n	Valid Edge Specification (n = 1, 9)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution Be sure to set the INTF3n and INTR3n bits to 00 if the corresponding pin is not used as the INTP7 and INTP8 pins.

Remark n = 1: Control of INTP7 pin
n = 9: Control of INTP8 pin

(4) External interrupt rising, falling edge specification registers 4 (INTR4, INTF4)

The INTR4 and INTF4 registers are 8-bit registers that specify detection of the rising and falling edges of the INTP14 pin.

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

Reset sets these registers to 00H.

Cautions 1. When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF40 and INTR40 bits to 00, and then set the port mode.

2. INTP14 is assigned to pins P40 and P80. To use the P40 pin as the INTP14 input, clear the INTF8.INTF80 and INTR8.INTR80 bits to 0 (no edge detected).

After reset: 00H R/W Address: INTF4 FFFFFC08H, INTR4 FFFFFC28H

	7	6	5	4	3	2	1	0
INTF4	0	0	0	0	0	0	0	INTF40
								INTP14
	7	6	5	4	3	2	1	0
INTR4	0	0	0	0	0	0	0	INTR40
								INTP14

Remark For the valid edge specification combinations, see **Table 18-9**.

Table 18-9. Valid Edge Specification

INTF40	INTR40	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 set the INTF40 and INTR40 bits to 00 if the corresponding pin is not used as the INTP14 pin.

(5) External interrupt rising, falling edge specification register 6L (INTR6L, INTF6L)

The INTR6L and INTF6L registers are 8-bit registers that specify detection of the rising and falling edges of the INTP11 to INTP13 pins.

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

Reset sets these registers to 00H.

Caution When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF6n and INTR6n bits to 00, and then set the port mode.

After reset: 00H R/W Address: INTF6L FFFFFC0CH, INTR6L FFFFFC2CH

	7	6	5	4	3	2	1	0
INTF6L	0	0	0	0	0	INTF62	INTF61	INTF60
						INTP13	INTP12	INTP11

	7	6	5	4	3	2	1	0
INTR6L	0	0	0	0	0	INTR62	INTR61	INTR60
						INTP13	INTP12	INTP11

Remark For the valid edge specification combinations, see **Table 18-10**.

Table 18-10. Valid Edge Specification

INTF6n	INTR6n	Valid Edge Specification (n = 0 to 2)
0	0	No edge detected
0	1	Rising edge
1	0	Falling edge
1	1	Both rising and falling edges

Caution Be sure to set the INTF6n and INTR6n bits to 00 if the corresponding pin is not used as the INTP11 to INTP13 pins.

Remark n = 0: Control of INTP11 pin
n = 1: Control of INTP12 pin
n = 2: Control of INTP13 pin

(6) External interrupt falling, rising edge specification registers 8 (INTF8, INTR8)

The INTF8 and INTR8 registers are 8-bit registers that specify detection of the falling and rising edges of the external interrupt pin (INTP14).

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

Reset sets these registers to 00H.

Cautions 1. When the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF80 and INTR80 bits to 00, and then set the port mode.

2. The INTP14 pin and RXDD3 pin are alternate-function pins. When using the pin as the RXDD3 pin, disable edge detection for the INTP14 alternate-function pin (clear the INTF8.INTF80 bit and the INTR8.INTR80 bit to 0). When using the pin as the INTP14 pin, stop UARTD3 reception (clear the UA3CTL0.UA3RXE bit to 0).

3. INTP14 is assigned to pins P40 and P80. To use the P80 pin as the INTP14 input, clear the INTF4.INTF40 and INTR4.INTR40 bits to 0 (no edge detected).

After reset: 00H R/W Address: INTF8 FFFFFC10H, INTR8 FFFFFC30H

	7	6	5	4	3	2	1	0
INTF8	0	0	0	0	0	0	0	INTF80
								INTP14
	7	6	5	4	3	2	1	0
INTR8	0	0	0	0	0	0	0	INTR80
								INTP14

Remark For the valid edge specification combinations, see **Table 18-11**.

Table 18-11. Valid Edge Specification

INTF80	INTR80	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 set the INTF80 and INTR80 bits to 00 if the corresponding pin is not used as the INTP14 pin.

(7) 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 the function is changed from the external interrupt function (alternate function) to the port function, an edge may be detected. Therefore, clear the INTF9n and INTR9n bits to 0, and then set the port mode.

After reset: 00H R/W Address: INTF9H FFFFFC13H, INTR9H FFFFFC33H

	15	14	13	12	11	10	9	8
INTF9H	INTF915	INTF914	INTF913	0	0	0	0	0
	INTP6	INTP5	INTP4					
	15	14	13	12	11	10	9	8
INTR9H	INTR915	INTR914	INTR913	0	0	0	0	0
	INTP6	INTP5	INTP4					

Remark For the valid edge specification combinations, see **Table 18-12**.

Table 18-12. 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 set the INTF9n and INTR9n bits to 00 if the corresponding pin is not used as the INTP4 to INTP6 pins.

Remark n = 13 to 15: Control of INTP4 to INTP6 pins

(8) Noise elimination control register (NFC)

Digital noise elimination can be selected for the INTP3 pin. The noise elimination settings are performed using the NFC register.

When digital noise elimination is selected, the sampling clock for digital sampling can be selected from among $f_{XX}/64$, $f_{XX}/128$, $f_{XX}/256$, $f_{XX}/512$, $f_{XX}/1{,}024$, and f_{XT}. Sampling times are set by the NFC.NFSTS bit.

When digital noise elimination is selected, if the clock that performs sampling in the standby mode is stopped, then the INTP3 interrupt request signal cannot be used for releasing the standby mode. When f_{XT} is used as the sampling clock, the INTP3 interrupt request signal can be used for releasing either the subclock operating mode or the IDLE1/IDLE2/STOP/sub-IDLE mode.

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

Reset sets this register to 00H.

Caution Time equal to the sampling clock × the number of times set by the NFSTS bit is required until the digital noise eliminator is initialized after the sampling clock has been changed. If the valid edge of INTP3 is input after the sampling clock has been changed and before the time of the sampling clock × the number of times set by the NFSTS bit passes, therefore, the interrupt request signal may be generated. Therefore, note the following points when using the interrupt and DMA functions.

- **When using the interrupt function, after the sampling clock × the number of times set by the NFSTS bit 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 the sampling clock × the number of times set by the NFSTS bit have elapsed.**

After reset: 00H R/W Address: FFFFF318H

	7	6	5	4	3	2	1	0
NFC	NFEN	NFSTS	0	0	0	NFC2	NFC1	NFC0

NFEN	Settings of INTP3 pin noise elimination
0	Analog noise elimination
1	Digital noise elimination

NFSTS	Setting of number of times of sampling of digital noise elimination
0	Number of times of sampling × 3 times
1	Number of times of sampling × twice

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 clocks.

2. In the case of noise with a width smaller than 2 sampling clocks, an interrupt request signal is generated if noise synchronized with the sampling clock is input.

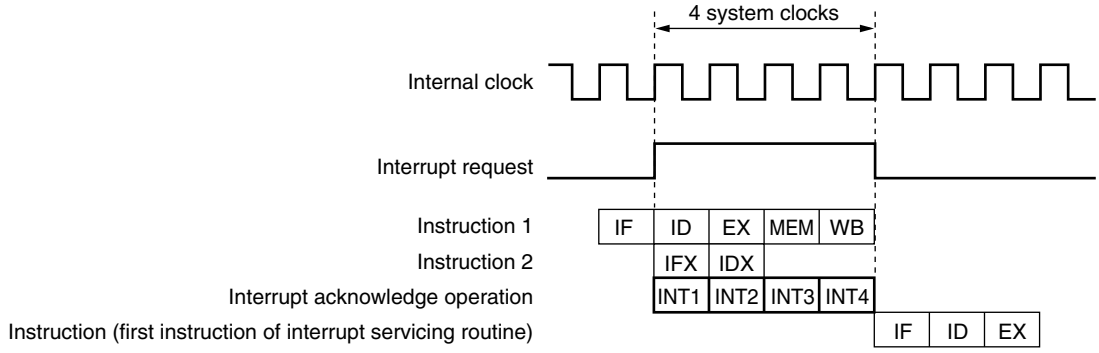
18.7 Interrupt Acknowledge Time of CPU

Except the following cases, the interrupt acknowledge time of the CPU is 4 clocks minimum. To input interrupt request signals successively, input the next interrupt request signal at least 5 clocks after the preceding interrupt.

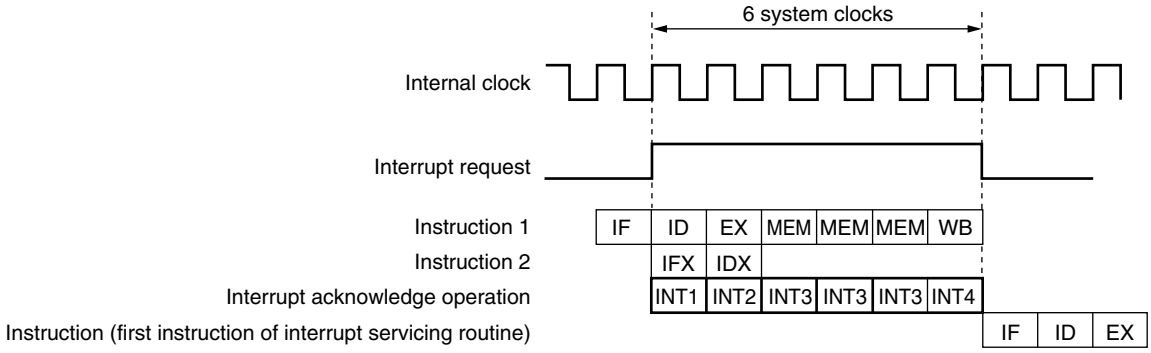
- In IDLE1/IDLE2/STOP mode
- When the external bus is accessed
- When interrupt request non-sampling instructions are successively executed (see **18.8 Periods in Which Interrupts Are Not Acknowledged by CPU**.)
- When the interrupt control register is accessed

Figure 18-15. Pipeline Operation at Interrupt Request Signal Acknowledgment (Outline)

(1) Minimum interrupt response time



(2) Maximum interrupt response time



Remark INT1 to INT4: Interrupt acknowledgment processing
IFX: Invalid instruction fetch
IDX: Invalid instruction decode

Interrupt acknowledge time (internal system clock)			Condition
	Internal interrupt	External interrupt	
Minimum	4	4 + Analog delay time	The following cases are exceptions. • In IDLE1/IDLE2/STOP mode • External bus access • Two or more interrupt request non-sample instructions are executed in succession • Access to peripheral I/O register
Maximum	6[Note]	6[Note] + Analog delay time	

Note This value is 7 in the μPD70F3757.

18.8 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 (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 5 (IMR0 to IMR5)
  - Command register (PRCMD)
  - Power save control register (PSC)
  - On-chip debug mode register (OCDM)
  - Peripheral emulation register 1 (PEMU1):

Remark xx: Identification name of each peripheral unit (see **Table 18-5 Interrupt Control Registers (xxICn)**)
n: Peripheral unit number (see **Table 18-5 Interrupt Control Registers (xxICn)**).

18.9 Cautions

The NMI pin alternately functions as the P02 pin. It functions as a normal port pin after reset. To enable the NMI pin, validate the NMI pin with the PMC0 register. The initial setting of the NMI pin is “No edge detected”. Select the NMI pin valid edge using the INTF0 and INTR0 registers.

CHAPTER 19 KEY INTERRUPT FUNCTION

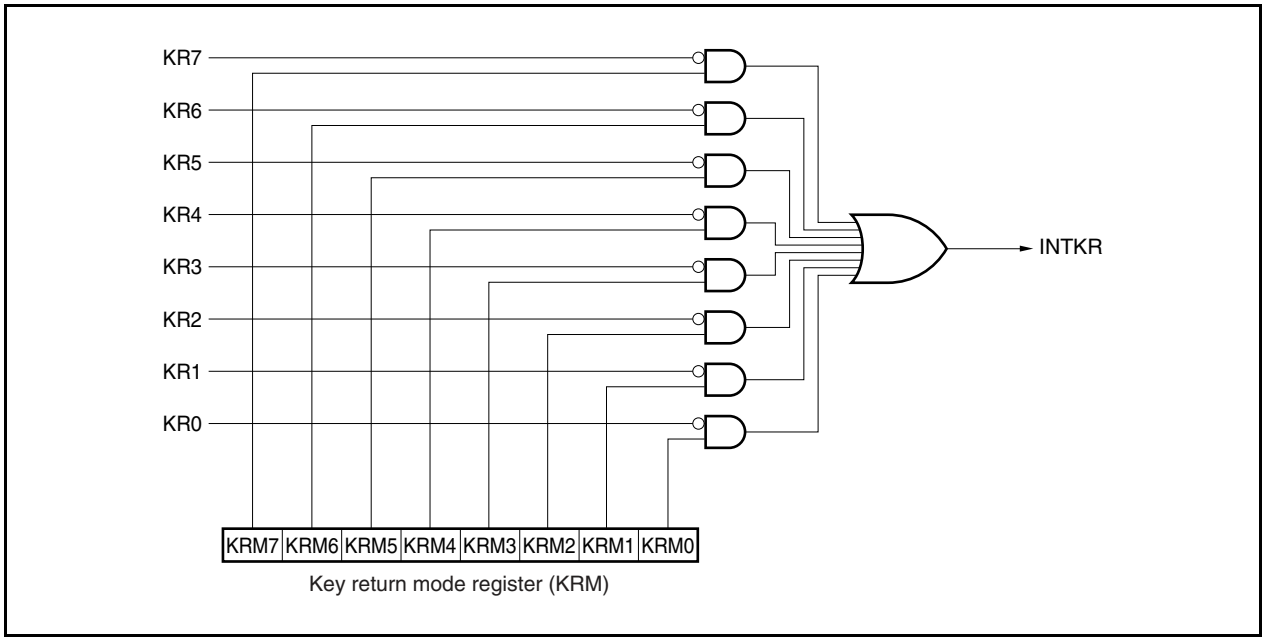
19.1 Function

A key interrupt request signal (INTKR) can be generated by detecting a falling edge to the eight key input pins (KR0 to KR7) by setting the key return mode register (KRM).

Table 19-1. Key Return Flag Function

Flag	Pin to Be Set	Alternate-Function Port
KRM0	KR0 signal	P40
KRM1	KR1 signal	P41/P51
KRM2	KR2 signal	P42/P52
KRM3	KR3 signal	P53
KRM4	KR4 signal	P54
KRM5	KR5 signal	P55
KRM6	KR6 signal	P90
KRM7	KR7 signal	P91

Figure 19-1. Key Return Block Diagram



19.2 Register

(1) Key return mode register (KRM)

The KRM register controls the KRM0 to KRM7 bits using the KR0 to KR7 signals.

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: FFFFF300H

	7	6	5	4	3	2	1	0
KRM	KRM7	KRM6	KRM5	KRM4	KRM3	KRM2	KRM1	KRM0

KRMn	Control of key return mode
0	Does not detect key return signal
1	Detects key return signal

Caution Rewrite the KRM register after once clearing the KRM register to 00H.

Remark For the alternate-function pin settings, see **Table 4-25 Using Port Pin as Alternate Function Pin**.

19.3 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 of another pin is input.

(2) The RXDD1 and KR7 pins must not be used at the same time. To use the RXDD1 pin, do not use the KR7 pin. To use the KR7 pin, do not use the RXDD1 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 interrupts (DI) or masking, then clear the interrupt request flag (KRIC.KRIF bit) to 0, and enable interrupts (EI) or clear the mask.

(4) To use the key interrupt function, be sure to set the port pin to the key return pin and then enable the operation with the KRM register. To switch from the key return pin to the port pin, disable the operation with the KRM register and then set the port pin.

CHAPTER 20 STANDBY FUNCTION

20.1 Overview

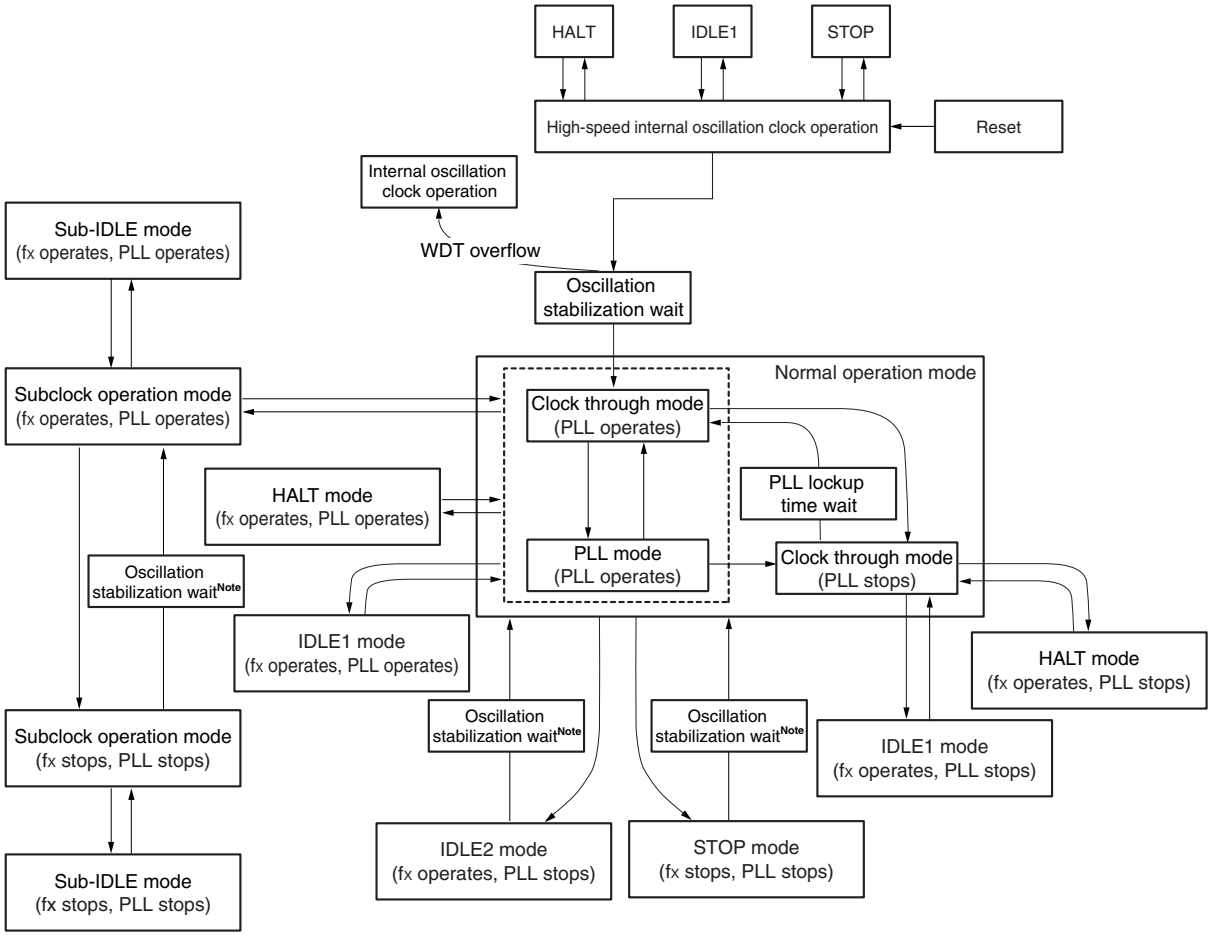
The power consumption of the system can be effectively reduced by using the standby modes in combination and selecting the appropriate mode for the application. The available standby modes are listed in Table 20-1.

Table 20-1. Standby Modes

Mode	Functional Outline
HALT mode	Mode in which only the operating clock of the CPU is stopped
IDLE1 mode	Mode in which all the operations of the internal circuits except the oscillator, PLL[Note], and flash memory are stopped
IDLE2 mode	Mode in which all the internal operations of the chip except the oscillator are stopped
STOP mode	Mode in which all the internal operations of the chip except the subclock oscillator are stopped
Subclock operation mode	Mode in which the subclock is used as the internal system clock
Sub-IDLE mode	Mode in which all the internal operations of the chip except the oscillator are stopped, in the subclock operation mode

Note The PLL holds the previous operating status.

Figure 20-1. Status Transition



Note If a WDT overflow occurs during an oscillation stabilization time, the CPU operates on the internal oscillation clock.

Remark fx: Main clock oscillation frequency

20.2 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 STOP mode. This register is a special register that can be written only by the special sequence combinations (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 mode
1	Standby mode

Note Standby mode set by STP bit: IDLE1, IDLE2, STOP, or sub-IDLE mode

Cautions 1. Before setting the IDLE1, IDLE2, STOP, or sub-IDLE mode, set the PSMR.PSM1 and PSMR.PSM0 bits and then set the STP bit.

2. 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.

4. Be sure to set bits 7, 3, 2, and 0 to "0".

(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
0	1	STOP mode
1	0	IDLE2, sub-IDLE modes
1	1	STOP mode

Cautions 1. Be sure to set bits 7 to 2 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 oscillation of the main clock is enabled, until the oscillation stabilizes after the STOP mode is released, or until the internal flash memory stabilizes after the IDLE2 mode is released is controlled by the OSTS register after reset release.

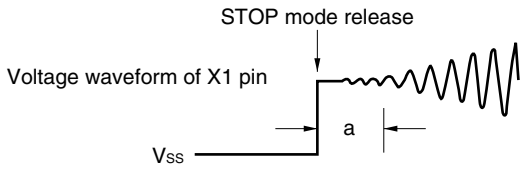
The OSTS register can be read or written 8-bit units.

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	OSTS4	OSTS3	OSTS2	OSTS1	OSTS0

Cautions 1. The wait time following release of the STOP mode does not include the time until the clock oscillation starts ("a" in the figure below) following release of the STOP mode, regardless of whether the STOP mode is released by reset or the occurrence of an interrupt request signal.



2. Be sure to set bits 3 to 5 to "0".

3. The oscillation stabilization time following reset release is 2^{16}/fx (because the initial value of the OSTS register = 06H).

4. The OSTS register cannot be written while the oscillation stabilization time is counted.

Remarks 1. For selecting the oscillation stabilization time, see **Table 20-2**.

2. fx = Main clock oscillation frequency

Table 20-2. Selecting Oscillation Stabilization Time

OSTS4	OSTS3	OSTS2	OSTS1	OSTS0	Selection of Oscillation Stabilization Time				
						f_X = 4 MHz	f_X = 5 MHz	f_X = 12 MHz	f_X = 16 MHz
0	0	0	0	0	$2^{10}/f_X$	256 μs	205 μs	86 μs	64 μs
	0	0	0	1	$2^{11}/f_X$	512 μs	410 μs	171 μs	128 μs
	0	0	1	0	$2^{12}/f_X$	1.03 ms	820 μs	342 μs	256 μs
	0	0	1	1	$2^{13}/f_X$	2.05 ms	1.64 ms	683 μs	512 μs
	0	1	0	0	$2^{14}/f_X$	4.10 ms	3.28 ms	1.37 ms	1.03 ms
	0	1	0	1	$2^{15}/f_X$	8.20 ms	6.54 ms	2.74 ms	2.05 ms
	0	1	1	0	$2^{16}/f_X$ (default value)	16.39 ms	13.11 ms	5.47 ms	4.10 ms
	0	1	1	1	$2^{17}/f_X$	32.77 ms	26.22 ms	10.93 ms	8.20 ms
	1	0	0	0	$2^{18}/f_X$	65.54 ms	52.43 ms	21.85 ms	16.39 ms
	1	0	0	1	$2^{19}/f_X$	131.08 ms	104.86 ms	43.70 ms	32.77 ms
	1	0	1	0	$2^{20}/f_X$	262.15 ms	209.72 ms	87.39 ms	65.54 ms
	1	0	1	1	$2^{21}/f_X$	524.29 ms	349.53 ms	174.77 ms	131.08 ms
1	0	0	0	0	Setting prohibited				
	0	0	0	1	Setting prohibited				
	0	0	1	0	$2^4/f_X$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
	0	0	1	1	$2^5/f_X$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
	0	1	0	0	$2^6/f_X$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
	0	1	0	1	$2^7/f_X$	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
	0	1	1	0	$2^8/f_X$	64 μs	Setting prohibited	Setting prohibited	Setting prohibited
	0	1	1	1	$2^9/f_X$	128 μs	103 μs	Setting prohibited	Setting prohibited
	1	0	0	0	$2^{10}/f_X$	256 μs	205 μs	86 μs	64 μs
	1	0	0	1	$2^{11}/f_X$	512 μs	410 μs	171 μs	128 μs
	1	0	1	0	$2^{12}/f_X$	1.03 ms	820 μs	342 μs	256 μs
	1	0	1	1	$2^{13}/f_X$	2.05 ms	1.64 ms	683 μs	512 μs
Other than above					Setting prohibited				

Cautions 1. The OSTS4 bit is valid only when the IDLE2 mode is released.
If the STOP mode is set when the OSTS4 bit = 1, the oscillation stabilization time after the STOP mode is released is the time set by the OSTS3 to OSTS0 bits (it is assumed that OSTS4 = 0).

2. Make sure that the following stabilization time passes when the IDLE2 mode is released.
- **When PLL operates: 800 μs min. (to secure PLL lockup time)**
- **When PLL stops: 54 μs min. (to secure setup time of flash memory)**
- **When SSCG operates: 1 ms min. (to secure SSCG lockup time)**

3. Make sure that the following stabilization time passes when the STOP mode is released.
- **When PLL operates: 1600 μs min. (to secure PLL lockup time)**
- **When PLL stops: 54 μs min. (to secure setup time of flash memory)**
- **When SSCG operates: 2 ms min. (to secure SSCG lockup time)**

If the oscillation stabilization time of the main clock is higher than above, set so as to secure the oscillation stabilization time of the main clock.

(4) Oscillation stabilization time count status register (OSTC)

The OSTC register is used to check the status of the main clock.

This register is read-only in 8-bit or 1-bit units.

Reset sets this register to 00H.

After reset: 00H R Address: FFFFF6C2H

	7	6	5	4	3	2	1	0
OSTC	0	0	0	0	0	0	0	MSTS

MSTS	Status of main clock
0	Main clock stops or waits for oscillation to stabilize.
1	Main clock oscillation stabilization time has passed.

Cautions 1. The OSTC register does not directly monitor the oscillation status of the main clock but indicates the lapse of the oscillation stabilization time selected by the OSTS register.

2. If oscillation of the main clock is stopped by software (PCC.MCK bit = 1) or is in the STOP mode, the OSTC register is cleared to 00H, but it retains the current value if the main clock oscillation is abnormal and stopped.

20.3 HALT Mode

20.3.1 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 20-3 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 for intermittent operation.

Cautions 1. Insert five or more NOP instructions after the HALT instruction.

2. If the HALT instruction is executed while an unmasked interrupt request signal is being held pending, the status shifts to HALT mode, but the HALT mode is then released immediately by the pending interrupt request.

20.3.2 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 INTP14 pin input), unmasked internal interrupt request signal from a peripheral function operable in the HALT mode, or reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, power-on clear circuit (POC), 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 signal. If the HALT mode is set in an interrupt servicing routine, however, an interrupt request signal that is issued later is serviced as follows.

(a) If an interrupt request signal with a priority lower than that of the interrupt request currently being serviced is issued, the HALT mode is released, but that interrupt request signal is not acknowledged. The interrupt request signal itself is retained.

(b) If an interrupt request signal with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request signal), the HALT mode is released and that interrupt request signal is acknowledged.

Table 20-3. Operation After Releasing HALT Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address.	
Maskable interrupt request signal	Execution branches to the handler address or the next instruction is executed.	The next instruction is executed.

(2) Releasing HALT mode by reset

The same operation as the normal reset operation is performed.

Table 20-4. Operating Status in HALT Mode

Item \ Setting of HALT Mode		Operating Status	
		When Subclock Is Not Used	When Subclock Is Used
Main clock oscillator		Oscillation enabled	
Subclock oscillator		–	Oscillates
Low-speed internal oscillator (f_{RL})		Oscillation enabled	
High-speed internal oscillator (f_{RH})		Oscillation enabled	
PLL		Operable	
SSCG		Operable	
CPU		Stops operation	
Port function		Retains status before HALT mode was set	
External bus interface		See **2.2 Pin States**.	
Timer AA (TAA0 to TAA4)		TAA0, TAA2, TAA4: Operable TAA1, TAA3: Operable when a clock other than f_{XT} is selected as the count clock	Operable
Timer AB (TAB0 to TAB2)		Operable	
Timer M (TMM0)		Operable when a clock other than f_{XT} is selected as the count clock	Operable
Watch timer		Operable when f_X (divided BRG) is selected as the count clock	Operable
Watchdog timer 2		Operable	
A/D converter		Operable	
Serial interface	UARTD0 to UARTD5	Operable	
	CSIB0 to CSIB2	Operable	
	I^2C00	Operable	
DMA		Operable	
Interrupt controller		Operable	
Key interrupt function (KR)		Operable	
Clock monitor		Operable	
Power-on clear circuit		Operable	
Low-voltage detector		Operable	
Regulator		Continues operation	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the HALT mode was set.	

Remark Table 20-4 shows the maximum specifications of the V850ES/HJ3. Some of the functions shown in this table are not supported by the V850ES/HE3, V850ES/HF3, and V850ES/HG3. For details, see **Table 1-1 V850ES/Hx3 Function List**.

20.4 IDLE1 Mode

20.4.1 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 20-6 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 IDLE1 mode is released immediately by the pending interrupt request.

20.4.2 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 INTP14 pin input), unmasked internal interrupt request signal from a peripheral function operable in the IDLE1 mode, or reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, power-on clear circuit (POC), low-voltage detector (LVI), or clock monitor (CLM)).

After the IDLE1 mode has been released, the normal operation mode is restored.

(1) Releasing IDLE1 mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The IDLE1 mode is released by a non-maskable interrupt request signal or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request signal. If the IDLE1 mode is set in an interrupt servicing routine, however, an interrupt request signal that is issued later is processed as follows.

Cautions 1. An interrupt request signal that is disabled by setting the PSC.NMI1M, PSC.NMI0M, and PSC.INTM bits to 1 becomes invalid and IDLE1 mode is not released.

2. If digital noise elimination is selected for the INTP3 pin by using the NFC register and if the sampling clock is selected from f_{XX}/64, f_{XX}/128, f_{XX}/256, f_{XX}/512, and f_{XX}/1024, the IDLE1 mode cannot be released by the interrupt request signal. The IDLE1 mode can be released by selecting the subclock (f_{XT}) as the sampling clock or selecting analog noise elimination for the INTP3 pin. For details, see 18.6.2 (8) Noise elimination control register (NFC).

(a) If an interrupt request signal with a priority lower than that of the interrupt request currently being serviced is issued, the IDLE1 mode is released, but that interrupt request signal is not acknowledged. The interrupt request signal itself is retained.

(b) If an interrupt request signal with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request signal), the IDLE1 mode is released and that interrupt request signal is acknowledged.

Table 20-5. Operation After Releasing IDLE1 Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address.	
Maskable interrupt request signal	Execution branches to the handler address or the next instruction is executed.	The next instruction is executed.

(2) Releasing IDLE1 mode by reset

The same operation as the normal reset operation is performed.

Table 20-6. Operating Status in IDLE1 Mode

Item \ Setting of IDLE1 Mode		Operating Status	
		When Subclock Is Not Used	When Subclock Is Used
Main clock oscillator		Oscillation enabled	
Subclock oscillator		–	Oscillates
Low-speed internal oscillator (f_{RL})		Oscillation enabled	
High-speed internal oscillator (f_{RH})		Oscillation enabled	
PLL		Operable	
SSCG		Operable	
CPU		Stops operation	
Port function		Retains status before IDLE1 mode was set	
External bus interface		See **2.2 Pin States**.	
Timer AA (TAA0 to TAA4)		Stops operation	
Timer AB (TAB0 to TAB2)		Stops operation	
Timer M (TMM0)		Operable when INTWT or $f_{RL}/8$ is selected as the count clock	Operable when INTWT, $f_{RL}/8$, or f_{XT} is selected as the count clock
Watch timer		Operable when f_X (divided BRG) is selected as the count clock	Operable
Watchdog timer 2		Operable	
A/D converter		Stops operation	
Serial interface	UARTD0 to UARTD5	Stops operation (but UARTD0 is operable when the ASCKD0 input clock is selected)	
	CSIB0 to CSIB2	Operable when the $\overline{SCKBn}$ input clock is selected as the count clock	
	I^2C00	Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation (but IDLE1 mode release is possible)	
Key interrupt function (KR)		Operable	
Clock monitor		Operable	
Power-on clear circuit		Operable	
Low-voltage detector		Operable	
Regulator		Continues operation	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the IDLE1 mode was set.	

Remark Table 20-6 shows the maximum specifications of the V850ES/HJ3. Some of the functions shown in this table are not supported by the V850ES/HE3, V850ES/HF3, and V850ES/HG3. For details, see **Table 1-1 V850ES/Hx3 Function List**.

20.5 IDLE2 Mode

20.5.1 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 20-8 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 IDLE2 mode is released immediately by the pending interrupt request.

20.5.2 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 INTP14 pin input), unmasked internal interrupt request signal from the peripheral functions operable in the IDLE2 mode, or reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, power-on clear circuit (POC), 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 IDLE2 mode is released by a non-maskable interrupt request signal or an unmasked maskable interrupt request signal, regardless of the priority of the interrupt request signal. If the IDLE2 mode is set in an interrupt servicing routine, however, an interrupt request signal that is issued later is processed as follows.

Cautions 1. The interrupt request signal that is disabled by setting the PSC.NMI1M, PSC.NMI0M, and PSC.INTM bits to 1 becomes invalid and IDLE2 mode is not released.

2. If digital noise elimination is selected for the INTP3 pin by using the NFC register and if the sampling clock is selected from f_{XX}/64, f_{XX}/128, f_{XX}/256, f_{XX}/512, and f_{XX}/1024, the IDLE2 mode cannot be released by the interrupt request signal. The IDLE2 mode can be released by selecting the subclock (f_{XT}) as the sampling clock or by selecting analog noise elimination. For details, see 18.6.2 (8) Noise elimination control register (NFC).

(a) If an interrupt request signal with a priority lower than that of the interrupt request currently being serviced is issued, the IDLE2 mode is released, but that interrupt request signal is not acknowledged. The interrupt request signal itself is retained.

(b) If an interrupt request signal with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request signal), the IDLE2 mode is released and that interrupt request signal is acknowledged.

Table 20-7. Operation After Releasing IDLE2 Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address after securing the prescribed setup time.	
Maskable interrupt request signal	Execution branches to the handler address or the next instruction is executed after securing the prescribed setup time.	The next instruction is executed after securing the prescribed setup time.

(2) Releasing IDLE2 mode by reset

The same operation as the normal reset operation is performed.

Table 20-8. Operating Status in IDLE2 Mode

Item \ Setting of IDLE2 Mode		Operating Status	
		When Subclock Is Not Used	When Subclock Is Used
Main clock oscillator		Oscillation enabled	
Subclock oscillator		–	Oscillates
Low-speed internal oscillator (f_{RL})		Oscillation enabled	
High-speed internal oscillator (f_{RH})		Oscillation enabled	
PLL		Stops operation	
SSCG		Stops operation	
CPU		Stops operation	
Port function		Retains status before IDLE2 mode was set	
External bus interface		See **2.2 Pin States**.	
Timer AA (TAA0 to TAA4)		Stops operation	
Timer AB (TAB0 to TAB2)		Stops operation	
Timer M (TMM0)		Operable when INTWT or f_{RL}/8 is selected as the count clock	Operable when INTWT, f_{RL}/8, or f_{XT} is selected as the count clock
Watch timer		Operable when f_X (divided BRG) is selected as the count clock	Operable
Watchdog timer 2		Operable	
A/D converter		Stops operation	
Serial interface	UARTD0 to UARTD5	Stops operation (but UARTD0 is operable when the ASCKD0 input clock is selected)	
	CSIB0 to CSIB2	Operable when the $\overline{SCKBn}$ input clock is selected as the count clock	
	I^2C00	Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation (but IDLE2 mode release is possible)	
Key interrupt function (KR)		Operable	
Clock monitor		Operable	
Power-on clear circuit		Operable	
Low-voltage detector		Operable	
Regulator		Continues operation	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the IDLE2 mode was set.	

Remark Table 20-8 shows the maximum specifications of the V850ES/HJ3. Some of the functions shown in this table are not supported by the V850ES/HE3, V850ES/HF3, and V850ES/HG3. For details, see **Table 1-1 V850ES/Hx3 Function List**.

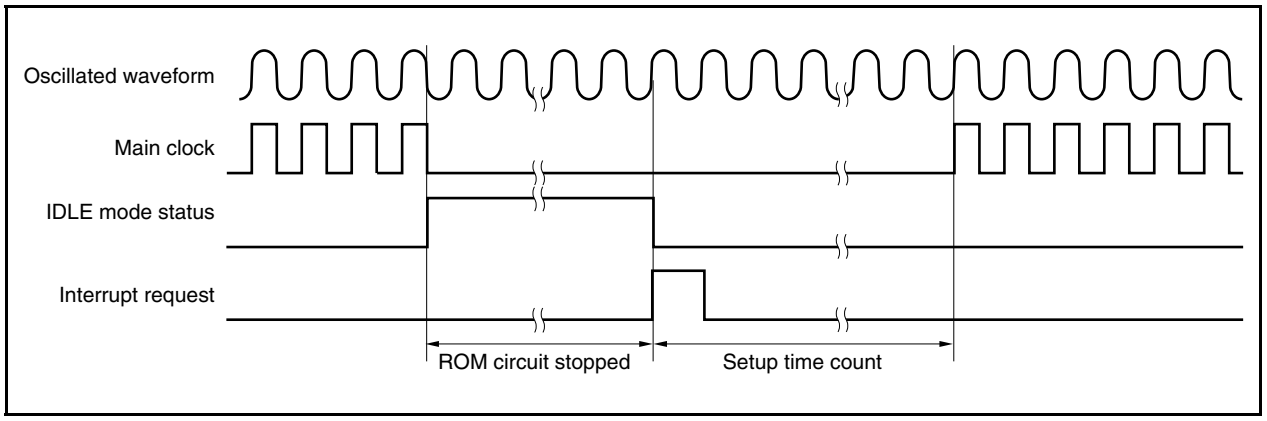
20.5.3 Securing setup time when releasing IDLE2 mode

Secure the setup time for the flash memory by the OSTS register after releasing the IDLE2 mode because the operation of the blocks other than the main clock oscillator stops after the IDLE2 mode is set. Also, secure the lockup time by the OSTS register when the PLL and SSCG are used.

(1) Releasing IDLE2 mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

Secure the specified setup time and lockup 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 is the initial value of the OSTS register, 2^{16}/fx.

20.6 STOP Mode

20.6.1 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.

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. The on-chip peripheral functions that operate with the clock oscillated by the subclock oscillator or an external clock continue operating.

Table 20-10 shows the operating status in the 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, and external clock are not used, the power consumption can be minimized with only leakage current flowing.

Cautions 1. Insert five or more NOP instructions after the instruction that stores data in the PSC register to set the STOP mode.

2. If the STOP mode is set while an unmasked interrupt request signal is being held pending, the STOP mode is released immediately by the pending interrupt request.

20.6.2 Releasing STOP mode

The STOP mode is released by a non-maskable interrupt request signal (NMI pin input, INTWDT2 signal), unmasked external interrupt request signal (INTP0 to INTP14 pin input), unmasked internal interrupt request signal from the peripheral functions operable in the STOP mode, or reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, power-on clear circuit (POC), or low-voltage detector (LVI)).

After the STOP mode has been released, the normal operation mode is restored after the oscillation stabilization time has been secured.

Cautions 1. The interrupt request that is disabled by setting the PSC.NMI1M, PSC.NMI0M, and PSC.INTM bits to 1 becomes invalid and STOP mode is not released.

2. If digital noise elimination is selected for the INTP3 pin by using the NFC register and if the sampling clock is selected from f_{XX}/64, f_{XX}/128, f_{XX}/256, f_{XX}/512, and f_{XX}/1024, the STOP mode cannot be released by the interrupt request signal. The STOP mode can be released by selecting the subclock (f_{XT}) as the sampling clock or selecting analog noise elimination. For details, see 18.6.2 (8) Noise elimination control register (NFC).

(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 signal. If the STOP mode is set in an interrupt servicing routine, however, an interrupt request signal that is issued later is serviced as follows.

(a) If an interrupt request signal with a priority lower than that of the interrupt request currently being serviced is issued, the STOP mode is released, but that interrupt request signal is not acknowledged. The interrupt request signal itself is retained.

(b) If an interrupt request signal with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request signal), the STOP mode is released and that interrupt request signal is acknowledged.

Table 20-9. Operation After Releasing STOP Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address after securing the oscillation stabilization time.	
Maskable interrupt request signal	Execution branches to the handler address or the next instruction is executed after securing the oscillation stabilization time.	The next instruction is executed after securing the oscillation stabilization time.

(2) Releasing STOP mode by reset

The same operation as the normal reset operation is performed.

Table 20-10. Operating Status in STOP Mode

Setting of STOP Mode / Item		Operating Status	
		When Subclock Is Not Used	When Subclock Is Used
Main clock oscillator		Stops oscillation	
Subclock oscillator		–	Oscillates
Low-speed internal oscillator (f_{RL})		Oscillation enabled	
High-speed internal oscillator (f_{RH})		Stops oscillation	
PLL		Stops operation	
SSCG		Stops operation	
CPU		Stops operation	
Port function		Retains status before STOP mode was set	
External bus interface		See **2.2 Pin States**.	
Timer AA (TAA0 to TAA4)		Stops operation	
Timer AB (TAB0 to TAB2)		Stops operation	
Timer M (TMM0)		Operable when f_{RL}/8 is selected as the count clock	Operable when INTWT, f_{RL}/8, or f_{XT} is selected as the count clock
Watch timer		Stops operation	Operable when f_{XT} is selected as the count clock
Watchdog timer 2		Operable when f_{RL} is selected as the count clock	
A/D converter		Stops operation	
Serial interface	UARTD0 to UARTD5	Stops operation (but UARTD0 is operable when the ASCKD0 input clock is selected)	
	CSIB0 to CSIB2	Operable when the $\overline{\text{SCKBn}}$ input clock is selected as the count clock	
	I^2C00	Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation (but STOP mode release is possible)	
Key interrupt function (KR)		Operable	
Clock monitor		Stops operation	
Power-on clear circuit		Operable	
Low-voltage detector		Operable	
Regulator		Continues operation	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the STOP mode was set.	

Remark Table 20-10 shows the maximum specifications of the V850ES/HJ3. Some of the functions shown in this table are not supported by the V850ES/HE3, V850ES/HF3, and V850ES/HG3. For details, see **Table 1-1 V850ES/Hx3 Function List**.

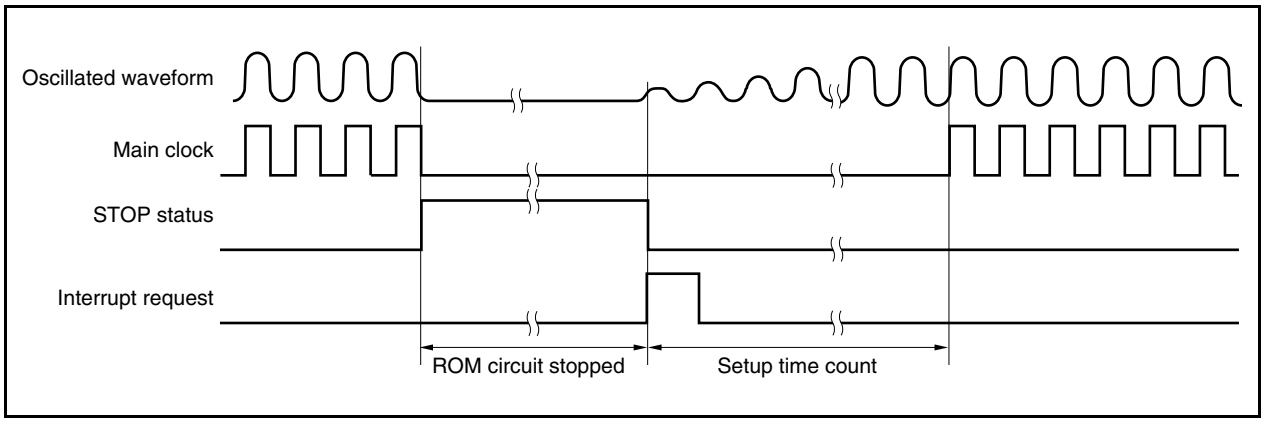
20.6.3 Securing oscillation stabilization time when releasing STOP mode

Secure the oscillation stabilization time for the main clock oscillator by the OSTS register after releasing the STOP mode because the operation of the main clock oscillator stops after STOP mode is set. Also, secure the setup time for the flash memory and the lockup time for the PLL and SSCG by the OSTS register.

(1) Releasing STOP mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

Secure the setup time, lockup time, and 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 is the initial value of the OSTS register, 2^{16}/fx.

20.7 Subclock Operation Mode

20.7.1 Setting and operation status

The subclock operation mode is set by setting the PCC.CK3 bit to 1 in the normal operation mode.

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, the 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, the power consumption can be further reduced to the level of the STOP mode by stopping the operation of the main clock oscillator.

Table 20-11 shows the operating status in subclock operation mode.

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 6.3 (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.
Internal system clock (f_{CLK}) > Subclock (f_{XT}) × 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

20.7.2 Releasing subclock operation mode

The subclock operation mode is released by a reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, power-on clear circuit (POC), low-voltage detector (LVI), or clock monitor (CLM)) when the CK3 bit is cleared 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 clear 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 6.3 (1) Processor clock control register (PCC).

Table 20-11. Operating Status in Subclock Operation Mode

Item \ Setting of Subclock Operation Mode		Operating Status	
		When Main Clock Is Oscillating	When Main Clock Is Stopped
Subclock oscillator		Oscillates	
Low-speed internal oscillator (f_{RL})		Oscillation enabled	
High-speed internal oscillator (f_{RH})		Oscillation enabled	
PLL		Operable	Stops operation[Note]
SSCG		Operable	Stops operation[Note]
CPU		Operable	
Port function		Settable	
External bus interface		See **2.2 Pin States**.	
Timer AA (TAA0 to TAA4)		Operable	Stops operation
Timer AB (TAB0 to TAB2)		Operable	Stops operation
Timer M (TMM0)		Operable	Operable when INTWT, $f_{RL}/8$, or f_{XT} is selected as the count clock
Watch timer		Operable	Operable when f_{XT} is selected as the count clock
Watchdog timer 2		Operable	Operable when f_{RL} is selected as the count clock
A/D converter		Operable	Stops operation
Serial interface	UARTD0 to UARTD5	Operable	Stops operation (but UARTD0 is operable when the ASCKD0 input clock is selected)
	CSIB0 to CSIB2	Operable	Operable when the $\overline{SCKBn}$ input clock is selected as the count clock
	I^2C00	Operable	Stops operation
DMA		Operable	
Interrupt controller		Operable	
Key interrupt function (KR)		Operable	
Clock monitor		Operable	Stops operation
Power-on clear circuit		Operable	
Low-voltage detector		Operable	
Regulator		Continues operation	
Internal data		Settable	

Note Be sure to stop the PLL and SSCG (PLLON bit = 0, SSCGON bit = 0) before stopping the main clock oscillator.

Caution When the CPU is operating on the subclock and main clock oscillation is stopped, accessing a register in which a wait occurs is disabled. If a wait is generated, it can be released only by reset (see 3.4.8 (2)).

Remark Table 20-11 shows the maximum specifications of the V850ES/HJ3. Some of the functions shown in this table are not supported by the V850ES/HE3, V850ES/HF3, and V850ES/HG3. For details, see **Table 1-1 V850ES/Hx3 Function List**.

20.8 Sub-IDLE Mode

20.8.1 Setting and operation status

The sub-IDLE mode is set by setting the PSMR.PSM1 and PSMR.PSM0 bits to 10 and setting the PSC.STP bit to 1 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.

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.

Table 20-13 shows the operating status in the sub-IDLE mode.

Cautions 1. Following the store instruction to set the PSC register to the sub-IDLE mode, insert five or more NOP instructions.

2. If the sub-IDLE mode is set while an unmasked interrupt request signal is being held pending, the sub-IDLE mode is then released immediately by the pending interrupt request.

20.8.2 Releasing sub-IDLE mode

The sub-IDLE mode is released by a non-maskable interrupt request signal (NMI pin input, INTWDT2 signal), unmasked external interrupt request signal (INTP0 to INTP14 pin input), unmasked internal interrupt request signal from the peripheral functions operable in the sub-IDLE mode, or reset signal (reset by $\overline{\text{RESET}}$ pin input, WDT2RES signal, power-on clear circuit (POC), 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.

When the sub-IDLE mode is released by an interrupt request signal, the subclock operation mode is set.

(1) Releasing sub-IDLE mode by non-maskable interrupt request signal or unmasked maskable interrupt request signal

The 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 signal.

If the sub-IDLE mode is set in an interrupt servicing routine, however, an interrupt request signal that is issued later is serviced as follows.

Cautions 1. The interrupt request signal that is disabled by setting the PSC.NMI1M, PSC.NMI0M, and PSC.INTM bits to 1 becomes invalid and sub-IDLE mode is not released.

2. When the sub-IDLE mode is released, 12 cycles of the subclock (about 366 μs) elapse from when the interrupt request signal that releases the sub-IDLE mode is generated to when the mode is released.

3. If digital noise elimination is selected for the INTP3 pin by using the NFC register and if the sampling clock is selected from f_{XX}/64, f_{XX}/128, f_{XX}/256, f_{XX}/512, and f_{XX}/1024, the sub-IDLE mode cannot be released by the interrupt request signal. The sub-IDLE mode can be released by selecting the subclock (f_{XT}) as the sampling clock or selecting analog noise elimination. For details, see 18.6.2 (8) Noise elimination control register (NFC).

(a) If an interrupt request signal with a priority lower than that of the interrupt request currently being serviced is issued, the sub-IDLE mode is released, but that interrupt request signal is not acknowledged. The interrupt request signal itself is retained.

(b) If an interrupt request signal with a priority higher than that of the interrupt request currently being serviced is issued (including a non-maskable interrupt request signal), the sub-IDLE mode is released and that interrupt request signal is acknowledged.

Table 20-12. Operation After Releasing Sub-IDLE Mode by Interrupt Request Signal

Release Source	Interrupt Enabled (EI) Status	Interrupt Disabled (DI) Status
Non-maskable interrupt request signal	Execution branches to the handler address.	
Maskable interrupt request signal	Execution branches to the handler address or the next instruction is executed.	The next instruction is executed.

(2) Releasing sub-IDLE mode by reset

The same operation as the normal reset operation is performed.

Table 20-13. Operating Status in Sub-IDLE Mode

Setting of Sub-IDLE Mode / Item		Operating Status	
		When Main Clock Is Oscillating	When Main Clock Is Stopped
Subclock oscillator		Oscillates	
Low-speed internal oscillator (f_{RL})		Oscillation enabled	
High-speed internal oscillator (f_{RH})		Oscillation enabled	
PLL		Operable	Stops operation[Note]
SSCG		Operable	Stops operation[Note]
CPU		Stops operation	
Port function		Retains status before sub-IDLE mode was set	
External bus interface		See **2.2 Pin States**	
Timer AA (TAA0 to TAA4)		Stops operation	
Timer AB (TAB0 to TAB2)		Stops operation	
Timer M (TMM0)		Operable when INTWT, $f_{RL}/8$, or f_{XT} is selected as the count clock	
Watch timer		Operable	Operable when f_{XT} is selected as the count clock
Watchdog timer 2		Operable	Operable when f_{RL} is selected as the count clock
A/D converter		Stops operation	
Serial interface	UARTD0 to UARTD5	Stops operation (but UARTD0 is operable when the ASCKD0 input clock is selected)	
	CSIB0 to CSIB2	Operable when the $\overline{\text{SCKBn}}$ input clock is selected as the count clock	
	I^2C00	Stops operation	
DMA		Stops operation	
Interrupt controller		Stops operation (but sub-IDLE mode release is possible)	
Key interrupt function (KR)		Operable	
Clock monitor		Operable	Stops operation
Power-on clear circuit		Operable	
Low-voltage detector		Operable	
Regulator		Continues operation	
Internal data		The CPU registers, statuses, data, and all other internal data such as the contents of the internal RAM are retained as they were before the sub-IDLE mode was set.	

Note Be sure to stop the PLL and SSCG (PLLON bit = 0, SSCGON bit = 0) before stopping the main clock oscillator.

Remark Table 20-13 shows the maximum specifications of the V850ES/HJ3. Some of the functions shown in this table are not supported by the V850ES/HE3, V850ES/HF3, and V850ES/HG3. For details, see **Table 1-1 V850ES/Hx3 Function List**.

CHAPTER 21 RESET FUNCTIONS

21.1 Overview

The following reset functions are available.

(1) Four kinds of reset sources

- External reset input via the $\overline{\text{RESET}}$ pin
- Reset via the watchdog timer 2 (WDT2) overflow (WDT2RES)
- System reset via the comparison of the low-voltage detector (LVI) supply voltage and detected voltage
- System reset via the detecting clock monitor (CLM) oscillation stop
- System reset via the power-on clear circuit

After a reset is released, the source of the reset can be confirmed with the reset source flag register (RESF).

(2) Emergency operation mode

If the WDT2 overflows during the main clock oscillation stabilization time inserted after reset, a main clock oscillation anomaly is judged and the CPU starts operating on the internal oscillation clock.

Caution When the CPU is being operated with the low-speed internal oscillation clock, access to the register in which a wait state is generated is prohibited. For the register in which a wait state is generated, see 3.4.8 (2) Accessing specific on-chip peripheral I/O registers.

21.2 Registers to Check Reset Source

The V850ES/Hx3 has four kinds of reset sources. After a reset has been released, the source of the reset that occurred can be checked with the reset source flag register (RESF).

(1) Reset source flag register (RESF)

The RESF register is a special register that can be written only by a combination of specific sequences (see **3.4.7 Special registers**).

The RESF register indicates the source from which a reset signal is generated.

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

$\overline{\text{RESET}}$ pin input or POC reset sets this register to 00H. The default value differs if the source of reset is other than the $\overline{\text{RESET}}$ pin signal.

After reset: 00H[Note] R/W Address: FFFFF888H

	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), low-voltage detector (LVI), or clock monitor (CLM), the reset flags of this register (WDT2RF bit, CLMRF bit, and LVIRF bit) are set. However, other sources are retained.

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

2. 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.

21.3 Operation

21.3.1 Reset operation via $\overline{\text{RESET}}$ pin

When a low level is input to the $\overline{\text{RESET}}$ pin, the system is reset, and each hardware unit is initialized.

When the level of the $\overline{\text{RESET}}$ pin is changed from low to high, the reset status is released. The CPU starts program execution after reset release and securing the setup time of the high-speed internal oscillator.

Table 21-1. Hardware Status on $\overline{\text{RESET}}$ Pin Input

Item	During Reset	After Reset
Main clock oscillator (f_X)	Stops oscillation	Stops oscillation (oscillation can be set starting by software after securing oscillation stabilization time of f_{RH}).
Subclock oscillator (f_{XT})	Continues oscillation	
Low-speed internal oscillator (f_{RL})	Stops oscillation	Starts oscillation
High-speed internal oscillator (f_{RH})	Stops oscillation	Starts oscillation
PLL	Stops oscillation	Stops operation (operation can be set starting by software after securing oscillation stabilization time)
SSCG	Stops oscillation	Stops operation (operation can be set starting by software after securing oscillation stabilization time)
Peripheral clock (f_X to f_X/1,024)	Stops operation	Starts operation after securing oscillation stabilization time
Internal system clock (f_{CLK}), CPU clock (f_{CPU})	Stops operation	Starts operation after securing oscillation stabilization time
CPU	Initialized	Program execution starts after securing oscillation stabilization time
Watchdog timer 2	Stops operation (initialized to 0)	Starts operation
Internal RAM	Undefined if power-on reset or writing data to RAM (by CPU or DMAC) and reset input conflict (data is damaged). Otherwise value immediately after reset input is retained[Note 1].	
I/O lines (ports/alternate-function pins)	High impedance[Note 2]	
On-chip peripheral I/O registers	Initialized to specified status, OCDM register is set (01H).	
Other on-chip peripheral functions	Stops operation	Operation can be started after securing oscillation stabilization time

Notes 1. The firmware of the V850ES/Hx3 uses a part of the internal RAM after the internal system reset status has been released. Therefore, the contents of some internal RAM areas are not retained after power-on reset. For details, see **21.4 Operation After Reset Release**.

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

- P53/KR3/TIAB00/TOAB00/TOAB0B2/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 after a reset release before the OCDM.OCDM0 bit is cleared, the on-chip debug mode is entered. For details, see CHAPTER 4 PORT FUNCTIONS.

Figure 21-1. Timing of Reset Operation by $\overline{\text{RESET}}$ Pin Input

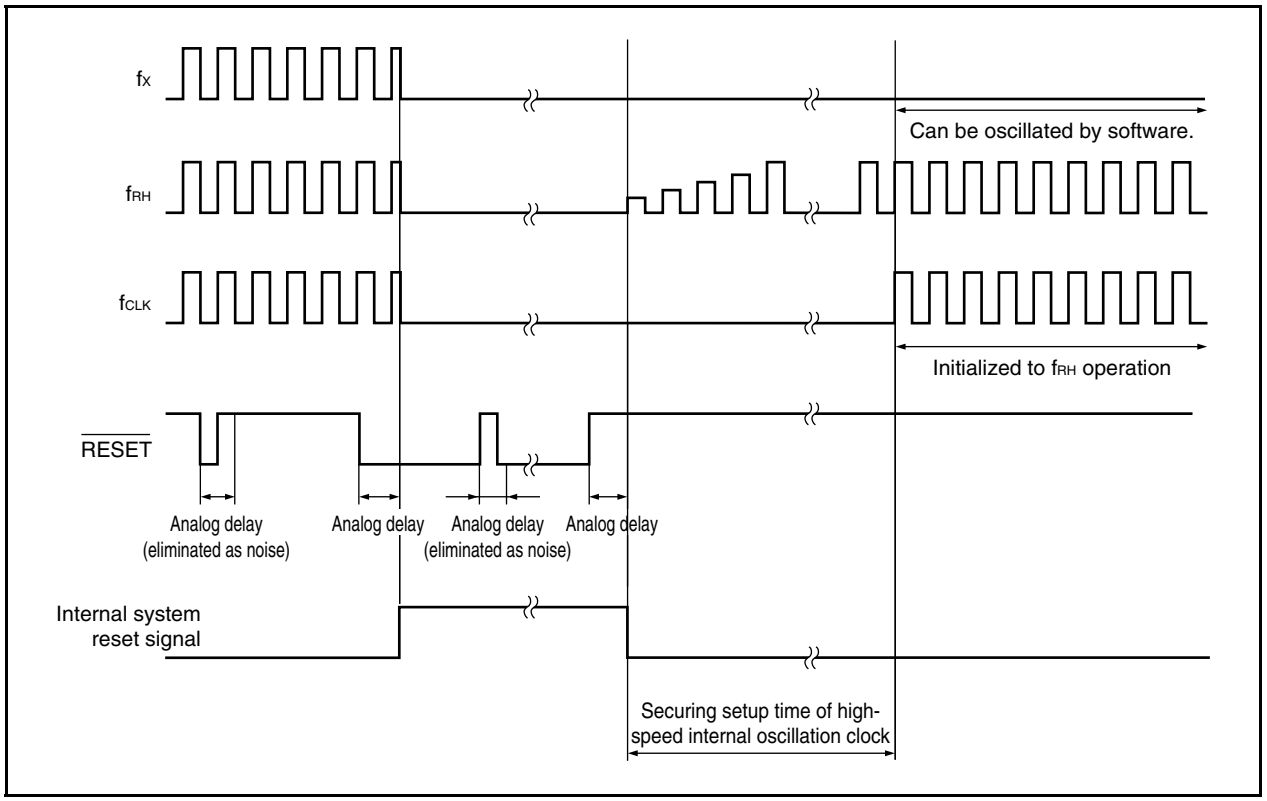
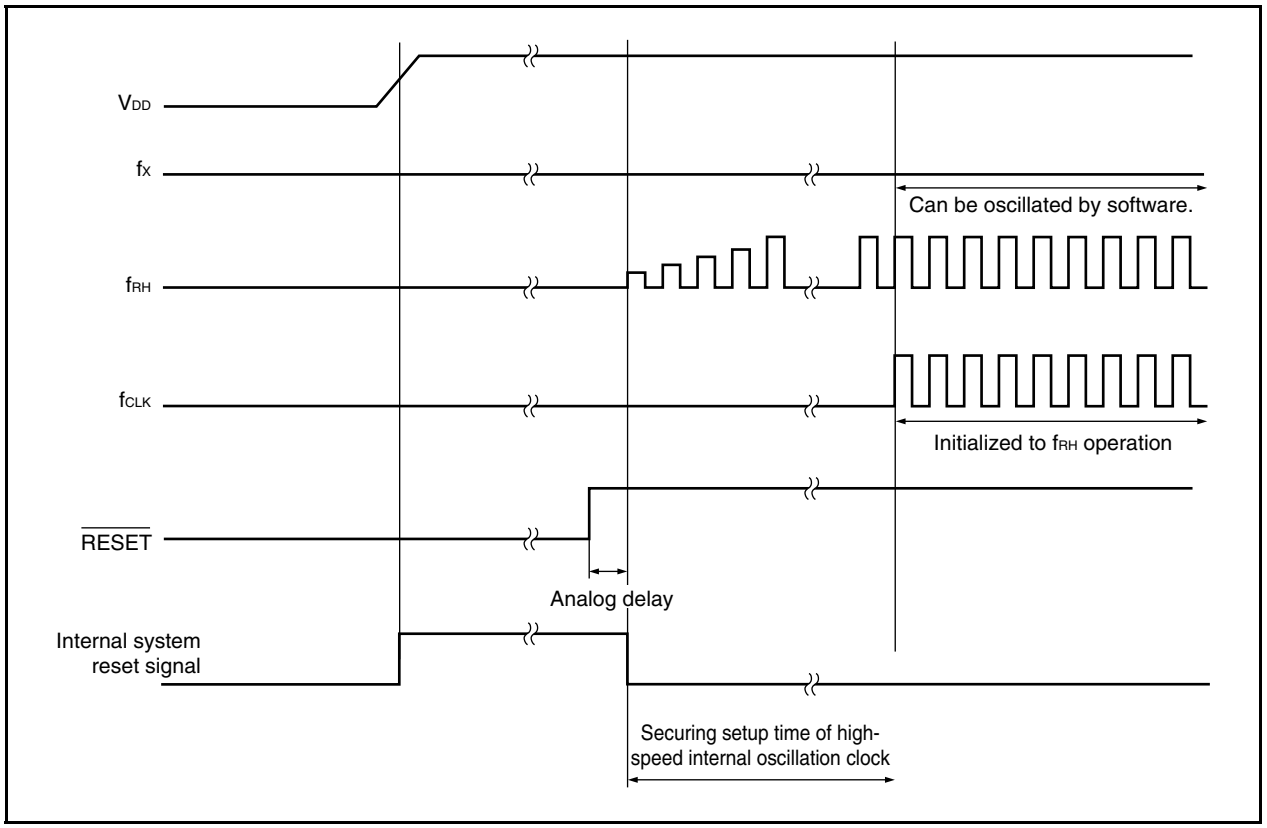


Figure 21-2. Timing of Power-on Reset Operation



21.3.2 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 the reset status is then automatically released.

Table 21-2. Hardware Status During Watchdog Timer 2 Reset Operation

Item	During Reset	After Reset
Main clock oscillator (f_X)	Stops oscillation	Stops oscillation (oscillation can be set starting by software after securing oscillation stabilization time of f_{RH}).
Subclock oscillator (f_{XT})	Continues oscillation	
Low-speed internal oscillator (f_{RL})	Stops oscillation	Starts oscillation
High-speed internal oscillator (f_{RH})	Stops oscillation	Starts oscillation
PLL	Stops oscillation	Stops operation (operation can be set starting by software after securing oscillation stabilization time)
SSCG	Stops oscillation	Stops operation (operation can be set starting by software after securing oscillation stabilization time)
Peripheral clock (f_{XX} to f_{XX}/1,024)	Stops operation	Starts operation after securing oscillation stabilization time
Internal system clock (f_{XX}), CPU clock (f_{CPU})	Stops operation	Starts operation after securing oscillation stabilization time
CPU	Initialized	Program execution after securing oscillation stabilization time
Watchdog timer 2	Stops operation (initialized to 0)	Starts operation
Internal RAM	Undefined if power-on reset or writing data to RAM (by CPU or DMAC) and reset input conflict (data is damaged). Otherwise value immediately after reset input is retained[Note].	
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	Stops operation	Operation can be started after securing oscillation stabilization time.

Note The firmware of the V850ES/Hx3 uses a part of the internal RAM after the internal system reset status has been released. Therefore, the contents of some internal RAM areas are not retained after power-on reset. For details, see **21.4 Operation After Reset Release**.

21.3.3 Reset operation by power-on clear circuit

The supply voltage and detection voltage are compared when the power-on clear operation is enabled. If the supply voltage drops below the detection voltage (including when power is applied), the system is reset and each hardware unit is initialized to the default status.

The reset status lasts since the voltage drop has been detected until the supply voltage rises above the detection voltage, and then is automatically cleared. After reset is released, the setup time of the high-speed internal oscillator (f_{RH}) elapses, and then the CPU starts program execution. For details, see **Figure 23-2 Timing of Reset Signal Generation by Power-on-Clear Circuit**.

21.3.4 Reset operation by low-voltage detector

When LVI operation is enabled and when the LVIM.LVIMD bit is set to “1”, the supply voltage and detection voltage are compared. If the supply voltage drops below the detection voltage, the system is reset and each hardware unit is initialized to the default status.

The reset status lasts from detection of the voltage drop until the supply voltage rises above the detection voltage, and then is automatically cleared. After reset is released, the setup time of the high-speed internal oscillator (f_{RH}) elapses, and then the CPU starts program execution.

For details, see **CHAPTER 24 LOW-VOLTAGE DETECTOR**.

21.3.5 Reset operation by clock monitor

When the clock monitor operation is enabled, the main clock is monitored by using the sampling clock (low-speed internal oscillator). If stoppage of the main clock is detected, the system is reset and each hardware unit is initialized to the default status.

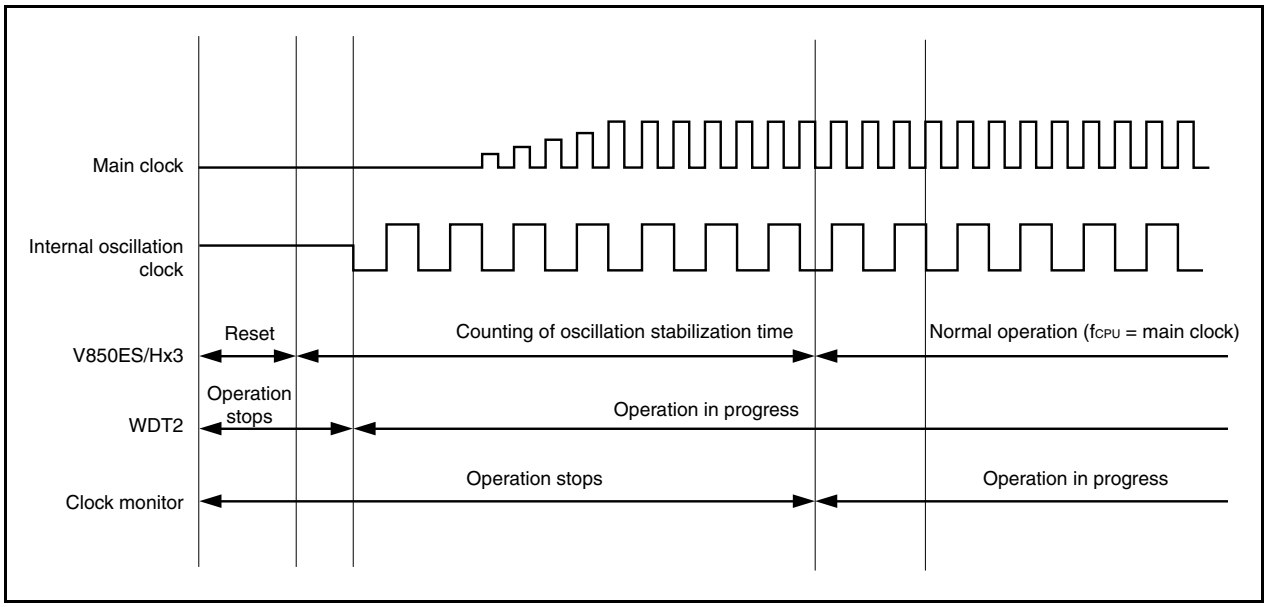
For details, see **CHAPTER 22 CLOCK MONITOR**.

21.4 Operation After Reset Release

After the reset is released, the main clock starts oscillation and oscillation stabilization time (OSTS register initial value: 2^{16}/fx) is secured, and the CPU starts program execution.

WDT2 immediately begins to operate after a reset has been released using the internal oscillation clock as a source clock.

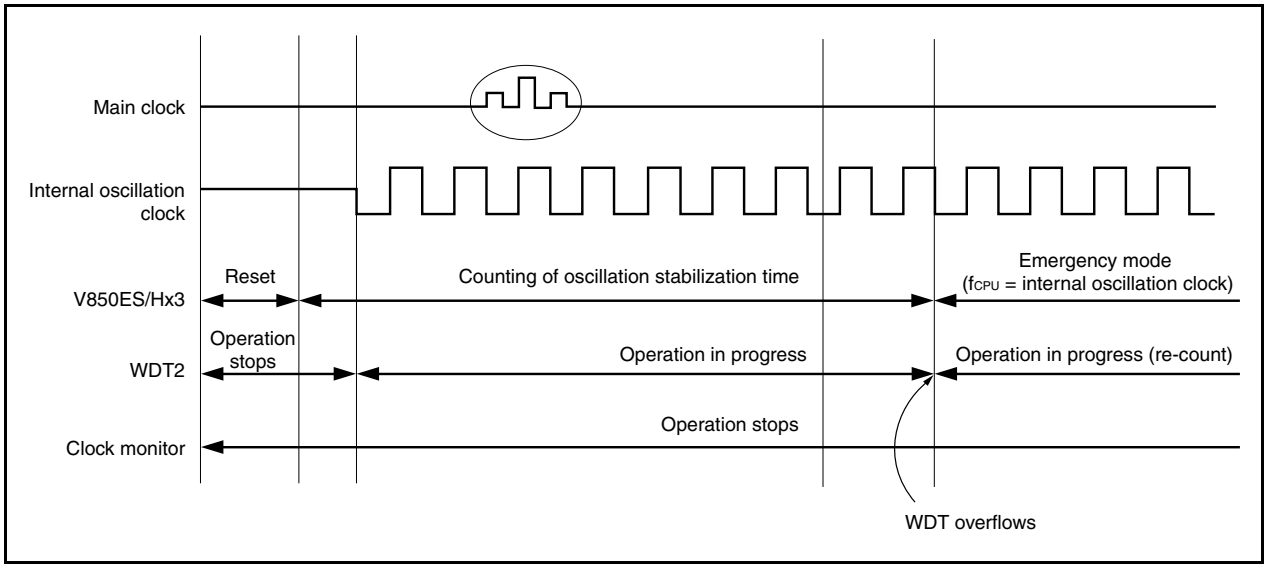
Figure 21-3. Operation After Reset Release



(1) Emergent operation mode

If an anomaly occurs in the main clock before oscillation stabilization time is secured, the WDT2 overflows before executing the CPU program. At this time, the CPU starts program execution by using the internal oscillation clock as the source clock.

Figure 21-4. Operation After Reset Release



The CPU operation clock states can be checked with the CPU operation clock status register (CCLS).

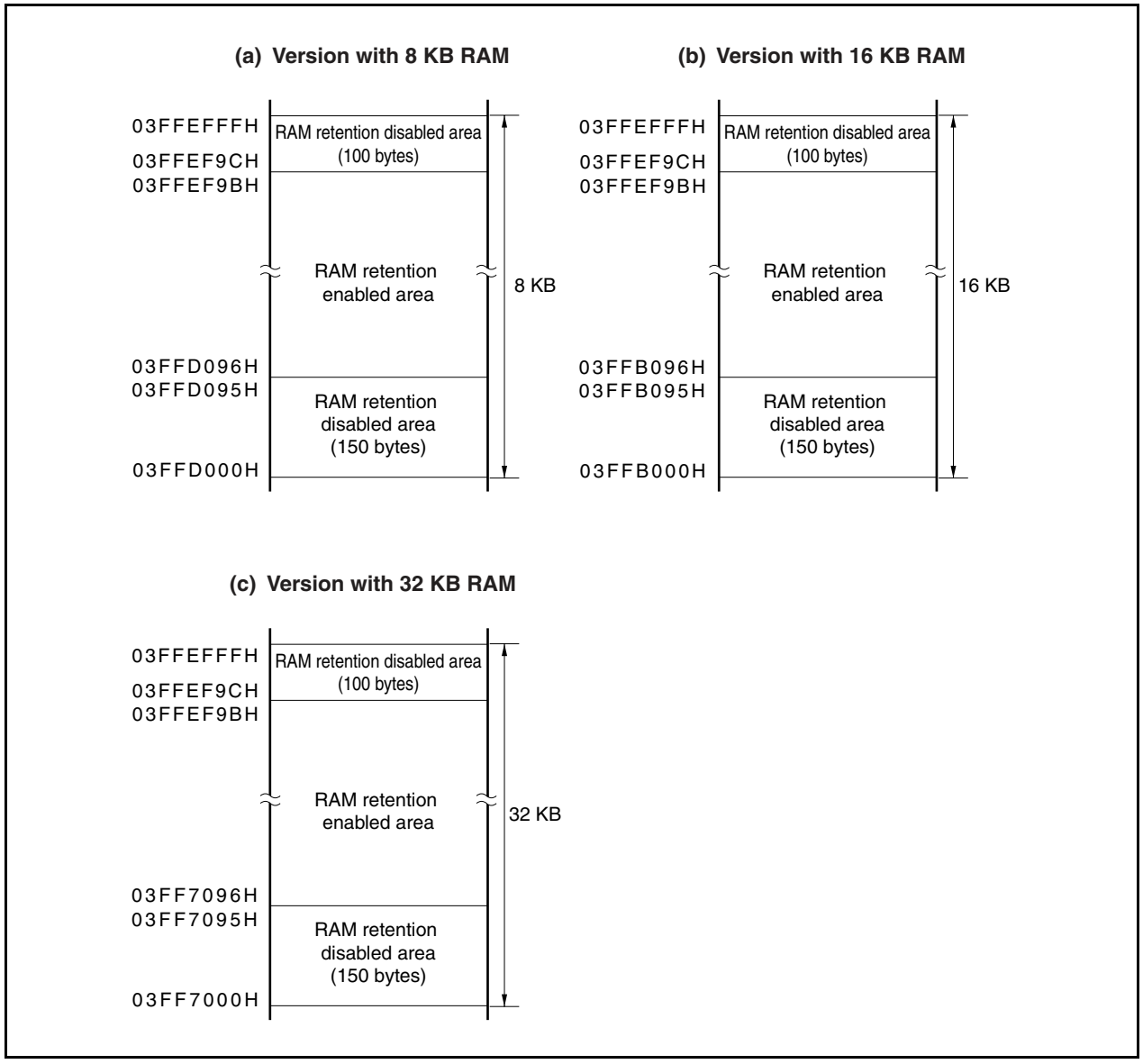
(2) Firmware operation

In the V850ES/Hx3, after a reset is released, the on-chip firmware operates before starting the user program to support the boot switch function.

Since the firmware uses a portion of the internal RAM, the contents of the following RAM areas are not retained through a reset even in power on status.

- Version with 8 KB RAM: 03FFD000H to 03FFD095H, 03FFEF9CH to 03FFEFFFH
- Version with 16 KB RAM: 03FFB000H to 03FFB095H, 03FFEF9CH to 03FFEFFFH
- Version with 32 KB RAM: 03FF7000H to 03FF7095H, 03FFEF9CH to 03FFEFFFH

Figure 21-5. RAM Retention Enabled Area



CHAPTER 22 CLOCK MONITOR

22.1 Functions

The clock monitor samples the main clock by using the low-speed internal oscillation 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 reset.

When a reset by the clock monitor occurs, the RESF.CLMRF bit is set. For details on the RESF register, see **21.2 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 sampling clock (low-speed internal oscillation clock) is stopped
- When the CPU operates with the high-speed internal oscillation clock (MCM.MCS bit = 0)
- When the CPU operates with the low-speed internal oscillation clock (CCLS.CCLSF bit = 1)

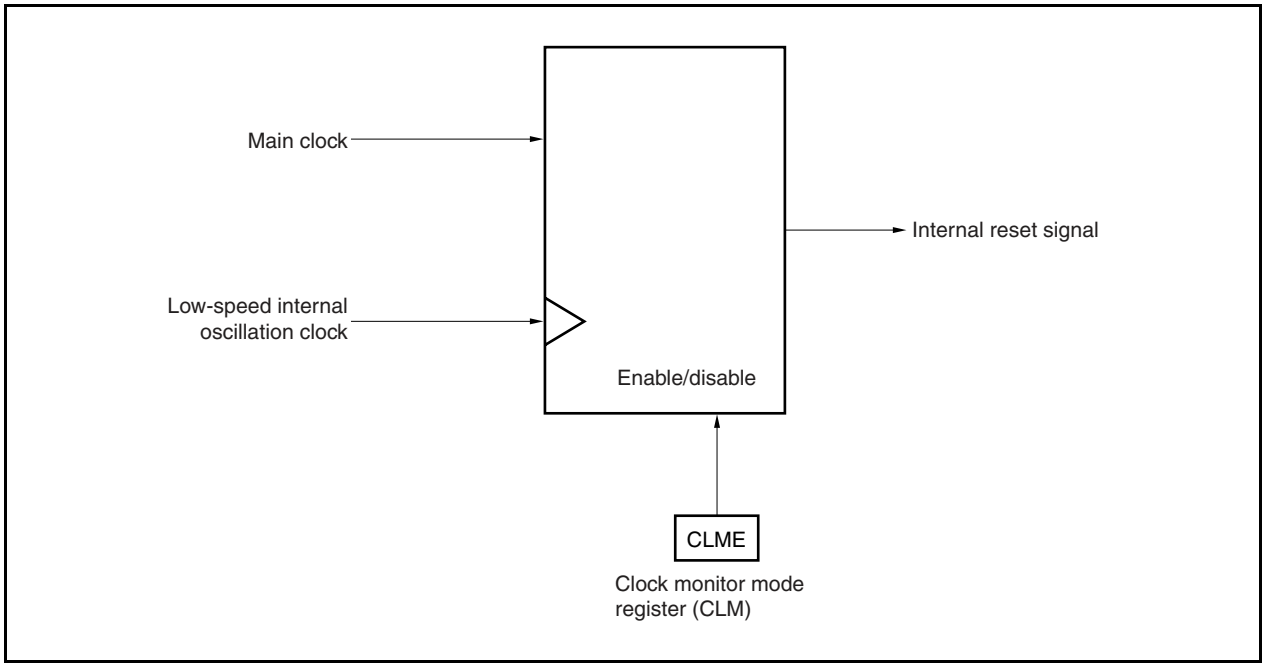
22.2 Configuration

The clock monitor includes the following hardware.

Table 22-1. Configuration of Clock Monitor

Item	Configuration
Control register	Clock monitor mode register (CLM)

Figure 22-1. Block Diagram of Clock Monitor



22.3 Register

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. This can be written only in a special combination of 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 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.

22.4 Operation

This section explains the functions of the clock monitor. The start and stop conditions are as follows.

<Start condition>

Enabling operation by setting the CLM.CLME bit to 1

<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 = 1 during subclock operation to when PCC.CLS bit = 0 during main clock operation)
- When the sampling clock (low-speed internal oscillation clock) is stopped
- When the CPU operates using the low-speed internal oscillation clock
- When the CPU operates using the high-speed internal oscillation clock

Table 22-2. Operation Status of Clock Monitor
(When CLM.CLME Bit = 1, During Low-Speed Internal Oscillation Clock Operation)

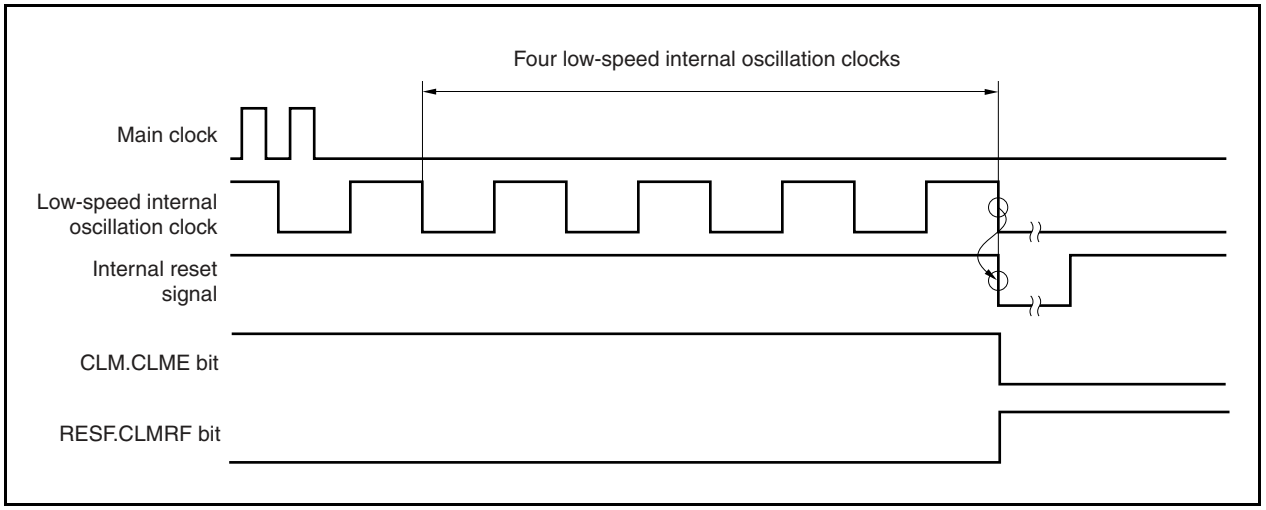
CPU Operating Clock	Operation Mode	Status of Main Clock	Status of Low-Speed Internal Oscillation Clock	Status of Clock Monitor
Main clock	HALT mode	Oscillates	Oscillates	Operates[Note]
	IDLE1, IDLE2 modes	Oscillates	Oscillates	Operates[Note]
	STOP mode	Stops	Oscillates	Stops
Subclock (MCK bit of PCC register = 0)	Sub-IDLE mode	Oscillates	Oscillates	Operates[Note]
Subclock (MCK bit of PCC register = 1)	Sub-IDLE mode	Stops	Oscillates	Stops
High-speed internal oscillation clock (f_{RH}) (PCC.MCK bit = 0)	–	Oscillates	Oscillates	Stops[Note]
High-speed internal oscillation clock (f_{RH}) (PCC.MCK bit = 1)	–	Stops	Oscillates	Stops
Low-speed internal oscillation clock (f_{RL})	–	Stops	Oscillates	Stops
During reset	–	Stops	Stops	Stops

Note The clock monitor is stopped while the internal oscillator is stopped.

(1) Operation when main clock oscillation is stopped (CLME bit = 1)

If oscillation of the main clock is stopped when the CLME bit = 1, an internal reset signal is generated as shown in Figure 22-2.

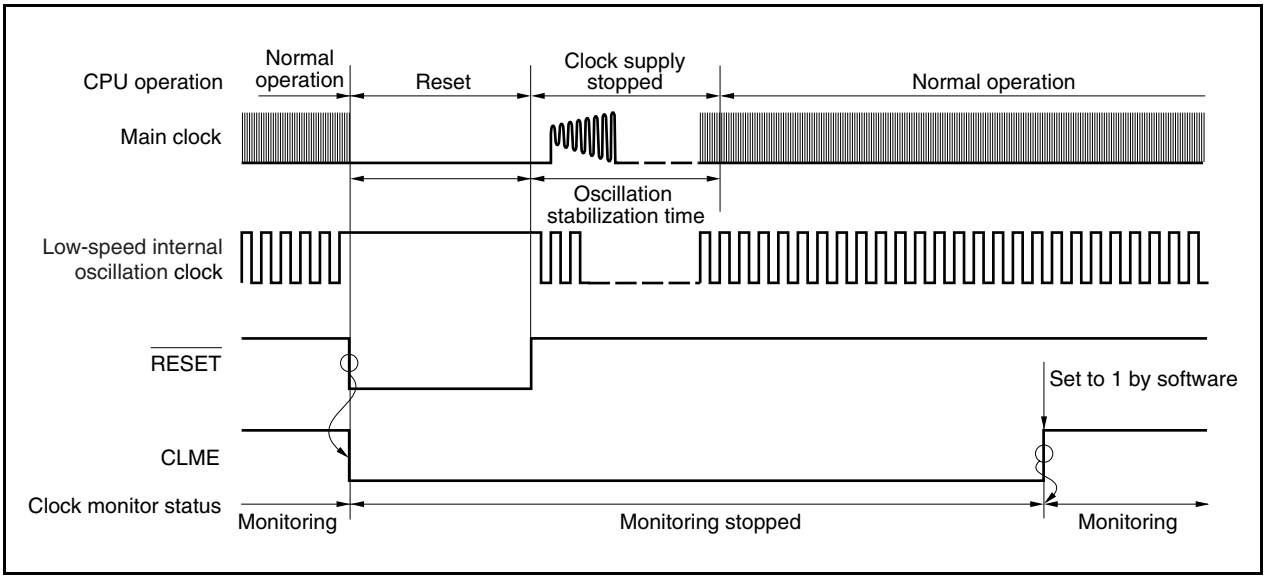
Figure 22-2. Reset Period Due to That Oscillation of Main Clock Is Stopped



(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 CLME bit is set to 1 by software at the end of the oscillation stabilization time of the main clock, monitoring is started.

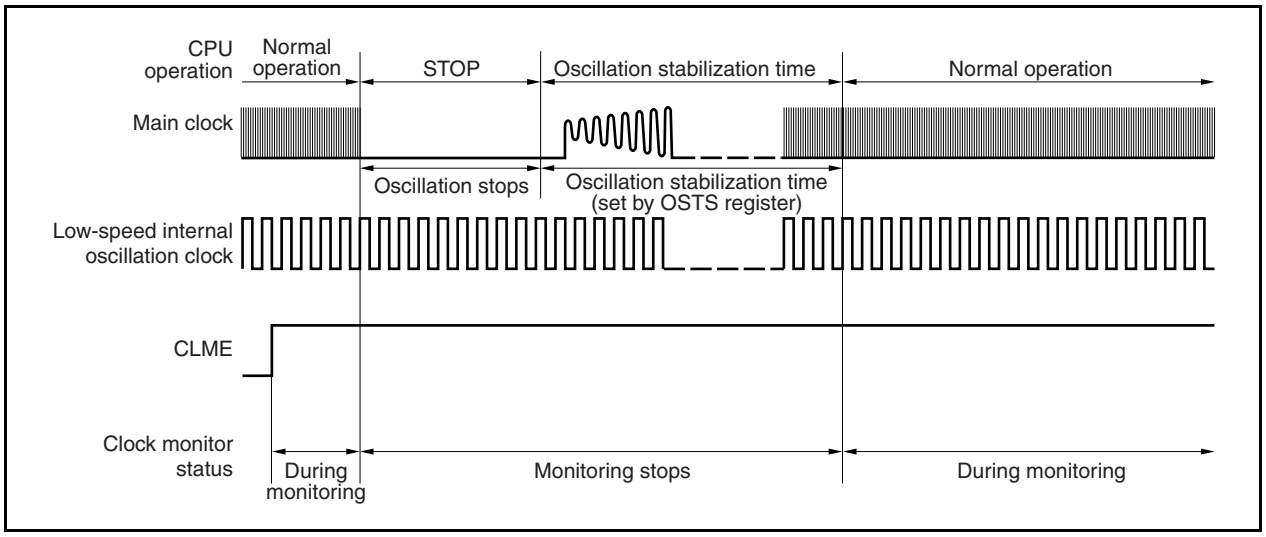
Figure 22-3. Clock Monitor Status After $\overline{\text{RESET}}$ Input
(CLM.CLME bit = 1 is set after $\overline{\text{RESET}}$ input and main clock oscillation stabilization time)



(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.

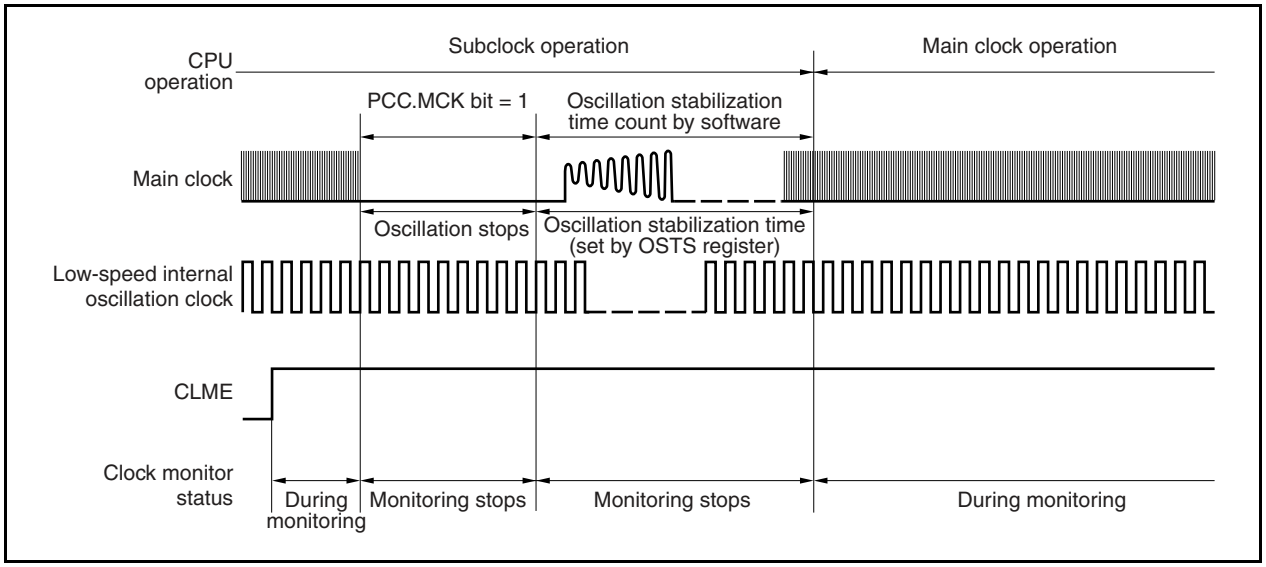
Figure 22-4. Operation in STOP Mode or After STOP Mode Is Released



(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 22-5. Operation When Main Clock Is Stopped (Arbitrary)



(5) Operation while CPU is operating on low-speed internal oscillation clock (CCLS.CCLSF bit = 1)

The monitor operation is not started when the CCLSF bit is 1, even if the CLME bit is set to 1.

(6) Operation while CPU is operating on high-speed internal oscillation clock (MCM.MCS bit = 0)

The monitor operation is not started when the MCM.MCS bit is 0, even if the CLM.CLME bit is set to 1.

CHAPTER 23 POWER-ON CLEAR CIRCUIT

23.1 Function

Functions of the power-on-clear (POC) circuit are shown below.

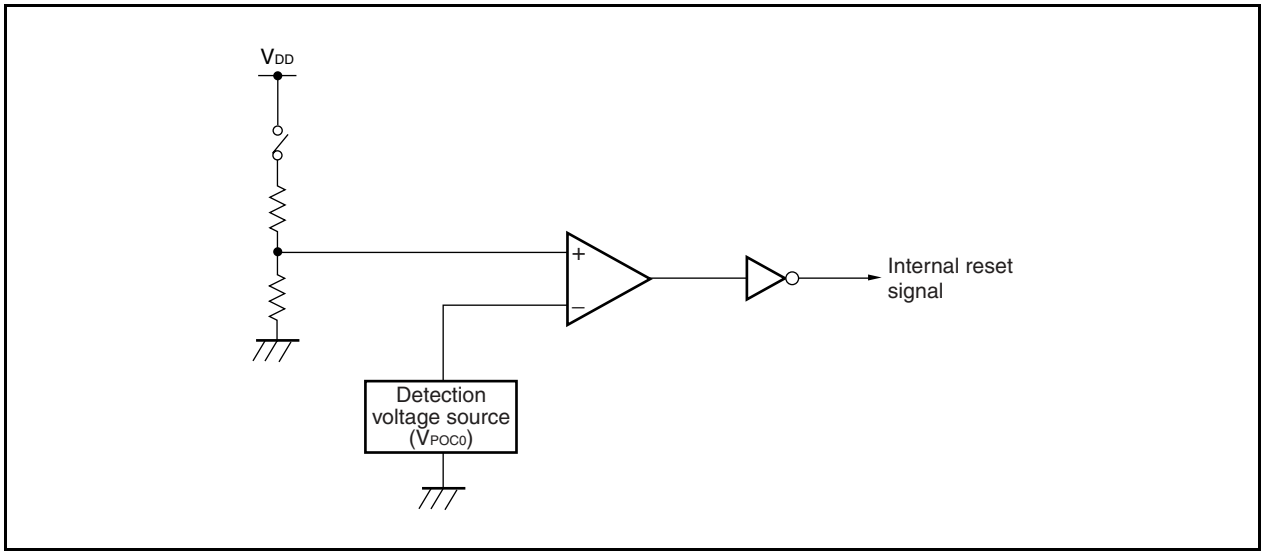
- Generates a reset signal upon power application.
- Compares the supply voltage (V_{DD}) and detection voltage (V_{POC0}), and generates a reset signal when $V_{DD} < V_{POC0}$ (detection voltage (V_{POC0}): 3.5 V ±0.2 V).

Remark The V850ES/Hx3 has plural internal hardware units that generate an internal reset signal. When the system is reset by watchdog timer 2 (WDT2RES), low-voltage detector (LVI), or clock monitor (CLM), a flag corresponding to the reset source is allocated to the reset source flag register (RESF).
The RESF register is not cleared when an internal reset signal is generated by WDT2RES, LVI, or clock monitor, and its flag corresponding to the reset source is set to 1. For details of the RESF register, see **CHAPTER 21 RESET FUNCTIONS**.

23.2 Configuration

The block diagram is shown below.

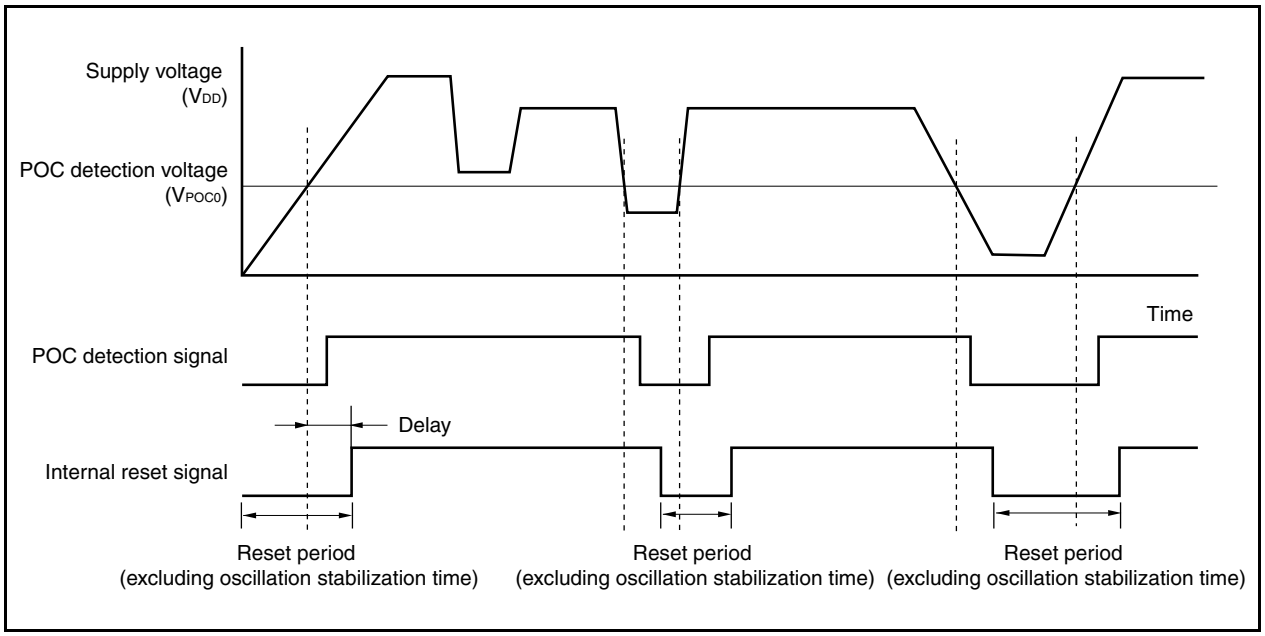
Figure 23-1. Block Diagram of Power-on-Clear Circuit



23.3 Operation

When the supply voltage and detection voltage are compared and if the supply voltage is lower than the detection voltage (including at power application), the system is reset and each hardware is returned to the specific status.

Figure 23-2. Timing of Reset Signal Generation by Power-on-Clear Circuit



CHAPTER 24 LOW-VOLTAGE DETECTOR

24.1 Functions

The low-voltage detector (LVI) has the following functions.

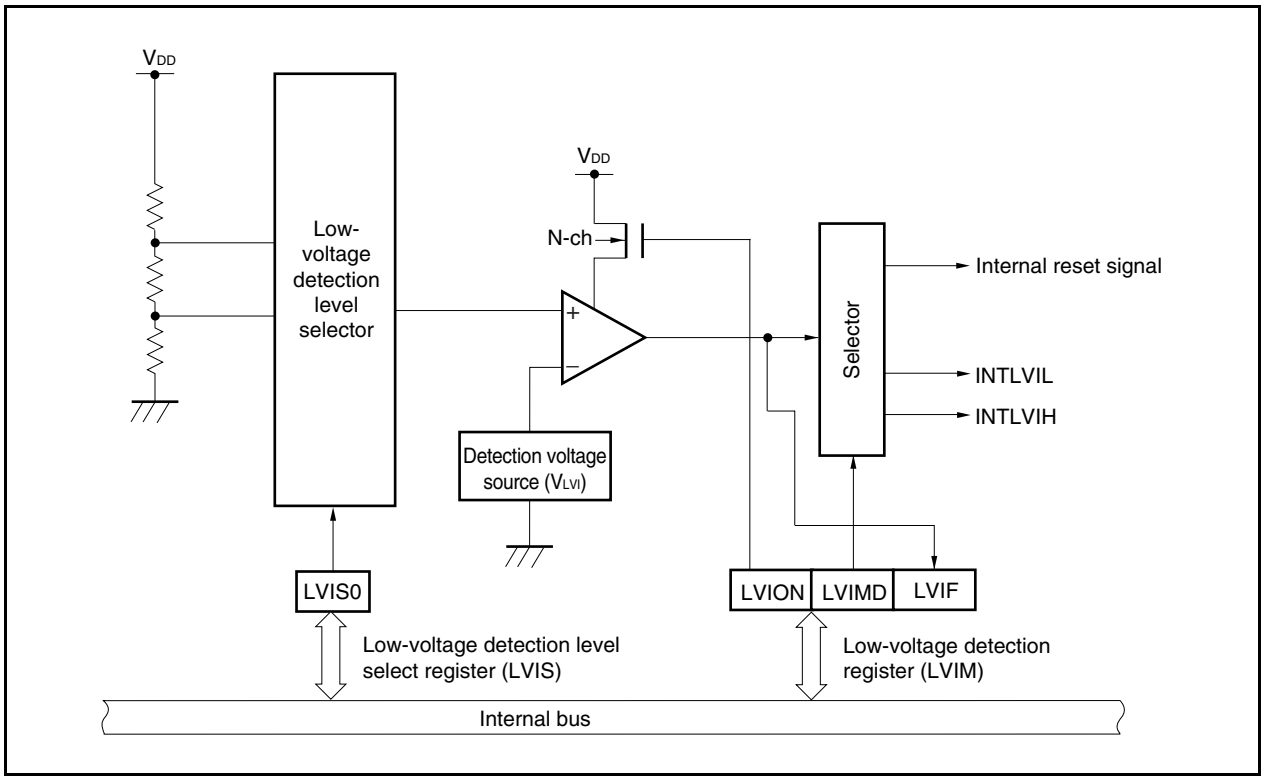
- If it is selected to generate an interrupt when a low voltage is detected, the supply voltage (V_{DD}) and detection voltage (V_{LVI}) are compared and an internal interrupt signal is generated if the supply voltage falls below or rises above the detection voltage.
- If it is selected to generate a reset signal when a low voltage is detected, the supply voltage (V_{DD}) and detection voltage (V_{LVI}) are compared and an internal reset signal is generated when $V_{DD} < V_{LV1}$.
- The level of the supply voltage to be detected can be changed by software (in two steps).
- An interrupt request signal or internal reset signal can be selected.
- Can operate in STOP mode.
- Operation can be stopped by software.

If the low-voltage detector is used to generate a reset signal, the RESF.LVIRF bit is set to 1 when the reset signal is generated. For details of the RESF register, see **CHAPTER 21 RESET FUNCTIONS**.

24.2 Configuration

The block diagram is shown below.

Figure 24-1. Block Diagram of Low-Voltage Detector



24.3 Registers

(1) Low-voltage detection register (LVIM)

The LVIM register is used to enable or disable low voltage detection, and to set the operation mode of the low-voltage detector. The LVIM register is a special register. It can be written only by a combination of specific sequences (see **3.4.7 Special registers**).

This register can be read or written in 8-bit or 1-bit units. However, bit 0 is read-only.

After reset: 00H R/W Address: FFFFF890H

	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVION	0	0	0	0	0	LVIMD	LVIF

LVION	Low voltage detection operation enable or disable
0	Disables operation.
1	Enables operation.

LVIMD	Selection of operation mode of low voltage detection
0	Generates interrupt request signal INTLVIL when the supply voltage falls below the detection voltage. Generates interrupt request signal INTLVIH when the supply voltage rises above the detection voltage.
1	Generates internal reset signal LVIRES when supply voltage < detection voltage.

LVIF	Low voltage detection flag
0	When supply voltage > detection voltage, or when operation is disabled
1	When supply voltage < detection voltage

Cautions 1. After setting the LVION bit to 1, wait for 0.2 ms (max.) before checking the voltage using the LVIF bit.

2. The value of the LVIF flag is output as the output signal INTLVIL or INTLVIH when the LVION bit = 1 and LVIMD bit = 0.

3. Be sure to set bits 2 to 6 to "0".

4. The low-voltage detector cannot be stopped until a reset request due to something other than low-voltage detection is generated after the LVIM.LVION and LVIM.LVIMD bits are set to 1.

(2) Low-voltage detection level select register (LVIS)

The LVIS register is used to select the level of low voltage to be detected.

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

After reset: 00H R/W Address: FFFFF891H

	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	0	0	0	LVIS0

LVIS0	Detection level
0	4.0 V ±0.2 V
1	3.7 V ±0.2 V

Cautions 1. This register cannot be written until a reset request due to something other than low-voltage detection is generated after the LVIM.LVION and LVIM.LVIMD bits are set to 1.

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

(3) Internal RAM data status register (RAMS)

The RAMS register is a flag register that indicates whether the internal RAM is valid or not. The RAMS register is a special register. It can be written only by a combination of specific sequences (see **3.4.7 Special registers**).

For the RAMS register, see **24.5 RAM Retention Voltage Detection Operation**.

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

Caution The following shows the specific sequence after reset.

- **Setting conditions: Detection of voltage lower than detection level (2.0 V ±0.1 V)**
  Set by instruction
- **Clearing condition: Writing of 0 in specific sequence**

After reset: 01H R/W Address: FFFFF892H

	7	6	5	4	3	2	1	<0>
RAMS	0	0	0	0	0	0	0	RAMF

RAMF	Internal RAM data valid/invalid
0	Does not detect voltage equal to or lower than RAM retention voltage.
1	Detects voltage equal to or lower than RAM retention voltage.

24.4 Operation

Depending on the setting of the LVIM.LVIMD bit, an interrupt request signal (INTLVIL, INTLVIH) or an internal reset signal is generated.

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).

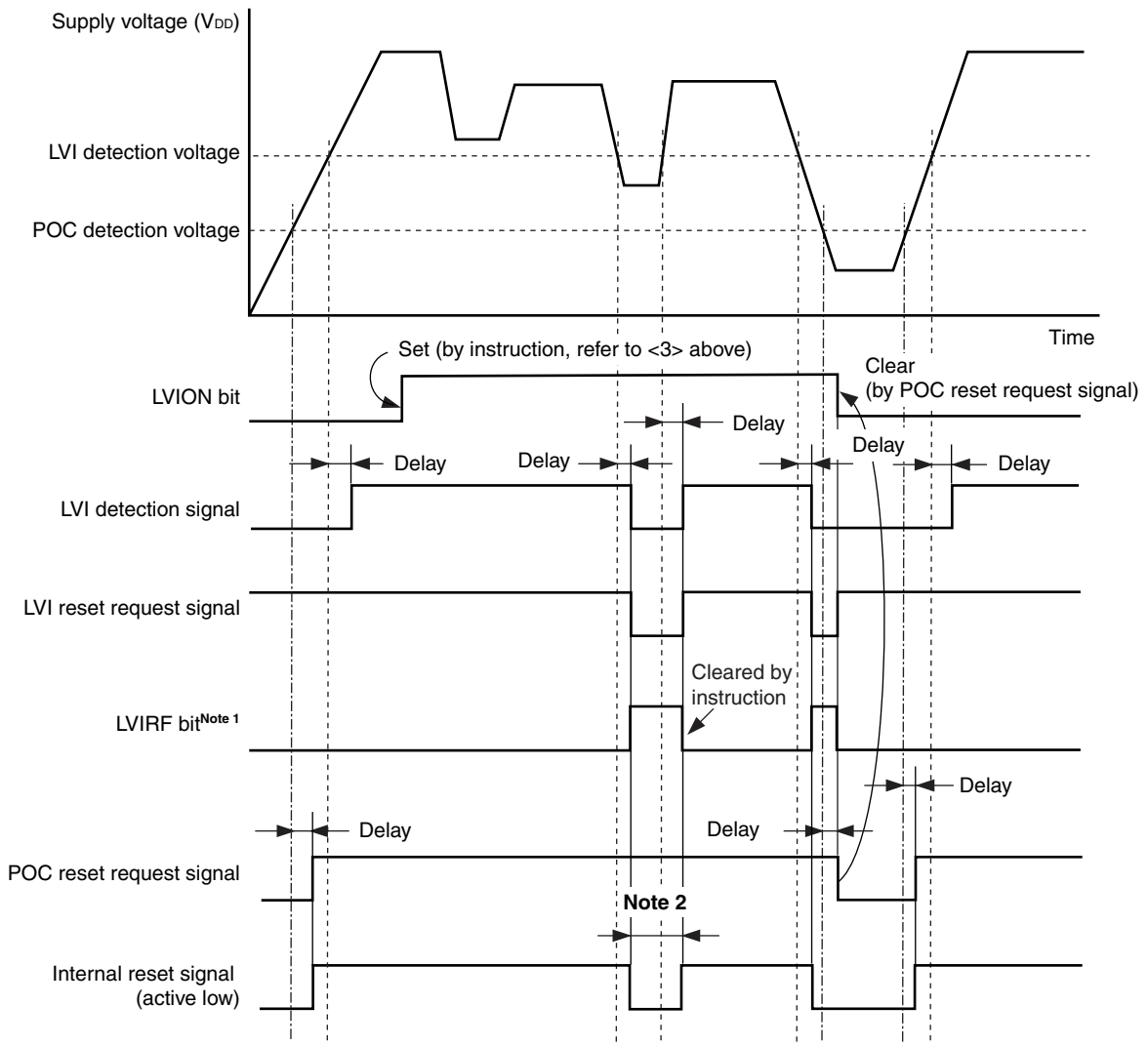
<4> Insert a wait cycle of 0.2 ms max. by software.

<5> By using the LVIM.LVIF bit, check if the supply voltage > detection voltage.

<6> Set the LVIM.LVIMD bit to 1 (to generate an internal reset signal).

Caution If the LVIMD bit is set to 1, the contents of the LVIM and LVIS registers cannot be changed until a reset request other than LVI is generated.

Figure 24-2. Operation Timing of Low-Voltage Detector (LVIMD Bit = 1)



Notes 1. The LVIRF bit is bit 0 of the reset source flag register (RESF). For details of RESF, see **CHAPTER 21 RESET FUNCTIONS**.

2. During the period in which the supply voltage is the set voltage or lower, the internal reset signal is retained (internal reset state).

24.4.2 To use for interrupt

<To start operation>

<1> Mask the interrupt of LVI (LVILIC.LVILMK = 1, LVIHIC.LVIHMK = 1).

<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., by software.

<5> By using the LVIM.LVIF bit, check if the supply voltage > detection voltage.

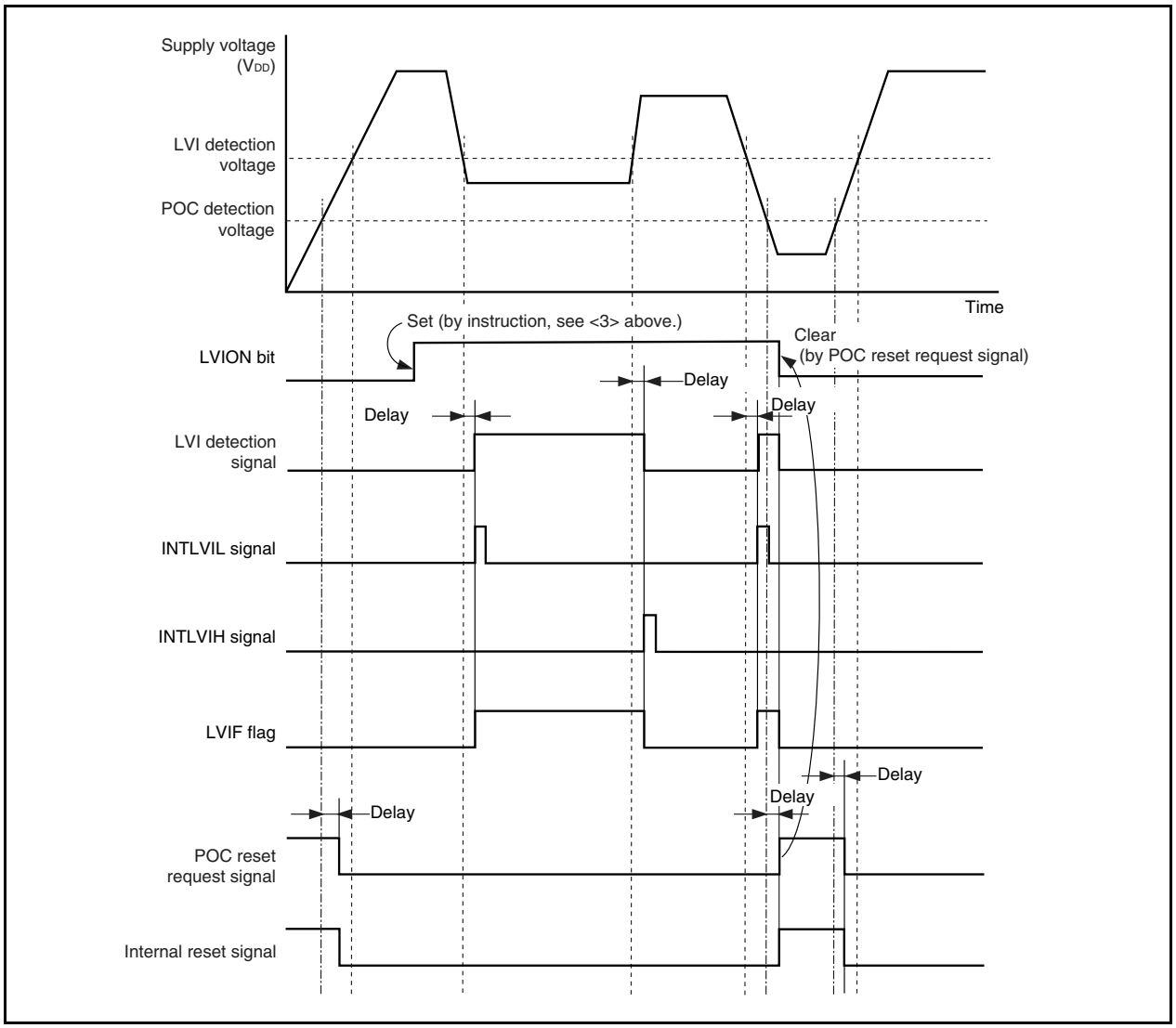
<6> Clear the interrupt request flag of LVI.

<7> Unmask the interrupt of LVI.

<To stop operation>

Clear the LVION bit to 0.

Figure 24-3. Operation Timing of Low-Voltage Detector (LVIMD Bit = 0)



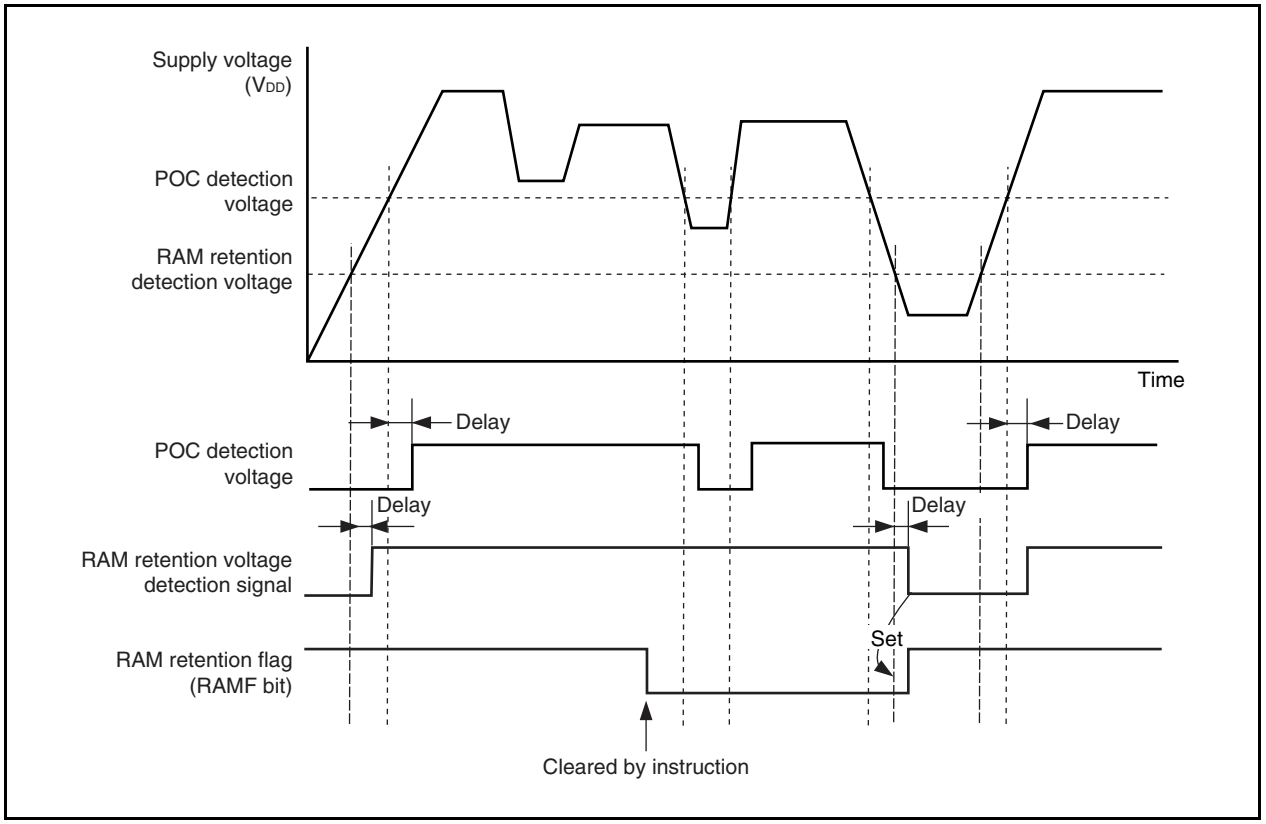
Caution If V_{DD} fluctuates in the neighborhood of the LVI detection level, the INTLVIL or INTLVIH interrupt may be generated more than once. In this case, which interrupt has been generated last cannot be identified. Identify the interrupt by software, such as by monitoring the LVIF flag.

24.5 RAM Retention Voltage Detection Operation

The supply voltage and detection voltage are compared. When the supply voltage drops below the detection voltage (including on power application), the RAMS.RAMF bit is set to 1.

When the POC function is not used and when the RAM retention voltage detection function is used, be sure to input an external reset signal if the detected voltage falls below the operating voltage.

Figure 24-4. Operation Timing of RAM Retention Voltage Detection Function



24.6 Emulation Function

When an in-circuit emulator is used, the operation of the RAM retention flag (RAMS.RAMF bit) can be pseudo-controlled and emulated by manipulating the PEMU1 register on the debugger.

This register is valid only in the emulation mode. It is invalid in the normal mode.

(1) Peripheral emulation register 1 (PEMU1)

After reset: 00H R/W Address: FFFFF9FEH

	7	6	5	4	3	2	1	0
PEMU1	0	0	0	0	0	EVARAMIN	0	0

EVARAMIN	Pseudo specification of RAM retention voltage detection signal
0	Do not detect voltage lower than RAM retention voltage.
1	Detect voltage lower than RAM retention voltage (set RAMF flag).

Caution This bit is not automatically cleared.

[Usage]

When an in-circuit emulator is used, pseudo emulation of RAMF is realized by rewriting this register on the debugger.

<1> CPU break (CPU operation stops.)

<2> Set the EVARAMIN bit to 1 by using a register write command.
By setting the EVARAMIN bit to 1, the RAMF bit is set to 1 on hardware (the internal RAM data is invalid).

<3> Clear the EVARAMIN bit to 0 by using a register write command again.
Unless this operation is performed (clearing the EVARAMIN bit to 0), the RAMF bit cannot be cleared to 0 by a CPU operation instruction.

<4> Run the CPU and resume emulation.

CHAPTER 25 REGULATOR

25.1 Overview

The V850ES/Hx3 includes 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 and output buffers). The regulator output voltage is set to 2.5 V (TYP.).

Figure 25-1. Regulator (1/2)

(a) V850ES/HE3, V850ES/HF3

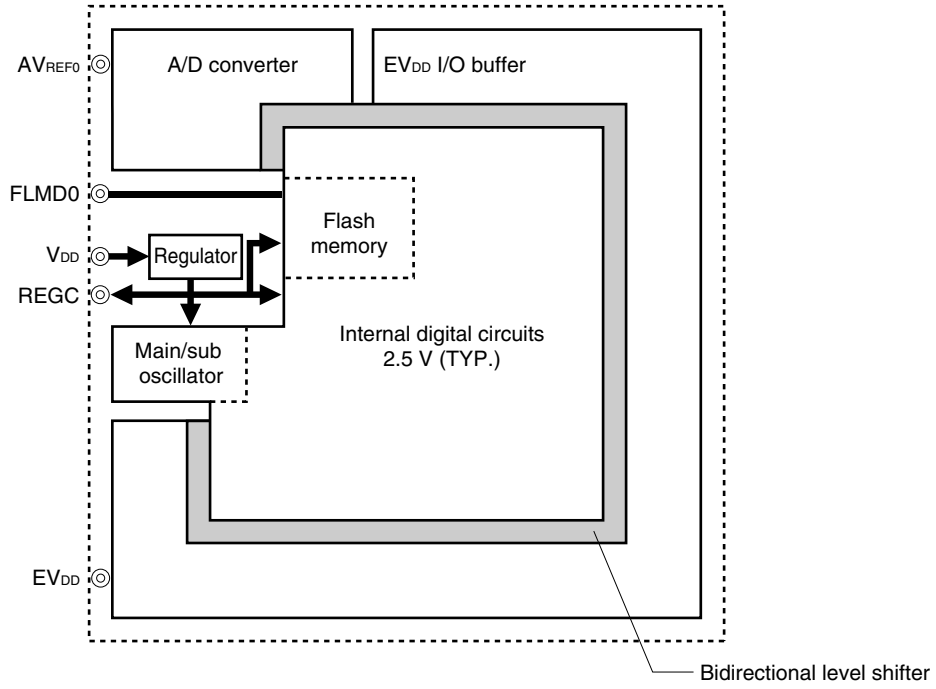
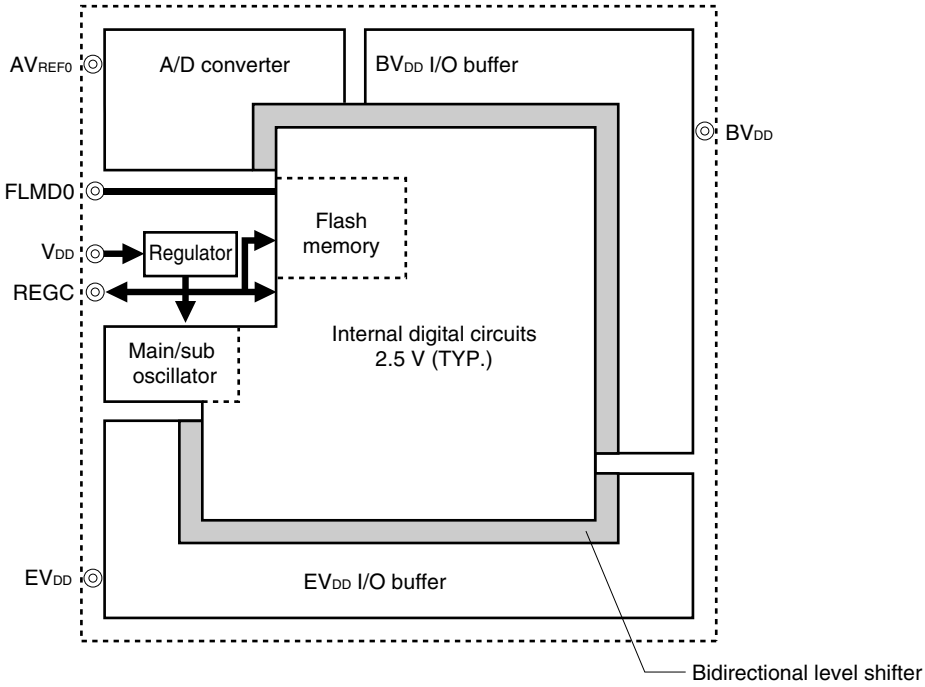
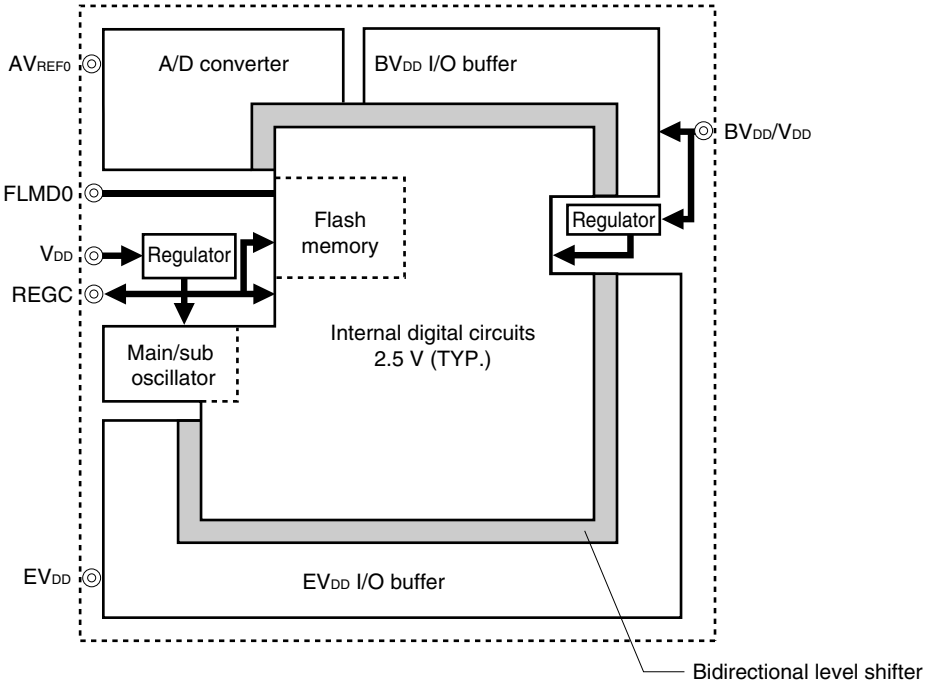


Figure 25-1. Regulator (2/2)

(b) V850ES/HG3



(c) V850ES/HJ3



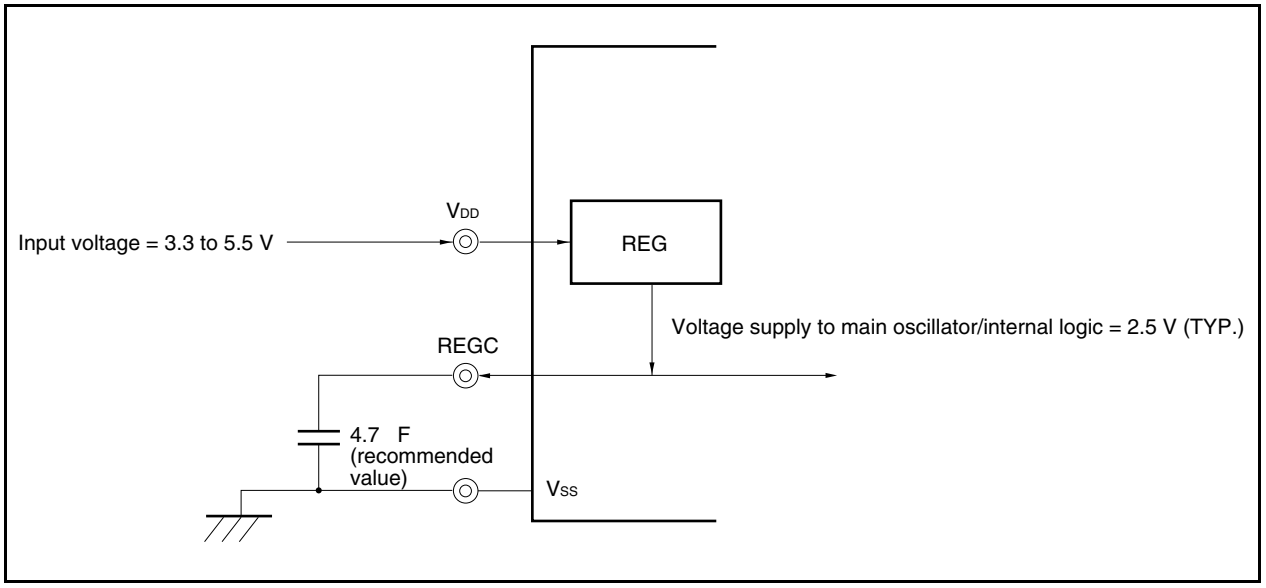
25.2 Operation

The regulator of this product always operates in any mode (normal operation mode, HALT mode, IDLE1 mode, IDLE2 mode, STOP mode, or during reset).

Be sure to connect a capacitor (4.7 F (recommended value)) to the REGC pin to stabilize the regulator output.

A diagram of the regulator pin connection method is shown below.

Figure 25-2. REGC Pin Connection



CHAPTER 26 FLASH MEMORY

The V850ES/Hx3 incorporates a flash memory.

- 128 KB flash memory: μPD70F3747
- 256 KB flash memory: μPD70F3750, 70F3752, 70F3755
- 512 KB flash memory: μPD70F3757

Flash memory versions offer the following advantages for development environments and mass production applications.

- For altering software after the V850ES/Hx3 is soldered onto the target system.
- For data adjustment when starting mass production.
- For differentiating software according to the specification in small scale production of various models.
- For facilitating inventory management.
- For updating software after shipment.

26.1 Features

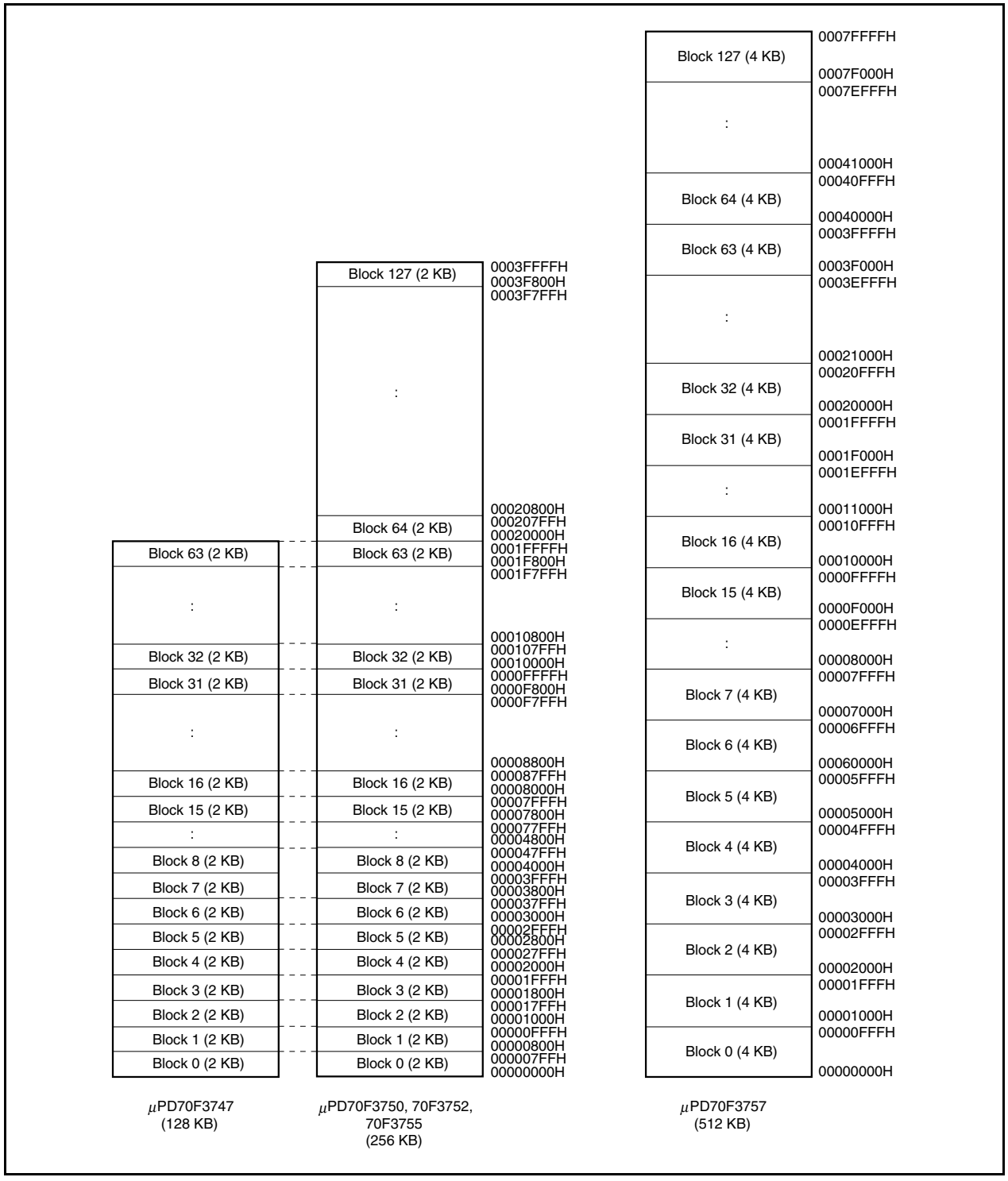
- 4-byte/1-clock access (when instruction is fetched)
- Capacity: 512/256/128 KB
- Write voltage: Erase/write with a single power supply
- Rewriting method
  - Rewriting by communication with dedicated flash programmer via serial interface (on-board/off-board programming)
  - Rewriting flash memory by 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.

26.2 Memory Configuration

The V850ES/Hx3 internal flash memory area is divided into 64 or 128 blocks and can be programmed/erased in block units. All the blocks can also be erased at once.

When the boot swap function is used, the set boot swap cluster size (see **Table 26-13**) is replaced. For details of the boot swap function, see **26.5 Rewriting by Self Programming**.

Figure 26-1. Flash Memory Mapping



26.3 Functional Outline

The internal flash memory of the V850ES/Hx3 can be rewritten by using the rewrite function of the dedicated flash programmer, regardless of whether the V850ES/Hx3 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 is 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 is supported during self programming, so that the flash memory can be rewritten under various conditions, such as while communicating with an external device.

Table 26-1. Rewrite Method

Rewrite Method	Functional Outline	Operation Mode
On-board programming	Flash memory can be rewritten after the device is mounted on the target system, by using a dedicated flash programmer.	Flash memory programming mode
Off-board programming	Flash memory can be rewritten before the device is mounted on the target system, by using a dedicated flash programmer and a dedicated program adapter board (FA series).	
Self programming	Flash memory can be rewritten by executing a user program that has been written to the flash memory in advance by means of 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 or external memory in advance).	Normal operation mode

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

Table 26-2. Basic Functions

Function	Functional Outline	Support (√: Supported, ×: Not supported)	
		On-Board/Off-Board Programming	On-Board/Off-Board 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 write level is secured are performed.	√	√
Verify/checksum	Data read from the flash memory is compared with data transferred from the flash programmer.	√	× (Can be read by user program)
Read	Data written to the flash memory is read.	√	×
Security setting	Use of the chip erase command, block erase command, program command, and read command is prohibited, and rewriting of the boot area is prohibited.	√	× (Supported only when setting is changed from enable to disable)

Note Chip erasure can be executed by specifying the entire memory area by using 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 setting functions are enabled by default after shipment, and security can be set by rewriting via on-board/off-board programming. Each security function can be used in combination with the others at the same time.

Table 26-3. 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 is prohibited. Setting of prohibition can be initialized by execution of the chip erase command.
Read command prohibit	Execution of a read command on all of the blocks is prohibited. Setting of the prohibition can be initialized by execution of a chip erase command.
Boot block cluster rewrite prohibit setting	Boot block cluster from block 0 to a specified block can be protected. Once a boot block cluster has been protected, it cannot be rewritten (or erased/written) afterward. Even if a chip erase command is executed, the prohibited setting cannot be initialized. The maximum block that can be specified is as follows. μPD70F3747: Block 63 μPD70F3750, 70F3752, 70F3755, 70F3757: Block 127

Table 26-4. Security Setting

Function	Erase, Write, Read Operations When Each Security Is Set (√: Executable, ×: Not Executable, –: Not Supported)		Notes on Security Setting	
	On-Board/ Off-Board Programming	Self Programming	On-Board/ Off-Board Programming	Self Programming
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 block cluster 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 different from the data already written in the flash memory cannot be written.

2. Executable except in boot area.

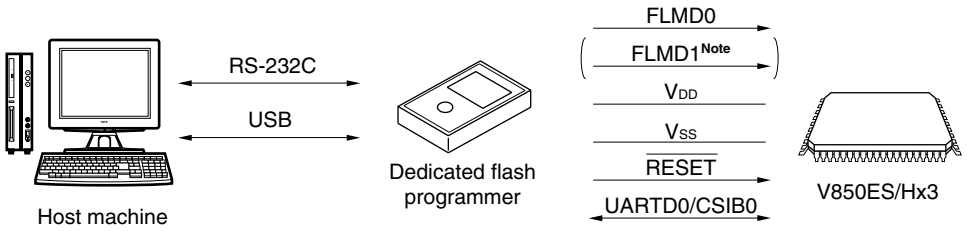
26.4 Rewriting by Dedicated Flash Programmer

The flash memory can be rewritten by using a dedicated flash programmer after the V850ES/Hx3 is mounted on the target system (on-board programming). The flash memory can also be rewritten before the device is mounted on the target system (off-board programming) by using a dedicated program adapter (FA series).

26.4.1 Programming environment

The following shows the environment required for writing programs to the flash memory of the V850ES/Hx3.

Figure 26-2. Environment Required for Writing Programs to Flash Memory



Note Connect the FLMD1 pin to the flash programmer or connect to GND via a pull-down resistor on the board.

A host machine is required for controlling the dedicated flash programmer.

UARTD0 or CSIB0 is used for the interface between the dedicated flash programmer and the V850ES/Hx3 to perform writing, erasing, etc. A dedicated program adapter (FA series) required for off-board writing.

For V850ES/HE3

- FA-70F3371GB-GAH-RX (already wired)Note
- FA-64GB-GAH-B (not wired: wiring required)

For V850ES/HF3

- FA-70F3373GK-GAK-RX (already wired)Note
- FA-80GK-GAK-B (not wired: wiring required)

For V850ES/HG3

- FA-70F3375GC-UEU-RX (already wired)Note
- FA-100GC-UEU-B (not wired: wiring required)

For V850ES/HJ3

- FA-70F3378GJ-GAE-RX (already wired)Note
- FA-144GJ-GAE-B (not wired: wiring required)

Note Under development

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

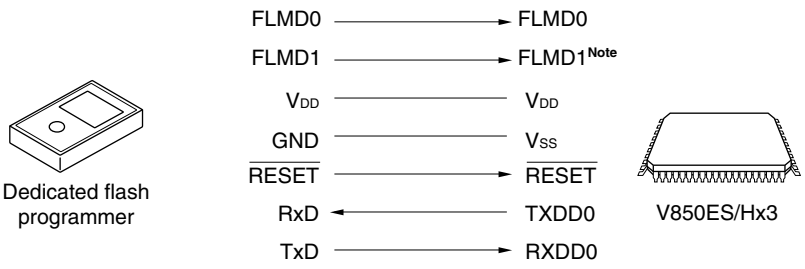
26.4.2 Communication mode

Communication between the dedicated flash programmer and the V850ES/Hx3 is performed by serial communication using the UARTD0 or CSIB0 interface of the V850ES/Hx3.

(1) UARTD0

Transfer rate: 9,600 to 153,600 bps

Figure 26-3. Communication with Dedicated Flash Programmer (UARTD0)

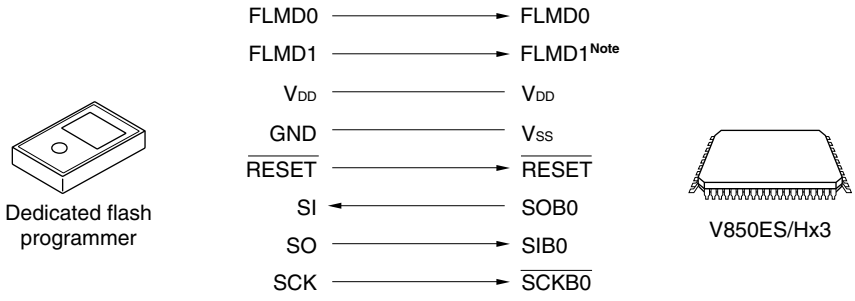


Note Connect the FLMD1 pin to the flash programmer or connect to GND via a pull-down resistor on the board.

(2) CSIB0

Serial clock: 2.4 kHz to 5.0 MHz (MSB first)

Figure 26-4. Communication with Dedicated Flash Programmer (CSIB0)

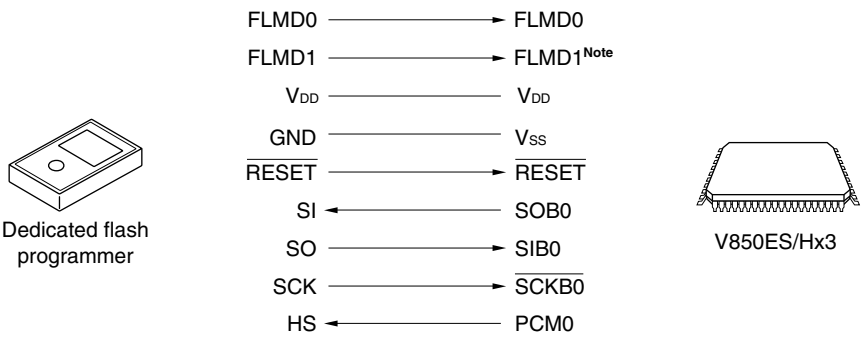


Note Connect the FLMD1 pin to the flash programmer or connect to GND via a pull-down resistor on the board.

(3) CSIB0 + HS

Serial clock: 2.4 kHz to 5.0 MHz (MSB first)

Figure 26-5. Communication with Dedicated Flash Programmer (CSIB0 + HS)



Note Connect the FLMD1 pin to the flash programmer or connect to GND via a pull-down resistor on the board.

The dedicated flash programmer outputs the transfer clock, and the V850ES/Hx3 operates as a slave.

When the PG-FP5 is used as the dedicated flash programmer, it generates the following signals to the V850ES/Hx3. For details, refer to the **PG-FP5 User's Manual (U18865E)**.

Table 26-5. Signal Connections of Dedicated Flash Programmer (PG-FP5)

PG-FP5			V850ES/Hx3	Processing for Connection		
Signal Name	I/O	Pin Function	Pin Name	UARTD0	CSIB0	CSIB0 + HS
FLMD0	Output	Write enable/disable	FLMD0	◎	◎	◎
FLMD1	Output	Write enable/disable	FLMD1	◎Note 1	◎Note 1	◎Note 1
VDD	–	V_{DD} voltage generation/voltage monitor	V_{DD}	◎	◎	◎
GND	–	Ground	V_{SS}	◎	◎	◎
CLK	Output	Clock output to V850ES/Hx3	X1, X2	×Note 2	×Note 2	×Note 2
$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	◎	◎	◎
SI/RxD	Input	Receive signal	SOB0/TXDD0	◎	◎	◎
SO/TxD	Output	Transmit signal	SIB0/RXDD0	◎	◎	◎
SCK	Output	Transfer clock	$\overline{\text{SCKB0}}$	×	◎	◎
HS	Input	Handshake signal for CSIB0 + HS communication	PCM0	×	×	◎

Notes 1. Wire these pins as shown in Figure 26-6 or connect then to GND via pull-down resistor on board.

2. Clock cannot be supplied via the CLK pin of the flash programmer. Create an oscillator on board and supply the clock.

Remark ◎: Must be connected.

×: Does not have to be connected.

Table 26-6. Wiring of V850ES/HE3 Flash Writing Adapters (FA-64GB-GAH-B)

Flash Programmer (FG-FP5) Connection Pin			Name of FA Board Pin	CSIB0 + HS Used		CSIB0 Used		UARTD0 Used	
Signal Name	I/O	Pin Function		Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	SI	P41/SOB0/ KR1	20	P41/SOB0/ KR1	20	P30/TXDD0	22
SO/TxD	Output	Transmit signal	SO	P40/SIB0/ KR0	19	P40/SIB0/ KR0	19	P31/RXDD0/ INTP7	23
SCK	Output	Transfer clock	SCK	P42/$\overline{\text{SCKB0}}$/ KR2	21	P42/$\overline{\text{SCKB0}}$/ KR2	21	Not needed	–
CLK	Output	Clock to V850ES/HE3	X1	Not needed	–	Not needed	–	Not needed	–
			X2	Not needed	–	Not needed	–	Not needed	–
/RESET	Output	Reset signal	/RESET	$\overline{\text{RESET}}$	9	$\overline{\text{RESET}}$	9	$\overline{\text{RESET}}$	9
FLMD0	Output	Write voltage	FLMD0	FLMD0	3	FLMD0	3	FLMD0	3
FLMD1	Output	Write voltage	FLMD1	PDL5/FLMD1	52	PDL5/FLMD1	52	PDL5/FLMD1	52
HS	Input	Handshake signal for CSI0 + HS communication	RESERVE/ HS	PCM0	45	Not needed	–	Not needed	–
VDD	–	VDD voltage generation/ voltage monitoring	VDD	V_{DD}	4	V_{DD}	4	V_{DD}	4
				EV_{DD}	33	EV_{DD}	33	EV_{DD}	33
				AV_{REF0}	1	AV_{REF0}	1	AV_{REF0}	1
GND	–	Ground	GND	V_{SS}	6	V_{SS}	6	V_{SS}	6
				AV_{SS}	2	AV_{SS}	2	AV_{SS}	2
				EV_{SS}	32	EV_{SS}	32	EV_{SS}	32

Cautions 1. Be sure to connect the REGC pin to GND via 4.7 μF capacitor.

2. Clock cannot be supplied from the CLK pin of the flash programmer. Create an oscillator on the board and supply clock.

Table 26-7. Wiring of V850ES/HF3 Flash Writing Adapters (FA-80GK-GAK-B)

Flash Programmer (FG-FP5) Connection Pin			Name of FA Board Pin	CSIB0 + HS Used		CSIB0 Used		UARTD0 Used	
Signal Name	I/O	Pin Function		Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	SI	P41/SOB0/ KR1	20	P41/SOB0/ KR1	20	P30/TXDD0	22
SO/TxD	Output	Transmit signal	SO	P40/SIB0/ KR0	19	P40/SIB0/ KR0	19	P31/RXDD0/ INTP7	23
SCK	Output	Transfer clock	SCK	P42/$\overline{\text{SCKB0}}$/ KR2	21	P42/$\overline{\text{SCKB0}}$/ KR2	21	Not needed	–
CLK	Output	Clock to V850ES/HF3	X1	Not needed	–	Not needed	–	Not needed	–
			X2	Not needed	–	Not needed	–	Not needed	–
/RESET	Output	Reset signal	/RESET	$\overline{\text{RESET}}$	14	$\overline{\text{RESET}}$	14	$\overline{\text{RESET}}$	14
FLMD0	Output	Write voltage	FLMD0	FLMD0	8	FLMD0	8	FLMD0	8
FLMD1	Output	Write voltage	FLMD1	PDL5/FLMD1	62	PDL5/FLMD1	62	PDL5/FLMD1	62
HS	Input	Handshake signal for CSI0 + HS communication	RESERVE/ HS	PCM0	49	Not needed	–	Not needed	–
VDD	–	VDD voltage generation/ voltage monitoring	VDD	V_{DD}	9	V_{DD}	9	V_{DD}	9
				EV_{DD}	31	EV_{DD}	31	EV_{DD}	31
				AV_{REF0}	1	AV_{REF0}	1	AV_{REF0}	1
GND	–	Ground	GND	V_{SS}	11	V_{SS}	11	V_{SS}	11
				AV_{SS}	2	AV_{SS}	2	AV_{SS}	2
				EV_{SS}	30	EV_{SS}	30	EV_{SS}	30

Cautions 1. Be sure to connect the REGC pin to GND via 4.7 μF capacitor.

2. Clock cannot be supplied from the CLK pin of the flash programmer. Create an oscillator on the board and supply clock.

Table 26-8. Wiring of V850ES/HG3 Flash Writing Adapters (FA-100GC-UEU-B)

Flash Programmer (FG-FP5) Connection Pin			Name of FA Board Pin	CSIB0 + HS Used		CSIB0 Used		UARTD0 Used	
Signal Name	I/O	Pin Function		Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	SI	P41/SOB0/KR1	23	P41/SOB0/KR1	23	P30/TXDD0	25
SO/TxD	Output	Transmit signal	SO	P40/SIB0/KR0	22	P40/SIB0/KR0	22	P31/RXDD0/INTP7	26
SCK	Output	Transfer clock	SCK	P42/$\overline{\text{SCKB0}}$/KR2	24	P42/$\overline{\text{SCKB0}}$/KR2	24	Not needed	–
CLK	Output	Clock to V850ES/HG3	X1	Not needed	–	Not needed	–	Not needed	–
			X2	Not needed	–	Not needed	–	Not needed	–
/RESET	Output	Reset signal	/RESET	$\overline{\text{RESET}}$	14	$\overline{\text{RESET}}$	14	$\overline{\text{RESET}}$	14
FLMD0	Output	Write voltage	FLMD0	FLMD0	8	FLMD0	8	FLMD0	8
FLMD1	Output	Write voltage	FLMD1	PDL5/FLMD1	76	PDL5/FLMD1	76	PDL5/FLMD1	76
HS	Input	Handshake signal for CSI0 + HS communication	RESERVE/HS	PCM0	61	Not needed	–	Not needed	–
VDD	–	VDD voltage generation/voltage monitoring	VDD	V_{DD}	9	V_{DD}	9	V_{DD}	9
				EV_{DD}	5, 34	EV_{DD}	5, 34	EV_{DD}	5, 34
				AV_{REF0}	1	AV_{REF0}	1	AV_{REF0}	1
				BV_{DD}	70	BV_{DD}	70	BV_{DD}	70
GND	–	Ground	GND	V_{SS}	11	V_{SS}	11	V_{SS}	11
				AV_{SS}	2	AV_{SS}	2	AV_{SS}	2
				EV_{SS}	33	EV_{SS}	33	EV_{SS}	33
				BV_{SS}	69	BV_{SS}	69	BV_{SS}	69

Cautions 1. Be sure to connect the REGC pin to GND via 4.7 μF capacitor.

2. Clock cannot be supplied from the CLK pin of the flash programmer. Create an oscillator on the board and supply clock.

Table 26-9. Wiring of V850ES/HJ3 Flash Writing Adapters (FA-144GJ-GAE-B)

Flash Programmer (FG-FP5) Connection Pin			Name of FA Board Pin	CSIB0 + HS Used		CSIB0 Used		UARTD0 Used	
Signal Name	I/O	Pin Function		Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.
SI/RxD	Input	Receive signal	SI	P41/SOB0/ KR1/ TXDD3[Note]	23	P41/SOB0/ KR1/ TXDD3[Note]	23	P30/TXDD0	25
SO/TxD	Output	Transmit signal	SO	P40/SIB0/ KR0/ RXDD3[Note]/ INTP14[Note]	22	P40/SIB0/ KR0/ RXDD3[Note]/ INTP14[Note]	22	P31/RXDD0/ INTP7	26
SCK	Output	Transfer clock	SCK	P42/$\overline{\text{SCKB0}}$/ KR2	24	P42/$\overline{\text{SCKB0}}$/ KR2	24	Not needed	–
CLK	Output	Clock to V850ES/HJ3	X1	Not needed	–	Not needed	–	Not needed	–
			X2	Not needed	–	Not needed	–	Not needed	–
/RESET	Output	Reset signal	/RESET	$\overline{\text{RESET}}$	14	$\overline{\text{RESET}}$	14	$\overline{\text{RESET}}$	14
FLMD0	Output	Write voltage	FLMD0	FLMD0	8	FLMD0	8	FLMD0	8
FLMD1	Output	Write voltage	FLMD1	PDL5/AD5/ FLMD1	110	PDL5/AD5/ FLMD1	110	PDL5/AD5/ FLMD1	110
HS	Input	Handshake signal for CSI0 + HS communication	RESERVE/ HS	PCM0/$\overline{\text{WAIT}}$	85	Not needed	–	Not needed	–
VDD	–	VDD voltage generation/ voltage monitoring	VDD	V_{DD}	9	V_{DD}	9	V_{DD}	9
				EV_{DD}	5, 34	EV_{DD}	5, 34	EV_{DD}	5, 34
				BV_{DD}	104	BV_{DD}	104	BV_{DD}	104
				AV_{REF0}	1	AV_{REF0}	1	AV_{REF0}	1
GND	–	Ground	GND	V_{SS}	11	V_{SS}	11	V_{SS}	11
				AV_{SS}	2	AV_{SS}	2	AV_{SS}	2
				EV_{SS}	33	EV_{SS}	33	EV_{SS}	33
				BV_{SS}	103	BV_{SS}	103	BV_{SS}	103

Cautions 1. Be sure to connect the REGC pin to GND via 4.7 μF capacitor.

2. Clock cannot be supplied from the CLK pin of the flash programmer.
Create an oscillator on the board and supply clock.

Figure 26-6. Wiring Example of V850ES/HE3 Flash Writing Adapter (FA-64GB-GAH-B) (In CSIB0 + HS Mode) (1/2)

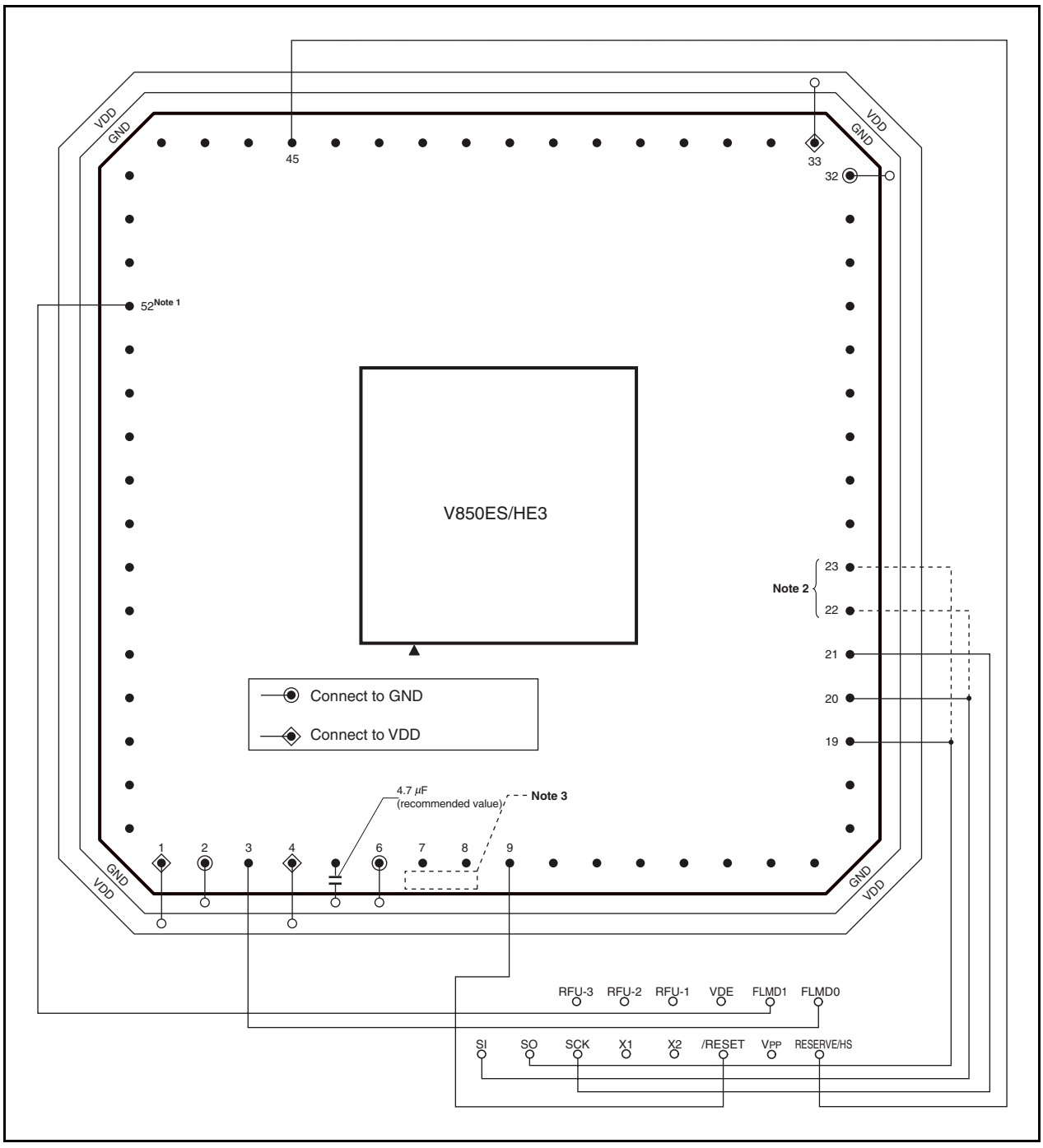


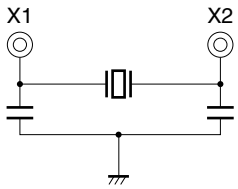
Figure 26-6. Wiring Example of V850ES/HE3 Flash Writing Adapter (FA-64GB-GAH-B) (In CSIB0 + HS Mode) (2/2)

Notes 1. Wire the FLMD1 pin as shown below, or connect it to GND on board via a pull-down resistor.

2. Pins used when UARTD0 is used.

3. Supply a clock by creating an oscillator on the flash writing adapter (enclosed by broken lines). An example of the oscillator is shown below.

Example:



Caution Do not input a high level to the $\overline{\text{DRST}}$ pin.

Remarks 1. Handle the pins not shown, in accordance with the handling of unused pins (see **2.3 Pin I/O Circuit Types, I/O Buffer Power Supplies, and Connection of Unused Pins**).

2. This adapter is for the 64-pin plastic LQFP package.

Figure 26-7. Wiring Example of V850ES/HF3 Flash Writing Adapter (FA-80GK-GAK-B) (In CSIB0 + HS Mode) (1/2)

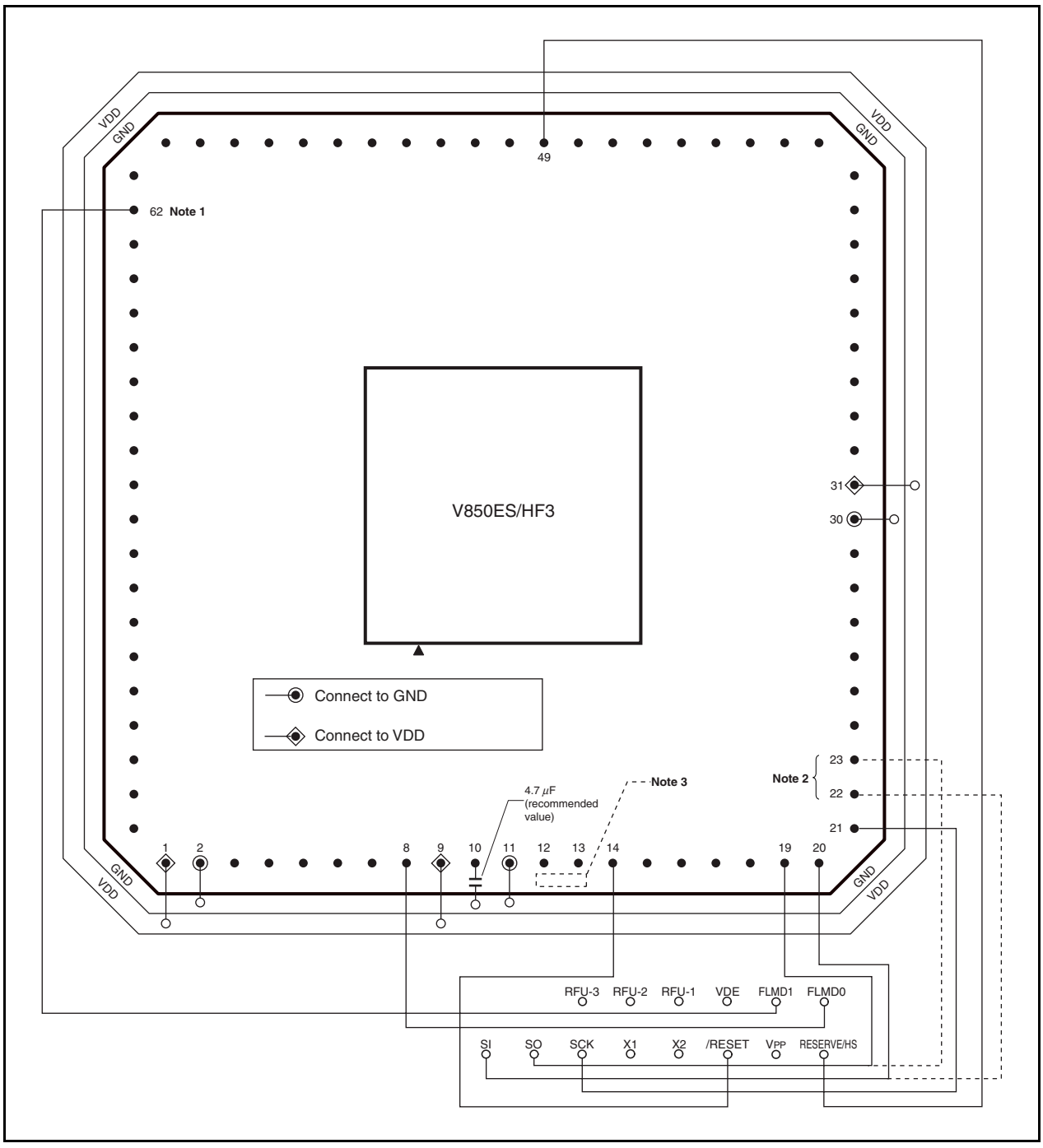


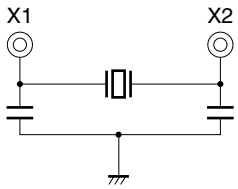
Figure 26-7. Wiring Example of V850ES/HF3 Flash Writing Adapter (FA-80GK-GAK-B) (In CSIB0 + HS Mode) (2/2)

Notes 1. Wire the FLMD1 pin as shown below, or connect it to GND on board via a pull-down resistor.

2. Used when UARTD0 is used.

3. Supply a clock by creating an oscillator on the flash writing adapter (enclosed by broken lines). An example of the oscillator is shown below.

Example:



Caution Do not input a high level to the $\overline{\text{DRST}}$ pin.

Remarks 1. Handle the pins not shown in accordance with the handling of unused pins (see **2.3 Pin I/O Circuit Types, I/O Buffer Power Supplies, and Connection of Unused Pins**).

2. This adapter is for the 80-pin plastic TQFP package.

Figure 26-8. Wiring Example of V850ES/HG3 Flash Writing Adapter (FA-100GC-UEU-B) (In CSIB0 + HS Mode) (1/2)

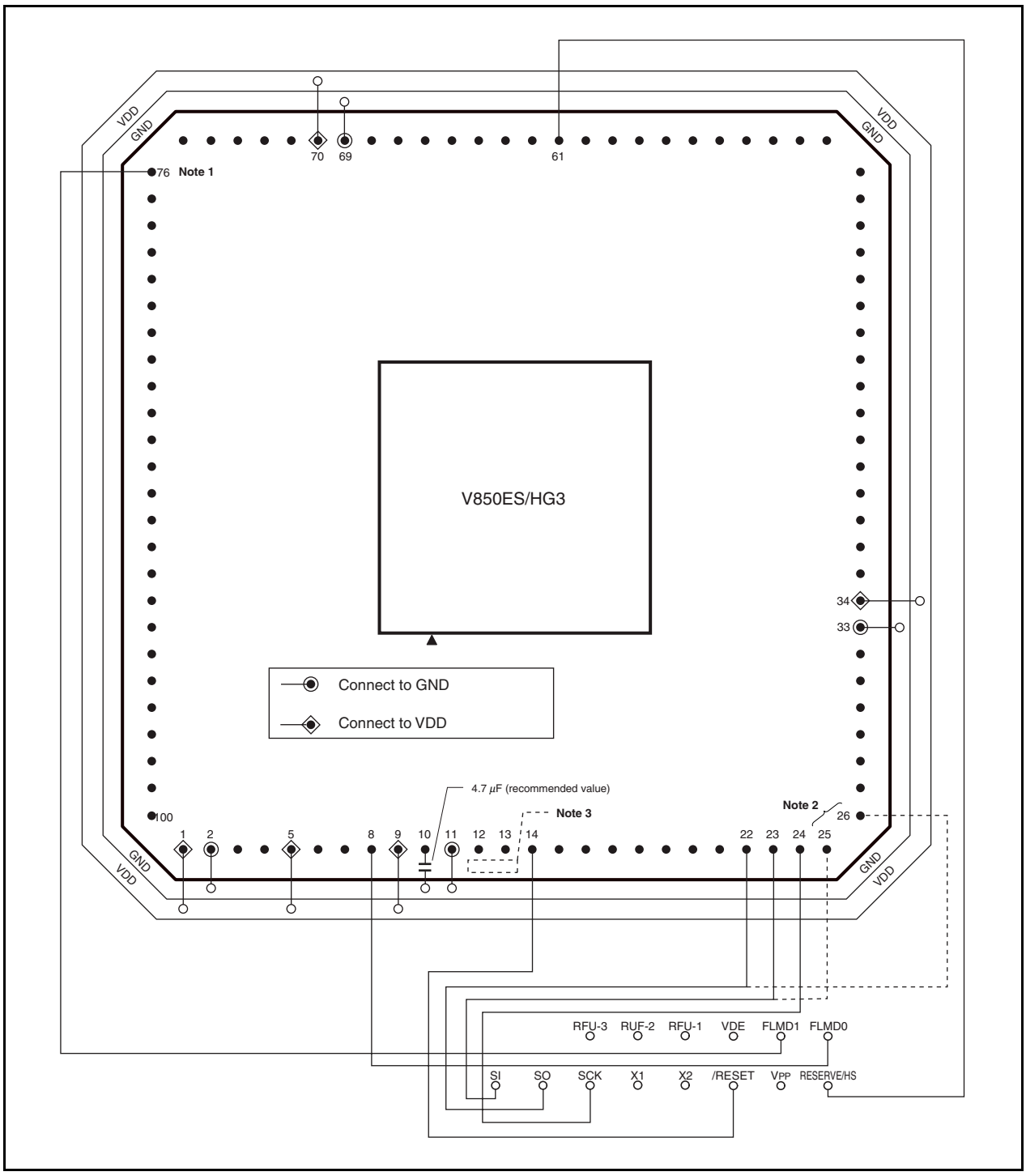


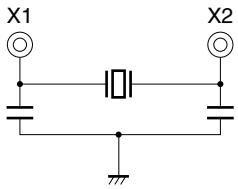
Figure 26-8. Wiring Example of V850ES/HG3 Flash Writing Adapter (FA-100GC-UEU-B) (In CSIB0 + HS Mode) (2/2)

Notes 1. Wire the FLMD1 pin as shown below, or connect it to GND on board via a pull-down resistor.

2. Used when UARTD0 is used.

3. Supply a clock by creating an oscillator on the flash writing adapter (enclosed by broken lines). An example of the oscillator is shown below.

Example:



Caution Do not input a high level to the $\overline{\text{DRST}}$ pin.

Remarks 1. Handle the pins not shown in accordance with the handling of unused pins (see **2.3 Pin I/O Circuit Types, I/O Buffer Power Supplies, and Connection of Unused Pins**).

2. This adapter is for the 100-pin plastic LQFP package.

Figure 26-9. Wiring Example of V850ES/HJ3 Flash Writing Adapter (FA-144GJ-GAE-B) (In CSIB0 + HS Mode) (1/2)

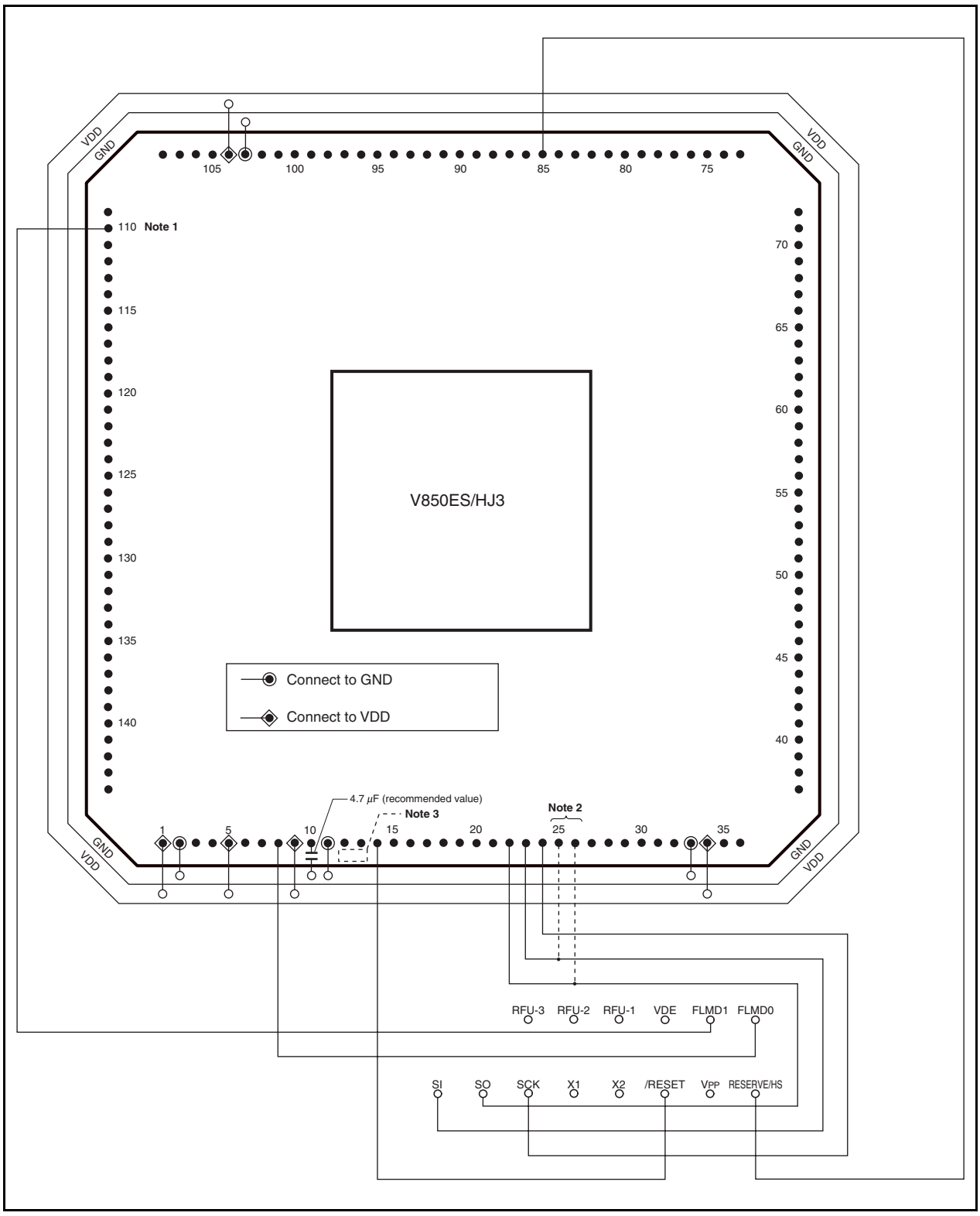


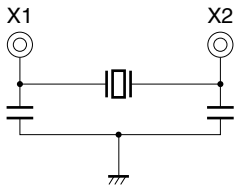
Figure 26-9. Wiring Example of V850ES/HJ3 Flash Writing Adapter (FA-144GJ-GAE-B) (In CSIB0 + HS Mode) (2/2)

Notes 1. Wire the FLMD1 pin as shown below, or connect it to GND on board via a pull-down resistor.

2. Used when UARTD0 is used.

3. Supply a clock by creating an oscillator on the flash writing adapter (enclosed by broken lines). An example of the oscillator is shown below.

Example:



Caution Do not input a high level to the $\overline{\text{DRST}}$ pin.

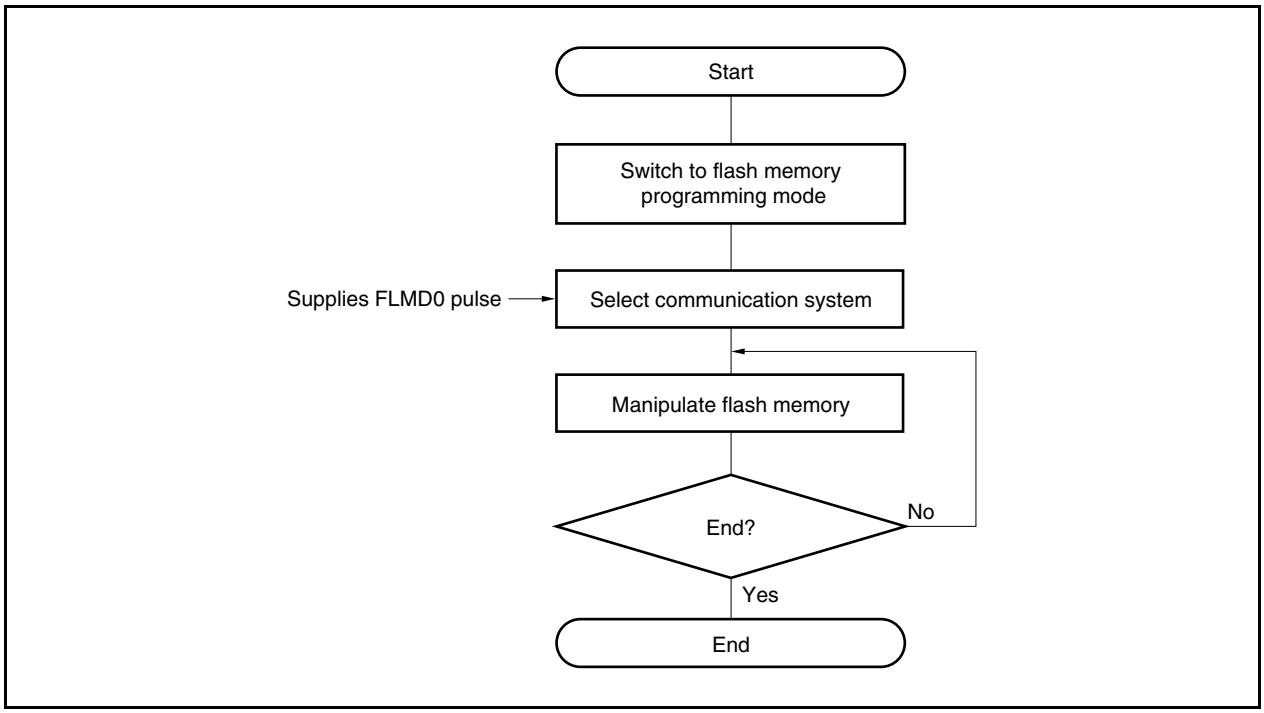
Remarks 1. Handle the pins not shown in accordance with the handling of unused pins (see **2.3 Pin I/O Circuit Types, I/O Buffer Power Supplies, and Connection of Unused Pins**).

2. This adapter is for the 144-pin plastic LQFP package.

26.4.3 Flash memory control

The following shows the procedure for manipulating the flash memory.

Figure 26-10. Procedure for Manipulating Flash Memory

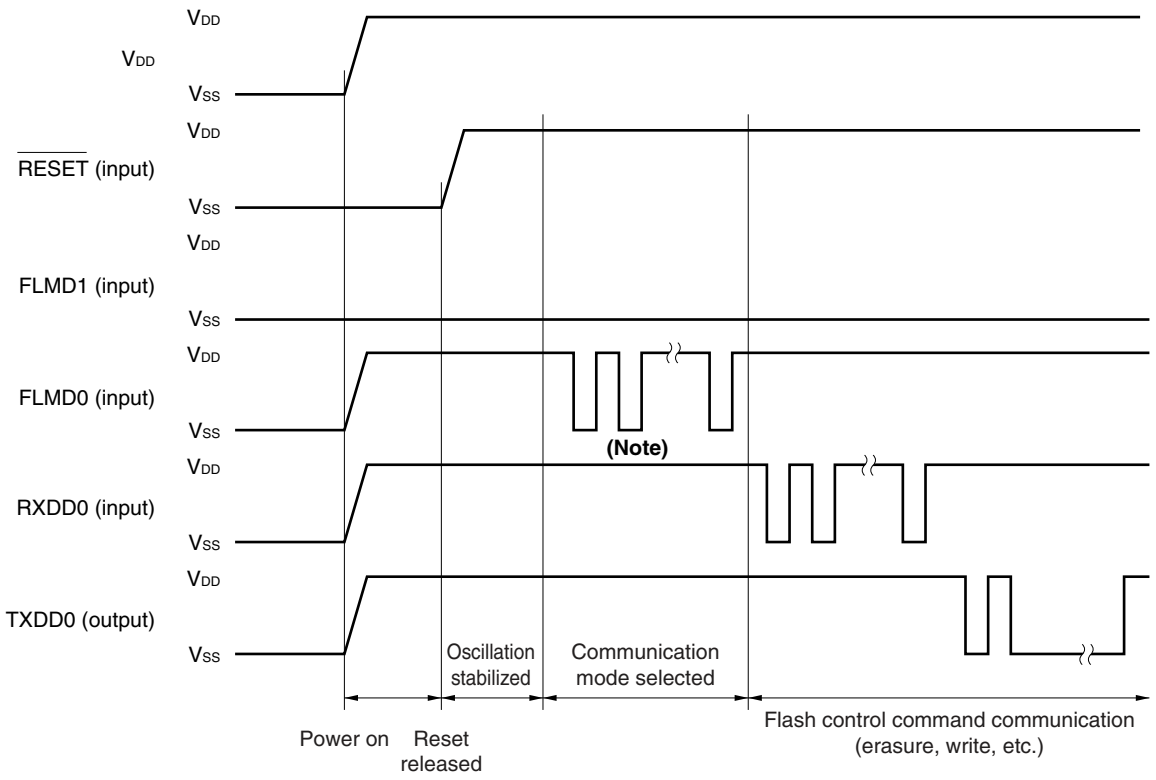


26.4.4 Selection of communication mode

In the V850ES/Hx3, the communication mode is selected by inputting pulses (11 pulses max.) to the FLMD0 pin after switching to the flash memory programming mode. The FLMD0 pulse is generated by the dedicated flash programmer.

The following shows the relationship between the number of pulses and the communication mode.

Figure 26-11. Selection of Communication Mode



Note The number of clocks is as follows depending on the communication mode.

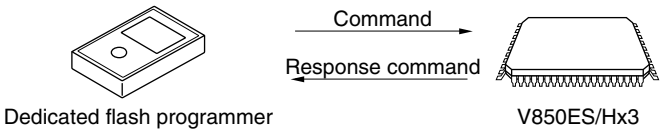
FLMD0 Pulse	Communication Mode	Remarks
0	UARTD0	Communication rate: 9,600 bps (after reset), LSB first
8	CSIB0	V850ES/Hx3 performs slave operation, MSB first
11	CSIB0 + HS	V850ES/Hx3 performs slave operation, MSB first
Other	RFU	Setting prohibited

Caution When UARTD0 is selected, the receive clock is calculated based on the reset command sent from the dedicated flash programmer after receiving the FLMD0 pulse.

26.4.5 Communication commands

The V850ES/Hx3 communicates with the dedicated flash programmer by means of commands. The signals sent from the dedicated flash programmer to the V850ES/Hx3 are called “commands”. The response signals sent from the V850ES/Hx3 to the dedicated flash programmer are called “response commands”.

Figure 26-12. Communication Commands



The following shows the commands for flash memory control in the V850ES/Hx3. All of these commands are issued from the dedicated flash programmer, and the V850ES/Hx3 performs the processing corresponding to the commands.

Table 26-10. Flash Memory Control Commands

Classification	Command Name	Support			Function
		CSIB0	CSIB0 + HS	UARTD0	
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.
Program	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 programmer.
	Checksum command	√	√	√	Reads the checksum in the specified address range.
Read	Read command	√	√	√	Reads the data written to the 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.

26.4.6 Pin connection

When performing on-board writing, mount a connector on the target system to connect to the dedicated flash programmer. Also, incorporate a function on-board to switch from the normal operation mode to the flash memory programming mode.

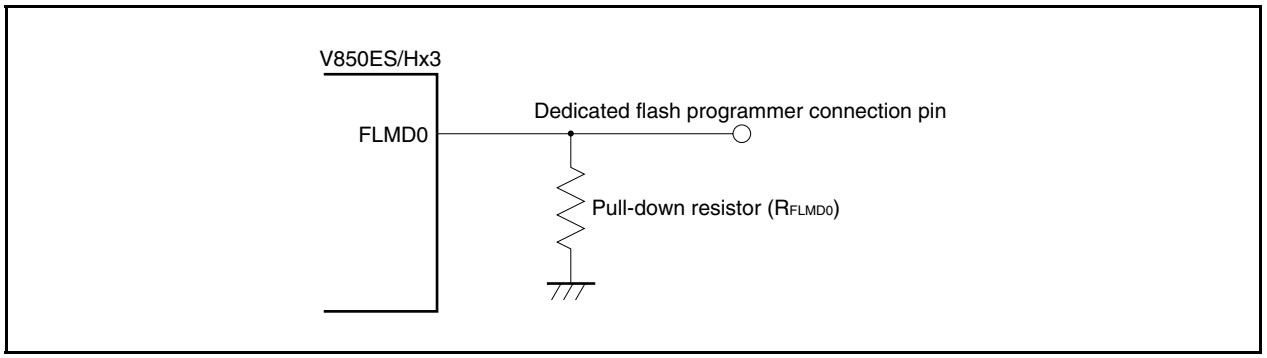
In the flash memory programming mode, all the pins not used for flash memory programming become the same status as that immediately after reset. Therefore, pin handling is required when the external device does not acknowledge the status immediately after a reset.

(1) FLMD0 pin

In the normal operation mode, input a voltage of V_{SS} level to the FLMD0 pin. In the flash memory programming mode, supply a write voltage of V_{DD} level to the FLMD0 pin.

Because the FLMD0 pin serves as a write protection pin in the self programming mode, 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 **26.5.5 (1) FLMD0 pin**.

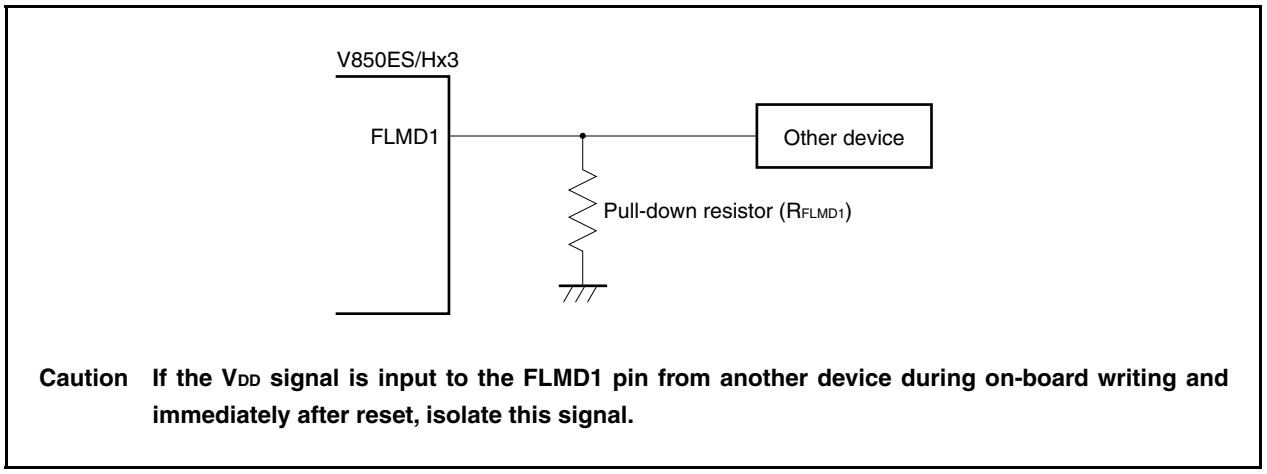
Figure 26-13. FLMD0 Pin Connection Example



(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 26-14. FLMD1 Pin Connection Example



Caution If the V_{DD} signal is input to the FLMD1 pin from another device during on-board writing and immediately after reset, isolate this signal.

Table 26-11. Relationship Between FLMD0 and FLMD1 Pins and Operation Mode When Reset Is Released

FLMD0	FLMD1	Operation Mode
0	Don't care	Normal operation mode
V_{DD}	0	Flash memory programming mode
V_{DD}	V_{DD}	Setting prohibited

(3) Serial interface pin

The following shows the pins used by each serial interface.

Table 26-12. Pins Used by Serial Interfaces

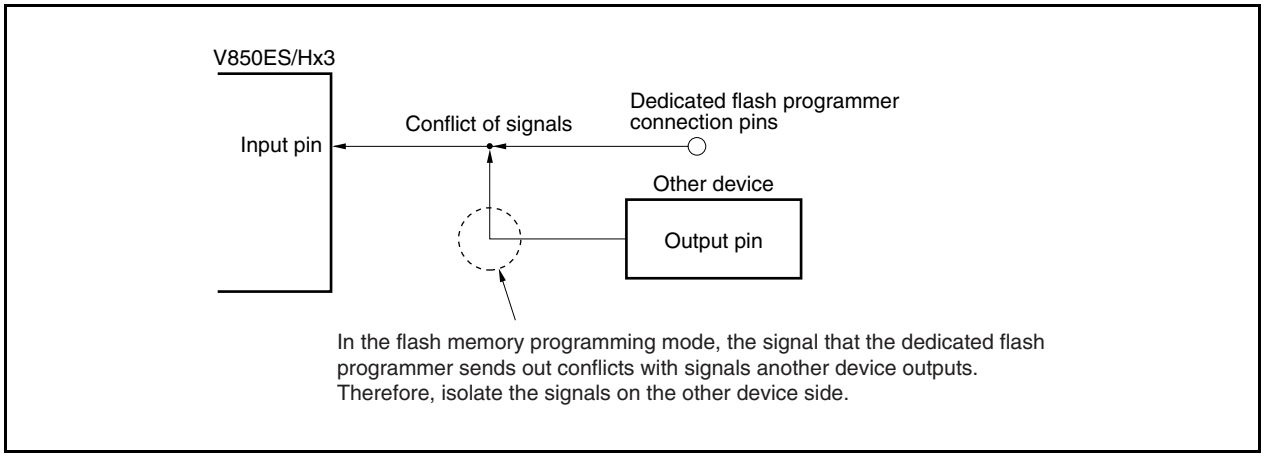
Serial Interface	Pins Used
UARTD0	TXDD0, RXDD0
CSIB0	SOB0, SIB0, $\overline{\text{SCKB0}}$
CSIB0 + HS	SOB0, SIB0, $\overline{\text{SCKB0}}$, PCM0

When connecting a dedicated flash 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 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.

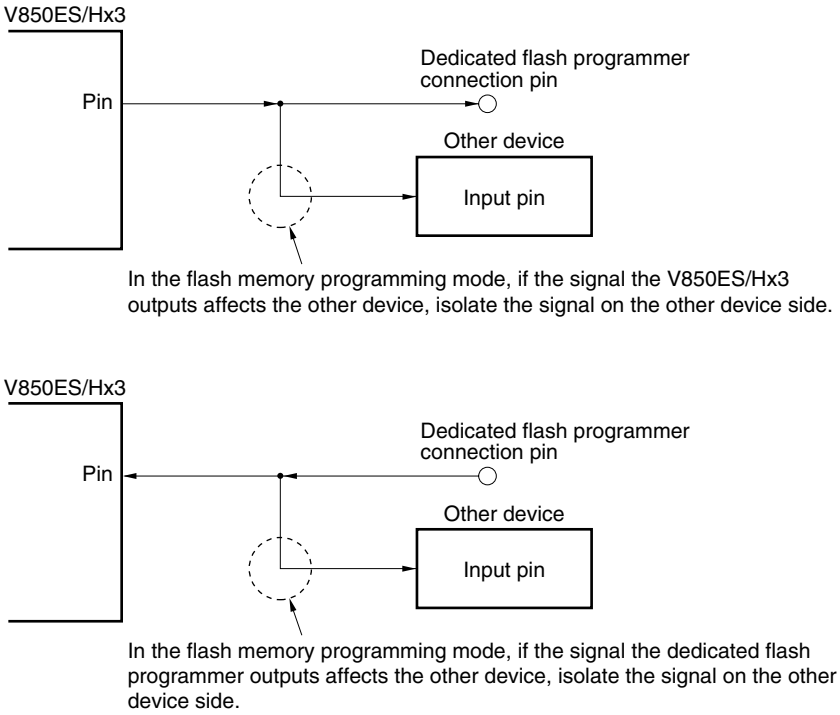
Figure 26-15. Conflict of Signals (Serial Interface Input Pin)



(b) Malfunction of other device

When the dedicated flash 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 26-16. Malfunction of Other Device

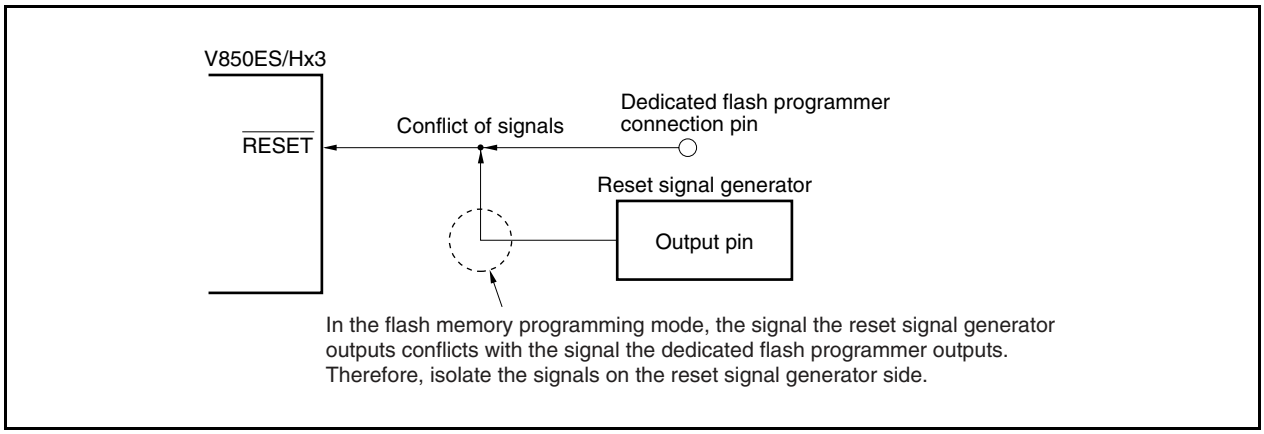


(4) $\overline{\text{RESET}}$ pin

When the reset signals of the dedicated flash 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 programmer.

Figure 26-17. 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}, BV_{DD}, BV_{SS}, AV_{REF0}, AV_{SS}) as in normal operation mode.

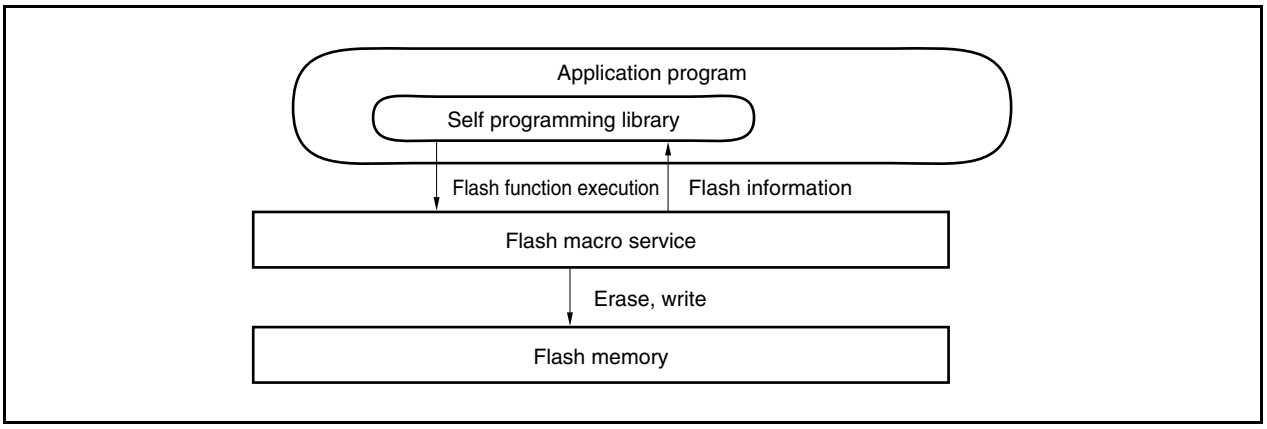
26.5 Rewriting by Self Programming

26.5.1 Overview

The V850ES/Hx3 supports a flash macro service that allows the user program to rewrite the internal flash memory by itself. By using this interface and a self programming library that is used to rewrite the flash memory with a user application program, the flash memory can be rewritten by a user application transferred in advance to the internal RAM or external memory. Consequently, the user program can be upgraded and constant data[Note] can be rewritten in the field.

Note Be sure not to allocate the program code to the block where the constant data to be rewritten is allocated. See **26.2 Memory Configuration** for the block configuration.
For flash self programming, refer to the **V850 Microcontroller Flash Memory Self-Programming Library Type04 User's Manual (U17819E)**.

Figure 26-18. Concept of Self Programming



26.5.2 Features

(1) Secure self programming (boot swap function)

The V850ES/Hx3 supports a boot swap function that can swap physical memory areas from block 0 to the specified block for a consecutive physical memory areas of the same size.

The number of blocks and the areas that are swapped by the boot swap function can be specified by flash self programming.

Remark The number of blocks and the area that can be swapped can be set for each product of the V850ES/Hx3 by flash self programming.
The settable range of each product is shown below.

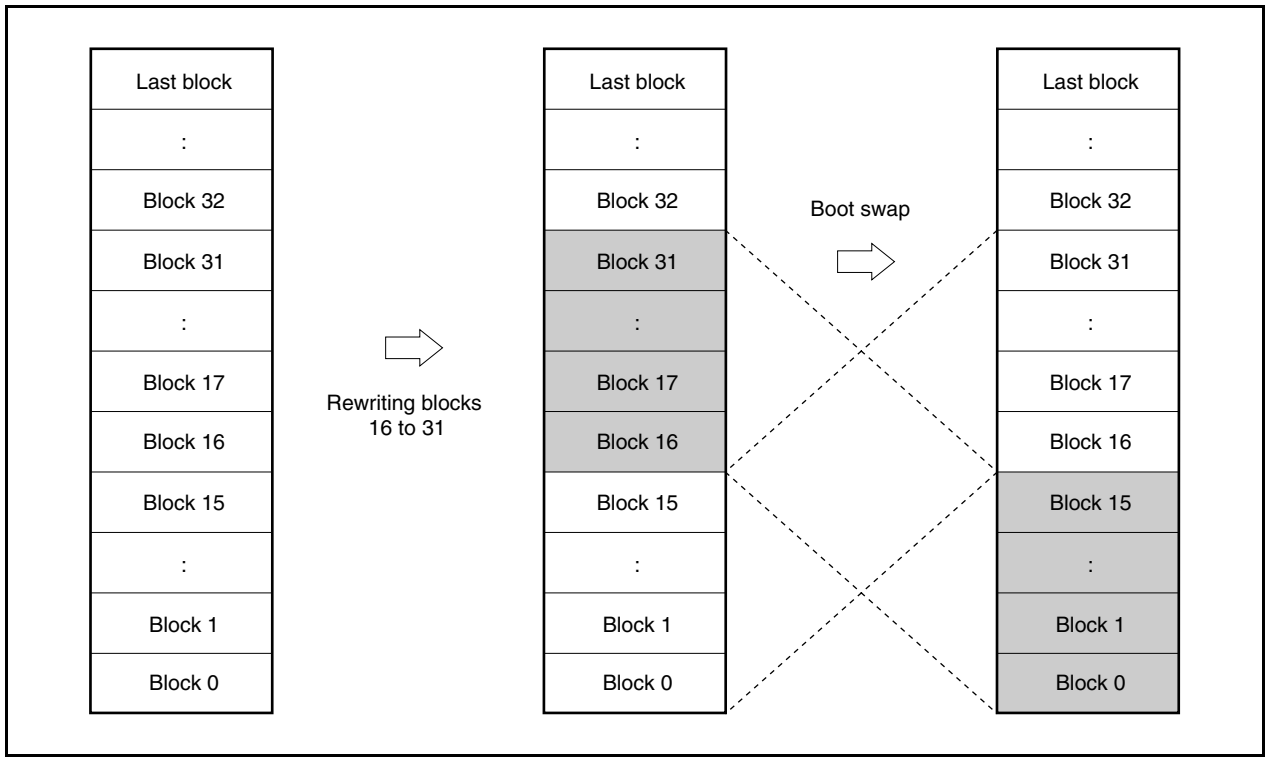
Table 26-13. Boot Swap Cluster Size

Product Name	Boot Swap Unit	Target Block	Target Area
μPD70F3747, μPD70F3750, μPD70F3752, μPD70F3755	8 KB	Blocks 0 to 3	0000 0000H to 0000 1FFFH
		Blocks 4 to 7	0000 2000H to 0000 3FFFH
	16 KB	Blocks 0 to 7	0000 0000H to 0000 3FFFH
		Blocks 8 to 15	0000 4000H to 0000 7FFFH
	32 KB	Blocks 0 to 15	0000 0000H to 0000 7FFFH
		Blocks 16 to 31	0000 8000H to 0000 FFFFH
	64 KB	Blocks 0 to 31	0000 0000H to 0000 FFFFH
		Blocks 32 to 63	0001 0000H to 0001 FFFFH
μPD70F3757	16 KB	Blocks 0 to 3	0000 0000H to 0000 3FFFH
		Blocks 4 to 7	0000 4000H to 0000 7FFFH
	32 KB	Blocks 0 to 7	0000 0000H to 0000 7FFFH
		Blocks 8 to 15	0000 8000H to 0000 FFFFH
	64 KB	Blocks 0 to 15	0000 0000H to 0000 FFFFH
		Blocks 16 to 31	0001 0000H to 0001 FFFFH
	128 KB	Blocks 0 to 31	0000 0000H to 0001 FFFFH
		Blocks 32 to 63	0002 0000H to 0003 FFFFH

Caution The boot swap function can be used only when the reset vector is at the default value (00000000H). It cannot be used if the reset vector is changed from the default value.

If blocks 0 to 15 are specified as a boot swap cluster, for example, write a start program to be rewritten to blocks 16 to 31. After blocks 16 to 31 have been correctly rewritten, the physical memory areas are swapped. If rewriting a new start program fails during rewriting due to instantaneous power interruption or inadvertent reset, the original start program can run in blocks 0 to 15. Therefore, the entire area can be safely rewritten.

Figure 26-19. Rewriting Entire Memory Area: When Swapping Blocks 0 to 15 for Blocks 16 to 31 (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.

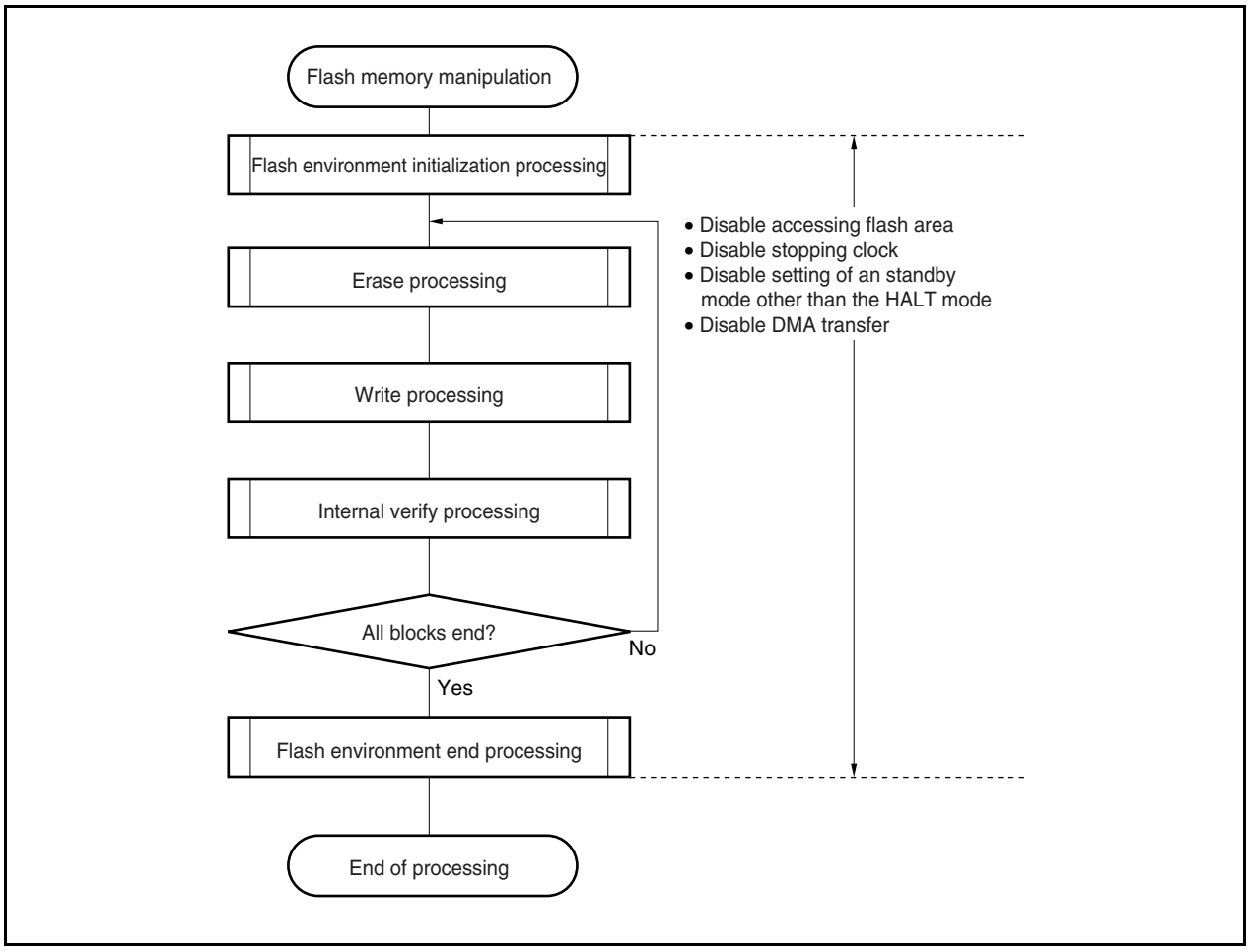
Therefore, in the V850ES/Hx3, to use an interrupt during self-programming, processing transits to the specific address[Note] in the internal RAM. Allocate the branch instruction that transits processing to the user interrupt servicing at the specific address[Note] in the internal RAM.

Note NMI interrupt: Start address of internal RAM
Maskable interrupt: Start address of internal RAM + 4 addresses

26.5.3 Standard self programming flow

The entire processing to rewrite the flash memory by flash self programming is illustrated below.

Figure 26-20. Standard Self Programming Flow



26.5.4 Flash functions

Table 26-14. 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	√
FlashBlockIVerify	Internal verification of block	√
FlashBlockBlankCheck	Blank check of block	√
FlashSetInfo	Flash information setting	√
FlashBootSwap	Boot swap execution	√

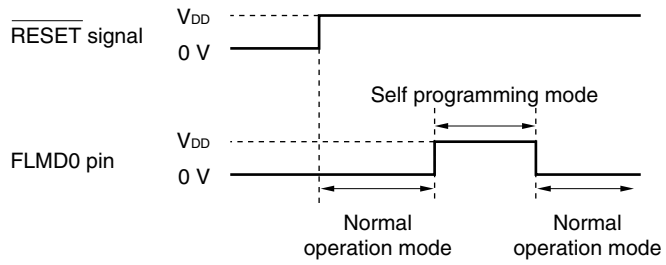
26.5.5 Pin processing

(1) FLMD0 pin

The FLMD0 pin is used to set the operation mode when reset is released and to protect the flash memory from being written during self rewriting. It is therefore necessary to keep the voltage applied to the FLMD0 pin at 0 V when reset is released and a normal operation is executed. It is also necessary to apply a voltage of 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 26-21. Mode Change Timing



Caution Make sure that the FLMD0 pin is at 0 V when reset is released.

26.5.6 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 26-15. Internal Resources Used

Resource Name	Description
Stack area[Note]	An extension of the stack used by the user is used by the library (can be used in both the internal RAM and external RAM).
Library code[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 transits the processing to the user interrupt servicing at the address of the internal RAM start address + 4 addresses in advance.
NMI 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, allocate the branch instruction that transits the processing to the user interrupt servicing at the address of the internal RAM start address in advance.

Note For the resources used, see the **V850 Microcontroller Flash Memory Self-Programming Library Type04 User's Manual (U17819E)**.

The option byte is stored in address 000007BH of the internal flash memory (internal ROM area) as 8-bit data.

When writing a program to the V850ES/Hx3, be sure to set the option data corresponding to the following option in the program at address 000007BH as default data.

The data in this area cannot be rewritten during program execution.

Address: 0000007BH

7	6	5	4	3	2	1	0
0	0	0	0	PLLO	0	0	1

PLLO	Setting of division ratio of output clock from PLL/SSCG
0	$f_{PLL} = f_{PLLO}$ or f_{SSCGO}
1	$f_{PLL} = f_{PLLO}/2$ or $f_{SSCGO}/2$

Caution Be sure to set bits 7 to 4, 2, and 1 to "0". Be sure to set bit 0 to "1".
Also be sure to set 00H to addresses 000007AH and 000007CH to 000007FH.

Remark f_{PLLO}: PLL output clock frequency (See **Figure 6-2**.)
f_{PLL}: Multiplication block output frequency (See **Figure 6-2**.)
f_{SSCGO}: SSCG output clock frequency (See **Figure 6-2**.)

A sample program for using the CA850 is shown below.

[Sample Program]

```
#-----------------------------------------------------------------
# OPTION_BYTES
#-----------------------------------------------------------------
.section "OPTION_BYTES"
.byte 0b00000000 -- 0x7a
.byte 0b00001001 -- 0x7b
.byte 0b00000000 -- 0x7c
.byte 0b00000000 -- 0x7d
.byte 0b00000000 -- 0x7e
.byte 0b00000000 -- 0x7f
```

Caution Be sure to write for 6 bytes in this section. If less than 6 bytes, an error occurs on a linker operation.

Error message: F4112: illegal “OPTION_BYTES” section size.

Remark Set 0x00 to addresses 000007AH and 000007CH to 000007FH.

CHAPTER 28 ON-CHIP DEBUG FUNCTION

The V850ES/Hx3 on-chip debug function can be implemented by the following two methods.

- Using the DCU (debug control unit)
  On-chip debug function is implemented by the on-chip DCU in the V850ES/Hx3, with using the $\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO pins as the debug interface pins.
- Not using the DCU
  On-chip debug function is implemented by MINICUBE2 or the like, using the user resources, instead of the DCU.

The following table shows the features of the two on-chip debug functions.

Table 28-1. On-Chip Debug Function Features

		Debugging Using DCU	Debugging Without Using DCU
Debug interface pins		$\overline{\text{DRST}}$, DCK, DMS, DDI, DDO	• When UARTD0 is used RXDD0, TXDD0 • When CSIB0 is used SIB0, SOB0, $\overline{\text{SCKB0}}$, HS (PCM0)
Securement of 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		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.	NINICUBE2, 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.

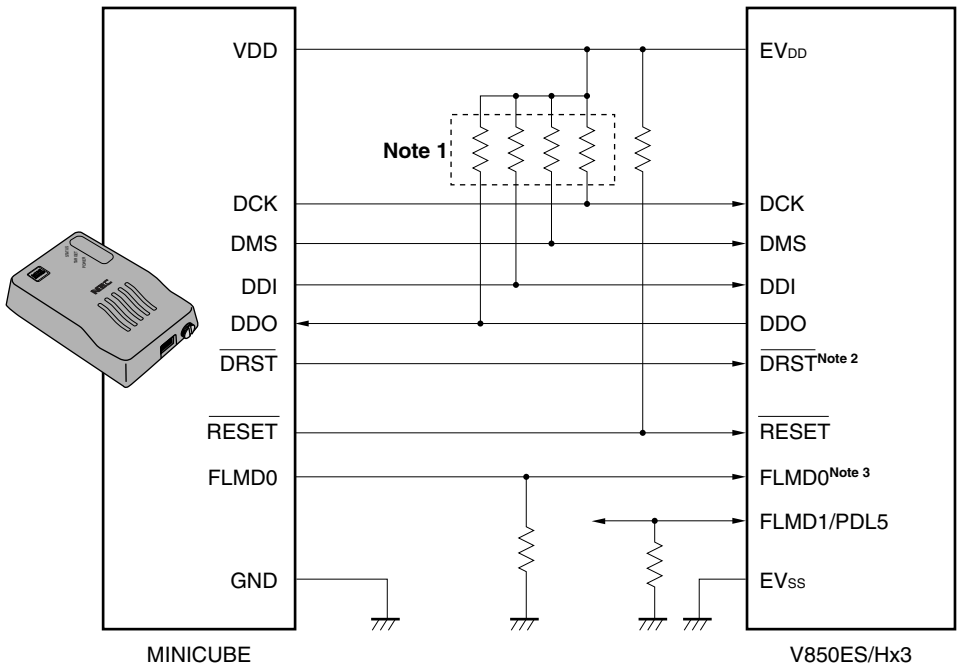
2. This is a function which rewrites RAM contents during program execution.

28.1 Debugging with DCU

Programs can be debugged using the debug interface pins ($\overline{\text{DRST}}$, DCK, DMS, DDI, and DDO) to connect the on-chip debug emulator (MINICUBE).

28.1.1 Connection circuit example

Figure 28-1. Circuit Connection Example When Debug Interface Pins Are Used for Communication Interface



Notes 1. Example of pin connection when MINICUBE is not connected

2. A pull-down resistor is provided on chip.

3. For flash memory rewriting

28.1.2 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 VDD of the target system. If VDD from the target system is not detected, the signals output from MINICUBE ($\overline{DRST}$, DCK, DMS, DDI, FLMD0, and $\overline{RESET}$) go into a high-impedance state.

(7) FLMD0

The flash self programming function is used for the function to download data to the flash memory via the integrated debugger. During flash self programming, the FLMD0 pin must be kept high. In addition, connect a pull-down resistor to the FLMD0 pin.

The FLMD0 pin can be controlled in either of the following two ways.

<1> To control from MINICUBE

Connect the FLMD0 signal of MINICUBE to the FLMD0 pin.

In the normal mode, nothing is driven by MINICUBE (high impedance).

During a break, MINICUBE raises the FLMD0 pin to the high level when the download function of the integrated debugger is executed.

<2> To control from port

Connect any port of the device to the FLMD0 pin.

The same port as the one used by the user program to realize the flash self programming function may be used.

On the console of the integrated debugger, make a setting to raise the port pin to high level before executing the download function, or lower the port pin after executing the download function.

For details, refer to the **ID850QB Ver. 3.40 Integrated Debugger Operation User's Manual (U18604E)**.

(8) $\overline{RESET}$

This is a system reset input pin. If the $\overline{DRST}$ pin is made invalid by the value of the OCDM0 bit of the OCDM register set by the user program, on-chip debugging cannot be executed. Therefore, reset is effected by MINICUBE, using the $\overline{RESET}$ pin, to make the $\overline{DRST}$ pin valid (initialization).

28.1.3 Maskable functions

Reset, NMI, INTWDT2, $\overline{\text{WAIT}}$, and $\overline{\text{HLDRQ}}$ (V850ES/HJ3 only) signals can be masked.

The maskable functions with the debugger (ID850QB) and the corresponding V850ES/Hx3 functions are listed below.

Table 28-2. Maskable Functions

Maskable Functions with ID850QB	Corresponding V850ES/Hx3 Functions
NMI0	NMI pin input
NMI1	Non-maskable interrupt request signal (INTWDT2) generation
NMI2	–
HLDRQ	$\overline{\text{HLDRQ}}$ pin input (V850ES/HJ3 only)
RESET	Reset signal generation by $\overline{\text{RESET}}$ pin input, low-voltage detector, clock monitor, or watchdog timer (WDT2) overflow
STOP	–
WAIT	$\overline{\text{WAIT}}$ pin input (V850ES/HJ3 only)
DBINT	–

28.1.4 Register

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

The OCDM 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.

After POC reset, the default value of the OCDM0 bit is “0” and the normal operation mode is set. To set the on-chip debug mode, therefore, the value of the OCDM0 bit must be changed to “1” by pin reset. If POC reset is generated during on-chip debugging, communication with MINICUBE is stopped. Therefore, reset by POC cannot be emulated.

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 $\overline{\text{DRST}}$ pin.

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

After reset: 01H[Note] R/W Address: FFFFF9FCH

	7	6	5	4	3	2	1	<0>
OCDM	0	0	0	0	0	0	0	OCDM0

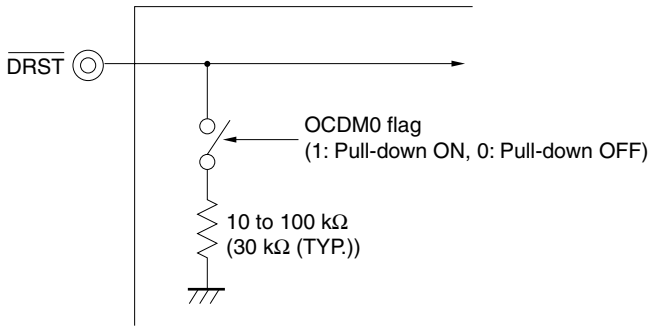
OCDM0	Operation mode
0	Selects normal operation mode (in which a pin that functions alternately as on-chip debug function pin is used as a port/peripheral function pin) and disconnects the on-chip pull-down resistor of the P05/INTP2/$\overline{\text{DRST}}$ pin.
1	When $\overline{\text{DRST}}$ pin is 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 $\overline{\text{DRST}}$ 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.
Reset by power-on clear sets this register to 00H.
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 $\overline{\text{DRST}}$, DDI, DDO, DCK, and DMS pins not as on-chip debug pins but as port pins after external reset, any 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 $\overline{\text{DRST}}$ pin has an on-chip pull-down resistor. This resistor is disconnected when the OCDM0 flag is cleared to 0.



28.1.5 Operation

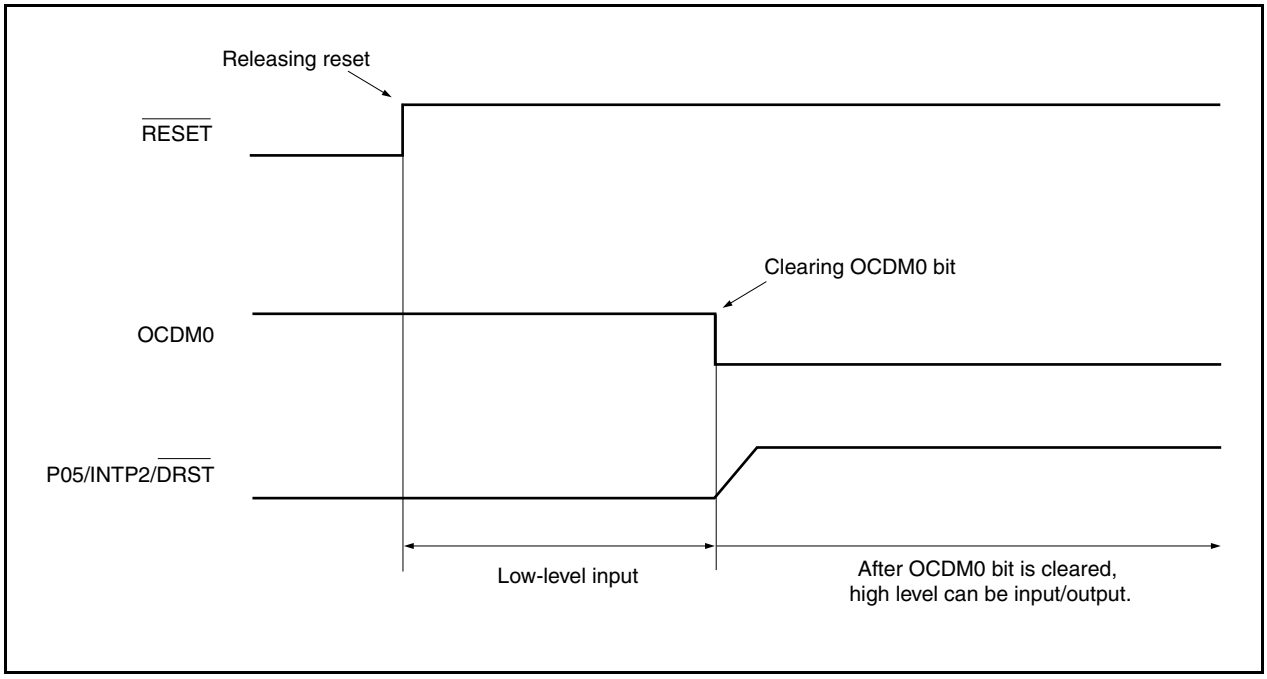
The on-chip debug function is made invalid under the conditions shown in the table below.

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 28-2. Timing When On-Chip Debug Function Is Not Used



28.1.6 Cautions

(1) If a reset signal is input (from the target system or a reset signal from an internal reset source) during RUN (program execution), the break function may malfunction.

(2) Even if the reset signal is masked by the mask function, the I/O buffer (port pin) may be reset if a reset signal is input from a pin.

(3) Pin reset during a break is masked and the CPU and peripheral I/O are not reset. If pin reset or internal reset is generated as soon as the flash memory is rewritten by 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.

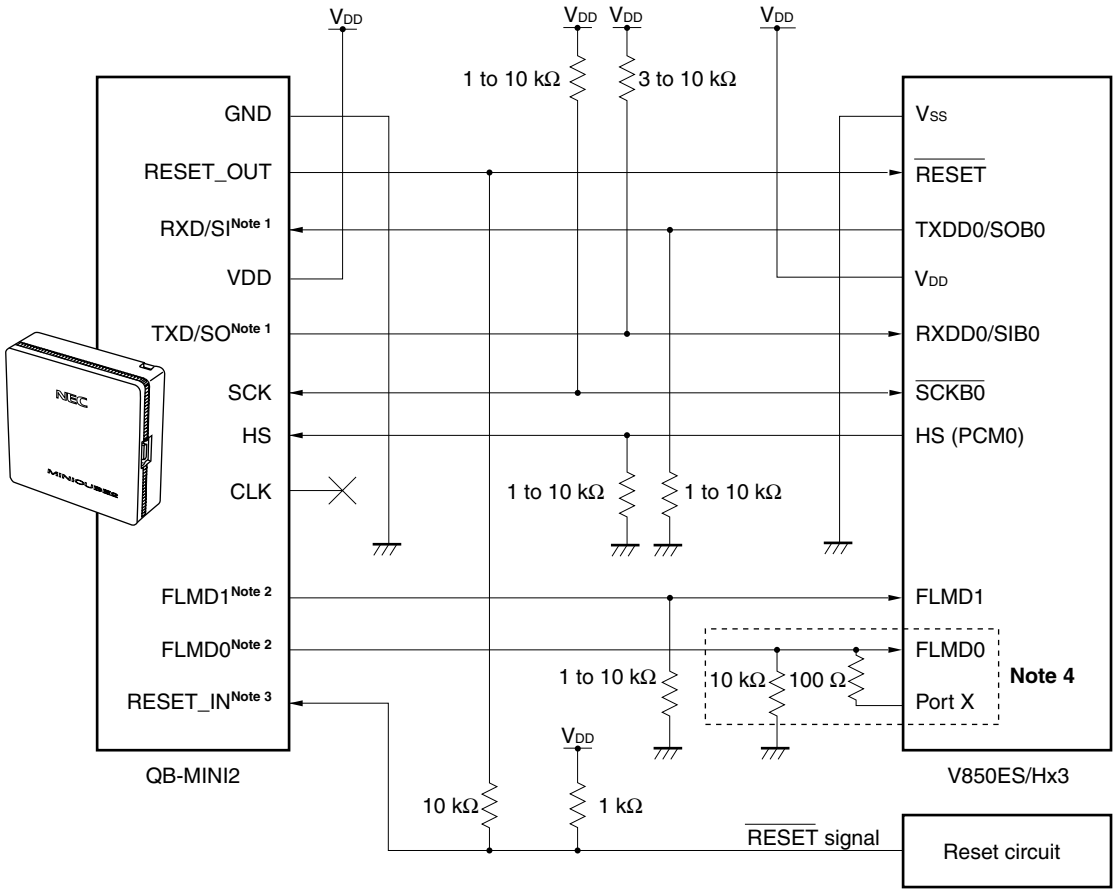
(4) In the on-chip debug mode, the DDO pin is forcibly set to the high-level output.

28.2 Debugging Without Using DCU

The following describes how to implement an on-chip debug function using MINICUBE2 with pins for UARTD0 (RXDD0 and TXDD0) or pins for CSIB0 (SIB0, SOB0, $\overline{\text{SCKB0}}$, and HS (PCM0)) as debug interfaces, without using the DCU.

28.2.1 Circuit connection examples

Figure 28-3. Circuit Connection Example When UARTD0/CSIB0 Is Used for Communication Interface



Notes 1. Connect TXDD0/SOB0 (transmit side) of the V850ES/Hx3 to RXD/SI (receive side) of the target connector, and TXD/SO (transmit side) of the target connector to RXDD0/SIB0 (receive side) of the V850ES/Hx3.

2. The V850ES/Hx3-side pin connected to this pin (FLMD0, FLMD1) can be used as an alternate-function pin other than while the memory is rewritten during a break in debugging, because this pin is in Hi-Z state.

3. This connection is designed assuming that the $\overline{\text{RESET}}$ signal is output from the N-ch open-drain buffer (output resistance: 100 Ω or less).

4. The circuit enclosed by broken lines is designed for flash self programming, which controls the FLMD0 pin via ports. Use the port for inputting or outputting the high level. When flash self programming is not performed, a pull-down resistance for the FLMD0 pin can be within 1 to 10 kΩ.

Remark See **Table 28-3** for pins used when UARTD0 or CSIB0 is used for communication interface.

Table 28-3. Wiring Between V850ES/Hx3 and MINICUBE2

Pin Configuration of MINICUBE2 (QB-MINI2)			With CSIB0-HS	With UARTD0
Signal Name	I/O	Pin Function	Pin Name	Pin Name
SI/RxD	Input	Pin to receive commands and data from V850ES/Hx3	P41/SOB0	P30/TXDD0
SO/TxD	Output	Pin to transmit commands and data to V850ES/Hx3	P40/SIB0	P31/RXDD0
SCK	Output	Clock output pin for 3-wire serial communication	P42/$\overline{\text{SCKB0}}$	Not needed
CLK	Output	Not used	Not needed	Not needed
RESET_OUT	Output	Reset output pin to V850ES/Hx3	$\overline{\text{RESET}}$	$\overline{\text{RESET}}$
FLMD0	Output	Output pin to set V850ES/Hx3 to debug mode or programming mode	FLMD0	FLMD0
FLMD1	Output	Output pin to set programming mode	PDL5/FLMD1	PDL5/FLMD1
HS	Input	Handshake signal for CSI0 + HS communication	PCM0/$\overline{\text{WAIT}}$	Not needed
GND	–	Ground	V_{SS}	V_{SS}
			AV_{SS}	AV_{SS}
			EV_{SS}	EV_{SS}
			BV_{SS}	BV_{SS}
RESET_IN	Input	Reset input pin on the target system		

28.2.2 Maskable functions

Only reset signals can be masked.

The maskable functions with the debugger (ID850QB) and the corresponding V850ES/Hx3 functions are listed below.

Table 28-4. Maskable Functions

Maskable Functions with ID850QB	Corresponding V850ES/Hx3 Functions
NMI0	–
NMI1	–
NMI2	–
HLDRQ	–
RESET	Reset signal generation by $\overline{\text{RESET}}$ pin input
STOP	–
WAIT	–
DBINT	–

28.2.3 Securement of user resources

The user must prepare the following 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.

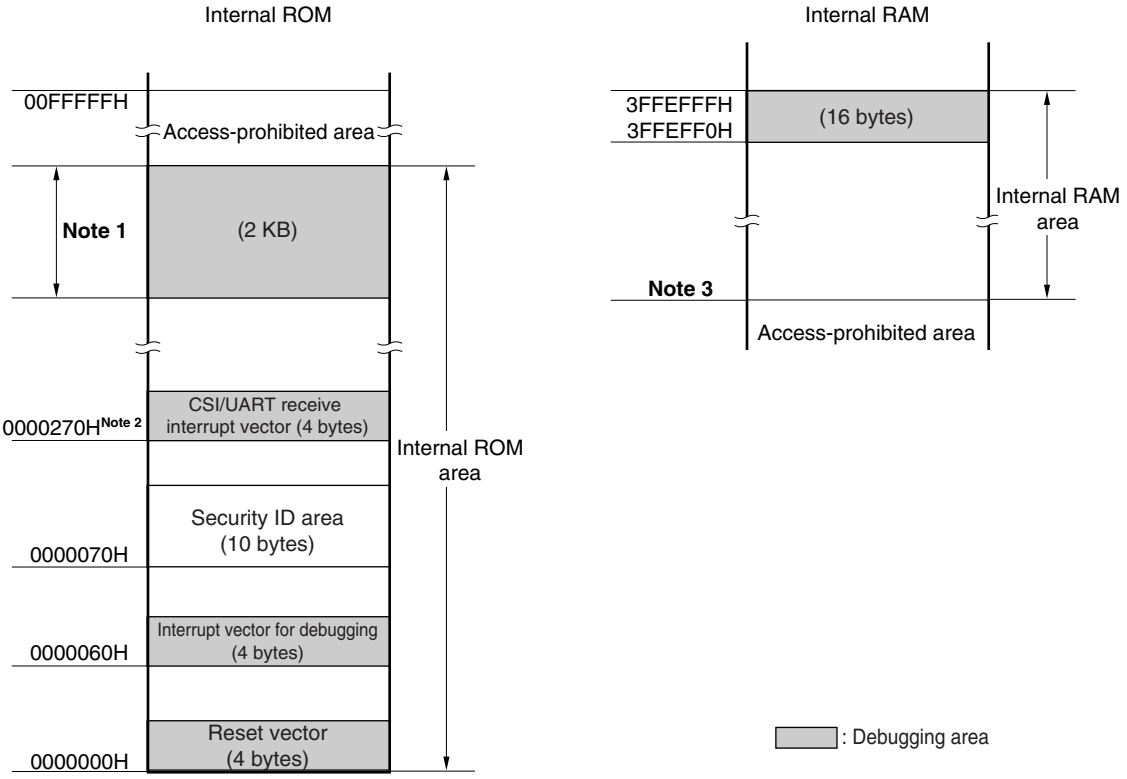
(1) Securement of memory space

The shaded portions in Figure 28-4 are the areas reserved for placing the debug monitor program, so user programs and data cannot be allocated in these spaces. These spaces must be secured so as not to be used by the user program.

(2) Security ID setting

The ID code must be embedded in the area between 0000070H and 0000079H in Figure 28-4, to prevent the memory from being read by an unauthorized person. For details, see **28.3 ROM Security Function**.

Figure 28-4. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address values vary depending on the product.

	Internal ROM size	Address value
μPD70F3747	128 KB	001F800H to 001FFFFH
μPD70F3750, μPD70F3752, μPD70F3755	256 KB	003F800H to 003FFFFH
μPD70F3757	512 KB	007F800H to 0007FFFFH

2. This is the address when CSIB0 is used. It starts at 00002C0H when UARTD0 is used.

3. Address values vary depending on the product.

	Internal RAM size	Address value
μPD70F3747	8 KB	3FFD000H
μPD70F3750, μPD70F3752, μPD70F3755	16 KB	3FFB000H
μPD70F3757	32 KB	3FF7000H

(3) Reset vector

A reset vector includes the jump instruction for the debug monitor program.

[How to secure areas]

It is not necessary to secure this area intentionally. When downloading a program, however, the debugger rewrites the reset vector in accordance with the following cases. If the rewritten pattern does not match the following cases, the debugger generates an error (F0C34 when using the ID850QB).

(a) When two nop instructions are placed in succession from address 0

Before rewriting		After rewriting
0x0 nop	→	Jumps to debug monitor program at 0x0
0x2 nop		0x4 xxxx
0x4 xxxx		

(b) When two 0xFFFF are successively placed from address 0 (already erased device)

Before rewriting		After rewriting
0x0 0xFFFF	→	Jumps to debug monitor program at 0x0
0x2 0xFFFF		0x4 xxxx
0x4 xxxx		

(c) The *jr* instruction is placed at address 0 (when using CA850)

Before rewriting		After rewriting
0x0 jr disp22	→	Jumps to debug monitor program at 0x0
		0x4 jr disp22 - 4

(d) *mov32* and *jmp* are placed in succession from address 0 (when using IAR compiler ICCV850)

Before rewriting		After rewriting
0x0 mov imm32,reg1	→	Jumps to debug monitor program at 0x0
0x6 jmp [reg1]		0x4 mov imm32,reg1
		0xa jmp [reg1]

(e) The jump instruction for the debug monitor program is placed at address 0

Before rewriting		After rewriting
Jumps to debug monitor program at 0x0	→	No change

(4) Securement of area for debug monitor program

The shaded portions in Figure 28-4 are the areas where the debug monitor program is allocated. The monitor program performs initialization processing for debug communication interface and RUN or break processing for the CPU. The internal ROM area must be filled with 0xFF. This area must not be rewritten by the user program.

[How to secure areas]

It is not necessarily required to secure this area if the user program does not use this area.

To avoid problems that may occur during the debugger startup, however, it is recommended to secure this area in advance, using the compiler.

The following shows examples for securing the area, using the NEC Electronics compiler CA850. Add the assemble source file and link directive code, as shown below.

- Assemble source (Add the following code as an assemble source file.)

```
-- Secures 2 KB space for monitor ROM section
.section "MonitorROM", const
.space   0x800, 0xff

-- Secures interrupt vector for debugging
.section "DBG0"
.space   4, 0xff

-- Secures interrupt vector for serial communication
-- Change the section name according to the serial communication mode used
.section "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 the internal ROM has 256 KB (end address is 003FFFFH) and internal RAM has 16 KB (end address is 3FFEFFFH).

```
MROMSEG       : !LOAD ?R V0x03f800{
              MonitorROM    = $PROGBITS   ?A MonitorROM;
};

MRAMSEG       : !LOAD ?RW V0x03ffeff0{
              MonitorRAM    = $NOBITS     ?AW MonitorRAM;
};
```

(5) Securement of communication serial interface

UARTD0 or CSIB0 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, communication serial interface must be secured in the user program.

[How to secure communication serial interface]

- On-chip debug mode register (OCDM)

  For the on-chip debug function using the UARTD0 or CSIB0, set the OCDM register functions 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 or UARTD0 in the user program.

- Interrupt mask registers

  When CSIB0 is used, do not mask the transmit end interrupt (INTCB0R). When UARTD0 is used, do not mask the receive end interrupt (INTUD0R).

(a) When CSIB0 is used

	7	6	5	4	3	2	1	0
CB0RIC	×	0	×	×	×	×	×	×

(b) When UARTD0 is used

	7	6	5	4	3	2	1	0
UD0RIC	×	0	×	×	×	×	×	×

Remark ×: don't care

- Port registers when UARTD0 is used

  When UARTD0 is used, port registers are set to make the TXDD0 and RXDD0 pins valid by the debug monitor program. Do not change the following register settings with the user program during debugging. (The same value can be overwritten.)

	7	6	5	4	3	2	1	0
PMC3L	×	×	×	×	×	×	1	1

Remark ×: don't care

- Port registers when CSIB0 is used

  When CSIB0 is used, port registers are set to make the SIB0, SOB0, $\overline{\text{SCKB0}}$, and HS (PCM0) pins valid by the debug monitor program. Do not change the following register settings with the user program during debugging. (The same value can be overwritten.)

(a) SIB0, SOB0, and $\overline{\text{SCKB0}}$ settings

	7	6	5	4	3	2	1	0
PMC4	×	×	×	×	×	1	1	1
PFC4	×	×	×	×	×	×	0	0
PFCE4	×	×	×	×	×	×	0	0

(μPD70F3757 only)

(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 perform port register settings in 8-bit units, the user program can usually use read-modify-write. If an interrupt for debugging occurs before writing, however, an unexpected operation may be performed.

Remark ×: don't care

28.2.4 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 UARTD0, 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 UARTD0, and the main clock has been stopped
- Mode for communication between MINICUBE2 and the target device is UARTD0, 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
- Mode for communication between MINICUBE2 and the target device is UARTD0, and the main clock has been supplied.

(5) Writing to peripheral I/O registers that requires a specific sequence, using DMM function

Peripheral I/O registers that requires a specific sequence cannot be written with the DMM function.

(6) Flash self programming

If a space where the debug monitor program is allocated is rewritten by flash self programming, the debugger can no longer operate normally.

28.3 ROM Security Function

28.3.1 Security ID

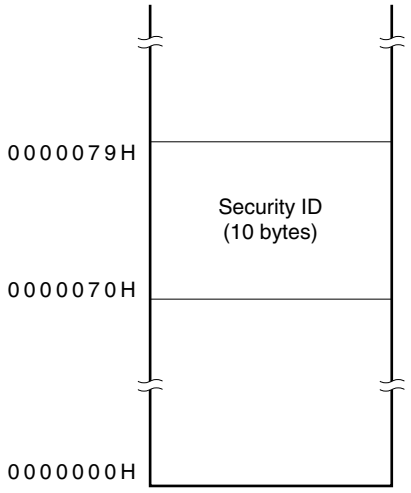
The flash memory versions of the V850ES/Hx3 perform authentication using a 10-byte ID code to prevent the contents of the flash memory from being read by an unauthorized person during on-chip debugging by the on-chip debug emulator.

Set the ID code in the 10-byte on-chip flash memory area from 0000070H to 0000079H to allow the debugger perform ID authentication.

If the IDs match, the security is released and reading flash memory and using the on-chip debug emulator are enabled.

- Set the 10-byte ID code to 0000070H to 0000079H.
- Bit 7 of 0000079H is the on-chip debug emulator enable flag.
  (0: Disable, 1: Enable)
- When the on-chip debug emulator is started, the debugger requests ID input. When the ID code input on the debugger and the ID code set in 0000070H to 0000079H match, the debugger starts.
- Debugging cannot be performed if the on-chip debug emulator enable flag is 0, even if the ID codes match.

Figure 28-5. Security ID Area



Caution After the flash memory is erased, 1 is written to the entire area.

28.3.2 Setting

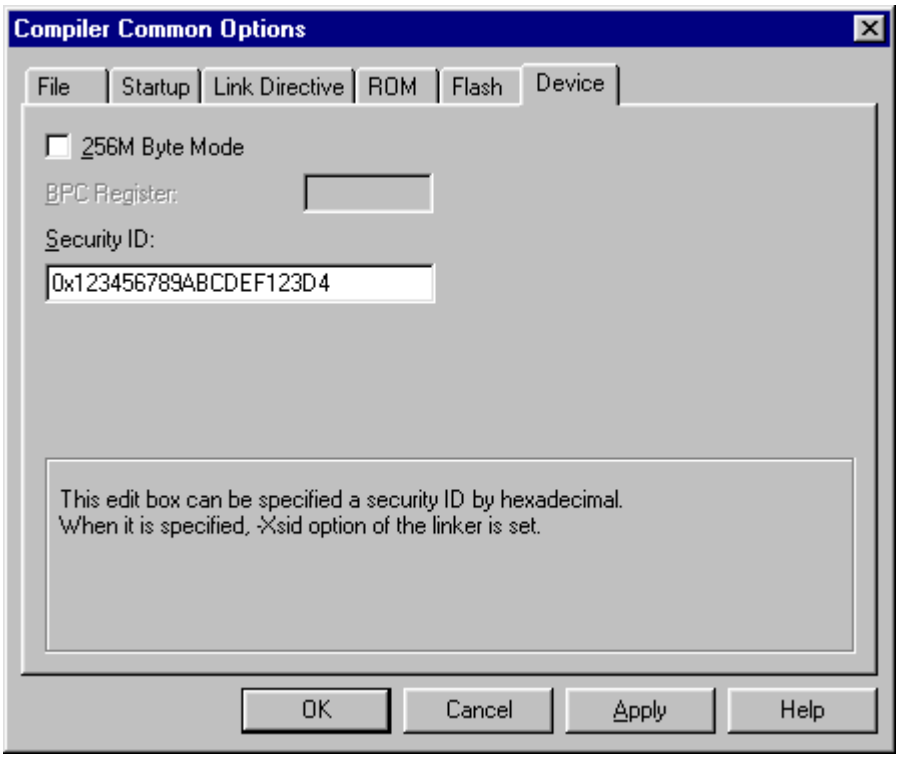
The following shows how to set the ID code as shown in Table 28-5.

When the ID code is set as shown in Table 28-5, the ID code input in the configuration dialog box of the ID850QB is “123456789ABCDEF123D4” (the ID code is case-insensitive).

Table 28-5. 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 by using the PM+ compiler common option setting.



CHAPTER 29 ELECTRICAL SPECIFICATIONS (V850ES/HE3)

29.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}	$V_{DD} = EV_{DD}$	−0.5 to +6.5	V
	EV_{DD}	$V_{DD} = EV_{DD}$	−0.5 to +6.5	V
	AV_{REF0}		−0.5 to +6.5	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}	P00 to P06, P30 to P35, P40 to P42, P50 to P55, P90, P91, P96 to P99, P913 to P915, PCM0, PCM1, PDL0 to PDL7, $\overline{RESET}$, FLMD0	−0.5 to EV_{DD} + 0.5[Note]	V
	V_{I3}	X1, X2, XT1, XT2	−0.5 to V_{RO} + 0.5	V
Analog input voltage	V_{IAN}	P70 to P79	−0.5 to AV_{REF0} + 0.5[Note]	V

Note Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

Cautions 1. Do not directly connect the output (or I/O) pins of IC products to each other, or to V_{DD}, 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 and AC characteristics represent the quality assurance range during normal operation.

Remark Unless specified otherwise, the characteristics 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}	P00 to P06, P30 to P35, P40 to P42, P50 to P55, P90, P91, P96 to P99, P913 to P915, PCM0, PCM1, PDL0 to PDL7	Per pin	4	mA
			Total of all pins	50	mA
		P70 to P79	Per pin	4	mA
			Total of all pins	20	mA
Output current, high	I_{OH}	P00 to P06, P30 to P35, P40 to P42, P50 to P55, P90, P91, P96 to P99, P913 to P915, PCM0, PCM1, PDL0 to PDL7	Per pin	−4	mA
			Total of all pins	−50	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	°C
		Flash memory programming mode			
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 and AC characteristics represent the quality assurance range during normal operation.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

29.2 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

29.3 Operating Conditions

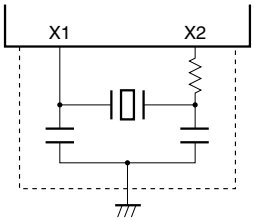
(T_A = –40 to +85°C, V_{DD} = EV_{DD}, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C = 4.7 μF)

Internal System Clock Frequency	Conditions	Supply Voltage		Unit
		V_{DD}, EV_{DD}	AV_{REF0}	
4 MHz ≤ f_{XX} ≤ 32 MHz		4.0 to 5.5	4.0 to 5.5	V
4 MHz ≤ f_{XX} ≤ 20 MHz	AD converter operating	3.7 to 5.5	4.0 to 5.5	V
	AD converter stop	3.7 to 5.5	3.7 to 5.5	V
f_{XT} = 32.768 kHz		3.7 to 5.5	3.7 to 5.5	V
f_{RL} = 240 kHz (TYP.)		3.7 to 5.5	3.7 to 5.5	V
f_{RH} = 8 MHz (TYP.)	AD converter operating	3.7 to 5.5	4.0 to 5.5	V
	AD converter stop	3.7 to 5.5	3.7 to 5.5	V

29.4 Oscillator Characteristics

29.4.1 Main clock oscillator characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions		MIN.	TYP.	MAX.	Unit
Ceramic resonator/ crystal resonator		Oscillation frequency (f_X)[Note 1]			4		16	MHz
		Oscillation stabilization time[Note 2]	After STOP mode release	PLL stopped	54[Note 3]	**Note 4**		μs
				PLL operating	1600[Note 5]	**Note 4**		μs
				SSCG operating	2000[Note 6]	**Note 4**		μs
			After IDLE2 mode release	PLL stopped	54[Note 3]	**Note 4**		μs
				PLL operating	800[Note 5]	**Note 4**		μs
				SSCG operating	1000[Note 6]	**Note 4**		μs

Notes 1. Indicates only oscillator characteristics.

2. Time required from start of oscillation until the resonator stabilizes.

3. Time required to stabilize access to the internal flash memory.

4. The value differs depending on the OSTS register settings.

5. PLL lockup time

6. SSCG lockup time

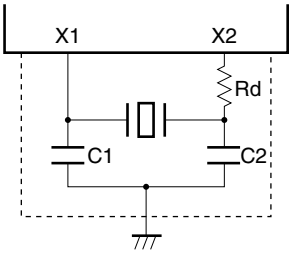
Cautions 1. When using the main clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **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. When the main clock is stopped and the subclock is operating, wait until the oscillation stabilization time has been secured by the program before switching back to the main clock.

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.

<R> **(i) KYOCERA KINSEKI CORPORATION: Crystal resonator (T_A = –10 to +70°C)**

Type	Circuit Example	Part Number	Oscillation Frequency f_X (MHz)	Recommended Circuit Constant		
				C1 (pF)	C2 (pF)	Rd (kΩ)
Lead		HC49SFWB06000D0PESZZ	6.000	8	8	0
		HC49SFWB08000D0PESZZ	8.000	8	8	0
		HC49SFWB10000D0PESZZ	10.000	8	8	0
Surface mounting		HC49GFWB06000D0PESZZ	6.000	8	8	0
		CX1255GB06000D0PESZZ	6.000	8	8	0
		CX8045GB06000D0PESZZ	6.000	8	8	0
		HC49GFWB08000D0PESZZ	8.000	8	8	0
		CX1255GB08000D0PESZZ	8.000	8	8	0
		CX8045GB08000D0PESZZ	8.000	8	8	0
		CX5032GB08000D0PESZZ	8.000	8	8	0
		HC49GFWB10000D0PESZZ	10.000	8	8	0
		CX1255GB10000D0PESZZ	10.000	8	8	0
		CX8045GB10000D0PESZZ	10.000	8	8	0
		CX5032GB10000D0PESZZ	10.000	8	8	0

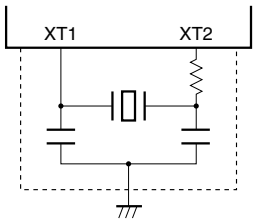
Caution This oscillator constant is a reference value based on evaluation under a specific environment by the resonator manufacturer.

If optimization of oscillator characteristics is necessary in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

The oscillation voltage and oscillation frequency indicate only oscillator characteristics. Use the V850ES/HE3 so that the internal operating conditions are within the specifications of the DC and AC characteristics.

29.4.2 Subclock oscillator characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	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. Indicates only oscillator characteristics. For the CPU operation clock, see **29.7 AC Characteristics**.

2. Time required from when V_{DD} reaches oscillation voltage range (MIN.: 3.3 V) to when the oscillation 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 current 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.

29.4.3 PLL characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
<R> Input frequency	f_X		6		12	MHz
Output frequency	f_{XX}		12		32	MHz
Lock time	t_{PLL}	After V_{DD} reaches MIN.: 3.3 V			800	μs

29.4.4 SSCG characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input frequency	f_X		4		16	MHz
Output frequency[Note]	f_{XX}		12		32	MHz
Lock time	t_{SSCG}	After V_{DD} reaches MIN.: 3.3 V			1000	μs

Note Indicates the characteristics of the SSCG output frequency when it is not modulated. The modulated operating frequency is as follows, depending on the settings of the SFC1.SCF16 to SFC1.SFC14 bits. Make sure that the maximum operating frequency does not exceed 32 MHz, taking the maximum value of the modulation rate of the operating frequency into consideration.

SFC1.SFC16 to SFC1.SFC14	Modulation Rate		Operating Frequency
	TYP.	MAX.	
000B	±0.5 %	±2.0 %	31.3 MHz
001B	±1.0 %	±2.5 %	31.2 MHz
010B	±2.0 %	±4.0 %	30.7 MHz
011B	±3.0 %	±6.0 %	30.0 MHz
100B	±4.0 %	±8.0 %	29.4 MHz
101B	±5.0 %	±10.0 %	28.8 MHz

29.4.5 Low-speed internal oscillator/high-speed internal oscillator characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

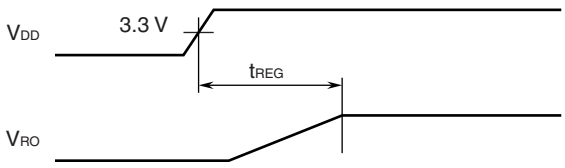
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output frequency	f_{RL}	Low-speed internal oscillator	204	240	276	kHz
	f_{RH}	High-speed internal oscillator	7.2	8.0	8.8	MHz
Oscillation stabilization time		High-speed internal oscillator operating			256	μs

29.5 Voltage Regulator Characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage	V_{DD}		3.3		5.5	V
Output voltage	V_{RO}			2.5		V
Output voltage stabilization time	t_{REG}	After V_{DD} reaches MIN.: 3.3 V, C = 4.7 μF connected to REGC pin			1	ms

Remark t_{REG} is secured by the POC function. Then reset is released.



29.6 DC Characteristics

29.6.1 I/O level

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(1/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P04, P30, P31, P34, P40, P91, P913 to P915	0.7EV_{DD}		EV_{DD}	V
	V_{IH2}	P00 to P03, P05, P06, P32, P33, P35, P41 P42, P50 to P55, P90, P96 to P99, PDL0 to PDL7	0.8EV_{DD}		EV_{DD}	V
	V_{IH3}	PCM0, PCM1	0.7EV_{DD}		EV_{DD}	V
	V_{IH4}	P70 to P79	0.7AV_{REF0}		AV_{REF0}	V
	V_{IH5}	$\overline{RESET}$, FLMD0	0.8EV_{DD}		EV_{DD}	V
Input voltage, low	V_{IL1}	P04, P30, P31, P34, P40, P91, P913 to P915	EV_{SS}		0.3EV_{DD}	V
	V_{IL2}	P00 to P03, P05, P06, P32, P33, P35, P41 P42, P50 to P55, P90, P96 to P99, PDL0 to PDL7	EV_{SS}		0.4EV_{DD}	V
	V_{IL3}	PCM0, PCM1	EV_{SS}		0.3EV_{DD}	V
	V_{IL4}	P70 to P79	AV_{SS}		0.3AV_{REF0}	V
	V_{IL5}	$\overline{RESET}$, FLMD0	EV_{SS}		0.2EV_{DD}	V

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} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(2/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high[Note 1]	V_{OH1}	P00 to P06, P30 to P35, P40 to P42, P50 to P55, P90, P91, P96 to P99, P913 to P915, PCM0, PCM1, PDL0 to PDL7	I_{OH} = –1.0 mA	EV_{DD} – 1.0		EV_{DD}	V
			I_{OH} = –0.1 mA	EV_{DD} – 0.5		EV_{DD}	V
	V_{OH3}	P70 to P79	I_{OH} = –1.0 mA	AV_{REF0} – 1.0		AV_{REF0}	V
			I_{OH} = –0.1 mA	AV_{REF0} – 0.5		AV_{REF0}	V
Output voltage, low[Note 1]	V_{OL1}	P00 to P06, P30 to P35, P40 to P42, P50 to P55, P90, P91, P96 to P99, P913 PCM0, PCM1, PDL0 to PDL7	I_{OL} = 1.0 mA	0		0.4	V
		P914, P915	I_{OL} = 3.0 mA	0		0.4	V
	V_{OL3}	P70 to P79	I_{OL} = 1.0 mA	0		0.4	V
Pull-up resistor	R_1	V_I = 0 V		10	30	100	kΩ
Pull-down resistor[Note 2]	R_2	V_I = V_{DD}		10	30	100	kΩ

Notes 1. The maximum value of the total of I_{OH}/I_{OL} is 20 mA/–20 mA for each power supply (EV_{DD}, AV_{REF0}).

2. $\overline{\text{DRST}}$ pin only

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

29.6.2 Pin leakage current

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	V_{IN} = V_{DD}	Analog pin			0.2	μA
			FLMD0 pin			2.0	
			Other than above pin			0.5	
Input leakage current, low	I_{LIL1}	V_{IN} = 0 V	Analog pin			–0.2	μA
			FLMD0 pin			–2.0	
			Other than above pin			–0.5	
Output leakage current, high	I_{LOH1}	V_O = V_{DD}	Analog pin			0.2	μA
			Other than analog pin			0.5	
Output leakage current, low	I_{LOL1}	V_O = 0 V	Analog pin			–0.2	μA
			Other than analog pin			–0.5	

<R> **29.6.3 Supply current**

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Normal operation mode[Note 2]	PLL operating	f_{XX} = 32 MHz (f_X = 8 MHz)	All peripheral function operating		39	51	mA
					All peripheral function stopped		32		mA
			PLL operating	f_{XX} = 20 MHz (f_X = 10 MHz)	All peripheral function operating		27	37	mA
					All peripheral function stopped		22		mA
			PLL stopped	f_{XX} = 16 MHz (f_X = 16 MHz)	All peripheral function operating		21	30	mA
					All peripheral function stopped		19		mA
	I_{DD2}	HALT mode[Note 2]	PLL operating	f_{XX} = 32 MHz (f_X = 8 MHz)	All peripheral function operating		24	34	mA
					All peripheral function stopped		18		mA
			PLL operating	f_{XX} = 20 MHz (f_X = 10 MHz)	All peripheral function operating		16	23	mA
					All peripheral function stopped		12		mA
			PLL stopped	f_{XX} = 16 MHz (f_X = 16 MHz)	All peripheral function operating		13	20	mA
					All peripheral function stopped		9		mA
	I_{DD3}	IDLE1 mode	PLL stopped[Note 3]	f_{XX} = 16 MHz (f_X = 16 MHz)	TAA, UARTD operating		2.4	3.6	mA
					All peripheral function stopped		1.6		mA
				f_{XX} = 8 MHz (f_X = 8 MHz)	TAA, UARTD operating		1.6	2.5	mA
					All peripheral function stopped		1.3		mA
			f_{XX} = High-speed internal oscillation (f_{RH})[Note 4]		TAA, UARTD operating		1.5	2.3	mA
					All peripheral function stopped		1.1		mA
	I_{DD4}	IDLE2 mode	PLL stopped[Note 3]	f_{XX} = 16 MHz (f_X = 16 MHz)			0.8	1.2	mA
				f_{XX} = 8 MHz (f_X = 8 MHz)			0.5	0.8	mA
			f_{XX} = High-speed internal oscillation (f_{RH})[Note 4]				0.2	0.5	mA
	I_{DD5}	Subclock operation mode[Notes 4, 5]	Crystal resonator (f_{XT} = 32.768 kHz)				80	400	μA
	I_{DD6}	Sub-IDLE mode[Notes 4, 5]	Crystal resonator (f_{XT} = 32.768 kHz)				20	190	μA
	I_{DD7}	STOP mode[Notes 4, 6]	Low-speed internal oscillator (f_{RL}) operating				18.5	100	μA
			Low-speed internal oscillator (f_{RL}) stopped				10.5	85	μA

Notes 1. Total current of V_{DD} and EV_{DD} (all ports stopped). The current of AV_{REF0} and the port buffer current including the current flowing through the on-chip pull-up/pull-down resistors are not included.

2. During SSCG operation, TYP. value + 2.5 mA, MAX. value + 4 mA

3. High-speed internal oscillator (f_{RH}) stopped.

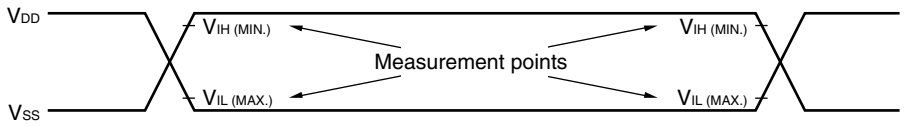
4. When the main clock oscillator (f_{XX}) is stopped.

5. Low-speed internal oscillator (f_{RL}) operating, high-speed internal oscillator (f_{RH}) stopped.

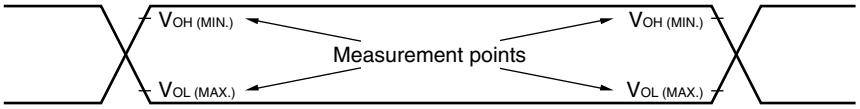
6. When the subclock oscillator (f_{XT}) is not used.

29.7 AC Characteristics

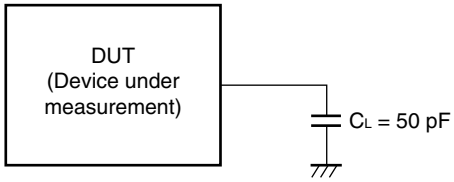
(1) AC test input measurement points (V_{DD}, AV_{REF0}, EV_{DD})



(2) AC test output measurement points



(3) Load conditions

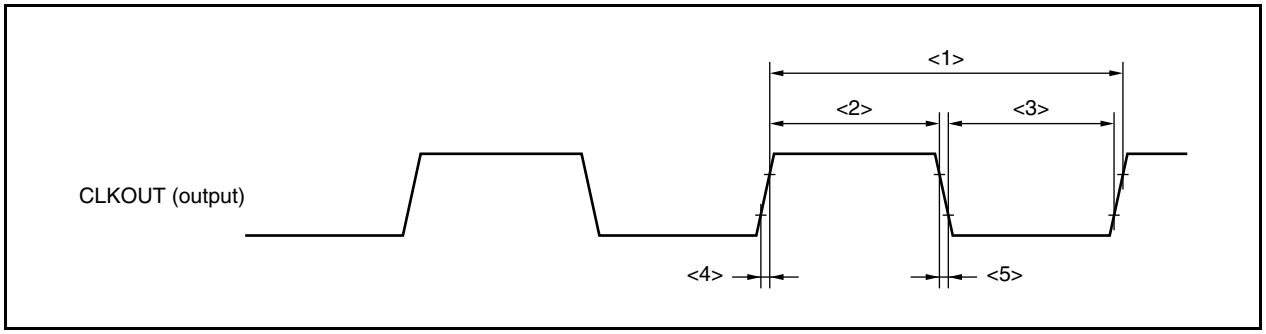


Caution If the load capacitance exceeds 50 pF due to the circuit configuration, bring the load capacitance of the device to 50 pF or less by inserting a buffer or by some other means.

29.7.1 CLKOUT output timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output cycle	t_{CYK}	<1>	V_{DD} = EV_{DD} = 3.7 V to 5.5 V	50 ns	80 μs	
			V_{DD} = EV_{DD} = 4.0 V to 5.5 V	31.25 ns	80 μs	
High-level width	t_{WKH}	<2>	V_{DD} = EV_{DD} = 3.7 V to 5.5 V	t_{CYK}/2 – 15		ns
			V_{DD} = EV_{DD} = 4.0 V to 5.5 V	t_{CYK}/2 – 13		ns
Low-level width	t_{WKL}	<3>	V_{DD} = EV_{DD} = 3.7 V to 5.5 V	t_{CYK}/2 – 15		ns
			V_{DD} = EV_{DD} = 4.0 V to 5.5 V	t_{CYK}/2 – 13		ns
Rise time	t_{KR}	<4>	V_{DD} = EV_{DD} = 3.7 V to 5.5 V		15	ns
			V_{DD} = EV_{DD} = 4.0 V to 5.5 V		13	ns
Fall time	t_{KF}	<5>	V_{DD} = EV_{DD} = 3.7 V to 5.5 V		15	ns
			V_{DD} = EV_{DD} = 4.0 V to 5.5 V		13	ns



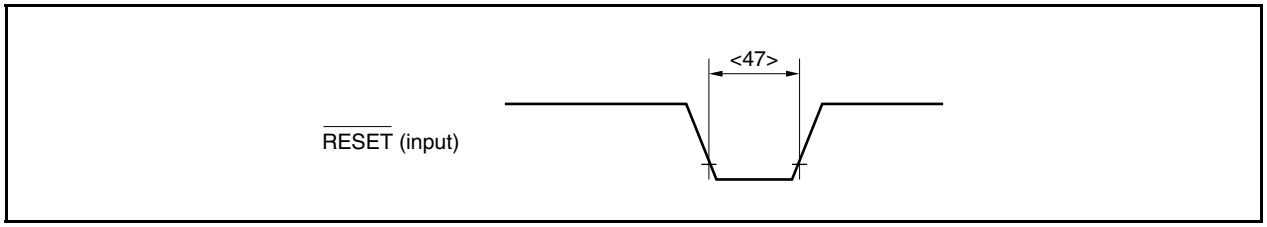
29.8 Basic Operation

(1) Reset timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{RESET}$ low-level width	t_{WRSL}	<47>		250		ns

Reset



(2) Interrupt timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

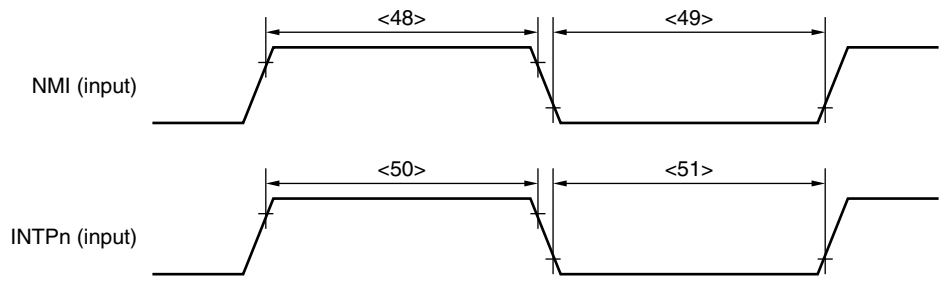
Parameter	Symbol		Conditions	MIN.	MAX.	Unit
NMI high-level width	t_{WNIH}	<48>	Analog noise elimination	250		ns
NMI low-level width	t_{WNIL}	<49>	Analog noise elimination	250		ns
INTPn[Note 1] high-level width	t_{WITH}	<50>	Analog noise elimination (n = 0 to 7)	250		ns
			Digital noise elimination (n = 3)	**Note 2**		ns
INTPn[Note 1] low-level width	t_{WITL}	<51>	Analog noise elimination (n = 0 to 7)	250		ns
			Digital noise elimination (n = 3)	**Note 2**		ns

Notes 1. The same value as the INTP0/P03 pin applies in the case of the ADTRG pin. The same value as the INTP2/P05 pin applies in the case of the $\overline{DRST}$ pin.

2. $2T_{samp}$ + 20 or $3T_{samp}$ + 20

T_{samp}: Sampling clock for noise elimination

Reset/interrupt

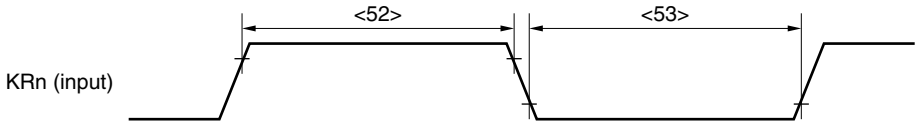


Remark n = 0 to 7

(3) Key interrupt timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
KRn input high-level width	t_{WKRH}	<52>	Analog noise elimination (n = 0 to 7)	250		ns
KRn input low-level width	t_{WKRL}	<53>		250		ns



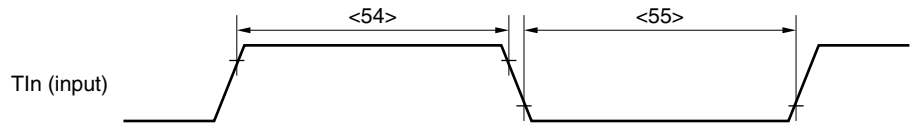
Remark n = 0 to 7

(4) Timer input timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions		MIN.	MAX.	Unit
TIn high-level width	t_{TIH}	<54>	TIAA00, TIAA01, TIAA10, TIAA11, TIAA20, TIAA21, TIAA30, TIAA31, TIAA40, TIAA41, TIAB00 to TIAB03[Note]		250		ns
TIn low-level width	t_{TIL}	<55>			250		ns
TOn output cycle	f_{TCYK}		TOAA00, TOAA01, TOAA10, TOAA11, TOAA20, TOAA21, TOAA30, TOAA31, TOAA40, TOAA41, TOAB00 to TOAB03	4.0 V ≤ V_{DD} ≤ 5.5 V		16	MHz
				3.7 V ≤ V_{DD} < 4.0 V		10	MHz

Note Noise on the TIAA00, TIAA10, TIAA20, TIAA30, and TIAB00 pins can be eliminated only when a capture signal is input.
The noise cannot be eliminated when an external trigger signal or an external event counter signal is input.



Remark TIn: TIAA00, TIAA01, TIAA10, TIAA11, TIAA20, TIAA21, TIAA30, TIAA31, TIAA40, TIAA41, TIAB00 to TIAB03

(5) CSIB timing

(a) Master mode

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKBn}$ cycle time	t_{KCY1}	<56>		125		ns
$\overline{SCKBn}$ high-level width	t_{KH1}	<57>		t_{KCY1}/2 – 15		ns
$\overline{SCKBn}$ low-level width	t_{KL1}	<58>		t_{KCY1}/2 – 15		ns
SIBn setup time (to $\overline{SCKBn}$↑)	t_{SIK1}	<59>		30		ns
SIBn hold time (from $\overline{SCKBn}$↑)	t_{KSI1}	<60>		25		ns
Output delay time from $\overline{SCKBn}$↓ to SOBn	t_{KSO1}	<61>			25	ns

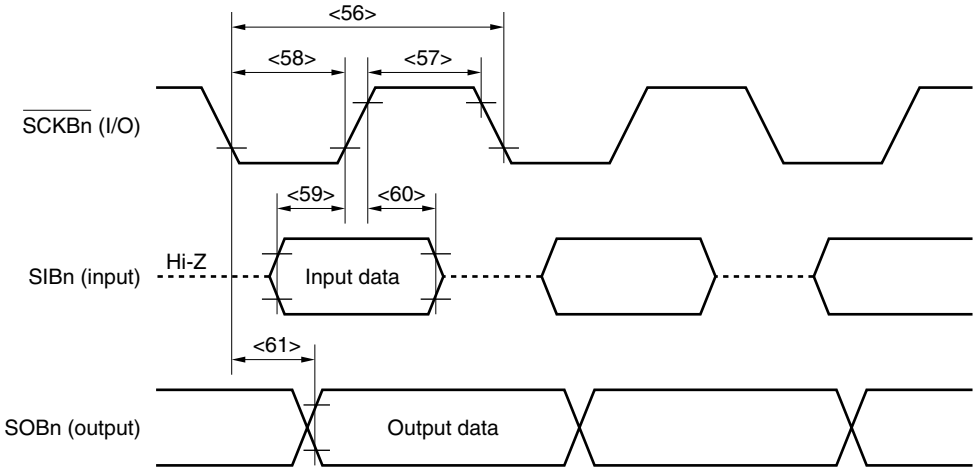
Remark n = 0, 1

(b) Slave mode

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKBn}$ cycle time	t_{KCY1}	<56>		200		ns
$\overline{SCKBn}$ high-level width	t_{KH1}	<57>		90		ns
$\overline{SCKBn}$ low-level width	t_{KL1}	<58>		90		ns
SIBn setup time (to $\overline{SCKBn}$↑)	t_{SIK1}	<59>		50		ns
SIBn hold time (from $\overline{SCKBn}$↑)	t_{KSI1}	<60>		50		ns
Output delay time from $\overline{SCKBn}$↓ to SOBn	t_{KSO1}	<61>			50	ns

Remark n = 0, 1



Remark n = 0, 1

(6) UARTD timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Communication rate				1.5	Mbps
ASCK0 cycle time				10	MHz

(7) I^2C bus timing

(T_A = –40 to +85°C, V_{DD} = V_{DD1} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = V_{SS1} = EV_{SS} = AV_{SS} = 0 V)

Parameter		Symbol		Normal Mode		High-Speed Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCL00 clock frequency		f_{CLK}		0	100	0	400	kHz
Bus free time (Between start and stop conditions)		t_{BUF}	<62>	4.7	–	1.3	–	μs
Hold time[Note 1]		$t_{HD:STA}$	<63>	4.0	–	0.6	–	μs
SCL00 clock low-level width		t_{LOW}	<64>	4.7	–	1.3	–	μs
SCL00 clock high-level width		t_{HIGH}	<65>	4.0	–	0.6	–	μs
Setup time for start/restart conditions		$t_{SU:STA}$	<66>	4.7	–	0.6	–	μs
Data hold time	CBUS compatible master	$t_{HD:DAT}$	<67>	5.0	–	–	–	μs
	I^2C mode			0[Note 2]	–	0[Note 2]	0.9[Note 3]	μs
Data setup time		$t_{SU:DAT}$	<68>	250	–	100[Note 4]	–	ns
SDA00 and SCL00 signal rise time		t_R	<69>	–	1000	20 + 0.1Cb[Note 5]	300	ns
SDA00 and SCL00 signal fall time		t_F	<70>	–	300	20 + 0.1Cb[Note 5]	300	ns
Stop condition setup time		$t_{SU:STO}$	<71>	4.0	–	0.6	–	μs
Pulse width of spike suppressed by input filter		t_{SP}	<72>	–	–	0	50	ns
Capacitance load of each bus line		Cb		–	400	–	400	pF

Notes 1. At the start condition, the first clock pulse is generated after the hold time.

2. The system requires a minimum of 300 ns hold time internally for the SDA00 signal (at $V_{IHmin.}$ of SCL00 signal) in order to occupy the undefined area at the falling edge of SCL00.

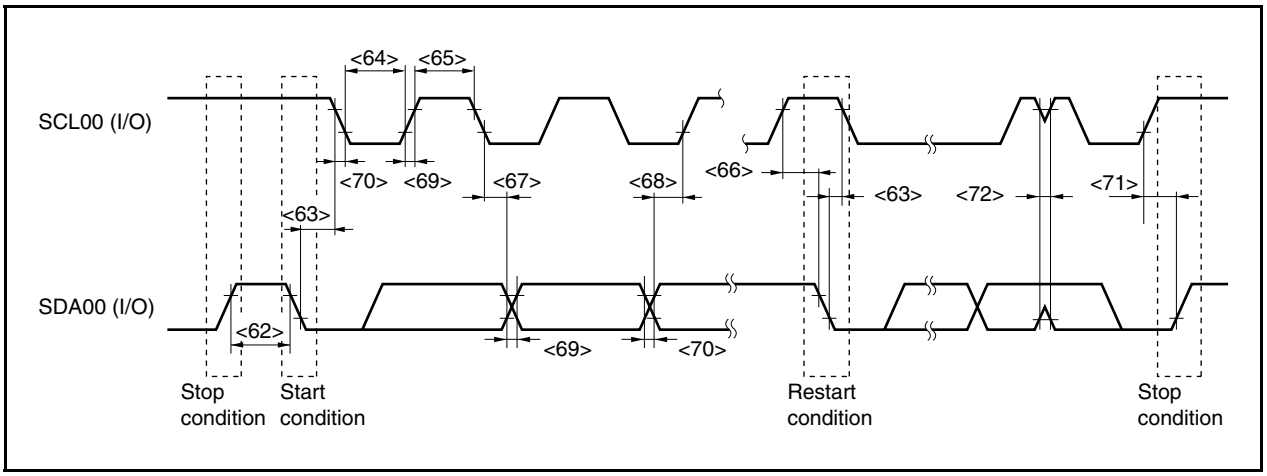
3. If the system does not extend the SCL00 signal low hold time (t_{LOW}), only the maximum data hold time ($t_{HD:DAT}$) needs to be satisfied.

4. The high-speed mode I^2C bus can be used in the normal-mode I^2C bus system. In this case, set the high-speed mode I^2C bus so that it meets the following conditions.

- If the system does not extend the SCL00 signal's low state hold time:
  $t_{SU:DAT} \geq 250$ ns
- If the system extends the SCL00 signal's low state hold time:
  Transmit the following data bit to the SDA00 line prior to the SCL00 line release ($t_{Rmax.} + t_{SU:DAT}$ = 1,000 + 250 = 1,250 ns: Normal mode I^2C bus specification).

5. Cb: Total capacitance of one bus line (unit: pF)

I²C bus mode



(8) A/D converter

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution					10	bit
Overall error[Note 1]		4.0 ≤ AV_{REF0} ≤ 5.5 V		±0.15	±0.3	%FSR
Conversion time	t_{CONV}		3.1		16	μs
Stabilization time	t_{STA}	After ADA0M0.ADA0PS bit changes from 0 to 1	2			μs
Power down recovery time	t_{DPU}	Starting operation after STOP mode is released	1			μs
Zero scale error[Note 1]	ZSE				±0.3	%FSR
Full scale error[Note 1]	FSE				±0.3	%FSR
Non-linearity error[Note 2]	INL				±2.5	LSB
Differential linearity error[Note 2]	DNL				±1.5	LSB
Analog input voltage	V_{IAN}		AV_{SS}		AV_{REF0}	V
AV_{REF0} current	I_{AREF0}	When using A/D converter		4	7	mA
		When not using A/D converter		1	10	μA

Notes 1. Excluding quantization error (±0.05 %FSR). Indicates the ratio to the full-scale value (%FSR).

2. Quantization error (±0.5LSB)

Remark FSR: Full Scale Range

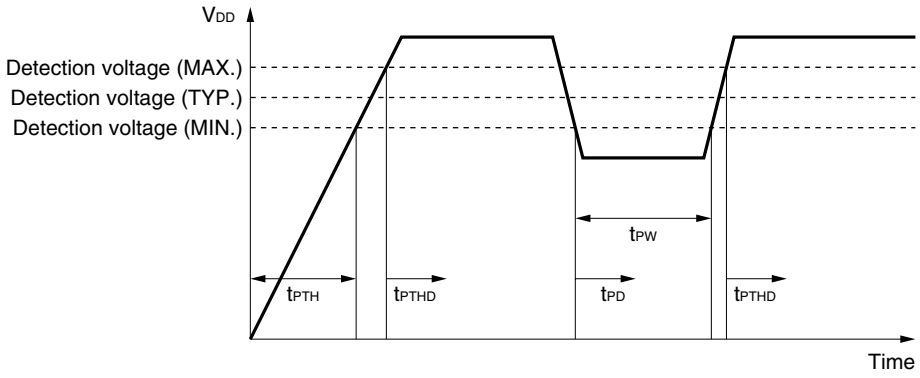
(9) POC circuit characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC0}		3.3	3.5	3.7	V
Power supply startup time	t_{PTH}	V_{DD} = 0 V → 3.3 V	0.002			ms
Response delay time 1 Note 1	t_{PTHD}	After V_{DD} reaches 3.7 V on power application			2.0	ms
Response delay time 2 Note 2	t_{PD}	After V_{DD} drops below 3.3 V on power drop			1.0	ms
Minimum V_{DD} width	t_{PW}		0.2			ms

Notes 1. The time required to release a reset after the detection voltage is detected.

2. The time required to output a reset after the detection voltage is detected.

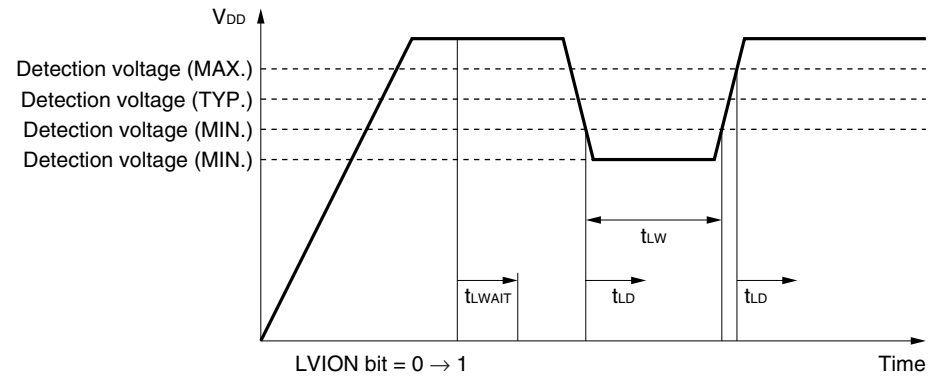


(10) LVI circuit characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LVI0}		3.8	4.0	4.2	V
	V_{LVI1}		3.5	3.7	3.9	V
Response time[Note]	t_{LD}	After V_{DD} reaches V_{LVI0}/V_{LVI1} (MAX.) or drops below V_{LVI0}/V_{LVI1} (MIN.)		0.2	2.0	ms
Minimum V_{DD} width	t_{LW}		0.2			ms
Reference voltage stabilization wait time	t_{LWAIT}	After LVION bit (LVIM.bit7) changes from 0 to 1		0.1	0.2	ms

Note The time required to output an interrupt/reset after the detection voltage is detected.

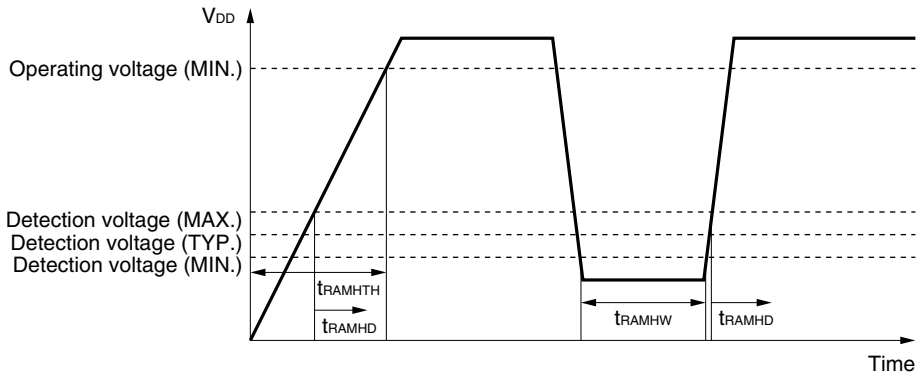


(11) RAM retention flag characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 1.9 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{RAMH}		1.9	2.0	2.1	V
Supply voltage rise time	t_{RAMHTH}	V_{DD} = 0 V → 3.3 V	0.002		1800	ms
Response time[Note]	t_{RAMHD}	After the supply voltage reaches the detection voltage (MAX.)		0.2	2.0	ms
Minimum V_{DD} width	t_{RAMHW}		0.2			ms

Note Time required to set the RAMF bit after the detection voltage is detected.



29.9 Flash Memory Programming Characteristics

(1) Basic characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD}, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operating frequency	f_{CPU}		4		32	MHz
Supply voltage	V_{DD}		3.8		5.5	V
Number of writes	C_{WRT} Note				100	Times
Input voltage, high	V_{IH}	FLMD0	0.8EV_{DD}		EV_{DD}	V
Input voltage, low	V_{IL}	FLMD0	EV_{SS}		0.2EV_{DD}	V
Programming temperature	t_{PRG}		–40		+85	°C

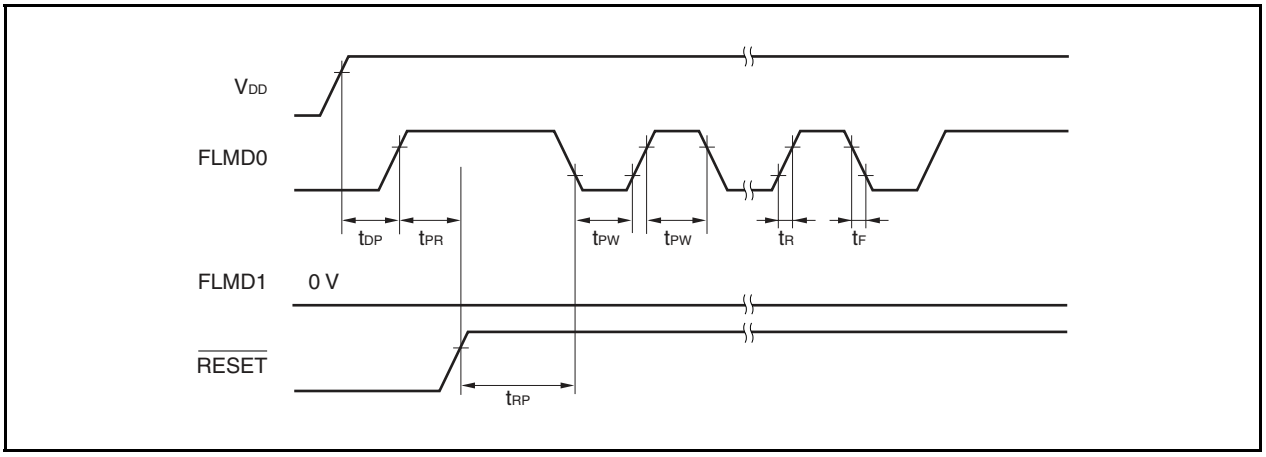
Note When writing initially to shipped products, it is counted as one rewrite for both “erase to write” and “write only”.

Example (P: Write, E: Erase)
Shipped product → P → E → P → E → P: 3 rewrites
Shipped product → E → P → E → P → E → P: 3 rewrites

(2) Serial write operation characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD}, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Setup time from VDD↑ to FLMD0↑	t_{DP}		1			ms
Time from FLMD0↑ to RESET release	t_{PR}		2			ms
FLMD0 pulse input start time from RESET↑	t_{RP}		800			μs
FLMD0 pulse high-level width/ low-level width	t_{PW}		10		100	μs
<R> FLMD0 rise time	t_R				1	μs
<R> FLMD0 fall time	t_F				1	μs



CHAPTER 30 ELECTRICAL SPECIFICATIONS (V850ES/HF3)

30.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}	$V_{DD} = EV_{DD}$	−0.5 to +6.5	V
	EV_{DD}	$V_{DD} = EV_{DD}$	−0.5 to +6.5	V
	AV_{REF0}		−0.5 to +6.5	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}	P00 to P06, P30 to 35, P38, P39, P40 to P42, P50 to P55, P90, P91, P96 to P99, P913 to P915, PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL11, $\overline{\text{RESET}}$, FLMD0	−0.5 to EV_{DD} + 0.5[Note]	V
	V_{I3}	X1, X2, XT1, XT2	−0.5 to V_{RO} + 0.5	V
Analog input voltage	V_{IAN}	P70 to P711	−0.5 to AV_{REF0} + 0.5[Note]	V

Note Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

Cautions 1. Do not directly connect the output (or I/O) pins of IC products to each other, or to V_{DD}, 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 and AC characteristics represent the quality assurance range during normal operation.

Remark Unless specified otherwise, the characteristics 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}	P00 to P06, P30 to P35, P38, P39, P40 to P42, P50 to P55, P90, P91, P96 to P99, P913 to P915, PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL11	Per pin	4	mA
			Total of all pins	50	mA
		P70 to P711	Per pin	4	mA
			Total of all pins	20	mA
Output current, high	I_{OH}	P00 to P06, P30 to P35, P38, P39, P40 to P42, P50 to P55, P90, P91, P96 to P99, P913 to P915, PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL11	Per pin	−4	mA
			Total of all pins	−50	mA
		P70 to P711	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			
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 and AC characteristics represent the quality assurance range during normal operation.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

30.2 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

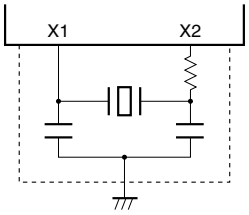
(T_A = –40 to +85°C, V_{DD} = EV_{DD}, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C = 4.7 μF)

Internal System Clock Frequency	Conditions	Supply Voltage		Unit
		V_{DD}, EV_{DD}	AV_{REF0}	
4 MHz ≤ f_{XX} ≤ 32 MHz		4.0 to 5.5	4.0 to 5.5	V
4 MHz ≤ f_{XX} ≤ 20 MHz	AD converter operating	3.7 to 5.5	4.0 to 5.5	V
	AD converter stop	3.7 to 5.5	3.7 to 5.5	V
f_{XT} = 32.768 kHz		3.7 to 5.5	3.7 to 5.5	V
f_{RL} = 240 kHz (TYP.)		3.7 to 5.5	3.7 to 5.5	V
f_{RH} = 8 MHz (TYP.)	AD converter operating	3.7 to 5.5	4.0 to 5.5	V
	AD converter stop	3.7 to 5.5	3.7 to 5.5	V

30.4 Oscillator Characteristics

30.4.1 Main clock oscillator characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions		MIN.	TYP.	MAX.	Unit
Ceramic resonator/ crystal resonator		Oscillation frequency (f_X)[Note 1]			4		16	MHz
		Oscillation stabilization time[Note 2]	After STOP mode release	PLL stopped	54[Note 3]	**Note 4**		μs
				PLL operating	1600[Note 5]	**Note 4**		μs
				SSCG operating	2000[Note 6]	**Note 4**		μs
			After IDLE2 mode release	PLL stopped	54[Note 3]	**Note 4**		μs
				PLL operating	800[Note 5]	**Note 4**		μs
				SSCG operating	1000[Note 6]	**Note 4**		μs

Notes 1. Indicates only oscillator characteristics.

2. Time required from start of oscillation until the resonator stabilizes.

3. Time required to stabilize access to the internal flash memory.

4. The value differs depending on the OSTS register settings.

5. PLL lockup time

6. SSCG lockup time

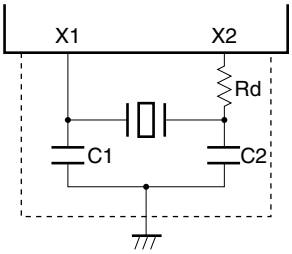
Cautions 1. When using the main clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **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. When the main clock is stopped and the subclock is operating, wait until the oscillation stabilization time has been secured by the program before switching back to the main clock.

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.

<R> **(i) KYOCERA KINSEKI CORPORATION: Crystal resonator (T_A = –10 to +70°C)**

Type	Circuit Example	Part Number	Oscillation Frequency f_X (MHz)	Recommended Circuit Constant		
				C1 (pF)	C2 (pF)	Rd (kΩ)
Lead		HC49SFWB06000D0PESZZ	6.000	8	8	0
		HC49SFWB08000D0PESZZ	8.000	8	8	0
		HC49SFWB10000D0PESZZ	10.000	8	8	0
Surface mounting		HC49GFWB06000D0PESZZ	6.000	8	8	0
		CX1255GB06000D0PESZZ	6.000	8	8	0
		CX8045GB06000D0PESZZ	6.000	8	8	0
		HC49GFWB08000D0PESZZ	8.000	8	8	0
		CX1255GB08000D0PESZZ	8.000	8	8	0
		CX8045GB08000D0PESZZ	8.000	8	8	0
		CX5032GB08000D0PESZZ	8.000	8	8	0
		HC49GFWB10000D0PESZZ	10.000	8	8	0
		CX1255GB10000D0PESZZ	10.000	8	8	0
		CX8045GB10000D0PESZZ	10.000	8	8	0
		CX5032GB10000D0PESZZ	10.000	8	8	0

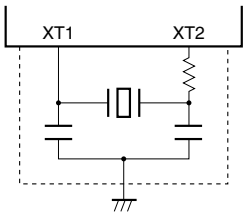
Caution This oscillator constant is a reference value based on evaluation under a specific environment by the resonator manufacturer.

If optimization of oscillator characteristics is necessary in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

The oscillation voltage and oscillation frequency indicate only oscillator characteristics. Use the V850ES/HF3 so that the internal operating conditions are within the specifications of the DC and AC characteristics.

30.4.2 Subclock oscillator characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	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. Indicates only oscillator characteristics. For the CPU operation clock, see **30.7 AC Characteristics**.

2. Time required from when V_{DD} reaches oscillation voltage range (MIN.: 3.3 V) to when the oscillation 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 current 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°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
<R> Input frequency	f_X		6		12	MHz
Output frequency	f_{XX}		12		32	MHz
Lock time	t_{PLL}	After V_{DD} reaches MIN.: 3.3 V			800	μs

30.4.4 SSCG characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input frequency	f_X		4		16	MHz
Output frequency[Note]	f_{XX}		12		32	MHz
Lock time	t_{SSCG}	After V_{DD} reaches MIN.: 3.3 V			1000	μs

Note Indicates the characteristics of the SSCG output frequency when it is not modulated. The modulated operating frequency is as follows, depending on the settings of the SFC1.SCF16 to SFC1.SFC14 bits. Make sure that the maximum operating frequency does not exceed 32 MHz, taking the maximum value of the modulation rate of the operating frequency into consideration.

SFC1.SFC16 to SFC1.SFC14	Modulation Rate		Operating Frequency
	TYP.	MAX.	
000B	±0.5 %	±2.0 %	31.3 MHz
001B	±1.0 %	±2.5 %	31.2 MHz
010B	±2.0 %	±4.0 %	30.7 MHz
011B	±3.0 %	±6.0 %	30.0 MHz
100B	±4.0 %	±8.0 %	29.4 MHz
101B	±5.0 %	±10.0 %	28.8 MHz

30.4.5 Low-speed internal oscillator/high-speed internal oscillator characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

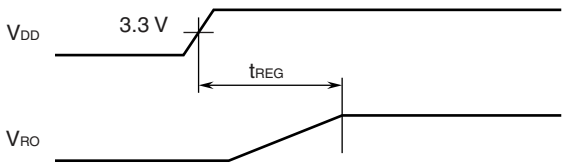
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output frequency	f_{RL}	Low-speed internal oscillator	204	240	276	kHz
	f_{RH}	High-speed internal oscillator	7.2	8.0	8.8	MHz
Oscillation stabilization time		High-speed internal oscillator operating			256	μs

30.5 Voltage Regulator Characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage	V_{DD}		3.3		5.5	V
Output voltage	V_{RO}			2.5		V
Output voltage stabilization time	t_{REG}	After V_{DD} reaches MIN.: 3.3 V, C = 4.7 μF connected to REGC pin			1	ms

Remark t_{REG} is secured by the POC function. Then reset is released.



30.6 DC Characteristics

30.6.1 I/O level

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(1/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P04, P30, P31, P34, P38, P39, P40, P91, P913 to P915	0.7EV_{DD}		EV_{DD}	V
	V_{IH2}	P00 to P03, P05, P06, P32, P33, P35, P41 P42, P50 to P55, P90, P96 to P99, PDL0 to PDL11	0.8EV_{DD}		EV_{DD}	V
	V_{IH3}	PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6	0.7EV_{DD}		EV_{DD}	V
	V_{IH4}	P70 to P711	0.7AV_{REF0}		AV_{REF0}	V
	V_{IH5}	$\overline{RESET}$, FLMD0	0.8EV_{DD}		EV_{DD}	V
Input voltage, low	V_{IL1}	P04, P30, P31, P34, P38, P39, P40, P91, P913 to P915	EV_{SS}		0.3EV_{DD}	V
	V_{IL2}	P00 to P03, P05, P06, P32, P33, P35, P41 P42, P50 to P55, P90, P96 to P99, PDL0 to PDL11	EV_{SS}		0.4EV_{DD}	V
	V_{IL3}	PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6	EV_{SS}		0.3EV_{DD}	V
	V_{IL4}	P70 to P711	AV_{SS}		0.3AV_{REF0}	V
	V_{IL5}	$\overline{RESET}$, FLMD0	EV_{SS}		0.2EV_{DD}	V

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} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

(2/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high[Note 1]	V_{OH1}	P00 to P06, P30 to P35, P38, P39, P40 to P42, P50 to P55, P90, P91, P96 to P99, P913 to P915, PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL11	I_{OH} = –1.0 mA	EV_{DD} – 1.0		EV_{DD}	V
			I_{OH} = –0.1 mA	EV_{DD} – 0.5		EV_{DD}	V
	V_{OH3}	P70 to P711	I_{OH} = –1.0 mA	AV_{REF0} – 1.0		AV_{REF0}	V
			I_{OH} = –0.1 mA	AV_{REF0} – 0.5		AV_{REF0}	V
Output voltage, low[Note 1]	V_{OL1}	P00 to P06, P30 to P35, P38, P39, P40 to P42, P50 to P55, P90, P91 P96 to P99, P913, PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL11	I_{OL} = 1.0 mA	0		0.4	V
		P914, P915	I_{OL} = 3.0 mA	0		0.4	V
	V_{OL3}	P70 to P711	I_{OL} = 1.0 mA	0		0.4	V
Pull-up resistor	R_1	V_I = 0 V		10	30	100	kΩ
Pull-down resistor[Note 2]	R_2	V_I = V_{DD}		10	30	100	kΩ

Notes 1. The maximum value of the total of I_{OH}/I_{OL} is 20 mA/–20 mA for each power supply (EV_{DD}, AV_{REF0}).

2. $\overline{\text{DRST}}$ pin only

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

30.6.2 Pin leakage current

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	V_{IN} = V_{DD}	Analog pin			0.2	μA
			FLMD0 pin			2.0	
			Other than above pin			0.5	
Input leakage current, low	I_{LIL1}	V_{IN} = 0 V	Analog pin			–0.2	μA
			FLMD0 pin			–2.0	
			Other than above pin			–0.5	
Output leakage current, high	I_{LOH1}	V_O = V_{DD}	Analog pin			0.2	μA
			Other than analog pin			0.5	
Output leakage current, low	I_{LOL1}	V_O = 0 V	Analog pin			–0.2	μA
			Other than analog pin			–0.5	

<R> **30.6.3 Supply current**

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Normal operation mode[Note 2]	PLL operating	f_{XX} = 32 MHz (f_X = 8 MHz)	All peripheral function operating		39	51	mA
					All peripheral function stopped		32		mA
			PLL operating	f_{XX} = 20 MHz (f_X = 10 MHz)	All peripheral function operating		27	37	mA
					All peripheral function stopped		22		mA
			PLL stopped	f_{XX} = 16 MHz (f_X = 16 MHz)	All peripheral function operating		21	30	mA
					All peripheral function stopped		19		mA
	I_{DD2}	HALT mode[Note 2]	PLL operating	f_{XX} = 32 MHz (f_X = 8 MHz)	All peripheral function operating		24	34	mA
					All peripheral function stopped		18		mA
			PLL operating	f_{XX} = 20 MHz (f_X = 10 MHz)	All peripheral function operating		16	23	mA
					All peripheral function stopped		12		mA
			PLL stopped	f_{XX} = 16 MHz (f_X = 16 MHz)	All peripheral function operating		13	20	mA
					All peripheral function stopped		9		mA
	I_{DD3}	IDLE1 mode	PLL stopped[Note 3]	f_{XX} = 16 MHz (f_X = 16 MHz)	TAA, UARTD operating		2.4	3.6	mA
					All peripheral function stopped		1.6		mA
				f_{XX} = 8 MHz (f_X = 8 MHz)	TAA, UARTD operating		1.6	2.5	mA
					All peripheral function stopped		1.3		mA
			f_{XX} = High-speed internal oscillation (f_{RH})[Note 4]		TAA, UARTD operating		1.5	2.3	mA
					All peripheral function stopped		1.1		mA
	I_{DD4}	IDLE2 mode	PLL stopped[Note 3]	f_{XX} = 16 MHz (f_X = 16 MHz)			0.8	1.2	mA
				f_{XX} = 8 MHz (f_X = 8 MHz)			0.5	0.8	mA
			f_{XX} = High-speed internal oscillation (f_{RH})[Note 4]				0.2	0.5	mA
	I_{DD5}	Subclock operation mode[Notes 4, 5]	Crystal resonator (f_{XT} = 32.768 kHz)				80	400	μA
	I_{DD6}	Sub-IDLE mode[Notes 4, 5]	Crystal resonator (f_{XT} = 32.768 kHz)				20	190	μA
	I_{DD7}	STOP mode[Notes 4, 6]	Low-speed internal oscillator (f_{RL}) operating				18.5	100	μA
			Low-speed internal oscillator (f_{RL}) stopped				10.5	85	μA

Notes 1. Total current of V_{DD} and EV_{DD} (all ports stopped). The current of AV_{REF0} and the port buffer current including the current flowing through the on-chip pull-up/pull-down resistors are not included.

2. During SSCG operation, TYP. value + 2.5 mA, MAX. value + 4 mA

3. High-speed internal oscillator (f_{RH}) stopped.

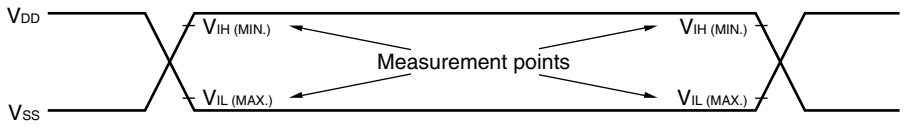
4. When the main clock oscillator (f_{XX}) is stopped.

5. Low-speed internal oscillator (f_{RL}) operating, high-speed internal oscillator (f_{RH}) stopped.

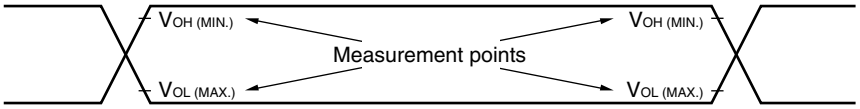
6. When the subclock oscillator (f_{XT}) is not used.

30.7 AC Characteristics

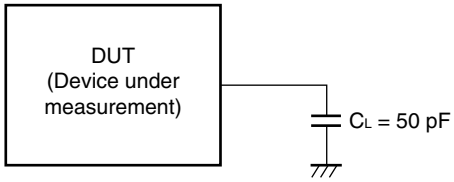
(1) AC test input measurement points (V_{DD}, AV_{REF0}, EV_{DD})



(2) AC test output measurement points



(3) Load conditions

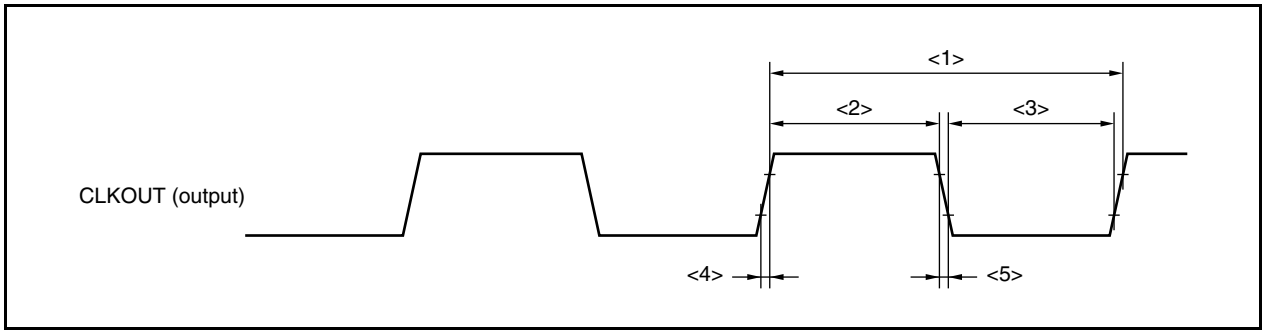


Caution If the load capacitance exceeds 50 pF due to the circuit configuration, bring the load capacitance of the device to 50 pF or less by inserting a buffer or by some other means.

30.7.1 CLKOUT output timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output cycle	t_{CYK}	<1>	V_{DD} = EV_{DD} = 3.7 V to 5.5 V	50 ns	80 μs	
			V_{DD} = EV_{DD} = 4.0 V to 5.5 V	31.25 ns	80 μs	
High-level width	t_{WKH}	<2>	V_{DD} = EV_{DD} = 3.7 V to 5.5 V	t_{CYK}/2 – 15		ns
			V_{DD} = EV_{DD} = 4.0 V to 5.5 V	t_{CYK}/2 – 13		ns
Low-level width	t_{WKL}	<3>	V_{DD} = EV_{DD} = 3.7 V to 5.5 V	t_{CYK}/2 – 15		ns
			V_{DD} = EV_{DD} = 4.0 V to 5.5 V	t_{CYK}/2 – 13		ns
Rise time	t_{KR}	<4>	V_{DD} = EV_{DD} = 3.7 V to 5.5 V		15	ns
			V_{DD} = EV_{DD} = 4.0 V to 5.5 V		13	ns
Fall time	t_{KF}	<5>	V_{DD} = EV_{DD} = 3.7 V to 5.5 V		15	ns
			V_{DD} = EV_{DD} = 4.0 V to 5.5 V		13	ns



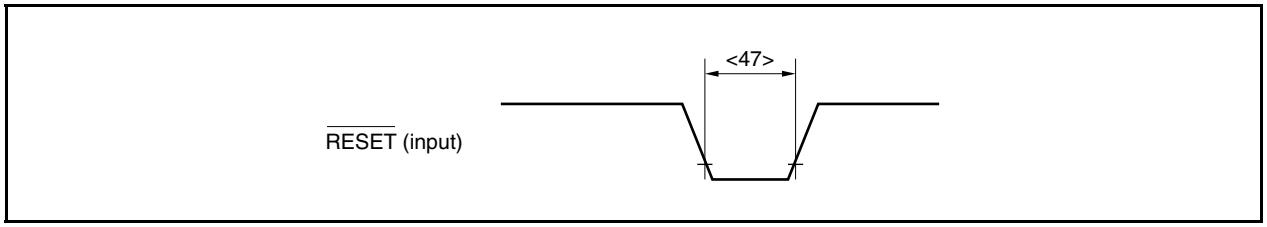
30.8 Basic Operation

(1) Reset timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ low-level width	t_{WRSL}	<47>		250		ns

Reset



(2) Interrupt timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

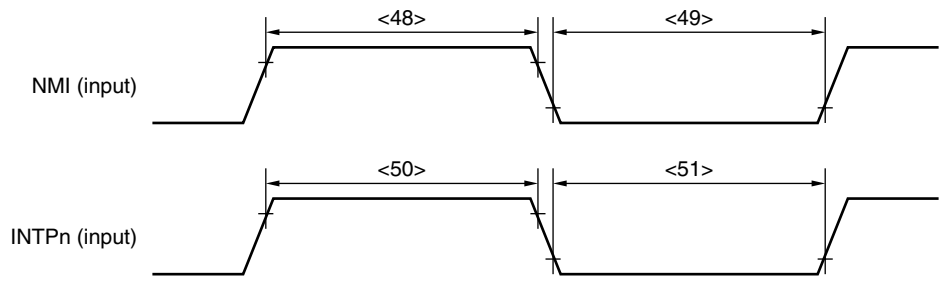
Parameter	Symbol		Conditions	MIN.	MAX.	Unit
NMI high-level width	t_{WNIH}	<48>	Analog noise elimination	250		ns
NMI low-level width	t_{WNIL}	<49>	Analog noise elimination	250		ns
INTPn[Note 1] high-level width	t_{WITH}	<50>	Analog noise elimination (n = 0 to 7)	250		ns
			Digital noise elimination (n = 3)	**Note 2**		ns
INTPn[Note 1] low-level width	t_{WITL}	<51>	Analog noise elimination (n = 0 to 7)	250		ns
			Digital noise elimination (n = 3)	**Note 2**		ns

Notes 1. The same value as the INTP0/P03 pin applies in the case of the ADTRG pin. The same value as the INTP2/P05 pin applies in the case of the $\overline{\text{DRST}}$ pin.

2. $2T_{samp} + 20$ or $3T_{samp} + 20$

T_{samp}: Sampling clock for noise elimination

Reset/interrupt

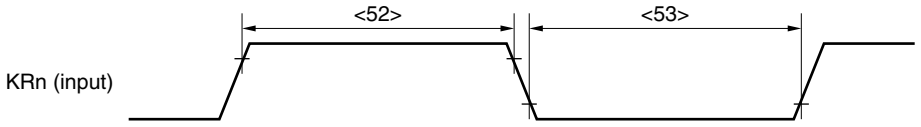


Remark n = 0 to 7

(3) Key interrupt timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
KRn input high-level width	t_{WKRH}	<52>	Analog noise elimination (n = 0 to 7)	250		ns
KRn input low-level width	t_{WKRL}	<53>		250		ns



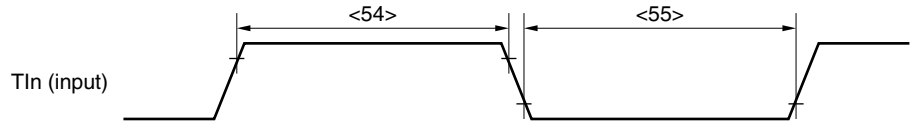
Remark n = 0 to 7

(4) Timer input timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions		MIN.	MAX.	Unit
TIn high-level width	t_{TIH}	<54>	TIAA00, TIAA01, TIAA10, TIAA11, TIAA20, TIAA21, TIAA30, TIAA31, TIAA40, TIAA41, TIAB00 to TIAB03[Note]		250		ns
TIn low-level width	t_{TIL}	<55>			250		ns
TOn output cycle	f_{TCYK}		TOAA00, TOAA01, TOAA10, TOAA11, TOAA20, TOAA21, TOAA30, TOAA31, TOAA40, TOAA41, TOAB00 to TOAB03	4.0 V ≤ V_{DD} ≤ 5.5 V		16	MHz
				3.7 V ≤ V_{DD} < 4.0 V		10	MHz

Note Noise on the TIAA00, TIAA10, TIAA20, TIAA30, and TIAB00 pins can be eliminated only when a capture signal is input.
The noise cannot be eliminated when an external trigger signal or an external event counter signal is input.



Remark TIn: TIAA00, TIAA01, TIAA10, TIAA11, TIAA20, TIAA21, TIAA30, TIAA31, TIAA40, TIAA41, TIAB00 to TIAB03

(5) CSIB timing

(a) Master mode

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKBn}$ cycle time	t_{KCY1}	<56>		125		ns
$\overline{SCKBn}$ high-level width	t_{KH1}	<57>		t_{KCY1}/2 – 15		ns
$\overline{SCKBn}$ low-level width	t_{KL1}	<58>		t_{KCY1}/2 – 15		ns
SIBn setup time (to $\overline{SCKBn}$↑)	t_{SIK1}	<59>		30		ns
SIBn hold time (from $\overline{SCKBn}$↑)	t_{KSI1}	<60>		25		ns
Output delay time from $\overline{SCKBn}$↓ to SOBn	t_{KSO1}	<61>			25	ns

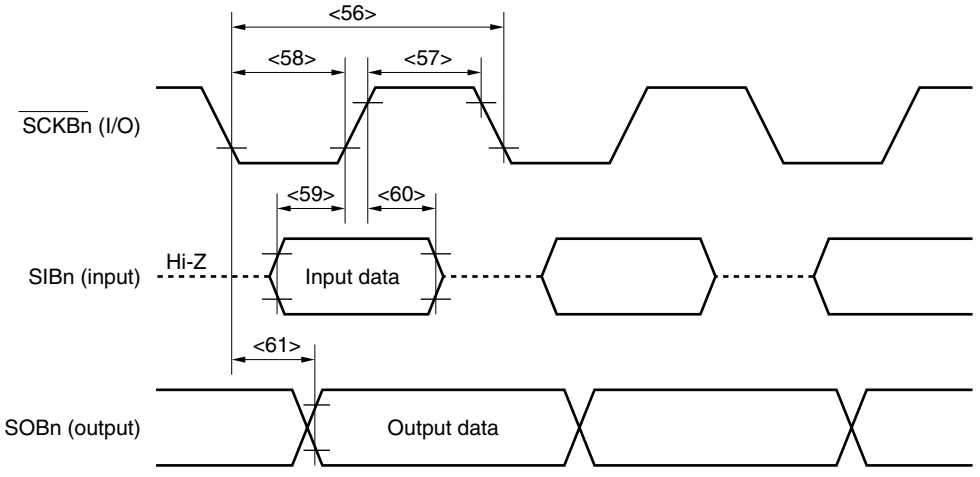
Remark n = 0, 1

(b) Slave mode

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKBn}$ cycle time	t_{KCY1}	<56>		200		ns
$\overline{SCKBn}$ high-level width	t_{KH1}	<57>		90		ns
$\overline{SCKBn}$ low-level width	t_{KL1}	<58>		90		ns
SIBn setup time (to $\overline{SCKBn}$↑)	t_{SIK1}	<59>		50		ns
SIBn hold time (from $\overline{SCKBn}$↑)	t_{KSI1}	<60>		50		ns
Output delay time from $\overline{SCKBn}$↓ to SOBn	t_{KSO1}	<61>			50	ns

Remark n = 0, 1



Remark n = 0, 1

(6) UARTD timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Communication rate				1.5	Mbps
ASCK0 cycle time				10	MHz

(7) I^2C bus timing

(T_A = –40 to +85°C, V_{DD} = V_{DD1} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = V_{SS1} = EV_{SS} = AV_{SS} = 0 V)

Parameter		Symbol		Normal Mode		High-Speed Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCL00 clock frequency		f_{CLK}		0	100	0	400	kHz
Bus free time (Between start and stop conditions)		t_{BUF}	<62>	4.7	–	1.3	–	μs
Hold time[Note 1]		$t_{HD:STA}$	<63>	4.0	–	0.6	–	μs
SCL00 clock low-level width		t_{LOW}	<64>	4.7	–	1.3	–	μs
SCL00 clock high-level width		t_{HIGH}	<65>	4.0	–	0.6	–	μs
Setup time for start/restart conditions		$t_{SU:STA}$	<66>	4.7	–	0.6	–	μs
Data hold time	CBUS compatible master	$t_{HD:DAT}$	<67>	5.0	–	–	–	μs
	I^2C mode			0[Note 2]	–	0[Note 2]	0.9[Note 3]	μs
Data setup time		$t_{SU:DAT}$	<68>	250	–	100[Note 4]	–	ns
SDA00 and SCL00 signal rise time		t_R	<69>	–	1000	20 + 0.1Cb[Note 5]	300	ns
SDA00 and SCL00 signal fall time		t_F	<70>	–	300	20 + 0.1Cb[Note 5]	300	ns
Stop condition setup time		$t_{SU:STO}$	<71>	4.0	–	0.6	–	μs
Pulse width of spike suppressed by input filter		t_{SP}	<72>	–	–	0	50	ns
Capacitance load of each bus line		Cb		–	400	–	400	pF

Notes 1. At the start condition, the first clock pulse is generated after the hold time.

2. The system requires a minimum of 300 ns hold time internally for the SDA00 signal (at $V_{IHmin.}$ of SCL00 signal) in order to occupy the undefined area at the falling edge of SCL00.

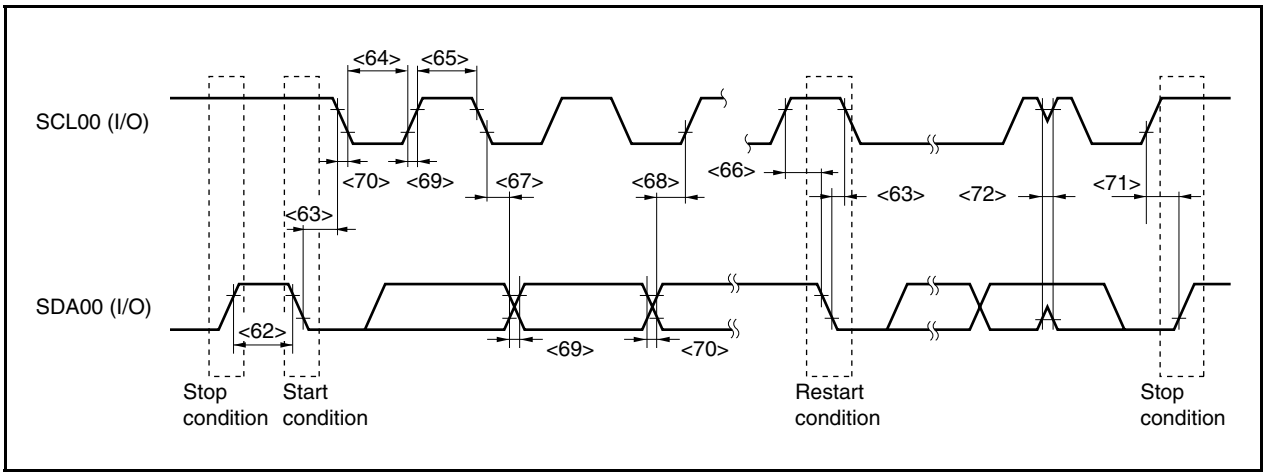
3. If the system does not extend the SCL00 signal low hold time (t_{LOW}), only the maximum data hold time ($t_{HD:DAT}$) needs to be satisfied.

4. The high-speed mode I^2C bus can be used in the normal-mode I^2C bus system. In this case, set the high-speed mode I^2C bus so that it meets the following conditions.

- If the system does not extend the SCL00 signal's low state hold time:
  $t_{SU:DAT} \geq 250$ ns
- If the system extends the SCL00 signal's low state hold time:
  Transmit the following data bit to the SDA00 line prior to the SCL00 line release ($t_{Rmax.} + t_{SU:DAT}$ = 1,000 + 250 = 1,250 ns: Normal mode I^2C bus specification).

5. Cb: Total capacitance of one bus line (unit: pF)

I²C bus mode



(8) A/D converter

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution					10	bit
Overall error[Note 1]		4.0 ≤ AV_{REF0} ≤ 5.5 V		±0.15	±0.3	%FSR
Conversion time	t_{CONV}		3.1		16	μs
Stabilization time	t_{STA}	After ADA0M0.ADA0PS bit changes from 0 to 1	2			μs
Power down recovery time	t_{DPU}	Starting operation after STOP mode is released	1			μs
Zero scale error[Note 1]	ZSE				±0.3	%FSR
Full scale error[Note 1]	FSE				±0.3	%FSR
Non-linearity error[Note 2]	INL				±2.5	LSB
Differential linearity error[Note 2]	DNL				±1.5	LSB
Analog input voltage	V_{IAN}		AV_{SS}		AV_{REF0}	V
AV_{REF0} current	I_{AREF0}	When using A/D converter		4	7	mA
		When not using A/D converter		1	10	μA

Notes 1. Excluding quantization error (±0.05 %FSR). Indicates the ratio to the full-scale value (%FSR).

2. Quantization error (±0.5LSB)

Remark FSR: Full Scale Range

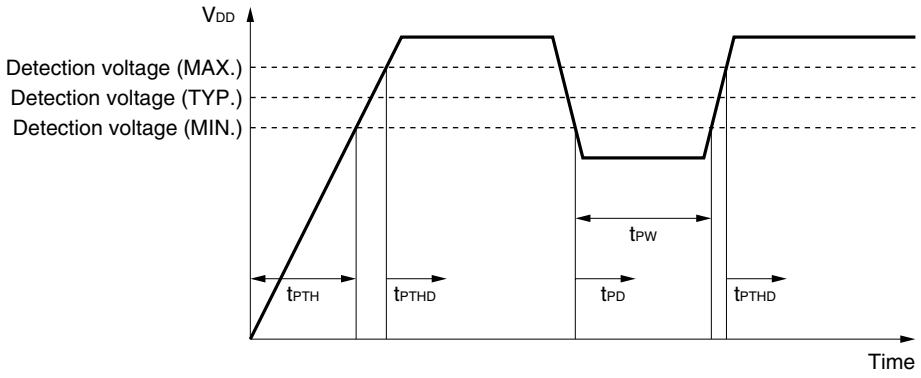
(9) POC circuit characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC0}		3.3	3.5	3.7	V
Power supply startup time	t_{PTH}	V_{DD} = 0 V → 3.3 V	0.002			ms
Response delay time 1 Note 1	t_{PTHD}	After V_{DD} reaches 3.7 V on power application			2.0	ms
Response delay time 2 Note 2	t_{PD}	After V_{DD} drops below 3.3 V on power drop			1.0	ms
Minimum V_{DD} width	t_{PW}		0.2			ms

Notes 1. The time required to release a reset after the detection voltage is detected.

2. The time required to output a reset after the detection voltage is detected.

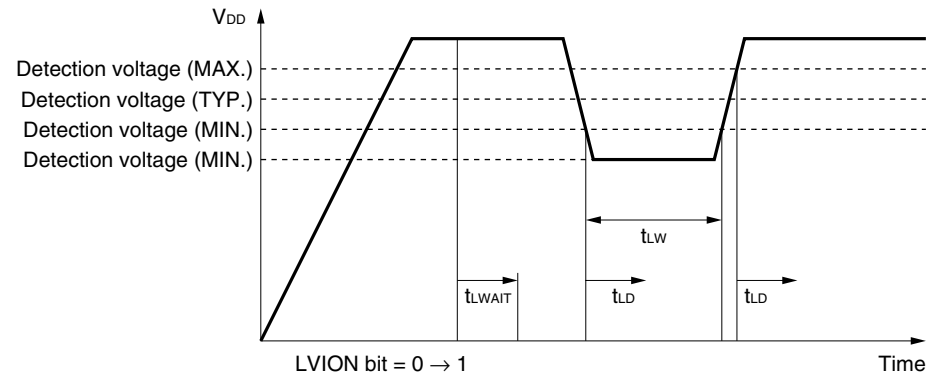


(10) LVI circuit characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LVI0}		3.8	4.0	4.2	V
	V_{LVI1}		3.5	3.7	3.9	V
Response time[Note]	t_{LD}	After V_{DD} reaches V_{LVI0}/V_{LVI1} (MAX.) or drops below V_{LVI0}/V_{LVI1} (MIN.)		0.2	2.0	ms
Minimum V_{DD} width	t_{LW}		0.2			ms
Reference voltage stabilization wait time	t_{LWAIT}	After LVION bit (LVIM.bit7) changes from 0 to 1		0.1	0.2	ms

Note The time required to output an interrupt/reset after the detection voltage is detected.

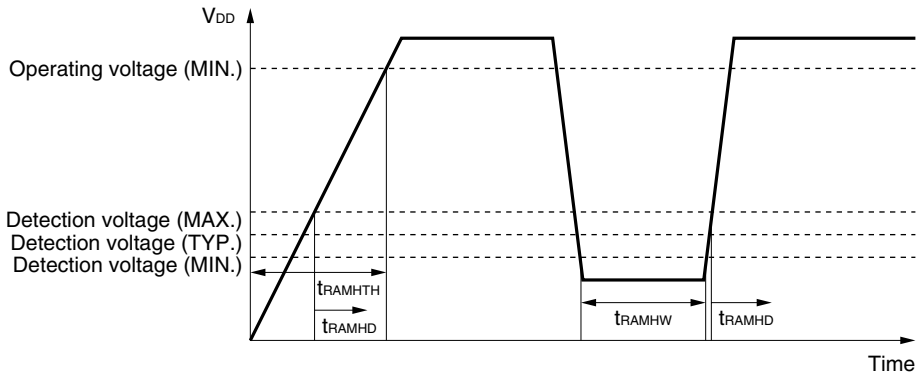


(11) RAM retention flag characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = 1.9 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{RAMH}		1.9	2.0	2.1	V
Supply voltage rise time	t_{RAMHTH}	V_{DD} = 0 V → 3.3 V	0.002		1800	ms
Response time[Note]	t_{RAMHD}	After the supply voltage reaches the detection voltage (MAX.)		0.2	2.0	ms
Minimum V_{DD} width	t_{RAMHW}		0.2			ms

Note Time required to set the RAMF bit after the detection voltage is detected.



30.9 Flash Memory Programming Characteristics

(1) Basic characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD}, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operating frequency	f_{CPU}		4		32	MHz
Supply voltage	V_{DD}		3.8		5.5	V
Number of writes	C_{WRT} Note				100	Times
Input voltage, high	V_{IH}	FLMD0	0.8EV_{DD}		EV_{DD}	V
Input voltage, low	V_{IL}	FLMD0	EV_{SS}		0.2EV_{DD}	V
Programming temperature	t_{PRG}		–40		+85	°C

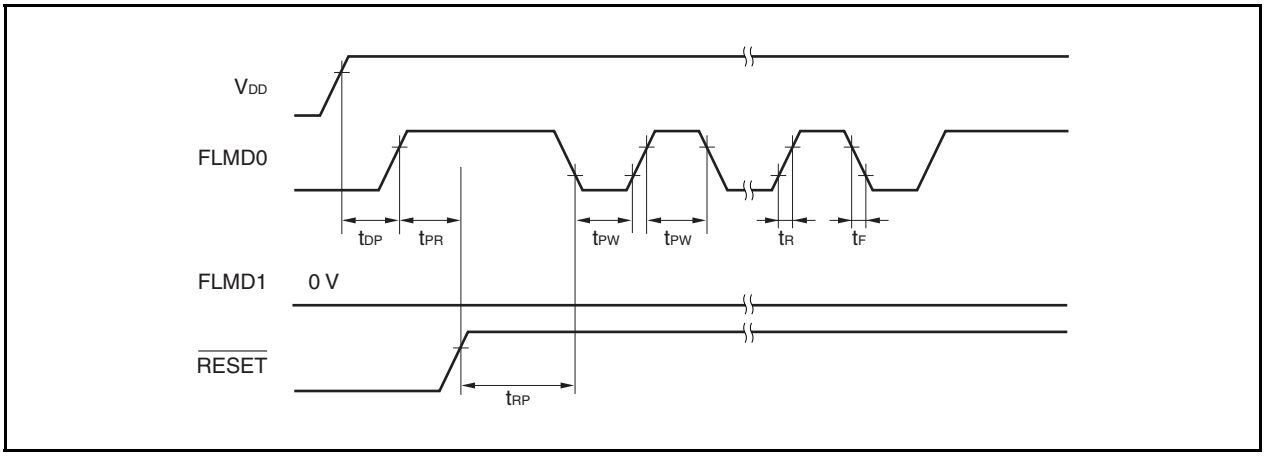
Note When writing initially to shipped products, it is counted as one rewrite for both “erase to write” and “write only”.

Example (P: Write, E: Erase)
Shipped product → P → E → P → E → P: 3 rewrites
Shipped product → E → P → E → P → E → P: 3 rewrites

(2) Serial write operation characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD}, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Setup time from VDD↑ to FLMD0↑	t_{DP}		1			ms
Time from FLMD0↑ to RESET release	t_{PR}		2			ms
FLMD0 pulse input start time from RESET↑	t_{RP}		800			μs
FLMD0 pulse high-level width/ low-level width	t_{PW}		10		100	μs
<R> FLMD0 rise time	t_R				1	μs
<R> FLMD0 fall time	t_F				1	μs



CHAPTER 31 ELECTRICAL SPECIFICATIONS (V850ES/HG3)

31.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}	$V_{DD} = EV_{DD} = BV_{DD}$	−0.5 to +6.5	V
	BV_{DD}	$V_{DD} = EV_{DD} = BV_{DD}$	−0.5 to +6.5	V
	EV_{DD}	$V_{DD} = EV_{DD} = BV_{DD}$	−0.5 to +6.5	V
	AV_{REF0}		−0.5 to +6.5	V
	V_{SS}	$V_{SS} = EV_{SS} = BV_{SS} = AV_{SS}$	−0.5 to +0.5	V
	AV_{SS}	$V_{SS} = EV_{SS} = BV_{SS} = AV_{SS}$	−0.5 to +0.5	V
	BV_{SS}	$V_{SS} = EV_{SS} = BV_{SS} = AV_{SS}$	−0.5 to +0.5	V
	EV_{SS}	$V_{SS} = EV_{SS} = BV_{SS} = AV_{SS}$	−0.5 to +0.5	V
Input voltage	V_{I1}	P00 to P06, P10, P11, P30 to P39, P40 to P42, P50 to P55, P90 to P915, $\overline{\text{RESET}}$, FLMD0	−0.5 to EV_{DD} + 0.5[Note]	V
	V_{I2}	PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL13	−0.5 to BV_{DD} + 0.5[Note]	V
	V_{I3}	X1, X2, XT1, XT2	−0.5 to V_{RO} + 0.5	V
Analog input voltage	V_{IAN}	P70 to P715	−0.5 to AV_{REF0} + 0.5[Note]	V

Note Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

Cautions 1. Do not directly connect the output (or I/O) pins of IC products to each other, or to V_{DD}, 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 and AC characteristics represent the quality assurance range during normal operation.

Remark Unless specified otherwise, the characteristics 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}	P00 to P06, P10, P11, P30 to P39, P40 to P42, P50 to P55, P90 to P915	Per pin	4	mA
			Total of all pins	50	mA
		P70 to P715	Per pin	4	mA
			Total of all pins	20	mA
		PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL13	Per pin	4	mA
			Total of all pins	50	mA
Output current, high	I_{OH}	P00 to P06, P10, P11, P30 to P39, P40 to P42, P50 to P55, P90 to P915	Per pin	−4	mA
			Total of all pins	−50	mA
		P70 to P715	Per pin	−4	mA
			Total of all pins	−20	mA
		PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL13	Per pin	−4	mA
			Total of all pins	−50	mA
Operating ambient temperature	T_A	Normal operation mode		−40 to +85	°C
		Flash memory programming mode			
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 and AC characteristics represent the quality assurance range during normal operation.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

31.2 Capacitance

(T_A = 25°C, V_{DD} = EV_{DD} = BV_{DD} = AV_{REF0} = V_{SS} = EV_{SS} = BV_{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

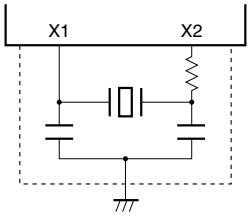
(T_A = –40 to +85°C, V_{DD} = EV_{DD}= BV_{DD}, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C = 4.7 μF)

Internal System Clock Frequency	Conditions	Supply Voltage		Unit
		V_{DD}, EV_{DD}, BV_{DD}	AV_{REF0}	
4 MHz ≤ f_{XX} ≤ 32 MHz		4.0 to 5.5	4.0 to 5.5	V
4 MHz ≤ f_{XX} ≤ 20 MHz	AD converter operating	3.7 to 5.5	4.0 to 5.5	V
	AD converter stop	3.7 to 5.5	3.7 to 5.5	V
32 kHz ≤ f_{XT} ≤ 35 kHz		3.7 to 5.5	3.7 to 5.5	V
f_{RL} = 240 kHz (TYP.)		3.7 to 5.5	3.7 to 5.5	V
f_{RH} = 8 MHz (TYP.)	AD converter operating	3.7 to 5.5	4.0 to 5.5	V
	AD converter stop	3.7 to 5.5	3.7 to 5.5	V

31.4 Oscillator Characteristics

31.4.1 Main clock oscillator characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions		MIN.	TYP.	MAX.	Unit
Ceramic resonator/ crystal resonator		Oscillation frequency (f_X)[Note 1]			4		16	MHz
		Oscillation stabilization time[Note 2]	After STOP mode release	PLL stopped	54[Note 3]	**Note 4**		μs
				PLL operating	1600[Note 5]	**Note 4**		μs
				SSCG operating	2000[Note 6]	**Note 4**		μs
			After IDLE2 mode release	PLL stopped	54[Note 3]	**Note 4**		μs
				PLL operating	800[Note 5]	**Note 4**		μs
				SSCG operating	1000[Note 6]	**Note 4**		μs

Notes 1. Indicates only oscillator characteristics.

2. Time required from start of oscillation until the resonator stabilizes.

3. Time required to stabilize access to the internal flash memory.

4. The value differs depending on the OSTS register settings.

5. PLL lockup time

6. SSCG lockup time

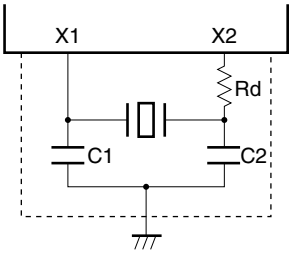
Cautions 1. When using the main clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **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. When the main clock is stopped and the subclock is operating, wait until the oscillation stabilization time has been secured by the program before switching back to the main clock.

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.

<R> **(i) KYOCERA KINSEKI CORPORATION: Crystal resonator (T_A = –10 to +70°C)**

Type	Circuit Example	Part Number	Oscillation Frequency f_X (MHz)	Recommended Circuit Constant		
				C1 (pF)	C2 (pF)	Rd (kΩ)
Lead		HC49SFWB06000D0PESZZ	6.000	8	8	0
		HC49SFWB08000D0PESZZ	8.000	8	8	0
		HC49SFWB10000D0PESZZ	10.000	8	8	0
Surface mounting		HC49GFWB06000D0PESZZ	6.000	8	8	0
		CX1255GB06000D0PESZZ	6.000	8	8	0
		CX8045GB06000D0PESZZ	6.000	8	8	0
		HC49GFWB08000D0PESZZ	8.000	8	8	0
		CX1255GB08000D0PESZZ	8.000	8	8	0
		CX8045GB08000D0PESZZ	8.000	8	8	0
		CX5032GB08000D0PESZZ	8.000	8	8	0
		HC49GFWB10000D0PESZZ	10.000	8	8	0
		CX1255GB10000D0PESZZ	10.000	8	8	0
		CX8045GB10000D0PESZZ	10.000	8	8	0
		CX5032GB10000D0PESZZ	10.000	8	8	0

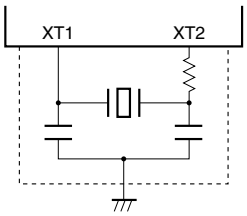
Caution This oscillator constant is a reference value based on evaluation under a specific environment by the resonator manufacturer.

If optimization of oscillator characteristics is necessary in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

The oscillation voltage and oscillation frequency indicate only oscillator characteristics. Use the V850ES/HG3 so that the internal operating conditions are within the specifications of the DC and AC characteristics.

31.4.2 Subclock oscillator characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	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. Indicates only oscillator characteristics. For the CPU operation clock, see **31.7 AC Characteristics**.

2. Time required from when V_{DD} reaches oscillation voltage range (MIN.: 3.3 V) to when the oscillation 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 current 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°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
<R> Input frequency	f_X		6		12	MHz
Output frequency	f_{XX}		12		32	MHz
Lock time	t_{PLL}	After V_{DD} reaches MIN.: 3.3 V			800	μs

31.4.4 SSCG characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input frequency	f_X		4		16	MHz
Output frequency[Note]	f_{XX}		12		32	MHz
Lock time	t_{SSCG}	After V_{DD} reaches MIN.: 3.3 V			1000	μs

Note Indicates the characteristics of the SSCG output frequency when it is not modulated. The modulated operating frequency is as follows, depending on the settings of the SFC1.SCF16 to SFC1.SFC14 bits. Make sure that the maximum operating frequency does not exceed 32 MHz, taking the maximum value of the modulation rate of the operating frequency into consideration.

SFC1.SFC16 to SFC1.SFC14	Modulation Rate		Operating Frequency
	TYP.	MAX.	
000B	±0.5 %	±2.0 %	31.3 MHz
001B	±1.0 %	±2.5 %	31.2 MHz
010B	±2.0 %	±4.0 %	30.7 MHz
011B	±3.0 %	±6.0 %	30.0 MHz
100B	±4.0 %	±8.0 %	29.4 MHz
101B	±5.0 %	±10.0 %	28.8 MHz

31.4.5 Low-speed internal oscillator/high-speed internal oscillator characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

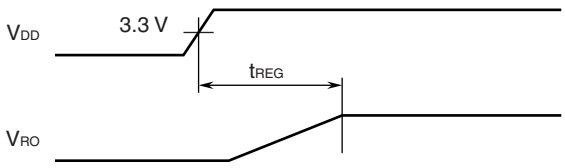
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output frequency	f_{RL}	Low-speed internal oscillator	204	240	276	kHz
	f_{RH}	High-speed internal oscillator	7.2	8.0	8.8	MHz
Oscillation stabilization time		High-speed internal oscillator operating			256	μs

31.5 Voltage Regulator Characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD}, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage	V_{DD}		3.3		5.5	V
Output voltage	V_{RO}			2.5		V
Output voltage stabilization time	t_{REG}	After V_{DD} reaches MIN.: 3.3 V, C = 4.7 μF connected to REGC pin			1	ms

Remark t_{REG} is secured by the POC function. Then reset is released.



31.6 DC Characteristics

31.6.1 I/O level

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

(1/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P04, P30, P31, P34, P36 to P39, P40, P91, P911, P913 to P915	$0.7EV_{DD}$		EV_{DD}	V
	V_{IH2}	P00 to P03, P05, P06, P10, P11, P32, P33, P35, P41 P42, P50 to P55, P90, P92 to P910, P912	$0.8EV_{DD}$		EV_{DD}	V
		PDL0 to PDL13	$0.8BV_{DD}$		BV_{DD}	V
	V_{IH3}	PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6	$0.7BV_{DD}$		BV_{DD}	V
	V_{IH4}	P70 to P715	$0.7AV_{REF0}$		AV_{REF0}	V
	V_{IH5}	$\overline{RESET}$, FLMD0	$0.8EV_{DD}$		EV_{DD}	V
Input voltage, low	V_{IL1}	P04, P30, P31, P34, P36 to P39, P40, P91, P911, P913 to P915	EV_{SS}		$0.3EV_{DD}$	V
	V_{IL2}	P00 to P03, P05, P06, P10, P11, P32, P33, P35, P41 P42, P50 to P55, P90, P92 to P910, P912	EV_{SS}		$0.4EV_{DD}$	V
		PDL0 to PDL13	BV_{SS}		$0.4BV_{DD}$	V
	V_{IL3}	PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6	BV_{SS}		$0.3BV_{DD}$	V
	V_{IL4}	P70 to P715	AV_{SS}		$0.3AV_{REF0}$	V
	V_{IL5}	$\overline{RESET}$, FLMD0	EV_{SS}		$0.2EV_{DD}$	V

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} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

(2/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high[Note 1]	V_{OH1}	P00 to P06, P10, P11, P30 to P39, P40 to P42, P50 to P55, P90 to P915	I_{OH} = –1.0 mA	EV_{DD} – 1.0		EV_{DD}	V
			I_{OH} = –0.1 mA	EV_{DD} – 0.5		EV_{DD}	V
	V_{OH2}	PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL13	I_{OH} = –1.0 mA	BV_{DD} – 1.0		BV_{DD}	V
			I_{OH} = –0.1 mA	BV_{DD} – 0.5		BV_{DD}	V
	V_{OH3}	P70 to P715	I_{OH} = –1.0 mA	AV_{REF0} – 1.0		AV_{REF0}	V
			I_{OH} = –0.1 mA	AV_{REF0} – 0.5		AV_{REF0}	V
Output voltage, low[Note 1]	V_{OL1}	P00 to P06, P10, P11, P30 to P39, P40 to P42, P50 to P55, P90 to P913	I_{OL} = 1.0 mA	0		0.4	V
		P914, P915	I_{OL} = 3.0 mA	0		0.4	V
	V_{OL2}	PCM0 to PCM3, PCS0, PCS1, PCT0, PCT1, PCT4, PCT6, PDL0 to PDL13	I_{OL} = 1.0 mA	0		0.4	V
	V_{OL3}	P70 to P715	I_{OL} = 1.0 mA	0		0.4	V
Pull-up resistor	R_1	V_I = 0 V		10	30	100	kΩ
Pull-down resistor[Note 2]	R_2	V_I = V_{DD}		10	30	100	kΩ

Notes 1. The maximum value of the total of I_{OH}/I_{OL} is 20 mA/–20 mA for each power supply (EV_{DD}, BV_{DD}, AV_{REF0}).

2. $\overline{\text{DRST}}$ pin only

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

31.6.2 Pin leakage current

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	V_{IN} = V_{DD}	Analog pin			0.2	μA
			FLMD0 pin			2.0	
			Other than above pin			0.5	
Input leakage current, low	I_{LIL1}	V_{IN} = 0 V	Analog pin			–0.2	μA
			FLMD0 pin			–2.0	
			Other than above pin			–0.5	
Output leakage current, high	I_{LOH1}	V_O = V_{DD}	Analog pin			0.2	μA
			Other than analog pin			0.5	
Output leakage current, low	I_{LOL1}	V_O = 0 V	Analog pin			–0.2	μA
			Other than analog pin			–0.5	

<R> **31.6.3 Supply current**

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Normal operation mode[Note 2]	PLL operating	f_{XX} = 32 MHz (f_X = 8 MHz)	All peripheral function operating		40	53	mA
					All peripheral function stopped		32		mA
			PLL operating	f_{XX} = 20 MHz (f_X = 10 MHz)	All peripheral function operating		28	38	mA
					All peripheral function stopped		22		mA
			PLL stopped	f_{XX} = 16 MHz (f_X = 16 MHz)	All peripheral function operating		22	30	mA
					All peripheral function stopped		19		mA
	I_{DD2}	HALT mode[Note 2]	PLL operating	f_{XX} = 32 MHz (f_X = 8 MHz)	All peripheral function operating		27	39	mA
					All peripheral function stopped		18		mA
			PLL operating	f_{XX} = 20 MHz (f_X = 10 MHz)	All peripheral function operating		18	26	mA
					All peripheral function stopped		12		mA
			PLL stopped	f_{XX} = 16 MHz (f_X = 16 MHz)	All peripheral function operating		13	20	mA
					All peripheral function stopped		9		mA
	I_{DD3}	IDLE1 mode	PLL stopped[Note 3]	f_{XX} = 16 MHz (f_X = 16 MHz)	TAA, UARTD operating		3.3	4.7	mA
					All peripheral function stopped		1.6		mA
				f_{XX} = 8 MHz (f_X = 8 MHz)	TAA, UARTD operating		2.1	3.0	mA
					All peripheral function stopped		1.3		mA
			f_{XX} = High-speed internal oscillation (f_{RH})[Note 4]		TAA, UARTD operating		1.5	2.3	mA
					All peripheral function stopped		1.1		mA
	I_{DD4}	IDLE2 mode	PLL stopped[Note 3]	f_{XX} = 16 MHz (f_X = 16 MHz)			0.8	1.2	mA
				f_{XX} = 8 MHz (f_X = 8 MHz)			0.5	0.8	mA
			f_{XX} = High-speed internal oscillation (f_{RH})[Note 4]				0.2	0.5	mA
	I_{DD5}	Subclock operation mode[Notes 4, 5]	Crystal resonator (f_{XT} = 32.768 kHz)				80	400	μA
	I_{DD6}	Sub-IDLE mode[Notes 4, 5]	Crystal resonator (f_{XT} = 32.768 kHz)				20	190	μA
	I_{DD7}	STOP mode[Notes 4, 6]	Low-speed internal oscillator (f_{RL}) operating				18.5	100	μA
			Low-speed internal oscillator (f_{RL}) stopped				10.5	85	μA

Notes 1. Total current of V_{DD}, EV_{DD}, and BV_{DD} (all ports stopped). The current of AV_{REF0} and the port buffer current including the current flowing through the on-chip pull-up/pull-down resistors are not included.

2. During SSCG operation, TYP. value + 2.5 mA, MAX. value + 4 mA

3. High-speed internal oscillator (f_{RH}) stopped.

4. When the main clock oscillator (f_{XX}) is stopped.

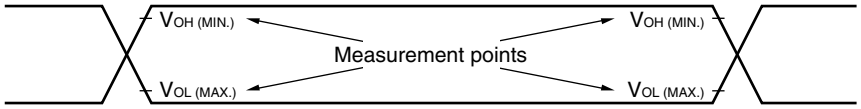
5. Low-speed internal oscillator (f_{RL}) operating, high-speed internal oscillator (f_{RH}) stopped.

6. When the subclock oscillator (f_{XT}) is not used.

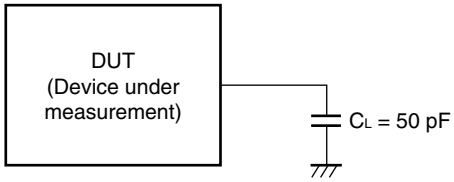
31.7 AC Characteristics

(1) AC test input measurement points (V_{DD}, AV_{REF0}, EV_{DD}, BV_{DD})

(2) AC test output measurement points



(3) Load conditions

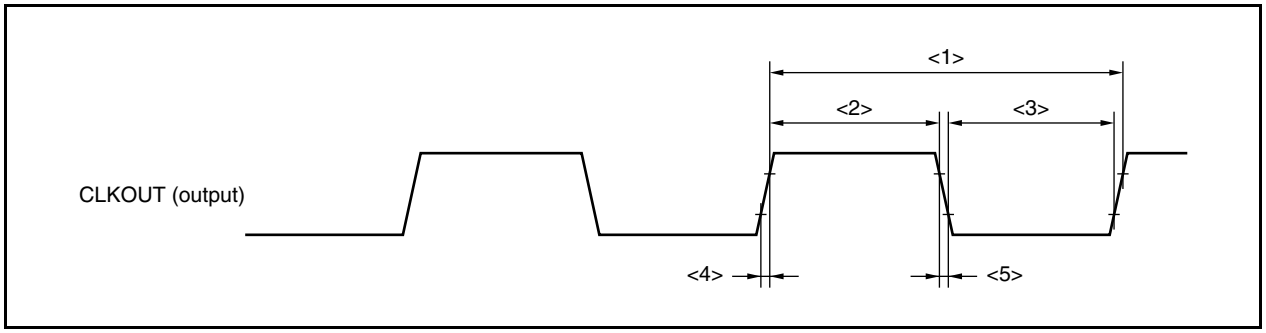


Caution If the load capacitance exceeds 50 pF due to the circuit configuration, bring the load capacitance of the device to 50 pF or less by inserting a buffer or by some other means.

31.7.1 CLKOUT output timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output cycle	t_{CYK}	<1>	V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V	50 ns	80 μs	
			V_{DD} = EV_{DD} = BV_{DD} = 4.0 V to 5.5 V	31.25 ns	80 μs	
High-level width	t_{WKH}	<2>	V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V	t_{CYK}/2 – 15		ns
			V_{DD} = EV_{DD} = BV_{DD} = 4.0 V to 5.5 V	t_{CYK}/2 – 13		ns
Low-level width	t_{WKL}	<3>	V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V	t_{CYK}/2 – 15		ns
			V_{DD} = EV_{DD} = BV_{DD} = 4.0 V to 5.5 V	t_{CYK}/2 – 13		ns
Rise time	t_{KR}	<4>	V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V		15	ns
			V_{DD} = EV_{DD} = BV_{DD} = 4.0 V to 5.5 V		13	ns
Fall time	t_{KF}	<5>	V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V		15	ns
			V_{DD} = EV_{DD} = BV_{DD} = 4.0 V to 5.5 V		13	ns



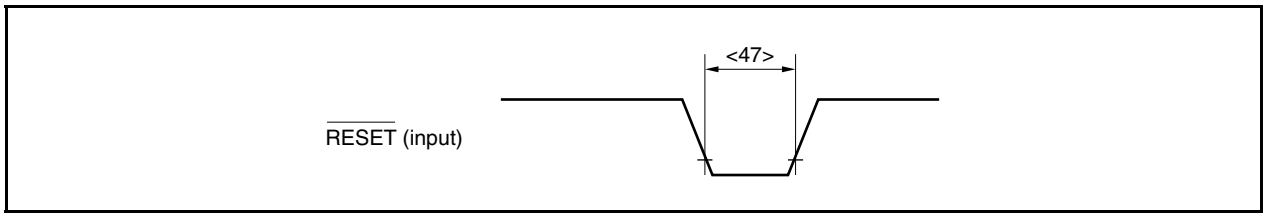
31.8 Basic Operation

(1) Reset timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{RESET}$ low-level width	t_{WRSL}	<47>		250		ns

Reset



(2) Interrupt timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

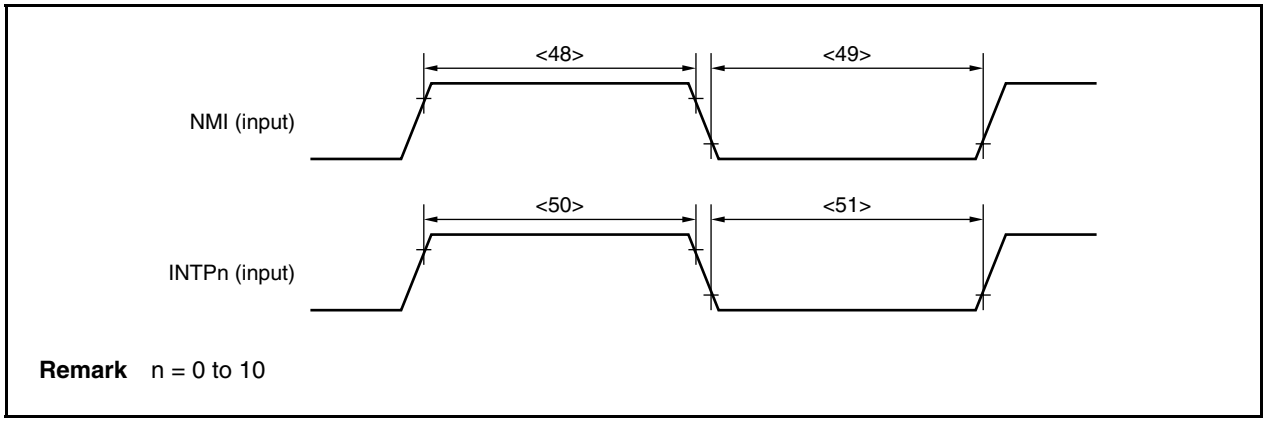
Parameter	Symbol		Conditions	MIN.	MAX.	Unit
NMI high-level width	t_{WNIH}	<48>	Analog noise elimination	250		ns
NMI low-level width	t_{WNIL}	<49>	Analog noise elimination	250		ns
INTPn[Note 1] high-level width	t_{WITH}	<50>	Analog noise elimination (n = 0 to 10)	250		ns
			Digital noise elimination (n = 3)	**Note 2**		ns
INTPn[Note 1] low-level width	t_{WITL}	<51>	Analog noise elimination (n = 0 to 10)	250		ns
			Digital noise elimination (n = 3)	**Note 2**		ns

Notes 1. The same value as the INTP0/P03 pin applies in the case of the ADTRG pin. The same value as the INTP2/P05 pin applies in the case of the $\overline{DRST}$ pin.

2. $2T_{samp}$ + 20 or $3T_{samp}$ + 20

T_{samp}: Sampling clock for noise elimination

Reset/interrupt

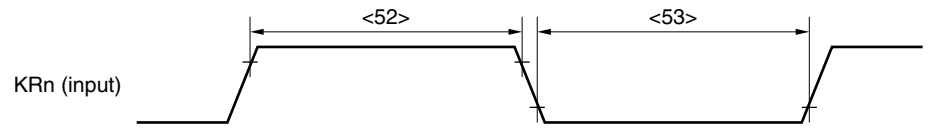


Remark n = 0 to 10

(3) Key interrupt timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
KRn input high-level width	t_{WKRH}	<52>	Analog noise elimination (n = 0 to 7)	250		ns
KRn input low-level width	t_{WKRL}	<53>		250		ns



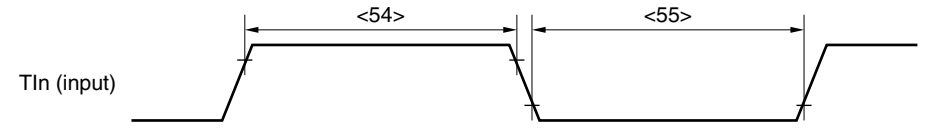
Remark n = 0 to 7

(4) Timer input timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions		MIN.	MAX.	Unit
TIn high-level width	t_{TIH}	<54>	TIAA00, TIAA01, TIAA10, TIAA11, TIAA20, TIAA21, TIAA30, TIAA31, TIAA40, TIAA41, TIAB00 to TIAB03, TIAB10 to TIAB13[Note]		250		ns
TIn low-level width	t_{TIL}	<55>			250		ns
TOn output cycle	f_{TCYK}		TOAA00, TOAA01, TOAA10, TOAA11, TOAA20, TOAA21, TOAA30, TOAA31, TOAA40, TOAA41, TOAB00 to TOAB03, TOAB10 to TOAB13	4.0 V ≤ V_{DD} ≤ 5.5 V		16	MHz
				3.7 V ≤ V_{DD} < 4.0 V		10	MHz

Note Noise on the TIAA00, TIAA10, TIAA20, TIAA30, TIAB00, and TIAB10 pins can be eliminated only when a capture signal is input.
The noise cannot be eliminated when an external trigger signal or an external event counter signal is input.



Remark TIn: TIAA00, TIAA01, TIAA10, TIAA11, TIAA20, TIAA21, TIAA30, TIAA31, TIAA40, TIAA41, TIAB00 to TIAB03, TIAB10 to TIAB13

(5) CSIB timing

(a) Master mode

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKBn}$ cycle time	t_{KCY1}	<56>		125		ns
$\overline{SCKBn}$ high-level width	t_{KH1}	<57>		$t_{KCY1}/2 - 15$		ns
$\overline{SCKBn}$ low-level width	t_{KL1}	<58>		$t_{KCY1}/2 - 15$		ns
SIBn setup time (to $\overline{SCKBn}\uparrow$)	t_{SIK1}	<59>		30		ns
SIBn hold time (from $\overline{SCKBn}\uparrow$)	t_{KSI1}	<60>		25		ns
Output delay time from $\overline{SCKBn}\downarrow$ to SOBn	t_{KSO1}	<61>			25	ns

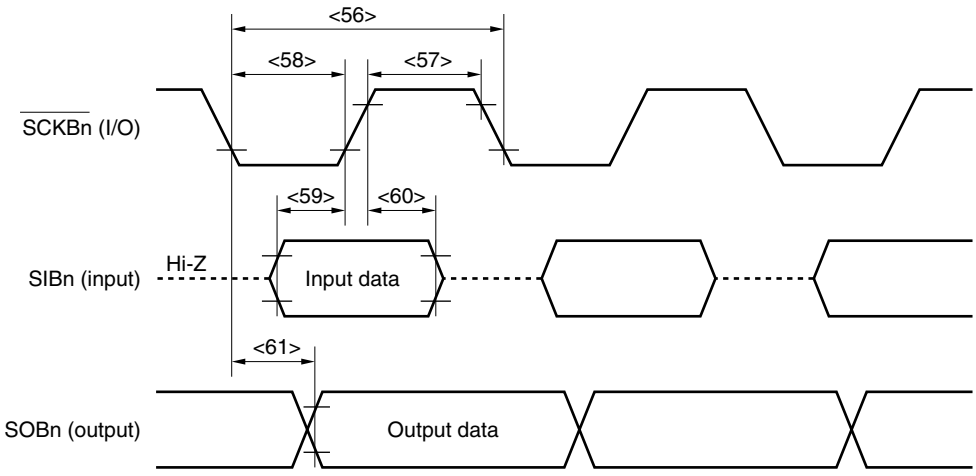
Remark n = 0, 1

(b) Slave mode

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKBn}$ cycle time	t_{KCY1}	<56>		200		ns
$\overline{SCKBn}$ high-level width	t_{KH1}	<57>		90		ns
$\overline{SCKBn}$ low-level width	t_{KL1}	<58>		90		ns
SIBn setup time (to $\overline{SCKBn}\uparrow$)	t_{SIK1}	<59>		50		ns
SIBn hold time (from $\overline{SCKBn}\uparrow$)	t_{KSI1}	<60>		50		ns
Output delay time from $\overline{SCKBn}\downarrow$ to SOBn	t_{KSO1}	<61>			50	ns

Remark n = 0, 1



Remark n = 0, 1

(6) UARTD timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Communication rate				1.5	Mbps
ASCK0 cycle time				10	MHz

(7) I^2C bus timing

(T_A = –40 to +85°C, V_{DD} = V_{DD1} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = V_{SS1} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter		Symbol		Normal Mode		High-Speed Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCL00 clock frequency		f_{CLK}		0	100	0	400	kHz
Bus free time (Between start and stop conditions)		t_{BUF}	<62>	4.7	–	1.3	–	μs
Hold time[Note 1]		$t_{HD:STA}$	<63>	4.0	–	0.6	–	μs
SCL00 clock low-level width		t_{LOW}	<64>	4.7	–	1.3	–	μs
SCL00 clock high-level width		t_{HIGH}	<65>	4.0	–	0.6	–	μs
Setup time for start/restart conditions		$t_{SU:STA}$	<66>	4.7	–	0.6	–	μs
Data hold time	CBUS compatible master	$t_{HD:DAT}$	<67>	5.0	–	–	–	μs
	I^2C mode			0[Note 2]	–	0[Note 2]	0.9[Note 3]	μs
Data setup time		$t_{SU:DAT}$	<68>	250	–	100[Note 4]	–	ns
SDA00 and SCL00 signal rise time		t_R	<69>	–	1000	20 + 0.1Cb[Note 5]	300	ns
SDA00 and SCL00 signal fall time		t_F	<70>	–	300	20 + 0.1Cb[Note 5]	300	ns
Stop condition setup time		$t_{SU:STO}$	<71>	4.0	–	0.6	–	μs
Pulse width of spike suppressed by input filter		t_{SP}	<72>	–	–	0	50	ns
Capacitance load of each bus line		Cb		–	400	–	400	pF

Notes 1. At the start condition, the first clock pulse is generated after the hold time.

2. The system requires a minimum of 300 ns hold time internally for the SDA00 signal (at $V_{IHmin.}$ of SCL00 signal) in order to occupy the undefined area at the falling edge of SCL00.

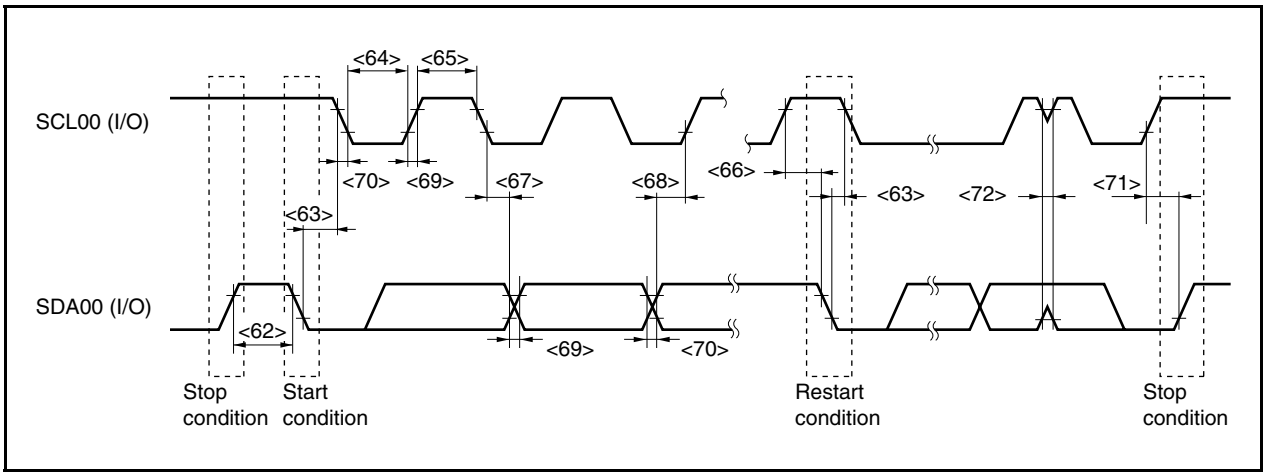
3. If the system does not extend the SCL00 signal low hold time (t_{LOW}), only the maximum data hold time ($t_{HD:DAT}$) needs to be satisfied.

4. The high-speed mode I^2C bus can be used in the normal-mode I^2C bus system. In this case, set the high-speed mode I^2C bus so that it meets the following conditions.

- If the system does not extend the SCL00 signal's low state hold time:
  $t_{SU:DAT} \geq 250$ ns
- If the system extends the SCL00 signal's low state hold time:
  Transmit the following data bit to the SDA00 line prior to the SCL00 line release ($t_{Rmax.} + t_{SU:DAT}$ = 1,000 + 250 = 1,250 ns: Normal mode I^2C bus specification).

5. Cb: Total capacitance of one bus line (unit: pF)

I²C bus mode



(8) A/D converter

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution					10	bit
Overall error[Note 1]		4.0 ≤ AV_{REF0} ≤ 5.5 V		±0.15	±0.3	%FSR
Conversion time	t_{CONV}		3.1		16	μs
Stabilization time	t_{STA}	After ADA0M0.ADA0PS bit changes from 0 to 1	2			μs
Power down recovery time	t_{DPU}	Starting operation after STOP mode is released	1			μs
Zero scale error[Note 1]	ZSE				±0.3	%FSR
Full scale error[Note 1]	FSE				±0.3	%FSR
Non-linearity error[Note 2]	INL				±2.5	LSB
Differential linearity error[Note 2]	DNL				±1.5	LSB
Analog input voltage	V_{IAN}		AV_{SS}		AV_{REF0}	V
AV_{REF0} current	I_{AREF0}	When using A/D converter		4	7	mA
		When not using A/D converter		1	10	μA

Notes 1. Excluding quantization error (±0.05 %FSR). Indicates the ratio to the full-scale value (%FSR).

2. Quantization error (±0.5LSB)

Remark FSR: Full Scale Range

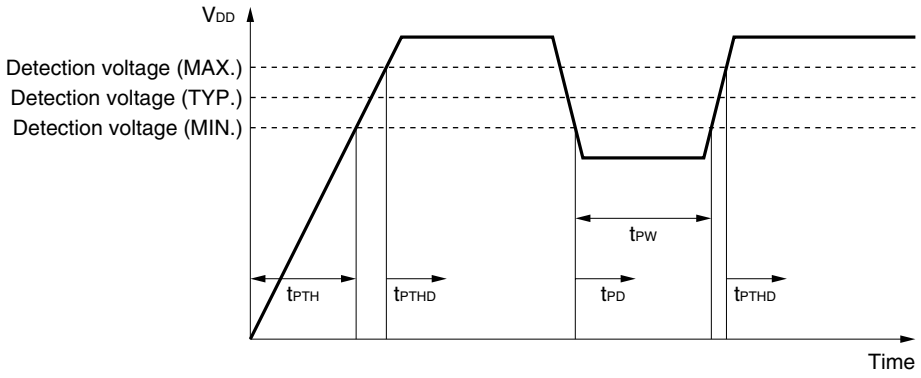
(9) POC circuit characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD}, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC0}		3.3	3.5	3.7	V
Power supply startup time	t_{PTH}	V_{DD} = 0 V → 3.3 V	0.002			ms
Response delay time 1 Note 1	t_{PTHD}	After V_{DD} reaches 3.7 V on power application			2.0	ms
Response delay time 2 Note 2	t_{PD}	After V_{DD} drops below 3.3 V on power drop			1.0	ms
Minimum V_{DD} width	t_{PW}		0.2			ms

Notes 1. The time required to release a reset after the detection voltage is detected.

2. The time required to output a reset after the detection voltage is detected.

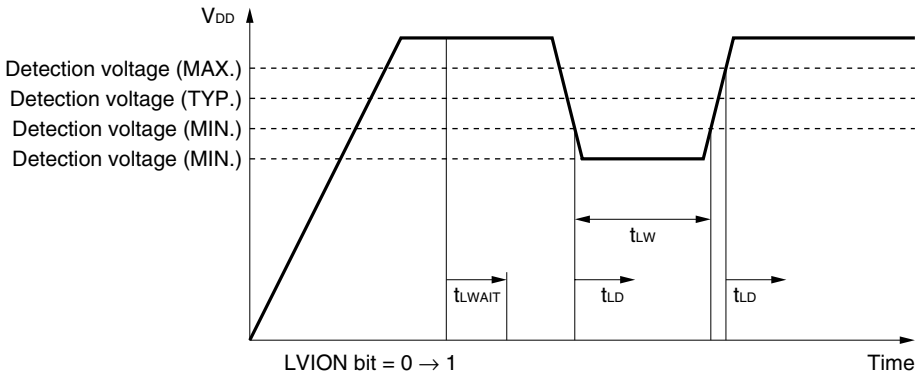


(10) LVI circuit characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LVI0}		3.8	4.0	4.2	V
	V_{LVI1}		3.5	3.7	3.9	V
Response time[Note]	t_{LD}	After V_{DD} reaches V_{LVI0}/V_{LVI1} (MAX.) or drops below V_{LVI0}/V_{LVI1} (MIN.)		0.2	2.0	ms
Minimum V_{DD} width	t_{LW}		0.2			ms
Reference voltage stabilization wait time	t_{LWAIT}	After LVION bit (LVIM.bit7) changes from 0 to 1		0.1	0.2	ms

Note The time required to output an interrupt/reset after the detection voltage is detected.

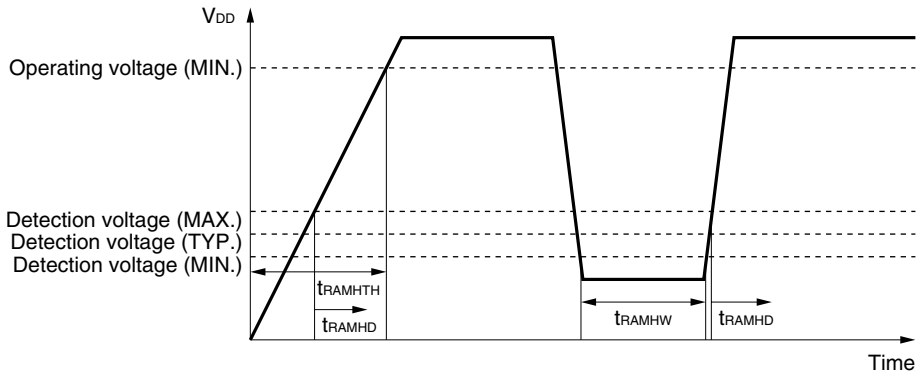


(11) RAM retention flag characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 1.9 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{RAMH}		1.9	2.0	2.1	V
Supply voltage rise time	t_{RAMHTH}	V_{DD} = 0 V → 3.3 V	0.002		1800	ms
Response time[Note]	t_{RAMHD}	After the supply voltage reaches the detection voltage (MAX.)		0.2	2.0	ms
Minimum V_{DD} width	t_{RAMHW}		0.2			ms

Note Time required to set the RAMF bit after the detection voltage is detected.



31.9 Flash Memory Programming Characteristics

(1) Basic characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD}, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operating frequency	f_{CPU}		4		32	MHz
Supply voltage	V_{DD}		3.8		5.5	V
Number of writes	C_{WRT} Note				100	Times
Input voltage, high	V_{IH}	FLMD0	0.8EV_{DD}		EV_{DD}	V
Input voltage, low	V_{IL}	FLMD0	EV_{SS}		0.2EV_{DD}	V
Programming temperature	t_{PRG}		–40		+85	°C

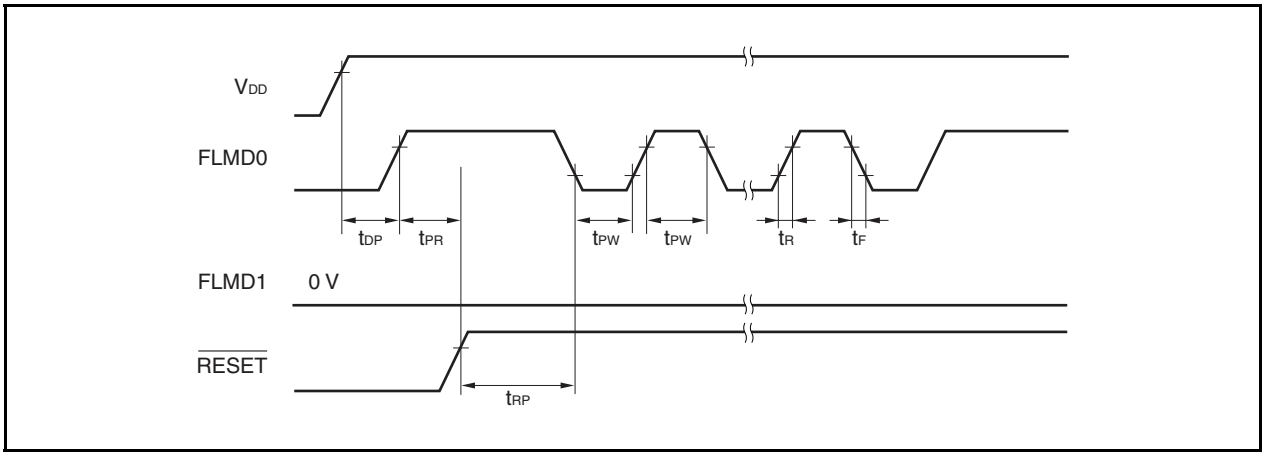
Note When writing initially to shipped products, it is counted as one rewrite for both “erase to write” and “write only”.

Example (P: Write, E: Erase)
Shipped product → P → E → P → E → P: 3 rewrites
Shipped product → E → P → E → P → E → P: 3 rewrites

(2) Serial write operation characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD}, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Setup time from VDD↑ to FLMD0↑	t_{DP}		1			ms
Time from FLMD0↑ to RESET release	t_{PR}		2			ms
FLMD0 pulse input start time from RESET↑	t_{RP}		800			μs
FLMD0 pulse high-level width/ low-level width	t_{PW}		10		100	μs
<R> FLMD0 rise time	t_R				1	μs
<R> FLMD0 fall time	t_F				1	μs



32.1 Absolute Maximum Ratings

Absolute Maximum Ratings (T_A = 25°C) (1/2)

Parameter	Symbol	Conditions	Ratings	Unit
Supply voltage	V_{DD}	$V_{DD} = EV_{DD} = BV_{DD}$	−0.5 to +6.5	V
	BV_{DD}	$V_{DD} = EV_{DD} = BV_{DD}$	−0.5 to +6.5	V
	EV_{DD}	$V_{DD} = EV_{DD} = BV_{DD}$	−0.5 to +6.5	V
	AV_{REF0}		−0.5 to +6.5	V
	V_{SS}	$V_{SS} = EV_{SS} = BV_{SS} = AV_{SS}$	−0.5 to +0.5	V
	AV_{SS}	$V_{SS} = EV_{SS} = BV_{SS} = AV_{SS}$	−0.5 to +0.5	V
	BV_{SS}	$V_{SS} = EV_{SS} = BV_{SS} = AV_{SS}$	−0.5 to +0.5	V
	EV_{SS}	$V_{SS} = EV_{SS} = BV_{SS} = AV_{SS}$	−0.5 to +0.5	V
Input voltage	V_{I1}	P00 to P06, P10, P11, P30 to P39, P40 to P42, P50 to P55, P60 to P615, P80, P81, P90 to P915, $\overline{\text{RESET}}$, FLMD0	−0.5 to EV_{DD} + 0.5[Note]	V
	V_{I2}	PCD0 to PCD3, PCM0 to PCM5, PCS0 to PCS7, PCT0 to PCT7, PDL0 to PDL15	−0.5 to BV_{DD} + 0.5[Note]	V
	V_{I3}	X1, X2, XT1, XT2	−0.5 to V_{RO} + 0.5	V
Analog input voltage	V_{IAN}	P70 to P715, P120 to P127	−0.5 to AV_{REF0} + 0.5[Note]	V

Note Be sure not to exceed the absolute maximum ratings (MAX. value) of each supply voltage.

Cautions 1. Do not directly connect the output (or I/O) pins of IC products to each other, or to V_{DD}, 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 and AC characteristics represent the quality assurance range during normal operation.

Remark Unless specified otherwise, the characteristics 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}	P00 to P06, P10, P11, P30 to P39, P40 to P42, P50 to P55, P60 to P615, P80, P81, P90 to P915	Per pin	4	mA
			Total of all pins	50	mA
		P70 to P715, P120 to P127	Per pin	4	mA
			Total of all pins	20	mA
		PCD0 to PCD3, PCM0 to PCM5, PCS0 to PCS7, PCT0 to PCT7, PDL0 to PDL15	Per pin	4	mA
			Total of all pins	50	mA
Output current, high	I_{OH}	P00 to P06, P10, P11, P30 to P39, P40 to P42, P50 to P55, P60 to P615, P80, P81, P90 to P915	Per pin	−4	mA
			Total of all pins	−50	mA
		P70 to P715, P120 to P127	Per pin	−4	mA
			Total of all pins	−20	mA
		PCD0 to PCD3, PCM0 to PCM5, PCS0 to PCS7, PCT0 to PCT7, PDL0 to PDL15	Per pin	−4	mA
			Total of all pins	−50	mA
Operating ambient temperature	T_A	Normal operation mode		−40 to +85	°C
		Flash memory programming mode			
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 and AC characteristics represent the quality assurance range during normal operation.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

32.2 Capacitance

(T_A = 25°C, V_{DD} = EV_{DD} = BV_{DD} = AV_{REF0} = V_{SS} = EV_{SS} = BV_{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

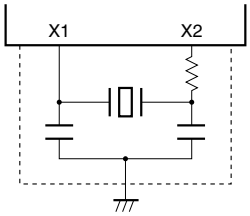
(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD}, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C = 4.7 μF)

Internal System Clock Frequency	Conditions	Supply Voltage		Unit
		V_{DD} = EV_{DD} = BV_{DD}	AV_{REF0}	
4 MHz ≤ f_{XX} ≤ 32 MHz		4.0 to 5.5	4.0 to 5.5	V
4 MHz ≤ f_{XX} ≤ 20 MHz	AD converter operating	3.7 to 5.5	4.0 to 5.5	V
	AD converter stop	3.7 to 5.5	3.7 to 5.5	V
32 kHz ≤ f_{XT} ≤ 35 kHz		3.7 to 5.5	3.7 to 5.5	V
f_{RL} = 240 kHz (TYP.)		3.7 to 5.5	3.7 to 5.5	V
f_{RH} = 8 MHz (TYP.)	AD converter operating	3.7 to 5.5	4.0 to 5.5	V
	AD converter stop	3.7 to 5.5	3.7 to 5.5	V

32.4 Oscillator Characteristics

32.4.1 Main clock oscillator characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions		MIN.	TYP.	MAX.	Unit
Ceramic resonator/ crystal resonator		Oscillation frequency (f_X)[Note 1]			4		16	MHz
		Oscillation stabilization time[Note 2]	After STOP mode release	PLL stopped	54[Note 3]	**Note 4**		μs
				PLL operating	1600[Note 5]	**Note 4**		μs
				SSCG operating	2000[Note 6]	**Note 4**		μs
			After IDLE2 mode release	PLL stopped	54[Note 3]	**Note 4**		μs
				PLL operating	800[Note 5]	**Note 4**		μs
				SSCG operating	1000[Note 6]	**Note 4**		μs

Notes 1. Indicates only oscillator characteristics.

2. Time required from start of oscillation until the resonator stabilizes.

3. Time required to stabilize access to the internal flash memory.

4. The value differs depending on the OSTS register settings.

5. PLL lockup time

6. SSCG lockup time

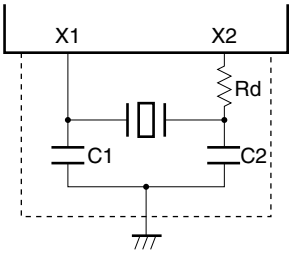
Cautions 1. When using the main clock oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

- **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. When the main clock is stopped and the subclock is operating, wait until the oscillation stabilization time has been secured by the program before switching back to the main clock.

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.

<R> **(i) KYOCERA KINSEKI CORPORATION: Crystal resonator (T_A = –10 to +70°C)**

Type	Circuit Example	Part Number	Oscillation Frequency f_X (MHz)	Recommended Circuit Constant		
				C1 (pF)	C2 (pF)	Rd (kΩ)
Lead		HC49SFWB06000D0PESZZ	6.000	8	8	0
		HC49SFWB08000D0PESZZ	8.000	8	8	0
		HC49SFWB10000D0PESZZ	10.000	8	8	0
Surface mounting		HC49GFWB06000D0PESZZ	6.000	8	8	0
		CX1255GB06000D0PESZZ	6.000	8	8	0
		CX8045GB06000D0PESZZ	6.000	8	8	0
		HC49GFWB08000D0PESZZ	8.000	8	8	0
		CX1255GB08000D0PESZZ	8.000	8	8	0
		CX8045GB08000D0PESZZ	8.000	8	8	0
		CX5032GB08000D0PESZZ	8.000	8	8	0
		HC49GFWB10000D0PESZZ	10.000	8	8	0
		CX1255GB10000D0PESZZ	10.000	8	8	0
		CX8045GB10000D0PESZZ	10.000	8	8	0
		CX5032GB10000D0PESZZ	10.000	8	8	0

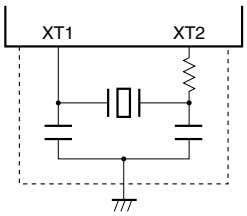
Caution This oscillator constant is a reference value based on evaluation under a specific environment by the resonator manufacturer.

If optimization of oscillator characteristics is necessary in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit.

The oscillation voltage and oscillation frequency indicate only oscillator characteristics. Use the V850ES/HJ3 so that the internal operating conditions are within the specifications of the DC and AC characteristics.

32.4.2 Subclock oscillator characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	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. Indicates only oscillator characteristics. For the CPU operation clock, see **32.7 AC Characteristics**.

2. Time required from when V_{DD} reaches oscillation voltage range (MIN.: 3.3 V) to when the oscillation 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 current 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°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
<R> Input frequency	f_X		6		12	MHz
Output frequency	f_{XX}		12		32	MHz
Lock time	t_{PLL}	After V_{DD} reaches MIN.: 3.3 V			800	μs

32.4.4 SSCG characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input frequency	f_X		4		16	MHz
Output frequency[Note]	f_{XX}		12		32	MHz
Lock time	t_{SSCG}	After V_{DD} reaches MIN.: 3.3 V			1000	μs

Note Indicates the characteristics of the SSCG output frequency when it is not modulated. The modulated operating frequency is as follows, depending on the settings of the SFC1.SCF16 to SFC1.SFC14 bits. Make sure that the maximum operating frequency does not exceed 32 MHz, taking the maximum value of the modulation rate of the operating frequency into consideration.

SFC1.SFC16 to SFC1.SFC14	Modulation Rate		Operating Frequency
	TYP.	MAX.	
000B	±0.5 %	±2.0 %	31.3 MHz
001B	±1.0 %	±2.5 %	31.2 MHz
010B	±2.0 %	±4.0 %	30.7 MHz
011B	±3.0 %	±6.0 %	30.0 MHz
100B	±4.0 %	±8.0 %	29.4 MHz
101B	±5.0 %	±10.0 %	28.8 MHz

32.4.5 Low-speed internal oscillator/high-speed internal oscillator characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

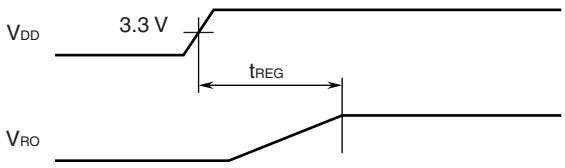
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output frequency	f_{RL}	Low-speed internal oscillator	204	240	276	kHz
	f_{RH}	High-speed internal oscillator	7.2	8.0	8.8	MHz
Oscillation stabilization time		High-speed internal oscillator operating			256	μs

32.5 Voltage Regulator Characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD}, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage	V_{DD}		3.3		5.5	V
Output voltage	V_{RO}			2.5		V
Output voltage stabilization time	t_{REG}	After V_{DD} reaches MIN.: 3.3 V, C = 4.7 μF connected to REGC pin			1	ms

Remark t_{REG} is secured by the POC function. Then reset is released.



32.6 DC Characteristics

32.6.1 I/O level

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

(1/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P04, P30, P31, P34, P36 to P39, P40, P63 to P69, P614, P615, P80, P81, P91, P911, P913 to P915	0.7EV_{DD}		EV_{DD}	V
	V_{IH2}	P00 to P03, P05, P06, P10, P11, P32, P33, P35, P41 P42, P50 to P55, P60 to P62, P610 to P613, P90, P92 to P910, P912	0.8EV_{DD}		EV_{DD}	V
		PDL0 to PDL15	0.8BV_{DD}		BV_{DD}	V
	V_{IH3}	PCD0 to PCD3, PCM0 to PCM5, PCS0 to PCS7, PCT0 to PCT7	0.7BV_{DD}		BV_{DD}	V
	V_{IH4}	P70 to P715, P120 to P127	0.7AV_{REF0}		AV_{REF0}	V
	V_{IH5}	$\overline{RESET}$, FLMD0	0.8EV_{DD}		EV_{DD}	V
Input voltage, low	V_{IL1}	P04, P30, P31, P34, P36 to P39, P40, P63 to P69, P614, P615, P80, P81, P91, P911, P913 to P915	EV_{SS}		0.3EV_{DD}	V
	V_{IL2}	P00 to P03, P05, P06, P10, P11, P32, P33, P35, P41, P42, P50 to P55, P60 to P62, P610 to P613, P90, P92 to P910, P912	EV_{SS}		0.4EV_{DD}	V
		PDL0 to PDL15	BV_{SS}		0.4BV_{DD}	V
	V_{IL3}	PCD0 to PCD3, PCM0 to PCM5, PCS0 to PCS7, PCT0 to PCT7	BV_{SS}		0.3BV_{DD}	V
	V_{IL4}	P70 to P715, P120 to P127	AV_{SS}		0.3AV_{REF0}	V
	V_{IL5}	$\overline{RESET}$, FLMD0	EV_{SS}		0.2EV_{DD}	V

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} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

(2/2)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high[Note 1]	V_{OH1}	P00 to P06, P10, P11, P30 to P39, P40 to P42, P50 to P55, P60 to P615, P80, P81, P90 to P915	I_{OH} = –1.0 mA	EV_{DD} – 1.0		EV_{DD}	V
			I_{OH} = –0.1 mA	EV_{DD} – 0.5		EV_{DD}	V
	V_{OH2}	PCD0 to PCD3, PCM0 to PCM5, PCS0 to PCS7, PCT0 to PCT7, PDL0 to PDL15	I_{OH} = –1.0 mA	BV_{DD} – 1.0		BV_{DD}	V
			I_{OH} = –0.1 mA	BV_{DD} – 0.5		BV_{DD}	V
	V_{OH3}	P70 to P715, P120 to P127	I_{OH} = –1.0 mA	AV_{REF0} – 1.0		AV_{REF0}	V
			I_{OH} = –0.1 mA	AV_{REF0} – 0.5		AV_{REF0}	V
Output voltage, low[Note 1]	V_{OL1}	P00 to P06, P10, P11, P30 to P39, P40 to P42, P50 to P55, P60 to P615, P80, P81, P90 to P913	I_{OL} = 1.0 mA	0		0.4	V
		P914, P915	I_{OL} = 3.0 mA	0		0.4	V
	V_{OL2}	PCD0 to PCD3, PCM0 to PCM5, PCS0 to PCS7, PCT0 to PCT7, PDL0 to PDL15	I_{OL} = 1.0 mA	0		0.4	V
	V_{OL3}	P70 to P715, P120 to P127	I_{OL} = 1.0 mA	0		0.4	V
Pull-up resistor	R_1	V_I = 0 V		10	30	100	kΩ
Pull-down resistor[Note 2]	R_2	V_I = V_{DD}		10	30	100	kΩ

Notes 1. The maximum value of the total of I_{OH}/I_{OL} is 20 mA/–20 mA for each power supply (EV_{DD}, BV_{DD}, AV_{REF0}).

2. $\overline{\text{DRST}}$ pin only

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

32.6.2 Pin leakage current

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	V_{IN} = V_{DD}	Analog pin			0.2	μA
			FLMD0 pin			2.0	
			Other than above pin			0.5	
Input leakage current, low	I_{LIL1}	V_{IN} = 0 V	Analog pin			–0.2	μA
			FLMD0 pin			–2.0	
			Other than above pin			–0.5	
Output leakage current, high	I_{LOH1}	V_O = V_{DD}	Analog pin			0.2	μA
			Other than analog pin			0.5	
Output leakage current, low	I_{LOL1}	V_O = 0 V	Analog pin			–0.2	μA
			Other than analog pin			–0.5	

32.6.3 Supply current

<R> **(1) μPD70F3755**

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Normal operation mode[Note 2]	PLL operating	f_{XX} = 32 MHz (f_X = 8 MHz)	All peripheral function operating		40	53	mA
					All peripheral function stopped		32		mA
			PLL operating	f_{XX} = 20 MHz (f_X = 10 MHz)	All peripheral function operating		28	38	mA
					All peripheral function stopped		22		mA
			PLL stopped	f_{XX} = 16 MHz (f_X = 16 MHz)	All peripheral function operating		22	30	mA
					All peripheral function stopped		19		mA
	I_{DD2}	HALT mode[Note 2]	PLL operating	f_{XX} = 32 MHz (f_X = 8 MHz)	All peripheral function operating		27	39	mA
					All peripheral function stopped		18		mA
			PLL operating	f_{XX} = 20 MHz (f_X = 10 MHz)	All peripheral function operating		18	26	mA
					All peripheral function stopped		12		mA
			PLL stopped	f_{XX} = 16 MHz (f_X = 16 MHz)	All peripheral function operating		13	20	mA
					All peripheral function stopped		9		mA
	I_{DD3}	IDLE1 mode	PLL stopped[Note 3]	f_{XX} = 16 MHz (f_X = 16 MHz)	TAA, UARTD operating		3.3	4.7	mA
					All peripheral function stopped		1.6		mA
				f_{XX} = 8 MHz (f_X = 8 MHz)	TAA, UARTD operating		2.1	3.0	mA
					All peripheral function stopped		1.3		mA
			f_{XX} = High-speed internal oscillation (f_{RH})[Note 4]		TAA, UARTD operating		1.5	2.3	mA
					All peripheral function stopped		1.1		mA
	I_{DD4}	IDLE2 mode	PLL stopped[Note 3]	f_{XX} = 16 MHz (f_X = 16 MHz)			0.8	1.2	mA
				f_{XX} = 8 MHz (f_X = 8 MHz)			0.5	0.8	mA
			f_{XX} = High-speed internal oscillation (f_{RH})[Note 4]				0.2	0.5	mA
	I_{DD5}	Subclock operation mode[Notes 4, 5]	Crystal resonator (f_{XT} = 32.768 kHz)				80	400	μA
	I_{DD6}	Sub-IDLE mode[Notes 4, 5]	Crystal resonator (f_{XT} = 32.768 kHz)				20	190	μA
	I_{DD7}	STOP mode[Notes 4, 6]	Low-speed internal oscillator (f_{RL}) operating				18.5	100	μA
			Low-speed internal oscillator (f_{RL}) stopped				10.5	85	μA

Notes 1. Total current of V_{DD}, EV_{DD}, and BV_{DD} (all ports stopped). The current of AV_{REF0} and the port buffer current including the current flowing through the on-chip pull-up/pull-down resistors are not included.

2. During SSCG operation, TYP. value + 2.5 mA, MAX. value + 4 mA

3. High-speed internal oscillator (f_{RH}) stopped.

4. When the main clock oscillator (f_{XX}) is stopped.

5. Low-speed internal oscillator (f_{RL}) operating, high-speed internal oscillator (f_{RH}) stopped.

6. When the subclock oscillator (f_{XT}) is not used.

<R> **(2) μPD70F3757**

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Normal operation mode[Note 2]	PLL operating	f_{XX} = 32 MHz (f_X = 8 MHz)	All peripheral function operating		41	54	mA
					All peripheral function stopped		32		mA
			PLL operating	f_{XX} = 20 MHz (f_X = 10 MHz)	All peripheral function operating		28	39	mA
					All peripheral function stopped		22		mA
			PLL stopped	f_{XX} = 16 MHz (f_X = 16 MHz)	All peripheral function operating		23	31	mA
					All peripheral function stopped		19		mA
	I_{DD2}	HALT mode[Note 2]	PLL operating	f_{XX} = 32 MHz (f_X = 8 MHz)	All peripheral function operating		27	39	mA
					All peripheral function stopped		18		mA
			PLL operating	f_{XX} = 20 MHz (f_X = 10 MHz)	All peripheral function operating		18	26	mA
					All peripheral function stopped		12		mA
			PLL stopped	f_{XX} = 16 MHz (f_X = 16 MHz)	All peripheral function operating		13	20	mA
					All peripheral function stopped		9		mA
	I_{DD3}	IDLE1 mode	PLL stopped[Note 3]	f_{XX} = 16 MHz (f_X = 16 MHz)	TAA, UARTD operating		3.3	4.7	mA
					All peripheral function stopped		1.6		mA
				f_{XX} = 8 MHz (f_X = 8 MHz)	TAA, UARTD operating		2.1	3.0	mA
					All peripheral function stopped		1.3		mA
			f_{XX} = High-speed internal oscillation (f_{RH})[Note 4]		TAA, UARTD operating		1.5	2.3	mA
					All peripheral function stopped		1.1		mA
	I_{DD4}	IDLE2 mode	PLL stopped[Note 3]	f_{XX} = 16 MHz (f_X = 16 MHz)			0.8	1.2	mA
				f_{XX} = 8 MHz (f_X = 8 MHz)			0.5	0.8	mA
			f_{XX} = High-speed internal oscillation (f_{RH})[Note 4]				0.2	0.5	mA
	I_{DD5}	Subclock operation mode[Notes 4, 5]	Crystal resonator (f_{XT} = 32.768 kHz)				80	400	μA
	I_{DD6}	Sub-IDLE mode[Notes 4, 5]	Crystal resonator (f_{XT} = 32.768 kHz)				20	190	μA
	I_{DD7}	STOP mode[Notes 4, 6]	Low-speed internal oscillator (f_{RL}) operating				18.5	110	μA
			Low-speed internal oscillator (f_{RL}) stopped				10.5	95	μA

Notes 1. Total current of V_{DD}, EV_{DD}, and BV_{DD} (all ports stopped). The current of AV_{REF0} and the port buffer current including the current flowing through the on-chip pull-up/pull-down resistors are not included.

2. During SSCG operation, TYP. value + 2.5 mA, MAX. value + 4 mA

3. High-speed internal oscillator (f_{RH}) stopped.

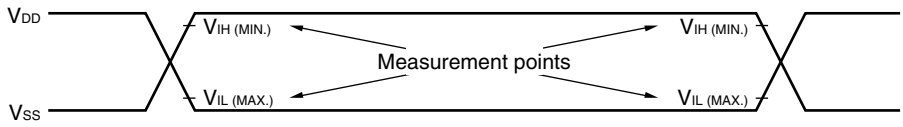
4. When the main clock oscillator (f_{XX}) is stopped.

5. Low-speed internal oscillator (f_{RL}) operating, high-speed internal oscillator (f_{RH}) stopped.

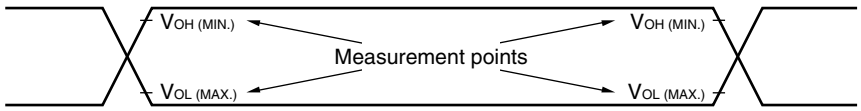
6. When the subclock oscillator (f_{XT}) is not used.

32.7 AC Characteristics

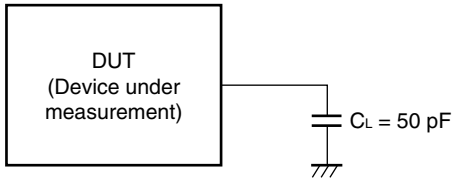
(1) AC test input measurement points (V_{DD}, AV_{REF0}, EV_{DD}, BV_{DD})



(2) AC test output measurement points



(3) Load conditions

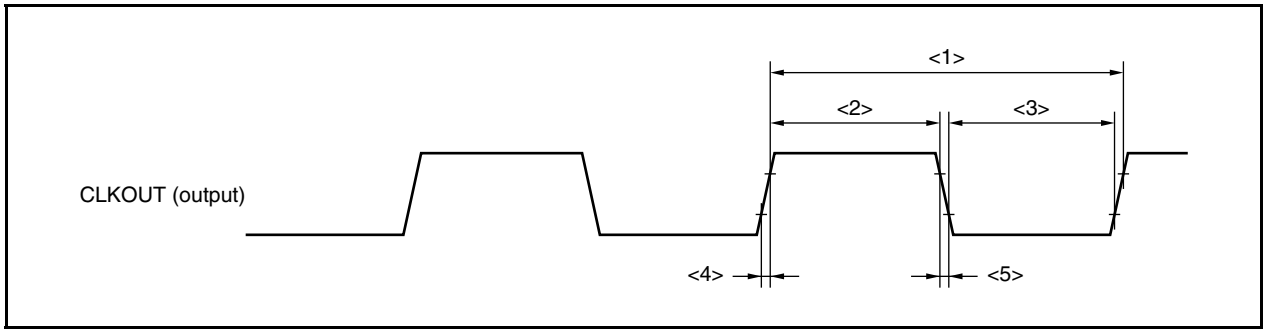


Caution If the load capacitance exceeds 50 pF due to the circuit configuration, bring the load capacitance of the device to 50 pF or less by inserting a buffer or by some other means.

32.7.1 CLKOUT output timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Output cycle	t_{CYK}	<1>	V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V	50 ns	80 μs	
			V_{DD} = EV_{DD} = BV_{DD} = 4.0 V to 5.5 V	31.25 ns	80 μs	
High-level width	t_{WKH}	<2>	V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V	t_{CYK}/2 – 15		ns
			V_{DD} = EV_{DD} = BV_{DD} = 4.0 V to 5.5 V	t_{CYK}/2 – 13		ns
Low-level width	t_{WKL}	<3>	V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V	t_{CYK}/2 – 15		ns
			V_{DD} = EV_{DD} = BV_{DD} = 4.0 V to 5.5 V	t_{CYK}/2 – 13		ns
Rise time	t_{KR}	<4>	V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V		15	ns
			V_{DD} = EV_{DD} = BV_{DD} = 4.0 V to 5.5 V		13	ns
Fall time	t_{KF}	<5>	V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V		15	ns
			V_{DD} = EV_{DD} = BV_{DD} = 4.0 V to 5.5 V		13	ns



32.7.2 Bus timing

(1) CLKOUT asynchronous

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Address setup time (to ASTB↓)	t_{SAST}	<6>		(0.5 + t_{ASW}) – 20		ns
Address hold time (from ASTB↓)	t_{HSTA}	<7>		(0.5 + t_{AHW}) – 15		ns
Delay time from $\overline{RD}$↓ to address float	t_{FRDA}	<8>			16	ns
Data input setup time from address	t_{SAID}	<9>			(2 + n + t_{ASW} + t_{AHW})T – 40	ns
Data input setup time from $\overline{RD}$↓	t_{SRDID}	<10>			(1 + n)T – 30	ns
Delay time from ASTB↓ to $\overline{RD}$, $\overline{WRm}$↓	$t_{DSTRDWR}$	<11>		(0.5 + t_{AHW})T – 15		ns
Data input hold time (from $\overline{RD}$↑)	t_{HRDID}	<12>		0		ns
Address output time from $\overline{RD}$↑	t_{DRDA}	<13>		(1 + i)T – 15		ns
Delay time from $\overline{RD}$, $\overline{WRm}$↑ to ASTB↑	$t_{DRDWRST}$	<14>		0.5T – 15		ns
Delay time from $\overline{RD}$↑ to ASTB↓	t_{DRDST}	<15>		(1.5 + i + t_{ASW})T – 15		ns
$\overline{RD}$, $\overline{WRm}$ low-level width	t_{WRDWRL}	<16>		(1 + n)T – 20		ns
ASTB high-level width	t_{WSTH}	<17>		(1 + i + t_{ASW})T – 15		ns
Data output time from $\overline{WRm}$↓	t_{DWROD}	<18>			15	ns
Data output setup time (to $\overline{WRm}$↑)	t_{SODWR}	<19>		(1 + n)T – 25		ns
Data output hold time (from $\overline{WRm}$↑)	t_{HWROD}	<20>		T – 15		ns
$\overline{WAIT}$ setup time (to address)	t_{SAWT1}	<21>	n ≥ 1		(1.5 + t_{ASW} + t_{AHW})T – 45	ns
	t_{SAWT2}	<22>			(1.5 + n + t_{ASW} + t_{AHW})T – 45	ns
$\overline{WAIT}$ hold time (from address)	t_{HAWT1}	<23>	n ≥ 1	(0.5 + n + t_{ASW} + t_{AHW})T		ns
	t_{HAWT2}	<24>		(1.5 + n + t_{ASW} + t_{AHW})T		ns
$\overline{WAIT}$ setup time (to ASTB↓)	t_{SSTWT1}	<25>	n ≥ 1		(1 + t_{AHW})T – 35	ns
	t_{SSTWT2}	<26>			(1 + n + t_{AHW})T – 35	ns
$\overline{WAIT}$ hold time (from ASTB↓)	t_{HSTWT1}	<27>	n ≥ 1	(n + t_{AHW})T		ns
	t_{HSTWT2}	<28>		(1 + n + t_{AHW})T		ns
$\overline{HLDRQ}$ high-level width	t_{WHQH}	<29>		T + 10		ns
$\overline{HLDAK}$ low-level width	t_{WHAL}	<30>		T – 20		ns
Delay time from $\overline{HLDAK}$↑ to bus output	t_{DHAC}	<31>		–3		ns
Delay time from $\overline{HLDRQ}$↓ to $\overline{HLDAK}$↓	t_{DHQHA1}	<32>			(2n + 7.5)T + 25	ns
Delay time from $\overline{HLDRQ}$↑ to $\overline{HLDAK}$↑	t_{DHQHA2}	<33>		0.5T	1.5T + 35	ns

Remarks 1. T = 1/f_{CPU} (f_{CPU}: CPU operating clock frequency)

2. n: Number of wait clocks inserted in the bus cycle.
The sampling timing changes when a programmable wait is inserted.

3. m = 0, 1

4. i: Number of idle states inserted after a read cycle (0 or 1).

5. The values in the above specifications are values for when clocks with a 1:1 duty ratio are input from X1.

6. t_{ASW}: Number of address setup wait clocks (0 or 1)
t_{AHW}: Number of address hold wait clocks (0 or 1)

(2) CLKOUT synchronous

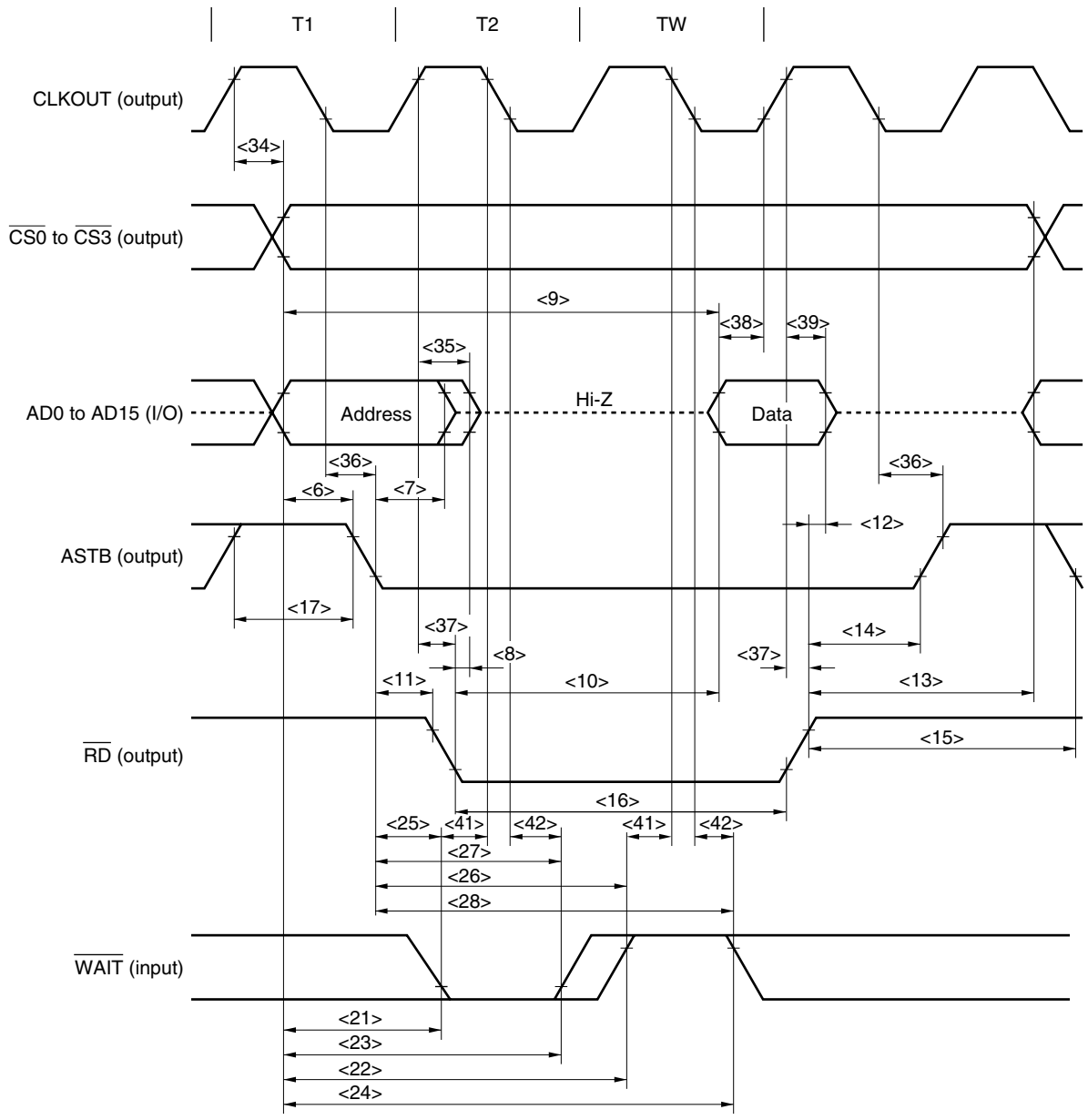
(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
Delay time from CLKOUT↑ to address	t_{DKA}	<34>		0	24	ns
Delay time from CLKOUT↑ to address float	t_{FKA}	<35>		0	24	ns
Delay time from CLKOUT↓ to ASTB	t_{DKST}	<36>		–12	12	ns
Delay time from CLKOUT↑ to $\overline{RD}$, $\overline{WRm}$	t_{DKRDWR}	<37>		–5	14	ns
Data input setup time (to CLKOUT↑)	t_{SIDK}	<38>		20		ns
Data input hold time (from CLKOUT↑)	t_{HKID}	<39>		5		ns
Data output delay time from CLKOUT↑	t_{DKOD}	<40>			22	ns
$\overline{WAIT}$ setup time (to CLKOUT↓)	t_{SWTK}	<41>		30		ns
$\overline{WAIT}$ hold time (from CLKOUT↓)	t_{HKWT}	<42>		5		ns
$\overline{HLDRQ}$ setup time (to CLKOUT↓)	t_{SHQK}	<43>		30		ns
$\overline{HLDRQ}$ hold time (from CLKOUT↓)	t_{HKHQ}	<44>		5		ns
Delay time from CLKOUT↑ to $\overline{HLDAK}$	t_{DKHA}	<45>			24	ns
Delay time from CLKOUT↑ to bus float	t_{DKF}	<46>			25	ns

Remarks 1. m = 0, 1

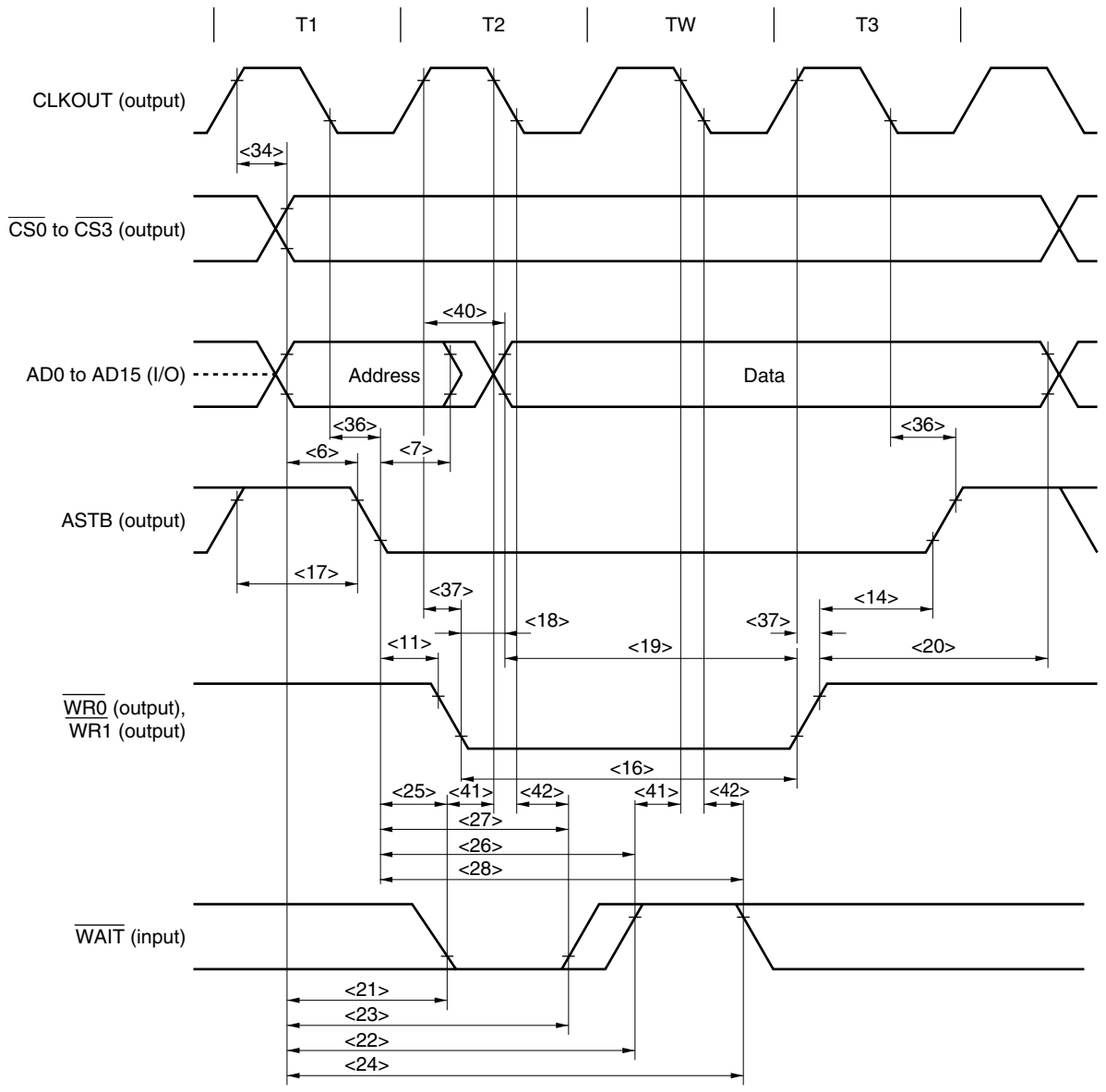
2. The values in the above specifications are values for when clocks with a 1:1 duty ratio are input from X1.

Read cycle (CLKOUT synchronous/asynchronous, 1 wait)



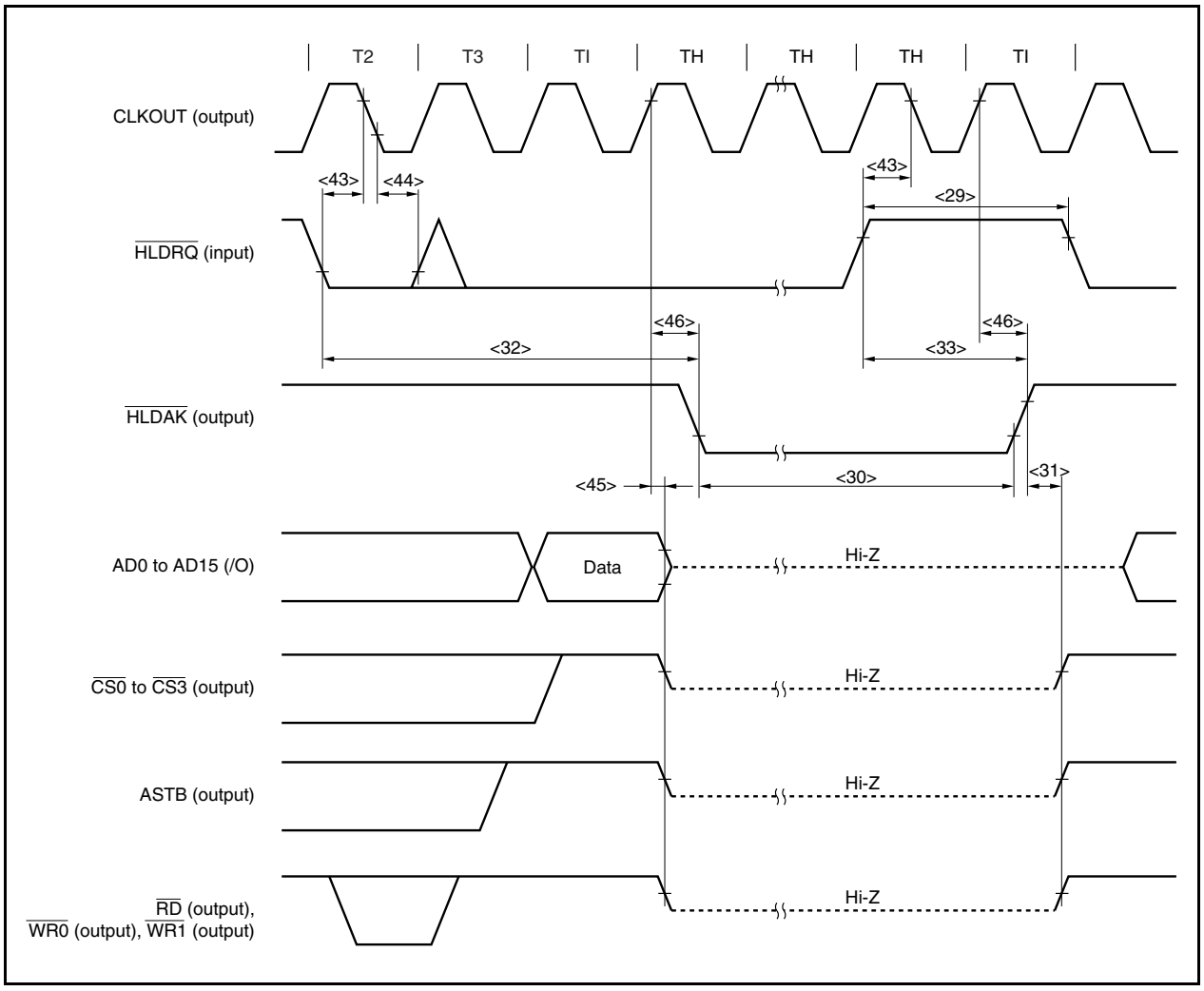
Remark $\overline{WR0}$ and $\overline{WR1}$ are high level.

Write cycle (CLKOUT synchronous/asynchronous, 1 wait)



Remark $\overline{RD}$ is high level.

Bus hold


T2
T3
TI
TH
TH
TH
TI
CLKOUT (output)
<43>
<44>
<43>
<29>
HLDRQ (input)
<46>
<46>
<32>
<33>
HLDAK (output)
<45>
<30>
<31>
AD0 to AD15 (/O)
Data
Hi-Z
CS0 to CS3 (output)
Hi-Z
ASTB (output)
Hi-Z
RD (output),
WR0 (output), WR1 (output)
Hi-Z

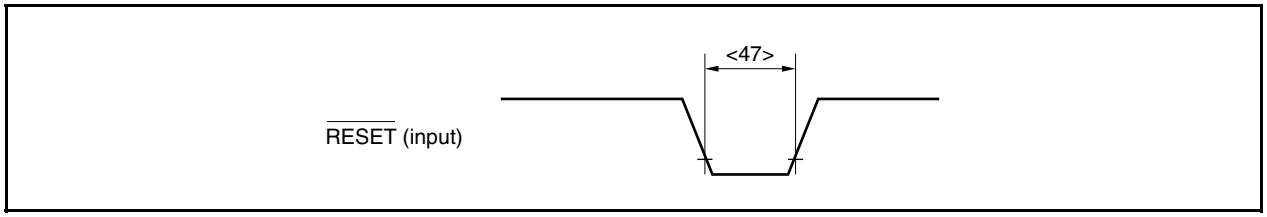
32.8 Basic Operation

(1) Reset timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ low-level width	t_{WRSL}	<47>		250		ns

Reset



(2) Interrupt timing

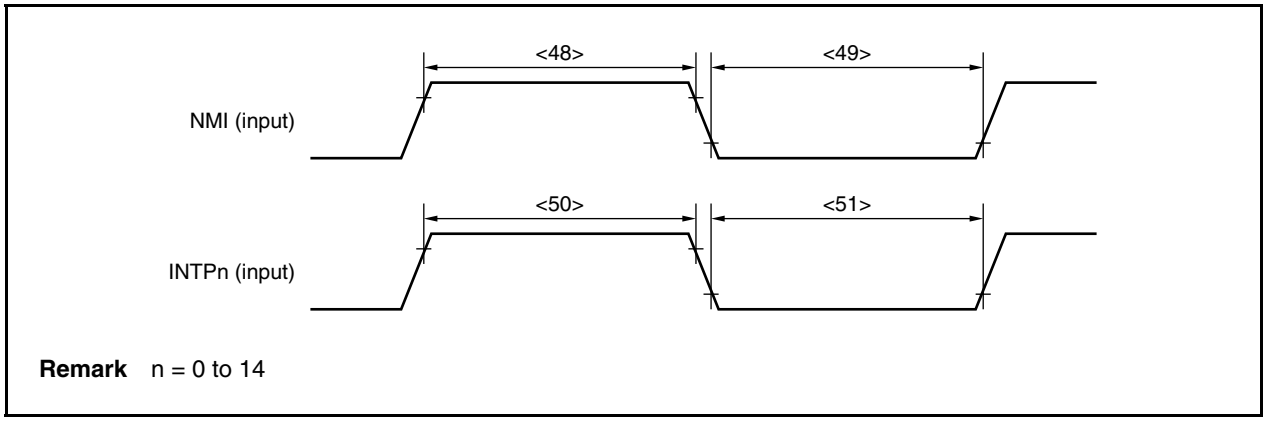
(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
NMI high-level width	t_{WNIH}	<48>	Analog noise elimination	250		ns
NMI low-level width	t_{WNIL}	<49>	Analog noise elimination	250		ns
INTPn[Note 1] high-level width	t_{WITH}	<50>	Analog noise elimination (n = 0 to 14)	250		ns
			Digital noise elimination (n = 3)	**Note 2**		ns
INTPn[Note 1] low-level width	t_{WITL}	<51>	Analog noise elimination (n = 0 to 14)	250		ns
			Digital noise elimination (n = 3)	**Note 2**		ns

Notes 1. The same value as the INTP0/P03 pin applies in the case of the ADTRG pin. The same value as the INTP2/P05 pin applies in the case of the $\overline{\text{DRST}}$ pin.

2. $2T_{samp}$ + 20 or $3T_{samp}$ + 20
T_{samp}: Sampling clock for noise elimination

Reset/interrupt

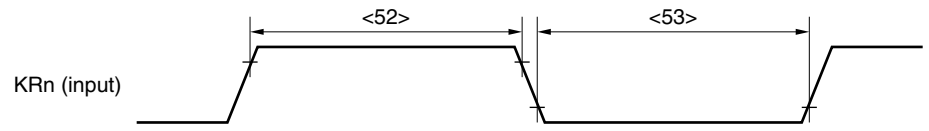


Remark n = 0 to 14

(3) Key interrupt timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
KRn input high-level width	t_{WKRH}	<52>	Analog noise elimination (n = 0 to 7)	250		ns
KRn input low-level width	t_{WKRL}	<53>		250		ns



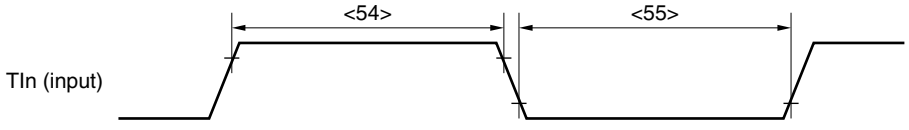
Remark n = 0 to 7

(4) Timer input timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions		MIN.	MAX.	Unit
TIn high-level width	t_{TIH}	<54>	TIAA00, TIAA01, TIAA10, TIAA11, TIAA20, TIAA21, TIAA30, TIAA31, TIAA40, TIAA41, TIAB00 to TIAB03, TIAB10 to TIAB13, TIAB20 to TIAB23[Note]		250		ns
TIn low-level width	t_{TIL}	<55>			250		ns
TOn output cycle	f_{TCYK}		TOAA00, TOAA01, TOAA10, TOAA11, TOAA20, TOAA21, TOAA30, TOAA31, TOAA40, TOAA41, TOAB00 to TOAB03, TOAB10 to TOAB13, TOAB20 to TOAB23	4.0 V ≤ V_{DD} ≤ 5.5 V		16	MHz
				3.7 V ≤ V_{DD} < 4.0 V		10	MHz

Note Noise on the TIAA00, TIAA10, TIAA20, TIAA30, TIAB00, TIAB10, and TIAB20 pins can be eliminated only when a capture signal is input.
The noise cannot be eliminated when an external trigger signal or an external event counter signal is input.



Remark TIn: TIAA00, TIAA01, TIAA10, TIAA11, TIAA20, TIAA21, TIAA30, TIAA31, TIAA40, TIAA41, TIAB00 to TIAB03, TIAB10 to TIAB13, TIAB20 to TIAB23

(5) CSIB timing

(a) Master mode

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKBn}$ cycle time	t_{KCY1}	<56>		125		ns
$\overline{SCKBn}$ high-level width	t_{KH1}	<57>		$t_{KCY1}/2 - 15$		ns
$\overline{SCKBn}$ low-level width	t_{KL1}	<58>		$t_{KCY1}/2 - 15$		ns
SIBn setup time (to $\overline{SCKBn}\uparrow$)	t_{SIK1}	<59>		30		ns
SIBn hold time (from $\overline{SCKBn}\uparrow$)	t_{KSI1}	<60>		25		ns
Output delay time from $\overline{SCKBn}\downarrow$ to SOBn	t_{KSO1}	<61>			25	ns

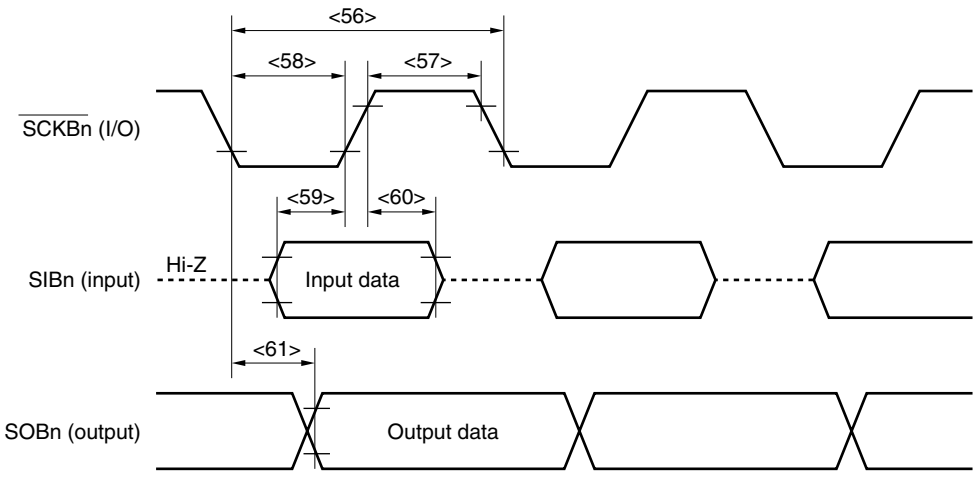
Remark n = 0 to 3

(b) Slave mode

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{SCKBn}$ cycle time	t_{KCY1}	<56>		200		ns
$\overline{SCKBn}$ high-level width	t_{KH1}	<57>		90		ns
$\overline{SCKBn}$ low-level width	t_{KL1}	<58>		90		ns
SIBn setup time (to $\overline{SCKBn}\uparrow$)	t_{SIK1}	<59>		50		ns
SIBn hold time (from $\overline{SCKBn}\uparrow$)	t_{KSI1}	<60>		50		ns
Output delay time from $\overline{SCKBn}\downarrow$ to SOBn	t_{KSO1}	<61>			50	ns

Remark n = 0 to 3



Remark n = 0 to 3

(6) UARTD timing

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
Communication rate				1.5	Mbps
ASCK0 cycle time				10	MHz

(7) I^2C bus timing

(T_A = –40 to +85°C, V_{DD} = V_{DD1} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = V_{SS1} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V)

Parameter		Symbol		Normal Mode		High-Speed Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCL00 clock frequency		f_{CLK}		0	100	0	400	kHz
Bus free time (Between start and stop conditions)		t_{BUF}	<62>	4.7	–	1.3	–	μs
Hold time[Note 1]		$t_{HD:STA}$	<63>	4.0	–	0.6	–	μs
SCL00 clock low-level width		t_{LOW}	<64>	4.7	–	1.3	–	μs
SCL00 clock high-level width		t_{HIGH}	<65>	4.0	–	0.6	–	μs
Setup time for start/restart conditions		$t_{SU:STA}$	<66>	4.7	–	0.6	–	μs
Data hold time	CBUS compatible master	$t_{HD:DAT}$	<67>	5.0	–	–	–	μs
	I^2C mode			0[Note 2]	–	0[Note 2]	0.9[Note 3]	μs
Data setup time		$t_{SU:DAT}$	<68>	250	–	100[Note 4]	–	ns
SDA00 and SCL00 signal rise time		t_R	<69>	–	1000	20 + 0.1Cb[Note 5]	300	ns
SDA00 and SCL00 signal fall time		t_F	<70>	–	300	20 + 0.1Cb[Note 5]	300	ns
Stop condition setup time		$t_{SU:STO}$	<71>	4.0	–	0.6	–	μs
Pulse width of spike suppressed by input filter		t_{SP}	<72>	–	–	0	50	ns
Capacitance load of each bus line		Cb		–	400	–	400	pF

Notes 1. At the start condition, the first clock pulse is generated after the hold time.

2. The system requires a minimum of 300 ns hold time internally for the SDA00 signal (at $V_{IHmin.}$ of SCL00 signal) in order to occupy the undefined area at the falling edge of SCL00.

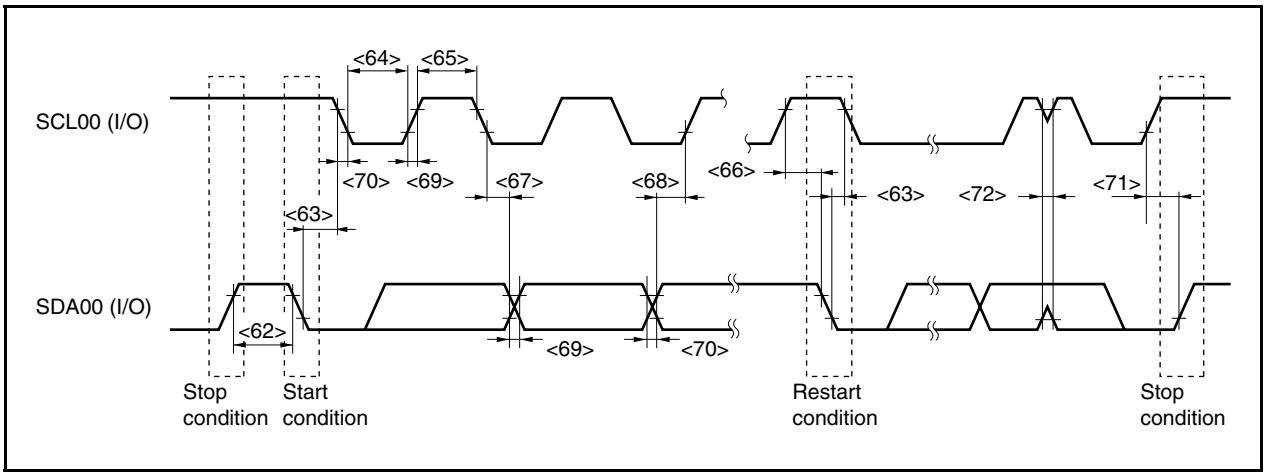
3. If the system does not extend the SCL00 signal low hold time (t_{LOW}), only the maximum data hold time ($t_{HD:DAT}$) needs to be satisfied.

4. The high-speed mode I^2C bus can be used in the normal-mode I^2C bus system. In this case, set the high-speed mode I^2C bus so that it meets the following conditions.

- If the system does not extend the SCL00 signal's low state hold time:
  $t_{SU:DAT} \geq 250$ ns
- If the system extends the SCL00 signal's low state hold time:
  Transmit the following data bit to the SDA00 line prior to the SCL00 line release ($t_{Rmax.} + t_{SU:DAT}$ = 1,000 + 250 = 1,250 ns: Normal mode I^2C bus specification).

5. Cb: Total capacitance of one bus line (unit: pF)

I²C bus mode



(8) A/D converter

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.7 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution					10	bit
Overall error[Note 1]		4.0 ≤ AV_{REF0} ≤ 5.5 V		±0.15	±0.3	%FSR
Conversion time	t_{CONV}		3.1		16	μs
Stabilization time	t_{STA}	After ADA0M0.ADA0PS bit changes from 0 to 1	2			μs
Power down recovery time	t_{DPU}	Starting operation after STOP mode is released	1			μs
Zero scale error[Note 1]	ZSE				±0.3	%FSR
Full scale error[Note 1]	FSE				±0.3	%FSR
Non-linearity error[Note 2]	INL				±2.5	LSB
Differential linearity error[Note 2]	DNL				±1.5	LSB
Analog input voltage	V_{IAN}		AV_{SS}		AV_{REF0}	V
AV_{REF0} current	I_{AREF0}	When using A/D converter		4	7	mA
		When not using A/D converter		1	10	μA

Notes 1. Excluding quantization error (±0.05 %FSR). Indicates the ratio to the full-scale value (%FSR).

2. Quantization error (±0.5LSB)

Remark FSR: Full Scale Range

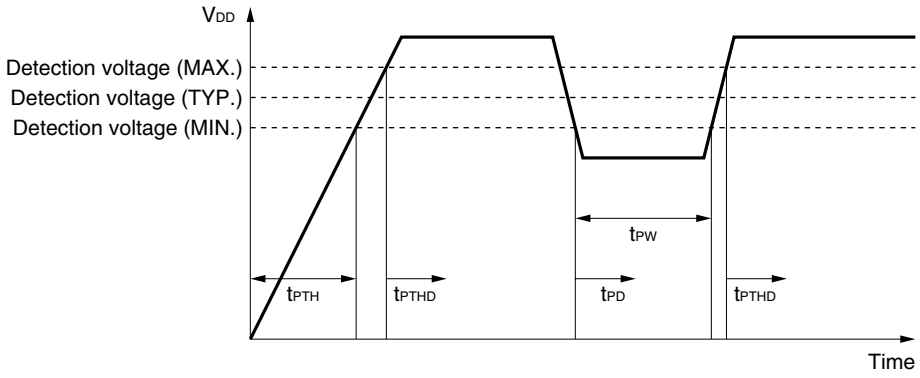
(9) POC circuit characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD}, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POC0}		3.3	3.5	3.7	V
Power supply startup time	t_{PTH}	V_{DD} = 0 V → 3.3 V	0.002			ms
Response delay time 1 Note 1	t_{PTHD}	After V_{DD} reaches 3.7 V on power application			2.0	ms
Response delay time 2 Note 2	t_{PD}	After V_{DD} drops below 3.3 V on power drop			1.0	ms
Minimum V_{DD} width	t_{PW}		0.2			ms

Notes 1. The time required to release a reset after the detection voltage is detected.

2. The time required to output a reset after the detection voltage is detected.

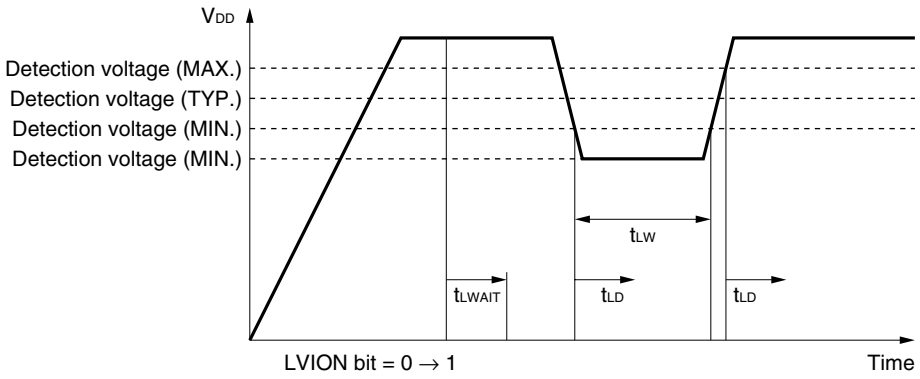


(10) LVI circuit characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 3.3 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{LVI0}		3.8	4.0	4.2	V
	V_{LVI1}		3.5	3.7	3.9	V
Response time[Note]	t_{LD}	After V_{DD} reaches V_{LVI0}/V_{LVI1} (MAX.) or drops below V_{LVI0}/V_{LVI1} (MIN.)		0.2	2.0	ms
Minimum V_{DD} width	t_{LW}		0.2			ms
Reference voltage stabilization wait time	t_{LWAIT}	After LVION bit (LVIM.bit7) changes from 0 to 1		0.1	0.2	ms

Note The time required to output an interrupt/reset after the detection voltage is detected.

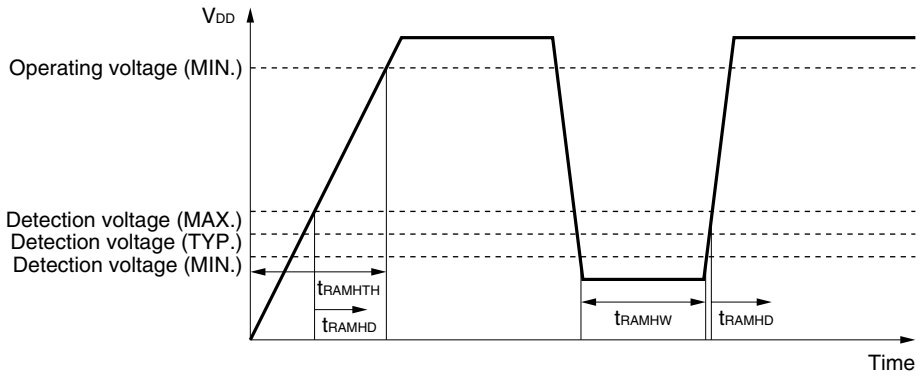


(11) RAM retention flag characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD} = 1.9 V to 5.5 V, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{RAMH}		1.9	2.0	2.1	V
Supply voltage rise time	t_{RAMHTH}	V_{DD} = 0 V → 3.3 V	0.002		1800	ms
Response time[Note]	t_{RAMHD}	After the supply voltage reaches the detection voltage (MAX.)		0.2	2.0	ms
Minimum V_{DD} width	t_{RAMHW}		0.2			ms

Note Time required to set the RAMF bit after the detection voltage is detected.



32.9 Flash Memory Programming Characteristics

(1) Basic characteristics

(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD}, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Operating frequency	f_{CPU}		4		32	MHz
Supply voltage	V_{DD}		3.8		5.5	V
Number of writes	C_{WRT} Note				100	Times
Input voltage, high	V_{IH}	FLMD0	0.8EV_{DD}		EV_{DD}	V
Input voltage, low	V_{IL}	FLMD0	EV_{SS}		0.2EV_{DD}	V
Programming temperature	t_{PRG}		–40		+85	°C

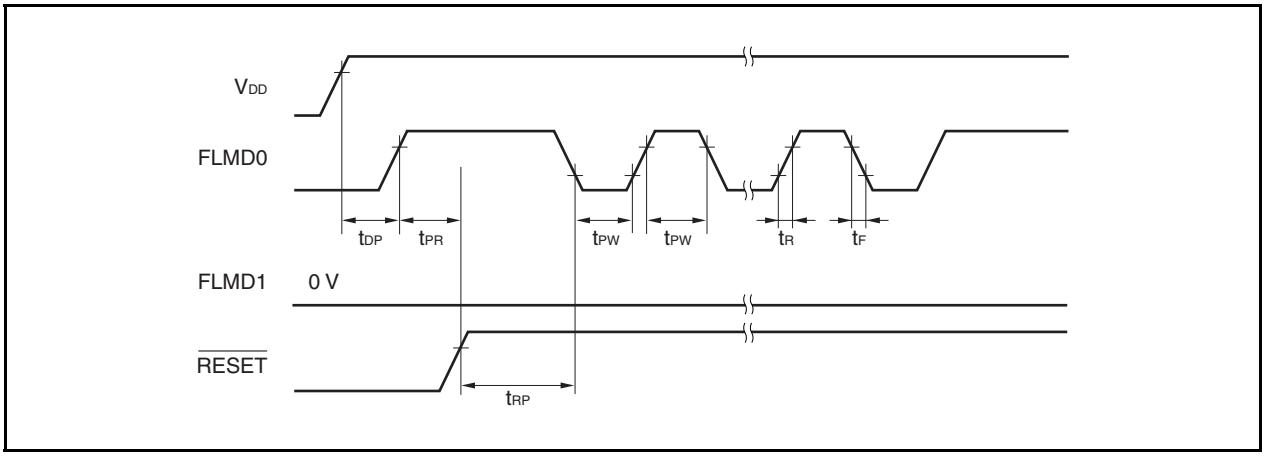
Note When writing initially to shipped products, it is counted as one rewrite for both “erase to write” and “write only”.

Example (P: Write, E: Erase)
Shipped product → P → E → P → E → P: 3 rewrites
Shipped product → E → P → E → P → E → P: 3 rewrites

(2) Serial write operation characteristics

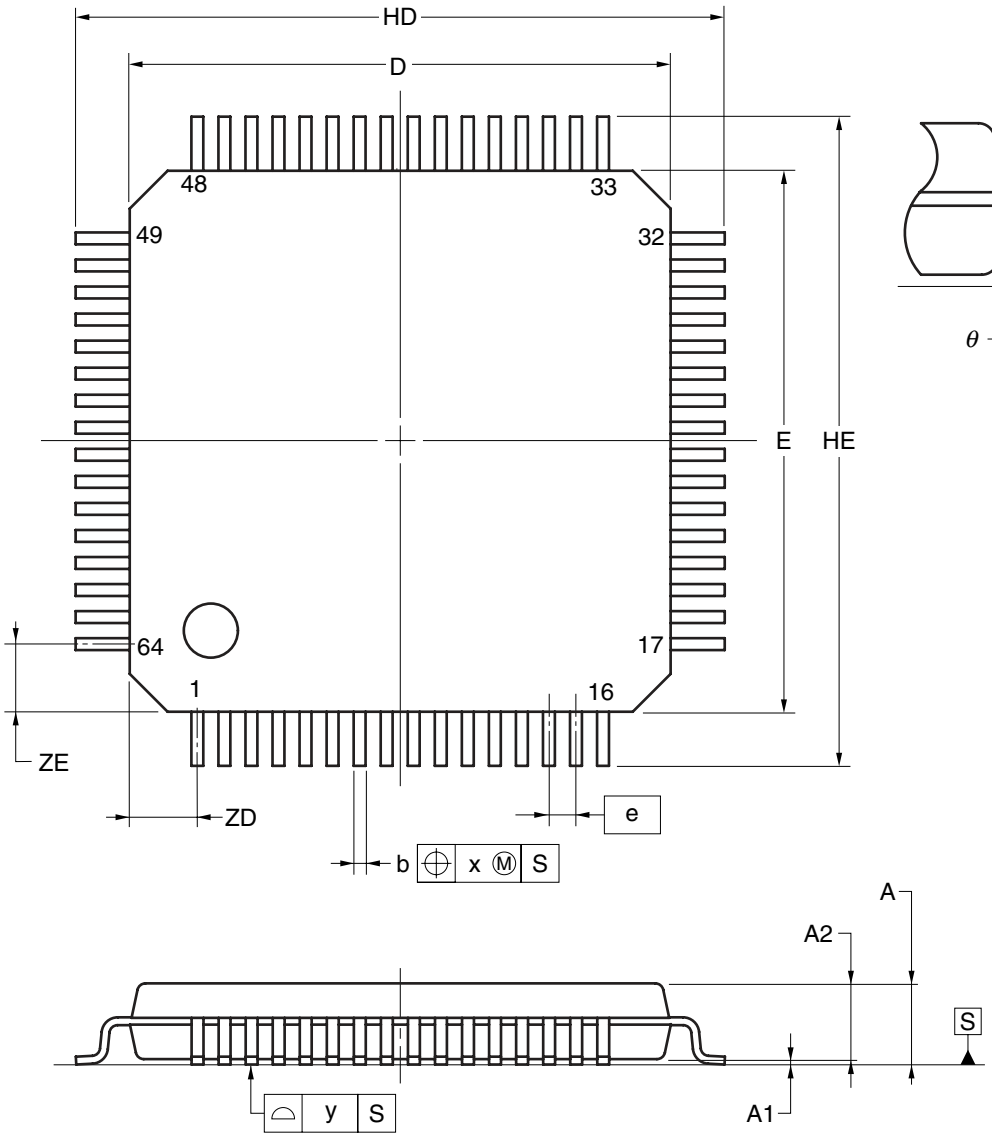
(T_A = –40 to +85°C, V_{DD} = EV_{DD} = BV_{DD}, 4.0 V ≤ AV_{REF0} ≤ 5.5 V, V_{SS} = EV_{SS} = BV_{SS} = AV_{SS} = 0 V, C_L = 50 pF)

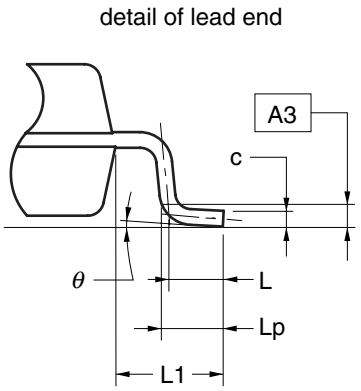
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Setup time from VDD↑ to FLMD0↑	t_{DP}		1			ms
Time from FLMD0↑ to RESET release	t_{PR}		2			ms
FLMD0 pulse input start time from RESET↑	t_{RP}		800			μs
FLMD0 pulse high-level width/ low-level width	t_{PW}		10		100	μs
<R> FLMD0 rise time	t_R				1	μs
<R> FLMD0 fall time	t_F				1	μs



• V850ES/HE3

64-PIN PLASTIC LQFP(FINE PITCH)(10x10)





(UNIT:mm)

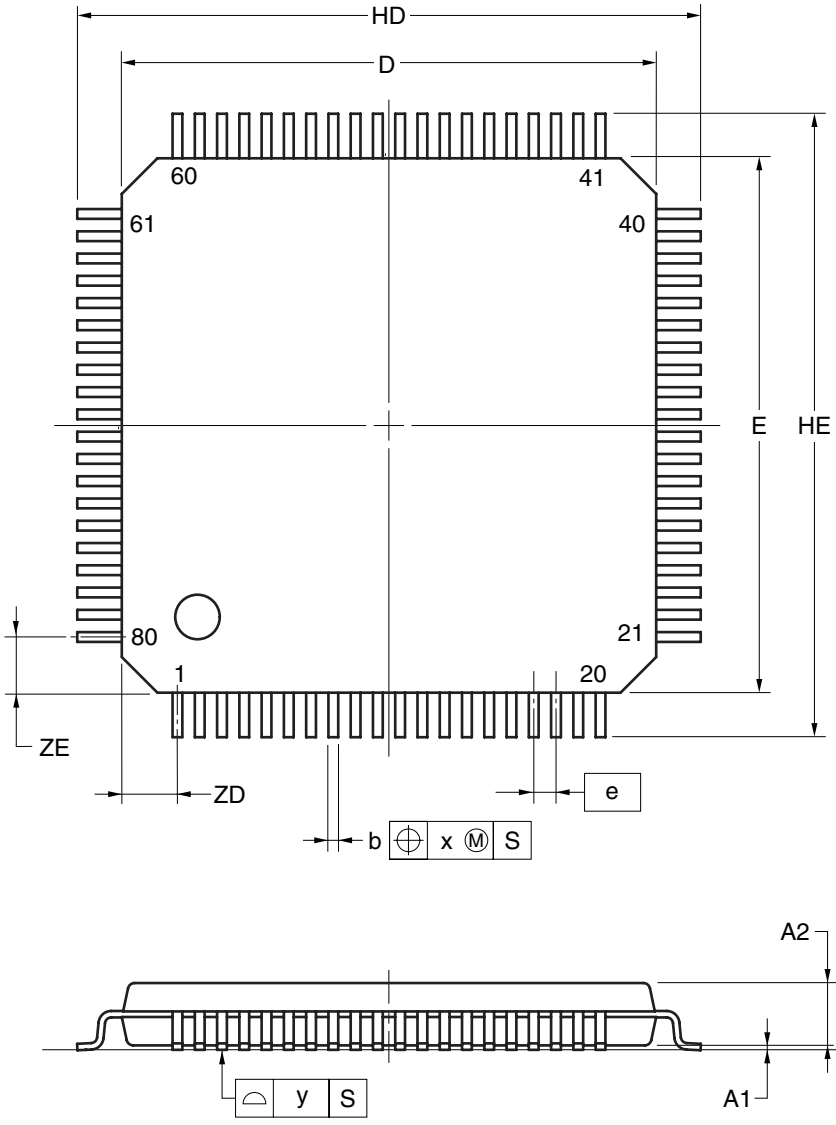
ITEM	DIMENSIONS
D	10.00±0.20
E	10.00±0.20
HD	12.00±0.20
HE	12.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	1.25
ZE	1.25

P64GB-50-GAH

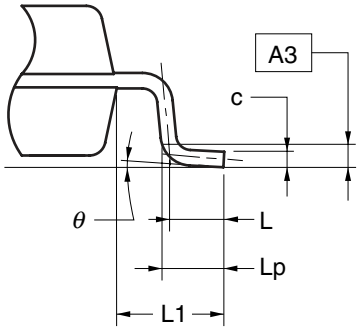
NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

• V850ES/HF3

80-PIN PLASTIC LQFP (FINE PITCH) (12x12)



detail of lead end



(UNIT:mm)

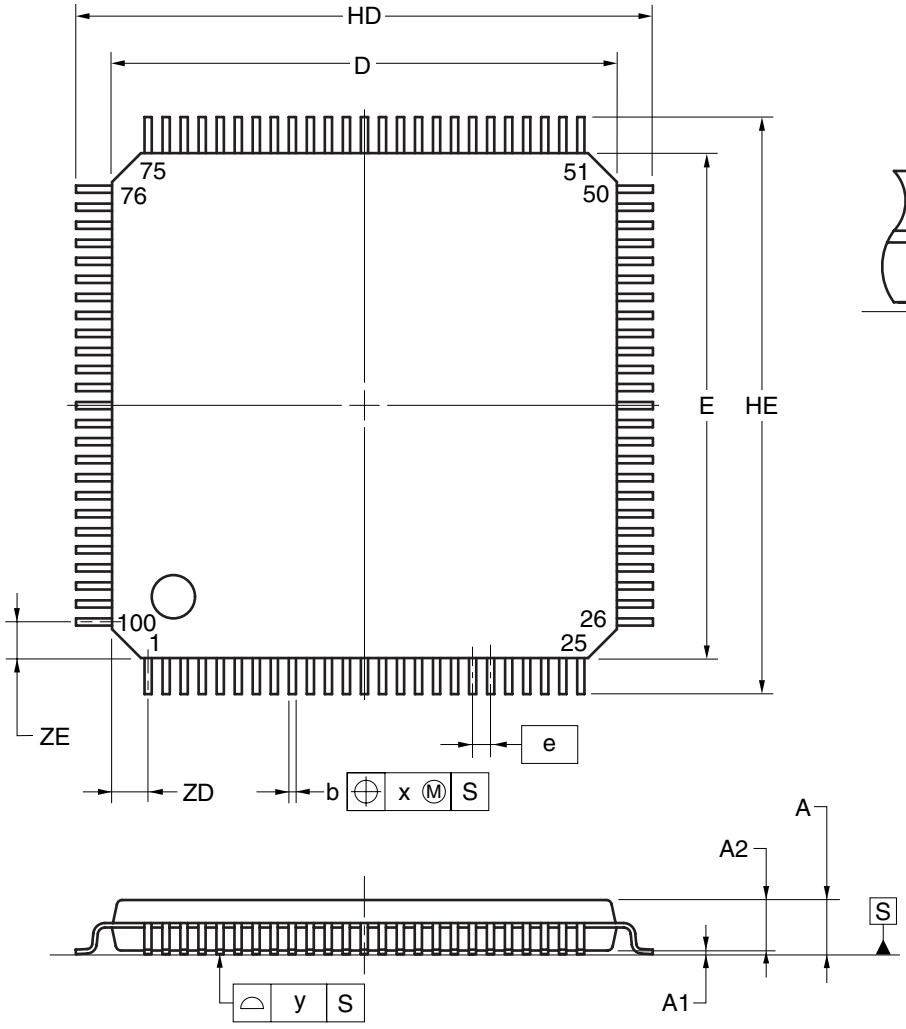
ITEM	DIMENSIONS
D	12.00±0.20
E	12.00±0.20
HD	14.00±0.20
HE	14.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	1.25
ZE	1.25

P80GK-50-GAK

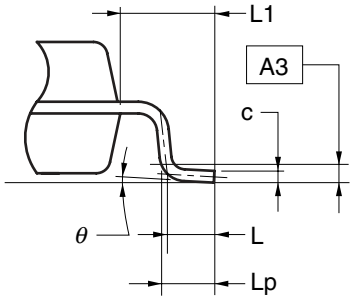
NOTE
Each lead centerline is located within 0.08 mm of its true position at maximum material condition.

• V850ES/HG3

100-PIN PLASTIC LQFP (FINE PITCH) (14x14)



detail of lead end



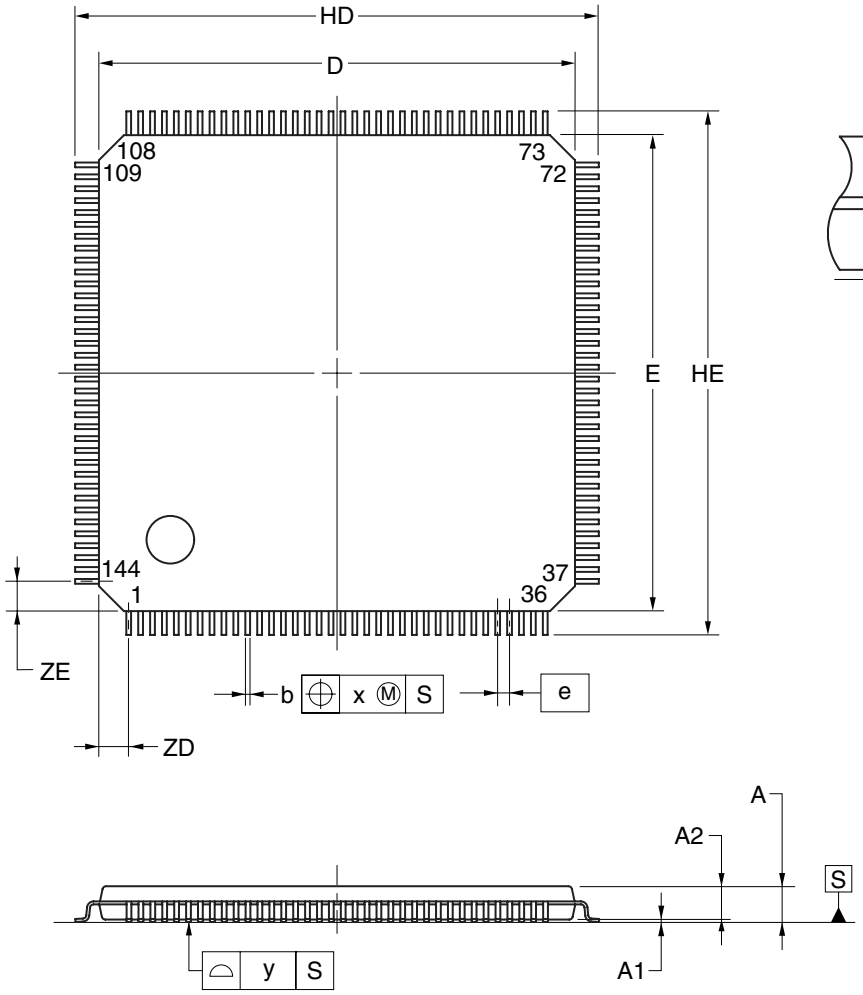
(UNIT:mm)

ITEM	DIMENSIONS
D	14.00±0.20
E	14.00±0.20
HD	16.00±0.20
HE	16.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.20^{+0.07}_{-0.03}$
c	$0.125^{+0.075}_{-0.025}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	1.00
ZE	1.00

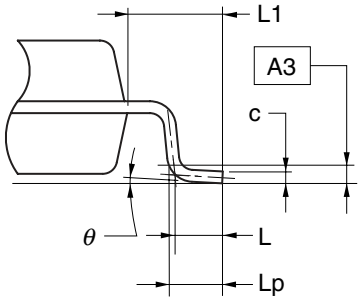
P100GC-50-UEU-1

• V850ES/HJ3

144-PIN PLASTIC LQFP (FINE PITCH) (20x20)



detail of lead end



(UNIT:mm)

ITEM	DIMENSIONS
D	20.00±0.20
E	20.00±0.20
HD	22.00±0.20
HE	22.00±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	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.50
x	0.08
y	0.08
ZD	1.25
ZE	1.25

P144GJ-50-GAE-1

<R>

CHAPTER 34 RECOMMENDED SOLDERING CONDITIONS

The V850ES/Hx3 should be soldered and mounted under the following recommended conditions.
For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.necel.com/pkg/en/mount/index.html)

Table 34-1. Surface Mounting Type Soldering Conditions

PD70F3747GB-GAH-AX: 64-pin plastic LQFP (fine pitch) (10 10)
PD70F3750GK-GAK-AX: 80-pin plastic LQFP (fine pitch) (12 12)
PD70F3752GC-UEU-AX: 100-pin plastic LQFP (fine pitch) (14 14)
PD70F3755GJ-GAE-AX: 144-pin plastic LQFP (fine pitch) (20 20)
PD70F3757GJ-GAE-AX: 144-pin plastic LQFP (fine pitch) (20 20)

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: Three times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 20 to 72 hours)	IR60-207-3
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Remarks 1. V850ES/Hx3 microcontrollers are lead-free products.

2. For soldering methods and conditions other than those recommended above, please contact an NEC Electronics sales representative.

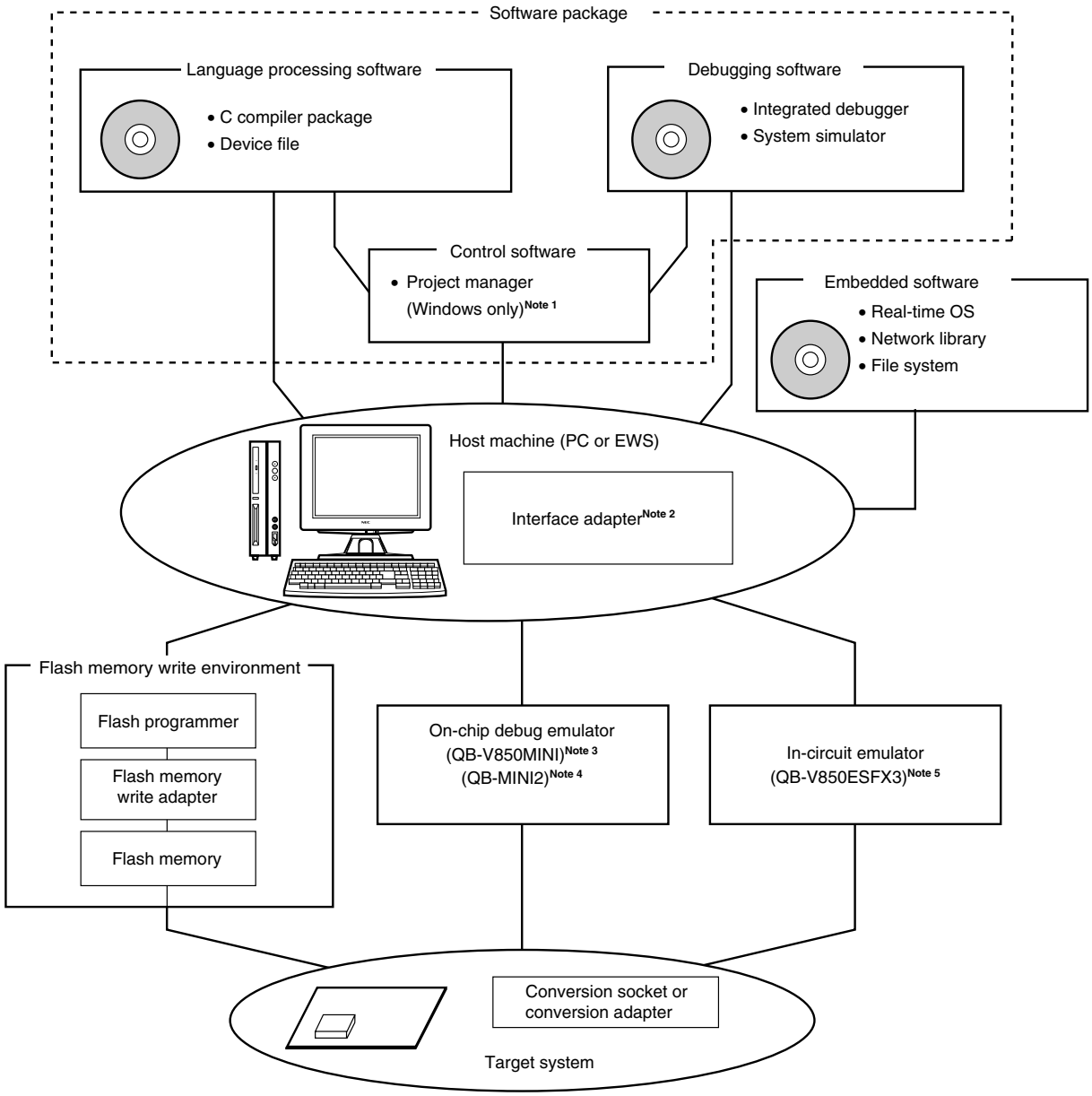
APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the V850ES/Hx3.
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 1. Project manager PM+ is included in the C compiler package.
PM+ is only used in Windows.

2. The QB-V850MINI, QB-MINI2, and QB-V850ESFX3 support the USB interface only.

3. The QB-V850MINI is supplied with the ID850QB, USB interface cable, OCD cable, self-check board, KEL adapter, and KEL connector. All other products are optional.

4. 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.

5. The QB-V850ESFX3 is supplied with the ID850QB, flash memory programmer (QB-MINI2), 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: μS××××SP850

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××SP850

××××	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: μS××××CA703000
DF703757 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 ×××× in the part number differs depending on the host machine and OS used.

μS××××CA703000

××××	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

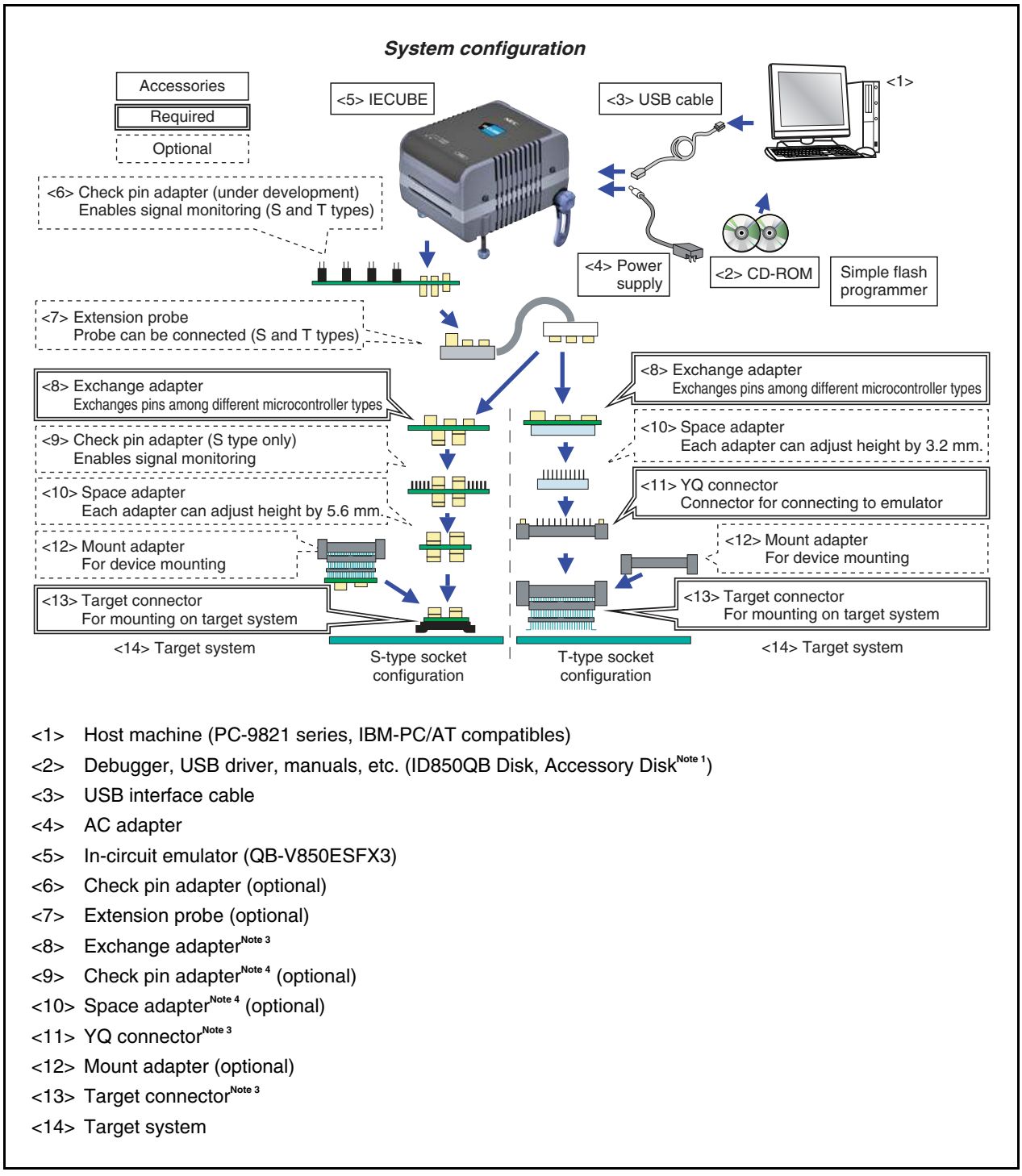
PM+ Project manager	This is control software designed to enable efficient user program development in the Windows environment. All operations used in development of a user program, such as starting the editor, building, and starting the debugger, can be performed from PM+. **<Caution>** PM+ is included in C compiler package CA850. It can only be used in Windows.

A.4 Debugging Tools (Hardware)

A.4.1 When using IECUBE QB-V850ESFX3

The system configuration when connecting the QB-V850ESFX3 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-V850ESFX3) (1/2)



<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-V850ESFX3)
<6> Check pin adapter (optional)
<7> Extension probe (optional)
<8> Exchange adapter[Note 3]
<9> Check pin adapter[Note 4] (optional)
<10> Space adapter[Note 4] (optional)
<11> YQ connector[Note 3]
<12> Mount adapter (optional)
<13> Target connector[Note 3]
<14> Target system

Figure A-2. System Configuration (When Using QB-V850ESFX3) (2/2)

Notes 1. Download the device file from the NEC Electronics website.
http://www.necel.com/micro/ods/eng/

2. Under development

3. Supplied with the device depending on the ordering number.

- When QB-V850ESFX3-ZZZ is ordered
  The exchange adapter and the target connector are not supplied.
- When QB-V850ESFX3-S64GB, QB-V850ESFX3-S80GK, QB-V850ESFX3-S100GC, or QB-V850ESFX3-S144GJ is ordered
  The exchange adapter (S type) and the target connector (S type) are supplied.
- When QB-V850ESFX3-T64GB, QB-V850ESFX3-T80GK, QB-V850ESFX3-T100GC, or QB-V850ESFX3-T144GJ is ordered
  The exchange adapter (T type) and the target connector (T type) are supplied.

4. When using both <9> and <10>, the order between <9> and <10> is not cared.

<5> QB-V850ESFX3[Note] In-circuit emulator	In-circuit emulator to debug hardware and software when developing application systems using the V850ES/Hx3. 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-V850ESFX3.
<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	Connector to connect target connector and exchange adapter
<12> Mount adapter	Adapter to mount the V850ES/Hx3 with socket.
<13> Target connector	Connector to solder on the target system.

Note The QB-V850ESFX3 is supplied with a power supply unit, USB interface cable, and flash memory programmer (QB-MINI2). It is also supplied with integrated debugger ID850QB as control software.

Remark The numbers in the angle brackets correspond to the numbers in Figure A-2. See **Table A-1** for the part number of each component.

Table A-1. Part Numbers of Components Related to IECUBE

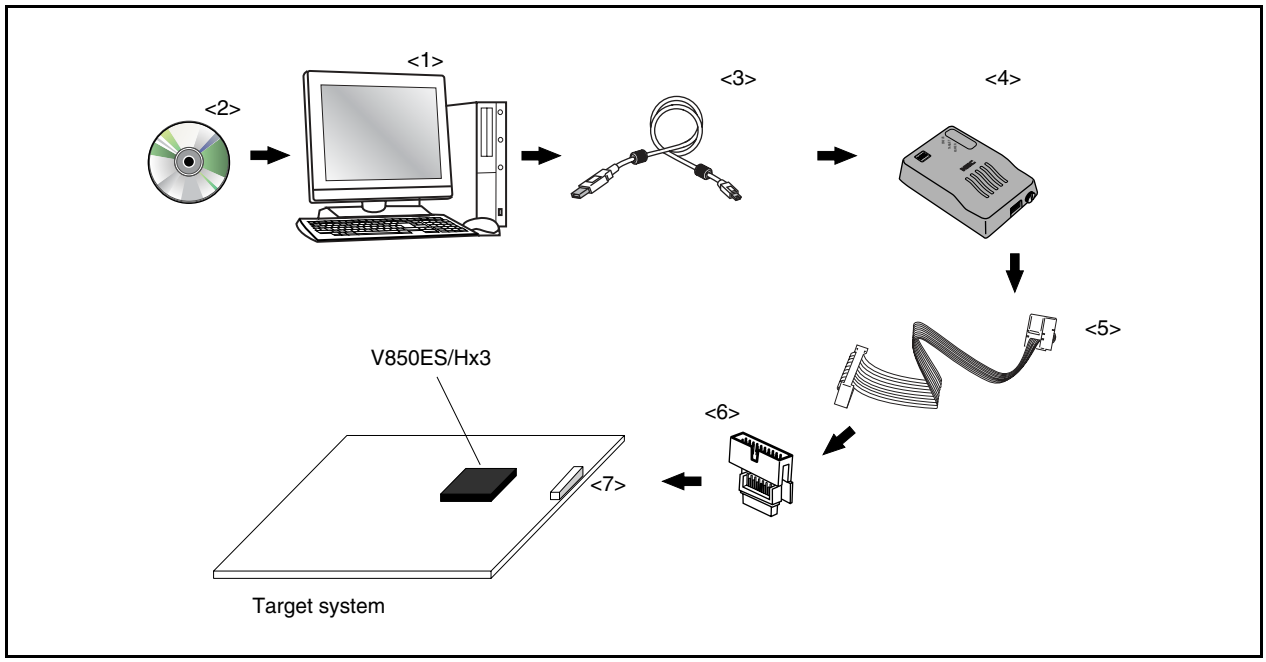
Target Device		V850ES/HE3	V850ES/HF3	V850ES/HG3	V850ES/HJ3
<6> Check pin adapter 2	S type	QB-144-CA-01			
	T type				
<7> Emulation probe	S type	QB-144-EP-01S			
	T type				
<8> Exchange adapter	S type	QB-64GB-EA-01S	QB-80GK-EA-02S	QB-100GC-EA-01S	QB-144GJ-EA-03S
	T type	QB-64GB-EA-02T	QB-80GK-EA-02T	QB-100GC-EA-01T	QB-144GJ-EA-03T
<9> Check pin adapter 1	S type	QB-64-CA-01S	QB-80-CA-01S	QB-100-CA-01S	QB-144-CA-01S
<10> Space adapter	S type	QB-64-SA-01S	QB-80-SA-01S	QB-100-SA-01S	QB-144-SA-01S
	T type	QB-64GB-YS-01T	QB-80GK-YS-01T	QB-100GC-YS-01T	QB-144GJ-YS-01T
<11> YQ connector	T type	QB-64GB-YQ-01T	QB-80GK-YQ-01T	QB-100GC-YQ-01T	QB-144GJ-YQ-01T
<12> Mount adapter	S type	QB-64GB-MA-01S	QB-80GK-MA-01S	QB-100GC-MA-01S	QB-144GJ-MA-01S
	T type	QB-64GB-HQ-01T	QB-80GK-HQ-01T	QB-100GC-HQ-01T	QB-144GJ-HQ-01T
<13> Target connector	S type	QB-64GB-TC-01S	QB-80GK-TC-01S	QB-100GC-TC-01S	QB-144GJ-TC-01S
	T type	QB-64GB-NQ-01T	QB-80GK-NQ-01T	QB-100GC-NQ-01T	QB-144GJ-NQ-01T

A.4.2 When using MINICUBE QB-V850MINI

(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> CD-ROM[Note 1]	Contents such as integrated debugger ID850QB, N-Wire Checker, device driver, and documents are included in CD-ROM. It is supplied with MINICUBE.
<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/Hx3. 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.
<6> Connector conversion board KEL adapter	This conversion board is supplied with MINICUBE.
<7> MINICUBE connector KEL connector[Note 2]	8830E-026-170S (supplied with MINICUBE) 8830E-026-170L (sold separately)

Notes 1. Download the device file from the NEC Electronics website.
http://www.necel.com/micro/ods/eng/index.html

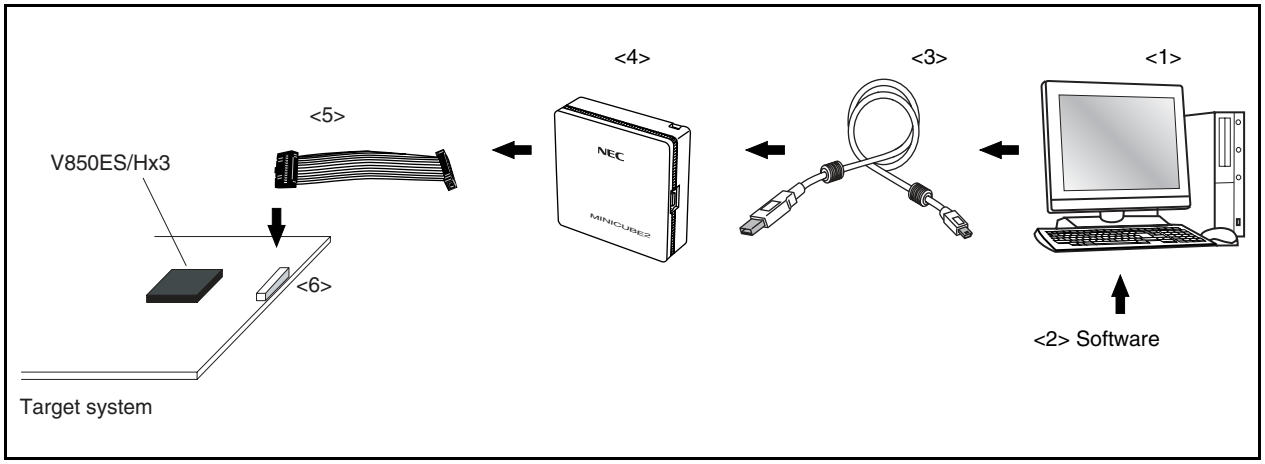
2. Product of KEL Corporation

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 NEC Electronics website. http://www.necel.com/micro/ods/eng/
<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/Hx3. 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.
	Part number: μS×××× ID703000-QB (ID850QB)

Remark ×××× in the part number differs depending on the host machine and OS used.

μS××××ID703000-QB

××××	Host Machine	OS	Supply Medium
AB17	PC-9800 series, IBM PC/AT compatibles	Windows (Japanese version)	CD-ROM
BB17		Windows (English version)	

A.6 Embedded Software

RX850, RX850 Pro Real-time OS	The RX850 and RX850 Pro are real-time OSs conforming to μITRON 3.0 specifications. A tool (configurator) for generating multiple information tables is supplied. RX850 Pro has more functions than the RX850.
	Part number: μS××××RX703000-ΔΔΔΔ (RX850) μS××××RX703100-ΔΔΔΔ (RX850 Pro)
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 ×××× and ΔΔΔΔ in the part number differ depending on the host machine and OS used.

μS××××RX703000-ΔΔΔΔ
μS××××RX703100-ΔΔΔΔ

ΔΔΔΔ	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

××××	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 IV (part number: PG-FP4) Flashpro V (part number: PG-FP5) Flash programmer	Flash programmer dedicated to microcontrollers with on-chip flash memory.
QB-MINI2 (MINICUBE2)	On-chip debug emulator with programming function.
FA-64GB-GAH-B FA-80GK-GAK-B FA-100GC-UEU-B FA-144GJ-GAE-B Flash memory writing adapter	Flash memory writing adapter used connected to the Flashpro IV, Flashpro V, etc. (not wired). • FA-64GB-GAH-B: For 64-pin plastic LQFP (V850ES/HE3) • FA-80GK-GAK-B: For 80-pin plastic LQFP (V850ES/HF3) • FA-100GC-UEU-B: For 100-pin plastic LQFP (V850ES/HG3) • FA-144GJ-GAE-B: For 144-pin plastic LQFP (V850ES/HJ3)
FA-70F3371GB-GAH-RX (under development) FA-70F3373GK-GAK-RX (under development) FA-70F3375GC-UEU-RX (under development) FA-70F3378GJ-GAE-RX (under development) Flash memory writing adapter	Flash memory writing adapter used connected to the Flashpro IV, Flashpro V, etc. (already wired). • FA-70F3371GB-GAH-RX: For 64-pin plastic LQFP (V850ES/HE3) • FA-70F3373GK-GAK-RX: For 80-pin plastic LQFP (V850ES/HF3) • FA-70F3375GC-UEU-RX: For 100-pin plastic LQFP (V850ES/HG3) • FA-70F3378GJ-GAE-RX: For 144-pin plastic LQFP (V850ES/HJ3)

Remark FA-64GB-GAH-B, FA-80GK-GAK-B, FA-100GC-UEU-B, FA-144GJ-GAE-B, FA-70F3371GB-GAH-RX, FA-70F3373GK-GAK-RX, FA-70F3375GC-UEU-RX, and FA-70F3378GJ-GAE-RX are products of Naito Densei Machida Mfg. Co., Ltd.
TEL: +81-42-750-4172

APPENDIX B REGISTER INDEX

(1/12)

Symbol	Name	Unit	Page
ADA0CR0	A/D conversion result register 0	ADC	591
ADA0CR0H	A/D conversion result register 0H	ADC	591
ADA0CR1	A/D conversion result register 1	ADC	591
ADA0CR1H	A/D conversion result register 1H	ADC	591
ADA0CR2	A/D conversion result register 2	ADC	591
ADA0CR2H	A/D conversion result register 2H	ADC	591
ADA0CR3	A/D conversion result register 3	ADC	591
ADA0CR3H	A/D conversion result register 3H	ADC	591
ADA0CR4	A/D conversion result register 4	ADC	591
ADA0CR4H	A/D conversion result register 4H	ADC	591
ADA0CR5	A/D conversion result register 5	ADC	591
ADA0CR5H	A/D conversion result register 5H	ADC	591
ADA0CR6	A/D conversion result register 6	ADC	591
ADA0CR6H	A/D conversion result register 6H	ADC	591
ADA0CR7	A/D conversion result register 7	ADC	591
ADA0CR7H	A/D conversion result register 7H	ADC	591
ADA0CR8	A/D conversion result register 8	ADC	591
ADA0CR8H	A/D conversion result register 8H	ADC	591
ADA0CR9	A/D conversion result register 9	ADC	591
ADA0CR9H	A/D conversion result register 9H	ADC	591
ADA0CR10	A/D conversion result register 10	ADC	591
ADA0CR10H	A/D conversion result register 10H	ADC	591
ADA0CR11	A/D conversion result register 11	ADC	591
ADA0CR11H	A/D conversion result register 11H	ADC	591
ADA0CR12	A/D conversion result register 12	ADC	591
ADA0CR12H	A/D conversion result register 12H	ADC	591
ADA0CR13	A/D conversion result register 13	ADC	591
ADA0CR13H	A/D conversion result register 13H	ADC	591
ADA0CR14	A/D conversion result register 14	ADC	591
ADA0CR14H	A/D conversion result register 14H	ADC	591
ADA0CR15	A/D conversion result register 15	ADC	591
ADA0CR15H	A/D conversion result register 15H	ADC	591
ADA0CR16	A/D conversion result register 16	ADC	591
ADA0CR16H	A/D conversion result register 16H	ADC	591
ADA0CR17	A/D conversion result register 17	ADC	591
ADA0CR17H	A/D conversion result register 17H	ADC	591
ADA0CR18	A/D conversion result register 18	ADC	591
ADA0CR18H	A/D conversion result register 18H	ADC	591
ADA0CR19	A/D conversion result register 19	ADC	591

(2/12)

Symbol	Name	Unit	Page
ADA0CR19H	A/D conversion result register 19H	ADC	591
ADA0CR20	A/D conversion result register 20	ADC	591
ADA0CR20H	A/D conversion result register 20H	ADC	591
ADA0CR21	A/D conversion result register 21	ADC	591
ADA0CR21H	A/D conversion result register 21H	ADC	591
ADA0CR22	A/D conversion result register 22	ADC	591
ADA0CR22H	A/D conversion result register 22H	ADC	591
ADA0CR23	A/D conversion result register 23	ADC	591
ADA0CR23H	A/D conversion result register 23H	ADC	591
ADA0M0	A/D converter mode register 0	ADC	586
ADA0M1	A/D converter mode register 1	ADC	588
ADA0M2	A/D converter mode register 2	ADC	589
ADA0PFM	Power-fail compare mode register	ADC	593
ADA0PFT	Power-fail compare threshold value register	ADC	593
ADA0S	A/D converter channel specification register 0	ADC	590
ADIC	Interrupt control register	INTC	841
AWC	Address wait control register	BCU	252
BCC	Bus cycle control register	BCU	253
BSC	Bus size configuration register bus	BCU	242
CB0CTL0	CSIB0 control register 0	CSI	675
CB0CTL1	CSIB0 control register 1	CSI	678
CB0CTL2	CSIB0 control register 2	CSI	679
CB0RIC	Interrupt control register	INTC	841
CB0RX	CSIB0 receive data register	CSI	674
CB0RXL	CSIB0 receive data register L	CSI	674
CB0STR	CSIB0 status register	CSI	681
CB0TIC	Interrupt control register	INTC	841
CB0TX	CSIB0 transmit data register	CSI	674
CB0TXL	CSIB0 transmit data register L	CSI	674
CB1CTL0	CSIB1 control register 0	CSI	675
CB1CTL1	CSIB1 control register 1	CSI	678
CB1CTL2	CSIB1 control register 2	CSI	679
CB1RIC	Interrupt control register	INTC	841
CB1RX	CSIB1 receive data register	CSI	674
CB1RXL	CSIB1 receive data register L	CSI	674
CB1STR	CSIB1 status register	CSI	681
CB1TIC	Interrupt control register	INTC	841
CB1TX	CSIB1 transmit data register	CSI	674
CB1TXL	CSIB1 transmit data register L	CSI	674
CB2CTL0	CSIB2 control register 0	CSI	675
CB2CTL1	CSIB2 control register 1	CSI	678
CB2CTL2	CSIB2 control register 2	CSI	679
CB2RIC	Interrupt control register	INTC	841
CB2RX	CSIB2 receive data register	CSI	674

(3/12)

Symbol	Name	Unit	Page
CB2RXL	CSIB2 receive data register L	CSI	674
CB2STR	CSIB2 status register	CSI	681
CB2TIC	Interrupt control register	INTC	841
CB2TX	CSIB2 transmit data register	CSI	674
CB2TXL	CSIB2 transmit data register L	CSI	674
CCLS	CPU operation clock status register	CG	271
CLM	Clock monitor mode register	CLM	902
DADC0	DMA addressing control register 0	DMA	799
DADC1	DMA addressing control register 1	DMA	799
DADC2	DMA addressing control register 2	DMA	799
DADC3	DMA addressing control register 3	DMA	799
DBC0	DMA transfer count register 0	DMA	798
DBC1	DMA transfer count register 1	DMA	798
DBC2	DMA transfer count register 2	DMA	798
DBC3	DMA transfer count register 3	DMA	798
DCHC0	DMA channel control register 0	DMA	800
DCHC1	DMA channel control register 1	DMA	800
DCHC2	DMA channel control register 2	DMA	800
DCHC3	DMA channel control register 3	DMA	800
DDA0H	DMA destination address register 0H	DMA	797
DDA0L	DMA destination address register 0L	DMA	797
DDA1H	DMA destination address register 1H	DMA	797
DDA1L	DMA destination address register 1L	DMA	797
DDA2H	DMA destination address register 2H	DMA	797
DDA2L	DMA destination address register 2L	DMA	797
DDA3H	DMA destination address register 3H	DMA	797
DDA3L	DMA destination address register 3L	DMA	797
DMAIC0	Interrupt control register	INTC	841
DMAIC1	Interrupt control register	INTC	841
DMAIC2	Interrupt control register	INTC	841
DMAIC3	Interrupt control register	INTC	841
DSA0H	DMA source address register 0H	DMA	796
DSA0L	DMA source address register 0L	DMA	796
DSA1H	DMA source address register 1H	DMA	796
DSA1L	DMA source address register 1L	DMA	796
DSA2H	DMA source address register 2H	DMA	796
DSA2L	DMA source address register 2L	DMA	796
DSA3H	DMA source address register 3H	DMA	796
DSA3L	DMA source address register 3L	DMA	796
DTFR0	DMA trigger factor register 0	DMA	801
DTFR1	DMA trigger factor register 1	DMA	801
DTFR2	DMA trigger factor register 2	DMA	801
DTFR3	DMA trigger factor register 3	DMA	801
DWC0	Data wait control register 0	BCU	250

(4/12)

Symbol	Name	Unit	Page
HZA0CTL0	High-impedance output control register 0	Motor	517
HZA0CTL1	High-impedance output control register 1	Motor	517
IIC0	IIC shift register 0	I^2C	736
IIC0IC	Interrupt control register	INTC	841
IICC0	IIC control register 0	I^2C	723
IICCL0	IIC clock select register 0	I^2C	733
IICF0	IIC flag register 0	I^2C	731
IICS0	IIC status register 0	I^2C	728
IICX0	IIC function expansion register 0	I^2C	734
IMR0	Interrupt mask register 0	INTC	844
IMR0H	Interrupt mask register 0H	INTC	844
IMR0L	Interrupt mask register 0L	INTC	844
IMR1	Interrupt mask register 1	INTC	844
IMR1H	Interrupt mask register 1H	INTC	844
IMR1L	Interrupt mask register 1L	INTC	844
IMR2	Interrupt mask register 2	INTC	844
IMR2H	Interrupt mask register 2H	INTC	844
IMR2L	Interrupt mask register 2L	INTC	844
IMR3	Interrupt mask register 3	INTC	844
IMR3H	Interrupt mask register 3H	INTC	844
IMR3L	Interrupt mask register 3L	INTC	844
IMR4	Interrupt mask register 4	INTC	844
IMR4H	Interrupt mask register 4H	INTC	844
IMR4L	Interrupt mask register 4L	INTC	844
IMR5	Interrupt mask register 5	INTC	844
IMR5H	Interrupt mask register 5H	INTC	844
IMR5L	Interrupt mask register 5L	INTC	844
INTF0	External interrupt falling edge specification register 0	INTC	858
INTF1	External interrupt falling edge specification register 1	INTC	859
INTF3	External interrupt falling edge specification register 3	INTC	860
INTF3H	External interrupt falling edge specification register 3H	INTC	860
INTF3L	External interrupt falling edge specification register 3L	INTC	860
INTF4	External interrupt falling edge specification register 4	INTC	861
INTF6L	External interrupt falling edge specification register 6L	INTC	862
INTF8	External interrupt falling edge specification register 8	INTC	863
INTF9H	External interrupt falling edge specification register 9H	INTC	864
INTR0	External interrupt rising edge specification register 0	INTC	858
INTR1	External interrupt rising edge specification register 1	INTC	859
INTR3	External interrupt rising edge specification register 3	INTC	860
INTR3H	External interrupt rising edge specification register 3H	INTC	860
INTR3L	External interrupt rising edge specification register 3L	INTC	860
INTR4	External interrupt rising edge specification register 4	INTC	861
INTR6L	External interrupt rising edge specification register 6L	INTC	862
INTR8	External interrupt rising edge specification register 8	INTC	863

(5/12)

Symbol	Name	Unit	Page
INTR9H	External interrupt rising edge specification register 9H	INTC	864
ISPR	In-service priority register	INTC	848
KRIC	Interrupt control register	INTC	841
KRM	Key return mode register	KR	870
LOCKR	Lock register	CG	277
LVIHIC	Interrupt control register	INTC	841
LVILIC	Interrupt control register	INTC	841
LVIM	Low-voltage detection register	LVI	909
LVIS	Low-voltage detection level select register	LVI	910
MCM	Main clock mode register	CG	270
NFC	Noise elimination control register	INTC	865
OCDM	On-chip debug mode register	DCU	957
OCKS0	IIC division clock select register 0	I^2C	736
OSTC	Oscillation stabilization time count status register	Standby	877
OSTS	Oscillation stabilization time select	Standby	875
P0	Port 0	Port	115
P1	Port 1	Port	118
P3	Port 3	Port	121
P3H	Port 3H	Port	121
P3L	Port 3L	Port	121
P4	Port 4	Port	128
P5	Port 5	Port	131
P6	Port 6	Port	136
P6H	Port 6H	Port	136
P6L	Port 6L	Port	136
P7H	Port 7H	Port	141
P7L	Port 7L	Port	141
P8	Port 8	Port	144
P9	Port 9	Port	147
P9H	Port 9H	Port	147
P9L	Port 9L	Port	147
P12	Port 12	Port	165
PCC	Processor clock control register	CG	264
PCD	Port CD	Port	167
PCLM	Programmable clock mode register	CG	272
PCM	Port CM	Port	169
PCS	Port CS	Port	173
PCT	Port CT	Port	177
PDL	Port DL	Port	181
PDLH	Port DLH	Port	181
PDLL	Port DLL	Port	181
PEMU1	Peripheral emulation register 1	CPU	915
PF9H	Port function register 9H	Port	164
PFC0	Port function control register 0	Port	116

(6/12)

Symbol	Name	Unit	Page
PFC3L	Port function control register 3L	Port	125
PFC4	Port function control register 4	Port	129
PFC5	Port function control register 5	Port	133
PFC6	Port function control register 6	Port	138
PFC6H	Port function control register 6H	Port	138
PFC6L	Port function control register 6L	Port	138
PFC9	Port function control register 9	Port	154
PFC9H	Port function control register 9H	Port	154
PFC9L	Port function control register 9L	Port	154
PFCE0	Port function control expansion register 0	Port	117
PFCE3L	Port function control expansion register 3L	Port	125
PFCE4	Port function control expansion register 4	Port	130
PFCE5	Port function control expansion register 5	Port	133
PFCE9	Port function control expansion register 9	Port	155
PFCE9H	Port function control expansion register 9H	Port	155
PFCE9L	Port function control expansion register 9L	Port	155
PIC0	Interrupt control register	INTC	841
PIC1	Interrupt control register	INTC	841
PIC2	Interrupt control register	INTC	841
PIC3	Interrupt control register	INTC	841
PIC4	Interrupt control register	INTC	841
PIC5	Interrupt control register	INTC	841
PIC6	Interrupt control register	INTC	841
PIC7	Interrupt control register	INTC	841
PIC8	Interrupt control register	INTC	841
PIC9	Interrupt control register	INTC	841
PIC10	Interrupt control register	INTC	841
PIC11	Interrupt control register	INTC	841
PIC12	Interrupt control register	INTC	841
PIC13	Interrupt control register	INTC	841
PIC14	Interrupt control register	INTC	841
PLLCTL	PLL control register	CG	276
PLLS	PLL lockup time specification register	CG	278
PM0	Port mode register 0	Port	115
PM1	Port mode register 1	Port	118
PM3	Port mode register 3	Port	122
PM3H	Port mode register 3H	Port	122
PM3L	Port mode register 3L	Port	122
PM4	Port mode register 4	Port	129
PM5	Port mode register 5	Port	132
PM6	Port mode register 6	Port	136
PM6H	Port mode register 6H	Port	136
PM6L	Port mode register 6L	Port	136
PM7H	Port mode register 7H	Port	142

(7/12)

Symbol	Name	Unit	Page
PM7L	Port mode register 7L	Port	142
PM8	Port mode register 8	Port	145
PM9	Port mode register 9	Port	148
PM9H	Port mode register 9H	Port	148
PM9L	Port mode register 9L	Port	148
PM12	Port mode register 12	Port	166
PMC0	Port mode control register 0	Port	116
PMC1	Port mode control register 1	Port	119
PMC3	Port mode control register 3	Port	123
PMC3H	Port mode control register 3H	Port	123
PMC3L	Port mode control register 3L	Port	123
PMC4	Port mode control register 4	Port	129
PMC5	Port mode control register 5	Port	132
PMC6	Port mode control register 6	Port	137
PMC6H	Port mode control register 6H	Port	137
PMC6L	Port mode control register 6L	Port	137
PMC7H	Port mode control register 7H	Port	143
PMC7L	Port mode control register 7L	Port	143
PMC8	Port mode control register 8	Port	145
PMC9	Port mode control register 9	Port	149
PMC9H	Port mode control register 9H	Port	149
PMC9L	Port mode control register 9L	Port	149
PMC12	Port mode control register 12	Port	166
PMCCM	Port mode control register CM	Port	171
PMCCS	Port mode control register CS	Port	175
PMCCT	Port mode control register CT	Port	179
PMCD	Port mode register CD	Port	167
PMCDL	Port mode control register DL	Port	184
PMCDLH	Port mode control register DLH	Port	184
PMCDLL	Port mode control register DLL	Port	184
PMCM	Port mode register CM	Port	170
PMCS	Port mode register CS	Port	174
PMCT	Port mode register CT	Port	178
PMDL	Port mode register DL	Port	182
PMDLH	Port mode register DLH	Port	182
PMDLL	Port mode register DLL	Port	182
PRCMD	Command register	CPU	99
PRSCM0	Prescaler compare register 0	WT	570
PRSM0	Prescaler mode register 0	WT	569
PSC	Power save control register	CG	873
PSMR	Power save mode register	CG	874
PU0	Pull-up resistor option register 0	Port	117
PU1	Pull-up resistor option register 1	Port	119
PU3	Pull-up resistor option register 3	Port	127

(8/12)

Symbol	Name	Unit	Page
PU3H	Pull-up resistor option register 3H	Port	127
PU3L	Pull-up resistor option register 3L	Port	127
PU4	Pull-up resistor option register 4	Port	130
PU5	Pull-up resistor option register 5	Port	134
PU6	Pull-up resistor option register 6	Port	139
PU6H	Pull-up resistor option register 6H	Port	139
PU6L	Pull-up resistor option register 6L	Port	139
PU8	Pull-up resistor option register 8	Port	145
PU9	Pull-up resistor option register 9	Port	163
PU9H	Pull-up resistor option register 9H	Port	163
PU9L	Pull-up resistor option register 9L	Port	163
RAMS	Internal RAM data status register	CG	910
RCM	Internal oscillation mode register	CG	269
RESF	Reset source flag register	CG	894
SELCNT0	Selector operation control register 0	Timer	393
SELCNT1	Selector operation control register 1	Timer	394
SELCNT3	Selector operation control register 3	Timer	394
SELCNT4	Selector operation control register 4	Timer	282
SFC0	SSCG frequency control register 0	CG	280
SFC1	SSCG frequency control register 1	CG	281
SSCGCTL	SSCG control register	CG	279
SVA0	Slave address register	I^2C	737
SYS	System status register	Timer	100
TAA0CCIC0	Interrupt control register	INTC	841
TAA0CCIC1	Interrupt control register	INTC	841
TAA0CCR0	TAA0 capture/compare register 0	Timer	300
TAA0CCR1	TAA0 capture/compare register 1	Timer	302
TAA0CNT	TAA0 counter read buffer register	Timer	304
TAA0CTL0	TAA0 control register 0	Timer	292
TAA0CTL1	TAA0 control register 1	Timer	293
TAA0IOC0	TAA0 I/O control register 0	Timer	295
TAA0IOC1	TAA0 I/O control register 1	Timer	296
TAA0IOC2	TAA0 I/O control register 2	Timer	297
TAA0OPT0	TAA0 option register 0	Timer	298
TAA0OVIC	Interrupt control register	INTC	841
TAA1CCIC0	Interrupt control register	INTC	841
TAA1CCIC1	Interrupt control register	INTC	841
TAA1CCR0	TAA1 capture/compare register 0	Timer	300
TAA1CCR1	TAA1 capture/compare register 1	Timer	302
TAA1CNT	TAA1 counter read buffer register	Timer	304
TAA1CTL0	TAA1 control register 0	Timer	292
TAA1CTL1	TAA1 control register 1	Timer	293
TAA1IOC0	TAA1 I/O control register 0	Timer	295
TAA1IOC1	TAA1 I/O control register 1	Timer	296

(9/12)

Symbol	Name	Unit	Page
TAA1IOC2	TAA1 I/O control register 2	Timer	297
TAA1OPT0	TAA1 option register 0	Timer	298
TAA1OPT1	TAA1 option register 1	Timer	299
TAA1OVIC	Interrupt control register	INTC	841
TAA2CCIC0	Interrupt control register	INTC	841
TAA2CCIC1	Interrupt control register	INTC	841
TAA2CCR0	TAA2 capture/compare register 0	Timer	300
TAA2CCR1	TAA2 capture/compare register 1	Timer	302
TAA2CNT	TAA2 counter read buffer register	Timer	304
TAA2CTL0	TAA2 control register 0	Timer	292
TAA2CTL1	TAA2 control register 1	Timer	293
TAA2IOC0	TAA2 I/O control register 0	Timer	295
TAA2IOC1	TAA2 I/O control register 1	Timer	296
TAA2IOC2	TAA2 I/O control register 2	Timer	297
TAA2OPT0	TAA2 option register	Timer	298
TAA2OVIC	Interrupt control register	INTC	841
TAA3CCIC0	Interrupt control register	INTC	841
TAA3CCIC1	Interrupt control register	INTC	841
TAA3CCR0	TAA3 capture/compare register 0	Timer	300
TAA3CCR1	TAA3 capture/compare register 1	Timer	302
TAA3CNT	TAA3 counter read buffer register	Timer	304
TAA3CTL0	TAA3 control register 0	Timer	292
TAA3CTL1	TAA3 control register 1	Timer	293
TAA3IOC0	TAA3 I/O control register 0	Timer	295
TAA3IOC1	TAA3 I/O control register 1	Timer	296
TAA3IOC2	TAA3 I/O control register 2	Timer	297
TAA3OPT0	TAA3 option register 0	Timer	298
TAA3OPT1	TAA3 option register 1	Timer	299
TAA3OVIC	Interrupt control register	INTC	841
TAA3OVIC	Interrupt control register	INTC	841
TAA4CCIC0	Interrupt control register	INTC	841
TAA4CCIC1	Interrupt control register	INTC	841
TAA4CCR0	TAA4 capture/compare register 0	Timer	300
TAA4CCR1	TAA4 capture/compare register 1	Timer	302
TAA4CNT	TAA4 counter read buffer register	Timer	304
TAA4CTL0	TAA4 control register 0	Timer	292
TAA4CTL1	TAA4 control register 1	Timer	293
TAA4IOC0	TAA4 I/O control register 0	Timer	295
TAA4IOC1	TAA4 I/O control register 1	Timer	296
TAA4IOC2	TAA4 I/O control register 2	Timer	297
TAA4OPT0	TAA4 option register 0	Timer	298
TAB0CCIC0	Interrupt control register	INTC	841
TAB0CCIC1	Interrupt control register	INTC	841
TAB0CCIC2	Interrupt control register	INTC	841

(10/12)

Symbol	Name	Unit	Page
TAB0CCIC3	Interrupt control register	INTC	841
TAB0CCR0	TAB0 capture/compare register 0	Timer	407
TAB0CCR1	TAB0 capture/compare register 1	Timer	409
TAB0CCR2	TAB0 capture/compare register 2	Timer	411
TAB0CCR3	TAB0 capture/compare register 3	Timer	413
TAB0CNT	TAB0 counter read buffer register	Timer	415
TAB0CTL0	TAB0 control register 0	Timer	400
TAB0CTL1	TAB0 control register 1	Timer	401
TAB0DTC	TAB0 dead-time compare register	Timer	510
TAB0IOC0	TAB0 I/O control register 0	Timer	403
TAB0IOC1	TAB0 I/O control register 1	Timer	404
TAB0IOC2	TAB0 I/O control register 2	Timer	405
TAB0IOC3	TAB0 I/O control register 3	Timer	515
TAB0OPT0	TAB0 option register 0	Timer	406
TAB0OPT1	TAB0 option register 1	Timer	512
TAB0OPT2	TAB0 option register 2	Timer	513
TAB0OVIC	Interrupt control register	INTC	841
TAB1CCIC0	Interrupt control register	INTC	841
TAB1CCIC1	Interrupt control register	INTC	841
TAB1CCIC2	Interrupt control register	INTC	841
TAB1CCIC3	Interrupt control register	INTC	841
TAB1CCR0	TAB1 capture/compare register 0	Timer	407
TAB1CCR1	TAB1 capture/compare register 1	Timer	409
TAB1CCR2	TAB1 capture/compare register 2	Timer	411
TAB1CCR3	TAB1 capture/compare register 3	Timer	413
TAB1CNT	TAB1 counter read buffer register	Timer	415
TAB1CTL0	TAB1 control register 0	Timer	400
TAB1CTL1	TAB1 control register 1	Timer	401
TAB1IOC0	TAB1 I/O control register 0	Timer	403
TAB1IOC1	TAB1 I/O control register 1	Timer	404
TAB1IOC2	TAB1 I/O control register 2	Timer	405
TAB1OPT0	TAB1 option register 0	Timer	406
TAB1OVIC	Interrupt control register	INTC	841
TAB2CCIC0	Interrupt control register	INTC	841
TAB2CCIC1	Interrupt control register	INTC	841
TAB2CCIC2	Interrupt control register	INTC	841
TAB2CCIC3	Interrupt control register	INTC	841
TAB2CCR0	TAB2 capture/compare register 0	Timer	407
TAB2CCR1	TAB2 capture/compare register 1	Timer	409
TAB2CCR2	TAB2 capture/compare register 2	Timer	411
TAB2CCR3	TAB2 capture/compare register 3	Timer	413
TAB2CNT	TAB2 counter read buffer register	Timer	415
TAB2CTL0	TAB2 control register 0	Timer	400
TAB2CTL1	TAB2 control register 1	Timer	401

(11/12)

Symbol	Name	Unit	Page
TAB2IOC0	TAB2 I/O control register 0	Timer	403
TAB2IOC1	TAB2 I/O control register 1	Timer	404
TAB2IOC2	TAB2 I/O control register 2	Timer	405
TAB2OPT0	TAB2 option register 0	Timer	406
TAB2OVIC	Interrupt control register	INTC	841
TM0CMP0	TMM0 compare register 0	Timer	499
TM0CTL0	TMM0 control register 0	Timer	500
TM0EQIC0	Interrupt control register	INTC	841
UD0CTL0	UARTD0 control register 0	UART	621
UD0CTL1	UARTD0 control register 1	UART	661
UD0CTL2	UARTD0 control register 2	UART	662
UD0OPT0	UARTD0 option control register 0	UART	623
UD0OPT1	UARTD0 option control register 1	UART	625
UD0RIC	Interrupt control register	INTC	841
UD0RX	UARTD0 receive data register	UART	629
UD0SIC	Interrupt control register	INTC	841
UD0STR	UARTD0 status register	UART	626
UD0TIC	Interrupt control register	INTC	841
UD0TX	UARTD0 transmit data register	UART	630
UD1CTL0	UARTD1 control register 0	UART	621
UD1CTL1	UARTD1 control register 1	UART	661
UD1CTL2	UARTD1 control register 2	UART	662
UD1OPT0	UARTD1 option control register 0	UART	623
UD1OPT1	UARTD1 option control register 1	UART	625
UD1RIC	Interrupt control register	INTC	841
UD1RX	UARTD1 receive data register	UART	629
UD1SIC	Interrupt control register	INTC	841
UD1STR	UARTD1 status register	UART	626
UD1TIC	Interrupt control register	INTC	841
UD1TX	UARTD1 transmit data register	UART	630
UD2CTL0	UARTD2 control register 0	UART	621
UD2CTL1	UARTD2 control register 1	UART	661
UD2CTL2	UARTD2 control register 2	UART	662
UD2OPT0	UARTD2 option control register 0	UART	623
UD2OPT1	UARTD2 option control register 1	UART	625
UD2RIC	Interrupt control register	INTC	841
UD2RX	UARTD2 receive data register	UART	629
UD2SIC	Interrupt control register	INTC	841
UD2STR	UARTD2 status register	UART	626
UD2TIC	Interrupt control register	INTC	841
UD2TX	UARTD2 transmit data register	UART	630
UD3CTL0	UARTD3 control register 0	UART	621
UD3CTL1	UARTD3 control register 1	UART	661
UD3CTL2	UARTD3 control register 2	UART	662

(12/12)

Symbol	Name	Unit	Page
UD3OPT0	UARTD3 option control register 0	UART	623
UD3OPT1	UARTD3 option control register 1	UART	625
UD3RIC	Interrupt control register	INTC	841
UD3RX	UARTD3 receive data register	UART	629
UD3SIC	Interrupt control register	INTC	841
UD3STR	UARTD3 status register	UART	626
UD3TIC	Interrupt control register	INTC	841
UD3TX	UARTD3 transmit data register	UART	630
UD4CTL0	UARTD4 control register 0	UART	621
UD4CTL1	UARTD4 control register 1	UART	661
UD4CTL2	UARTD4 control register 2	UART	662
UD4OPT0	UARTD4 option control register 0	UART	623
UD4OPT1	UARTD4 option control register 1	UART	625
UD4RIC	Interrupt control register	INTC	841
UD4RX	UARTD4 receive data register	UART	629
UD4SIC	Interrupt control register	INTC	841
UD4STR	UARTD4 status register	UART	626
UD4TIC	Interrupt control register	INTC	841
UD4TX	UARTD4 transmit data register	UART	630
UD5CTL0	UARTD5 control register 0	UART	621
UD5CTL1	UARTD5 control register 1	UART	661
UD5CTL2	UARTD5 control register 2	UART	662
UD5OPT0	UARTD5 option control register 0	UART	623
UD5OPT1	UARTD5 option control register 1	UART	625
UD5RIC	Interrupt control register	INTC	841
UD5RX	UARTD5 receive data register	UART	629
UD5SIC	Interrupt control register	INTC	841
UD5STR	UARTD5 status register	UART	626
UD5TIC	Interrupt control register	INTC	841
UD5TX	UARTD5 transmit data register	UART	630
VSWC	System wait control register	CPU	101
WDTE	Watchdog timer enable register	WDT	580
WDTM2	Watchdog timer mode register 2	WDT	578
WTIC	Interrupt control register	INTC	841
WTIIC	Interrupt control register	INTC	841
WTM	Watch timer operation mode register	WT	571

APPENDIX C INSTRUCTION SET LIST

C.1 Conventions

(1) Register symbols used to describe operands

Register Symbol	Explanation
reg1	General-purpose registers: Used as source registers.
reg2	General-purpose registers: Used mainly as destination registers. Also used as source register in some instructions.
reg3	General-purpose registers: Used mainly to store the remainders of division results and the higher 32 bits of multiplication results.
bit#3	3-bit data for specifying the bit number
immX	X bit immediate data
dispX	X bit displacement data
regID	System register number
vector	5-bit data that specifies the trap vector (00H to 1FH)
cccc	4-bit data that shows the conditions code
sp	Stack pointer (r3)
ep	Element pointer (r30)
listX	X item register list

(2) Register symbols used to describe opcodes

Register Symbol	Explanation
R	1-bit data of a code that specifies reg1 or regID
r	1-bit data of the code that specifies reg2
w	1-bit data of the code that specifies reg3
d	1-bit displacement data
I	1-bit immediate data (indicates the higher bits of immediate data)
i	1-bit immediate data
cccc	4-bit data that shows the condition codes
CCCC	4-bit data that shows the condition codes of Bcond instruction
bbb	3-bit data for specifying the bit number
L	1-bit data that specifies a program register in the register list

(3) Register symbols used in operations

Register Symbol	Explanation
←	Input for
GR [ ]	General-purpose register
SR [ ]	System register
zero-extend (n)	Expand n with zeros until word length.
sign-extend (n)	Expand n with signs until word length.
load-memory (a, b)	Read size b data from address a.
store-memory (a, b, c)	Write data b into address a in size c.
load-memory-bit (a, b)	Read bit b of address a.
store-memory-bit (a, b, c)	Write c to bit b of address a.
saturated (n)	Execute saturated processing of n (n is a 2's complement). If, as a result of calculations, n ≥ 7FFFFFFFH, let it be 7FFFFFFFH. n ≤ 80000000H, let it be 80000000H.
result	Reflects the results in a flag.
Byte	Byte (8 bits)
Halfword	Half word (16 bits)
Word	Word (32 bits)
+	Addition
–	Subtraction
ll	Bit concatenation
×	Multiplication
÷	Division
%	Remainder from division results
AND	Logical product
OR	Logical sum
XOR	Exclusive OR
NOT	Logical negation
logically shift left by	Logical shift left
logically shift right by	Logical shift right
arithmetically shift right by	Arithmetic shift right

(4) Register symbols used in execution clock

Register Symbol	Explanation
i	If executing another instruction immediately after executing the first instruction (issue).
r	If repeating execution of the same instruction immediately after executing the first instruction (repeat).
l	If using the results of instruction execution in the instruction immediately after the execution (latency).

(5) Register symbols used in flag operations

Identifier	Explanation
(Blank)	No change
0	Clear to 0
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 or Z) = 1	Not higher (Less than or equal)
1 0 1 1	(CY 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 xor OV) = 1	Less than signed
1 1 1 0	(S xor OV) = 0	Greater than or equal signed
0 1 1 1	((S xor OV) or Z) = 1	Less than or equal signed
1 1 1 1	((S xor OV) or Z) = 0	Greater than signed

C.2 Instruction Set (in Alphabetical Order)

(1/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010010iiiii	GR[reg2]←GR[reg2]+sign-extend(imm5)		1	1	1	×	×	×	×	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1	×	×	×	×	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]		1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND zero-extend(imm16)		1	1	1		0	×	×	
Bcond	disp9	ddddd1011dddcccc **Note 1**	if conditions are satisfied then PC←PC+sign-extend(disp9)	When conditions are satisfied	2 Note 2	2 Note 2	2 Note 2					
				When conditions are not satisfied	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 wwwww01101000010	GR[reg3]←GR[reg2] (23 : 16) ll GR[reg2] (31 : 24) ll GR[reg2] (7 : 0) ll GR[reg2] (15 : 8)		1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 wwwww01101000000	GR[reg3]←GR[reg2] (7 : 0) ll GR[reg2] (15 : 8) ll GR [reg2] (23 : 16) ll GR[reg2] (31 : 24)		1	1	1	×	0	×	×	
CALLT	imm6	0000001000iiiiii	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+zero-extend(imm6 logically shift left by 1) PC←CTBP+zero-extend(Load-memory(adr,Halfword))		4	4	4					
CLR1	bit#3,disp16[reg1]	10bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)		3 Note 3	3 Note 3	3 Note 3				×	
CMOV	cccc,imm5,reg2,reg3	rrrrr111111iiiii wwwww011000cccc0	if conditions are satisfied then GR[reg3]←sign-extended(imm5) else GR[reg3]←GR[reg2]		1	1	1					
	cccc,reg1,reg2,reg3	rrrrr111111RRRR wwwww011001cccc0	if conditions are satisfied then GR[reg3]←GR[reg1] else GR[reg3]←GR[reg2]		1	1	1					
CMP	reg1,reg2	rrrrr001111RRRRR	result←GR[reg2]–GR[reg1]		1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010011iiiii	result←GR[reg2]–sign-extend(imm5)		1	1	1	×	×	×	×	
CTRET		0000011111100000 0000000101000100	PC←CTPC PSW←CTPSW		3	3	3	R	R	R	R	R
DBRET		0000011111100000 0000000101000110	PC←DBPC PSW←DBPSW		3	3	3	R	R	R	R	R

(2/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
DBTRAP		1111100001000000	DBPC←PC+2 (restored PC) DBPSW←PSW PSW.NP←1 PSW.EP←1 PSW.ID←1 PC←00000060H	3	3	3					
DI		0000011111100000 0000000101100000	PSW.ID←1	1	1	1					
DISPOSE	imm5,list12	0000011001iiiiiL LLLLLLLLLLL00000	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	imm5,list12,[reg1]	0000011001iiiiiL LLLLLLLLLLRRRRR **Note 5**	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded PC←GR[reg1]	n+3 Note 4	n+3 Note 4	n+3 Note 4					
DIV	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000000	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVH	reg1,reg2	rrrrr000010RRRRR	GR[reg2]←GR[reg2]÷GR[reg1]Note 6	35	35	35		×	×	×	
	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000000	GR[reg2]←GR[reg2]÷GR[reg1]Note 6 GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVHU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000010	GR[reg2]←GR[reg2]÷GR[reg1]Note 6 GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
DIVU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000010	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
EI		1000011111100000 0000000101100000	PSW.ID←0	1	1	1					
HALT		0000011111100000 0000000100100000	Stop	1	1	1					
HSW	reg2,reg3	rrrrr11111100000 wwwww01101000100	GR[reg3]←GR[reg2](15 : 0) ll GR[reg2] (31 : 16)	1	1	1	×	0	×	×	
JARL	disp22,reg2	rrrrr11110ddddddd ddddddddddddddd0 **Note 7**	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	3	3	3					
JR	disp22	0000011110ddddddd ddddddddddddddd0 **Note 7**	PC←PC+sign-extend(disp22)	2	2	2					
LD.B	disp16[reg1],reg2	rrrrr111000RRRRR ddddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 11					
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRRR ddddddddddddddd1 **Notes 8, 10**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 11					

(3/6)

Mnemonic	Operand	Opcode	Operation		Execution Clock			Flags				
					i	r	l	CY	OV	S	Z	SAT
LD.H	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Halfword))		1	1	Note 11					
LDSR	reg2,regID	rrrrr111111RRRRR 0000000000100000 **Note 12**	SR[regID]←GR[reg2]	Other than regID = PSW	1	1	1					
				regID = PSW	1	1	1	×	×	×	×	×
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Halfword)		1	1	Note 11					
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←Load-memory(adr,Word)		1	1	Note 11					
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]		1	1	1					
	imm5,reg2	r r r r r 0 1 0 0 0 0 i i i i i	GR[reg2]←sign-extend(imm5)		1	1	1					
	imm32,reg1	00000110001RRRRR i i i i i i i i i i i i i i i i l l l l l l l l l l l l l l l l	GR[reg1]←imm32		2	2	2					
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1]+sign-extend(imm16)		1	1	1					
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1]+(imm16 ll 0^{16})		1	1	1					
MUL	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000 **Note 14**	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1]		1	4	5					
	imm9,reg2,reg3	r r r r r 1 1 1 1 1 1 i i i i i w w w w w 0 1 0 0 1 l l l l 0 0 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xsign-extend(imm9)		1	4	5					
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2]Note 6xGR[reg1]Note 6		1	1	2					
	imm5,reg2	r r r r r 0 1 0 1 1 1 i i i i i	GR[reg2]←GR[reg2]Note 6xsign-extend(imm5)		1	1	2					
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1]Note 6ximm16		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	r r r r r 1 1 1 1 1 1 i i i i i w w w w w 0 1 0 0 1 l l l l 1 0 **Note 13**	GR[reg3] ll GR[reg2]←GR[reg2]xzero-extend(imm9)		1	4	5					
NOP		0000000000000000	Pass at least one clock cycle doing nothing.		1	1	1					
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])		1	1	1		0	×	×	
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,Z flag)		3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,Z flag)		3 Note 3	3 Note 3	3 Note 3				×	

(4/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
OR	reg1,reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×	
ORI	imm16,reg1,reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×	
PREPARE	list12,imm5	0000011110iiiiiL LLLLLLLLLLL00001	Store-memory(sp–4,GR[reg in list12],Word) sp←sp–4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5)	n+1 Note 4	n+1 Note 4	n+1 Note 4					
	list12,imm5, sp/imm Note 15	0000011110iiiiiL LLLLLLLLLLLff011 imm16/imm32 **Note 16**	Store-memory(sp–4,GR[reg in list12],Word) sp←sp+4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend (imm5) ep←sp/imm	n+2 Note 4 Note 17	n+2 Note 4 Note 17	n+2 Note 4 Note 17					
RETI		0000011111100000 0000000101000000	if PSW.EP=1 then PC ←EIPC PSW ←EIPSW else if PSW.NP=1 then PC ←FEPC PSW ←FEPSW else PC ←EIPC PSW ←EIPSW	3	3	3	R	R	R	R	R
SAR	reg1,reg2	rrrrr111111RRRRR 0000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010101iiiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend (imm5)	1	1	1	×	0	×	×	
SASF	cccc,reg2	rrrrr1111110cccc 0000001000000000	if conditions are satisfied then GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000001H else GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000000H	1	1	1					
SATADD	reg1,reg2	rrrrr000110RRRRR	GR[reg2]←saturated(GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×
	imm5,reg2	rrrrr010001iiiii	GR[reg2]←saturated(GR[reg2]+sign-extend(imm5)	1	1	1	×	×	×	×	×
SATSUB	reg1,reg2	rrrrr000101RRRRR	GR[reg2]←saturated(GR[reg2]–GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16,reg1,reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiii	GR[reg2]←saturated(GR[reg1]–sign-extend(imm16)	1	1	1	×	×	×	×	×
SATSUBR	reg1,reg2	rrrrr000100RRRRR	GR[reg2]←saturated(GR[reg1]–GR[reg2])	1	1	1	×	×	×	×	×
SETF	cccc,reg2	rrrrr1111110cccc 0000000000000000	If conditions are satisfied then GR[reg2]←00000001H else GR[reg2]←00000000H	1	1	1					

(5/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SET1	bit#3,disp16[reg1]	00bbb111110RRRRR 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]	rrrrr111111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 Note 3	3 Note 3	3 Note 3				×	
SHL	reg1,reg2	rrrrr111111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010110iiiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×	
SHR	reg1,reg2	rrrrr111111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010100iiiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SLD.B	disp7[ep],reg2	rrrrr0110ddddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.BU	disp4[ep],reg2	rrrrr0000110dddd **Note 18**	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	Note 9					
SLD.H	disp8[ep],reg2	rrrrr1000ddddddd **Note 19**	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Halfword))	1	1	Note 9					
SLD.HU	disp5[ep],reg2	rrrrr0000111dddd **Notes 18, 20**	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,Halfword))	1	1	Note 9					
SLD.W	disp8[ep],reg2	rrrrr1010ddddddd0 **Note 21**	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	Note 9					
SST.B	reg2,disp7[ep]	rrrrr0111ddddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1					
SST.H	reg2,disp8[ep]	rrrrr1001ddddddd **Note 19**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Halfword)	1	1	1					
SST.W	reg2,disp8[ep]	rrrrr1010ddddddd1 **Note 21**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1					
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR ddddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1					
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Halfword)	1	1	1					
ST.W	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Word)	1	1	1					
STSR	regID,reg2	rrrrr111111RRRRR 0000000001000000	GR[reg2]←SR[regID]	1	1	1					

(6/6)

Mnemonic	Operand	Opcode	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1,reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]–GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]–GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	00000000010RRRRR	adr←(PC+2) + (GR [reg1] logically shift left by 1) PC←(PC+2) + (sign-extend (Load-memory (adr,Halfword)) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←sign-extend (GR[reg1] (7 : 0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←sign-extend (GR[reg1] (15 : 0))	1	1	1					
TRAP	vector	0 0 0 0 0 1 1 1 1 1 1 iiiii 0000000100000000	EIPC ←PC+4 (Restored PC) EIPSW ←PSW ECR.EICC ←Interrupt code PSW.EP ←1 PSW.ID ←1 PC ←00000040H (when vector is 00H to 0FH) 00000050H (when vector is 10H to 1FH)	3	3	3					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR ddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit (adr,bit#3))	3 Note 3	3 Note 3	3 Note 3				×	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←GR[reg1] Z flag←Not (Load-memory-bit (adr,reg2))	3 Note 3	3 Note 3	3 Note 3				×	
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16,reg1,reg2	rrrrr110101RRRRR i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1] XOR zero-extend (imm16)	1	1	1		0	×	×	
ZXB	reg1	00000000100RRRRR	GR[reg1]←zero-extend (GR[reg1] (7 : 0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←zero-extend (GR[reg1] (15 : 0))	1	1	1					

Notes 1. ddddddddd: Higher 8 bits of disp9.

2. 3 if there is an instruction that rewrites the contents of the PSW immediately before.

3. If there is no wait state (3 + the number of read access wait states).

4. n is the total number of list12 load registers. (According to the number of wait states. Also, if there are no wait states, n is the total number of list12 registers. If n = 0, same operation as when n = 1)

5. RRRRR: other than 00000.

6. The lower halfword data only are valid.

7. ddddddddddddddddddddddd: The higher 21 bits of disp22.

8. ddddddddddddddd: The higher 15 bits of disp16.

9. According to the number of wait states (1 if there are no wait states).

10. b: bit 0 of disp16.

11. According to the number of wait states (2 if there are no wait states).

Notes 12. In this instruction, for convenience of mnemonic description, the source register is made reg2, but the reg1 field is used in the opcode. Therefore, the meaning of register specification in the mnemonic description and in the opcode differs from other instructions.

rrrrr = regID specification

RRRRR = reg2 specification

13. iiiii: Lower 5 bits of imm9.

IIII: Higher 4 bits of imm9.

14. Do not specify the same register for general-purpose registers reg1 and reg3.

15. sp/imm: specified by bits 19 and 20 of the sub-opcode.

16. ff = 00: Load sp in ep.

01: Load sign expanded 16-bit immediate data (bits 47 to 32) in ep.

10: Load 16-bit logically left shifted 16-bit immediate data (bits 47 to 32) in ep.

11: Load 32-bit immediate data (bits 63 to 32) in ep.

17. If imm = imm32, n + 3 clocks.

18. rrrrr: Other than 00000.

19. ddddddd: Higher 7 bits of disp8.

20. dddd: Higher 4 bits of disp5.

21. dddddd: Higher 6 bits of disp8.

<R>

APPENDIX D REVISION HISTORY

D.1 Major Revisions in This Edition

Page	Description
Throughout	• Under development → mass production μPD70F3747GB-GAH-AX, μPD70F3750GK-GAK-AX, μPD70F3752GC-UEU-AX, μPD70F3755GJ-GAE-AX, μPD70F3757GJ-GAE-AX
p. 241	Modification of **Table 5-4 Number of Clocks for Access (μPD70F3757)**
p. 260	Modification of **6.1 Overview**
p. 263	Modification of **6.2 (1) Main clock oscillator**
p. 276	Modification of **6.5.1 Overview**
p. 371	Modification of **Figure 7-36 Register Setting in Pulse Width Measurement Mode**
p. 814	Modification of **17.13 (4) (a) Temporarily stop transfer of all DMA channels**
p. 975	Addition of **29.4.1 (i) KYOCERA KINSEKI CORPORATION: Crystal resonator (T_A = –10 to +70°C)**
p. 977	Modification of **29.4.3 PLL characteristics**
p. 981	Modification of **29.6.3 Supply current**
p. 992	Modification of **29.9 (2) Serial write operation characteristics**
p. 997	Addition of **30.4.1 (i) KYOCERA KINSEKI CORPORATION: Crystal resonator (T_A = –10 to +70°C)**
p. 999	Modification of **30.4.3 PLL characteristics**
p. 1003	Modification of **30.6.3 Supply current**
p. 1014	Modification of **30.9 (2) Serial write operation characteristics**
p. 1019	Addition of **30.4.1 (i) KYOCERA KINSEKI CORPORATION: Crystal resonator (T_A = –10 to +70°C)**
p. 1021	Modification of **31.4.3 PLL characteristics**
p. 1025	Modification of **31.6.3 Supply current**
p. 1036	Modification of **31.9 (2) Serial write operation characteristics**
p. 1041	Addition of **32.4.1 (i) KYOCERA KINSEKI CORPORATION: Crystal resonator (T_A = –10 to +70°C)**
p. 1043	Modification of **32.4.3 PLL characteristics**
pp. 1047, 1048	Modification of **32.6.3 Supply current**
p. 1064	Modification of **32.9 (2) Serial write operation characteristics**
p. 1069	Addition of **CHAPTER 34 RECOMMENDED SOLDERING CONDITIONS**
p. 1103	Addition of **APPENDIX D REVISION HISTORY**

For further information,
please contact:

NEC Electronics Corporation
1753, Shimonumabe, Nakahara-ku,
Kawasaki, Kanagawa 211-8668,
Japan
Tel: 044-435-5111
http://www.necel.com/

[America]

NEC Electronics America, Inc.
2880 Scott Blvd.
Santa Clara, CA 95050-2554, U.S.A.
Tel: 408-588-6000
800-366-9782
http://www.am.necel.com/

[Europe]

NEC Electronics (Europe) GmbH
Arcadiastrasse 10
40472 Düsseldorf, Germany
Tel: 0211-65030
http://www.eu.necel.com/

Hanover Office
Podbielskistrasse 166 B
30177 Hannover
Tel: 0 511 33 40 2-0

Munich Office
Werner-Eckert-Strasse 9
81829 München
Tel: 0 89 92 10 03-0

Stuttgart Office
Industriestrasse 3
70565 Stuttgart
Tel: 0 711 99 01 0-0

United Kingdom Branch
Cygnus House, Sunrise Parkway
Linford Wood, Milton Keynes
MK14 6NP, U.K.
Tel: 01908-691-133

Succursale Française
9, rue Paul Dautier, B.P. 52
78142 Velizy-Villacoublay Cédex
France
Tel: 01-3067-5800

Sucursal en España
Juan Esplandiu, 15
28007 Madrid, Spain
Tel: 091-504-2787

Tyskland Filial
Täby Centrum
Entrance S (7th floor)
18322 Täby, Sweden
Tel: 08 638 72 00

Filiale Italiana
Via Fabio Filzi, 25/A
20124 Milano, Italy
Tel: 02-667541

Branch The Netherlands
Steijgerweg 6
5616 HS Eindhoven
The Netherlands
Tel: 040 265 40 10

[Asia & Oceania]

NEC Electronics (China) Co., Ltd
7th Floor, Quantum Plaza, No. 27 ZhiChunLu Haidian
District, Beijing 100083, P.R.China
Tel: 010-8235-1155
http://www.cn.necel.com/

Shanghai Branch
Room 2509-2510, Bank of China Tower,
200 Yincheng Road Central,
Pudong New Area, Shanghai, P.R.China P.C:200120
Tel:021-5888-5400
http://www.cn.necel.com/

Shenzhen Branch
Unit 01, 39/F, Excellence Times Square Building,
No. 4068 Yi Tian Road, Futian District, Shenzhen,
P.R.China P.C:518048
Tel:0755-8282-9800
http://www.cn.necel.com/

NEC Electronics Hong Kong Ltd.
Unit 1601-1613, 16/F., Tower 2, Grand Century Place,
193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: 2886-9318
http://www.hk.necel.com/

NEC Electronics Taiwan Ltd.
7F, No. 363 Fu Shing North Road
Taipei, Taiwan, R. O. C.
Tel: 02-8175-9600
http://www.tw.necel.com/

NEC Electronics Singapore Pte. Ltd.
238A Thomson Road,
#12-08 Novena Square,
Singapore 307684
Tel: 6253-8311
http://www.sg.necel.com/

NEC Electronics Korea Ltd.
11F., Samik Lavied'or Bldg., 720-2,
Yeoksam-Dong, Kangnam-Ku,
Seoul, 135-080, Korea
Tel: 02-558-3737
http://www.kr.necel.com/

G0706